



The next step in big data starts with IBM.
Extreme speed. Increased flexibility. Enhanced performance.



Fonctions Analytiques de DB2 V10 et V11

Cécile Benhamou

Technical Sales DB2 z/OS et Tools DB2
cecile_benhamou@fr.ibm.com



DB2 11 for z/OS

The Enterprise Data Server for Business
Critical Transactions and Analytics.





DB2 V10: Fonctions OLAP



- Moving Sum et Moving Average
 - Fonctions colonnes (aggregate) qui calculent une valeur unique basée sur une ligne ou un groupe de lignes
 - Supporte des sommes ou des moyennes cumulées en utilisant une “fenêtre”.
 - Une “fenêtre” peut spécifier 3 options:
 - Partitionnement de la table résultante en utilisant la clause PARTITION BY
 - Tri des lignes à l’intérieur d’une partition en utilisant la clause ORDER BY
 - Aggrégation par groupe en utilisant la clause ROW ou RANGE
 - 2 options sont obligatoires pour spécifier un groupe d’aggrégation
 - Mot clé ROWS pour indiquer un groupe physique ... ou
 - Mot clé RANGE pour indiquer un groupe logique... et
 - La ligne de début et de fin pour le groupe d’aggrégation
 - Ne peut pas être utilisé avec XMLQUERY, XMLEXISTS, ni comme un argument d’une fonction colonne
 - Nouveau code SQL -20117 pour indiquer une fenêtre invalide





DB2 V10: Fonctions OLAP ... Exemples



- Table SALES_HISTORY
 - Chaque TERRITORY a 5 lignes.

<u>TERRITORY</u>	<u>MONTH</u>	<u>SALES</u>
EAST	200910	10.00
WEST	200910	8.00
EAST	200911	4.00
WEST	200911	12.00
EAST	200912	10.00
WEST	200912	7.00
EAST	201001	7.00
WEST	201001	11.00
EAST	201002	9.00
WEST	201002	7.00





DB2 V10: Fonctions OLAP ... Exemples



```
SELECT TERRITORY, MONTH, SALES, AVG(SALES)
OVER(PARTITION BY TERRITORY ORDER BY MONTH
ROWS 2 PRECEDING) AS MOVING_AVG
FROM SALES_HISTORY
```

TERRITORY	MONTH	SALES	MOVING_AVG
EAST	200910	10.00	10.00
EAST	200911	4.00	7.00
EAST	200912	10.00	8.00
EAST	201001	7.00	7.00
EAST	201002	9.00	8.66
WEST	200910	8.00	8.00
WEST	200911	12.00	10.00
WEST	200912	7.00	9.00
WEST	201001	11.00	10.00
WEST	201002	7.00	8.33

Résultats:

1. Groupé par TERRITORY
2. Trié par MONTH
3. MOVING_AVG: les valeurs sont calculées sur la valeur moyenne de SALES pour la ligne courante avec les 2 lignes précédentes à l'intérieur de TERRITORY.





DB2 V10: Fonctions OLAP ... Exemples



```
SELECT TERRITORY, MONTH, SALES, SUM(SALES)
OVER(PARTITION BY TERRITORY ORDER BY MONTH
     ROWS UNBOUNDED PRECEDING) AS CUMULATIVE_SUM
FROM SALES_HISTORY
```

“ROWS UNBOUNDED PRECEDING” est optionnel. Résultats identiques sans.

TERRITORY	MONTH	SALES	CUMULATIVE SUM
EAST	200910	10.00	10.00
EAST	200911	4.00	14.00
EAST	200912	10.00	24.00
EAST	201001	7.00	31.00
EAST	201002	9.00	40.00
WEST	200910	8.00	8.00
WEST	200911	12.00	20.00
WEST	200912	7.00	27.00
WEST	201001	11.00	38.00
WEST	201002	7.00	45.00

Résultats:

1. Groupé par TERRITORY
2. Trié par MONTH
3. CUMULATIVE_SUM: les valeurs sont calculées sur la somme de SALES pour la ligne courante avec les lignes précédentes dans chaque TERRITORY.





