SQL for the 21st Century

David Simpson - dsimpson@themisinc.com

Themis Training – www.themisinc.com

@ThemisDavid

@ThemisTraining

Themis inc.

Agenda

- Online Analytical Processing (OLAP) Functions
- Moving Aggregates
- Advanced Aggregation
- MERGE and SELECT from FINAL (or OLD) TABLE
- Common Table Expressions and Recursion

Themis inc.

Online Analytical Processing (OLAP) Functions (DB2 9)

Rank Function

```
SELECT LASTNAME, SALARY,
   RANK() OVER (ORDER BY SALARY DESC) AS R
FROM EMP
ORDER BY SALARY DESC
```

LASTNAME	SALARY	R
HAAS	52750.00	1
MOON	52750.00	1
LUCCHESI	46500.00	3
THOMPSON	41250.00	4
GEYER	40175.00	5
KWAN	38250.00	6
PULASKI	36170.00	7
…	…	…

Rank Function

```
SELECT LASTNAME, SALARY,
  RANK() OVER (ORDER BY SALARY DESC) AS R
FROM EMP
ORDER BY LASTNAME
```

LASTNAME	SALARY	R
ADAMSON	25280.00	19
BROWN	27740.00	14
GEYER	40175.00	5
GOUNOT	23840.00	21
HAAS	52750.00	1
HENDERSON	29750.00	10
JEFFERSON	22180.00	24
…	…	…

Rank Function with Partitioning

```
SELECT DEPTNO, LASTNAME, SALARY,
       RANK() OVER(PARTITION BY DEPTNO
                   ORDER BY SALARY DESC) AS R
FROM EMP
ORDER BY DEPTNO, SALARY DESC
```

DEPTNO	LASTNAME	SALARY	R
A00	HAAS	52750.00	1
A00	MOON	52750.00	1
A00	LUCCHESI	46500.00	3
A00	O'CONNEL	29250.00	4
B01	THOMPSON	41250.00	1
C01	KWAN	38250.00	1
C01	NICHOLS	28420.00	2
C01	QUINTANA	23800.00	3
	…	…	…

Dense Rank Function

```
SELECT LASTNAME, SALARY,
   DENSE_RANK() OVER (ORDER BY SALARY DESC) AS R
FROM EMP
ORDER BY SALARY DESC
```

LASTNAME	SALARY	R
HAAS	52750.00	1
MOON	52750.00	1
LUCCHESI	46500.00	2
THOMPSON	41250.00	3
GEYER	40175.00	4
KWAN	38250.00	5
PULASKI	36170.00	6
…	…	…

Row Numbering

```
SELECT LASTNAME, SALARY,
   ROW_NUMBER() OVER (ORDER BY SALARY DESC) AS R
FROM EMP
ORDER BY SALARY DESC
```

LASTNAME	SALARY	R
HAAS	52750.00	1
MOON	52750.00	2
LUCCHESI	46500.00	3
THOMPSON	41250.00	4
GEYER	40175.00	5
KWAN	38250.00	6
PULASKI	36170.00	7
…	…	…

Moving Aggregates

Moving Sum / Avg

Example 1:

```
SELECT EMPNO, DEPTNO, SALARY,
   SUM (SALARY)
          OVER (ORDER BY SALARY ASC
                ROWS UNBOUNDED PRECEDING
                ) AS SUM_SAL
FROM EMP
```

Themis inc.

