IBM ILOG Plant PowerOps Data Schema

This document describes the IBM® ILOG® Plant PowerOps Data Schema. The data schema consists of numerous tables which define how to input data to Plant PowerOps (PPO). The data includes items such as resources, materials, recipes, activities, and production orders.

Schema tables appear in the following table, organized by group or topic. Note that some groups contain a large number of tables, and so have their own topic page. For an alphabetical list of all tables, see Index of schema tables.

Each topic also has an Entity Relationship Diagram (ERD); select the topic title (General tables, Manufacturing Resources, and so on) to see the ERD.

Successful implementation of the schema requires proper use of *prototypes* and *instances*. Note that prototype tables (_PROTO) are within the *Recipes, Activities, and Modes* group, while the corresponding instance tables are in several other groups including *Production Orders*. For example, the recommended procedure is to model your manufacturing process using prototype tables (such as PPO_ACTIVITY_PROTO), and the corresponding instances (such as PPO_ACTIVITY) are stored in the corresponding *Production Orders* group.

List of Topics

All tables	Index of schema tables
General tables	Buckets, criterion weights, model, optimization profile, setting
Manufacturing Resources	Calendars, resources
Materials and Storage Units	Inventory, material, storage, units
Recipes, Activities, and Modes	Prototype modeling tables: Activities, modes, recipes, setups
Setup Times and Setup Costs	PPO_ACTIVITY_SETUP_STATE PPO_RESOURCE_SETUP_MODEL PPO_RESOURCE_SETUP_STATE PPO_SETUP_MATRIX
Cleanup constraints	PPO_RESOURCE_CLEANUP_STATE
Demands	PPO_DEMAND PPO_DEMAND_COMPATIBILITY PPO_DUE_DATE
Procurements	PPO_PROCUREMENT
Production Plans	PPO_PLANNED_DELIVERY PPO_PLANNED_PRODUCTION PPO_PLANNED_PRODUCTION_MODE
Production Orders	Production order instance tables: activity, mode, setups
Material Flows	Material flow tables: _ARC

You can use database files (*.mdb) or spreadsheet files (*.csv) for data input to PPO. The use of database files to model problem data is described in the *Data modeling tutorials and examples*. For information about using spreadsheet files, see Data input with csv files.

For detailed information about the data types used in the PPO Data Schema, see Data schema types.

Note that flexible fields can be added to predefined PPO Objects using the PROPERTY mechanism. To add a user-defined property, just name the column `PROPERTY_`*type*_*name* where *type* is `INT`, `FLOAT` or `STRING`, and *name* is any character string.

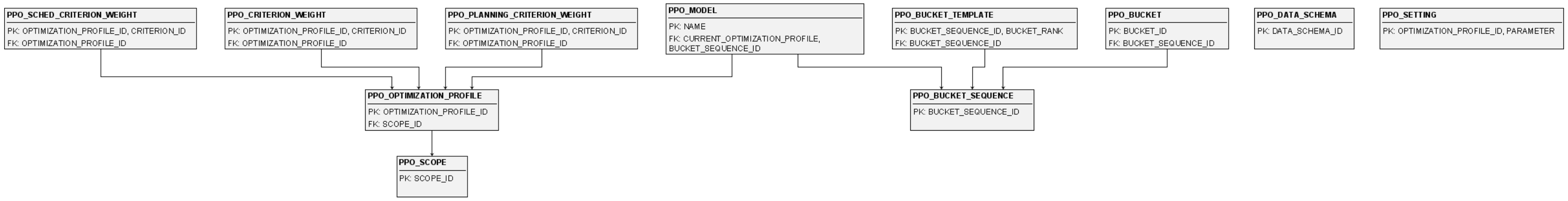
For information about data model checking in PPO see Data model checking.

Index of schema tables

- PPO_MODEL
- PPO_OPTIMIZATION_PROFILE
- PPO_PLANNED_DELIVERY
- PPO_PLANNED_PRODUCTION
- PPO_PLANNED_PRODUCTION_MODE
- PPO_PLANNING_CRITERION_WEIGHT
- PPO_PROCUREMENT
- PPO_PROCUREMENT_TO_DEMAND_ARC
- PPO_PROCUREMENT_TO_PROD_ARC
- PPO_PROD_ORDER_PLANNED_MODE
- PPO_PROD_PROD_COMPAT
- PPO_PROD_PROD_COMPAT_PROTO
- PPO_PROD_PROD_PRECED
- PPO_PROD_PROD_PRECED_PROTO
- PPO_PROD_SETUP_COMPAT
- PPO_PROD_SETUP_COMPAT_PROTO
- PPO_PROD_SETUP_PRECED
- PPO_PROD_SETUP_PRECED_PROTO
- PPO_PROD_TO_DEMAND_ARC
- PPO_PROD_TO_PROD_ARC
- PPO_PRODUCTION_ORDER
- PPO_PRODUCTION_ORDER_ACTIVITY
- PPO_QUALITY
- PPO_RECIPE
- PPO_RECIPE_FAMILY
- PPO_RECIPE_FAMILY_FILTER
- PPO_RECIPE_RECIPE_FAMILY
- PPO_RESOURCE
- PPO_RESOURCE_CAPACITY_COST
- PPO_RESOURCE_CAPACITY_COST_FCT
- PPO_RESOURCE_CLEANUP_STATE
- PPO_RESOURCE_CONNECTION
- PPO_RESOURCE_FAMILY

- PPO_RESOURCE_HIERARCHY
- PPO_RESOURCE_RESOURCE_FAMILY
- PPO_RESOURCE_SETUP_MODEL
- PPO_RESOURCE_SETUP_STATE
- PPO_SCHED_CRITERION_WEIGHT
- PPO_SCHEDULED_ACTIVITY
- PPO_SCHEDULED_SETUP_ACTIVITY
- PPO_SCOPE
- PPO_SECONDARY_RESOURCE
- PPO_SECONDARY_RESOURCE_PROTO
- PPO_SETTING
- PPO_SETUP_ACTIVITY
- PPO_SETUP_ACTIVITY_PROTO
- PPO_SETUP_MATRIX
- PPO_SETUP_MODE
- PPO_SETUP_MODE_PROTO
- PPO_SETUP_PROD_COMPAT
- PPO_SETUP_PROD_COMPAT_PROTO
- PPO_SETUP_PROD_PRECED
- PPO_SETUP_PROD_PRECED_PROTO
- PPO_SETUP_SECONDARY_RESOURCE
- PPO_SETUP_SECONDARY_RES_PROTO
- PPO_SETUP_SETUP_COMPAT
- PPO_SETUP_SETUP_COMPAT_PROTO
- PPO_SETUP_SETUP_PRECED_PROTO
- PPO_SETUP_SETUP_PRECED
- PPO_SPANNING
- PPO_SPANNING_PROTO
- PPO_STORAGE_UNIT
- PPO_STORAGE_UNIT_MATERIAL
- PPO_UNIT

Return to Data Schema Home Page.

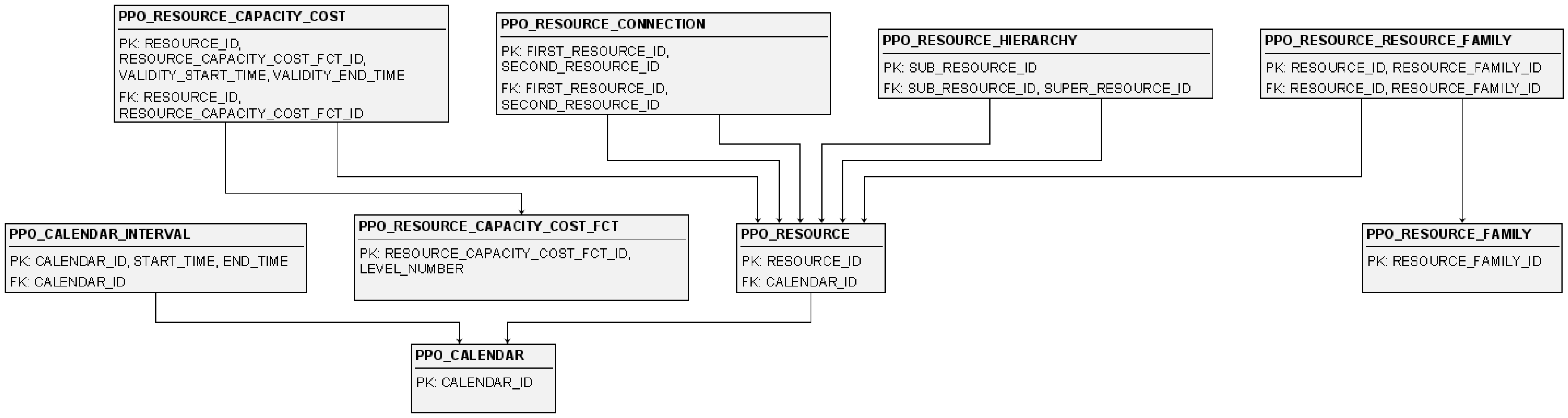

PPO_SCHED_CRITERION_WEIGHT
PK: OPTIMIZATION_PROFILE_ID, CRITERION_ID
FK: OPTIMIZATION_PROFILE_ID
PPO_CRITERION_WEIGHT
PK: OPTIMIZATION_PROFILE_ID, CRITERION_ID
FK: OPTIMIZATION_PROFILE_ID
PPO_PLANNING_CRITERION_WEIGHT
PK: OPTIMIZATION_PROFILE_ID, CRITERION_ID
FK: OPTIMIZATION_PROFILE_ID
PPO_MODEL
PK: NAME
FK: CURRENT_OPTIMIZATION_PROFILE, BUCKET_SEQUENCE_ID
PPO_BUCKET_TEMPLATE
PK: BUCKET_SEQUENCE_ID, BUCKET_RANK
FK: BUCKET_SEQUENCE_ID
PPO_BUCKET
PK: BUCKET_ID
FK: BUCKET_SEQUENCE_ID
PPO_DATA_SCHEMA
PK: DATA_SCHEMA_ID
PPO_SETTING
PK: OPTIMIZATION_PROFILE_ID, PARAMETER
PPO_OPTIMIZATION_PROFILE
PK: OPTIMIZATION_PROFILE_ID
FK: SCOPE_ID
PPO_BUCKET_SEQUENCE
PK: BUCKET_SEQUENCE_ID
PPO_SCOPE
PK: SCOPE_ID

General tables

This page lists the general data schema tables: Buckets, criterion weights, model, optimization profile, and setting tables.

- PPO_BUCKET
- PPO_BUCKET_SEQUENCE
- PPO_BUCKET_TEMPLATE
- PPO_CRITERION_WEIGHT
- PPO_DATA_SCHEMA
- PPO_MODEL
- PPO_OPTIMIZATION_PROFILE
- PPO_PLANNING_CRITERION_WEIGHT
- PPO_SCHED_CRITERION_WEIGHT
- PPO_SCOPE
- PPO_SETTING

Return to Data Schema Home Page.

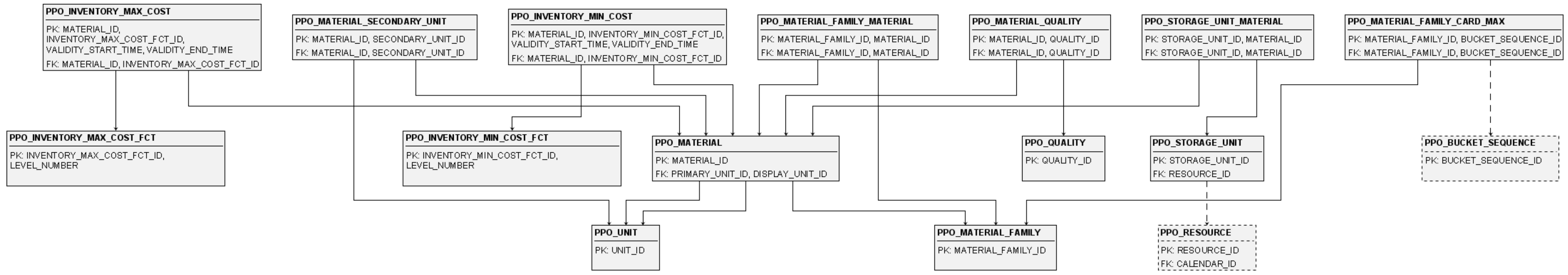

PPO_RESOURCE_CAPACITY_COST
PK: RESOURCE_ID, RESOURCE_CAPACITY_COST_FCT_ID, VALIDITY_START_TIME, VALIDITY_END_TIME
FK: RESOURCE_ID, RESOURCE_CAPACITY_COST_FCT_ID
PPO_RESOURCE_CONNECTION
PK: FIRST_RESOURCE_ID, SECOND_RESOURCE_ID
FK: FIRST_RESOURCE_ID, SECOND_RESOURCE_ID
PPO_RESOURCE_HIERARCHY
PK: SUB_RESOURCE_ID
FK: SUB_RESOURCE_ID, SUPER_RESOURCE_ID
PPO_RESOURCE_RESOURCE_FAMILY
PK: RESOURCE_ID, RESOURCE_FAMILY_ID
FK: RESOURCE_ID, RESOURCE_FAMILY_ID
PPO_CALENDAR_INTERVAL
PK: CALENDAR_ID, START_TIME, END_TIME
FK: CALENDAR_ID
PPO_RESOURCE_CAPACITY_COST_FCT
PK: RESOURCE_CAPACITY_COST_FCT_ID, LEVEL_NUMBER
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID
PPO_RESOURCE_FAMILY
PK: RESOURCE_FAMILY_ID
PPO_CALENDAR
PK: CALENDAR_ID

Manufacturing resource tables

This page lists the manufacturing resource data schema tables: Calendars and resources.

- PPO_CALENDAR
- PPO_CALENDAR_INTERVAL
- PPO_RESOURCE
- PPO_RESOURCE_CAPACITY_COST
- PPO_RESOURCE_CAPACITY_COST_FCT
- PPO_RESOURCE_CONNECTION
- PPO_RESOURCE_FAMILY
- PPO_RESOURCE_HIERARCHY
- PPO_RESOURCE_RESOURCE_FAMILY

Return to Data Schema Home Page.

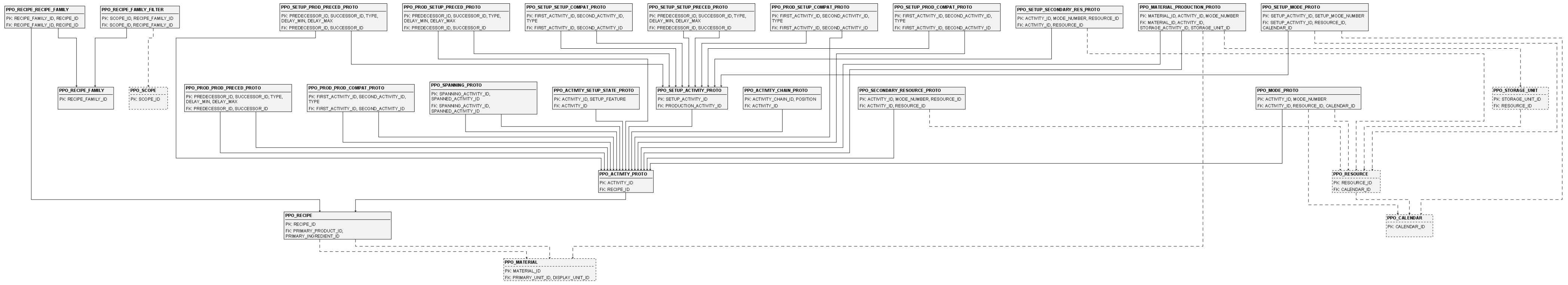

PPO_INVENTORY_MAX_COST
PK: MATERIAL_ID, INVENTORY_MAX_COST_FCT_ID, VALIDITY_START_TIME, VALIDITY_END_TIME
FK: MATERIAL_ID, INVENTORY_MAX_COST_FCT_ID
PPO_MATERIAL_SECONDARY_UNIT
PK: MATERIAL_ID, SECONDARY_UNIT_ID
FK: MATERIAL_ID, SECONDARY_UNIT_ID
PPO_INVENTORY_MIN_COST
PK: MATERIAL_ID, INVENTORY_MIN_COST_FCT_ID, VALIDITY_START_TIME, VALIDITY_END_TIME
FK: MATERIAL_ID, INVENTORY_MIN_COST_FCT_ID
PPO_MATERIAL_FAMILY_MATERIAL
PK: MATERIAL_FAMILY_ID, MATERIAL_ID
FK: MATERIAL_FAMILY_ID, MATERIAL_ID
PPO_MATERIAL_QUALITY
PK: MATERIAL_ID, QUALITY_ID
FK: MATERIAL_ID, QUALITY_ID
PPO_STORAGE_UNIT_MATERIAL
PK: STORAGE_UNIT_ID, MATERIAL_ID
FK: STORAGE_UNIT_ID, MATERIAL_ID
PPO_MATERIAL_FAMILY_CARD_MAX
PK: MATERIAL_FAMILY_ID, BUCKET_SEQUENCE_ID
FK: MATERIAL_FAMILY_ID, BUCKET_SEQUENCE_ID
PPO_INVENTORY_MAX_COST_FCT
PK: INVENTORY_MAX_COST_FCT_ID, LEVEL_NUMBER
PPO_INVENTORY_MIN_COST_FCT
PK: INVENTORY_MIN_COST_FCT_ID, LEVEL_NUMBER
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID
PPO_QUALITY
PK: QUALITY_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_BUCKET_SEQUENCE
PK: BUCKET_SEQUENCE_ID
PPO_UNIT
PK: UNIT_ID
PPO_MATERIAL_FAMILY
PK: MATERIAL_FAMILY_ID
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID

Materials and storage units

This page lists the materials and storage units schema tables: Inventory, material, storage, and units.

- PPO_INVENTORY_MAX_COST
- PPO_INVENTORY_MAX_COST_FCT
- PPO_INVENTORY_MIN_COST
- PPO_INVENTORY_MIN_COST_FCT
- PPO_MATERIAL
- PPO_MATERIAL_FAMILY
- PPO_MATERIAL_FAMILY_CARD_MAX
- PPO_MATERIAL_FAMILY_MATERIAL
- PPO_MATERIAL_QUALITY
- PPO_MATERIAL_SECONDARY_UNIT
- PPO_QUALITY
- PPO_STORAGE_UNIT
- PPO_STORAGE_UNIT_MATERIAL
- PPO_UNIT

Return to Data Schema Home Page.

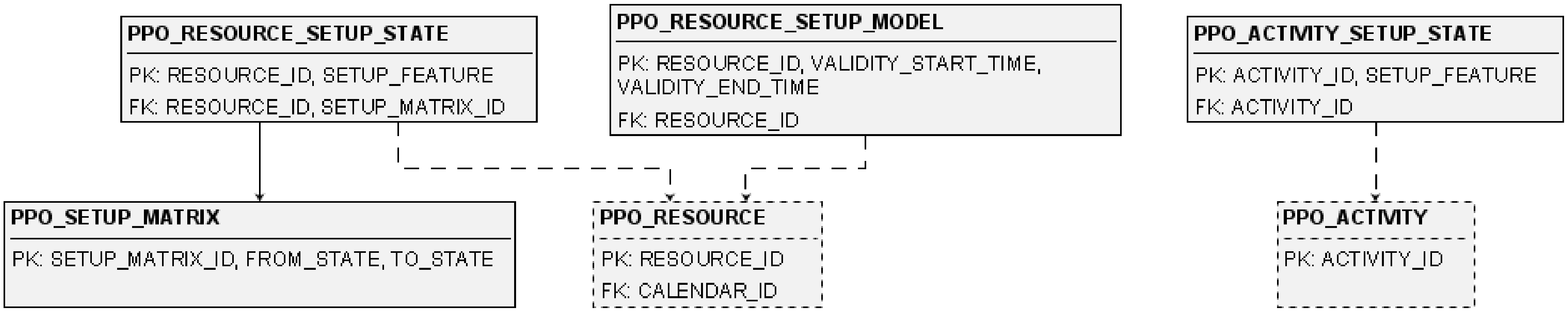

PPO_RECIPE_RECIPE_FAMILY
PK: RECIPE_FAMILY_ID, RECIPE_ID
FK: RECIPE_FAMILY_ID, RECIPE_ID
PPO_RECIPE_FAMILY_FILTER
PK: SCOPE_ID, RECIPE_FAMILY_ID
FK: SCOPE_ID, RECIPE_FAMILY_ID
PPO_SETUP_PROD_PRECED_PROTO
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_PROD_SETUP_PRECED_PROTO
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_SETUP_SETUP_COMPAT_PROTO
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_SETUP_PRECED_PROTO
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_PROD_SETUP_COMPAT_PROTO
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_PROD_COMPAT_PROTO
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_SECONDARY_RES_PROTO
PK: ACTIVITY_ID, MODE_NUMBER, RESOURCE_ID
FK: ACTIVITY_ID, RESOURCE_ID
PPO_MATERIAL_PRODUCTION_PROTO
PK: MATERIAL_ID, ACTIVITY_ID, MODE_NUMBER
FK: MATERIAL_ID, ACTIVITY_ID, STORAGE_ACTIVITY_ID, STORAGE_UNIT_ID
PPO_SETUP_MODE_PROTO
PK: SETUP_ACTIVITY_ID, SETUP_MODE_NUMBER
FK: SETUP_ACTIVITY_ID, RESOURCE_ID, CALENDAR_ID
PPO_RECIPE_FAMILY
PK: RECIPE_FAMILY_ID
PPO_SCOPE
PK: SCOPE_ID
PPO_PROD_PROD_PRECED_PROTO
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_PROD_PROD_COMPAT_PROTO
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SPANNING_PROTO
PK: SPANNING_ACTIVITY_ID, SPANNED_ACTIVITY_ID
FK: SPANNING_ACTIVITY_ID, SPANNED_ACTIVITY_ID
PPO_ACTIVITY_SETUP_STATE_PROTO
PK: ACTIVITY_ID, SETUP_FEATURE
FK: ACTIVITY_ID
PPO_SETUP_ACTIVITY_PROTO
PK: SETUP_ACTIVITY_ID
FK: PRODUCTION_ACTIVITY_ID
PPO_ACTIVITY_CHAIN_PROTO
PK: ACTIVITY_CHAIN_ID, POSITION
FK: ACTIVITY_ID
PPO_SECONDARY_RESOURCE_PROTO
PK: ACTIVITY_ID, MODE_NUMBER, RESOURCE_ID
FK: ACTIVITY_ID, RESOURCE_ID
PPO_MODE_PROTO
PK: ACTIVITY_ID, MODE_NUMBER
FK: ACTIVITY_ID, RESOURCE_ID, CALENDAR_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_ACTIVITY_PROTO
PK: ACTIVITY_ID
FK: RECIPE_ID
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID
PPO_CALENDAR
PK: CALENDAR_ID
PPO_RECIPE
PK: RECIPE_ID
FK: PRIMARY_PRODUCT_ID, PRIMARY_INGREDIENT_ID
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID

Recipes, activities, and modes

This page lists the prototype modeling tables: Activities, modes, recipes, setups.

- PPO_ACTIVITY_CHAIN_PROTO
- PPO_ACTIVITY_PROTO
- PPO_ACTIVITY_SETUP_STATE_PROTO
- PPO_MATERIAL_PRODUCTION_PROTO
- PPO_MODE_PROTO
- PPO_PROD_PROD_COMPAT_PROTO
- PPO_PROD_PROD_PRECED_PROTO
- PPO_PROD_SETUP_COMPAT_PROTO
- PPO_PROD_SETUP_PRECED_PROTO
- PPO_RECIPE
- PPO_RECIPE_FAMILY
- PPO_RECIPE_FAMILY_FILTER
- PPO_RECIPE_RECIPE_FAMILY
- PPO_SECONDARY_RESOURCE_PROTO
- PPO_SETUP_ACTIVITY_PROTO
- PPO_SETUP_MODE_PROTO
- PPO_SETUP_PROD_COMPAT_PROTO
- PPO_SETUP_PROD_PRECED_PROTO
- PPO_SETUP_SECONDARY_RES_PROTO
- PPO_SETUP_SETUP_COMPAT_PROTO
- PPO_SETUP_SETUP_PRECED_PROTO
- PPO_SPANNING_PROTO

Return to Data Schema Home Page.