DB2 V11: Fonctions Data Warehouse



- SQL Grouping Sets -> Rollup, Cube
 - Requêtes plus simples et rapides en produisant un seul résultat en exécutant automatiquement un UNION ALL de 2 (ou plus) groupes de lignes
- Clause Group By améliorée:
 - *grouping-sets*
 - Plusieurs clauses de *grouping* peuvent être spécifiées dans un seul ordre
 - Permet aux groupes d'être calculés en une seule passe sur les données
 - Peut être utilisé pour déterminer les sous-totaux et grands totaux
 - i.e. Pour un groupe donné, calcule son sous-total ainsi que le grand total pour tous les groupes
 - *super-groups*
 - *grouping-sets* prédéfinis
 - ROLLUP: permet de fournir des sous-totaux sur une dimension hiérarchique comme le temps ou la géographie
 - CUBE: utile pour les requêtes qui font des agrégations basées sur des colonnes de plusieurs dimensions





DB2 V11: Grouping Sets : ROLLUP



- Crée des sous-totaux qui remontent ('roll up') depuis le niveau le plus détaillé jusqu'au grand total
- N éléments translatés en N+1 grouping sets:

GROUP BY ROLLUP (C1,C2,C3) équivalent à GROUP BY GROUPING SETS ((C1,C2,C3),
(C1,C2),
(C1),
());

- Le résultat aurait pu être obtenu en utilisant UNION ALL des 4 (N+1) grouping sets:

```
SELECT .... FROM ... GROUP BY C1, C2, C3
UNION ALL
SELECT .... FROM ... GROUP BY C1, C2
UNION ALL
SELECT .... FROM ... GROUP BY C1
UNION ALL
SELECT .. FROM.... ;
```





DB2 V11: Grouping Sets : ROLLUP



- L'ordre spécifié est important pour le résultat

GROUP BY ROLLUP (C1,C2) équivalent à GROUP BY GROUPING SETS ((C1,C2),
(C1),
());

VERSUS

GROUP BY ROLLUP (C2,C1) équivalent à GROUP BY GROUPING SETS ((C2,C1),
(C2),
());





DB2 V11: Grouping Sets : CUBE



- Crée des sous-totaux pour toutes les permutations
- N éléments traduits en 2^n grouping sets:

GROUP BY CUBE (C1,C2,C3) équivalent à GROUP BY GROUPING SETS ((C1,C2,C3),
(C1,C2), (C1,C3), (C2,C3),
(C1), (C2),(C3),
());

3 éléments

8 groupes

grand total

- A la différence de ROLLUP, l'ordre spécifié n'est pas significatif pour le résultat
 - i.e. utiliser ORDER BY pour garantir l'ordre
- “grand-total”
 - L'agrégation des sous totaux précédents
 - CUBE et ROLLUP retournent cette ligne en tant que dernière ligne





DB2 V11: Grouping Sets : CUBE



- Le résultat aurait pu être obtenu en utilisant UNION ALL des 8 (2 puissance n) grouping sets:

```
SELECT .... FROM ... GROUP BY C1, C2, C3
UNION ALL
SELECT .... FROM ... GROUP BY C1, C2
UNION ALL
SELECT .... FROM ... GROUP BY C1, C3
UNION ALL
SELECT .... FROM ... GROUP BY C2, C3
UNION ALL
SELECT .... FROM ... GROUP BY C1
UNION ALL
SELECT .... FROM ... GROUP BY C2
UNION ALL
SELECT .... FROM ... GROUP BY C3
UNION ALL
SELECT .. FROM.... ;
```





DB2 V11: Grouping Sets on DSN8B10.PROJ



DEPTNO	RESPEMP	PRSTAFF
B01	000020	1.00
C01	000030	2.00
C01	000030	1.00
D01	000010	12.00
D01	000010	6.50
D11	000060	9.00
D11	000220	2.00
D11	000150	3.00
D11	000160	3.00
D21	000070	6.00
D21	000230	2.00
D21	000250	1.00
D21	000270	2.00
E01	000050	6.00
E01	000050	5.00
E11	000090	5.00
E21	000100	4.00
E21	000320	1.00
E21	000330	1.00
E21	000340	1.00