Moving Sum / Avg

EMPNO	DEPTNO	SALARY	SUM_SAL
000290	E11	15340.00	15340.00
000310	E11	15900.00	31240.00
000260	D21	17250.00	48490.00
000300	E11	17750.00	66240.00
000210	D11	18270.00	84510.00
000250	D21	19180.00	103690.00
000320	E21	19950.00	123640.00
000190	D11	20450.00	144090.00
000180	D11	21340.00	165430.00
000230	D21	22180.00	187610.00
…	…	…	…

Moving Sum / Avg

Example 2:

```
SELECT DEPTNO, EMPNO, SALARY,
   SUM  (SALARY)
            OVER (PARTITION BY DEPTNO
                  ORDER BY SALARY ASC
                  ROWS UNBOUNDED PRECEDING
                 ) AS SUM_SAL
FROM EMP
ORDER BY DEPTNO
```

Moving Sum / Avg

DEPTNO	EMPNO	SALARY	SUM_SAL
A00	000120	29250.00	29250.00
A00	000110	46500.00	75750.00
A00	000010	52750.00	128500.00
A00	000011	52750.00	181250.00
B01	000020	41250.00	41250.00
C01	000130	23800.00	23800.00
C01	000140	28420.00	52220.00
C01	000030	38250.00	90470.00
…	…	…	…

Moving Sum / Avg

Example 3: Rows n Preceding

```
SELECT DEPTNO, EMPNO, SALARY,
  AVG(SALARY)
  OVER (PARTITION BY DEPTNO
        ORDER BY SALARY ASC
        ROWS 2 PRECEDING
        ) AS AVG_SAL
FROM EMP
```

```
SELECT DEPTNO, EMPNO, SALARY,
  DEC(ROUND(AVG(SALARY)
  OVER (PARTITION BY DEPTNO
        ORDER BY SALARY ASC
        ROWS 2 PRECEDING
        ) ,2) ,7,2) AS AVG_SAL
FROM EMP
```

Themis inc.

OLAP Moving Ranges

ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING

ROWS BETWEEN 2 PRECEDING AND CURRENT ROW

ROWS BETWEEN 2 PRECEDING AND 1 PRECEDING

Themis inc.

OLAP Moving Ranges

```
RANGE BETWEEN 10000 PRECEDING
              AND 10000 FOLLOWING

RANGE BETWEEN 10000 PRECEDING
              AND CURRENT ROW
```

Themis inc.

Advanced Aggregation

The GROUP BY Clause

```
SELECT      DEPTNO,
            SUM(SALARY) AS PAYROLL
FROM        EMP
WHERE       DEPTNO LIKE 'D%'
GROUP BY    DEPTNO
```

DEPTNO	SALARY
D11	32250.00
D11	25280.00
D11	22250.00
D11	24680.00
D11	21340.00
D11	20450.00
D11	27740.00
D11	18270.00
D11	29840.00
D21	22180.00
D21	28760.00
D21	19180.00
D21	17250.00
D21	27380.00
D21	36170.00

222100.00

150920.00

DEPTNO	PAYROLL
D11	222100.00
D21	150920.00

GROUPing by Multiple Columns

```
SELECT      DEPTNO, JOB, AVG(SALARY) AS AVG
FROM        EMP
WHERE       DEPTNO < 'B99'
GROUP BY    DEPTNO, JOB
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
B01	MANAGER	41250.00

GROUPING SETS

```
SELECT      DEPTNO, JOB, AVG(SALARY) AS AVG
FROM        EMP
WHERE       DEPTNO < 'B99'
GROUP BY    GROUPING SETS
            ( (DEPTNO, JOB) )
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
B01	MANAGER	41250.00

GROUPING SETS With 2 Groups

```
SELECT      DEPTNO, JOB, AVG(SALARY) AS AVG
FROM        EMP
WHERE       DEPTNO < 'B99'
GROUP BY    GROUPING SETS
            ( (DEPTNO, JOB) , (DEPTNO) )
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
A00		45312.50
B01	MANAGER	41250.00
B01		41250.00

GROUPING SETS With 3 Groups

```
SELECT      DEPTNO, JOB, AVG(SALARY) AS AVG
FROM        EMP
WHERE       DEPTNO < 'B99'
GROUP BY    GROUPING SETS
            ( (DEPTNO, JOB) , (DEPTNO), ( ) )
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
A00		45312.50
B01	MANAGER	41250.00
B01		41250.00
		44500.00

