Optim™

Optim EDM

Technical Overview

Agenda

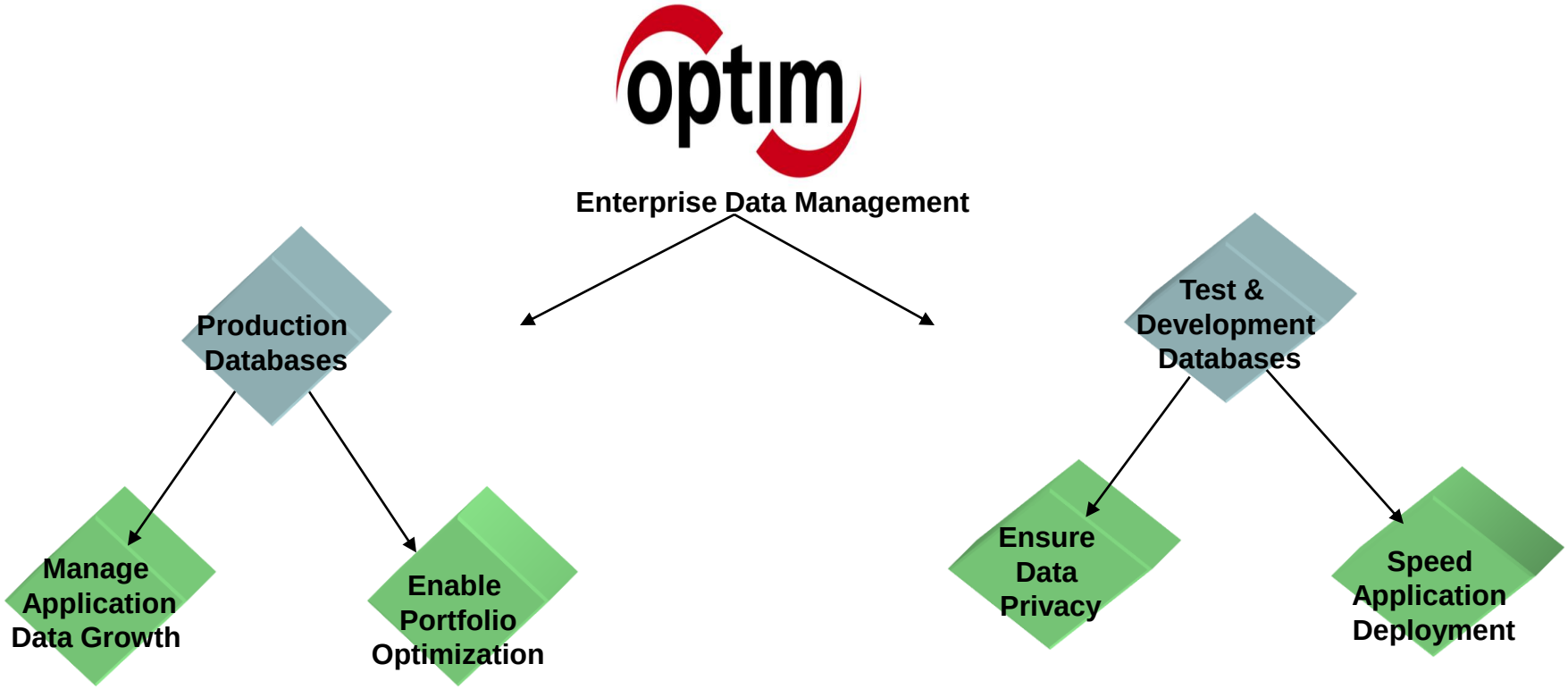
OPTIM Architecture

- OPTIM TDM (Test Data Management)
- OPTIM Data Privacy
- OPTIM Archive
- How do we compete?
- Q & A

The Princeton Softech Vision

Helping clients worldwide find better ways to manage their data and applications for greater efficiency and performance

Value Proposition



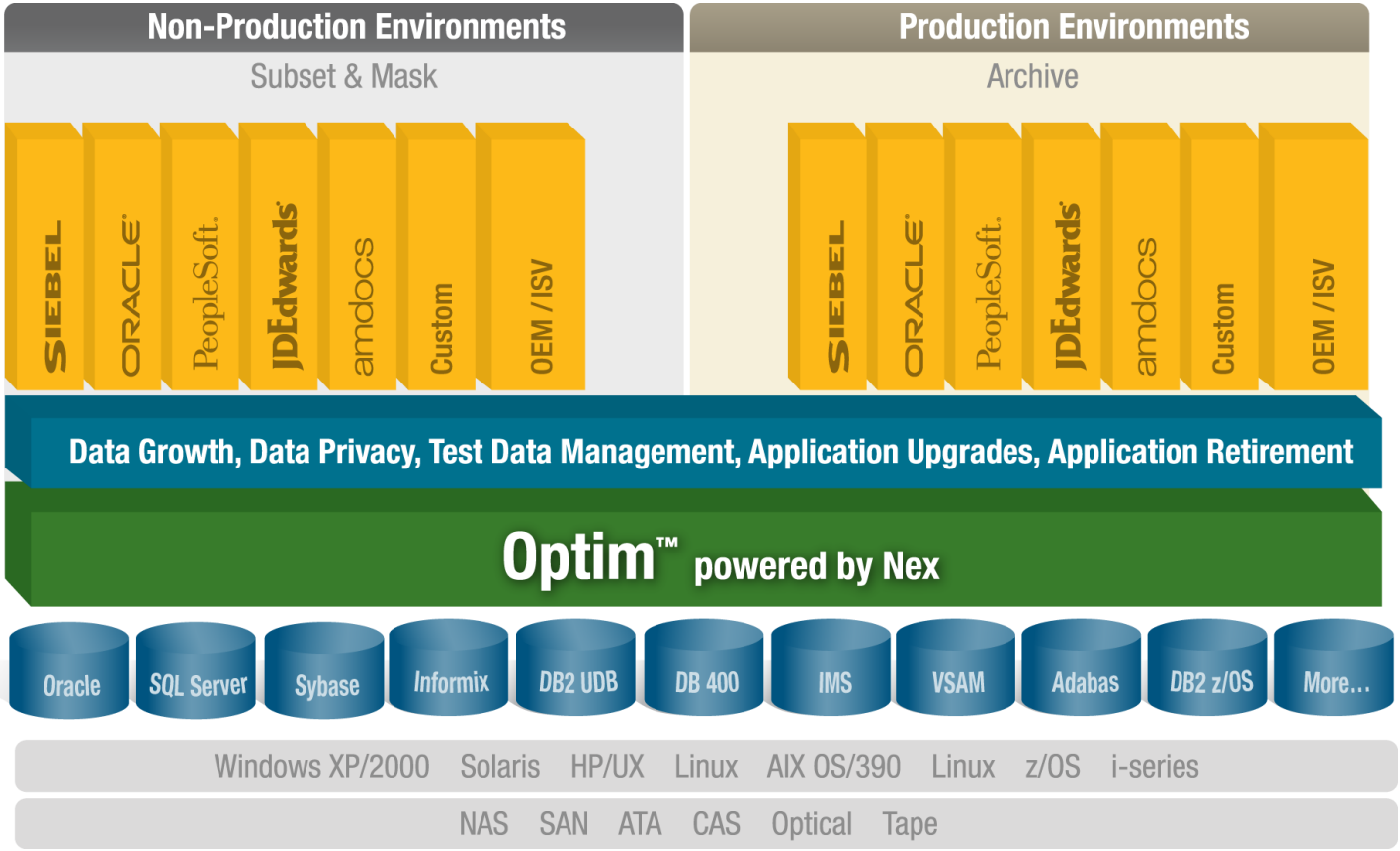
- Segregate Data & Move to Archive
- Deploy Tiered Storage Strategies
- Retain Data According to Value
- Simplify Infrastructure

- Decommission Redundant or Obsolete Apps
- Gain Control of Application Portfolio
- Retain Access to Legacy Data
- Retire Apps and Repurpose IT Assets
- Migrate Apps from High to Low Cost Platforms
- Preserve Historical Data

- Protect PII Data
- Apply Single Data Masking Solution
- Use Range of Masking Techniques
- Maintain Referential Integrity
- Maintain Contextual Look and Feel

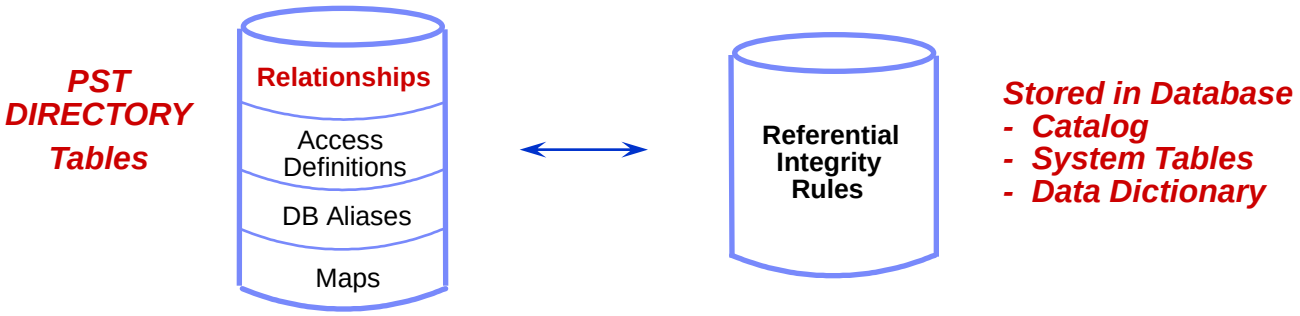
- Rightsize Test Apps
- Repeatable Process
- Quickly Deploy New Apps
- Futureproof Apps

Enterprise Architecture



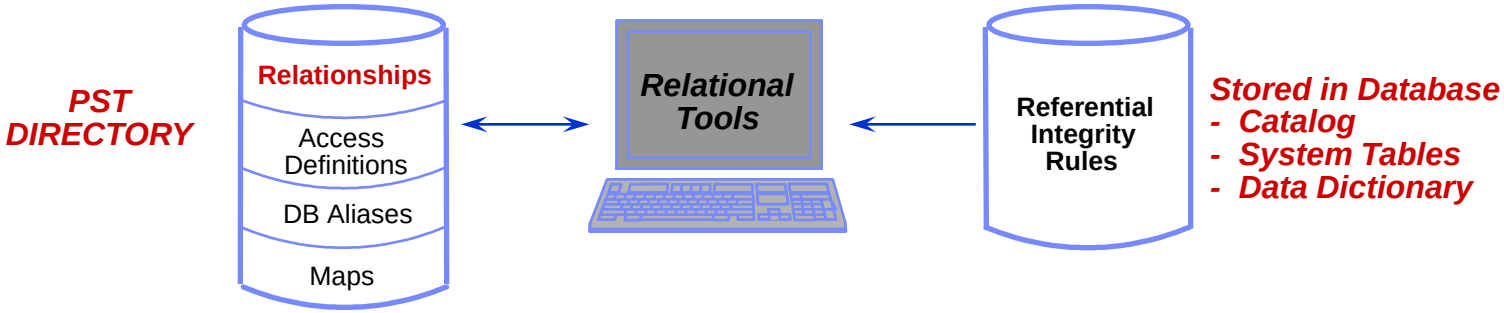
- **Single, scalable, interoperable EDM solution provides a central point to deploy policies to extract, store, port, and protect application data records from creation to deletion**

The Princeton Softech Directory



- **Supplements information stored in the database**
- **Maintains product definitions and tracks processing**
- **Stores database connection information (DB Aliases)**
- **Stores user-defined relationships**

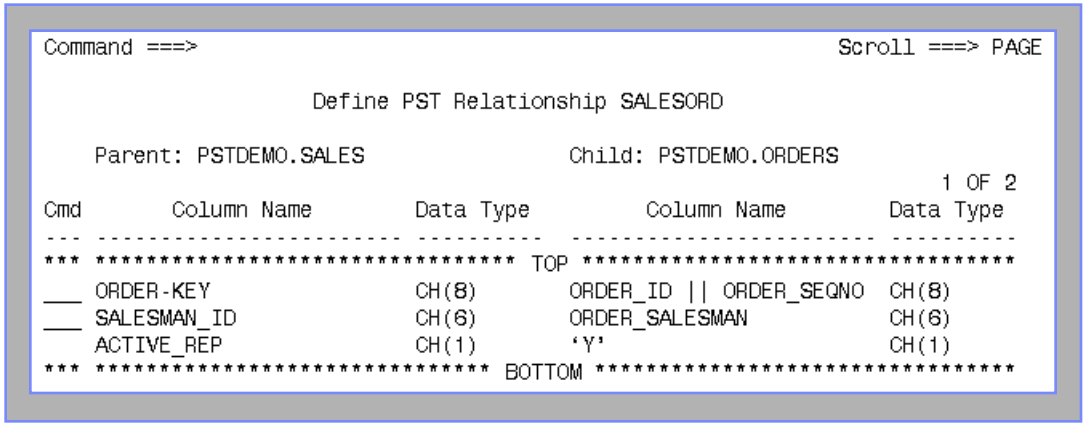
A Word About Relationships...



- **Automatically derived from database RI rules**
- **Can be defined to OPTIM or imported**
  - ➲ Shared by all Relational Tools components

OPTIM Extended Relationships

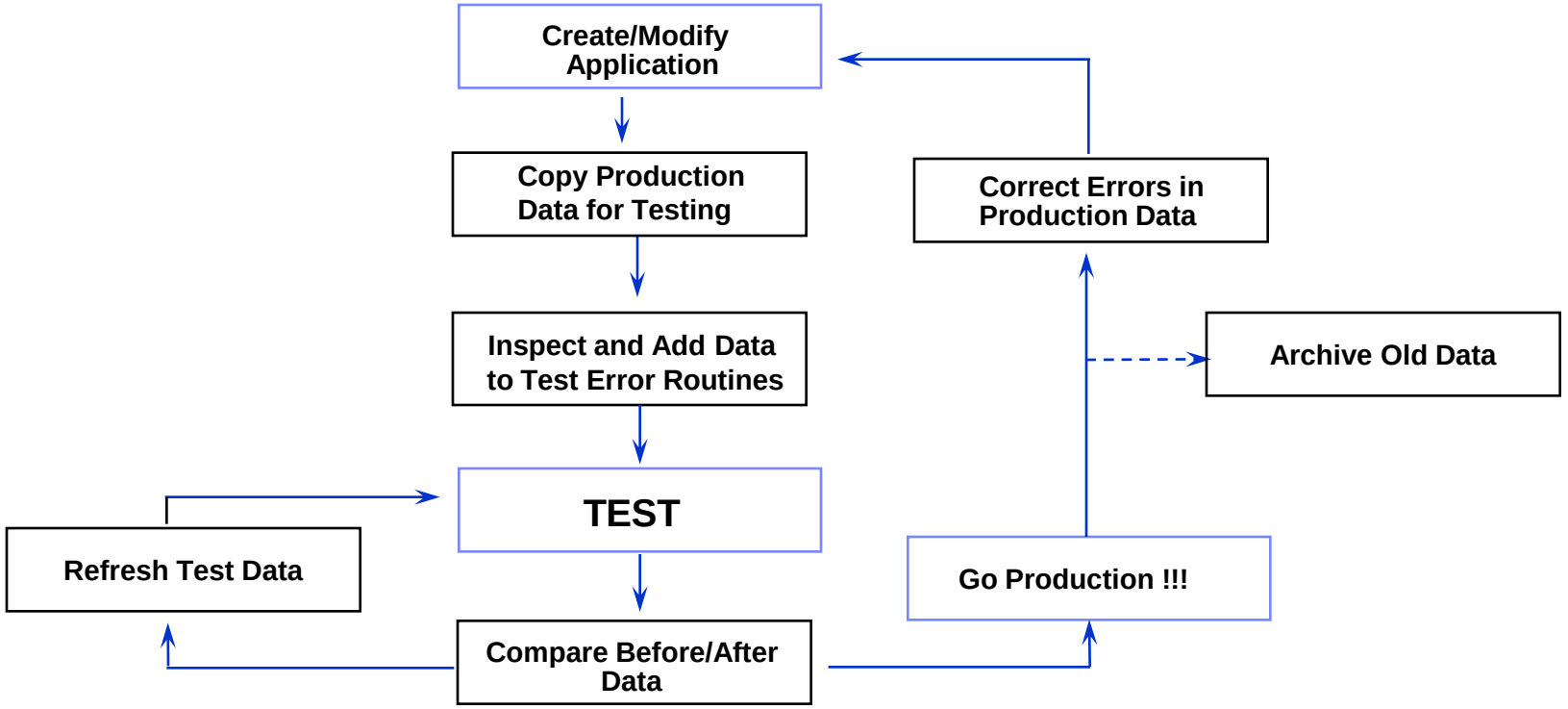
- **No need for primary key**
- **Relate column lists**
  - Single column related to multiple columns
  - Partial column related to single column
- **Flexible column attributes**
- **‘Data-driven’ relationships**



```
Command ===>                                                Scroll ===> PAGE

                    Define PST Relationship SALESORD

    Parent: PSTDEMO.SALES                Child: PSTDEMO.ORDERS
                                                                    1 OF 2
Cmd       Column Name          Data Type         Column Name         Data Type
--- ------------------------ ----------  ------------------------ ----------
*** ********************************* TOP ************************************
___ ORDER-KEY                  CH(8)       ORDER_ID || ORDER_SEQNO  CH(8)
___ SALESMAN_ID                CH(6)       ORDER_SALESMAN           CH(6)
    ACTIVE_REP                 CH(1)       'Y'                      CH(1)
*** ******************************** BOTTOM ***********************************
```