20 rows originally

DEPTNO	RESPEMP	PRSTAFF
B01	000020	1.00
C01	000030	3.00
D01	000010	18.50
D11	000060	9.00
D11	000150	3.00
D11	000160	3.00
D11	000220	2.00
D21	000070	6.00
D21	000230	2.00
D21	000250	1.00
D21	000270	2.00
E01	000050	11.00
E11	000090	5.00
E21	000100	4.00
E21	000320	1.00
E21	000330	1.00
E21	000340	1.00

17 rows after grouping

```
SELECT DEPTNO, RESPEMP,
       SUM(PRSTAFF)
FROM DSN8B10.PROJ
GROUP BY DEPTNO,RESPEMP;
```

equivalent

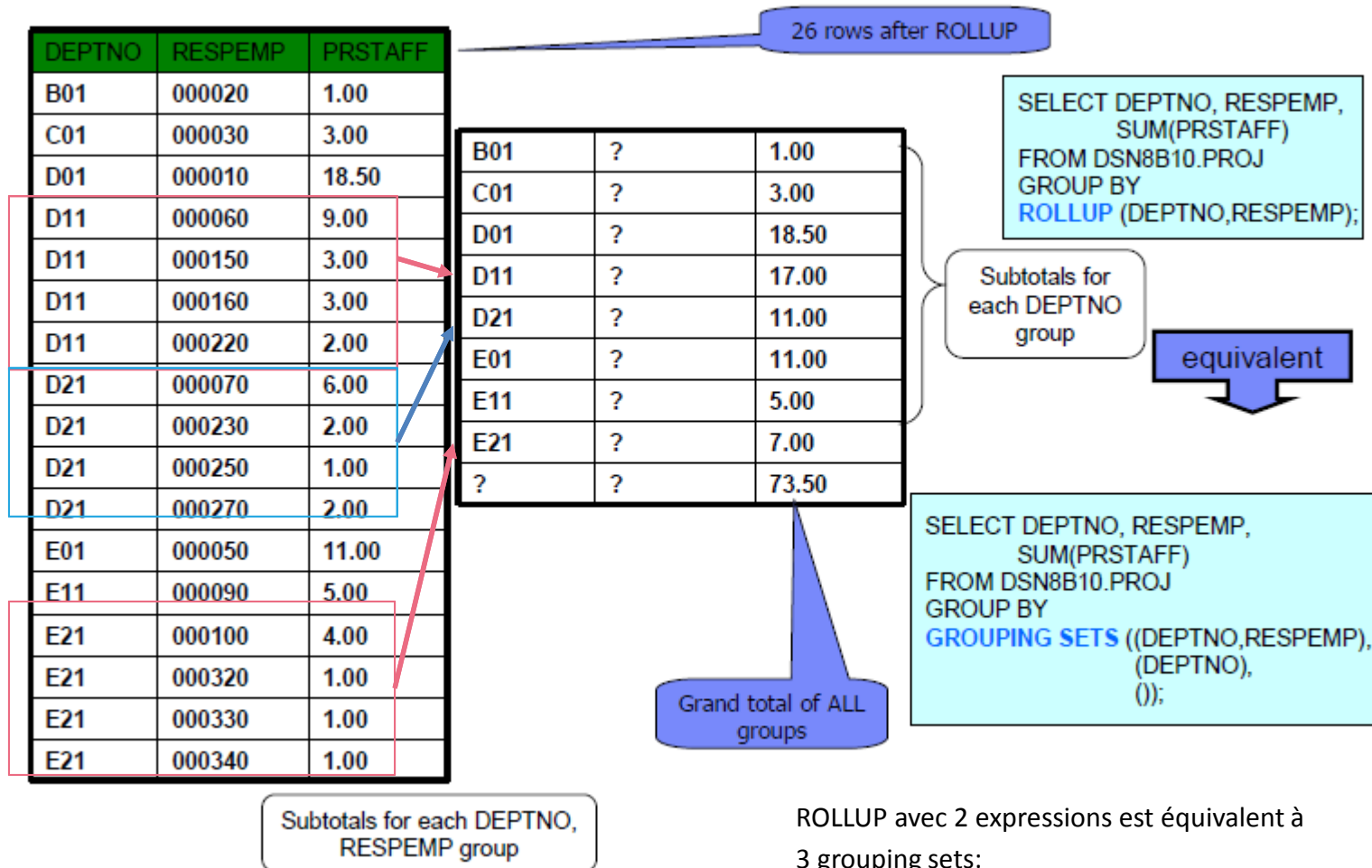
```
SELECT DEPTNO, RESPEMP,
       SUM(PRSTAFF)
FROM DSN8B10.PROJ
GROUP BY
GROUPING SETS ((DEPTNO,RESPEMP));
```