ROLLUP

```
SELECT      DEPTNO, JOB, AVG(SALARY) AS AVG
FROM        EMP
WHERE       DEPTNO < 'B99'
GROUP BY    ROLLUP  (DEPTNO, JOB)
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
A00		45312.50
B01	MANAGER	41250.00
B01		41250.00
		44500.00

ROLLUP With 2 Columns

```
GROUP BY   ROLLUP  (DEPTNO, JOB)
```

=

```
GROUP BY   GROUPING SETS  (
                          (DEPTNO, JOB),
                          (DEPTNO),
                          ( )
                          )
```

ROLLUP With 3 Columns

```
GROUP BY   ROLLUP  (DEPTNO, JOB, EDLEVEL)
```

=

```
GROUP BY   GROUPING SETS  (
                          (DEPTNO, JOB, EDLEVEL),
                          (DEPTNO, JOB),
                          (DEPTNO)
                          ( )
                          )
```

CUBE

```
SELECT    DEPTNO, JOB, AVG(SALARY) AS AVG
FROM      EMP
WHERE     DEPTNO < 'B99'
GROUP BY  CUBE  (DEPTNO, JOB)
ORDER BY DEPTNO, JOB
```

DEPTNO	JOB	AVG
A00	CLERK	29250.00
A00	PRES	52750.00
A00	SALESREP	46500.00
A00		45312.50
B01	MANAGER	41250.00
B01		41250.00
	CLERK	29250.00
	PRES	52750.00
	SALESREP	46500.00
	MANAGER	41250.00
		44500.00

CUBE With 2 Columns

```
GROUP BY   CUBE (DEPTNO, JOB)
```

=

```
GROUP BY   GROUPING SETS  (
                          (DEPTNO, JOB),
                          (DEPTNO),
                          (JOB),
                          ( )
                          )
```

CUBE With 3 Columns

```
GROUP BY   CUBE (DEPTNO, JOB, EDLEVEL)
```

=

```
GROUP BY GROUPING SETS (  (DEPTNO, JOB, EDLEVEL),
                          (DEPTNO, JOB),
                          (DEPTNO, EDLEVEL),
                          (JOB, EDLEVEL),
                          (DEPTNO),
                          (JOB),
                          (EDLEVEL),
                          ( )       )
```

Themis inc.

MERGE and SELECT FROM FINAL (or OLD) TABLE

MERGE Statement

```
MERGE INTO ITEM I
   USING (VALUES (1, 'WIDGIT') )
      AS NEWITEM (ITEMNO, ITEMNAME)
  ON I.ITEMNO = NEWITEM.ITEMNO
WHEN MATCHED THEN
   UPDATE SET ITEMNAME = NEWITEM.ITEMNAME
WHEN NOT MATCHED THEN
   INSERT (ITEMNO,ITEMNAME)
     VALUES (NEWITEM.ITEMNO, NEWITEM.ITEMNAME)
```

MERGE Two Tables (LUW only)

```
MERGE INTO DEPT D
 USING (SELECT DEPTNO, DEPTNAME, MGRNO, LOCATION, ADMRDEPT
        FROM THEMIS81.DEPT) AS OD
 ON D.DEPTNO = OD.DEPTNO
 WHEN MATCHED THEN
   UPDATE SET D.DEPTNAME = OD.DEPTNAME,
              D.MGRNO = OD.MGRNO,
              D.LOCATION = OD.LOCATION,
              D.ADMRDEPT = OD.ADMRDEPT
 WHEN NOT MATCHED THEN
  INSERT (DEPTNO, DEPTNAME, MGRNO, LOCATION, ADMRDEPT)
  VALUES (OD.DEPTNO, OD.DEPTNAME, OD.MGRNO, OD.LOCATION,
                     OD.ADMRDEPT)
```

SELECT FROM