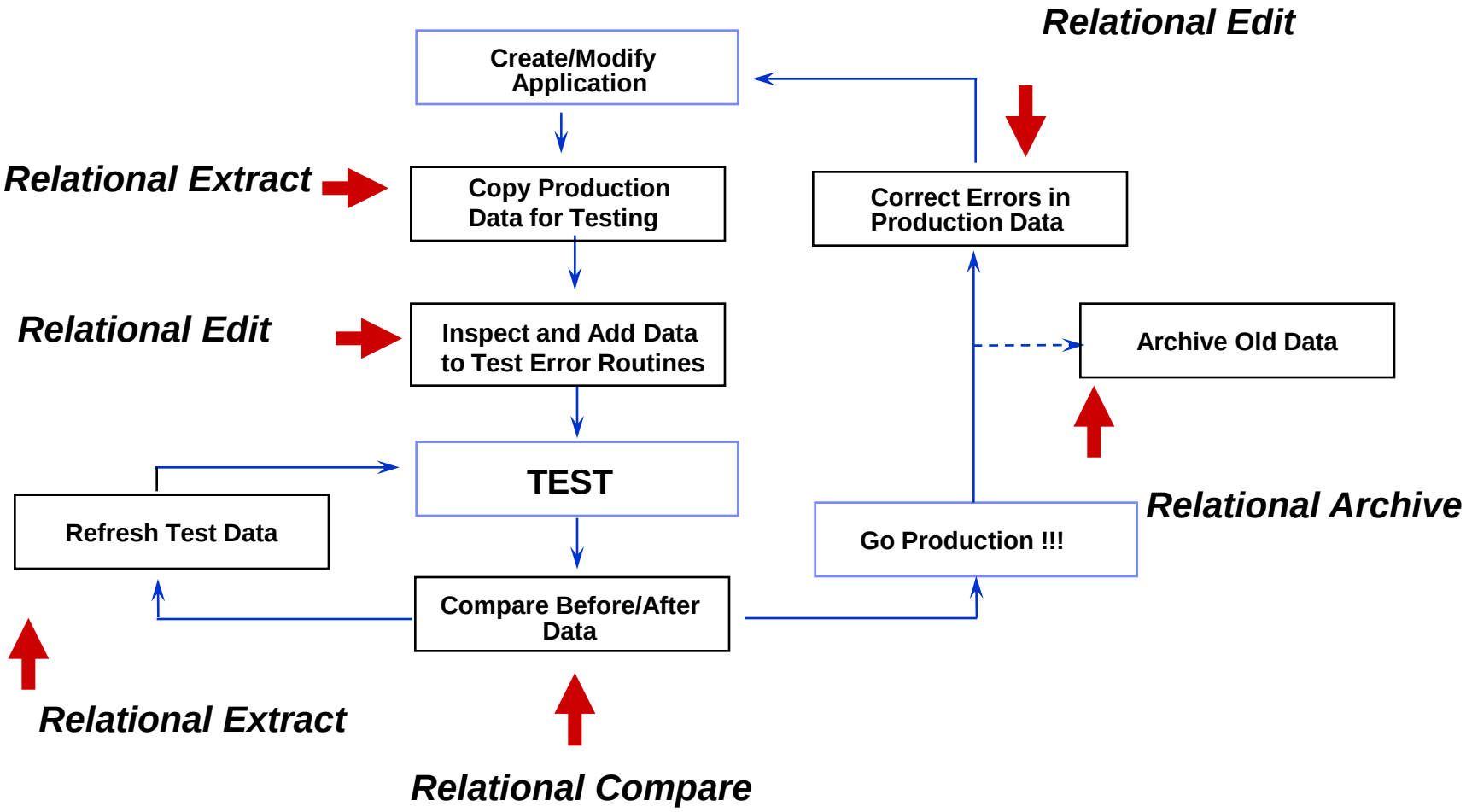
Managing Relational Data

Typical Development Activities



Princeton Softech's OPTIM Solution

The Solution for Managing Relational Sets of Data

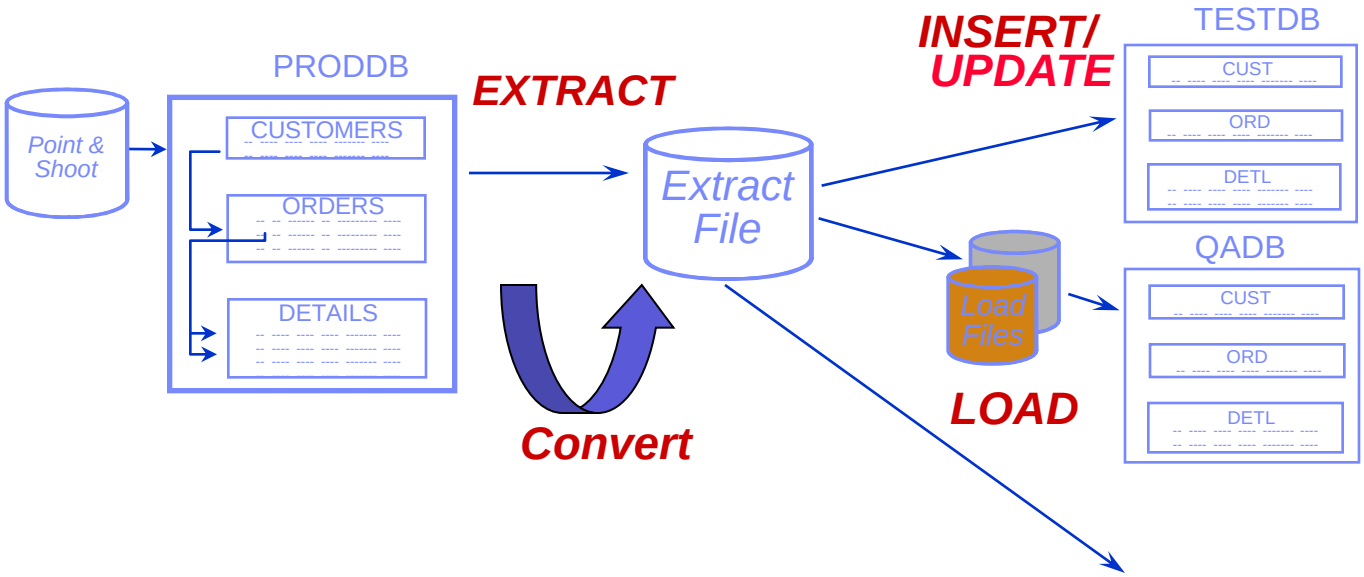


Optim™

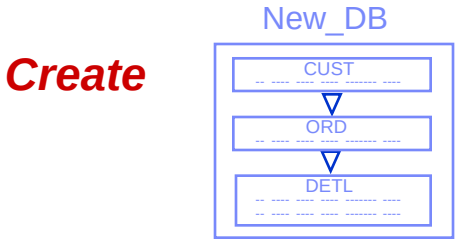
princeton softech
an IBM® Company

The Relational Extract Facility

The Relational Extract Facility

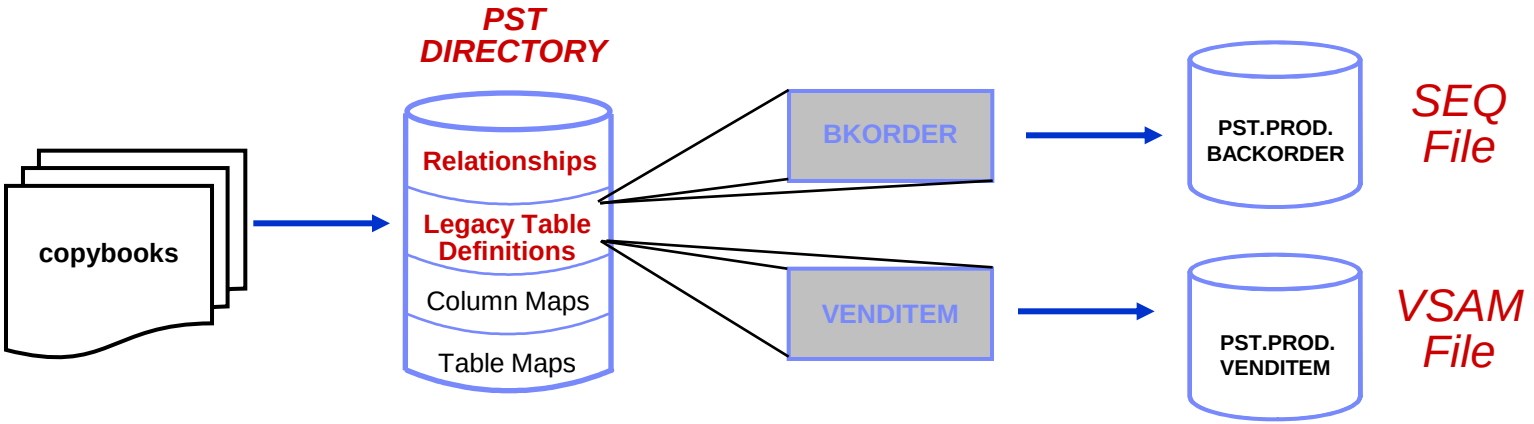


- **Creating and maintaining test data bases**
- **Migrating data**
- **Masking sensitive data**



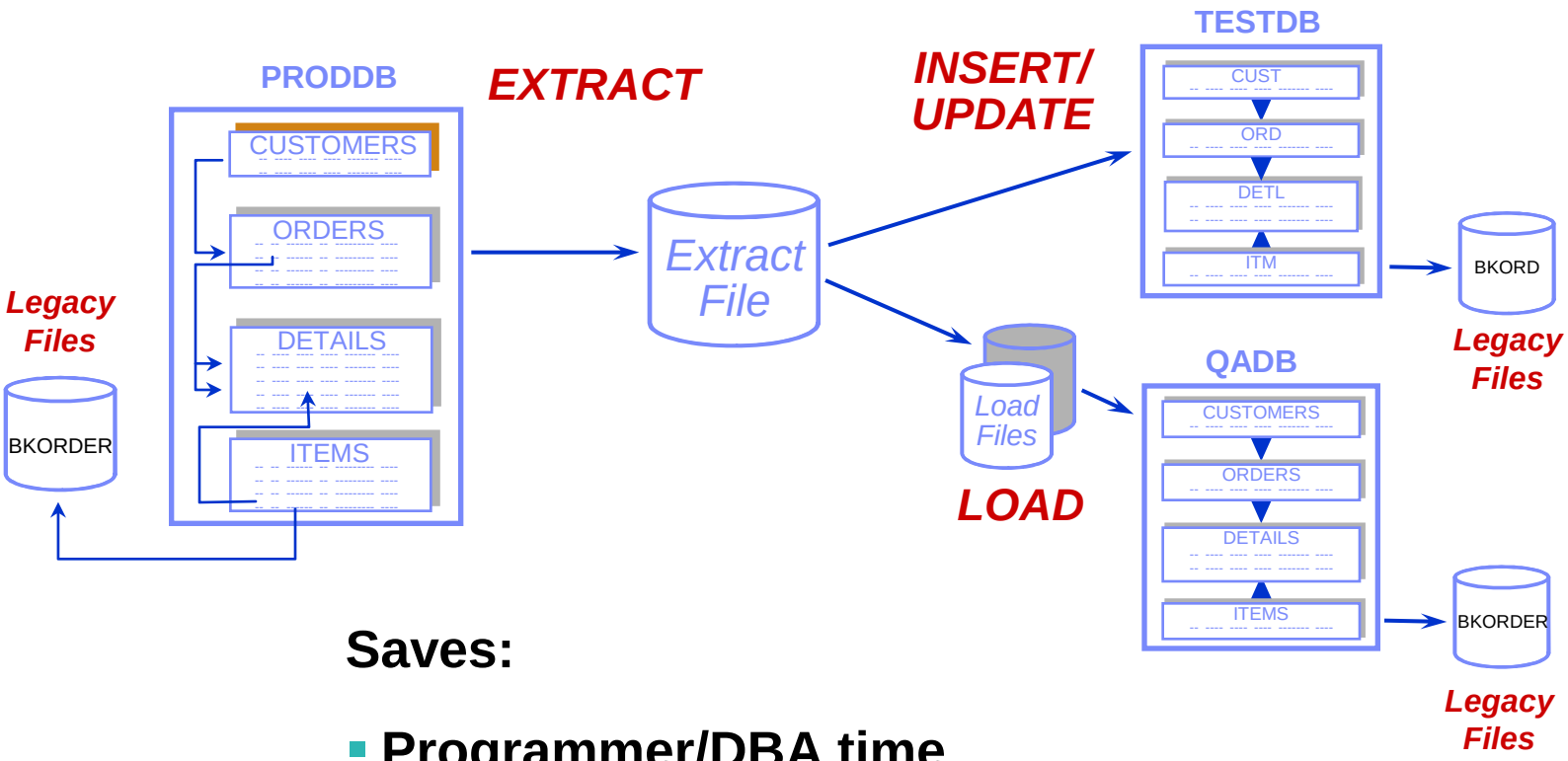
Legacy Data Files

- **Create Legacy Table definition from copybook**
- **Associate Table with Sequential or VSAM dataset**
- **Relate to other tables via PST Relationship**



Extracting Relational Sets of Data

Overview



Saves:

- **Programmer/DBA time**
- **Disk space utilization**
- **Testing interference**

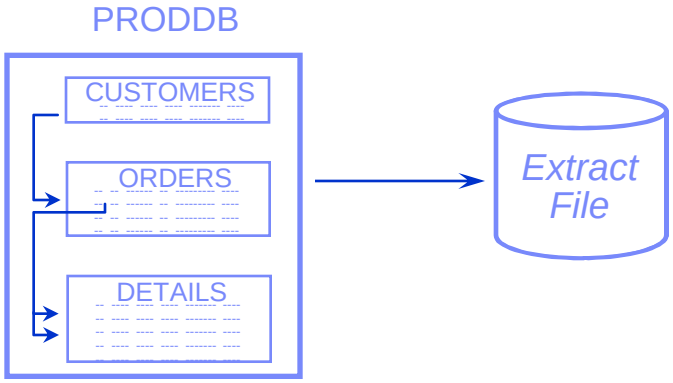
Defining the Extract…..

Tables

Views

Synonyms

Aliases



Required:

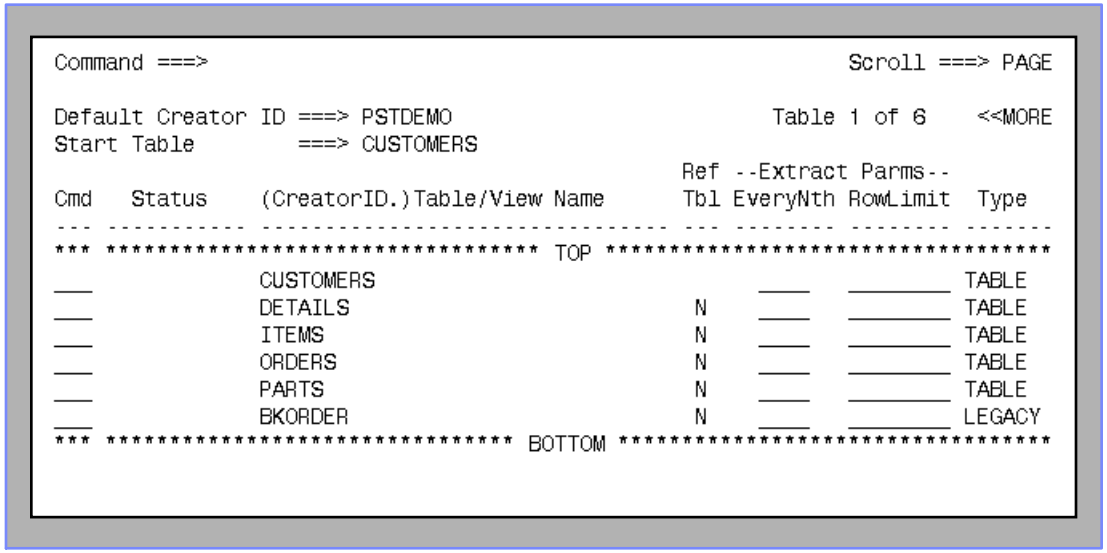
- **Start Table**
- **Set of Tables**

Optional:

- Selection Criteria
- Data Sampling
- Data Partitioning
- Point and Shoot
- Relationship Usage