PPO_RESOURCE_SETUP_STATE
PK: RESOURCE_ID, SETUP_FEATURE
FK: RESOURCE_ID, SETUP_MATRIX_ID
PPO_RESOURCE_SETUP_MODEL
PK: RESOURCE_ID, VALIDITY_START_TIME, VALIDITY_END_TIME
FK: RESOURCE_ID
PPO_ACTIVITY_SETUP_STATE
PK: ACTIVITY_ID, SETUP_FEATURE
FK: ACTIVITY_ID
PPO_SETUP_MATRIX
PK: SETUP_MATRIX_ID, FROM_STATE, TO_STATE
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID
PPO_ACTIVITY
PK: ACTIVITY_ID

PPO_ACTIVITY_SETUP_STATE

Synopsis

`PPO_ACTIVITY_SETUP_STATE,ACTIVITY_ID,SETUP_FEATURE,SETUP_STATE`

Topic

Setup Times and Setup Costs

Description

The PPO_ACTIVITY_SETUP_STATE table defines the setup states required by each activity instance for each setup feature.

Primary keys

ACTIVITY_ID,SETUP_FEATURE

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of the activity.	id		mandatory
SETUP_FEATURE	The identifier of the feature. If you don't need the notion of feature, use `NoFeature`.	id		"NoFeature"
SETUP_STATE	The state required by the activity for the setup feature.	id		mandatory

PPO_RESOURCE_SETUP_MODEL

Synopsis

`PPO_RESOURCE_SETUP_MODEL,RESOURCE_ID,VALIDITY_START_TIME,VALIDITY_END_TIME,SETUP_MODEL`

Topic

Setup Times and Setup Costs

Description

The table PPO_RESOURCE_SETUP_MODEL defines which approximation of the setup model the planning engine must take into account for each resource in a given time interval. `NoSetup` means that setup activities are not taken into account. `PerBucketPerRecipe` means that fixed capacity requirements including setups are counted independently for each recipe and each bucket. `CrossBucketPerRecipe` is similar except that if a recipe extends from one bucket to the next, the setups are not repeated in the second time bucket. `PerBucketPerFeature` means that fixed capacity requirements including setups are counted independently for each setup feature and bucket. `CrossBucketPerFeature` is similar except that if a setup feature continues from one bucket to the next, the setups are not repeated in the second time bucket.

Primary keys

RESOURCE_ID,VALIDITY_START_TIME,VALIDITY_END_TIME

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	The identifier of the resource.	id		mandatory
VALIDITY_START_TIME	The start time of the period over which the given setup model applies.	integer		-INF
VALIDITY_END_TIME	The end time of the period over which the given setup model applies.	integer		+INF
SETUP_MODEL	The name of the setup model approximation. The different approximation models are: `NoSetup`, `PerBucketPerRecipe`, `CrossBucketPerRecipe`, `PerBucketPerFeature`, `CrossBucketPerFeature`.	id		"NoSetup"

PPO_RESOURCE_SETUP_STATE

Synopsis

`PPO_RESOURCE_SETUP_STATE,RESOURCE_ID,SETUP_FEATURE,SETUP_MATRIX_ID,INITIAL_SETUP_STATE, FINAL_SETUP_STATE`

Topic

Setup Times and Setup Costs

Description

The PPO_RESOURCE_SETUP_STATE table defines the setup matrix as well as the initial and final setup states of the resource for each supported setup feature.

Primary keys

RESOURCE_ID,SETUP_FEATURE

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	The identifier of the resource.	id		mandatory
SETUP_FEATURE	The identifier of the feature. If you don't need the notion of feature, use `NoFeature`.	id		"NoFeature"
SETUP_MATRIX_ID	The identifier of the setup matrix.	id		mandatory
INITIAL_SETUP_STATE	The initial state of the resource (NULL if unknown).	id		NULL
FINAL_SETUP_STATE	The desired final state of the resource at the end of horizon (NULL if no requirement applies).	id		NULL

PPO_SETUP_MATRIX

Synopsis

`PPO_SETUP_MATRIX,SETUP_MATRIX_ID,FROM_STATE,TO_STATE,SETUP_TIME,SETUP_COST,CLEANUP`

Topic

Setup Times and Setup Costs

Description

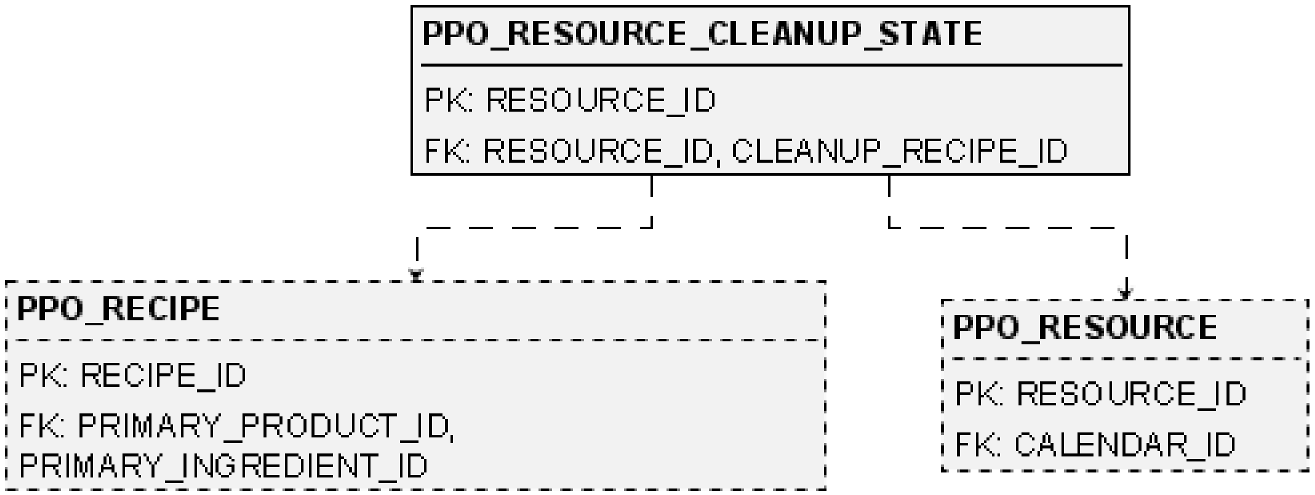
The PPO_SETUP_MATRIX table is used to store the setup time and cost incurred when two production activities follow each other on the same resource. PPO supports the notion of multiple setup features (cleaning, mold changing, and so forth). An activity may require several features to each be in a specific state. The setup times and costs incurred by each feature are additive. Each line in the table corresponds to a transition between two values of a feature. The CLEANUP field states if a cleanup activity is required for this transition (PPO_ACTIVITY table with CLEANUP_STATUS set to `Cleaning`).

Primary keys

SETUP_MATRIX_ID,FROM_STATE,TO_STATE

Field descriptions

Field Name	Description	Type	Range	Default
SETUP_MATRIX_ID	This data is the mandatory unique identifier of the setup matrix instance.	id		mandatory
FROM_STATE	The previous state of the feature for the transition under consideration. An unknown previous state is encoded as the NULL identifier. The values of setup time and setup cost given from the NULL state override the values inferred by the planning engine as setup approximations. Only the planning engine uses these values. Note that these values can be retrieved in the object model using `IloMSSetupMatrix::getDefaultSetupCost` and `IloMSSetupMatrix::getDefaultSetupTime`.	id		mandatory
TO_STATE	The next state of the feature for the transition under consideration.	id		mandatory
SETUP_TIME	The time incurred for switching between the two states of the specified feature. Note that setup times are not required to be smaller than bucket durations. However if a setup time is greater than the time bucket, then the planning engine will use a group of buckets to implement the associated resource capacity constraint.	integer		0
SETUP_COST	The cost incurred for switching between the two states of the specified feature.	double		0.
CLEANUP	If true, this transition requires a major clean up.	boolean		0


PPO_RESOURCE_CLEANUP_STATE
PK: RESOURCE_ID
FK: RESOURCE_ID, CLEANUP_RECIPE_ID
PPO_RECIPE
PK: RECIPE_ID
FK: PRIMARY_PRODUCT_ID, PRIMARY_INGREDIENT_ID
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID

PPO_RESOURCE_CLEANUP_STATE

Synopsis

`PPO_RESOURCE_CLEANUP_STATE,RESOURCE_ID,CLEANUP_RECIPE_ID,MAX_TIME_BEFORE_CLEANUP, MAX_NB_BATCHES_BEFORE_CLEANUP,MAX_IDLE_TIME_BEFORE_CLEANUP,TIME_OF_LAST_CLEANUP, NB_BATCHES_SINCE_LAST_CLEANUP`

Topic

Cleanup constraints

Description

The PPO_RESOURCE_CLEANUP_STATE table defines resource cleanup requirements. Each row states how often (in time or batches) the resource with the given RESOURCE_ID must be cleaned. It also defines the time at which the last cleanup of this resource was performed (TIME_OF_LAST_CLEANUP) and the number of batches executed since the last cleanup (NB_BATCHES_SINCE_LAST_CLEANUP). The recipe with the given CLEANUP_RECIPE_ID must have one and only one activity, and that activity must have a cleanup status of `Cleaning`. The activity must also have one and only one mode (in the PPO_MODE_PROTO table) executable on the resource under consideration, defining the time and cost, and possibly the secondary resources, required to clean the resource with the given RESOURCE_ID.

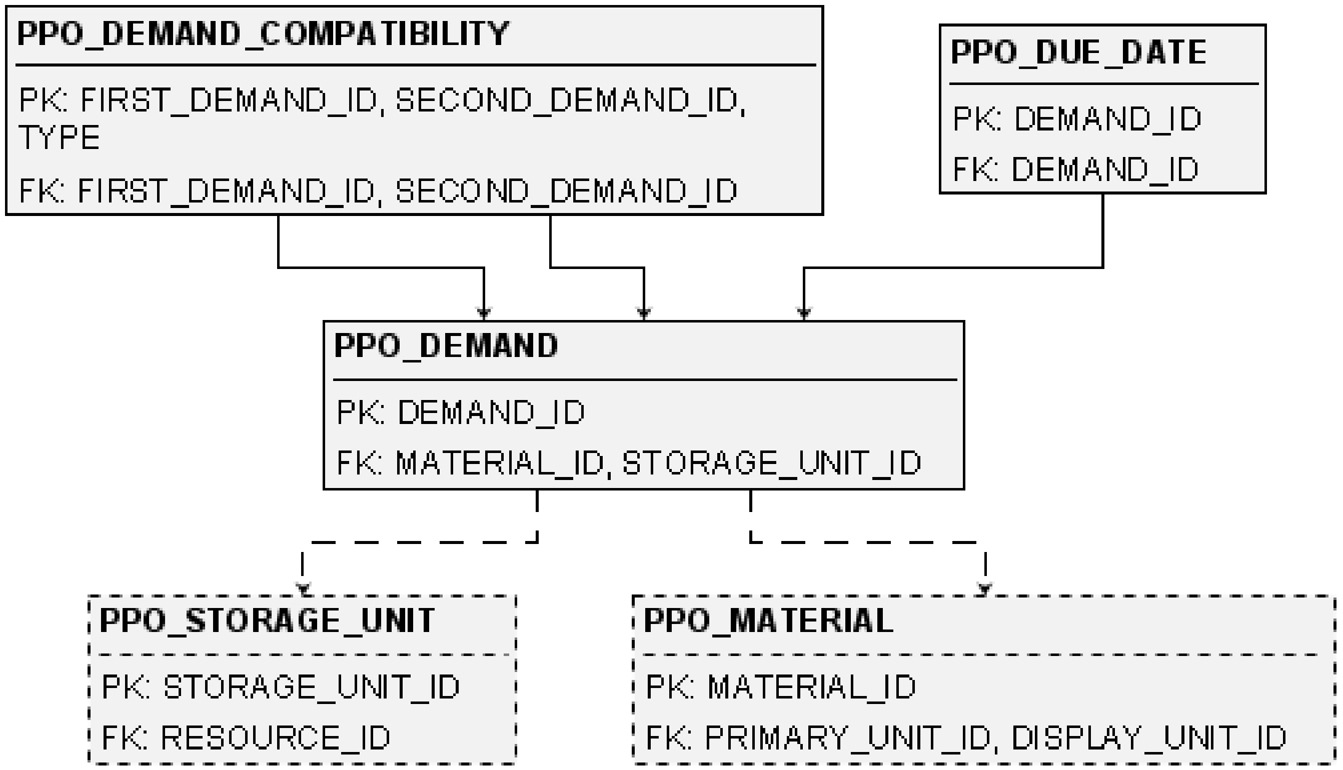
Primary keys

RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	The identifier of the resource.	id		mandatory
CLEANUP_RECIPE_ID	The identifier of the recipe that is used to clean the resource.	id		mandatory
MAX_TIME_BEFORE_CLEANUP	The maximal number of time units between two cleanups. After this elapsed time, a cleanup is required before any subsequent resource usage.	integer		+INF
MAX_NB_BATCHES_BEFORE_CLEANUP	The maximal number of batches between two cleanups. After this number of processed batches, a cleanup is required before any subsequent resource usage.	integer		+INF
MAX_IDLE_TIME_BEFORE_CLEANUP	The maximal number of time units that a resource can remain idle between two cleanups. After this elapsed idle time, a cleanup is required before any subsequent resource usage.	integer		+INF

TIME_OF_LAST_CLEANUP	The end time of the last cleanup executed before the START_MIN of the resource. The default is used when the cleanup end time is not known, in which case the START_MIN of the resource will apply.	integer	+INF
NB_BATCHES_SINCE_LAST_CLEANUP	The number of batches processed since the last cleanup.	integer	0


PPO_DEMAND_COMPATIBILITY
PK: FIRST_DEMAND_ID, SECOND_DEMAND_ID, TYPE
FK: FIRST_DEMAND_ID, SECOND_DEMAND_ID
PPO_DUE_DATE
PK: DEMAND_ID
FK: DEMAND_ID
PPO_DEMAND
PK: DEMAND_ID
FK: MATERIAL_ID, STORAGE_UNIT_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID

PPO_DEMAND

Synopsis

```
PPO_DEMAND,DEMAND_ID,NAME,MATERIAL_ID,STORAGE_UNIT_ID,QUANTITY,REVENUE,DELIVERY_START_MIN,
DELIVERY_END_MAX,NON_DELIVERY_VARIABLE_COST,MAX_NUMBER_OF_PEGGING_ARCS,REMAINING_SHELF_LIFE,
PROMISED
```

Topic

Demands

Description

The PPO_DEMAND table represents the request of a certain amount of material deliverable in a time window, with an optional preferred due date.

Primary keys

DEMAND_ID

Field descriptions

Field Name	Description	Type	Range	Default
DEMAND_ID	This data is the mandatory unique identifier of the demand.	id		mandatory
NAME	Optional name.	string		""
MATERIAL_ID	The requested material.	id		mandatory
STORAGE_UNIT_ID	Optional field. The storage unit from which the material must be shipped.	id		NULL
QUANTITY	The quantity requested.	double		mandatory
REVENUE	The revenue per unit of satisfied demand.	double		0.
DELIVERY_START_MIN	The earliest time at which the delivery of the demand can start.	integer		-INF
DELIVERY_END_MAX	The latest time at which the delivery of the demand can end.	integer		+INF
NON_DELIVERY_VARIABLE_COST	This data sets a penalty for nondelivery per unit of demand.	double		0.
MAX_NUMBER_OF_PEGGING_ARCS	Ensures that the number of incoming pegging arcs does not exceed a maximal value.	integer		+INF

REMAINING_SHELF_LIFE	The minimal shelf life of the material that must remain after the delivery time of the demand. The delivery time of a demand is the maximum between the end of the production of the material for the demand and the delivery start min of the demand	integer	0
PROMISED	If this field is true, the demand is already promised to a customer; such a demand is not included in the "Available-To-Promise" (ATP) calculation. Note that the fact that a demand is promised is not enough to make the demand delivery mandatory for the optimizer: An infinite nondelivery cost must still be set in order to make the delivery of this demand a hard constraint. The promised demand is used only in the ATP calculation for master production scheduling.	boolean	0

PPO_DEMAND_COMPATIBILITY

Synopsis

```
PPO_DEMAND_COMPATIBILITY,FIRST_DEMAND_ID,SECOND_DEMAND_ID,TYPE
```

Topic

Demands

Description

The PPO_DEMAND_COMPATIBILITY table places logical constraints on demand deliveries in the planning model. This table is taken into account only in the Planning module and is usable only with unsplittable demand. Unsplittable demand is specified by putting the max number of pegging arcs of a demand to one.

Primary keys

FIRST_DEMAND_ID,SECOND_DEMAND_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_DEMAND_ID	A demand.	id		mandatory
SECOND_DEMAND_ID	Another demand.	id		mandatory
TYPE	The type of demand compatibility constraint. The types are: `SameCoveringStatus` or `CoveredImpliesCovered`. `CoveredImpliesCovered` means that if *demand1* is fully satisfied then *demand2* must also be fully satisfied. `SameCoveringStatus` means that *demand1* and *demand2* are either both fully satisfied or both unsatisfied.	id		mandatory

PPO_DUE_DATE

Synopsis

```
PPO_DUE_DATE,DEMAND_ID,DUE_TIME,EARLINESS_VARIABLE_COST,TARDINESS_VARIABLE_COST,
EARLINESS_FIXED_COST,TARDINESS_FIXED_COST
```

Topic

Demands

Description

The PPO_DUE_DATE table is used to define due dates that are attached to demands. Four cost penalties can be defined for any given due date: An earliness fixed cost, an earliness variable cost, a tardiness fixed cost, and a tardiness variable cost.

The earliness fixed cost is a fixed price to be paid if the material is delivered early. For example, it represents the cost of putting a finished product into storage in a warehouse, rather than shipping it directly to the customer. The earliness variable cost is the price to be paid (in addition to the earliness fixed cost) per time unit that the material is delivered early. For example, it represents the cost of additional space in the warehouse to store the finished product that was produced early. Likewise with tardiness costs: The tardiness fixed cost is a fixed price to be paid if the material is delivered late, and the tardiness variable cost is the price to be paid (in addition to the tardiness fixed cost) per time unit late.

For demand due dates, both variable and fixed costs are multiplied by the quantity of material that is delivered before or after the due date. A demand can have only one due date. Note that the default for all costs is zero, so by default there is no penalty for delivering the demand early as the earliness fixed and variable costs are null, and there is no penalty for delivering the demand late as the tardiness fixed and variable costs are null.

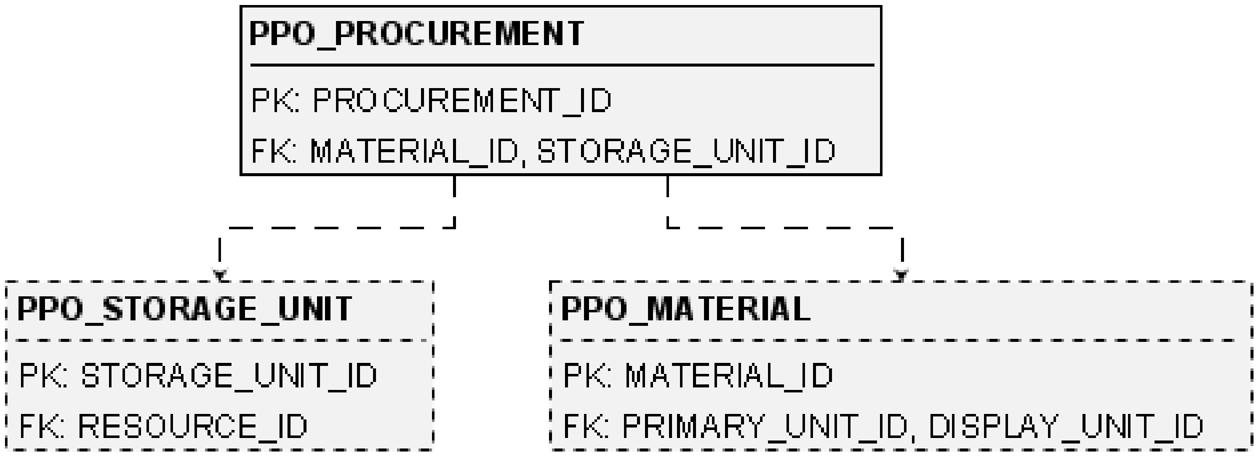
Primary keys

DEMAND_ID

Field descriptions

Field Name	Description	Type	Range	Default
DEMAND_ID	The identifier of the demand with which the due date is associated.	id		mandatory
DUE_TIME	The time at which the demand is due. This time is expressed with respect to the DATE_ORIGIN and TIME_UNIT of the model.	integer		mandatory
EARLINESS_VARIABLE_COST	This data defines the variable cost used to compute the earliness cost. This variable cost is multiplied by the quantity of material delivered early and by the difference between the DUE_TIME and the material delivery time.	double		0.

TARDINESS_VARIABLE_COST	This data defines the variable cost used to compute the tardiness cost. This variable cost is multiplied by the quantity of material delivered late and by the difference between the material delivery time and the DUE_TIME.	double	0.
EARLINESS_FIXED_COST	This data defines the fixed cost used to compute the earliness cost. This fixed cost is multiplied by the quantity of material delivered early.	double	0.
TARDINESS_FIXED_COST	This data defines the fixed cost used to compute the tardiness cost. This fixed cost is multiplied by the quantity of material delivered late.	double	0.


PPO_PROCUREMENT
PK: PROCUREMENT_ID
FK: MATERIAL_ID, STORAGE_UNIT_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID

PPO_PROCUREMENT

Synopsis

```
PPO_PROCUREMENT,PROCUREMENT_ID,NAME,MATERIAL_ID,STORAGE_UNIT_ID,QUANTITY,RECEIPT_TIME,
PRODUCTION_TIME,OVERRIDDEN_SHELF_LIFE,OVERRIDDEN_MATURITY
```

Topic

Procurements

Description

The PPO_PROCUREMENT table represents materials procured from outside the plant, and it can also be used to give an age to stock elements. Each procurement corresponds to a given quantity of material with an age according to its production date. A procurement may override the default shelf life and maturity characteristics of the material.