GROUP BY deptno, respemp est équivalent à 1 grouping set: (deptno,respemp)





DB2 V11: ROLLUP on DSN8B10.PROJ



ROLLUP avec 2 expressions est équivalent à

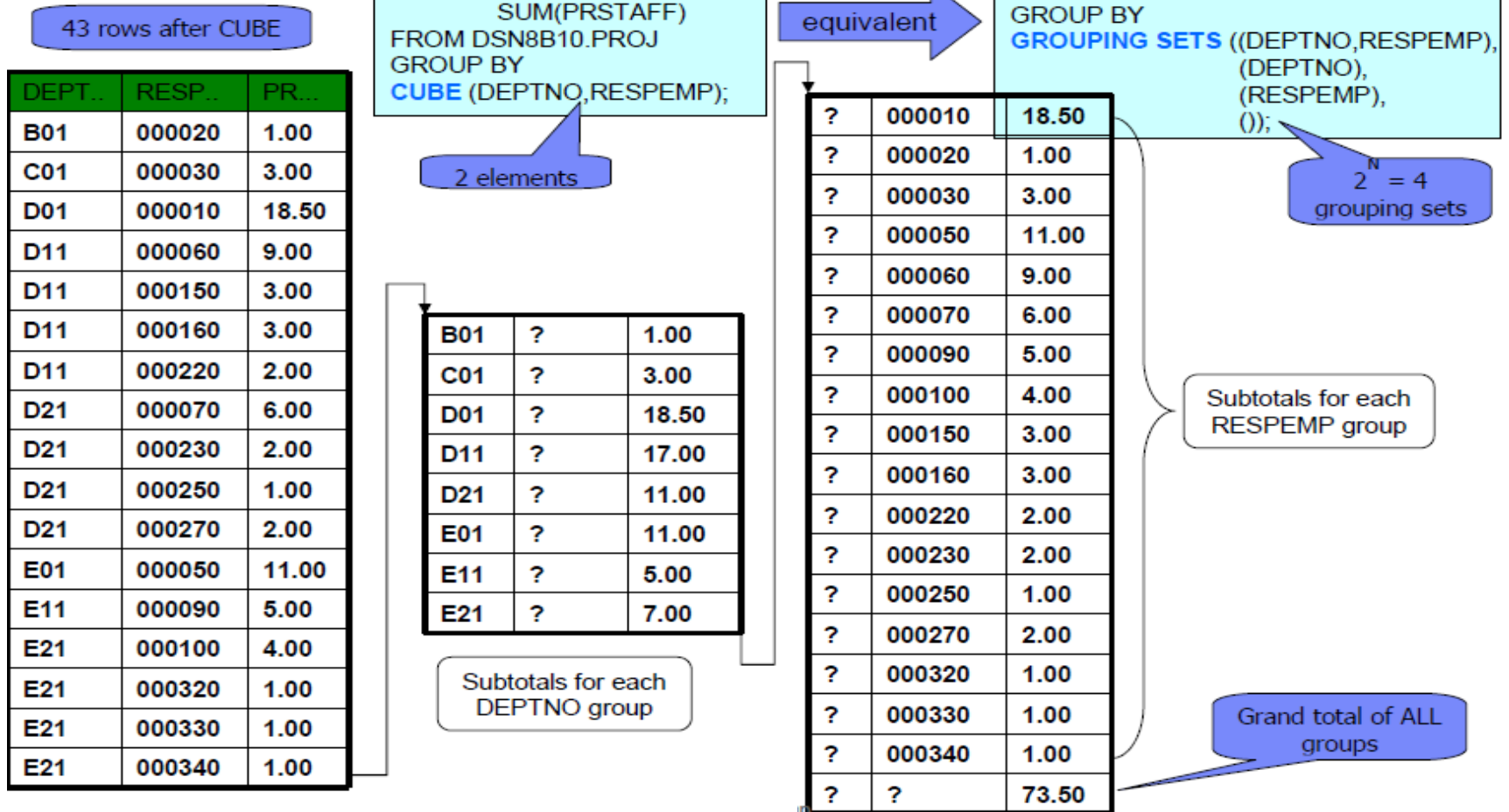
3 grouping sets:

- 1) (deptno,respemp) → 17 lignes
- 2) (deptno) → 8 lignes
- 3) () ← grand-total → 1 ligne





DB2 V11: CUBE on DSN8B10.PROJ



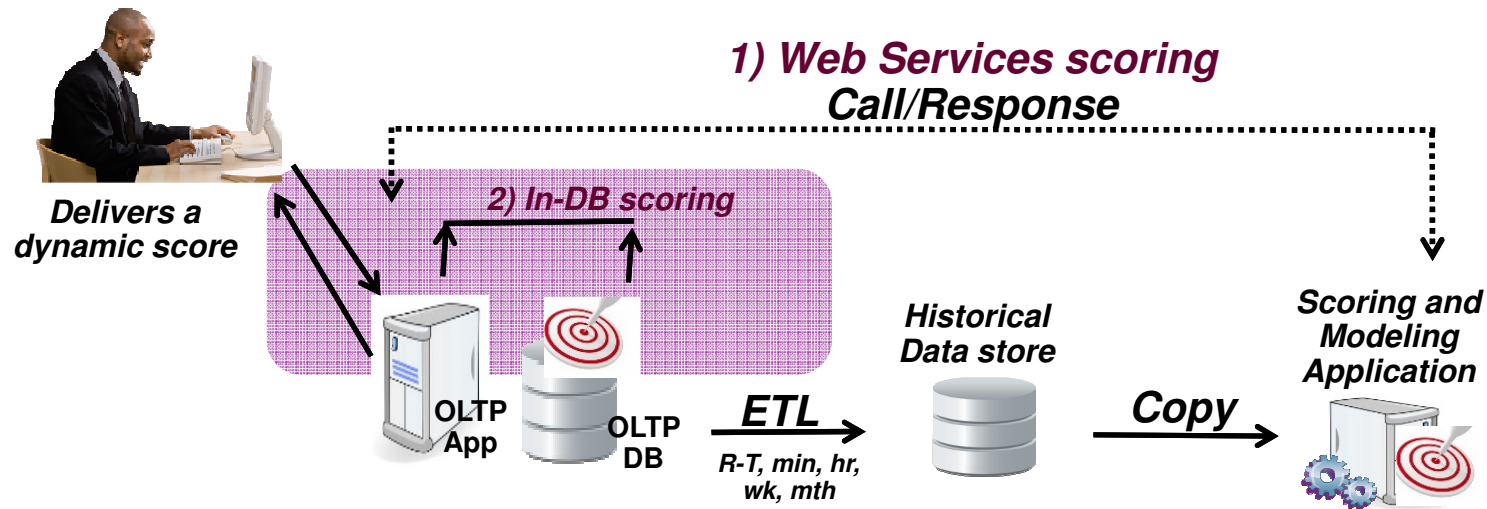
CUBE avec 2 expressions est équivalent à 4 grouping sets:

- 1) (deptno, respemp) → 17 lignes
- 2) (deptno) → 8 lignes
- 3) (respemp) → 17 lignes
- 4) () ← grand-total → 1 ligne





Détection des fraudes - DB2 V10: SPSS Scoring Adapter



- Analyse prédictive temps réel
 - Nécessaire pour détecter des tentatives de fraude
- Création du modèle dans SPSS Modeler
- Publication du modèle dans DB2 z/OS
- Utilisation de IBM SPSS Modeler 15 Scoring Adapter for DB2 z/OS
 - L'application peut appeler le modèle de scoring pour obtenir rapidement un taux (scoring)
 - Scoring via appel d'UDF DB2
- Remarque: l'Adapter initialement inclus dans DB2 Accessories Suite V2 fait maintenant partie de "SPSS Modeler with Scoring Adapter for zEnterprise V15"