Extract Process

The Table List

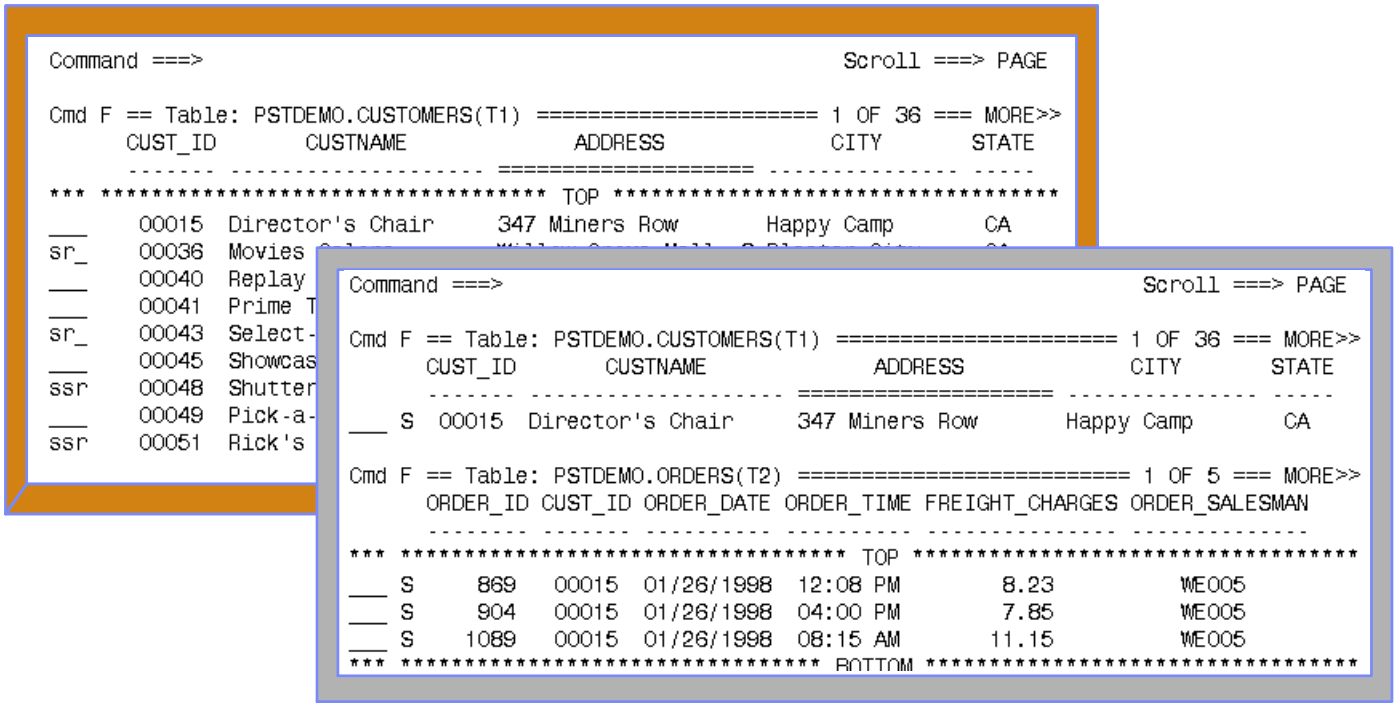


```
Command ===>                                                Scroll ===> PAGE

Default Creator ID ===> PSTDEMO                          Table 1 of 6    <<MORE
Start Table        ===> CUSTOMERS
                                                  Ref --Extract Parms--
Cmd   Status    (CreatorID.)Table/View Name       Tbl EveryNth RowLimit  Type
--- ----------- -------------------------------- --- -------- -------- -------
*** ********************************* TOP ***********************************
___             CUSTOMERS                                ____ ________ TABLE
___             DETAILS                           N      ____ ________ TABLE
___             ITEMS                             N      ____ ________ TABLE
___             ORDERS                            N      ____ ________ TABLE
___             PARTS                             N      ____ ________ TABLE
___             BKORDER                           N      ____ ________ LEGACY
*** ******************************** BOTTOM ***********************************
```

- **Identify the Start Table**
- **Use the RELATED functions to populate list**
- **Include random selection factor, extract limits and selection criteria**

Extract Process

Point-and-Shoot



- **Select individual rows from Start Table**
- **JOIN to view related rows**

Extract Process

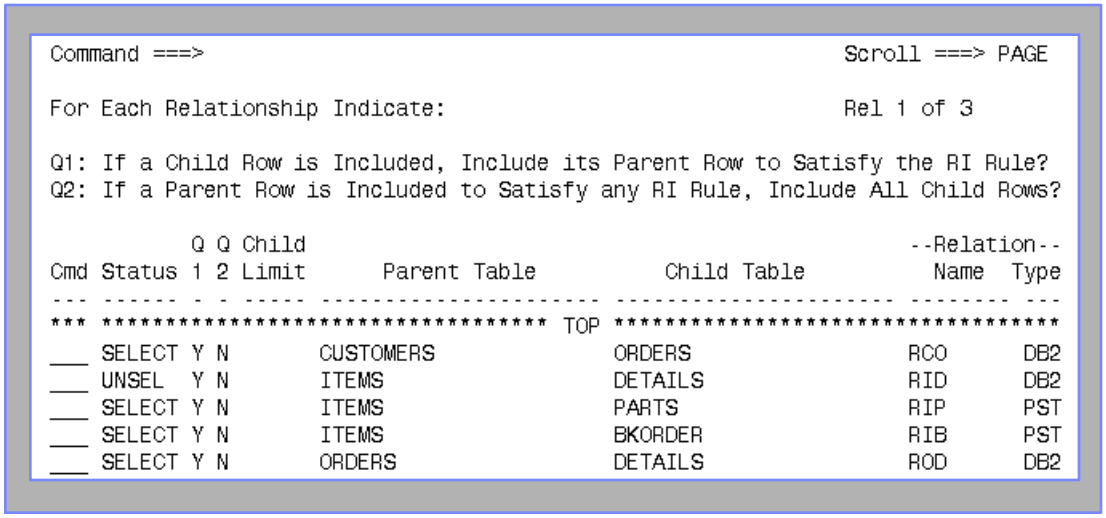
Show the Extract Steps

```
Command ===>                                                Scroll ===> PAGE

Step  1: Extract Rows from Start Table PSTDEMO.CUSTOMERS.  Row List is used
         and Determines the Rows Selected.

Step  2: Extract Rows from PSTDEMO.ORDERS which are Children of Rows
         Previously Extracted from PSTDEMO.CUSTOMERS in Step 1 using
         Relationship RCO.

Step  3: Extract Rows from PSTDEMO.DETAILS which are Children of Rows
         Previously Extracted from PSTDEMO.ORDERS in Step 2 using
         Relationship ROD.

Untraversed Table(s):        PSTDEMO.ITEMS
                             PSTDEMO.PARTS
                             PSTDEMO.BKORDER
```

- **Steps required to perform extract**
- **Cycles processed**
- **Untraversed tables**

Extract Process

Relationship Usage

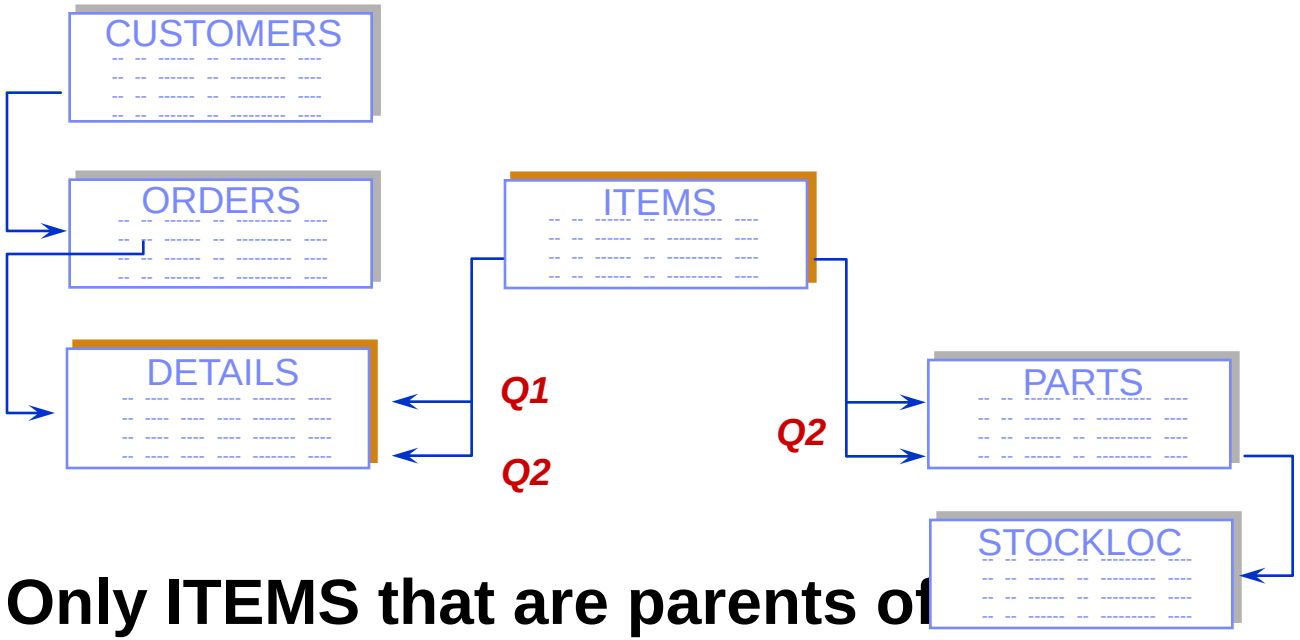


```
Command ===>                                                  Scroll ===> PAGE

For Each Relationship Indicate:                               Rel 1 of 3

Q1: If a Child Row is Included, Include its Parent Row to Satisfy the RI Rule?
Q2: If a Parent Row is Included to Satisfy any RI Rule, Include All Child Rows?

           Q Q Child                                                --Relation--
Cmd Status 1 2 Limit      Parent Table          Child Table           Name  Type
--- ------ - - ----- ---------------------- ---------------------- -------- ---
*** ************************************ TOP ************************************
___ SELECT Y N       CUSTOMERS             ORDERS                 RCO      DB2
___ UNSEL  Y N       ITEMS                 DETAILS                RID      DB2
___ SELECT Y N       ITEMS                 PARTS                  RIP      PST
___ SELECT Y N       ITEMS                 BKORDER                RIB      PST
___ SELECT Y N       ORDERS                DETAILS                ROD      DB2
```

- **Select relationship paths**
  - Defined to DB2 catalog or PST Directory
- **Designate relationship traversal**
- **Limit number of child rows extracted**

Extract Process

Relationship Traversal

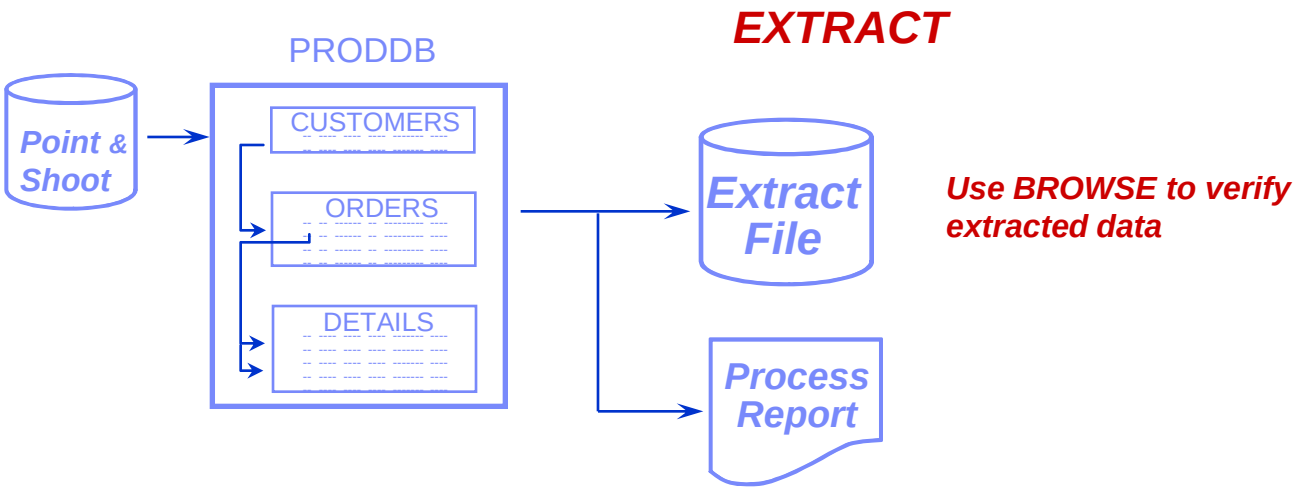


Q1 **Only ITEMS that are parents of**

Q2 **All other DETAILS for those ITEMS ...**
Each of the PARTS for those ITEMS

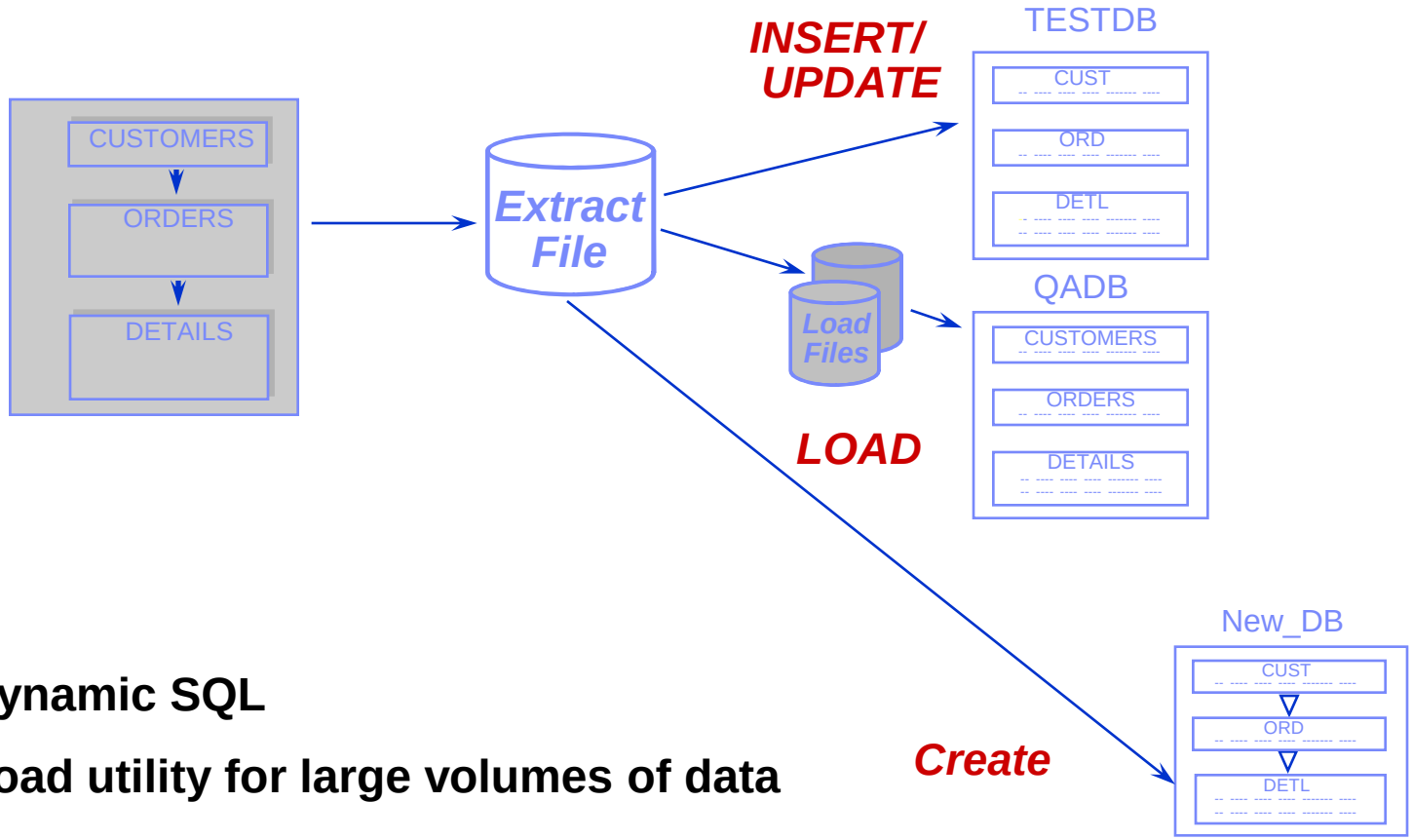
Extract Process

Extract Parameters



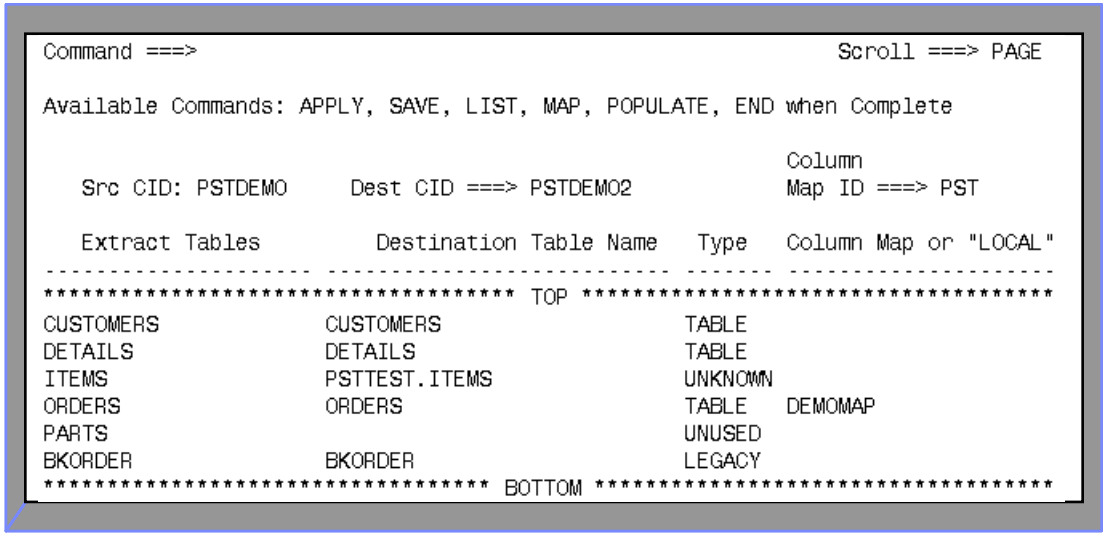
- **Extract from source tables using dynamic SQL**
- **Extract data and/or object definitions**

Populate Destination Tables



- **Dynamic SQL**
- **Load utility for large volumes of data**
- **Create a new set of tables**

Populate Destination Tables
Table Map



```
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY, SAVE, LIST, MAP, POPULATE, END when Complete

                                                              Column
   Src CID: PSTDEMO     Dest CID ===> PSTDEMO2                Map ID ===> PST

   Extract Tables         Destination Table Name    Type    Column Map or "LOCAL"
-------------------- ------------------------- ------- ---------------------
************************************ TOP ***************************************
CUSTOMERS            CUSTOMERS                 TABLE
DETAILS              DETAILS                   TABLE
ITEMS                PSTTEST.ITEMS             UNKNOWN
ORDERS               ORDERS                    TABLE   DEMOMAP
PARTS                                          UNUSED
BKORDER              BKORDER                   LEGACY
********************************** BOTTOM **************************************
```