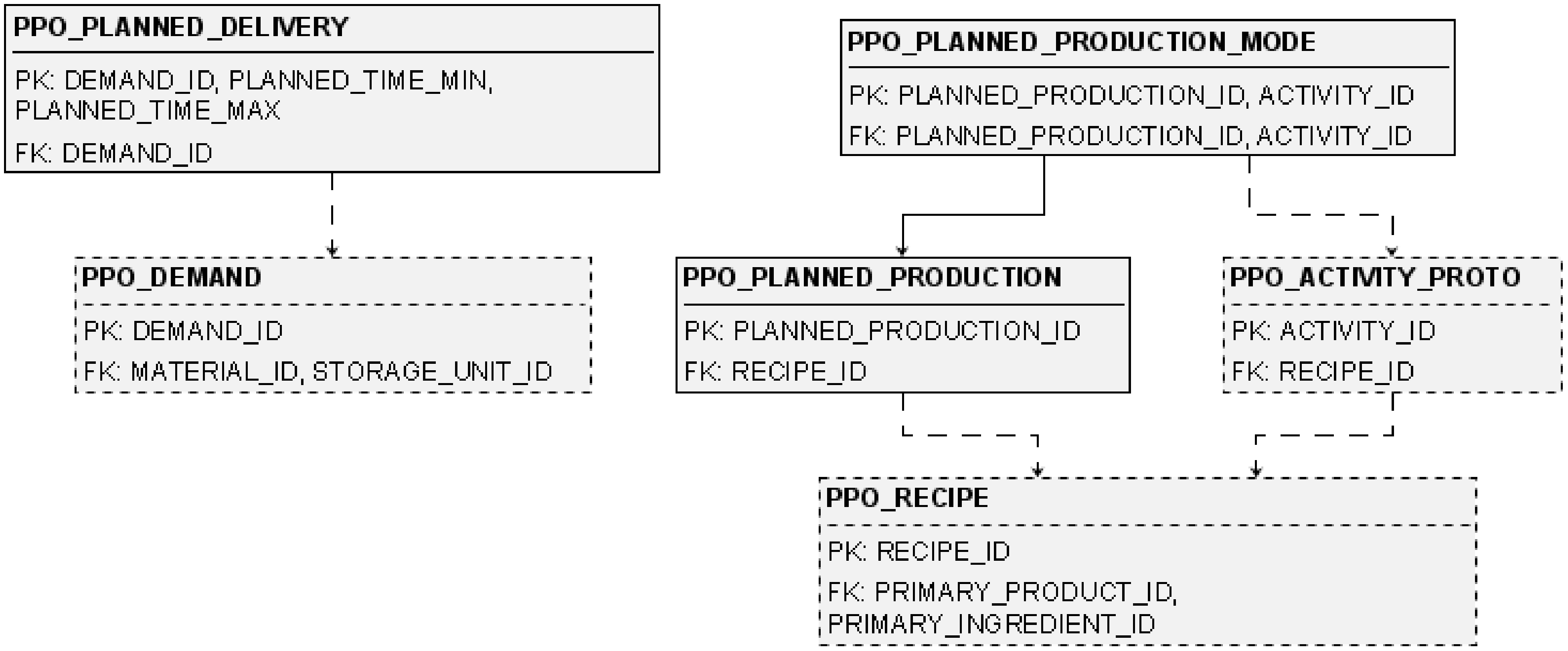
Primary keys

PROCUREMENT_ID

Field descriptions

Field Name	Description	Type	Range	Default
PROCUREMENT_ID	This data is the mandatory unique identifier of the instance.	id		mandatory
NAME	Optional name.	string		""
MATERIAL_ID	The procured material.	id		mandatory
STORAGE_UNIT_ID	Optional field. The storage unit in which the material must be received.	id		NULL
QUANTITY	The quantity procured.	double		mandatory
RECEIPT_TIME	The time at which the procurement is received.	integer		mandatory
PRODUCTION_TIME	The time at which the material was produced. This piece of information is used to compute when the material is usable if it is subjected to shelf life or maturity constraints. If this piece of information is not given, the procurement will be considered mature but possibly expired. It is not needed if there is no shelf life or maturity constraint.	integer		-INF
OVERRIDDEN_SHELF_LIFE	Optional. The overridden shelf life of the material provided by this procurement. If the value is **-1**, the standard shelf life of the material applies.	integer		-1

OVERRIDDEN_MATURITY	Optional. The overridden maturation time of the material provided by this procurement. If the value is **-1**, the standard maturity of the material applies.	integer	-1


PPO_PLANNED_DELIVERY
PK: DEMAND_ID, PLANNED_TIME_MIN, PLANNED_TIME_MAX
FK: DEMAND_ID
PPO_DEMAND
PK: DEMAND_ID
FK: MATERIAL_ID, STORAGE_UNIT_ID
PPO_PLANNED_PRODUCTION_MODE
PK: PLANNED_PRODUCTION_ID, ACTIVITY_ID
FK: PLANNED_PRODUCTION_ID, ACTIVITY_ID
PPO_PLANNED_PRODUCTION
PK: PLANNED_PRODUCTION_ID
FK: RECIPE_ID
PPO_ACTIVITY_PROTO
PK: ACTIVITY_ID
FK: RECIPE_ID
PPO_RECIPE
PK: RECIPE_ID
FK: PRIMARY_PRODUCT_ID, PRIMARY_INGREDIENT_ID

PPO_PLANNED_DELIVERY

Synopsis

```
PPO_PLANNED_DELIVERY,DEMAND_ID,PLANNED_TIME_MIN,PLANNED_TIME_MAX,QUANTITY,FIRM_QUANTITY_MIN,
FIRM_QUANTITY_MAX
```

Topic

Production Plans

Description

The PPO_PLANNED_DELIVERY table defines when a quantity of a specific demand is planned to be delivered. This quantity can be the total or a part of the demand quantity. Delivery means that the items exit from the inventory. Planned deliveries are in addition to any already fixed pegging arcs to the demand.

Primary keys

DEMAND_ID,PLANNED_TIME_MIN,PLANNED_TIME_MAX

Field descriptions

Field Name	Description	Type	Range	Default
DEMAND_ID	The identifier of a demand.	id		mandatory
PLANNED_TIME_MIN	An approximate minimal time at which the demand could be delivered.	integer		-INF
PLANNED_TIME_MAX	An approximate maximal time at which the demand is supposed to be delivered.	integer		+INF
QUANTITY	The quantity of the demand planned to be satisfied around the given planned time.	double		mandatory
FIRM_QUANTITY_MIN	When greater than `0`, This data specifies to the planning engine the minimum quantity to be shipped in the specified time bucket for the corresponding demand.	double		0.
FIRM_QUANTITY_MAX	When not equal to `+INF`, This data specifies to the planning engine the maximum quantity to be shipped in the specified time window for the corresponding demand.	double		+1.#INF

PPO_PLANNED_PRODUCTION

Synopsis

`PPO_PLANNED_PRODUCTION,PLANNED_PRODUCTION_ID,RECIPE_ID,PLANNED_TIME_MIN,PLANNED_TIME_MAX, BATCH_SIZE,PLANNED_NUMBER_OF_BATCHES,FIRM_BATCH_SIZE_MIN,FIRM_BATCH_SIZE_MAX`

Topic

Production Plans

Description

The PPO_PLANNED_PRODUCTION table defines which recipe executions are planned, approximately for when, and in what quantities. Recipe executions can be associated with more than one planning solution, thus representing alternative production plans; however, you must commit one of these planning solutions prior to proceeding with the batching.

Primary keys

PLANNED_PRODUCTION_ID

Field descriptions

Field Name	Description	Type	Range	Default
PLANNED_PRODUCTION_ID	This data is the mandatory unique identifier of the instance.	id		mandatory
RECIPE_ID	The identifier of a recipe.	id		mandatory
PLANNED_TIME_MIN	An approximate minimal time at which the recipe execution could start.	integer		-INF
PLANNED_TIME_MAX	An approximate maximal time at which the recipe execution is supposed to end.	integer		+INF
BATCH_SIZE	The quantity of the recipe planned to be executed around the given planned time.	double		mandatory
PLANNED_NUMBER_OF_BATCHES	The planned number of batches.	integer		0
FIRM_BATCH_SIZE_MIN	When greater than `0`, this data specifies the minimal batch size to the planning engine for executing the corresponding recipe in the corresponding time bucket.	double		0.
FIRM_BATCH_SIZE_MAX	When not equal to the default value of `+INF`, this data specifies maximal batch size to the planning engine for executing the corresponding recipe in the corresponding time bucket.	double		+1.#INF

PPO_PLANNED_PRODUCTION_MODE

Synopsis

`PPO_PLANNED_PRODUCTION_MODE,PLANNED_PRODUCTION_ID,ACTIVITY_ID,MODE_NUMBER`

Topic

Production Plans

Description

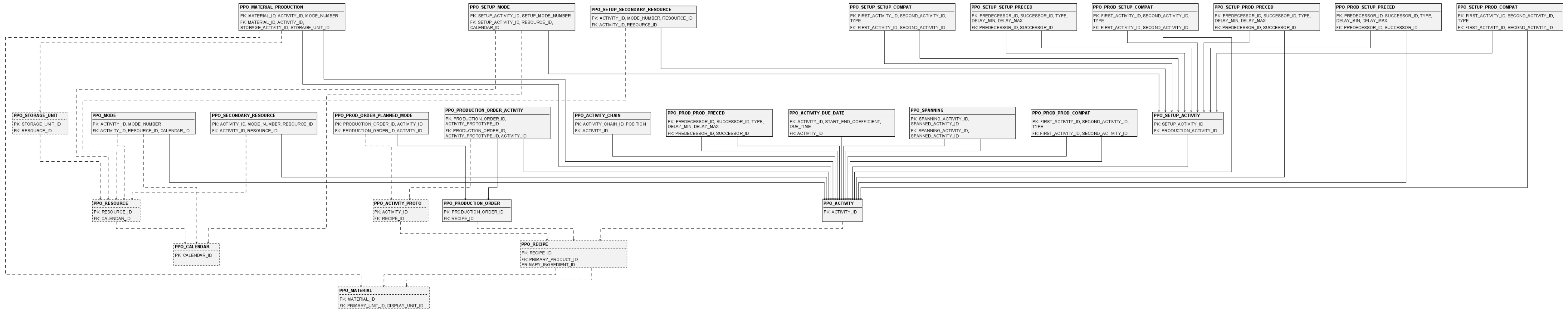
The PPO_PLANNED_PRODUCTION_MODE table defines which modes are used for a given planned production.

Primary keys

PLANNED_PRODUCTION_ID,ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
PLANNED_PRODUCTION_ID	This data is the mandatory identifier of the planned production instance.	id		mandatory
ACTIVITY_ID	The identifier of the activity prototype to which the planned production is associated.	id		mandatory
MODE_NUMBER	The index of the mode prototype for the activity prototype of the recipe.	integer		mandatory

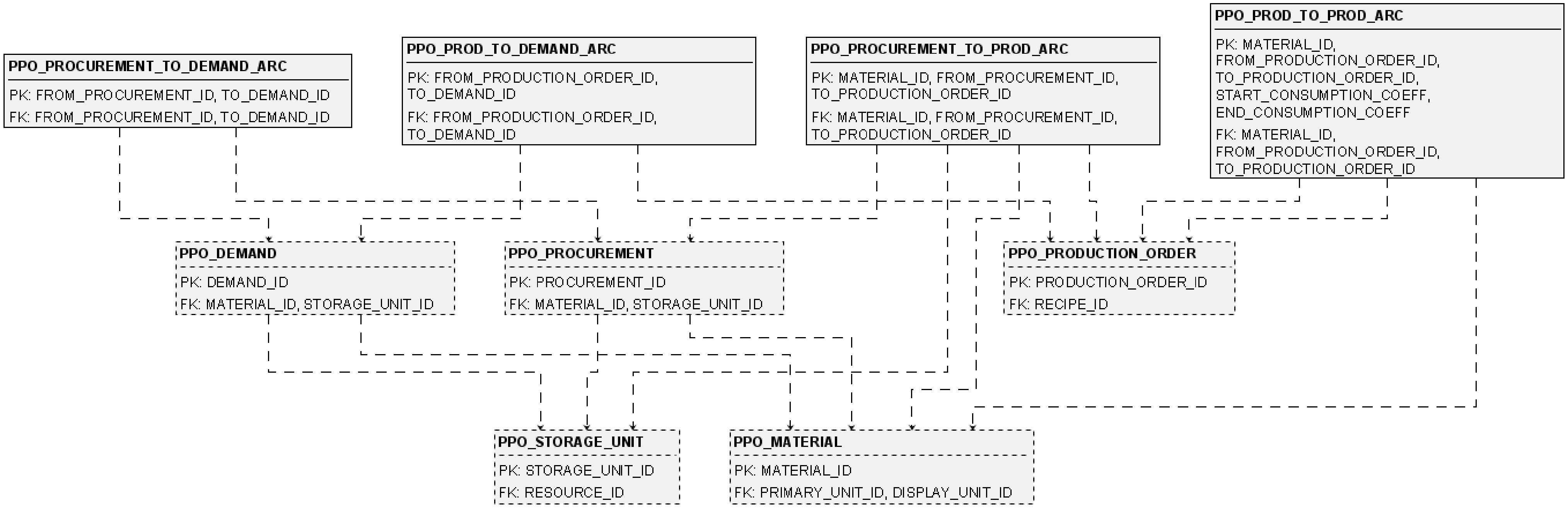

PPO_MATERIAL_PRODUCTION
PK: MATERIAL_ID, ACTIVITY_ID, MODE_NUMBER
FK: MATERIAL_ID, ACTIVITY_ID, STORAGE_ACTIVITY_ID, STORAGE_UNIT_ID
PPO_SETUP_MODE
PK: SETUP_ACTIVITY_ID, SETUP_MODE_NUMBER
FK: SETUP_ACTIVITY_ID, RESOURCE_ID, CALENDAR_ID
PPO_SETUP_SECONDARY_RESOURCE
PK: ACTIVITY_ID, MODE_NUMBER, RESOURCE_ID
FK: ACTIVITY_ID, RESOURCE_ID
PPO_SETUP_SETUP_COMPAT
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_SETUP_PRECED
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_PROD_SETUP_COMPAT
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_PROD_PRECED
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_PROD_SETUP_PRECED
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_SETUP_PROD_COMPAT
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_MODE
PK: ACTIVITY_ID, MODE_NUMBER
FK: ACTIVITY_ID, RESOURCE_ID, CALENDAR_ID
PPO_SECONDARY_RESOURCE
PK: ACTIVITY_ID, MODE_NUMBER, RESOURCE_ID
FK: ACTIVITY_ID, RESOURCE_ID
PPO_PROD_ORDER_PLANNED_MODE
PK: PRODUCTION_ORDER_ID, ACTIVITY_ID
FK: PRODUCTION_ORDER_ID, ACTIVITY_ID
PPO_PRODUCTION_ORDER_ACTIVITY
PK: PRODUCTION_ORDER_ID, ACTIVITY_PROTOTYPE_ID
FK: PRODUCTION_ORDER_ID, ACTIVITY_PROTOTYPE_ID, ACTIVITY_ID
PPO_ACTIVITY_CHAIN
PK: ACTIVITY_CHAIN_ID, POSITION
FK: ACTIVITY_ID
PPO_PROD_PROD_PRECED
PK: PREDECESSOR_ID, SUCCESSOR_ID, TYPE, DELAY_MIN, DELAY_MAX
FK: PREDECESSOR_ID, SUCCESSOR_ID
PPO_ACTIVITY_DUE_DATE
PK: ACTIVITY_ID, START_END_COEFFICIENT, DUE_TIME
FK: ACTIVITY_ID
PPO_SPANNING
PK: SPANNING_ACTIVITY_ID, SPANNED_ACTIVITY_ID
FK: SPANNING_ACTIVITY_ID, SPANNED_ACTIVITY_ID
PPO_PROD_PROD_COMPAT
PK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID, TYPE
FK: FIRST_ACTIVITY_ID, SECOND_ACTIVITY_ID
PPO_SETUP_ACTIVITY
PK: SETUP_ACTIVITY_ID
FK: PRODUCTION_ACTIVITY_ID
PPO_RESOURCE
PK: RESOURCE_ID
FK: CALENDAR_ID
PPO_ACTIVITY_PROTO
PK: ACTIVITY_ID
FK: RECIPE_ID
PPO_PRODUCTION_ORDER
PK: PRODUCTION_ORDER_ID
FK: RECIPE_ID
PPO_ACTIVITY
PK: ACTIVITY_ID
PPO_CALENDAR
PK: CALENDAR_ID
PPO_RECIPE
PK: RECIPE_ID
FK: PRIMARY_PRODUCT_ID, PRIMARY_INGREDIENT_ID
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID

Production order instance tables

This page lists the production order instance tables: Activities, modes, setups.

- PPO_ACTIVITY
- PPO_ACTIVITY_CHAIN
- PPO_ACTIVITY_DUE_DATE
- PPO_MATERIAL_PRODUCTION
- PPO_MODE
- PPO_PRODUCTION_ORDER
- PPO_PRODUCTION_ORDER_ACTIVITY
- PPO_PROD_ORDER_PLANNED_MODE
- PPO_PROD_PROD_COMPAT
- PPO_PROD_PROD_PRECED
- PPO_PROD_SETUP_COMPAT
- PPO_PROD_SETUP_PRECED
- PPO_SECONDARY_RESOURCE
- PPO_SETUP_ACTIVITY
- PPO_SETUP_MODE
- PPO_SETUP_PROD_COMPAT
- PPO_SETUP_PROD_PRECED
- PPO_SETUP_SECONDARY_RESOURCE
- PPO_SETUP_SETUP_COMPAT
- PPO_SETUP_SETUP_PRECED
- PPO_SPANNING

Return to Data Schema Home Page.


PPO_PROCUREMENT_TO_DEMAND_ARC
PK: FROM_PROCUREMENT_ID, TO_DEMAND_ID
FK: FROM_PROCUREMENT_ID, TO_DEMAND_ID
PPO_PROD_TO_DEMAND_ARC
PK: FROM_PRODUCTION_ORDER_ID, TO_DEMAND_ID
FK: FROM_PRODUCTION_ORDER_ID, TO_DEMAND_ID
PPO_PROCUREMENT_TO_PROD_ARC
PK: MATERIAL_ID, FROM_PROCUREMENT_ID, TO_PRODUCTION_ORDER_ID
FK: MATERIAL_ID, FROM_PROCUREMENT_ID, TO_PRODUCTION_ORDER_ID
PPO_PROD_TO_PROD_ARC
PK: MATERIAL_ID, FROM_PRODUCTION_ORDER_ID, TO_PRODUCTION_ORDER_ID, START_CONSUMPTION_COEFF, END_CONSUMPTION_COEFF
FK: MATERIAL_ID, FROM_PRODUCTION_ORDER_ID, TO_PRODUCTION_ORDER_ID
PPO_DEMAND
PK: DEMAND_ID
FK: MATERIAL_ID, STORAGE_UNIT_ID
PPO_PROCUREMENT
PK: PROCUREMENT_ID
FK: MATERIAL_ID, STORAGE_UNIT_ID
PPO_PRODUCTION_ORDER
PK: PRODUCTION_ORDER_ID
FK: RECIPE_ID
PPO_STORAGE_UNIT
PK: STORAGE_UNIT_ID
FK: RESOURCE_ID
PPO_MATERIAL
PK: MATERIAL_ID
FK: PRIMARY_UNIT_ID, DISPLAY_UNIT_ID

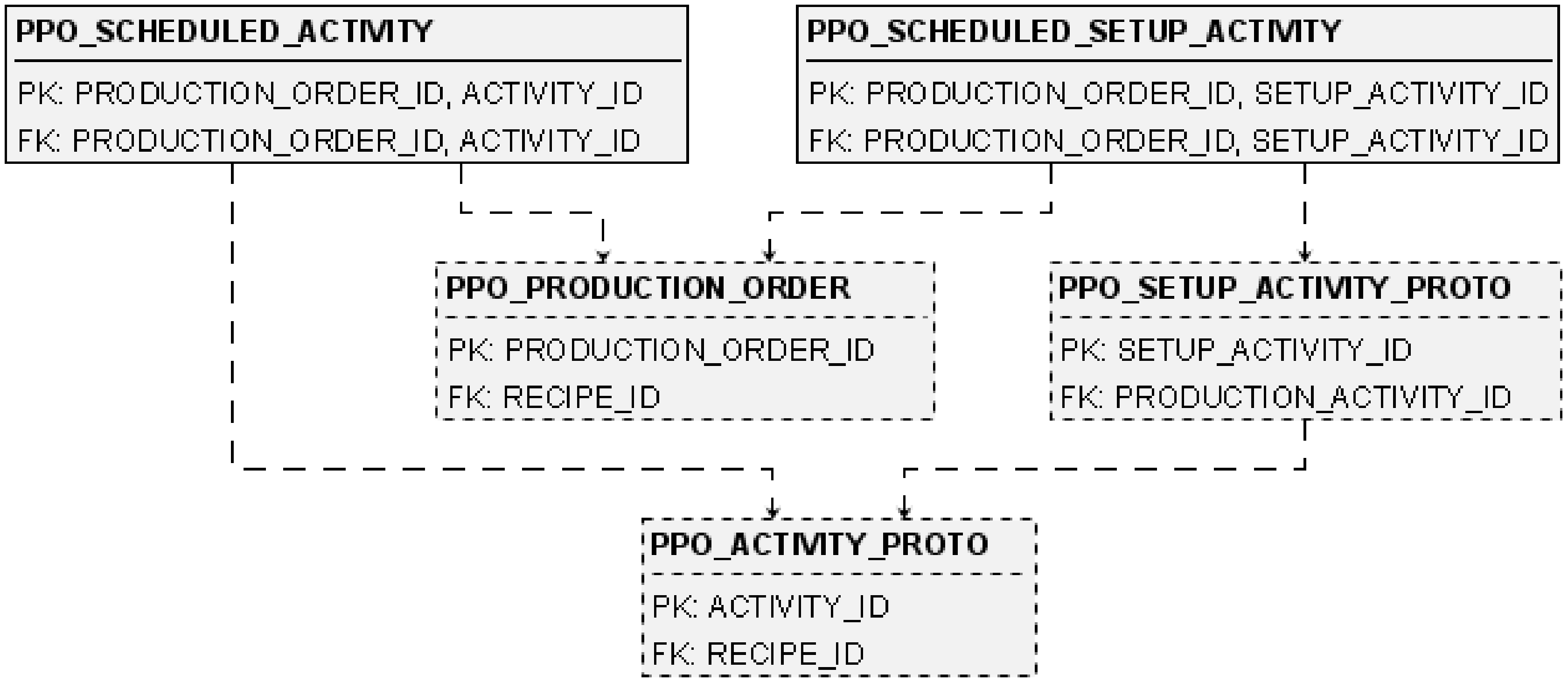
Material flow tables

This page lists the material flow arc tables, names ending with _ARC.

- PPO_PROCUREMENT_TO_DEMAND_ARC
- PPO_PROCUREMENT_TO_PROD_ARC
- PPO_PROD_TO_DEMAND_ARC
- PPO_PROD_TO_PROD_ARC

Return to

- Data Schema Home Page.


PPO_SCHEDULED_ACTIVITY
PK: PRODUCTION_ORDER_ID, ACTIVITY_ID
FK: PRODUCTION_ORDER_ID, ACTIVITY_ID
PPO_SCHEDULED_SETUP_ACTIVITY
PK: PRODUCTION_ORDER_ID, SETUP_ACTIVITY_ID
FK: PRODUCTION_ORDER_ID, SETUP_ACTIVITY_ID
PPO_PRODUCTION_ORDER
PK: PRODUCTION_ORDER_ID
FK: RECIPE_ID
PPO_SETUP_ACTIVITY_PROTO
PK: SETUP_ACTIVITY_ID
FK: PRODUCTION_ACTIVITY_ID
PPO_ACTIVITY_PROTO
PK: ACTIVITY_ID
FK: RECIPE_ID

PPO_SCHEDULED_ACTIVITY

Synopsis

```
PPO_SCHEDULED_ACTIVITY,PRODUCTION_ORDER_ID,ACTIVITY_ID,MODE_NUMBER,START_TIME,END_TIME,
PERFORMED_STATUS,FIRM_MODE_NUMBER,FIRM_START_MIN,FIRM_END_MIN,FIRM_START_MAX,FIRM_END_MAX,
FIRM_PERFORMED_STATUS
```

Topic

Production Schedules

Description

The PPO_SCHEDULED_ACTIVITY table is used to represent a production schedule. Each line in this table specifies when a given activity of a given production order is scheduled to execute and in which mode. Each PPO_SCHEDULED_ACTIVITY table may belong to an alternative scheduling solution.

Primary keys

PRODUCTION_ORDER_ID,ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
PRODUCTION_ORDER_ID	Mandatory identifier of a production order.	id		mandatory
ACTIVITY_ID	Mandatory identifier of an activity prototype of the recipe of the given production order.	id		mandatory
MODE_NUMBER	The number of the scheduled mode (`-1` if unknown).	integer		-1
START_TIME	The scheduled start time of the activity.	integer		mandatory
END_TIME	The scheduled end time of the activity.	integer		mandatory
PERFORMED_STATUS	The status of the activity; that is, either `Performed` or `Unperformed`.	id		"Performed"
FIRM_MODE_NUMBER	The index of the definitely imposed mode, if any (`-1` otherwise). When a value different from `-1` is given, it means the mode can no longer be changed; for example, because the scheduled activity is ready to execute in the factory.	integer		-1
FIRM_START_MIN	A firm minimal start time before which the scheduled activity can no longer start in the factory.	integer		-INF

FIRM_END_MIN	A firm minimal end time before which the scheduled activity can no longer end in the factory.	integer	-INF
FIRM_START_MAX	A firm maximal start time after which the scheduled activity can no longer start in the factory.	integer	+INF
FIRM_END_MAX	A firm maximal end time after which the scheduled activity can no longer end in the factory.	integer	+INF
FIRM_PERFORMED_STATUS	The status of the activity (that is, either `Performed` or `Unperformed`) if it is firm; the default value `PerformedOrUnperformed` otherwise.	id	"PerformedOrUnperformed"

PPO_SCHEDULED_SETUP_ACTIVITY

Synopsis

```
PPO_SCHEDULED_SETUP_ACTIVITY,PRODUCTION_ORDER_ID,SETUP_ACTIVITY_ID,MODE_NUMBER,START_TIME,
END_TIME,PERFORMED_STATUS,FIRM_MODE_NUMBER,FIRM_START_MIN,FIRM_END_MIN,FIRM_START_MAX,
FIRM_END_MAX,FIRM_PERFORMED_STATUS
```

Topic

Production Schedules

Description

The PPO_SCHEDULED_SETUP_ACTIVITY table is used to represent a production setup activity schedule. Each line in this table specifies when a given setup activity of a given production order is scheduled to execute and in which mode. Each PPO_SCHEDULED_SETUP_ACTIVITY table may belong to an alternative scheduling solution.