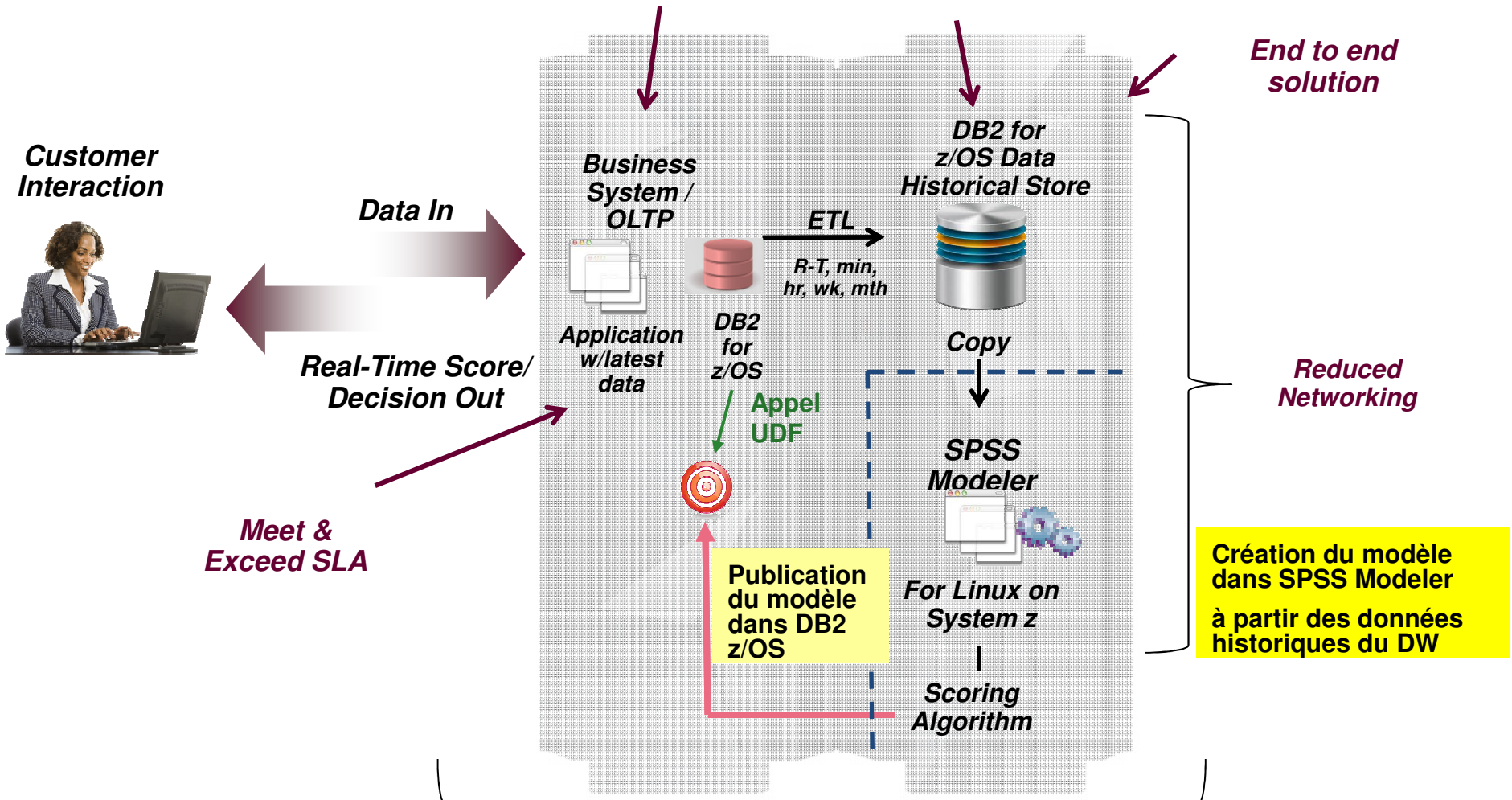




SPSS Modeler 15 Real-time Scoring with DB2 for z/OS V10



Support for both in-transaction and in-database scoring on the same platform



The next step in big data starts with IBM

Consolidates Resources





Merci !

The next step in big data starts with IBM.