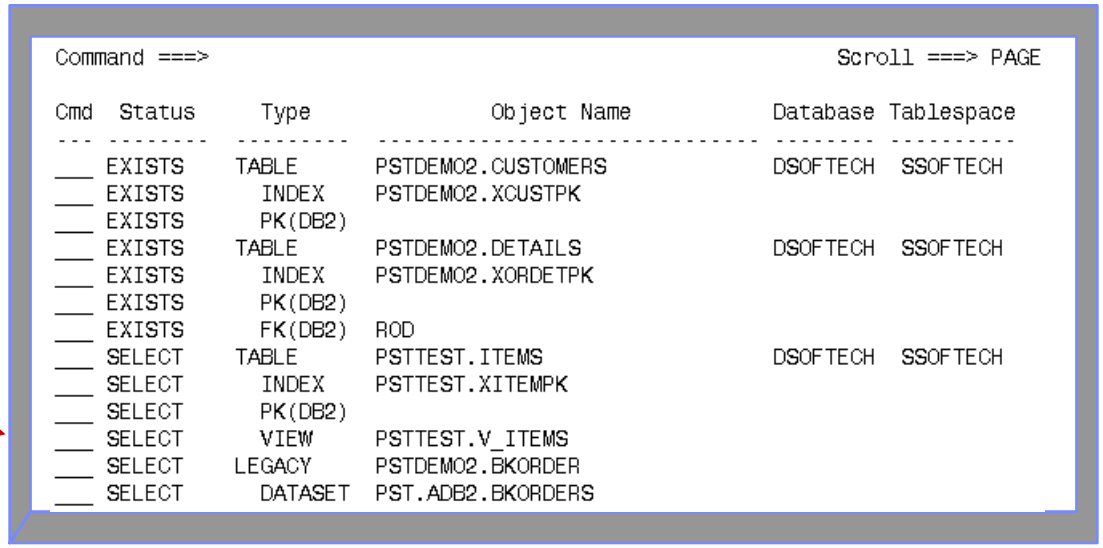
- **Table names need not match**
- **Change qualifier and/or table name**
- **Can be saved in PST Directory**

princeton softech an IBM® Company

Populate Destination Tables

Creating New Tables

Missing destination object(s)

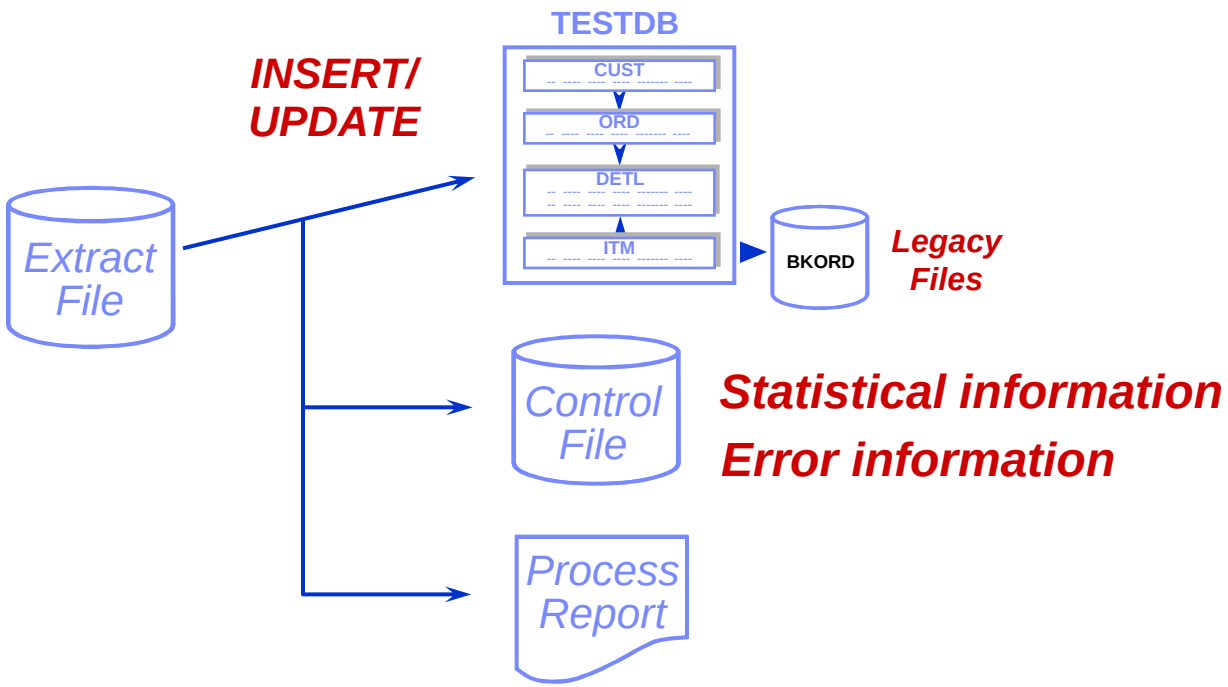


```
Command ===>                                                Scroll ===> PAGE

Cmd  Status     Type          Object Name                   Database Tablespace
--- --------  ---------  ------------------------------ -------- ----------
___ EXISTS    TABLE      PSTDEMO2.CUSTOMERS             DSOFTECH  SSOFTECH
___ EXISTS      INDEX    PSTDEMO2.XCUSTPK
___ EXISTS      PK(DB2)
___ EXISTS    TABLE      PSTDEMO2.DETAILS               DSOFTECH  SSOFTECH
___ EXISTS      INDEX    PSTDEMO2.XORDETPK
___ EXISTS      PK(DB2)
___ EXISTS      FK(DB2)  ROD
___ SELECT    TABLE      PSTTEST.ITEMS                  DSOFTECH  SSOFTECH
___ SELECT      INDEX    PSTTEST.XITEMPK
___ SELECT      PK(DB2)
___ SELECT      VIEW     PSTTEST.V_ITEMS
___ SELECT    LEGACY     PSTDEMO2.BKORDER
___ SELECT      DATASET  PST.ADB2.BKORDERS
```

- **Select destination object(s) to be created from source table definitions**
- **Functions include DROP, key conversion, and display of SQL**

Populate Destination Tables

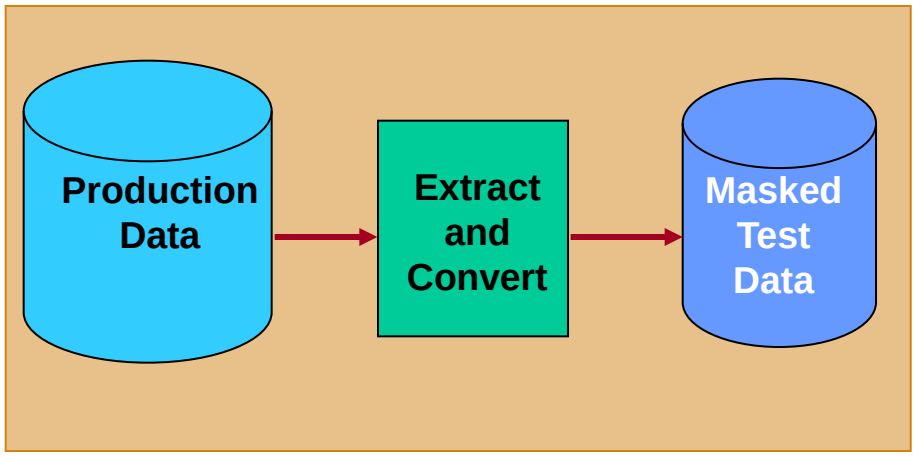
Control File



If INSERT/UPDATE errors occur:

1. BROWSE the control file for error information
2. RETRY/RESTART the INSERT/UPDATE

De-Identify test data



During Extract Process

Or

Standalone Convert Process

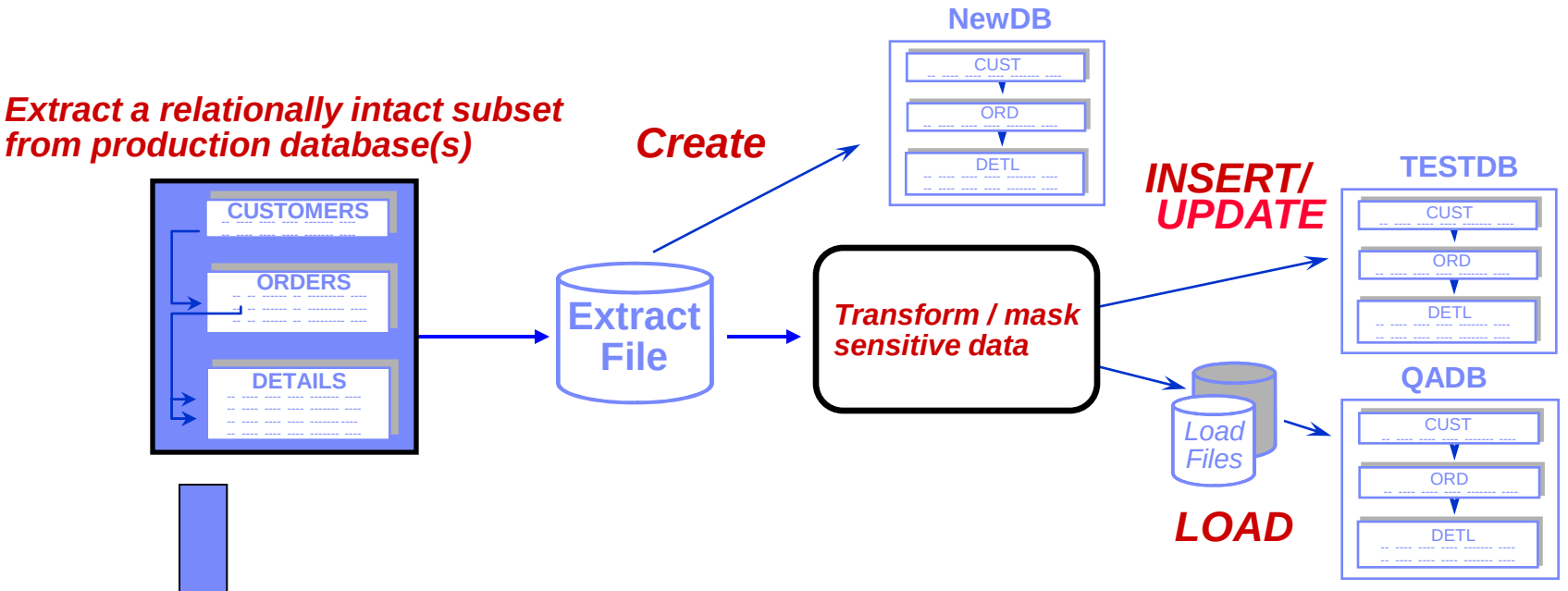
Or

During Insert/Load Process

Transform or mask sensitive data using

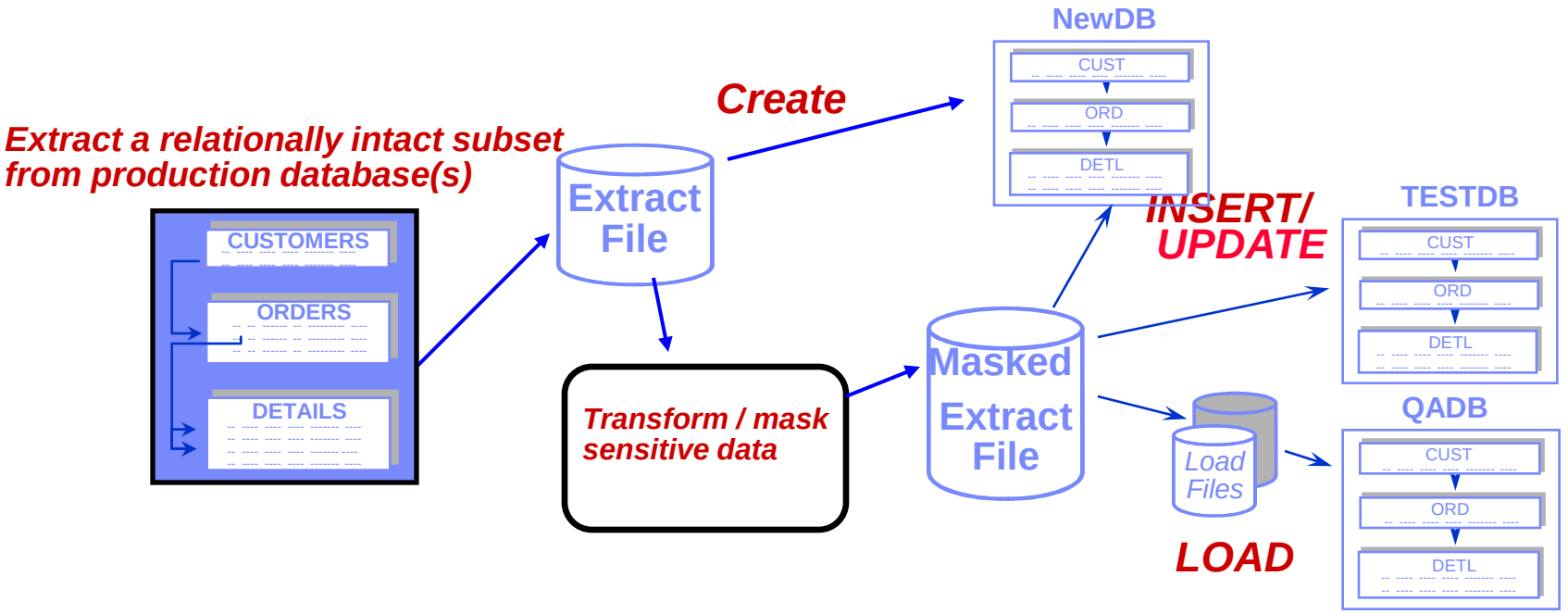
- Standard mapping rules: Literals, Special Registers, Expressions, Default Values, Look-up tables
- Complex mapping rules: User exits

Data Privacy in Application Testing



- Extract data and/or object definitions
- Define a new set of test tables
- Apply masking during population process
- Extract file may be reused but contains un-Masked data
- Good practice for testing masks

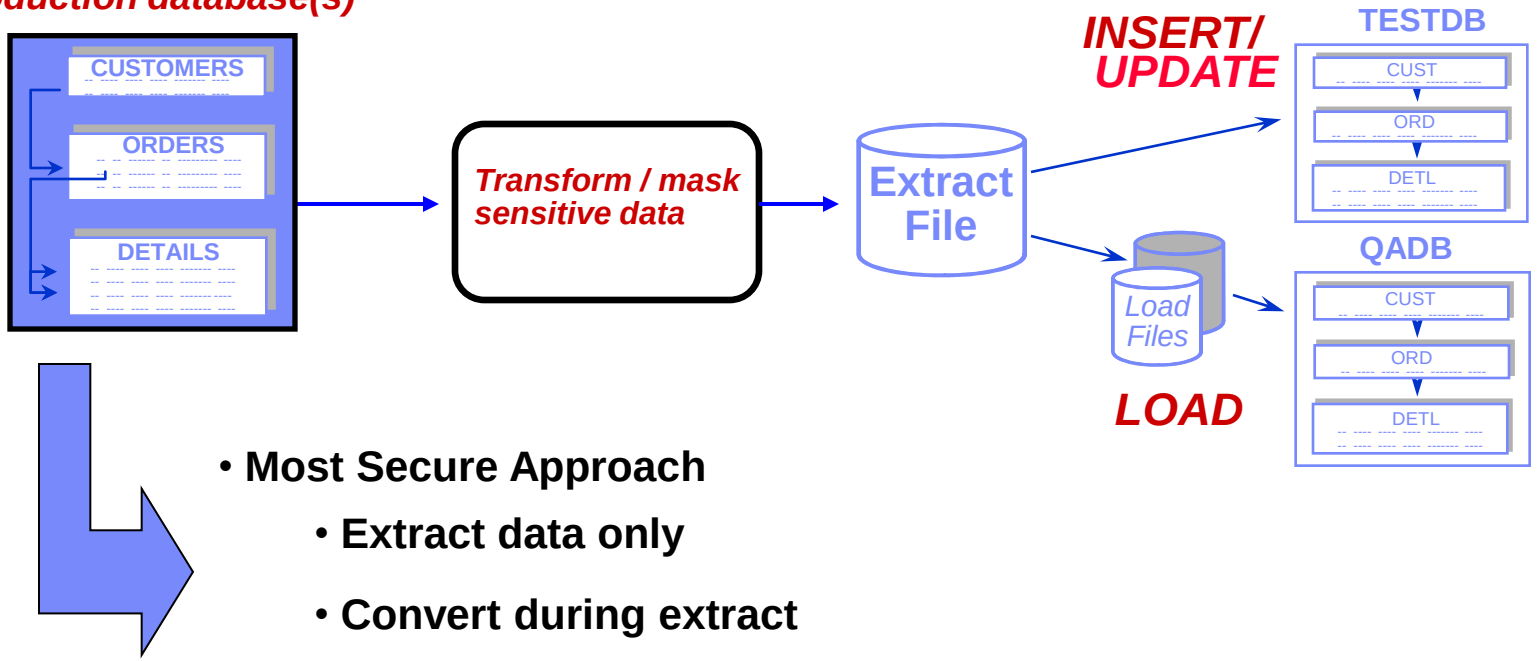
Data Privacy in Application Testing



- Extract data and/or object definitions in pre-masked file
- Use pre-masked Extract file to create new set of tables
- Convert Pre-masked extract file data into second masked extract file
- Share masked extract file to be reused for population step
- Good practice for testing masks using COMPARE