Primary keys

PRODUCTION_ORDER_ID,SETUP_ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
PRODUCTION_ORDER_ID	Mandatory identifier of a production order.	id		mandatory
SETUP_ACTIVITY_ID	Mandatory identifier of a setup activity prototype of the recipe of the given production order.	id		mandatory
MODE_NUMBER	The number of the scheduled mode (`-1` if unknown).	integer		-1
START_TIME	The scheduled start time of the activity.	integer		mandatory
END_TIME	The scheduled end time of the activity.	integer		mandatory
PERFORMED_STATUS	The status of the activity, that is, either `Performed` or `Unperformed`.	id		"Performed"
FIRM_MODE_NUMBER	The number of the definitely imposed mode, if any (`-1` otherwise). When a value different from `-1` is given, it means the mode can no longer be changed; for example, because the scheduled activity is ready to execute in the factory.	integer		-1

FIRM_START_MIN	A firm minimal start time before which the scheduled activity can no longer start in the factory.	integer	-INF
FIRM_END_MIN	A firm minimal end time before which the scheduled activity can no longer end in the factory.	integer	-INF
FIRM_START_MAX	A firm maximal start time after which the scheduled activity can no longer start in the factory.	integer	+INF
FIRM_END_MAX	A firm maximal end time after which the scheduled activity can no longer end in the factory.	integer	+INF
FIRM_PERFORMED_STATUS	The status of the activity (that is, either `Performed` or `Unperformed`) if it is firm; the default value `PerformedOrUnperformed` otherwise.	id	"PerformedOrUnperformed"

Data input with csv files

You can create comma separated values (*.csv) data files with any common spreadsheet. One csv file includes all the data tables (such as PPO_RESOURCE, PPO_MATERIAL, and so forth) that defines the problem data. Each data table has at least several rows, with commas that delimit or separate each field of the row. Every table row begins with the table name.

The file begins by declaring the CSV format:

```
ILOG_CSV_FORMAT, 1.0
```

This is followed by the data schema name and version.

```
ILOG_DATA_SCHEMA, PPO, 4.0
```

After these two lines you then add the data tables. Most tables start with three keyword lines: the NAMES row which lists each field of the table that you are specifying; the KEYS row which indicates the keys for the table; and TYPES which declares the data type of the field. The PPO_MODEL table does not need the KEYS line, but the other tables do require it. These three rows all begin with the table name, followed by a vertical separator character.

Next you enter the data rows for the table. Keywords and vertical separators are not used in data rows; simply start each row with the table name, followed by the values for each field, separated by commas. As an example, here is a simple PPO_MATERIAL table with two data fields (MATERIAL_ID and NAME), and it defines two materials:

```
PPO_MATERIAL|NAMES, MATERIAL_ID, NAME
PPO_MATERIAL|KEYS, 1, 0
PPO_MATERIAL|TYPES, id, string
PPO_MATERIAL, 0, PINE_CRADLE
PPO_MATERIAL, 1, TEAK_CRADLE
```

Spaces are not allowed in field names. Capital letters are suggested but not required. You do not have to declare all the fields available in any particular table. In fact you should declare only the fields that are at least one of the following: A) mandatory, B) contain default values that you need to change, or C) are fields that you really need to use. An entry is required on every row for every field declared in the NAMES row. Note that field order is not important; in the example above, you could put the NAME field before the MATERIAL_ID field. However, you must consistently follow that same order in all rows. It is suggested that data rows be entered in the order of their ID (if there is one), but this is not required.

Any line in a data file starting with the hash or pound sign (#) is treated as a comment line by PPO and ignored by the data file reader.

Return to Data Schema Home Page.

Data schema types

There are five types of PPO data: `boolean`, `float` (or `double`), `int` (or `integer`), `id`, and `string`.

Note that values in the TYPE column are for informational purposes; PPO checks for data type integrity regardless of the type declaration.

Values of type `boolean` represent Boolean values. Boolean values are integers; true is represented by 1 and false is represented by 0.

In this context, values of type `float` and `double` can be used interchangeably, and represent double precision floating-point values. The symbols +INF, +1.#INF, +Inf, and +Infinity are used to represent positive infinite double values. The symbols -INF, -1.#INF, -Inf, and -Infinity are used to represent negative infinite double values. Note that the leading + or - must be included in the value.

Values of type `int` and `integer` represent integer values. To ensure portability across platforms and to avoid overflows, all integers used in Plant PowerOps must be in the range from -INF = -999999999 to +INF = 999999999. Note that -999999999 and 999999999 are generally used to state that no constraint applies. For example, if a maximal delay is set to 999999999, it means that there is in fact no maximal delay. In the documentation, -INF and +INF are used to denote -999999999 and 999999999, but -999999999 and 999999999 are the integer values to be used in files and databases.

Values of type `id` represent unique identifiers. A valid id is any nonempty string other than the undefined id value of "NULL". As `id` is a string, do not accidentally add blank spaces to this field.

Values of type `string` are used to represent text.

Return to Data Schema Home Page.

Data model checking

Invalid data models can cause a variety of errors and warnings.

Some data model integrity tests are performed directly by the reader, or when using the API. In many cases of invalid data, the reader will ignore an invalid tuple or part of an invalid tuple.

For example, if the given productivity is negative for a calendar interval, then the negative value is ignored and the productivity is set to its default value (that is, 1). It is strongly recommended that whenever a reader error is signaled in the PPO console, action is taken to correct the problem at its source. In this example, that simply means replacing the negative productivity in the source database or CSV file with a valid entry.

More complex tests are done by the checker. There are four classes of messages that might display when a file is loaded in the PPO GUI. "Fatal errors" are the most severe, revealing inconsistency in the data that PPO cannot overcome. When a fatal error arises, it is very likely that no solution to the optimization problem will be found. Next in severity are "Errors," which are likely to be overcome, but strongly suggest there is either an integration problem or the violation of a constraint on the factory floor.

Least in severity are "Warnings" and "Messages," both of which you should examine to determine an appropriate response.

When the reader must ignore invalid data and use a default value, the data error is not reported again in the checker. In the previous example of negative productivity, the reader would discard the invalid data and use the default value of 1; so of course the default value is accepted by the checker.

Return to Data Schema Home Page.

PPO_ACTIVITY

Synopsis

`PPO_ACTIVITY,ACTIVITY_ID,NAME,PERFORMED_STATUS,CLEANUP_STATUS,COLOR`

Topic

Production Orders

Description

The PPO_ACTIVITY table is used to represent instances of production activities (see PPO_ACTIVITY_PROTO). This table includes cleanup activity instances (CLEANUP_STATUS) but not setup activities.

Primary keys

ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	This data is the mandatory unique identifier of the activity.	id		mandatory
NAME	The name of the activity. See PPO_ACTIVITY_PROTO for more information.	string		""
PERFORMED_STATUS	This data sets the status of an activity. The possible values are `Performed`, `Unperformed`, or `PerformedOrUnperformed`; the last status means that the engine has the option of relaxing the capacity constraint (as if the activity were to possibly be outsourced, so no longer under capacity limitations of modeled resources).	id		"Performed"
CLEANUP_STATUS	This data sets the cleanup status of an activity. The possible values are `Cleaning` or `NotCleaning`.	id		"NotCleaning"
COLOR	The color in the Gantt Chart.	string		""

PPO_ACTIVITY_CHAIN

Synopsis

```
PPO_ACTIVITY_CHAIN,ACTIVITY_CHAIN_ID,POSITION,ACTIVITY_ID
```

Topic

Production Orders

Description

The PPO_ACTIVITY_CHAIN table creates a chain of production activity instances (see PPO_ACTIVITY_CHAIN_PROTO).

Primary keys

ACTIVITY_CHAIN_ID,POSITION

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_CHAIN_ID	This data is the mandatory unique identifier of the activity chain.	id		mandatory
POSITION	The position of the activity in the chain (0-based index).	integer		mandatory
ACTIVITY_ID	The identifier of an activity belonging to the chain.	id		mandatory

PPO_ACTIVITY_CHAIN_PROTO

Synopsis

```
PPO_ACTIVITY_CHAIN_PROTO,ACTIVITY_CHAIN_ID,POSITION,ACTIVITY_ID
```

Topic

Recipes, Activities, and Modes

Description

The PPO_ACTIVITY_CHAIN_PROTO table creates a chain of production activity prototypes. The definition of an activity chain is: 1) The activities must be scheduled in one consistent order on the same primary resource. 2) The activities must be production activities. No setup activities are allowed. 3) The activities must all have the same setup state. 4) The required capacity on the primary resource must be the same for all activities (A1,A2,A3 will form a rectangle on the primary resource). 5) The resource is used from the beginning to the end of the chain.

Primary keys

ACTIVITY_CHAIN_ID,POSITION

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_CHAIN_ID	This data is the mandatory unique identifier of the activity chain.	id		mandatory
POSITION	The position of the activity in the chain (0-based index).	integer		mandatory
ACTIVITY_ID	The identifier of an activity belonging to the chain.	id		mandatory

PPO_ACTIVITY_DUE_DATE

Synopsis

```
PPO_ACTIVITY_DUE_DATE,ACTIVITY_ID,START_END_COEFFICIENT,DUE_TIME,EARLINESS_VARIABLE_COST,
TARDINESS_VARIABLE_COST,EARLINESS_FIXED_COST,TARDINESS_FIXED_COST
```

Topic

Production Orders

Description

The PPO_ACTIVITY_DUE_DATE table is used to define due dates that are attached to activities. An activity due date object can be attached to the start or end time of an activity. Four cost penalties can be defined for any given due date: An earliness fixed cost, an earliness variable cost, a tardiness fixed cost, and a tardiness variable cost. Note that the default for all costs is zero, so by default there is no penalty for earliness or tardiness.

The earliness fixed cost is a fixed price to be paid if the activity is early. For example, it represents the cost of entering a final product into a warehouse, rather than shipping it directly to the customer. The earliness variable cost is the price to be paid (in addition to the earliness fixed cost) per time unit that the activity is early. For example, it represents the cost of using additional space in a warehouse for an early final product. Likewise with tardiness costs: The tardiness fixed cost is a fixed price to be paid if the activity is late, and the tardiness variable cost is the price to be paid (in addition to the tardiness fixed cost) per time unit that the activity is late.

For example, suppose the due date for the end time of an activity is 1000, the earliness fixed cost is 10.0, the earliness variable cost is 2.0, the tardiness fixed cost is 200.0, and the tardiness variable cost is 5.0. If the activity ends at time 995, the earliness cost is 10.0 + 2.0 * (1000 - 995) = 20.0, and the tardiness cost is zero. If the activity ends at time 1010, the tardiness cost is 200.0 + 5.0 * (1010 - 1000) = 250.0, and the earliness cost is zero. And, of course, if the activity ends exactly at time 1000, both the earliness and the tardiness costs are zero.

Primary keys

ACTIVITY_ID,START_END_COEFFICIENT,DUE_TIME

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of the one-shot activity with which the due date is associated.	id		mandatory

START_END_COEFFICIENT	A floating point number between 0.0 and 1.0 stating if the due date applies to the start (0.0) or to the end (1.0) of the activity. Intermediate values can be used to specify intermediate points between the start time and the end time of the activity. For example, 0.4 means that the due date applies to start + 0.4 (end - start) = 0.6 start + 0.4 end.	double	1.0
DUE_TIME	The time at which the demand is due or at which the activity shall start or end. This time is expressed with respect to the DATE_ORIGIN and TIME_UNIT of the model.	integer	mandatory
EARLINESS_VARIABLE_COST	This data defines the variable cost used to compute the earliness cost if the activity is early. This variable cost will be multiplied by the difference between the DUE_TIME and (start + START_END_COEFFICIENT * (end - start)).	double	0.
TARDINESS_VARIABLE_COST	This data defines the variable cost used to compute the tardiness cost if the activity is tardy. This variable cost will be multiplied by the difference between (start + START_END_COEFFICIENT * (end - start)) and the DUE_TIME.	double	0.
EARLINESS_FIXED_COST	This data defines the fixed cost used to compute the earliness cost if the activity is early.	double	0.
TARDINESS_FIXED_COST	This data defines the fixed cost used to compute the tardiness cost if the activity is tardy.	double	0.

PPO_ACTIVITY_PROTO

Synopsis

`PPO_ACTIVITY_PROTO,ACTIVITY_ID,NAME,RECIPE_ID,PERFORMED_STATUS,CLEANUP_STATUS,COLOR`

Topic

Recipes, Activities, and Modes

Description

The PPO_ACTIVITY_PROTO table is used to represent the prototypes of production activities. This table includes cleanup activities (CLEANUP_STATUS) but not setup activities (see PPO_SETUP_ACTIVITY_PROTO and PPO_SETUP_ACTIVITY).

This table is used to create an activity *prototype* that belongs to a recipe for production. An activity typically has three phases in PPO: As a prototype, an instance, and as a scheduled activity. An activity prototype is a template or mold of an activity, represented through the recipes and modes in the data model. It is not an actual specific activity instance, but rather a model of an activity that typically is performed many times in a plant process. During optimization, the production orders are created and this process uses the prototypes as a template to create the actual, explicit instances of activities stored in PPO_ACTIVITY table. Optimization then schedules each of these instances to a particular time slot, and the results can be viewed in the **Scheduled Activities** data table in the GUI.

There are several ways to control the names of activities. To define a name that spans several lines in graphic objects in the GUI Gantt chart, use **\n** in the defined name to start a new line. Then to properly see multiline names in the Gantt, change the **Activity and Resource Chart Row Height** in the **Options** menu. By default, activities generated from activity prototypes have an automatic long name based on a concatenation of the production order name and the activity prototype name, separated by a dot. To keep the original name of the activity prototype, you must set the PPO_SETTING parameter `bShortName` to true. When the `bShortName` is true, the activity label from the activity prototypes interprets the following special characters as follows: **^a** = abbreviated main product name to 3 first letters (NOT SUPPORTING MULTIBYTE CHARACTERS); **^b** = abbreviated main ingredient name to 3 first letters (NOT SUPPORTING MULTIBYTE CHARACTERS); **^i** = main product identifier; **^j** = main ingredient identifier; **^m** = main ingredient name; **^n** = order index; **^o** = order identifier; **^p** = main product name; **^q** = quantity of main product or batch size (if missing round up as an integer); **^r** = recipe name (if missing then the recipe identifier, or if that's missing then recipe id); **^s** = batch size.

Primary keys

ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default

ACTIVITY_ID	This data is the mandatory unique identifier of the activity.	id	mandatory
NAME	The name of the activity. See **Description** above for more information.	string	""
RECIPE_ID	The recipe to which the activity prototype belongs.	id	mandatory
PERFORMED_STATUS	This data sets the status of an activity. The possible values are `Performed`, `Unperformed`, or `PerformedOrUnperformed`; the last status means that the engine has the option of relaxing the capacity constraint (as if the activity were to possibly be outsourced, so no longer under capacity limitations of modeled resources).	id	"Performed"
CLEANUP_STATUS	This data sets the cleanup status of an activity. The possible values are `Cleaning` or `NotCleaning`.	id	"NotCleaning"
COLOR	The color in the Gantt Chart.	string	""

PPO_ACTIVITY_SETUP_STATE_PROTO

Synopsis

`PPO_ACTIVITY_SETUP_STATE_PROTO,ACTIVITY_ID,SETUP_FEATURE,SETUP_STATE`

Topic

Recipes, Activities, and Modes

Description

The PPO_ACTIVITY_SETUP_STATE_PROTO table defines the setup states required by each activity prototype for each setup feature.

Primary keys

ACTIVITY_ID,SETUP_FEATURE

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of the activity.	id		mandatory
SETUP_FEATURE	The identifier of the feature. If you don't need the notion of feature, use `NoFeature`.	id		"NoFeature"
SETUP_STATE	The state required by the activity for the setup feature.	id		mandatory

PPO_BUCKET

Synopsis

```
PPO_BUCKET,BUCKET_ID,NAME,START_TIME,END_TIME,BUCKET_SEQUENCE_ID
```

Topic

General tables

Description

The PPO_BUCKET table defines time buckets for the planning engine. The planning engine does not check the constraints at each time unit as the scheduling engine does, but rather checks globally for each time bucket. Typical time buckets are months, weeks, or days. Buckets are organized by bucket sequence, and a model can have several sequences. For example, the planning module can use daily buckets, while the GUI displays weekly buckets.

Primary keys

BUCKET_ID

Field descriptions

Field Name	Description	Type	Range	Default
BUCKET_ID	The unique identifier of the time bucket.	id		mandatory
NAME	Optional name.	string		NULL
START_TIME	The start time of the time bucket expressed in time units since the origin.	integer		mandatory
END_TIME	The end time of the time bucket expressed in time units since the origin.	integer		mandatory
BUCKET_SEQUENCE_ID	The bucket sequence that contains the buckets. This data is optional, but a default sequence will be used if nothing is specified.	id		NULL

PPO_BUCKET_SEQUENCE

Synopsis

`PPO_BUCKET_SEQUENCE,BUCKET_SEQUENCE_ID,NAME,OFFSET`

Topic

General tables

Description

A bucket is always a member of a sequence. If no sequence is defined, a default one is automatically created.

Primary keys

BUCKET_SEQUENCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
BUCKET_SEQUENCE_ID	This data is the mandatory unique identifier of the sequence.	id		mandatory
NAME	This data is the name describing the sequence; for example, `daily` or `weekly`.	string		mandatory
OFFSET	This data sets the amount of time to add to the start time of the generated buckets (from the PPO_BUCKET_TEMPLATE table) of this bucket sequence. This offset is expressed in time units. For instance, when defining a bucket template of type `Day`, setting this offset to 3600 seconds makes the generated buckets start at 1:00 PM instead of midnight.	integer		0

PPO_BUCKET_TEMPLATE

Synopsis

`PPO_BUCKET_TEMPLATE,BUCKET_SEQUENCE_ID,BUCKET_RANK,BUCKET_TYPE,PERIOD_UNIT,NUMBER_OF_PERIODS`

Topic

General tables

Description

The PPO_BUCKET_TEMPLATE table defines a pattern or mold for bucket generation. When this table exists in a database, PPO automatically generates buckets according to the template(s). Several templates can exist in this table. Note that this table has no effect when loaded from a .csv file.

Primary keys

BUCKET_SEQUENCE_ID,BUCKET_RANK

Field descriptions

Field Name	Description	Type	Range	Default
BUCKET_SEQUENCE_ID	The unique identifier of the time bucket.	id		mandatory
BUCKET_RANK	The rank of the buckets generated from this template in the bucket sequence.	integer		mandatory
BUCKET_TYPE	The type of the generated buckets. The bucket type defines the default bucket duration. This data must have one of the following values: `Hour`, `EightHShift`, `Day`, `Week`, `Month`, `Quarter`, or `Year`. `Quarter` is for buckets of three months duration. `EightHShift` is for buckets of eight hours duration starting at midnight. You can use the column `OFFSET` of the PPO_BUCKET_SEQUENCE table to generate buckets that start before or after midnight.	id		mandatory
PERIOD_UNIT	The period unit is the time period over which the buckets will be generated; it is a calendar unit. The period unit must be greater than or equal to the bucket type. This data must have one of the following values: `Hour`, `EightHShift`, `Day`, `Week` (by default weeks start on Monday), `Month` (calendar month), `Quarter`, or `Year`. In PPO, quarters are defined as follows: `Q1` is from January through March; `Q2` is from April through June; `Q3` is from July through September; and `Q4` is from October through December.	id		mandatory

NUMBER_OF_PERIODS	Used in conjunction with PERIOD_UNIT to define the time period over which the buckets will be generated (*number of periods* times *period unit* equals coverage of bucket generation). For example, if `BUCKET_TYPE` equals to `Day`, `PERIOD_UNIT` equals `Week` and `NUMBER_OF_PERIODS` equals `2`, then PPO will generate buckets of 24 hour duration, for a period of 2 weeks. If the `START_MIN` of the model is a "Monday" then PPO will generate 14 buckets from this template. If the `START_MIN` of the model is a "Wednesday" then PPO will generate 12 buckets. If you want your buckets independent from the calendar (in order to always generate the same number of buckets), set the BUCKET_TYPE and the PERIOD_UNIT to the same value.	integer	1

PPO_CALENDAR

Synopsis

```
PPO_CALENDAR,CALENDAR_ID,NAME,ADDITIVE
```

Topic

Manufacturing Resources

Description

This table declares the existing calendars. Resources operate with respect to different calendars, thereby specifying the evolution of resource capacity and productivity over time. Calendars also specify breaks during which the resource may not operate, and changes of work shifts.

Primary keys

CALENDAR_ID

Field descriptions

Field Name	Description	Type	Range	Default
CALENDAR_ID	The mandatory unique identifier of the calendar.	id		mandatory
NAME	Optional name.	string		""
ADDITIVE	This boolean specifies how to manage overlapping calendar intervals within the calendar. If the calendar is additive, then each calendar interval provides capacity over a given time interval; if several intervals overlap, then the corresponding capacities shall be added to determine the actual capacity of the resource at a given time. If the calendar is not additive, then each calendar interval limits the capacity available over a given time interval; if several intervals overlap, the most constraining limit applies.	boolean		0

PPO_CALENDAR_INTERVAL

Synopsis

```
PPO_CALENDAR_INTERVAL,CALENDAR_ID,START_TIME,END_TIME,NAME,PERIODICITY,PERIOD_START_TIME,
PERIOD_END_TIME,CAPACITY,PRODUCTIVITY,IS_BREAK,END_OF_SHIFT
```

Topic

Manufacturing Resources

Description

Each calendar interval has capacity, productivity, break, and end of shift properties, with default values for each. The default capacity is the capacity of the resource to which the calendar is attached; the default productivity is 1.0; and by default, a calendar interval is neither a break nor an end of shift.

Calendar intervals can be periodic or aperiodic (without recurring regular intervals). A periodic interval of periodicity $p > 0$ is implicitly repeated every p units of time from a given PERIOD_START_TIME to a given PERIOD_END_TIME. An aperiodic interval (with a PERIODICITY of 0) occurs only once, from its START_TIME to its END_TIME (although on the GUI Calendars view, you can copy the interval using the "Repeat every" GUI function).

Primary keys

CALENDAR_ID,START_TIME,END_TIME

Field descriptions

Field Name	Description	Type	Range	Default
CALENDAR_ID	This data is the mandatory unique identifier of the calendar to which the calendar interval belongs.	id		mandatory
START_TIME	This data sets the start time of the corresponding calendar interval.	integer		mandatory
END_TIME	This data sets the end time of the corresponding calendar interval.	integer		mandatory
NAME	Optional name for the calendar interval.	string		""
PERIODICITY	This data sets the periodicity of the corresponding calendar interval. The value `0` means that the interval is aperiodic.	integer		0
PERIOD_START_TIME	This data sets the period start time of the corresponding calendar interval.	integer		-INF
PERIOD_END_TIME	This data sets the period end time of the corresponding calendar interval.	integer		+INF
CAPACITY	This data sets the capacity available over the calendar interval.	integer		+INF

PRODUCTIVITY	This data sets the productivity of the corresponding calendar interval.	double	1.0
IS_BREAK	This data states whether the calendar interval is a break.	boolean	0
END_OF_SHIFT	This data states whether the corresponding calendar interval is an end of shift.	boolean	0

PPO_CRITERION_WEIGHT

Synopsis

```
PPO_CRITERION_WEIGHT,OPTIMIZATION_PROFILE_ID,CRITERION_ID,WEIGHT
```

Topic

General tables

Description

The PPO_CRITERION_WEIGHT table defines the global weights of the directly optimizable Key Performance Indicators (KPIs). These weights can be associated with different optimization profiles. Key Performance Indicators and optimization criteria are discussed in detail in the *Key performance indicators*.