Data Privacy in Application Testing

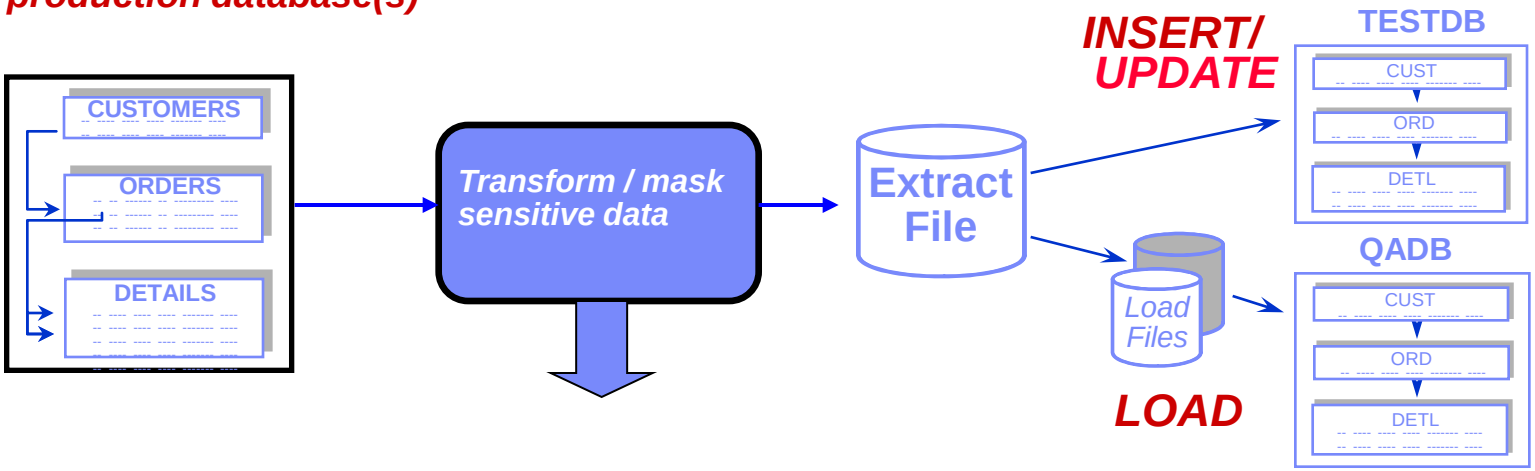
Extract a relationally intact subset from production database(s)



- **Most Secure Approach**
  - **Extract data only**
  - **Convert during extract**
- **Extract file already contains masked data**
  - **Can be shared with testers to reuse**

Data Privacy in Application Testing

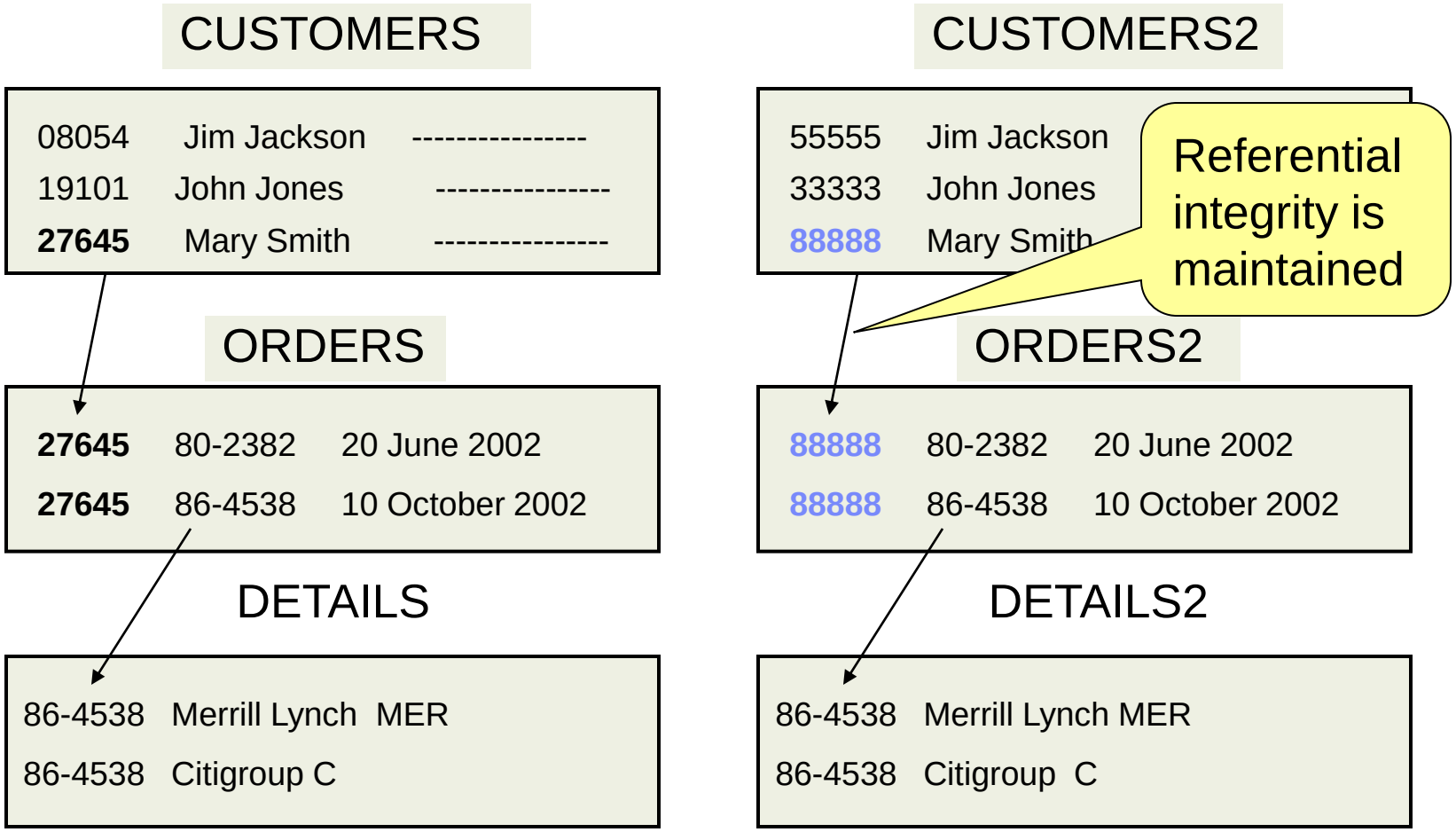
Extract a relationally intact subset from production database(s)



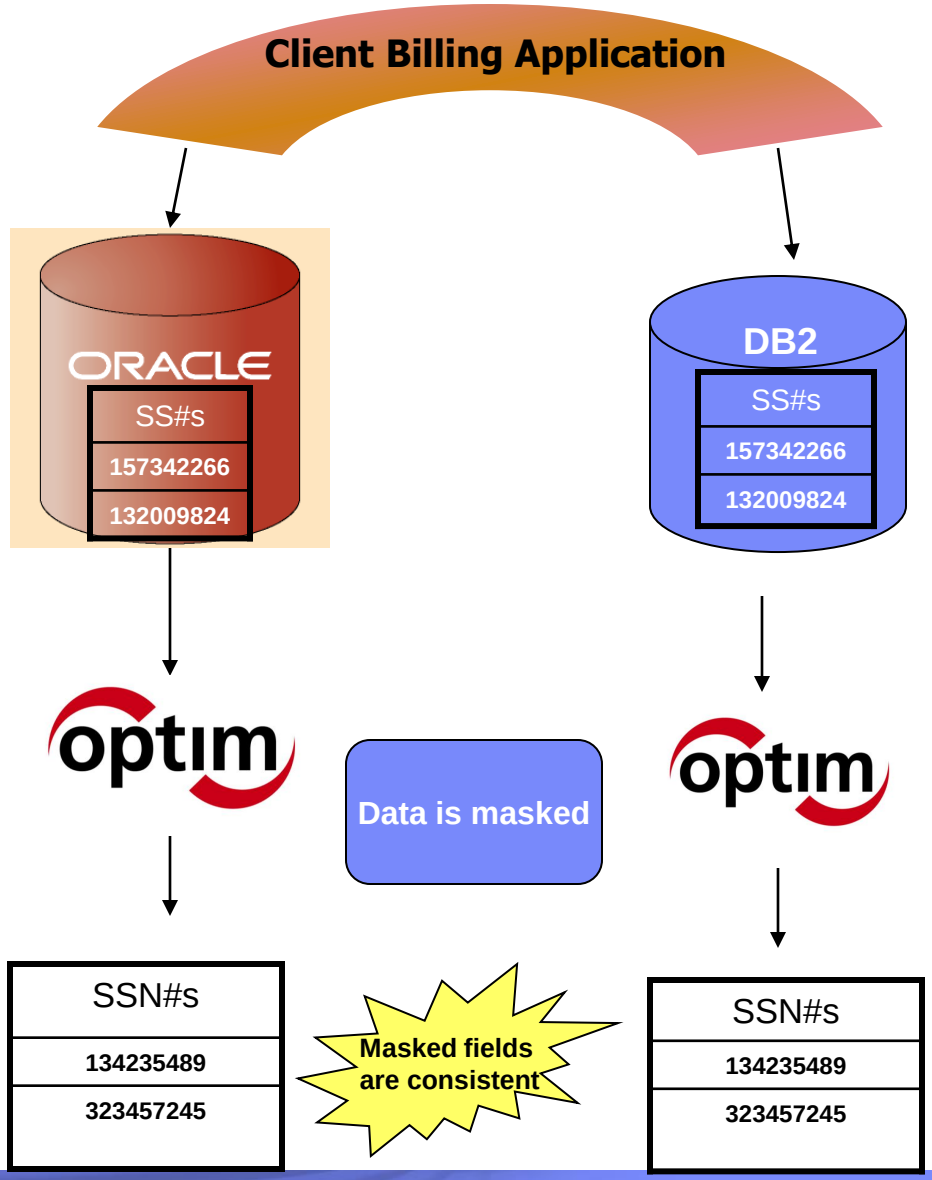
Data transformation functions:

- **Hard-code literals,**
- **special registers such as date, time**
- **Arithmetic calculations**
- **Sequential number generation**
- **Random number generation**
- **Substring and/or concatenation of values**
- **Lookup Table Functions Random, Specific or HASH**
- **Intellegent TRANformation Library – SSN, CCN,**
- **Access to client-defined exit routines to apply complex algorithms**
- **Propagation of masked primary keys to dependent foreign keys**

Propagating Keys



Consistent Masking across the Enterprise



First Names and Last Names Data Sets

Production Database

First Name	Last Name	GPA	High School	Advisor	State
Paul	Smith	3.2	Princeton	Johnson	NJ
Kate	Jones	2.7	Albany	Kline	NY

First Name Lookup Table

John
Bob
Danielle
Dave
Stacey

Last Name Lookup Table

Newton
Nelson
Kline
Howell
Reese

1) Client is a University who wishes to mask the first and last name fields in their admissions database

2) Optim now has a first name lookup table with over 5,000 male/female names and a last name lookup table with over 80,000 names

3) Use Lookup Tables to randomly replace table first and last names

Test Database

First Name	Last Name	GPA	High School	Advisor	State
Stacey	Nelson	3.2	Princeton	Johnson	NJ
Dave	Reese	2.7	Albany	Kline	NY

princeton softech an IBM® Company

Street Address/City/State/Zip Code Data Sets

Total Assets	Customers	Street	City	State	Zip Code
$534,674,233	**54,999**	**12 Buttercup Ln**	**Cleveland**	**OH**	**44101**
$8,777,733,811	**105,333**	**6767 Rte 1 S**	**Princeton**	**NJ**	**08540**

Address Lookup Table

1) Client is a Bank who wishes to mask its assets by location

288 Elm St	**Milwaukee**	**WI**	**53201**
12 Rodeo Dr	**Los Angeles**	**CA**	**90001**
3526 Diamond Rd	**Seattle**	**WA**	**98101**
12 Street Road	**Las Vegas**	**NV**	**89101**
2 Applegarth Ln	**Brunswick**	**ME**	**04011**

2) Optim provides corresponding Street Address/City/State/Zip Codes for masking

3) Leverage Multiple Column Replacement. Entire address row can be masked with a valid CASS address using enhanced random lookup function

New Table with Masked Data

Total Assets	Customers	Street	City	State	Zip Code
$534,674,233	**54,999**	3526 Diamond Rd	Seattle	WA	98101
$8,777,733,811	**105,333**	21 Street Rd	Las Vegas	NV	89101

princeton softech an IBM® Company

Intelligent Masking Capability

Production Database

F. Name	L. Name	Credit Card#	SSN#
John	**Denver**	**5298774132478855**	**254-77-6644**
Vanessa	**Jones**	**4324115574123634**	**154-74-7788**

Test Database

Valid

Valid

F. Name	L. Name	Credit Card#	SSN#
John	**Denver**	5326458711224956	854-77-6644
Vanessa	**Jones**	4972584612457744	154-74-7788

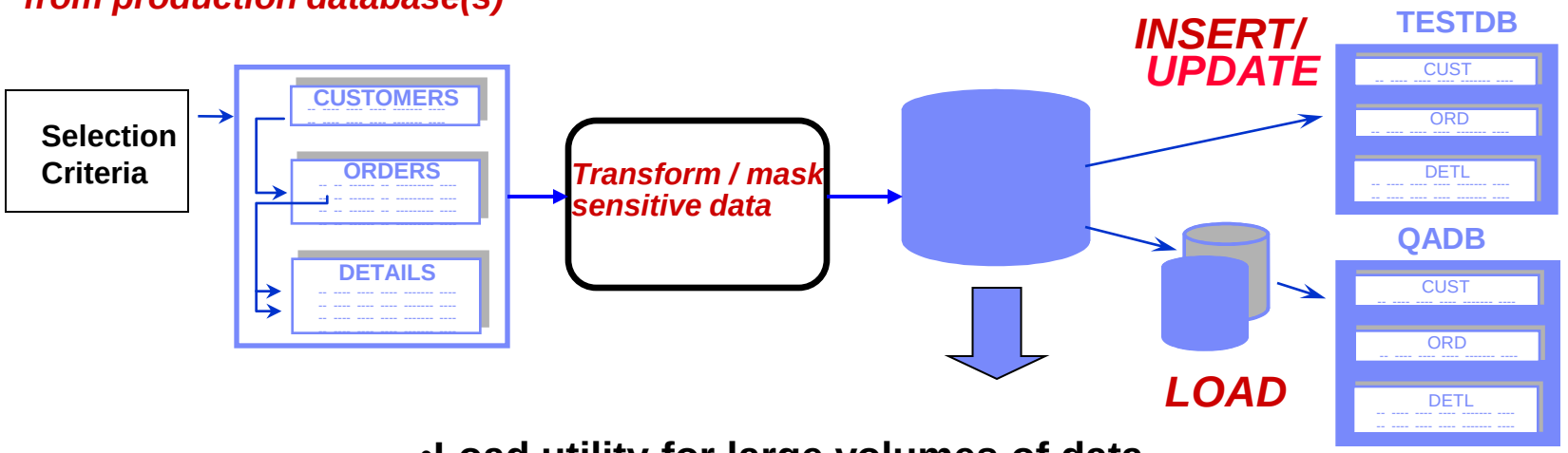
Data before Masking

Data after Masking… Masked with Valid CC# and SS#

How are these numbers valid?

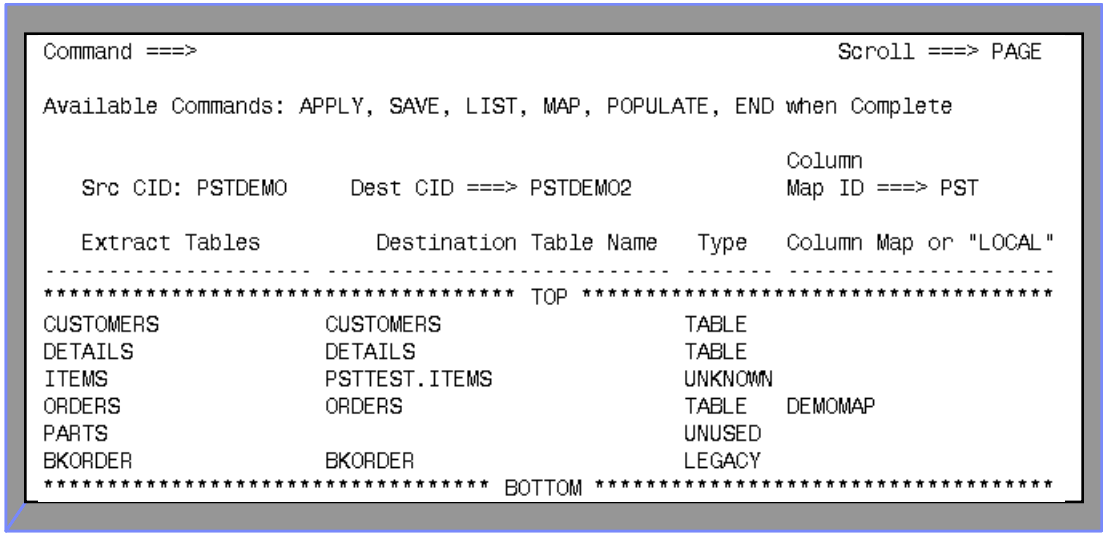
For Social Security Numbers	For Credit Card Numbers
A Social Security Number (SSN) consists of nine digits. The first three digits is called the "area number'. The central, two-digit field is called the "group Number". The final four-digit field is called the "serial Number". All numbers must fit the latest available criteria for each section.	Most credit card numbers are encoded with a "Check Digit". A check digit is a digit added to a number (either at the end or the beginning) that validates the authenticity of the number. A simple algorithm is applied to the other digits of the number which yields the check digit.

Data Privacy in Application Testing



- Load utility for large volumes of data
- Dynamic SQL
  - Insert new rows
  - Update existing rows; insert others
- Refresh from the Extract File
- Extract File maintains consistent baseline

Populate Destination Tables
Table Map



```
Command ===>                                                  Scroll ===> PAGE

Available Commands: APPLY, SAVE, LIST, MAP, POPULATE, END when Complete

                                                              Column
   Src CID: PSTDEMO     Dest CID ===> PSTDEMO2                Map ID ===> PST

   Extract Tables         Destination Table Name    Type    Column Map or "LOCAL"
--------------------- ------------------------- ------- ---------------------
*********************************** TOP ***************************************
CUSTOMERS             CUSTOMERS                 TABLE
DETAILS               DETAILS                   TABLE
ITEMS                 PSTTEST.ITEMS             UNKNOWN
ORDERS                ORDERS                    TABLE   DEMOMAP
PARTS                                           UNUSED
BKORDER               BKORDER                   LEGACY
********************************** BOTTOM *************************************
```

- **Table names need not match**
- **Change qualifier and/or table name**
- **Can be saved in PST Directory**

Populate Destination Tables
Column Map

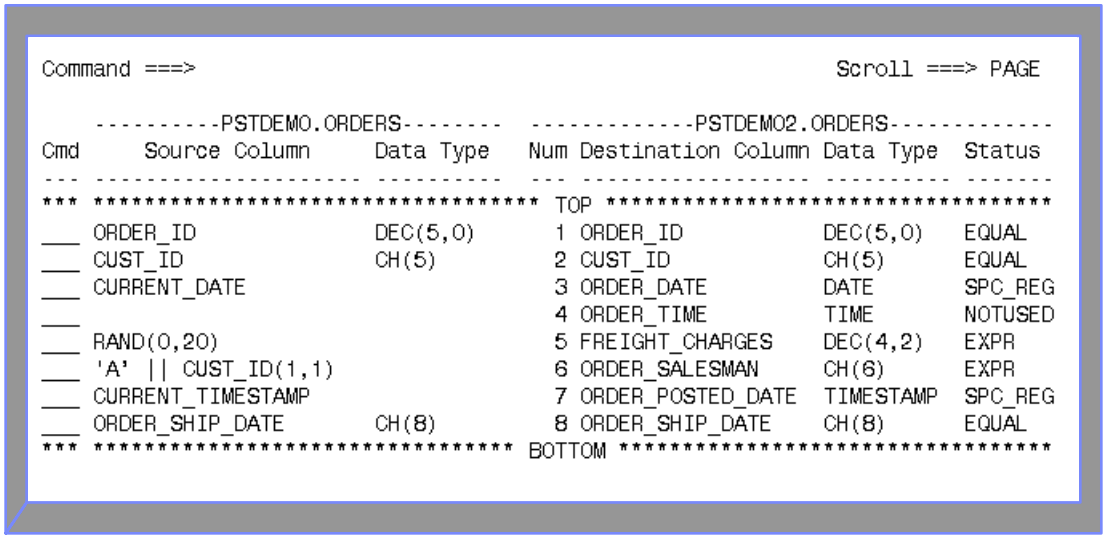
Literals

Special Registers

Expressions

Default Values

User exits

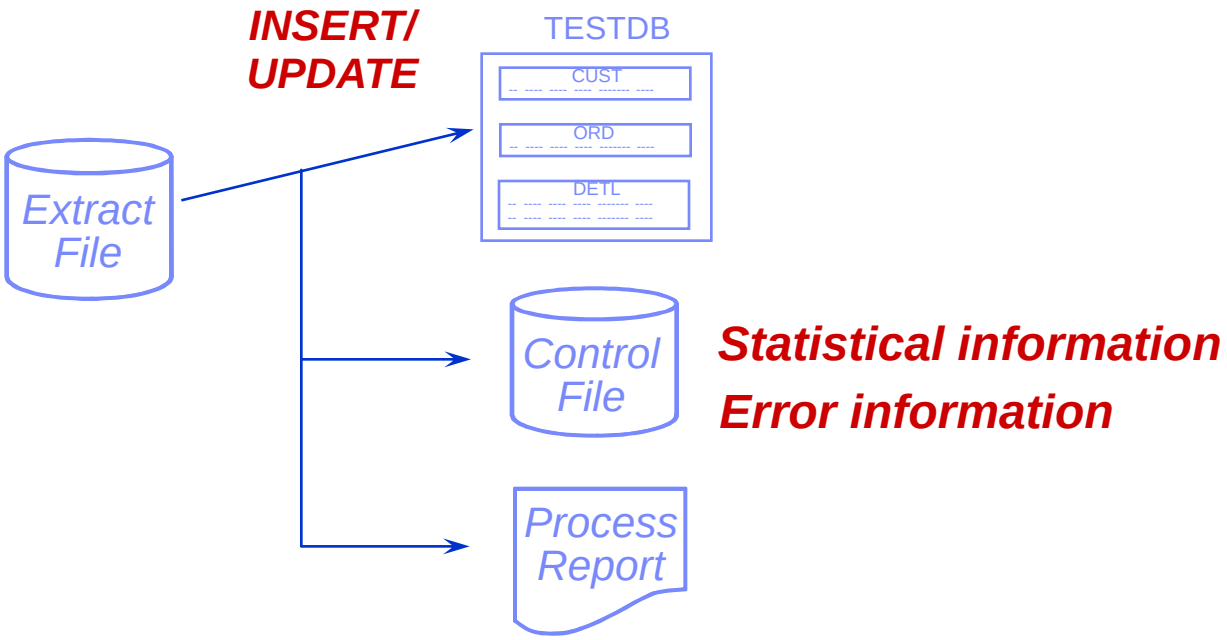


```
Command ===>                                                  Scroll ===> PAGE

    ----------PSTDEMO.ORDERS--------  -------------PSTDEMO2.ORDERS-------------
Cmd     Source Column      Data Type   Num Destination Column Data Type  Status
--- ---------------------- ----------  --- ------------------ ---------- -------
*** ********************************** TOP ************************************
___ ORDER_ID               DEC(5,0)     1 ORDER_ID           DEC(5,0)   EQUAL
___ CUST_ID                CH(5)        2 CUST_ID            CH(5)      EQUAL
___ CURRENT_DATE                        3 ORDER_DATE         DATE       SPC_REG
___                                     4 ORDER_TIME         TIME       NOTUSED
___ RAND(0,20)                          5 FREIGHT_CHARGES    DEC(4,2)   EXPR
___ 'A' || CUST_ID(1,1)                 6 ORDER_SALESMAN     CH(6)      EXPR
___ CURRENT_TIMESTAMP                   7 ORDER_POSTED_DATE  TIMESTAMP  SPC_REG
___ ORDER_SHIP_DATE        CH(8)        8 ORDER_SHIP_DATE    CH(8)      EQUAL
*** ********************************** BOTTOM *********************************
```

- **Map unlike column names**
- **Transform/mask sensitive data**
- **Datatype conversions**
- **Column-level date aging**

Populate Destination Tables

Control File

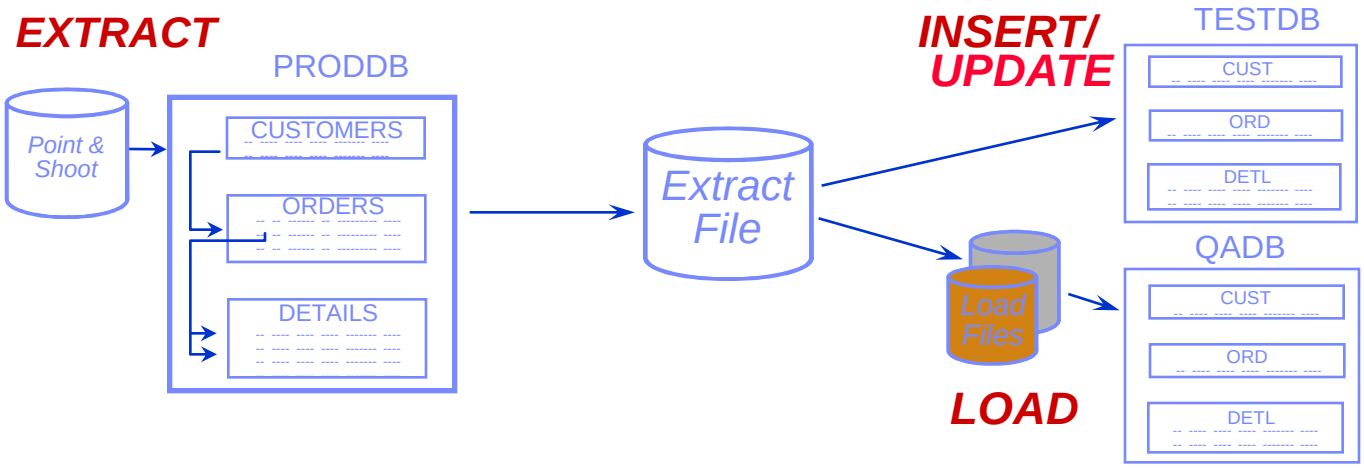


If INSERT/UPDATE errors occur:

- BROWSE the control file for error information
- RETRY/RESTART the INSERT/UPDATE process

The Relational Extract Facility

Summary



- **Creating and maintaining test data bases**
- **Migrating data**
- **Populating decision support data bases**

Optim™

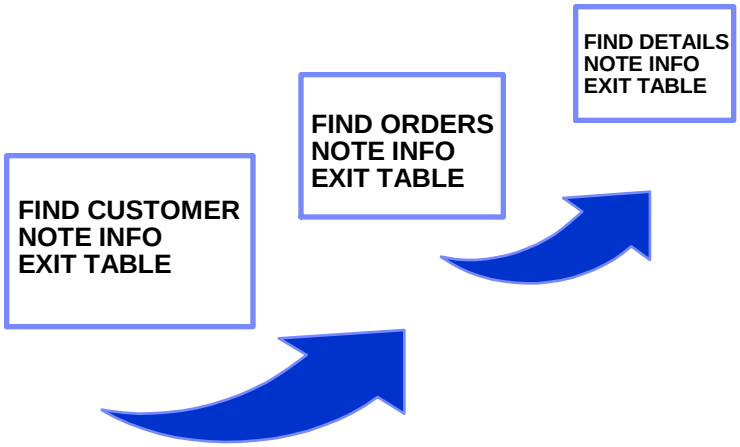
princeton softech
an IBM® Company

The Relational Editor

Traditional vs. Relational Tools

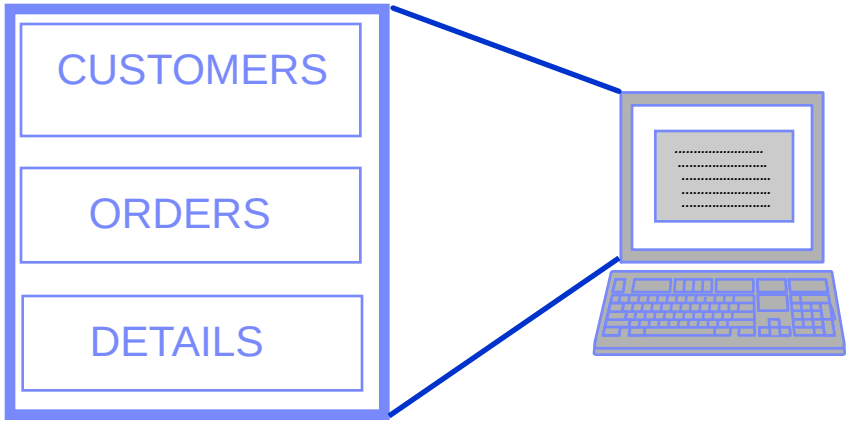
Single Table Editors

- **One table/view at a time**
- **No edit of related data from multiple tables**

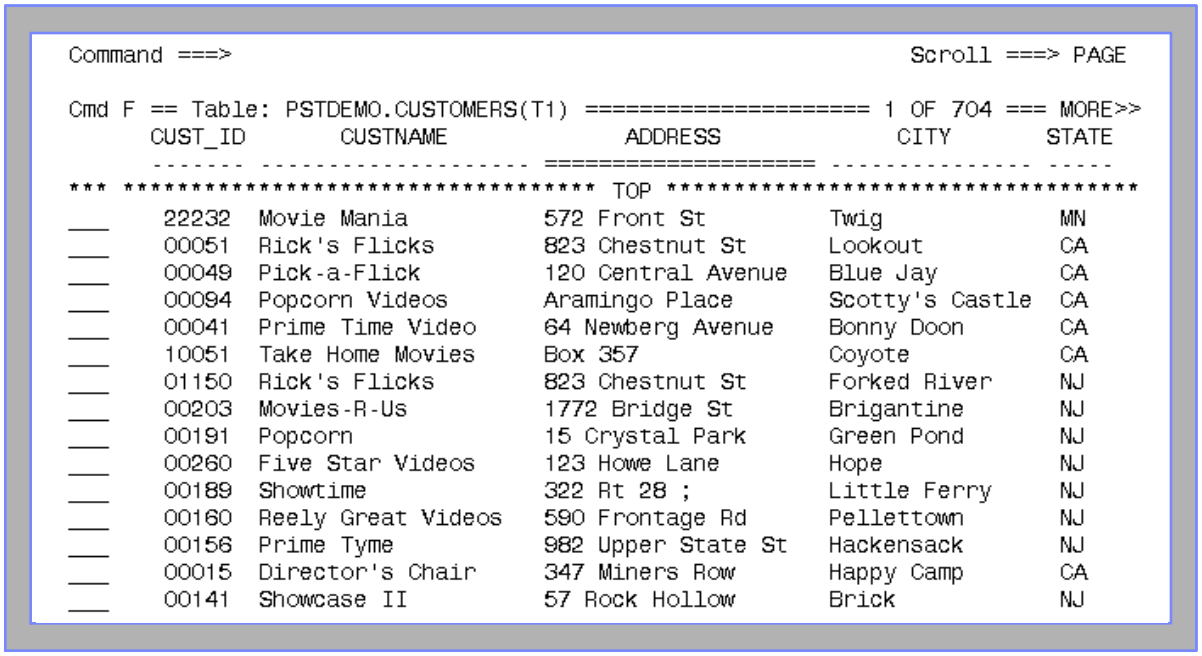


The Relational Editor

- ***Simultaneous* browse/edit of related data from multiple tables**



Browsing or Editing Data



```
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: PSTDEMO.CUSTOMERS(T1) ====================== 1 OF 704 === MORE>>
      CUST_ID       CUSTNAME              ADDRESS              CITY        STATE
      ------- ------------------- ==================== --------------- -----
*** ************************************ TOP ************************************
___     22232  Movie Mania          572 Front St         Twig            MN
___     00051  Rick's Flicks        823 Chestnut St      Lookout         CA
___     00049  Pick-a-Flick         120 Central Avenue   Blue Jay        CA
___     00094  Popcorn Videos       Aramingo Place       Scotty's Castle CA
___     00041  Prime Time Video     64 Newberg Avenue    Bonny Doon      CA
___     10051  Take Home Movies     Box 357              Coyote          CA
___     01150  Rick's Flicks        823 Chestnut St      Forked River    NJ
___     00203  Movies-R-Us          1772 Bridge St       Brigantine      NJ
___     00191  Popcorn              15 Crystal Park      Green Pond      NJ
___     00260  Five Star Videos     123 Howe Lane        Hope            NJ
___     00189  Showtime             322 Rt 28 ;          Little Ferry    NJ
___     00160  Reely Great Videos   590 Frontage Rd      Pellettown      NJ
___     00156  Prime Tyme           982 Upper State St   Hackensack      NJ
___     00015  Director's Chair     347 Miners Row       Happy Camp      CA
___     00141  Showcase II          57 Rock Hollow       Brick           NJ
```

- **User can define how data is displayed**
  - SORT, HEX, sidelabel/columnar format
- **All DB2 access authority enforced**

Joining to Another Table

JOIN [table]

```
Command ===>                                                  Scroll ===> PAGE

Cmd F == Table: PSTDEMO.CUSTOMERS(T1) ======================= 1 OF 36 === MORE>>
      CUST_ID       CUSTNAME              ADDRESS              CITY        STATE
      ------- -------------------- ==================== --------------- -----
___    00068  Audio-Video World    593 West 37th Street Angels Camp       CA

Cmd F == Table: PSTDEMO.ORDERS(T2) ============================ 1 OF 4 === MORE>>
      ORDER_ID CUST_ID ORDER_DATE ORDER_TIME FREIGHT_CHARGES ORDER_SALESMAN
      -------- ------- ---------- ---------- --------------- --------------
*** ********************************** TOP ************************************
___         23   00068  12/02/1997  08.16.09        14.80          WE005
___        222   00068  12/31/1997  14.22.31        19.05          WE005
___        278   00068  02/02/1998  11.51.47        21.97          WE005
___      30013   00068  01/12/1998  15.23.04        33.85          WE005
*** ******************************** BOTTOM ***********************************
```

- **Simultaneous edit/browse of data**
- **Scroll of higher-level table automatically synchronizes all lower-joined tables**

OPTIM Relational Editor

The Programmer's Solution

OPTIM Relational Editor helps you to:

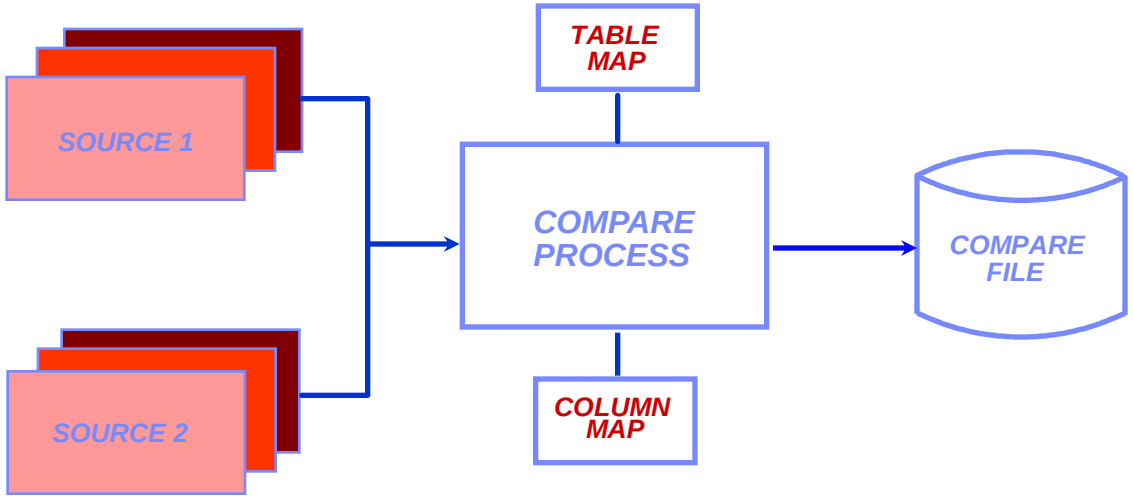
- **Understand the data your application is to process**
- **Create data values to test program logic**
- **Inspect and correct data that is causing problems**
- **Verify execution results**

Optim™

princeton softech
an IBM® Company

The Relational Compare Facility

OPTIM Relational Compare Facility



- **Single-table or multi-table compare**
- **Creates compare file of results**
- **Displays results on screen**

princeton softech an IBM® Company

Browsing the Compare File

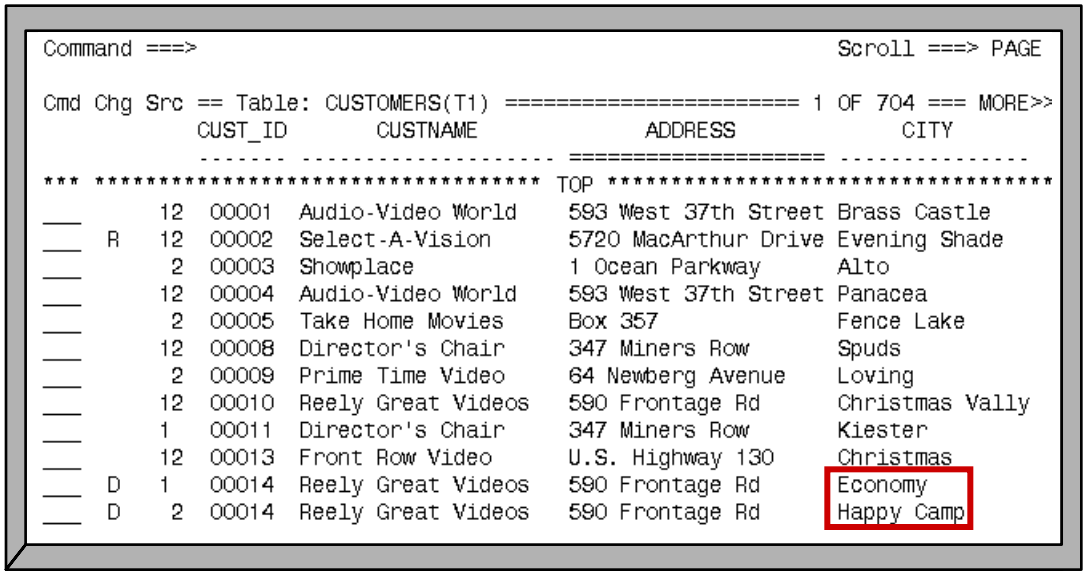
Compare Statistics

```
Command ===>                                                  Scroll ===> PAGE

Source 1: XF - PST.ADB2.PSTDEMO.EXTRACT, SUBSYS: PDB2
Source 2: DB2 Tables, SUBSYS: TDB2
                                                                   Rows    Non-
                                                        Changes    with    Unique
                                Total  UnMatched  Equal  (D)irect  Missing Match
Sel   Source:Table Name         Rows     Rows     Rows   (R)elated Parents  Keys
--- --------------------- -------- -------- -------- --------- ------- -------
___  1:PSTPROD.CUSTOMERS        702        0      689 D:     13       0       0
     2:PSTTEST.CUSTOMERS        704        2          R:     10       0       0

___  1:PSTPROD.ORDERS          1711        9     1698 D:      4       0       0
     2:PSTTEST.ORDERS          1709        7          R:      4       2       0
```

- **Shows statistics for each pair of tables**
- **Identifies tables containing orphan rows**
- **Identifies tables with duplicate match keys**

princeton softech an IBM® Company

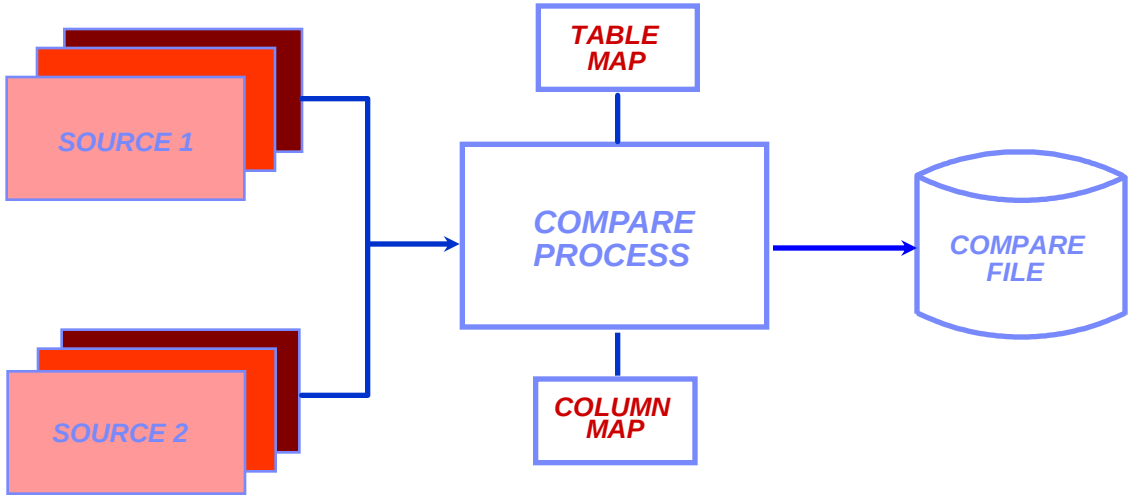
Browsing the Compare File



```
Command ===>                                                  Scroll ===> PAGE

Cmd Chg Src == Table: CUSTOMERS(T1) ======================= 1 OF 704 === MORE>>
            CUST_ID       CUSTNAME             ADDRESS              CITY
            ------- -------------------- ==================== ---------------
*** ********************************** TOP ************************************
___         12  00001  Audio-Video World    593 West 37th Street Brass Castle
___   R     12  00002  Select-A-Vision      5720 MacArthur Drive Evening Shade
___         2   00003  Showplace            1 Ocean Parkway      Alto
___         12  00004  Audio-Video World    593 West 37th Street Panacea
___         2   00005  Take Home Movies     Box 357              Fence Lake
___         12  00008  Director's Chair     347 Miners Row       Spuds
___         2   00009  Prime Time Video     64 Newberg Avenue    Loving
___         12  00010  Reely Great Videos   590 Frontage Rd      Christmas Vally
___         1   00011  Director's Chair     347 Miners Row       Kiester
___         12  00013  Front Row Video      U.S. Highway 130     Christmas
___   D     1   00014  Reely Great Videos   590 Frontage Rd      Economy
___   D     2   00014  Reely Great Videos   590 Frontage Rd      Happy Camp
```

- **SRC column identifies input source of row**
- **CHG column identifies the type of change**
- **Data differences are highlighted**

OPTIM Relational Compare Facility



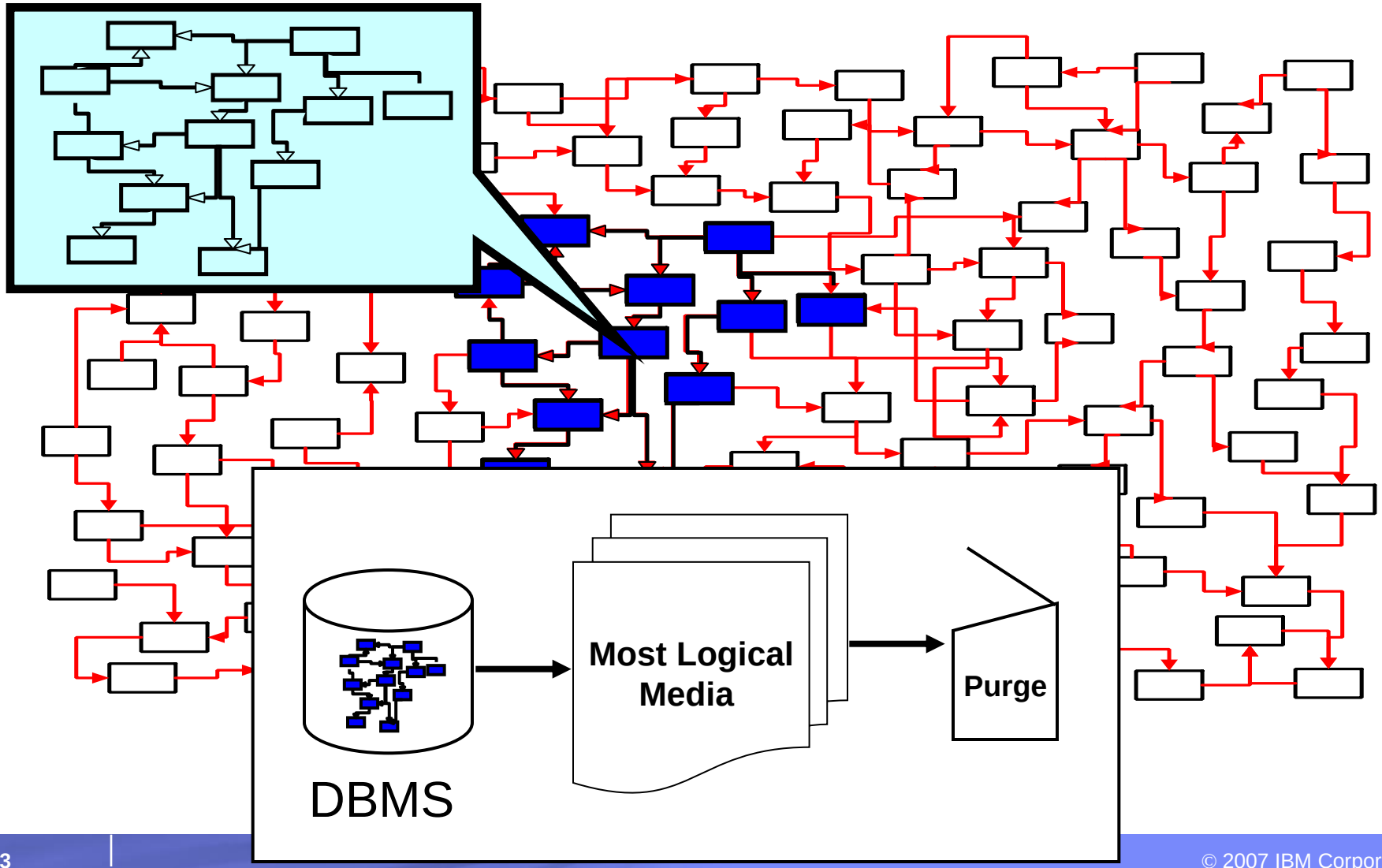
- **For application testing, QA, and to verify database contents**
- **Enhances productivity by finding unexpected changes in the data**

Optim™

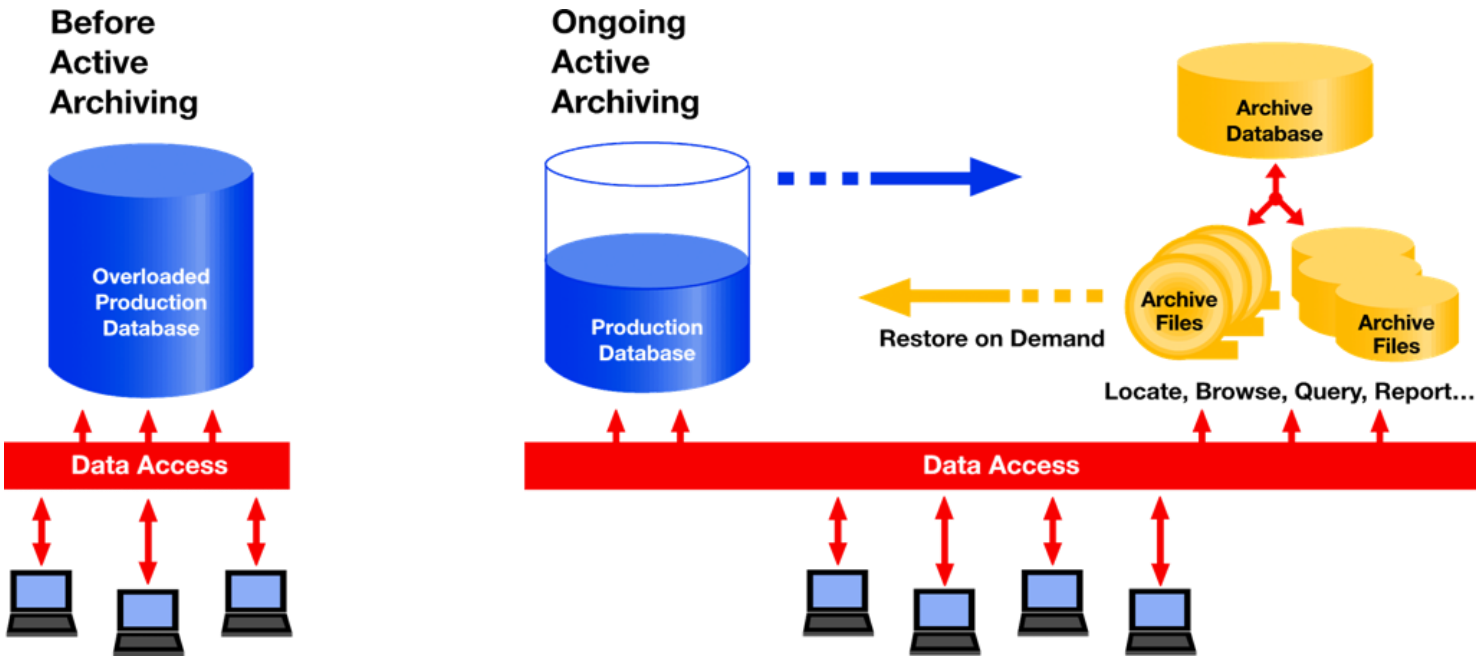
princeton softech
an IBM® Company

The Relational Archive Facility

Challenge: Referential Complexity



Active Archiving Defined



- **Reduce the amount of data in the application database by:**
  - Separating infrequently accessed data from transactional data
  - Preserve metadata and relationships of archived data outside db
  - Archive relational subsets vs. entire files
- **Enable easy user access to archived information**
  - View, research and restore as needed
- **Complementary to Information Lifecycle Management (ILM)**

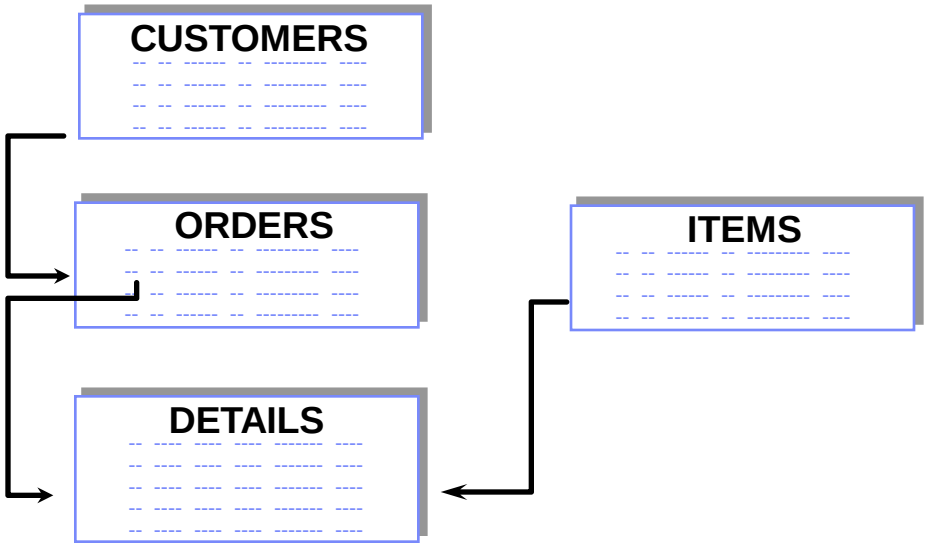
Steps for Archiving Data

- Identify the data to be archived
- Define the data to be deleted
- Create the archive & Delete the data
- Find Data in the Archives
- Browse, Report or Restore