Primary keys

OPTIMIZATION_PROFILE_ID,CRITERION_ID

Field descriptions

Field Name	Description	Type	Range	Default
OPTIMIZATION_PROFILE_ID	The optimization profile under consideration.	id		"NoProfile"
CRITERION_ID	The criterion. The possible values are: TotalRevenue, TotalNonDeliveryCost, TotalResourceCost, TotalIdleCost, TotalProcessingCost, TotalUnperformedCost, TotalSetupCost, TotalCleanupCost, TotalEarlinessCost, TotalTardinessCost, TotalInventoryCost, TotalInventoryDeficitCost, TotalWasteCost.	id		mandatory
WEIGHT	The weight assigned to the criterion.	double		0.

PPO_DATA_SCHEMA

Synopsis

`PPO_DATA_SCHEMA,DATA_SCHEMA_ID`

Topic

General tables

Description

This table stores the data schema version of PPO. It is used only in database persistence. Only a single record is stored in this table.

Primary keys

DATA_SCHEMA_ID

Field descriptions

Field Name	Description	Type	Range	Default
DATA_SCHEMA_ID	This data is the mandatory unique identifier of the data schema.	id		mandatory

PPO_INVENTORY_MAX_COST

Synopsis

```
PPO_INVENTORY_MAX_COST,MATERIAL_ID,INVENTORY_MAX_COST_FCT_ID,VALIDITY_START_TIME,
VALIDITY_END_TIME
```

Topic

Materials and Storage Units

Description

The PPO_INVENTORY_MAX_COST table specifies how the cost of maintaining inventory over time is evaluated. For each material, a piecewise or stepwise linear inventory cost function is defined. This function may vary over time.

Primary keys

MATERIAL_ID,INVENTORY_MAX_COST_FCT_ID,VALIDITY_START_TIME,VALIDITY_END_TIME

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material.	id		mandatory
INVENTORY_MAX_COST_FCT_ID	The identifier of the cost function.	id		mandatory
VALIDITY_START_TIME	The start time of the period over which the given function applies.	integer		-INF
VALIDITY_END_TIME	The end time of the period over which the given function applies.	integer		+INF

PPO_INVENTORY_MAX_COST_FCT

Synopsis

```
PPO_INVENTORY_MAX_COST_FCT,INVENTORY_MAX_COST_FCT_ID,LEVEL_NUMBER,INVENTORY_MAX,FIXED_COST,
VARIABLE_COST
```

Topic

Materials and Storage Units

Description

The PPO_INVENTORY_MAX_COST_FCT table defines a piecewise or stepwise linear cost function used to evaluate the cost of maintaining inventory over time.

Primary keys

INVENTORY_MAX_COST_FCT_ID,LEVEL_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
INVENTORY_MAX_COST_FCT_ID	The identifier of the cost function.	id		mandatory
LEVEL_NUMBER	The index (zero-based) of an inventory level.	integer		0
INVENTORY_MAX	The amount of inventory up to which this level applies.	double		+1.#INF
FIXED_COST	The fixed cost incurred to enter this level of inventory; that is, when the INVENTORY_MAX of the previous level is exceeded.	double		0.
VARIABLE_COST	The slope of the piecewise linear cost function for this level of inventory; that is, for each unit of inventory included in this level (from the INVENTORY_MAX of the previous level to the INVENTORY_MAX of this level).	double		0.

PPO_INVENTORY_MIN_COST

Synopsis

```
PPO_INVENTORY_MIN_COST,MATERIAL_ID,INVENTORY_MIN_COST_FCT_ID,VALIDITY_START_TIME,
VALIDITY_END_TIME
```

Topic

Materials and Storage Units

Description

The PPO_INVENTORY_MIN_COST table defines a piecewise or stepwise linear cost function used for penalizing inventory deficits. For each material, a piecewise or stepwise linear cost function is defined. This function may vary over time.

Primary keys

MATERIAL_ID,INVENTORY_MIN_COST_FCT_ID,VALIDITY_START_TIME,VALIDITY_END_TIME

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material.	id		mandatory
INVENTORY_MIN_COST_FCT_ID	The identifier of the cost function.	id		mandatory
VALIDITY_START_TIME	The start time of the interval over which the given function applies.	integer		-INF
VALIDITY_END_TIME	The end time of the interval over which the given function applies.	integer		+INF

PPO_INVENTORY_MIN_COST_FCT

Synopsis

```
PPO_INVENTORY_MIN_COST_FCT,INVENTORY_MIN_COST_FCT_ID,LEVEL_NUMBER,INVENTORY_MIN,FIXED_COST,
VARIABLE_COST
```

Topic

Materials and Storage Units

Description

The PPO_INVENTORY_MIN_COST_FCT table defines a piecewise or stepwise linear cost function for penalizing safety stock violations.

Primary keys

INVENTORY_MIN_COST_FCT_ID,LEVEL_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
INVENTORY_MIN_COST_FCT_ID	The identifier of the cost function.	id		mandatory
LEVEL_NUMBER	The index (zero-based) of an inventory level.	integer		0
INVENTORY_MIN	The amount of inventory below which this level applies.	double		0.
FIXED_COST	The fixed cost incurred if the inventory falls below the given INVENTORY_MIN.	double		0.
VARIABLE_COST	The absolute value of the negative slope of the piecewise linear cost function for this level of inventory deficit; that is, below the given INVENTORY_MIN down to the INVENTORY_MIN of the next level (0 if undefined).	double		0.

PPO_MATERIAL

Synopsis

```
PPO_MATERIAL,MATERIAL_ID,NAME,INVENTORY_CAPACITY,DAYS_OF_SUPPLY_TARGET_MIN,
DAYS_OF_SUPPLY_TARGET,DAYS_OF_SUPPLY_TARGET_MAX,TARGET_MIN_VARIABLE_COST,
TARGET_VARIABLE_COST,TARGET_MAX_VARIABLE_COST,INVENTORY_COST_FCT_TIME_STEP,SHELF_LIFE,
MATURITY,PRIMARY_UNIT_ID,DISPLAY_UNIT_ID,COLOR,VISIBLE,PEGGING_STRATEGY,SERVICE_LEVEL_TYPE,
TARGET_SERVICE_LEVEL,DEMAND_VARIABILITY,AVERAGE_LEAD_TIME,LEAD_TIME_STD_DEVIATION
```

Topic

Materials and Storage Units

Description

The PPO_MATERIAL table is used to represent finished products, raw materials, or intermediates. Each row in a material modeling table corresponds to a Stock-Keeping Unit (SKU). For example, two models of white shirts in the same inventory but of different sizes or styles would be two different SKUs; thus two different rows in a PPO_MATERIAL table. The same model of shirt held in two different inventories would also be two different SKUs, thus two rows in a table as well. To represent the stock for the same material in different warehouses, create separate rows, and if necessary group them in a PPO_MATERIAL_FAMILY table in order to aggregate the results in reports.

Primary keys

MATERIAL_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	This data is the mandatory unique identifier of the material.	id		mandatory
NAME	Optional name.	string		""
INVENTORY_CAPACITY	The maximal inventory allowed for this material. This data is useful in fixed location systems where a SKU is assigned a permanent location and no other items are stored there. In a floating location system where goods are stored wherever there is appropriate space for them, the column QUANTITY_MAX of the PPO_STORAGE_UNIT table must be used instead to limit the warehouse capacity.	double		+1.#INF

DAYS_OF_SUPPLY_TARGET_MIN	The minimum number of days of supply of the material that the user wants to hold in inventory at the end of each time bucket. This value is considered as a strong preference but is not necessarily obeyed.	double	0.
DAYS_OF_SUPPLY_TARGET	The ideal number of days of supply of the material that the user wants to hold in inventory at the end of each time bucket. This value is considered as a strong preference but is not necessarily obeyed. This value is ignored if it is not in between DAYS_OF_SUPPLY_TARGET_MIN and DAYS_OF_SUPPLY_TARGET_MAX. For short term planning, define only the min and max. The use of DAYS_OF_SUPPLY_TARGET is for mid-term planning when the bucket duration is greater than the width of the stock corridor (between min and max).	double	0.
DAYS_OF_SUPPLY_TARGET_MAX	The maximum number of days of supply of the material that the user wants to hold in inventory at the end of each time bucket. This value is considered as a strong preference but is not necessarily obeyed.	double	+1.#INF
TARGET_MIN_VARIABLE_COST	The variable cost incurred per day and per unit of material when inventory is below the minimal target inventory.	double	0.
TARGET_VARIABLE_COST	The variable cost incurred per day and per unit of material when inventory is above target min and below target max and not equal to the days of supply target.	double	0.
TARGET_MAX_VARIABLE_COST	The variable cost incurred per day and per unit of material when inventory is above the maximal target inventory.	double	0.
INVENTORY_COST_FCT_TIME_STEP	This value is used to set the precision at which inventory costs are computed. When greater than 1, inventory costs are approximated by computing values only at times of the form `START_MIN`(`MODEL`) + `k` * `INVENTORY_COST_FCT_TIME_STEP`.	integer	0
SHELF_LIFE	The shelf life of the material, which is the number of time units before the material expires and is no longer consumable. For example, if the production of a material ends at `t-1`, then at `t - 1 + shelf life` the material is still consumable; at `t + shelf life` the material is no longer consumable. When defining maturity or shelf life, initial quantities must be expressed as procurements received in the past with a clear production date.	integer	+INF

MATURITY	The maturation time defined for this material. Maturation time is the minimal number of time units that must elapse after material production completes before that material can be consumed. For example, if the production of a material ends at `t-1`, then at `t + maturity` the material is consumable; at `t - 1 + maturity` the material is not yet consumable. When defining maturity or shelf life, initial quantities must be expressed as procurements received in the past with a clear production date.	integer	0
PRIMARY_UNIT_ID	This field is used to indicate the primary unit in which all computation of quantities of this material must be performed.	id	NULL
DISPLAY_UNIT_ID	This field is used to indicate the unit in which all quantities of this material must be displayed.	id	NULL
COLOR	The defined color for this material representation.	string	""
VISIBLE	The visibility of this material inventory.	boolean	1
PEGGING_STRATEGY	The pegging strategy to consider for all arcs of this material. Possible values are: `Static`, `DynamicEarliestEndMax`, and `DynamicFirstInFirstOut`.	id	"Static"
SERVICE_LEVEL_TYPE	This field sets the service level type of the material. The service level is used to determine the acceptable level of unsatisfied demand that may occur due to insufficient available stock or inventory. The default value of `Disabled` means that no service level stock computations are performed. Other possible values are `ServiceLevel`, `ServiceLevelDynamic`, `FillRate`, `FillRateDynamic`. See *Implementation of Plant PowerOps* for more information.	id	"Disabled"
TARGET_SERVICE_LEVEL	This field contains the desired service level. The semantics of this value depends on the service level type. If SERVICE_LEVEL_TYPE is set to `ServiceLevel`, the target level measures the probability of avoiding stock-out events. If SERVICE_LEVEL_TYPE is set to `FillRate`, the target level measures the desired average ratio of demand that can be satisfied.	double	0.95
DEMAND_VARIABILITY	This field contains a ratio expressing the uncertainty about demands for this material. It is used in computing safety stock values from the target service level. More precisely, demands are assumed to follow Gaussian (normal) law where this ratio is the relative standard deviation. Hence, a zero value for demand variability will disable service-level computations of safety stocks. It states basically that the demand is certain.	double	0.

AVERAGE_LEAD_TIME	This field contains the duration from the start of production until the actual shipping to the customer. It is used to compute a safety stock based on a target service level. The higher the lead time, the higher also will be the demand uncertainty used to compute the safety stock. A zero value for the average lead time will disable the service-level computation of safety stocks.	integer	0
LEAD_TIME_STD_DEVIATION	This field contains the lead time standard deviation used in service-level computation of safety stock. By default this field is zero, meaning that the lead time has no uncertainty. This value should not be greater than the lead time itself.	integer	0

PPO_MATERIAL_FAMILY

Synopsis

```
PPO_MATERIAL_FAMILY,MATERIAL_FAMILY_ID,NAME,TYPE,COLOR
```

Topic

Materials and Storage Units

Description

This table is used to create a material family. A material may be a member of multiple families.

Primary keys

MATERIAL_FAMILY_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_FAMILY_ID	This data is the mandatory unique identifier of the family.	id		mandatory
NAME	This data is the name describing the family.	string		mandatory
TYPE	This data is the family type this family belongs to.	id		mandatory
COLOR	This data is the family color.	string		""

PPO_MATERIAL_FAMILY_CARD_MAX

Synopsis

```
PPO_MATERIAL_FAMILY_CARD_MAX,MATERIAL_FAMILY_ID,BUCKET_SEQUENCE_ID,CARDINALITY_MAX
```

Topic

Materials and Storage Units

Description

This table is used to create a constraint that limits the number of materials produced in certain intervals of time. For each time bucket of the bucket sequence, the constraint sets an upper bound on the number of materials produced on the bucket. Only materials belonging to the family are considered.

Primary keys

MATERIAL_FAMILY_ID,BUCKET_SEQUENCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_FAMILY_ID	This data is the unique identifier of the material family. The constraint only impacts materials belonging to this family.	id		mandatory
BUCKET_SEQUENCE_ID	This data is the unique identifier of the bucket sequence used to describe time intervals on which the cardinality is limited.	id		mandatory
CARDINALITY_MAX	This data is the maximum number of produced materials on each bucket of the bucket sequence	integer		mandatory

PPO_MATERIAL_FAMILY_MATERIAL

Synopsis

```
PPO_MATERIAL_FAMILY_MATERIAL,MATERIAL_FAMILY_ID,MATERIAL_ID
```

Topic

Materials and Storage Units

Description

This table is used to identify a material as a member of a material family. A material may be a member of several families.

Primary keys

MATERIAL_FAMILY_ID,MATERIAL_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_FAMILY_ID	This data is the mandatory unique identifier of the family.	id		mandatory
MATERIAL_ID	This data is the mandatory unique identifier of the material belonging to the family.	id		mandatory

PPO_MATERIAL_PRODUCTION

Synopsis

```
PPO_MATERIAL_PRODUCTION,MATERIAL_ID,ACTIVITY_ID,PRODUCED_QUANTITY,CONTINUOUS,MODE_NUMBER,
STORAGE_ACTIVITY_ID,STORAGE_UNIT_ID,TIME_OFFSET,RATIO_MIN,RATIO_MAX,
MAX_NUMBER_OF_PEGGING_ARCS,VARIABLE_QUANTITY_PROTO,FIXED_QUANTITY_PROTO
```

Topic

Production Orders

Description

The PPO_MATERIAL_PRODUCTION table specifies which activity and mode instances produce and consume materials, in what quantity, and to/from which storage units. Also see the PPO_MATERIAL_PRODUCTION_PROTO table.

Primary keys

MATERIAL_ID,ACTIVITY_ID,MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material that is produced or consumed.	id		mandatory
ACTIVITY_ID	The identifier of the activity which produces or consumes the material.	id		mandatory
PRODUCED_QUANTITY	The quantity produced for the execution of one batch of the recipe. If the recipe consumes the material, a negative number is used.	double		0.0
CONTINUOUS	If false, the production or consumption is performed in batch. Otherwise it is assumed that production or consumption occurs continuously as the activity executes.	boolean		0
MODE_NUMBER	The number of the mode which produces or consumes the material. The special value -1 can be used if all the modes of the activity produce or consume the material in the same manner.	integer		-1
STORAGE_ACTIVITY_ID	The identifier of the storage activity which stores the material produced for exclusive storage units. The default value NULL can be used if only planning is applied, for consumption and if the storage unit is not exclusive.	id		NULL

STORAGE_UNIT_ID	Optional. The identifier of the storage unit in which the material is produced (or from which it is consumed). The special value NULL can be used if you do not want to manage storage units for the material under consideration.	id	NULL
TIME_OFFSET	This data provides an approximation for the planning module regarding the availability of the produced material after the recipe begins to execute.	integer	0
RATIO_MIN	This data sets the ratio minimum in the recipe for the invoking material production/ consumption. For example, setting RATIO_MIN to **0.25** will force the outgoing/ incoming quantity of the corresponding material to represent at least 25% of the total production/consumption. Note that if the ratio minimum is different than the ratio maximum in a recipe, then the recipe is considered to be a flexible recipe template for planning. All positive quantities on material productions must then have the same absolute value; likewise, all negative quantities on material productions must then have the same absolute value.	double	1.0
RATIO_MAX	This method sets the ratio maximum in the recipe for the invoking material production/consumption. For example, setting RATIO_MAX to **0.5** will force the outgoing/incoming quantity of the corresponding material to represent at most 50% of the total production/consumption. Note that if the ratio minimum is different than the ratio maximum in a recipe, then the recipe is considered to be a flexible recipe template for planning. All positive quantities on material productions must then have the same absolute value; likewise, all negative quantities on material productions must then have the same absolute value.	double	1.0
MAX_NUMBER_OF_PEGGING_ARCS	Ensures that, for a given production order, the number of pegging arcs for this material production or consumption does not exceed a maximal value.	integer	+INF
VARIABLE_QUANTITY_PROTO	The quantity produced for the execution of one unit of recipe as defined in the prototype.	double	0.0
FIXED_QUANTITY_PROTO	The quantity produced for the execution of one batch of the recipe as defined in the prototype.	double	0.0

PPO_MATERIAL_PRODUCTION_PROTO

Synopsis

```
PPO_MATERIAL_PRODUCTION_PROTO,MATERIAL_ID,ACTIVITY_ID,VARIABLE_QUANTITY,FIXED_QUANTITY,
CONTINUOUS,MODE_NUMBER,STORAGE_ACTIVITY_ID,STORAGE_UNIT_ID,TIME_OFFSET,RATIO_MIN,RATIO_MAX,
MAX_NUMBER_OF_PEGGING_ARCS
```

Topic

Recipes, Activities, and Modes

Description

The PPO_MATERIAL_PRODUCTION_PROTO table specifies which activity prototypes and modes produce and consume materials, in what quantity, and to/from which storage units. Material consumption is defined as a negative material production. Within each recipe, a material can only be produced or consumed by one activity; the activity may have multiple modes to produce the material but all must produce the same quantity. No other activities of that recipe may produce that same material but they may produce other materials.

Primary keys

MATERIAL_ID,ACTIVITY_ID,MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material that is produced or consumed.	id		mandatory
ACTIVITY_ID	The identifier of the activity which produces or consumes the material.	id		mandatory
VARIABLE_QUANTITY	The quantity produced for the execution of one unit of recipe. If the recipe consumes the material, a negative number is used.	double		0.0
FIXED_QUANTITY	The quantity produced for the execution of one batch of the recipe. If the recipe consumes the material, a negative number is used.	double		0.0
CONTINUOUS	If false, the production or consumption is performed in batch. Otherwise it is assumed that production or consumption occurs continuously as the activity executes.	boolean		0
MODE_NUMBER	The number of the mode which produces or consumes the material. The special value -1 can be used if all the modes of the activity produce or consume the material in the same manner.	integer		-1

STORAGE_ACTIVITY_ID	The identifier of the storage activity which stores the material produced for exclusive storage units. The default value NULL can be used if only planning is applied, for consumption and if the storage unit is not exclusive.	id	NULL
STORAGE_UNIT_ID	Optional. The identifier of the storage unit in which the material is produced (or from which it is consumed). The special value NULL can be used if you do not want to manage storage units for the material under consideration.	id	NULL
TIME_OFFSET	This data provides an approximation for the planning module regarding the availability of the produced material after the recipe begins to execute.	integer	0
RATIO_MIN	This data sets the ratio minimum in the recipe for the invoking material production/ consumption. For example, setting RATIO_MIN to **0.25** will force the outgoing/ incoming quantity of the corresponding material to represent at least 25% of the total production/consumption. Note that if the ratio minimum is different than the ratio maximum in a recipe, then the recipe is considered to be a flexible recipe template for planning. All positive quantities on material productions must then have the same absolute value; likewise, all negative quantities on material productions must then have the same absolute value.	double	1.0
RATIO_MAX	This method sets the ratio maximum in the recipe for the invoking material production/consumption. For example, setting RATIO_MAX to **0.5** will force the outgoing/incoming quantity of the corresponding material to represent at most 50% of the total production/consumption. Note that if the ratio minimum is different than the ratio maximum in a recipe, then the recipe is considered to be a flexible recipe template for planning. All positive quantities on material productions must then have the same absolute value; likewise, all negative quantities on material productions must then have the same absolute value.	double	1.0
MAX_NUMBER_OF_PEGGING_ARCS	Ensures that, for a given production order, the number of pegging arcs for this material production or consumption does not exceed a maximal value.	integer	+INF

PPO_MATERIAL_QUALITY

Synopsis

```
PPO_MATERIAL_QUALITY,MATERIAL_ID,QUALITY_ID,QUALITY_LEVEL_MIN,QUALITY_LEVEL_MAX
```

Topic

Materials and Storage Units

Description

The table PPO_MATERIAL_QUALITY defines the qualities of materials.

Primary keys

MATERIAL_ID,QUALITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material.	id		mandatory
QUALITY_ID	The identifier of the quality.	id		mandatory
QUALITY_LEVEL_MIN	The minimal value of the quality necessary for the material to qualify.	double		-1.#INF
QUALITY_LEVEL_MAX	The maximal value of the quality necessary for the material to qualify.	double		+1.#INF

PPO_MATERIAL_SECONDARY_UNIT

Synopsis

```
PPO_MATERIAL_SECONDARY_UNIT,MATERIAL_ID,SECONDARY_UNIT_ID,NUMERATOR,DENOMINATOR
```

Topic

Materials and Storage Units

Description

The table PPO_MATERIAL_SECONDARY_UNIT defines the conversion between the primary unit of a material and its secondary units. Example: Let the *Piece* be the primary unit of a material. Each *Piece* is composed of three *Cups*, and six *Pieces* make a *Box*. Then the conversion factor from primary unit (*Piece*) to *Cup* is defined by the ratio 3/1 and the conversion factor from primary unit (*Piece*) to *Box* is defined by the ratio 1/6.

Note that several materials may share the same unit with different conversions. The standard conversion of a unit of measure to its dimension is defined in the PPO_UNIT table.

Primary keys

MATERIAL_ID,SECONDARY_UNIT_ID

Field descriptions

Field Name	Description	Type	Range	Default
MATERIAL_ID	The identifier of the material; its primary unit is identified in the PPO_MATERIAL table.	id		mandatory
SECONDARY_UNIT_ID	The identifier of the secondary unit.	id		mandatory
NUMERATOR	This data participates in the conversion from the primary unit to the secondary unit. This conversion is defined by the ratio NUMERATOR/DENOMINATOR. See the example in the Description, above.	double		mandatory
DENOMINATOR	This data participates in the conversion from the primary unit to the secondary unit. This conversion is defined by the ratio NUMERATOR/DENOMINATOR.	double		mandatory

PPO_MODE

Synopsis

```
PPO_MODE,ACTIVITY_ID,MODE_NUMBER,NAME,RESOURCE_ID,REQUIRED_CAPACITY,PROCESSING_TIME_MIN,
PROCESSING_TIME_MAX,COST,START_MIN,START_MAX,END_MIN,END_MAX,UNPERFORMED_COST,
UNPERFORMED_SETUP_TIME,UNPERFORMED_SETUP_COST,LINE_ID,CALENDAR_ID,BREAK_DURATION_MAX,
MAX_END_DURATION_IN_BREAK,SHIFT_BREAKABLE,BATCH_SIZE_MIN,BATCH_SIZE_MAX,
FIX_PROCESSING_TIME_MIN_PROTO,FIX_PROCESSING_TIME_MAX_PROTO,VARIABLE_PROCESSING_TIME_PROTO,
FIXED_COST_PROTO,UNPERFORMED_COST_PROTO
```

Topic

Production Orders

Description

PPO_MODE contains the production activity modes of the production activity instances (see PPO_MODE_PROTO).

Primary keys

ACTIVITY_ID,MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of the activity to which the mode belongs.	id		mandatory
MODE_NUMBER	The index of the mode (0-based indexing) for the corresponding activity (prototype or generated).	integer		0
NAME	Optional name.	string		""
RESOURCE_ID	This data defines the primary resource required by the corresponding mode.	id		NULL
REQUIRED_CAPACITY	Defines the amount of capacity of the primary resource that is required to execute the activity in this mode.	integer		1
PROCESSING_TIME_MIN	The lower bound of the part of the processing time which is independent of the batch size.	integer		0
PROCESSING_TIME_MAX	The upper bound of the part of the processing time which is independent of the batch size.	integer		0

COST	The fixed processing cost (per batch) incurred by performing the activity in this mode.	double	0.
START_MIN	This data sets the earliest possible start time of the activity in this mode.	integer	-INF
START_MAX	This data sets the latest possible start time of the activity in this mode.	integer	+INF
END_MIN	This data sets the earliest possible end time of the activity in this mode.	integer	-INF
END_MAX	This data sets the latest possible end time of the activity in this mode.	integer	+INF
UNPERFORMED_COST	This data defines the cost of leaving the activity left "unperformed" but planned to be executed in this mode.	double	0.
UNPERFORMED_SETUP_TIME	This data sets to the given value the setup time of the activity left "unperformed" but planned to be executed in this mode.	integer	0
UNPERFORMED_SETUP_COST	This data sets to the given value the setup cost of the activity left "unperformed" but planned to be executed in this mode.	double	0.
LINE_ID	This data sets the line identifier of the corresponding mode. If not specified the mode has the line id of its primary resource.	id	NULL
CALENDAR_ID	The calendar to be used if the activity is in this mode. If this field is not specified, then the mode uses the calendar of its primary resource. Breaks and work periods are taken into account on calendars assigned to modes, but resource capacities are not. Calendars on resources do account for capacities.	id	NULL
BREAK_DURATION_MAX	The maximal break duration over which the activity in this mode cannot be interrupted by a break.	integer	+INF
MAX_END_DURATION_IN_BREAK	The maximal duration over which the activity in this mode can end in a break.	integer	0
SHIFT_BREAKABLE	This data sets the shift breakable flag for the activity in this mode. An activity that is "shift-breakable" can overlap a shift change. An activity that is not shift-breakable must be completely executed within a shift.	boolean	1
BATCH_SIZE_MIN	This data sets the minimal size of production orders using this mode; it restricts the PPO_RECIPE|BATCH_SIZE_MIN. Typically used with a single multimode recipe to constrain the allowable batch size for orders flowing through a particular path.	double	0.
BATCH_SIZE_MAX	This data sets the maximal size of production orders using this mode; it restricts the PPO_RECIPE|BATCH_SIZE_MAX. Typically used with a single multimode recipe to constrain the allowable batch size for orders flowing through a particular path.	double	+1.#INF

FIX_PROCESSING_TIME_MIN_PROTO	The lower bound of the part of the processing time which is independent of the batch size as defined in the mode prototype.	integer	0
FIX_PROCESSING_TIME_MAX_PROTO	The upper bound of the part of the processing time which is independent of the batch size as defined in the mode prototype.	integer	0
VARIABLE_PROCESSING_TIME_PROTO	The part of the processing time which is dependent on the batch size as defined in the mode prototype.	double	0.
FIXED_COST_PROTO	The fixed processing cost (per batch) incurred by performing the activity in this mode as defined in the mode prototype.	double	0.
UNPERFORMED_COST_PROTO	This data defines the cost (as defined in the mode prototype) of leaving the activity left "unperformed" but planned to be executed in this mode.	double	0.

PPO_MODE_PROTO

Synopsis

```
PPO_MODE_PROTO,ACTIVITY_ID,MODE_NUMBER,NAME,RESOURCE_ID,REQUIRED_CAPACITY,
FIXED_PROCESSING_TIME_MIN,FIXED_PROCESSING_TIME_MAX,FIXED_PROCESSING_TIME,
VARIABLE_PROCESSING_TIME,FIXED_COST,VARIABLE_COST,START_MIN,START_MAX,END_MIN,END_MAX,
UNPERFORMED_COST,UNPERFORMED_SETUP_TIME,UNPERFORMED_SETUP_COST,LINE_ID,CALENDAR_ID,
BREAK_DURATION_MAX,MAX_END_DURATION_IN_BREAK,SHIFT_BREAKABLE,BATCH_SIZE_MIN,BATCH_SIZE_MAX
```

Topic

Recipes, Activities, and Modes

Description

A mode is a way of performing an activity; for example, which resource is used, how long the activity takes, how much it costs to perform. Depending on the resource used, an activity might have a longer or a shorter processing time or a lower or higher cost.

The processing time consists of two parts: a fixed part independent of the batch size, and a variable part that depends upon the batch size. The effective processing time of a generated activity (for a given production order with a given batch size) is computed as follows: effective_processing_time = FIXED_PROCESSING_TIME(prototype) + VARIABLE_PROCESSING_TIME(prototype) * BATCH_SIZE. Note that the prototype FIXED_PROCESSING_TIME may vary between a given FIXED_PROCESSING_TIME_MIN and a given FIXED_PROCESSING_TIME_MAX.

Processing costs also have a fixed part and a variable part. The processing cost of a generated activity is: effective_processing_cost = FIXED_COST (prototype) + VARIABLE_COST(prototype) * BATCH_SIZE. The total processing cost (a criterion) is the sum, over all activities, of the cost of the selected modes. Modes of a one-shot activity (not associated with a recipe) shall have their variable processing time and cost equal to 0.

Primary keys

ACTIVITY_ID,MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of the activity to which the mode belongs.	id		mandatory
MODE_NUMBER	The index of the mode (0-based indexing) for the corresponding activity (prototype or generated).	integer		0
NAME	Optional name.	string		""

RESOURCE_ID	This data defines the primary resource required by the corresponding mode.	id	NULL
REQUIRED_CAPACITY	Defines the amount of capacity of the primary resource that is required to execute the activity in this mode.	integer	1
FIXED_PROCESSING_TIME_MIN	The lower bound of the part of the processing time which is independent of the batch size.	integer	0
FIXED_PROCESSING_TIME_MAX	The upper bound of the part of the processing time which is independent of the batch size.	integer	0
FIXED_PROCESSING_TIME	The part of the processing time which is independent of the batch size. Only used in CSV format.	integer	0
VARIABLE_PROCESSING_TIME	The part of the processing time which is dependent on the batch size.	double	0.
FIXED_COST	The fixed processing cost (per batch) incurred by performing the activity in this mode.	double	0.
VARIABLE_COST	The variable processing cost (per unit) incurred by performing the activity in this mode.	double	0.
START_MIN	This data sets the earliest possible start time of the activity in this mode.	integer	-INF
START_MAX	This data sets the latest possible start time of the activity in this mode.	integer	+INF
END_MIN	This data sets the earliest possible end time of the activity in this mode.	integer	-INF
END_MAX	This data sets the latest possible end time of the activity in this mode.	integer	+INF
UNPERFORMED_COST	This data defines the cost of leaving the activity left "unperformed" but planned to be executed in this mode.	double	0.
UNPERFORMED_SETUP_TIME	This data sets to the given value the setup time of the activity left "unperformed" but planned to be executed in this mode.	integer	0
UNPERFORMED_SETUP_COST	This data sets to the given value the setup cost of the activity left "unperformed" but planned to be executed in this mode.	double	0.
LINE_ID	This data sets the line identifier of the corresponding mode. If not specified the mode has the line id of its primary resource.	id	NULL
CALENDAR_ID	The calendar to be used if the activity is in this mode. If this field is not specified, then the mode uses the calendar of its primary resource. Breaks and work periods are taken into account on calendars assigned to modes, but resource capacities are not. Calendars on resources do account for capacities.	id	NULL
BREAK_DURATION_MAX	The maximal break duration over which the activity in this mode cannot be interrupted by a break.	integer	+INF
MAX_END_DURATION_IN_BREAK	The maximal duration over which the activity in this mode can end in a break.	integer	0

SHIFT_BREAKABLE	This data sets the shift breakable flag for the activity in this mode. An activity that is "shift-breakable" can overlap a shift change. An activity that is not shift-breakable must be completely executed within a shift.	boolean	1
BATCH_SIZE_MIN	This data sets the minimal size of production orders using this mode; it restricts the PPO_RECIPE|BATCH_SIZE_MIN. Typically used with a single multimode recipe to constrain the allowable batch size for orders flowing through a particular path.	double	0.
BATCH_SIZE_MAX	This data sets the maximal size of production orders using this mode; it restricts the PPO_RECIPE|BATCH_SIZE_MAX. Typically used with a single multimode recipe to constrain the allowable batch size for orders flowing through a particular path.	double	+1.#INF

PPO_MODEL

Synopsis

```
PPO_MODEL,NAME,TIME_UNIT,DATE_ORIGIN,INT_DATE_ORIGIN,START_MIN,END_MAX,
CURRENT_OPTIMIZATION_PROFILE,TIME_ZONE,TIME_CHECKING_TOLERANCE,BUCKET_SEQUENCE_ID
```

Topic

General tables

Description

The PPO_MODEL table groups general parameters of the problem, such as time unit and horizon information.

Primary keys

NAME

Field descriptions

Field Name	Description	Type	Range	Default
NAME	The problem instance name.	string		"NoName"
TIME_UNIT	This data defines the time unit, expressed in seconds, used throughout the model. For example, a time unit of 60 means that an activity of duration 1 is one minute in length.	integer		1
DATE_ORIGIN	This field defines the date, expressed in "YYYY-MM-DD HH:MM:SS" format, that corresponds to the time 0 (zero) used in the model. The given date is understood to be in UTC. For example, "2001-02-01 00:00:00" means that time 0 corresponds to February 1st, 2001, at midnight 00:00:00 in 24-hour clock format. All relative times (given as integer) are interpreted with respect to this date origin. This field is omitted if INT_DATE_ORIGIN is not null.	string		"2001-01-01 00:00:00"

INT_DATE_ORIGIN	This field defines the date, expressed in seconds since January 1st 2001, that corresponds to the time 0 (zero) used in the model. The given date is understood to be in the defined time zone. For example, 31 * 24 * 3600 means that time 0 corresponds to February 1st, 2001, at midnight 00:00:00 in 24-hour clock format. Alternatively, you can use the DATE_ORIGIN column to provide the origin date as a string. When provided, this field overwrites the value of the DATE_ORIGIN field.	integer	0
START_MIN	This data defines a common earliest start time associated with all the activities of the model, except those known to have already started.	integer	-INF
END_MAX	This data defines a common latest end time associated with all the activities of the model.	integer	+INF
CURRENT_OPTIMIZATION_PROFILE	This data sets the current optimization profile to consider for retrieving default values of engine parameters such as criterion weights.	id	"NoProfile"
TIME_ZONE	This data sets the time zone code. There are hundreds of available choices; please see the *Date and time display* in the *Reference Documentation* section for a complete listing.	string	"UTC"
TIME_CHECKING_TOLERANCE	This data defines the tolerance expressed in time units used for filtering nonfatal error messages (with regard to time and capacity violations in the checkers).	integer	0
BUCKET_SEQUENCE_ID	This data defines the bucket sequence used by the optimizer.	id	"NULL"

PPO_OPTIMIZATION_PROFILE

Synopsis

```
PPO_OPTIMIZATION_PROFILE,OPTIMIZATION_PROFILE_ID,NAME,PLANNING_HORIZON,SCHEDULING_HORIZON,
PLANNING_TIME_LIMIT,SCHEDULING_TIME_LIMIT,REBALANCING_TIME_LIMIT,PLANNING_ALGORITHM,
BATCHING_ALGORITHM,PLANNING_REQUIRED,BATCHING_REQUIRED,SCHEDULING_REQUIRED,
REBALANCING_REQUIRED,SCOPE_ID
```

Topic

General tables

Description

The PPO_OPTIMIZATION_PROFILE table defines a set of profiles, each of which contains a set of parameters to tune the optimization engines.

Primary keys

OPTIMIZATION_PROFILE_ID

Field descriptions

Field Name	Description	Type	Range	Default
OPTIMIZATION_PROFILE_ID	The mandatory unique identifier of the optimization profile under consideration.	id		mandatory
NAME	The optimization profile name.	string		mandatory
PLANNING_HORIZON	This data sets the end time (horizon) to consider for production planning. Let **E** be the end of the latest bucket that ends exactly at or before the planning horizon. The planning engine generates only planned deliveries and corresponding planned productions for demands whose earliest possible delivery can start before **E** > (the DELIVERY_START_MIN is strictly less than **E**). Demands that are to be delivered after **E** (DELIVERY_START_MIN is greater than or equal to **E**) are taken into account in the inventory corridor by forcing the planning engine to create enough planned productions to stay between the minimum and the maximum number of days of supply at the end of the horizon.	integer		+INF

SCHEDULING_HORIZON	This field sets the end time of the scheduling horizon, and thereby affects the batching engine, scheduling engine, and GUI behavior. The batching engine does not consider planned productions or planned deliveries that start in time buckets equal to or after the scheduling horizon. Therefore, such planning information is not converted into production orders and material flow arcs, and no corresponding activities are created. In the scheduling solution, the nondelivery cost for a demand that has its delivery end max greater than this horizon is computed based on the pegged quantity, not on the quantity of the demand itself. Integrated planning and scheduling problems (where both planning and scheduling are required) have a further consideration regarding solution display in the GUI. If the scheduling horizon is less than the planning horizon, then for time periods before the scheduling horizon the solution data is based on the scheduling solution. For time periods after the scheduling horizon, the planning solution data (planned productions and planned deliveries) are displayed in the appropriate views of the GUI.	integer	+INF
PLANNING_TIME_LIMIT	This data sets the number of seconds allocated to the planning engine.	double	10.0
SCHEDULING_TIME_LIMIT	This data sets the number of seconds allocated to the scheduling engine.	double	10.0
REBALANCING_TIME_LIMIT	This data sets the number of seconds allocated to the material rebalancing engine.	double	10.0
PLANNING_ALGORITHM	This field sets the planning algorithm used by the planning engine. The possibilities are: `Automatic`, `OnePass` or `MultiPass`. Planning algorithms are useful to deal with infeasibilities in input data, providing different approaches to deal with problems due to the balance of intermediate materials using inflow recipes or waste recipes. `OnePass` solves a single mathematical programming model with a single weighted objective function, combining business objectives and "technical" costs. With this approach, the costs of processing these recipes must be carefully computed so that the optimizer uses them only as a fallback position, in case of infeasibility. The drawback of this approach is the potentially wide range in numerical values that coexist during optimization, which can lead to numerical stability issues. The business objectives could get diluted in an objective function containing high technical costs. A bad solution with respect to the business objective may result if the relative gap limit stops optimization when the solution is "good enough."	id	"OnePass"

	`MultiPass` uses goal programming to deal with infeasible material flow. The business objective is kept separate from technical costs such as inflow and waste recipe costs or nondelivery costs. This approach assumes it is always better to deliver a product, as compared to other considerations. The value `Automatic` lets PPO decide which of the two algorithms is used.		
BATCHING_ALGORITHM	This field sets the batching algorithm the solve procedure uses if batching is required. The possibilities are: `Heuristic`, `AdvancedHeuristic`, `ConstraintBased`, or `Automatic`. Automatic tries successively the previous engines, until one succeeds.	id	"ConstraintBased"
PLANNING_REQUIRED	This data expresses the need for a planning optimization with the planning engine.	boolean	1
BATCHING_REQUIRED	This data expresses the need for a batching and pegging pass with the batching engine.	boolean	1
SCHEDULING_REQUIRED	This data expresses the need for a scheduling optimization with the scheduling engine.	boolean	1
REBALANCING_REQUIRED	This data expresses the need for calling the material rebalancing engine.	boolean	0
SCOPE_ID	The identifier of the scope.	id	NULL

PPO_PLANNING_CRITERION_WEIGHT

Synopsis

```
PPO_PLANNING_CRITERION_WEIGHT,OPTIMIZATION_PROFILE_ID,CRITERION_ID,WEIGHT
```

Topic

General tables

Description

The PPO_PLANNING_CRITERION_WEIGHT table defines the global weights of the directly optimizable Key Performance Indicators (KPIs) for the planning engine. These weights can be associated with different optimization profiles. Key Performance Indicators and optimization criteria are discussed in detail in the section *Key performance indicators*.

Primary keys

OPTIMIZATION_PROFILE_ID,CRITERION_ID

Field descriptions

Field Name	Description	Type	Range	Default
OPTIMIZATION_PROFILE_ID	The optimization profile under consideration.	id		"NoProfile"
CRITERION_ID	The criterion. The possible values are: TotalRevenue, TotalNonDeliveryCost, TotalResourceCost, TotalIdleCost, TotalProcessingCost, TotalSetupCost, TotalEarlinessCost, TotalTardinessCost, TotalInventoryCost (excess), TotalInventoryDeficitCost, TotalWasteCost.	id		mandatory
WEIGHT	The weight assigned to the criterion.	double		0.

PPO_PROCUREMENT_TO_DEMAND_ARC

Synopsis

```
PPO_PROCUREMENT_TO_DEMAND_ARC,NAME,FROM_PROCUREMENT_ID,TO_DEMAND_ID,QUANTITY,
FIRM_QUANTITY_MIN,FIRM_QUANTITY_MAX
```

Topic

Material Flows

Description

The PPO_PROCUREMENT_TO_DEMAND_ARC table describes pegging arcs from procurements to demands.

Primary keys

FROM_PROCUREMENT_ID,TO_DEMAND_ID

Field descriptions

Field Name	Description	Type	Range	Default
NAME	Optional name.	string		""
FROM_PROCUREMENT_ID	The mandatory procurement origin node.	id		mandatory
TO_DEMAND_ID	The mandatory demand destination node.	id		mandatory
QUANTITY	The quantity of material that is moved.	double		mandatory
FIRM_QUANTITY_MIN	When not equal to `0`, this data specifies that the pegging arc can not be abandoned and that its quantity can not be reduced below the given value.	double		0.
FIRM_QUANTITY_MAX	When not equal to `+INF`, this data specifies whether the quantity on the arc can be increased and up to what limit.	double		+1.#INF

PPO_PROCUREMENT_TO_PROD_ARC

Synopsis

```
PPO_PROCUREMENT_TO_PROD_ARC,NAME,MATERIAL_ID,QUANTITY,FIRM_QUANTITY_MIN,FIRM_QUANTITY_MAX,
FROM_PROCUREMENT_ID,TO_PRODUCTION_ORDER_ID
```

Topic

Material Flows

Description

The PPO_PROCUREMENT_TO_PROD_ARC table describes pegging arcs from procurements to production orders. Material flow arcs can be associated with more than one batching solution, thus representing alternative production flows; however, you must commit one of these batching solutions prior to proceeding with the scheduling.

Primary keys

MATERIAL_ID,FROM_PROCUREMENT_ID,TO_PRODUCTION_ORDER_ID

Field descriptions

Field Name	Description	Type	Range	Default
NAME	Optional name.	string		""
MATERIAL_ID	The identifier of the material.	id		mandatory
QUANTITY	The quantity of material that is moved.	double		mandatory
FIRM_QUANTITY_MIN	When not equal to `0`, this data specifies that the pegging arc can not be abandoned and that its quantity can not be reduced below the given value.	double		0.
FIRM_QUANTITY_MAX	When not equal to `+INF`, this data specifies whether the quantity on the arc can be increased and up to what limit.	double		+1.#INF
FROM_PROCUREMENT_ID	Mandatory procurement origin node.	id		mandatory
TO_PRODUCTION_ORDER_ID	Mandatory production order destination node.	id		mandatory

PPO_PROD_ORDER_PLANNED_MODE

Synopsis

`PPO_PROD_ORDER_PLANNED_MODE,PRODUCTION_ORDER_ID,ACTIVITY_ID,MODE_NUMBER`

Topic

Production Orders

Description

The PPO_PRODUCTION_ORDER_PLANNED_MODE table records the choice of the planning module with respect to alternative modes for a given production order coming from a planned production.

Primary keys

PRODUCTION_ORDER_ID,ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
PRODUCTION_ORDER_ID	This data is the mandatory identifier of the production order instance.	id		mandatory
ACTIVITY_ID	The identifier of the activity prototype of the recipe to which the production order belongs.	id		mandatory
MODE_NUMBER	The index of the mode prototype in the activity prototype of the recipe chosen by the planning module.	integer		mandatory

PPO_PROD_PROD_COMPAT

Synopsis

`PPO_PROD_PROD_COMPAT,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE`

Topic

Production Orders

Description

The PPO_PROD_PROD_COMPAT table imposes constraints between the execution of different activity instances. See the PPO_PROD_PROD_COMPAT_PROTO table for more details.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of another production activity.	id		mandatory
TYPE	The type of compatibility constraint. See the PPO_PROD_PROD_COMPAT_PROTO table for possible values.	id		mandatory

PPO_PROD_PROD_COMPAT_PROTO

Synopsis

`PPO_PROD_PROD_COMPAT_PROTO,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE`

Topic

Recipes, Activities, and Modes

Description

The PPO_PROD_PROD_COMPAT_PROTO table imposes compatibility constraints between the execution of different activities. This constraint enables you to enforce that two given activities will be executed in compatible modes. When one states that two activities must execute on connected primary resources (type `ConnectedPrimaryResources`) then the optimizer will enforce that their respective modes refer to primary resources for which a connection has been defined. To specify the connections use the PPO_RESOURCE_CONNECTION table. When one states that two activities must execute on same line (type `SameLineId`) then the optimizer will choose modes that share a common line id. See the PPO_MODE_PROTO table for information on defining a line id.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of another production activity.	id		mandatory
TYPE	The type of compatibility constraint. The possible types are: `ConnectedPrimaryResources`, `SameLineId`, `SamePrimaryResource`, `SamePrimaryResourceAndCapacity`, `SamePerformedStatus`, `DifferentPerformedStatus`, `PerformedImpliesPerformed`, `UnperformedImpliesUnperformed`, `PerformedImpliesUnperformed`, and `UnperformedImpliesPerformed`.	id		mandatory

PPO_PROD_PROD_PRECED

Synopsis

```
PPO_PROD_PROD_PRECED,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Production Orders

Description

The PPO_PROD_PROD_PRECED table is used to create precedence constraints between production activity instances (see PPO_PROD_PROD_PRECED_PROTO) .

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_PROD_PROD_PRECED_PROTO

Synopsis

`PPO_PROD_PROD_PRECED_PROTO,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX`

Topic

Recipes, Activities, and Modes

Description

The PPO_PROD_PROD_PRECED_PROTO table is used to create precedence constraints between production activities. Four types of precedence constraints are distinguished: Start-to-start constraints relate the start times of the two activities; start-to-end constraints relate the start time of the predecessor with the end time of the successor; end-to-start constraints relate the end time of the predecessor with the start time of the successor; and end-to-end constraints relate the end times of the two activities.

You can define positive minimum (DELAY_MIN) and maximum (DELAY_MAX) delays between the relevant start or end points of the two activities. These delays are specified in the TIME_UNIT of the model. The special values -INF and +INF indicate that the corresponding delay constraint does not apply. In particular, when the minimum delay constraint does not apply, the relevant time point of the successor is not forced to follow the relevant time point of the predecessor.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor activity of the precedence constraint.	id		mandatory
TYPE	This data defines the type of the corresponding precedence constraint: `EndToStart`, `StartToStart`, `EndToEnd`, or `StartToEnd`.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_PROD_SETUP_COMPAT

Synopsis

```
PPO_PROD_SETUP_COMPAT,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Production Orders

Description

The PPO_PROD_SETUP_COMPAT table is similar to the PPO_PROD_PROD_COMPAT_PROTO table and is used to impose constraints between the execution of production and setup activities. See PPO_PROD_PROD_COMPAT_PROTO for more information.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of a setup activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_PROD_SETUP_COMPAT_PROTO

Synopsis

```
PPO_PROD_SETUP_COMPAT_PROTO,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Recipes, Activities, and Modes

Description

The PPO_PROD_SETUP_COMPAT_PROTO table is similar to the PPO_PROD_PROD_COMPAT_PROTO table and is used to impose constraints between the execution of production and setup activities. See PPO_PROD_PROD_COMPAT_PROTO for more information.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of a setup activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_PROD_SETUP_PRECED

Synopsis

```
PPO_PROD_SETUP_PRECED,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Production Orders

Description

The PPO_PROD_SETUP_PRECED table is similar to the PPO_PROD_PROD_PRECED table and used to create precedence constraints between production and setup activity instances.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor production activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_PROD_SETUP_PRECED_PROTO