Identify the data to be archived

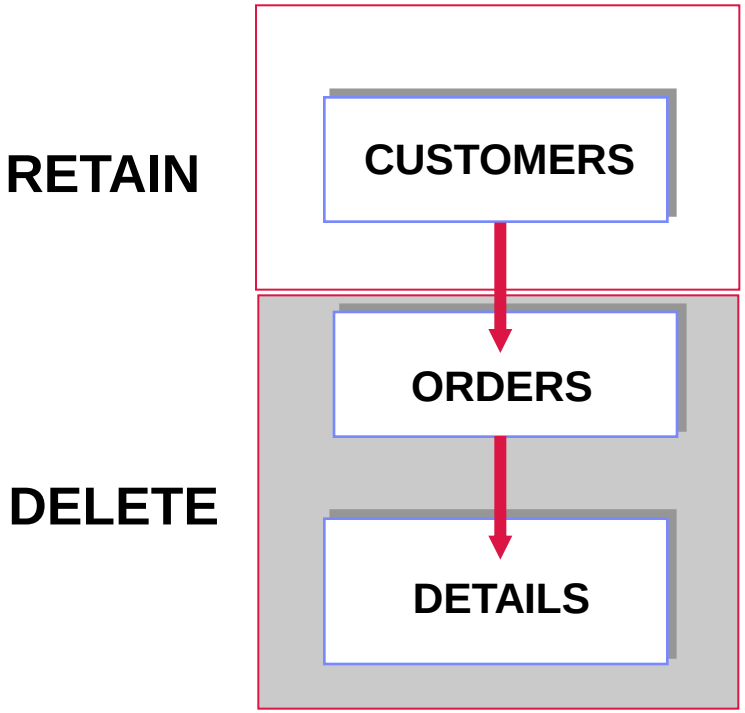
Access Definition

Defines a subset of of relational data



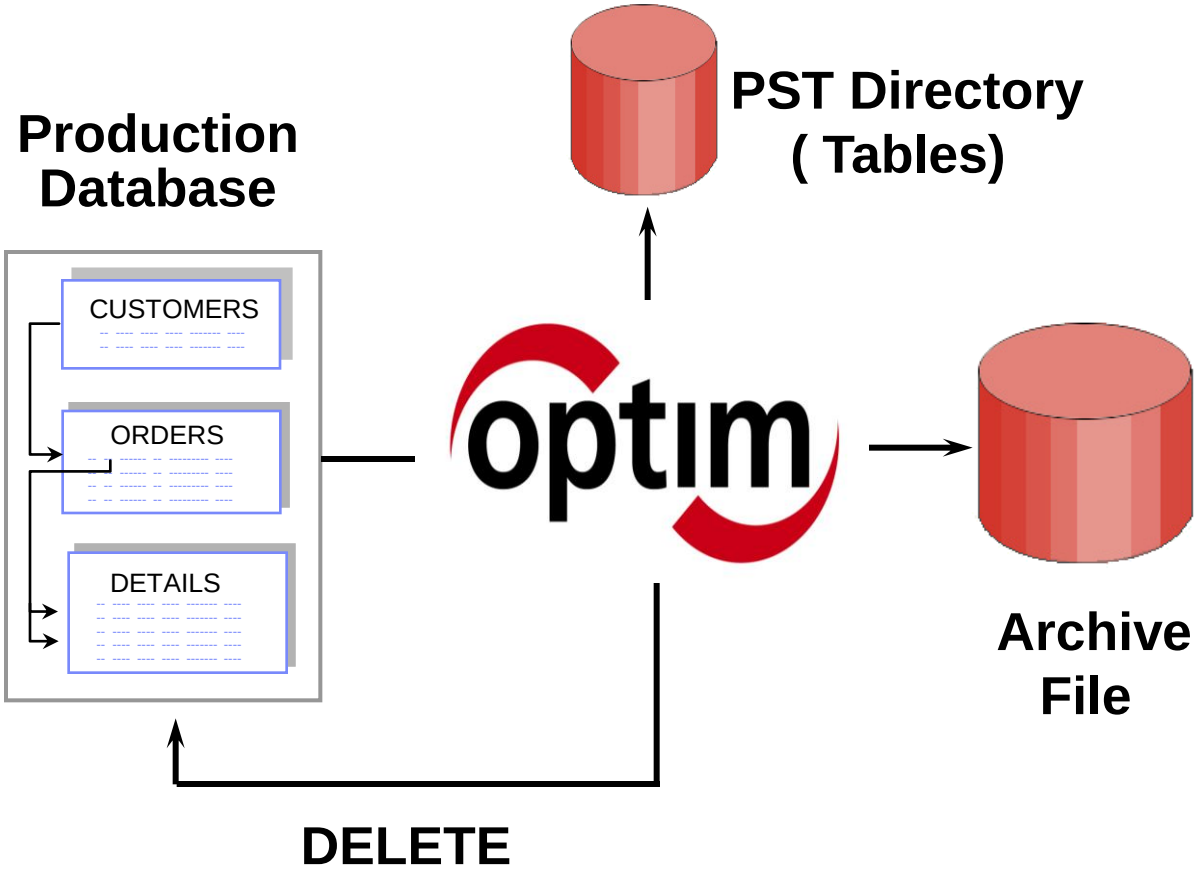
- **Start table**
- **Associated data**
- **Relationships**
- **Extraction rules**
- **Index specifications**

Define the data to be deleted

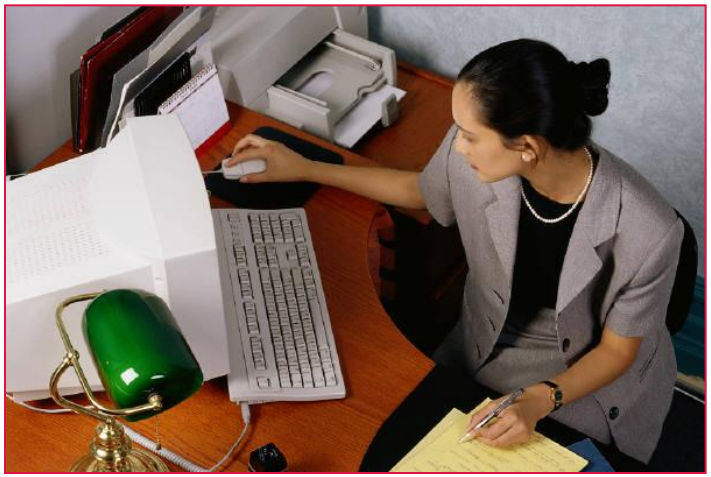


- Archive all data
- Delete orders and details after they are safely archived
- Preserve semantic intelligence

Create the archive



Researching the Archives

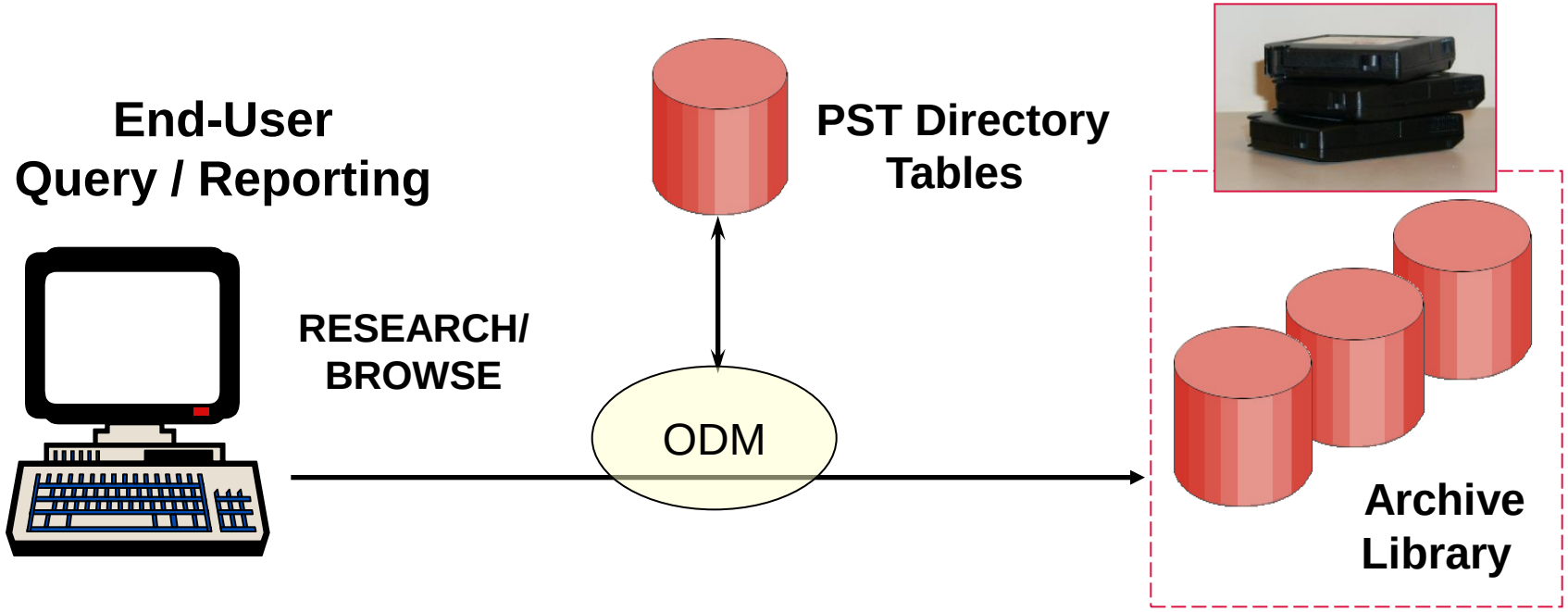


Direct access to archived data:

- **User maintainable indexes**
- **Global searches**
- **Simple or complex criteria**
- **Intelligent browse**
- **ODM access**

Restore archived data only when you need to

Applications accessing the Archive Files



Use the OPTIM Archive ODM Option

- ✓Direct Access within Your Application using standard SQL
- ✓Defines data-sources for any ODBC or JDBC application
- ✓Joins between multiple data-sources
  - ✓archive files and database tables

Why Restore?

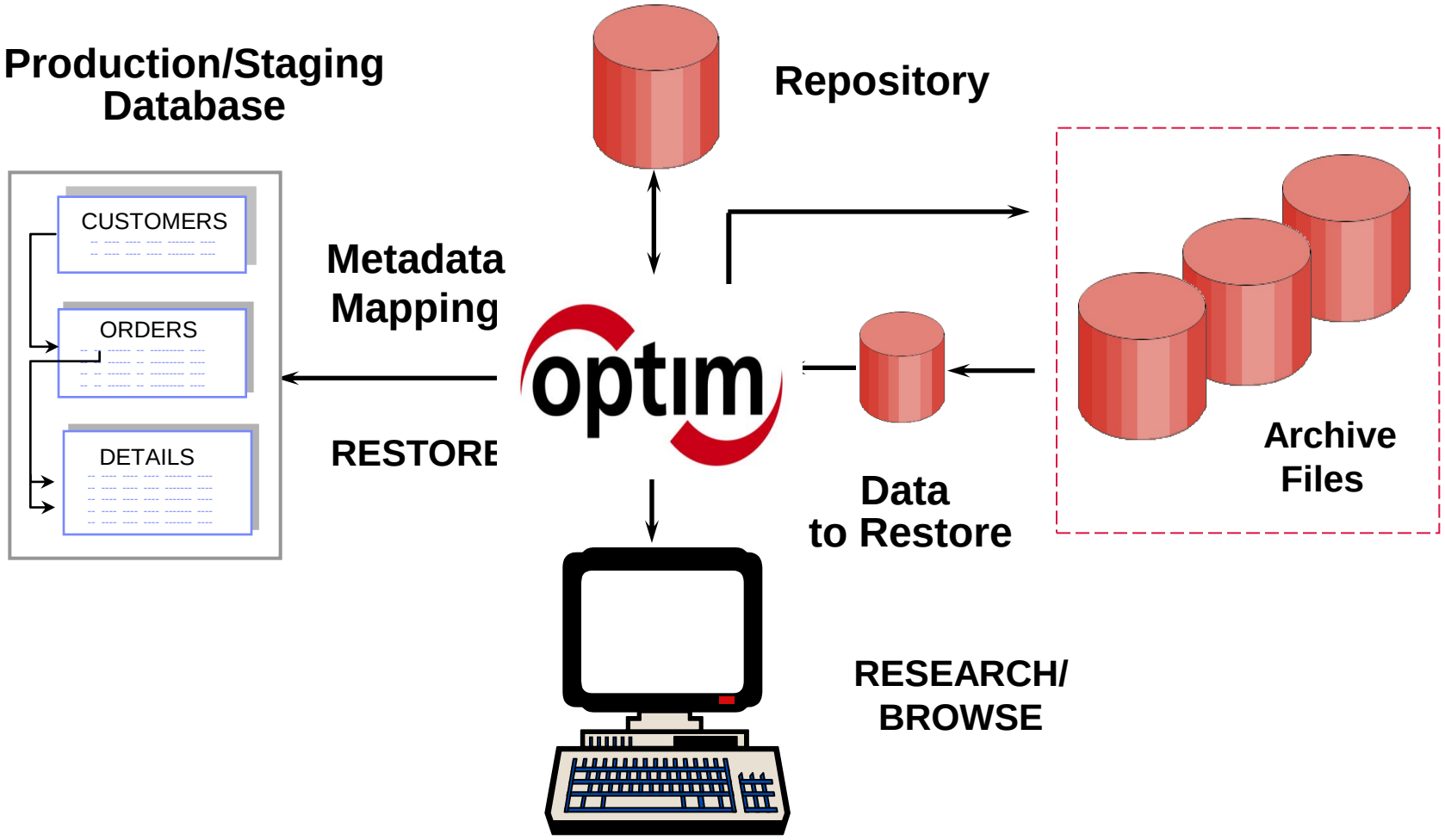
Browse archived data for:

- Customer service
- Answering questions
- Archive research

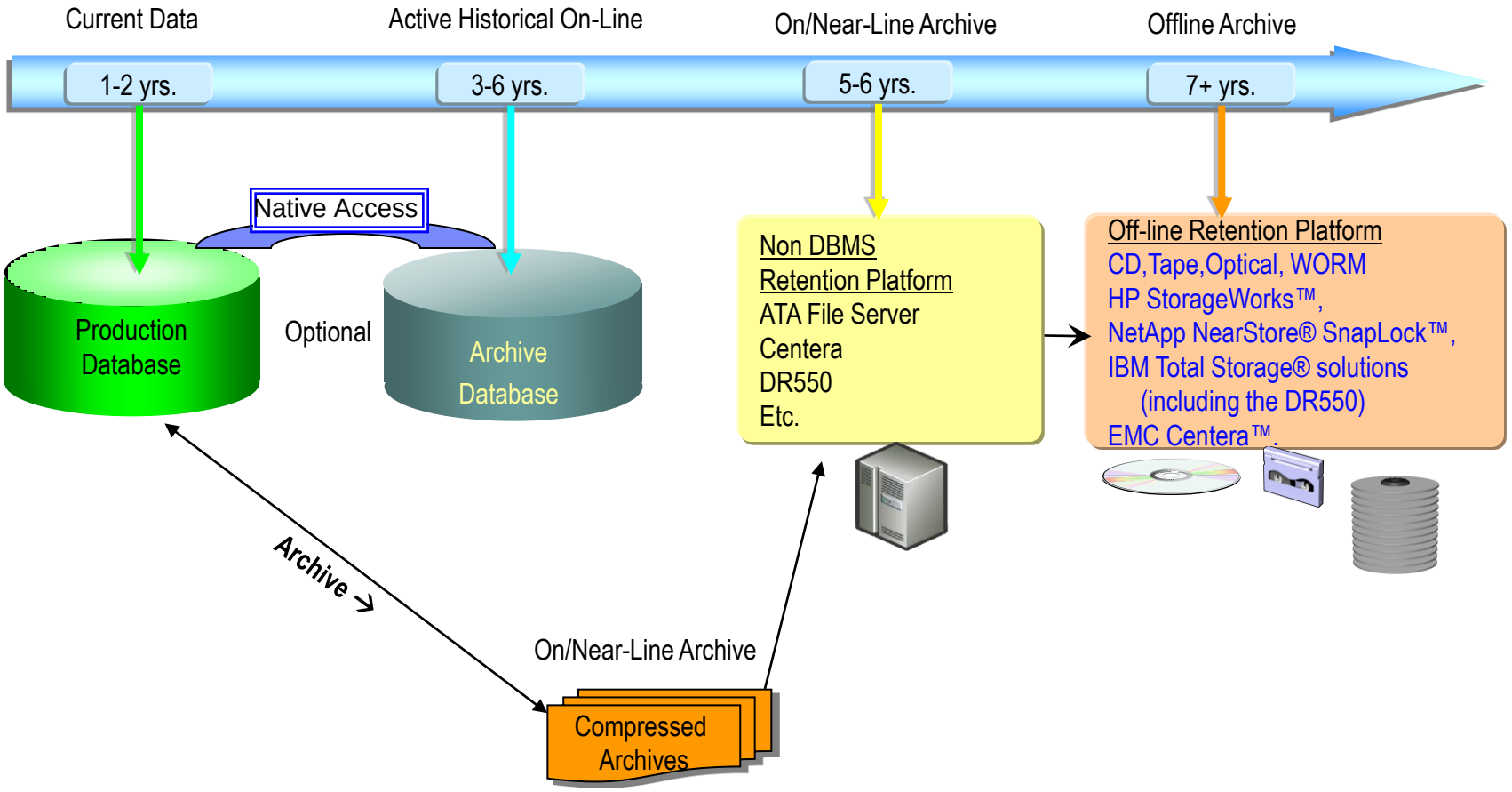
Restore archived data for:

- Transparent Access
- Audit situations
- Application-generated reports

Restoring Archived Data



Store - Data Retention Strategies



EDM Solution Requirements – The Four Pillars

1. **Enterprise Architecture**
2. **Complete Business Object**
3. **Extract, Store, Port and Protect**
4. **Universal Access**

These highlight the differentiators – use the COMPEDITIVE MATRIX to get more details.