Synopsis

```
PPO_PROD_SETUP_PRECED_PROTO,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Recipes, Activities, and Modes

Description

The PPO_PROD_SETUP_PRECED_PROTO table is similar to the PPO_PROD_PROD_PRECED_PROTO table and used to create precedence constraints between production and setup activity prototypes.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor production activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_PROD_TO_DEMAND_ARC

Synopsis

```
PPO_PROD_TO_DEMAND_ARC,NAME,FROM_PRODUCTION_ORDER_ID,TO_DEMAND_ID,QUANTITY,FIRM_QUANTITY_MIN,
FIRM_QUANTITY_MAX
```

Topic

Material Flows

Description

The PPO_PROD_TO_DEMAND_ARC table describes pegging arcs from production orders to demands.

Primary keys

FROM_PRODUCTION_ORDER_ID,TO_DEMAND_ID

Field descriptions

Field Name	Description	Type	Range	Default
NAME	Optional name.	string		""
FROM_PRODUCTION_ORDER_ID	The production order origin node.	id		mandatory
TO_DEMAND_ID	The demand destination node.	id		mandatory
QUANTITY	The quantity of material that is moved.	double		mandatory
FIRM_QUANTITY_MIN	When not equal to `0`, this data specifies that the pegging arc can not be abandoned and that its quantity can not be reduced below the given FIRM_QUANTITY_MIN.	double		0.
FIRM_QUANTITY_MAX	When not equal to `+INF`, this data specifies whether the quantity on the arc can be increased and up to what limit.	double		+1.#INF

PPO_PROD_TO_PROD_ARC

Synopsis

```
PPO_PROD_TO_PROD_ARC,NAME,MATERIAL_ID,QUANTITY,FIRM_QUANTITY_MIN,FIRM_QUANTITY_MAX,
FROM_PRODUCTION_ORDER_ID,TO_PRODUCTION_ORDER_ID,START_CONSUMPTION_COEFF,END_CONSUMPTION_COEFF
```

Topic

Material Flows

Description

The PPO_PROD_TO_PROD_ARC table describes pegging arcs between production orders. Material flow arcs can be associated with more than one batching solution, thus representing alternative production flows; however, you must commit one of these batching solutions prior to proceeding with the scheduling.

Primary keys

MATERIAL_ID,FROM_PRODUCTION_ORDER_ID,TO_PRODUCTION_ORDER_ID,START_CONSUMPTION_COEFF,END_CONSUMPTION_COEFF

Field descriptions

Field Name	Description	Type	Range	Default
NAME	Optional name.	string		""
MATERIAL_ID	This is the identifier of the material.	id		mandatory
QUANTITY	The quantity of material that is moved.	double		mandatory
FIRM_QUANTITY_MIN	When not equal to `0`, this data specifies that the pegging arc can not be abandoned and that its quantity can not be reduced below the given value.	double		0.
FIRM_QUANTITY_MAX	When not equal to `+INF`, this data specifies whether the quantity on the arc can be increased and up to what limit.	double		+1.#INF
FROM_PRODUCTION_ORDER_ID	Production order origin node.	id		mandatory
TO_PRODUCTION_ORDER_ID	Production order destination node.	id		mandatory
START_CONSUMPTION_COEFF	The start consumption coefficient.	double		0.
END_CONSUMPTION_COEFF	The end consumption coefficient.	double		1.

PPO_PRODUCTION_ORDER

Synopsis

```
PPO_PRODUCTION_ORDER,PRODUCTION_ORDER_ID,NAME,RECIPE_ID,BATCH_SIZE,EDITABLE,FIRMING_EDITABLE,
FIRM_BATCH_SIZE_MIN,FIRM_BATCH_SIZE_MAX,START_MIN,END_MAX,PLANNED_TIME_MIN,PLANNED_TIME_MAX,
EXPIRATION_TIME,COMMENTS
```

Topic

Production Orders

Description

The PPO_PRODUCTION_ORDER table is used to represent a production order implementing the recipe of a process. The order has a multiplication factor regarding the recipe called the batch size, which is used to compute the quantities of materials produced or consumed by the production order. The batch size is also used to adjust the processing times and costs of activities (if variable processing times and costs have been specified on the modes of the activity prototypes of the recipe).

A production order producing finished goods may partially or wholly satisfy a single demand, or may satisfy several demands. A production order producing only intermediates does not satisfy demand. The material flow between production orders can be modeled by adding material flow arcs between production orders. Production orders can be associated with more than one batching solution, thus representing alternative production flows; however, you must commit one of these batching solutions prior to proceeding with the scheduling.

Note when using a firm batch size: A firm production order must be scheduled in order to be taken into account in the planning module (a scheduled activity must exist for each of its activities). The planning module will not try to move it. The planning module will treat a firm order as a fixed order, as opposed to the scheduling module which may try to move it.

Primary keys

PRODUCTION_ORDER_ID

Field descriptions

Field Name	Description	Type	Range	Default
PRODUCTION_ORDER_ID	This data is the mandatory unique identifier of the production order.	id		mandatory
NAME	Optional name.	string		""
RECIPE_ID	The recipe the production order implements.	id		mandatory

BATCH_SIZE	The quantity of recipe to be executed by this production order.	double	mandatory
EDITABLE	If true, then changing the firm batch sizes interactively and changing its activities are allowed.	boolean	1
FIRMING_EDITABLE	If true, then changing the firm batch sizes interactively is allowed.	boolean	1
FIRM_BATCH_SIZE_MIN	When not equal to `0`, This data specifies that the production order can not be abandoned and that its batch size can not be reduced below the given FIRM_BATCH_SIZE_MIN. This occurs when the recipe execution is started, or more generally when actions have already been taken to execute the production order in the factory.	double	0.
FIRM_BATCH_SIZE_MAX	When the FIRM_BATCH_SIZE_MAX is not equal to the default value of `+INF`, it is used to specify up to what limit the batch size can be increased. In most cases, the FIRM_BATCH_SIZE_MIN and the FIRM_BATCH_SIZE_MAX will both be equal to the batch size as soon as action has been taken to execute the production order in the factory.	double	+1.#INF
START_MIN	This data defines the earliest start time of this production order.	integer	-INF
END_MAX	This data defines the latest end time of this production order.	integer	+INF
PLANNED_TIME_MIN	An approximate minimal time at which the execution of the production order could start.	integer	-INF
PLANNED_TIME_MAX	An approximate maximal time at which the execution of the production order is supposed to be done.	integer	+INF
EXPIRATION_TIME	The expiration time of the lot produced by the production order.	integer	-INF
COMMENTS	Optional comment.	string	""

PPO_PRODUCTION_ORDER_ACTIVITY

Synopsis

```
PPO_PRODUCTION_ORDER_ACTIVITY,PRODUCTION_ORDER_ID,ACTIVITY_PROTOTYPE_ID,ACTIVITY_ID
```

Topic

Production Orders

Description

The PPO_PRODUCTION_ORDER_ACTIVITY table is used to link generated activities to the production order they belong to.

Primary keys

PRODUCTION_ORDER_ID,ACTIVITY_PROTOTYPE_ID

Field descriptions

Field Name	Description	Type	Range	Default
PRODUCTION_ORDER_ID	The identifier of a production order.	id		mandatory
ACTIVITY_PROTOTYPE_ID	The identifier of an activity prototype belonging to the recipe of the production order.	id		mandatory
ACTIVITY_ID	The identifier of the corresponding one-shot activity, generated for the production order under consideration.	id		mandatory

PPO_QUALITY

Synopsis

```
PPO_QUALITY,QUALITY_ID,NAME
```

Topic

Materials and Storage Units

Description

The PPO_QUALITY table defines the different qualities that materials may have.

Primary keys

QUALITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
QUALITY_ID	The identifier of the quality.	id		mandatory
NAME	Optional name.	string		""

PPO_RECIPE

Synopsis

```
PPO_RECIPE,RECIPE_ID,NAME,RECIPE_TYPE,SPLIT_BATCH_SIZE_MIN,BATCH_SIZE_MIN,BATCH_SIZE_MAX,
START_MIN,END_MAX,CONSTANT_BATCH_SIZE,INTEGER_BATCH_SIZE,PRIMARY_PRODUCT_ID,
PRIMARY_INGREDIENT_ID,FLEXIBLE_INSTANCE,ALLOCATION_WEIGHT,CAMPAIGN_NB_ORDERS_MAX,
CAMPAIGN_CYCLE
```

Topic

Recipes, Activities, and Modes

Description

The PPO_RECIPE table is used to create recipes. A recipe object models a production process and is composed of activity prototypes. It contains a template of activity prototypes and constraints that can be used as a mold to generate a set of dependent activities. To effectively use a recipe, production orders must be instantiated.

Recipes are also used to model resource cleanups. When a recipe occurs in the PPO_RESOURCE_CLEANUP_STATE table (CLEANUP_RECIPE_ID field), then it is considered to be a cleanup recipe. Note the additional constraints on cleanup recipes: the recipe must have only one activity, and it must have a mode on the RESOURCE referenced in the same line of PPO_RESOURCE_CLEANUP_STATE table.

Primary keys

RECIPE_ID

Field descriptions

Field Name	Description	Type	Range	Default
RECIPE_ID	This data is the mandatory unique identifier of the recipe.	id		mandatory
NAME	Optional name.	string		""

RECIPE_TYPE	This data tags the corresponding recipe as one of the following types: `Fixed`, `Transport`, `Automatic`, `Make`, `Order`, or `Undefined`. If the type is `Fixed` then the recipe can be used only in manual mode; the planning engine will not be allowed to use it, and only already-existing fixed production orders or orders created interactively can use this recipe. If the type is `Transport`, then it means that consuming an SKU and producing another is just a matter of transportation of the same item to a different location. This is used in the GUI to compute the stock in transit (Stock Coverage view) and the transported quantity (Warehouse Summary view). The value `Automatic` tags the technical waste and inflow recipes created and deleted by PPO to deal with infeasibility. The other values are purely informative but can be used with plug-ins or in future versions of PPO. The type `Order` can be used for recipes simulating a good or material ordered from outside, typically acquired from a third party, as opposed to the type `Make` indicating an in-house production. The "make or order" decision can be taken by the planning engine based on capacity of the plant and recipe costs. Note that if the quantity and receipt date of procured goods are already known, no recipe has to be defined, only the procurements.	id	"Undefined"
SPLIT_BATCH_SIZE_MIN	This data sets the minimal possible size of production orders when splitting cross-bucket orders using the splitCrossBucketOrders API. If not defined then the BATCH_SIZE_MIN is used.	double	0.
BATCH_SIZE_MIN	This data sets the minimal size of production orders implementing this recipe.	double	0.
BATCH_SIZE_MAX	This data sets the maximal size of production orders implementing this recipe.	double	+1.#INF
START_MIN	This data sets a time before which the recipe cannot be used.	integer	-INF
END_MAX	This data sets a time after which the recipe cannot be used.	integer	+INF
CONSTANT_BATCH_SIZE	This data states whether PPO shall attempt to generate production orders with the same batch size, especially if these production orders follow each other in time. It is used as a strong preference.	boolean	0
INTEGER_BATCH_SIZE	This data states whether PPO shall attempt to generate production orders with integer batch size. It is used as a strong preference. If one unit of a recipe represents the production of one pallet and if fractional pallets are not allowed then this parameter must be set to true. Note that setting this parameter increases the complexity of the optimization problem to be solved.	boolean	0
PRIMARY_PRODUCT_ID	This data indicates the main material produced by the recipe. The GUI will display the batch size of production orders implementing this recipe using the display unit of this main material. If no primary material is indicated the GUI will display the batch sizes in recipe quantity.	id	NULL

PRIMARY_INGREDIENT_ID	This data indicates the main material consumed by the recipe. The GUI will display consumption of this ingredient in the inspector of the production order belonging to this recipe.	id	NULL
FLEXIBLE_INSTANCE	This data indicates that this recipe comes from the instantiation of a flexible recipe after solving a blending problem with the planning module.	boolean	0
ALLOCATION_WEIGHT	This data defines the allocation weight. When a material can be produced by using alternative recipes, this allocation weight controls distribution of the dependent demand for the semi-finished product across the different routings.	double	1.
CAMPAIGN_NB_ORDERS_MAX	This data sets an upper bound on the throughput of the process. It sets an upper bound on the number of production orders per time interval (the time interval to use is set by the CAMPAIGN_CYCLE field value). By default, this field has infinite value, that is, the constraint is disabled. This constraint is enforced by the planning engine only.	integer	+INF
CAMPAIGN_CYCLE	This data defines the cycle time of the maximum orders constraints, as described in field CAMPAIGN_NB_ORDERS_MAX. By default, this cycle time is null, meaning the constraint is disabled.	integer	0

PPO_RECIPE_FAMILY

Synopsis

```
PPO_RECIPE_FAMILY,RECIPE_FAMILY_ID,NAME,TYPE
```

Topic

Recipes, Activities, and Modes

Description

This table is used to create a recipe family. A recipe family can be used to group recipes together. A recipe may be a member of a several families.

Primary keys

RECIPE_FAMILY_ID

Field descriptions

Field Name	Description	Type	Range	Default
RECIPE_FAMILY_ID	This data is the mandatory unique identifier of the recipe family.	id		mandatory
NAME	The name of the recipe family.	string		""
TYPE	This data is the family type this family belongs to.	id		mandatory

PPO_RECIPE_FAMILY_FILTER

Synopsis

`PPO_RECIPE_FAMILY_FILTER,SCOPE_ID,RECIPE_FAMILY_ID,STATUS`

Topic

Recipes, Activities, and Modes

Description

This table defines the status of a recipe family in a given scope. A recipe family status can be: `Undefined`, `Frozen` or `Planned`. If the type is `Frozen` then recipes of this family and its production orders are fixed into the submodel. If the type is `Planned` then recipes of this family and its production orders are planned into the submodel.

Primary keys

SCOPE_ID,RECIPE_FAMILY_ID

Field descriptions

Field Name	Description	Type	Range	Default
SCOPE_ID	The identifier of the scope.	id		mandatory
RECIPE_FAMILY_ID	The identifier of the recipe family.	id		mandatory
STATUS	The status of the recipe family in this scope. This data should have one of the following values: `Undefined`, `Frozen` or `Planned`.	id		"Undefined"

PPO_RECIPE_RECIPE_FAMILY

Synopsis

```
PPO_RECIPE_RECIPE_FAMILY,RECIPE_FAMILY_ID,RECIPE_ID
```

Topic

Recipes, Activities, and Modes

Description

This table is used to associate a recipe with a recipe family. A recipe may be a member of a several families.

Primary keys

RECIPE_FAMILY_ID,RECIPE_ID

Field descriptions

Field Name	Description	Type	Range	Default
RECIPE_FAMILY_ID	This data is the mandatory unique identifier of the recipe family.	id		mandatory
RECIPE_ID	The identifier of the recipe.	id		mandatory

PPO_RESOURCE

Synopsis

```
PPO_RESOURCE,RESOURCE_ID,NAME,CAPACITY,CALENDAR_ID,START_MIN,TIME_FENCE,END_MAX,LINE_ID,
PLAN_CAPACITY_REDUCTION_FACTOR,VISIBLE,DISPLAY_RANK,CATEGORY,X,Y
```

Topic

Manufacturing Resources

Description

The PPO_RESOURCE table is used to model machines, tools, vehicles, equipment or workers. Do not use PPO_RESOURCE to model materials being consumed or produced; use PPO_MATERIAL for that purpose.

Resources can be aggregated together into larger groups; one reason to do so is to make solving the planning problem easier. Use of *super resources* allows the planning engine to consider within each time bucket the grouped capacity of all the resources of a super resource. Note that super resources must group only similar resources, which typically have identical connectivity and make the same products. To define super resources in order to lighten the planning problem, use the PPO_RESOURCE_HIERARCHY table.

You can also group resources into *resource families*. This allows you to tailor and organize resource data for reporting or visibility purposes (for example, to improve visibility on the Gantt Diagram). To create resource families, use the PPO_RESOURCE_FAMILY and PPO_RESOURCE_RESOURCE_FAMILY tables.

Primary keys

RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	This data is the mandatory unique identifier of the resource.	id		mandatory
NAME	Optional name.	string		""
CAPACITY	This data defines the maximal instantaneous capacity of the resource.	integer		1

CALENDAR_ID	This data defines an optional calendar for the resource. The calendar on the resource is the "default" calendar used when no calendar is specified on the mode. Note that calendars can be set on modes of activities to do block planning.	id	NULL
START_MIN	This data sets the time before which no production activity or setup (except those known to have already started) can start on the resource. It is the common earliest start time shared by all activities requiring the resource.	integer	-INF
TIME_FENCE	This data sets the time fence to apply to the model start min at each database session to obtain this resource start min. The previous value of startMin of this resource will be conserved if it is greater than this computed value.	integer	-INF
END_MAX	This data sets the time after which no production activity or setup can end on the resource. It is the common latest end time shared by all activities requiring the resource.	integer	+INF
LINE_ID	This data sets the line identifier of the corresponding resource to the given value.	id	NULL
PLAN_CAPACITY_REDUCTION_FACTOR	A factor between 0.0 and 1.0 used to limit the bucket capacity for the planning engine. This factor is used to provide a realistic estimate during planning optimization of the availability of the resource with regard to real scheduling constraints.	double	1.0
VISIBLE	The visibility of this resource.	boolean	1
DISPLAY_RANK	The rank of the resource in the GUI. The smaller the value, the higher the resource displays in the **Gantt Diagram**. The value **-1** means no ranking.	integer	-1
CATEGORY	The category of the resource to be interpreted or the symbolic representation in a specialized editor of the GUI. The expected values are: `column`, `cooler`, `filter`, `homogenizer`, `operator`, `packer`, `pasteurizer`, `separator`, `sterilizer`, `tank` or `team`.	id	""
X	The X-coordinate of the resource in the plant layout.	double	+1.#INF
Y	The Y-coordinate of the resource in the plant layout.	double	+1.#INF

PPO_RESOURCE_CAPACITY_COST

Synopsis

```
PPO_RESOURCE_CAPACITY_COST,RESOURCE_ID,RESOURCE_CAPACITY_COST_FCT_ID,VALIDITY_START_TIME,
VALIDITY_END_TIME
```

Topic

Manufacturing Resources

Description

The PPO_RESOURCE_CAPACITY_COST table specifies how the cost of using the capacity of a resource over time is evaluated. For each resource, a piecewise or stepwise linear capacity cost function is defined. This function may vary over time.

Primary keys

RESOURCE_ID,RESOURCE_CAPACITY_COST_FCT_ID,VALIDITY_START_TIME,VALIDITY_END_TIME

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	The identifier of the resource.	id		mandatory
RESOURCE_CAPACITY_COST_FCT_ID	The identifier of the cost function.	id		mandatory
VALIDITY_START_TIME	The start time of the period over which the given function applies.	integer		-INF
VALIDITY_END_TIME	The end time of the period over which the given function applies.	integer		+INF

PPO_RESOURCE_CAPACITY_COST_FCT

Synopsis

```
PPO_RESOURCE_CAPACITY_COST_FCT,RESOURCE_CAPACITY_COST_FCT_ID,LEVEL_NUMBER,CAPACITY,
FIXED_COST,VARIABLE_COST,IDLE_VARIABLE_COST
```

Topic

Manufacturing Resources

Description

The PPO_RESOURCE_CAPACITY_COST_FCT table defines a piecewise or stepwise linear cost function used to evaluate the cost of using a resource over time. It also allows you to penalize the idle resources using a piecewise linear idle cost (only used by the planning engine).

Primary keys

RESOURCE_CAPACITY_COST_FCT_ID,LEVEL_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_CAPACITY_COST_FCT_ID	The identifier of the cost function.	id		mandatory
LEVEL_NUMBER	The index (zero-based) of a capacity level.	integer		0
CAPACITY	The capacity limit (included between 0 and capacity of the resource) up to which this level applies. Planning example: For a resource of capacity 2, if we want the average capacity on a time bucket to be penalized above 66% then we must create two levels, one with a CAPACITY of 1.32 and a VARIABLE_COST of zero, and one with a CAPACITY of 2.0 or more and a positive VARIABLE_COST.	double		+1.#INF
FIXED_COST	The fixed cost incurred to enter this level of capacity; that is when the CAPACITY of the previous level is exceeded.	double		0.
VARIABLE_COST	The slope of the piecewise linear cost function for this level of capacity; that is, for each unit of capacity included in this level (from the CAPACITY of the previous level to the CAPACITY of this level).	double		0.
IDLE_VARIABLE_COST	The slope of the piecewise linear cost function when not using the capacity included in this level (from the CAPACITY of the previous level to the CAPACITY of this level). The idle cost is only used by the planning engine.	double		0.

PPO_RESOURCE_CONNECTION

Synopsis

```
PPO_RESOURCE_CONNECTION,FIRST_RESOURCE_ID,SECOND_RESOURCE_ID
```

Topic

Manufacturing Resources

Description

The PPO_RESOURCE_CONNECTION table is used to define physical connections between resources so that one can enforce the usage of two connected resources as primary resources of two activities.

Primary keys

FIRST_RESOURCE_ID,SECOND_RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_RESOURCE_ID	The identifier of one of the connected resources.	id		mandatory
SECOND_RESOURCE_ID	The identifier of one of the connected resources.	id		mandatory

PPO_RESOURCE_FAMILY

Synopsis

```
PPO_RESOURCE_FAMILY,RESOURCE_FAMILY_ID,NAME,TYPE,COLOR,DISPLAY_RANK
```

Topic

Manufacturing Resources

Description

For a given resource family type, a resource may be a member of one family of this type.

Primary keys

RESOURCE_FAMILY_ID

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_FAMILY_ID	This data is the mandatory unique identifier of the family.	id		mandatory
NAME	This data is the name describing the family.	string		mandatory
TYPE	This data is the family type this family belongs to.	id		mandatory
COLOR	This data is the family color.	string		""
DISPLAY_RANK	The rank of the resource family in the GUI. The smaller the value, the higher the resource family displays in the **Gantt Diagram**.	integer		-1

PPO_RESOURCE_HIERARCHY

Synopsis

```
PPO_RESOURCE_HIERARCHY,SUB_RESOURCE_ID,SUPER_RESOURCE_ID
```

Topic

Manufacturing Resources

Description

The PPO_RESOURCE_HIERARCHY table is used to allow the planning module to work on an aggregated representation of the plant resources. This table is optional. When it is not used, the planning and the scheduling modules work with the same resources. When it is used, the planning module plans the super resources while the scheduling module schedules the subresources as specified in the modes of the activities.

Primary keys

SUB_RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
SUB_RESOURCE_ID	The identifier of a resource.	id		mandatory
SUPER_RESOURCE_ID	The identifier of the super resource to which the subresource belongs.	id		mandatory

PPO_RESOURCE_RESOURCE_FAMILY

Synopsis

```
PPO_RESOURCE_RESOURCE_FAMILY,RESOURCE_ID,RESOURCE_FAMILY_ID
```

Topic

Manufacturing Resources

Description

A resource may be a member of a several resource families.

Primary keys

RESOURCE_ID,RESOURCE_FAMILY_ID

Field descriptions

Field Name	Description	Type	Range	Default
RESOURCE_ID	This data is the mandatory unique identifier of the resource belonging to the family.	id		mandatory
RESOURCE_FAMILY_ID	This data is the mandatory unique identifier of the family.	id		mandatory

PPO_SCHED_CRITERION_WEIGHT

Synopsis

```
PPO_SCHED_CRITERION_WEIGHT,OPTIMIZATION_PROFILE_ID,CRITERION_ID,WEIGHT
```

Topic

General tables

Description

The PPO_SCHED_CRITERION_WEIGHT table defines the global weights of the directly optimizable Key Performance Indicators (KPIs) for the scheduling engine. These weights can be associated with different optimization profiles. Key Performance Indicators and optimization criteria are discussed in detail in the section *Key performance indicators*.

Primary keys

OPTIMIZATION_PROFILE_ID,CRITERION_ID

Field descriptions

Field Name	Description	Type	Range	Default
OPTIMIZATION_PROFILE_ID	The optimization profile under consideration.	id		"NoProfile"
CRITERION_ID	The criterion. The possible values are: TotalCleanupCost, TotalEarlinessCost, TotalInventoryCost (excess), TotalInventoryDeficitCost, TotalNonDeliveryCost, TotalRevenue, TotalProcessingCost (mode costs), TotalSetupCost, TotalSetupTime, TotalTardinessCost, TotalUnperformedCost. Other values are possible but their use with the scheduling engine is discouraged (TotalIdleCost, TotalResourceCost, Makespan, MaxEarlinessCost, MaxTardinessCost).	id		mandatory
WEIGHT	The weight assigned to the criterion.	double		0.

PPO_SCOPE

Synopsis

```
PPO_SCOPE,SCOPE_ID,NAME,POSITION_INDEX
```

Topic

General tables

Description

This table lets you define a scope.

Primary keys

SCOPE_ID

Field descriptions

Field Name	Description	Type	Range	Default
SCOPE_ID	The identifier of the scope.	id		mandatory
NAME	The name of the scope.	string		""
POSITION_INDEX	The display index of the scope.	integer		0

PPO_SECONDARY_RESOURCE

Synopsis

```
PPO_SECONDARY_RESOURCE,ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID,REQUIRED_CAPACITY
```

Topic

Production Orders

Description

The PPO_SECONDARY_RESOURCE table declares the secondary resources of each mode (see PPO_SECONDARY_RESOURCE_PROTO).

Primary keys

ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of an activity.	id		mandatory
MODE_NUMBER	The mode index in the activity.	integer		mandatory
RESOURCE_ID	The secondary resource identifier.	id		mandatory
REQUIRED_CAPACITY	Defines the amount of capacity of the secondary resource that is required to execute the given activity in the given mode.	integer		mandatory

PPO_SECONDARY_RESOURCE_PROTO

Synopsis

```
PPO_SECONDARY_RESOURCE_PROTO,ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID,REQUIRED_CAPACITY
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SECONDARY_RESOURCE_PROTO table declares the secondary resources of each prototype mode.

Primary keys

ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of an activity.	id		mandatory
MODE_NUMBER	The mode index in the activity.	integer		mandatory
RESOURCE_ID	The secondary resource identifier.	id		mandatory
REQUIRED_CAPACITY	Defines the amount of capacity of the secondary resource that is required to execute the given activity in the given mode.	integer		mandatory

PPO_SETTING

Synopsis

`PPO_SETTING,OPTIMIZATION_PROFILE_ID,PARAMETER,PARAMETER_VALUE`

Topic

General tables

Description

The PPO_SETTING table defines the parameters used by the optimization engines and the PPO application interface. If the OPTIMIZATION_PROFILE_ID is "NULL" then the defined value for the given PARAMETER is valid for all optimization profiles, except for those profiles which have another value specified for the same PARAMETER. Some available parameters are `slackOnPlannedStartTime`, `slackOnPlannedEndTime`, `modeOfPlanningOnly`, `bShortName`, `MapFile`, and `MapGraphicFactor`.

The parameter `slackOnPlannedStartTime` expects an integer value (default is zero if inventory corridors are defined; otherwise `IloMSIntPlusInfinity`). It defines the number of time units of anticipation allowed to the scheduling engine with respect to the start of the time bucket chosen by the planning engine. The parameter `slackOnPlannedEndTime` (default is `IloMSIntPlusInfinity`) defines the number of time units of delay allowed to the scheduling engine with respect to the end of the time bucket chosen by the planning engine.

The parameter `modeOfPlanningOnly` expects a boolean value (the default is false). If the value is true, that forces the scheduling engine to respect the decisions taken by the planning engine with respect to chosen resources. Note that if super resources have been defined, then the scheduling engine choices are restricted to the subresources of the super resource chosen by the planning engine. This is a way to limit a possible combinatorial explosion of modes in scheduling.

The parameter `bShortName` is used to control activity names as they appear in the GUI Gantt chart; see the PPO_ACTIVITY_PROTO table. If the value is false (default) then the activity labels are obtained by concatenating the production order name to the activity prototype name. If the value is true then the activity label are obtained by a copy of the activity prototype name and interpreting the placeholders starting with the caret **^** symbol.

The parameter `MapFile` allows displaying a background map to the **Distribution Planning** view for multilocation problems. By default, the directory data and images are scanned.

The parameter `MapGraphicFactor` is used to tune the representation of nodes and arcs in the **Distribution Planning** view.

Primary keys

OPTIMIZATION_PROFILE_ID,PARAMETER

Field descriptions

Field Name	Description	Type	Range	Default
OPTIMIZATION_PROFILE_ID	The optimization profile to which this setting belongs.	id		"NULL"
PARAMETER	The name of the parameter.	id		mandatory
PARAMETER_VALUE	The value of the parameter.	id		mandatory

PPO_SETUP_ACTIVITY

Synopsis

```
PPO_SETUP_ACTIVITY,SETUP_ACTIVITY_ID,NAME,PRODUCTION_ACTIVITY_ID
```

Topic

Production Orders

Description

The PPO_SETUP_ACTIVITY table contains setup activity instances related to production activity instances (PPO_ACTIVITY). Also see the PPO_SETUP_ACTIVITY_PROTO table.

Primary keys

SETUP_ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
SETUP_ACTIVITY_ID	This data is the mandatory unique identifier of the setup activity.	id		mandatory
NAME	Optional name.	string		""
PRODUCTION_ACTIVITY_ID	The identifier of the production activity to which this setup activity belongs.	id		mandatory

PPO_SETUP_ACTIVITY_PROTO

Synopsis

```
PPO_SETUP_ACTIVITY_PROTO,SETUP_ACTIVITY_ID,NAME,PRODUCTION_ACTIVITY_ID
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_ACTIVITY_PROTO table is used only if specific constraints (for example, the use of specific resources) apply to setups. It allows the explicit definition of a setup activity prototype for a given production activity prototype (PPO_ACTIVITY_PROTO). If the setup activity does not need a specific additional resource, then defining the setup state required by the production activity is sufficient.

Primary keys

SETUP_ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
SETUP_ACTIVITY_ID	This data is the mandatory unique identifier of the setup activity.	id		mandatory
NAME	Optional name.	string		""
PRODUCTION_ACTIVITY_ID	The identifier of the production activity prototype to which this setup activity belongs.	id		mandatory

PPO_SETUP_MODE

Synopsis

```
PPO_SETUP_MODE,SETUP_ACTIVITY_ID,SETUP_MODE_NUMBER,NAME,RESOURCE_ID,START_MIN,START_MAX,
END_MIN,END_MAX,LINE_ID,CALENDAR_ID,BREAK_DURATION_MAX,MAX_END_DURATION_IN_BREAK,
SHIFT_BREAKABLE
```

Topic

Production Orders

Description

The PPO_SETUP_MODE table is used to define modes for explicitly-defined setup activity instances.

Primary keys

SETUP_ACTIVITY_ID,SETUP_MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
SETUP_ACTIVITY_ID	The identifier of the setup activity.	id		mandatory
SETUP_MODE_NUMBER	The index (0-based) of the mode.	integer		mandatory
NAME	Optional name.	string		""
RESOURCE_ID	This data defines the primary resource required by the corresponding mode.	id		NULL
START_MIN	This data sets the earliest possible start time of the setup activity in this mode.	integer		-INF
START_MAX	This data sets the latest possible start time of the setup activity in this mode.	integer		+INF
END_MIN	This data sets the earliest possible end time of the setup activity in this mode.	integer		-INF
END_MAX	This data sets the latest possible end time of the setup activity in this mode.	integer		+INF
LINE_ID	This data sets the line identifier of the corresponding mode. If not specified, the mode has the line id of its primary resource.	id		NULL
CALENDAR_ID	The calendar to be used if the setup activity is in this mode. If not specified, the mode has the calendar of its production activity. Breaks and work periods are taken into account on calendars assigned to modes, but resource capacities are not. Calendars on resources do account for capacities.	id		NULL

BREAK_DURATION_MAX	The maximal break duration over which the setup activity in this mode cannot be interrupted by a break.	integer	+INF
MAX_END_DURATION_IN_BREAK	The maximal duration over which the setup activity in this mode can end in a break.	integer	0
SHIFT_BREAKABLE	This data sets the shift-breakable flag for the setup activity in this mode. An activity that is "shift-breakable" can overlap a shift change. An activity that is not shift-breakable must be completely executed within a shift.	boolean	1

PPO_SETUP_MODE_PROTO

Synopsis

```
PPO_SETUP_MODE_PROTO,SETUP_ACTIVITY_ID,SETUP_MODE_NUMBER,NAME,RESOURCE_ID,START_MIN,
START_MAX,END_MIN,END_MAX,LINE_ID,CALENDAR_ID,BREAK_DURATION_MAX,MAX_END_DURATION_IN_BREAK,
SHIFT_BREAKABLE
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_MODE_PROTO table is used to define modes for explicitly-defined setup activity prototypes. Note that for a production activity and its setup activity, Plant PowerOps will select modes sharing the same primary resource.

Primary keys

SETUP_ACTIVITY_ID,SETUP_MODE_NUMBER

Field descriptions

Field Name	Description	Type	Range	Default
SETUP_ACTIVITY_ID	The identifier of the setup activity.	id		mandatory
SETUP_MODE_NUMBER	The index (0-based) of the mode.	integer		mandatory
NAME	Optional name.	string		""
RESOURCE_ID	This data defines the primary resource required by the corresponding mode.	id		NULL
START_MIN	This data sets the earliest possible start time of the setup activity in this mode.	integer		-INF
START_MAX	This data sets the latest possible start time of the setup activity in this mode.	integer		+INF
END_MIN	This data sets the earliest possible end time of the setup activity in this mode.	integer		-INF
END_MAX	This data sets the latest possible end time of the setup activity in this mode.	integer		+INF
LINE_ID	This data sets the line identifier of the corresponding mode. If not specified, the mode has the line id of its primary resource.	id		NULL

CALENDAR_ID	The calendar to be used if the setup activity is in this mode. If not specified, the mode has the calendar of its production activity. Breaks and work periods are taken into account on calendars assigned to modes, but resource capacities are not. Calendars on resources do account for capacities.	id	NULL
BREAK_DURATION_MAX	The maximal break duration over which the setup activity in this mode cannot be interrupted by a break.	integer	+INF
MAX_END_DURATION_IN_BREAK	The maximal duration over which the setup activity in this mode can end in a break.	integer	0
SHIFT_BREAKABLE	This data sets the shift-breakable flag for the setup activity in this mode. An activity that is "shift-breakable" can overlap a shift change. An activity that is not shift-breakable must be completely executed within a shift.	boolean	1

PPO_SETUP_PROD_COMPAT

Synopsis

```
PPO_SETUP_PROD_COMPAT,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Production Orders

Description

The PPO_SETUP_PROD_COMPAT table is similar to the PPO_PROD_PROD_COMPAT_PROTO table and is used to impose constraints between the execution of setup and production activities. See PPO_PROD_PROD_COMPAT_PROTO for more information.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of a setup activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_SETUP_PROD_COMPAT_PROTO

Synopsis

```
PPO_SETUP_PROD_COMPAT_PROTO,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Recipes, Activities, and Modes

Description

The PPO_PROD_SETUP_COMPAT_PROTO table is similar to the PPO_PROD_PROD_COMPAT_PROTO table and is used to impose constraints between the execution of production and setup activities. See PPO_PROD_PROD_COMPAT_PROTO for more information.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of a production activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of a setup activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_SETUP_PROD_PRECED

Synopsis

```
PPO_SETUP_PROD_PRECED,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Production Orders

Description

The PPO_SETUP_PROD_PRECED table is similar to the PPO_PROD_PROD_PRECED table and is used to create precedence constraints between setup and production activity instances.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor production activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_SETUP_PROD_PRECED_PROTO

Synopsis

```
PPO_SETUP_PROD_PRECED_PROTO,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_PROD_PRECED_PROTO table is similar to the PPO_PROD_PROD_PRECED_PROTO table and is used to create precedence constraints between setup and production activity prototypes.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor production activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_SETUP_SECONDARY_RESOURCE

Synopsis

```
PPO_SETUP_SECONDARY_RESOURCE,ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID,REQUIRED_CAPACITY
```

Topic

Production Orders

Description

The PPO_SETUP_SECONDARY_RESOURCE table declares the secondary resources of each mode for setup activities.

Primary keys

ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of an activity.	id		mandatory
MODE_NUMBER	The mode index in the activity.	integer		mandatory
RESOURCE_ID	The secondary resource identifier.	id		mandatory
REQUIRED_CAPACITY	Defines the amount of capacity of the secondary resource that is required to execute the given activity in the given mode.	integer		mandatory

PPO_SETUP_SECONDARY_RES_PROTO

Synopsis

```
PPO_SETUP_SECONDARY_RES_PROTO,ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID,REQUIRED_CAPACITY
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_SECONDARY_RES_PROTO table declares the secondary resource prototypes of each mode for setup activities.

Primary keys

ACTIVITY_ID,MODE_NUMBER,RESOURCE_ID

Field descriptions

Field Name	Description	Type	Range	Default
ACTIVITY_ID	The identifier of an activity.	id		mandatory
MODE_NUMBER	The mode index in the activity.	integer		mandatory
RESOURCE_ID	The secondary resource identifier.	id		mandatory
REQUIRED_CAPACITY	Defines the amount of capacity of the secondary resource that is required to execute the given activity in the given mode.	integer		mandatory

PPO_SETUP_SETUP_COMPAT

Synopsis

```
PPO_SETUP_SETUP_COMPAT,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Production Orders

Description

The PPO_SETUP_SETUP_COMPAT table is similar to the PPO_PROD_PROD_COMPAT table and is used to impose constraints between the execution of different setup activities. See PPO_PROD_PROD_COMPAT documentation.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of an activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of another activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_SETUP_SETUP_COMPAT_PROTO

Synopsis

```
PPO_SETUP_SETUP_COMPAT_PROTO,FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_SETUP_COMPAT_PROTO table is similar to the PPO_PROD_PROD_COMPAT_PROTO table and is used to impose constraints between the execution of different setup activities. See PPO_PROD_PROD_COMPAT_PROTO for more information.

Primary keys

FIRST_ACTIVITY_ID,SECOND_ACTIVITY_ID,TYPE

Field descriptions

Field Name	Description	Type	Range	Default
FIRST_ACTIVITY_ID	The identifier of an activity.	id		mandatory
SECOND_ACTIVITY_ID	The identifier of another activity.	id		mandatory
TYPE	See PPO_PROD_PROD_COMPAT_PROTO documentation for the list of possible values.	id		mandatory

PPO_SETUP_SETUP_PRECED_PROTO

Synopsis

```
PPO_SETUP_SETUP_PRECED_PROTO,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SETUP_SETUP_PRECED_PROTO table is similar to the PPO_PROD_PROD_PRECED_PROTO table and is used to create precedence constraints between two setup activity prototypes.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor setup activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_SETUP_SETUP_PRECED

Synopsis

`PPO_SETUP_SETUP_PRECED,PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX`

Topic

Production Orders

Description

The PPO_SETUP_SETUP_PRECED table is similar to the PPO_PROD_PROD_PRECED table and is used to create precedence constraints between two setup activity instances.

Primary keys

PREDECESSOR_ID,SUCCESSOR_ID,TYPE,DELAY_MIN,DELAY_MAX

Field descriptions

Field Name	Description	Type	Range	Default
PREDECESSOR_ID	This data defines the predecessor setup activity of the precedence constraint.	id		mandatory
SUCCESSOR_ID	This data defines the successor setup activity of the precedence constraint.	id		mandatory
TYPE	See PPO_PROD_PROD_PRECED_PROTO for possible values.	id		"EndToStart"
DELAY_MIN	This data defines the minimal delay between the relevant time points of the two activities.	integer		0
DELAY_MAX	This data defines the maximal delay between the relevant time points of the two activities.	integer		+INF

PPO_SPANNING

Synopsis

```
PPO_SPANNING,SPANNING_ACTIVITY_ID,SPANNED_ACTIVITY_ID,ON_START,ON_END
```

Topic

Production Orders

Description

The PPO_SPANNING table is used to define spanning constraints between activity instances (see table PPO_SPANNING_PROTO).

Primary keys

SPANNING_ACTIVITY_ID,SPANNED_ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
SPANNING_ACTIVITY_ID	This data defines the spanning activity.	id		mandatory
SPANNED_ACTIVITY_ID	This data defines a spanned activity for the spanning constraint.	id		mandatory
ON_START	When true, the spanning constraint is enforced for the start of the spanning activity.	boolean		1
ON_END	When true, the spanning constraint is enforced for the end of the spanning activity.	boolean		1

PPO_SPANNING_PROTO

Synopsis

```
PPO_SPANNING_PROTO,SPANNING_ACTIVITY_ID,SPANNED_ACTIVITY_ID,ON_START,ON_END
```

Topic

Recipes, Activities, and Modes

Description

The PPO_SPANNING_PROTO table is used to define spanning constraints between activities. It allows you to state that an activity `s` spans over a set of activities `{a}`. The activity `s` is called the spanning activity and its time bounds are computed as a propagation of the time bounds of its spanned activities. More precisely, it means that the start time of the spanning activity is equal to the earliest of the start times of the spanned activities, and that the end time of the spanning activity is equal to the latest of the end times of the spanned activities. The main difference between precedence constraints and spanning constraints is the following: With precedence constraints one can specify that an activity covers a set of activities but if there is no known order among them, one cannot constrain the covering activity so that its start time is equal to the start of the earliest covered activity (ON_START column), or that its end time is equal to the end time of the latest covered activity (ON_END column).

Primary keys

SPANNING_ACTIVITY_ID,SPANNED_ACTIVITY_ID

Field descriptions

Field Name	Description	Type	Range	Default
SPANNING_ACTIVITY_ID	This data defines the spanning activity.	id		mandatory
SPANNED_ACTIVITY_ID	This data defines a spanned activity for the spanning constraint.	id		mandatory
ON_START	When true, the spanning constraint is enforced for the start of the spanning activity.	boolean		1
ON_END	When true, the spanning constraint is enforced for the end of the spanning activity.	boolean		1

PPO_STORAGE_UNIT

Synopsis

`PPO_STORAGE_UNIT,STORAGE_UNIT_ID,NAME,QUANTITY_MAX,RESOURCE_ID,VISIBLE,LATITUDE,LONGITUDE,CATEGORY`

Topic

Materials and Storage Units

Description

The PPO_STORAGE_UNIT table is used to represent a physical storage facility for materials or a location. Storage units can store one or several materials. Activities may produce to or consume from a storage unit. Particular storage units can be specified for material procurement or storage. Note that if a storage unit is specified for a material, then any demand, procurement or material production in the recipes referring to that material must also specify the storage unit. You must fill in the table PPO_STORAGE_UNIT_MATERIAL so that the storage unit can store material. Storage units are useful to model multiple locations in multisite planning. A longitude, latitude and a category (warehouse, factory) can be specified for representation in the GUI. If not attached to a unary resource, a storage unit is simply a location (such as a warehouse) with a possible maximum storage capacity. A storage unit attached to a unary resource can model a storage tank. The unary resource ensures the sequencing of batches in the tank.

A group of tanks (or *super storage unit*) can be created by defining a storage unit **S** and attaching it to a super resource **R**. Then each storage unit associated to a subresource of the super resource **R** is a substorage unit of **S**. Such grouping simplifies the planning problem by considering globally the super resource capacities and the super storage unit aggregated volume. It is also useful for specifying the production or consumption of a material in a recipe. Note that the storage unit destination of a material must be consistent with the primary resource required by the activity producing it (if the storage unit is attached to a resource)

When using super storage units, be aware that any material exit (demand, activity consumption) must never refer to a subunit. That is, consumption must remain "fuzzy". Note, however, that initial stock or procurements must refer to precise subunits. When a material production on a mode of an activity prototype refers to a super unit at the recipe level, the mode generation on corresponding production orders will generate material productions on precise subunits. Thus any production in the transactional data is precise, and any consumption is fuzzy.

Primary keys

STORAGE_UNIT_ID

Field descriptions

Field Name	Description	Type	Range	Default

STORAGE_UNIT_ID	This data is the mandatory unique identifier of the instance.	id	mandatory
NAME	Optional name.	string	""
QUANTITY_MAX	The maximal quantity of any material (or sum of materials) that can be stored at the same time in the storage unit (capacity max of the storage unit).	double	+1.#INF
RESOURCE_ID	Optional column. This is the identifier of the associated unary resource if the storage unit is an exclusive multipurpose tank.	id	NULL
VISIBLE	The visibility of this storage unit in the level charts.	boolean	1
LATITUDE	This field is used together with longitude to locate the storage unit on a map for multilocation problems. Latitude gives the location of a place on Earth north or south of the equator. Technically, latitude is an angular measurement in degrees (marked with degree) ranging from 0 degrees at the Equator to +90 degrees at the North Pole and -90 degrees at the South Pole. Also see documentation about the PPO_SETTING table for other map considerations.	double	0.0
LONGITUDE	This field is used together with latitude to locate the storage unit on a map for multi-location problems. Longitude is given as an angular measurement ranging from 0 degrees at the prime meridian (Greenwich meridian) to +180 degrees eastward and -180 degrees westward. Also see documentation about the PPO_SETTING table for other map considerations.	double	0.0
CATEGORY	Category of the storage unit. Interpreted by the GUI for choosing a symbol on a map. Expected values are `warehouse` or `factory`.	id	NULL

PPO_STORAGE_UNIT_MATERIAL

Synopsis

```
PPO_STORAGE_UNIT_MATERIAL,STORAGE_UNIT_ID,MATERIAL_ID
```

Topic

Materials and Storage Units

Description

The PPO_STORAGE_UNIT_MATERIAL table specifies the materials that can be stored in the given storage unit as well as their initially available quantities. The initially available quantity is the quantity that is stored at time START_MIN of the model. Note that the quantities of the materials that share the same storage unit must be expressed in the same measurement unit.

Primary keys

STORAGE_UNIT_ID,MATERIAL_ID

Field descriptions

Field Name	Description	Type	Range	Default
STORAGE_UNIT_ID	This data specifies the storage unit.	id		mandatory
MATERIAL_ID	This data specifies the material.	id		mandatory

PPO_UNIT

Synopsis

`PPO_UNIT,UNIT_ID,NAME,DIMENSION,STANDARD_CONVERSION,CHECKING_TOLERANCE`

Topic

Materials and Storage Units

Description

The PPO_UNIT table is used to represent units of measure for quantities of material consumed or produced.

This table is used to declare units of measure of the model and conversion from standard units of the metric system. Some units of measure do not have a dimension or a standard conversion factor. For example, the pallet, the cup, and the box may have different conversion factors, depending on the contained material. Other units of measure do have a dimension, and thus a standard conversion to the metric system, such as length, surface area, volume, and mass. For example, the metric ton is a "mass" dimension, and its conversion factor from the standard unit of the dimension in the metric system (kilogram) is always 1/1000; it does not depend on the material being measured. The standard units are kilogram for mass, meter for length, square meter for surface area, cubic meter for volume.

All computations are performed in the primary unit of the material. However, you can define secondary units for a material that are used for display purposes in the GUI. Identify the primary and secondary units of a material in the PPO_MATERIAL table. Define the conversion factor between the primary and secondary units of a material in the PPO_MATERIAL_SECONDARY_UNIT table.

Note that several materials may share the same unit with different conversions.

Primary keys

UNIT_ID

Field descriptions

Field Name	Description	Type	Range	Default
UNIT_ID	This data is the mandatory unique identifier of the unit.	id		mandatory
NAME	This data is the name describing the unit.	string		mandatory

DIMENSION	This data contains the kind of dimension the unit belongs to. Possible values are `Length`, `Surface`, `Volume`, `Mass`, `Time` and `NoDimension`. One can define "ton" with DIMENSION `Mass` and "box" with `NoDimension` because a box has different conversions for different materials.	id	"NoDimension"
STANDARD_CONVERSION	This data defines the standard conversion from the standard unit of the corresponding dimension to this unit. The standard dimensions are meter for Length, square meter for Surface, cubic meter for Volume, kilogram for Mass, second for Time. When the dimension is NoDimension, the STANDARD_CONVERSION shall be set to 0 (zero). To express the fact that a ton is 1000 kilograms, you should put `0.001` in This data and specify that a ton has `Mass` as DIMENSION.	double	mandatory
CHECKING_TOLERANCE	Use This data to define the tolerance of the checkers when this unit is the display unit.	double	0.01