```
SELECT HIREDATE
FROM FINAL TABLE
(INSERT INTO EMP (EMPNO, LASTNAME, HIREDATE)
        VALUES ('999000', 'Caccavale', CURRENT DATE)
)
```

```
SELECT EMPNO
FROM OLD TABLE
(DELETE FROM EMP WHERE JOB = 'CLERK')
```

SELECT FROM

```
SELECT SALARY, SALARY * 1.1
  FROM OLD TABLE
 (UPDATE EMP SET SALARY = SALARY * 1.1
  WHERE DEPTNO = 'C01')
```

SELECT FROM MERGE (z/OS only)

```
SELECT ITEMNAME, UPD_IND FROM FINAL TABLE
(MERGE INTO ITEM I
  INCLUDE (UPD_IND CHAR(1))
  USING (VALUES (1, 'SOCKET') )
    AS NEWITEM (ITEMNO, ITEMNAME)
 ON I.ITEMNO = NEWITEM.ITEMNO
WHEN MATCHED THEN
  UPDATE SET ITEMNAME = NEWITEM.ITEMNAME,
           UPD_IND = 'U'
WHEN NOT MATCHED THEN
  INSERT (ITEMNO,ITEMNAME,UPD_IND)
    VALUES (NEWITEM.ITEMNO, NEWITEM.ITEMNAME,'I') )
```

Themis inc.

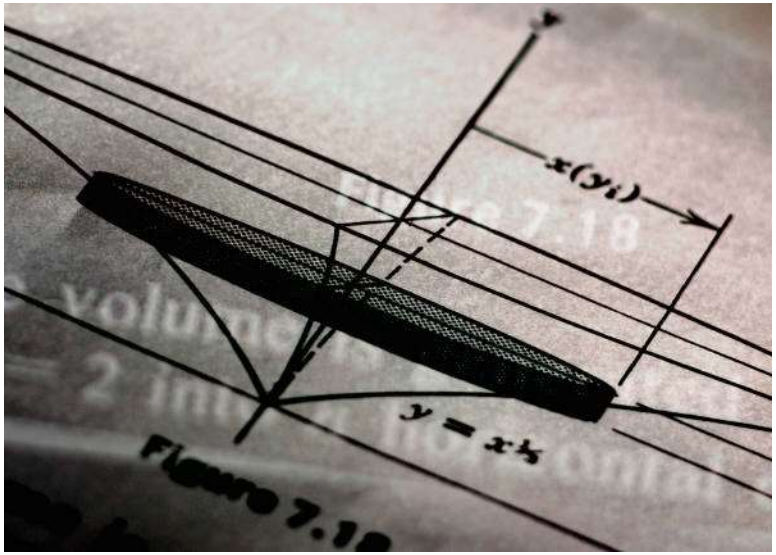
Common Table Expressions and Recursion

Common Table Expressions

```
WITH DEPTAVG AS
      (SELECT DEPTNO, AVG(SALARY) AS AVG_SAL
       FROM EMP
       GROUP BY DEPTNO)
  SELECT E.EMPNO, E.LASTNAME, E.SALARY,
         E.DEPTNO, DA.AVG_SAL
  FROM EMP E, DEPTAVG DA
  WHERE E.DEPTNO = DA.DEPTNO
```

Problem

Return the department number and the total payroll for the department that has the highest payroll. Payroll will be defined as the sum of all salaries and bonuses for the department.



Themis inc.

Solution #1: Use a View

```
CREATE VIEW DEPTPAY (DEPTNO, DEPT_TOT)
        AS SELECT DEPTNO, SUM(SALARY + BONUS)
                FROM EMP
              GROUP BY DEPTNO
```

```
SELECT DEPTNO, DEPT_TOT
        FROM DEPTPAY
WHERE DEPT_TOT =      (SELECT MAX(DEPT_TOT)
                              FROM DEPTPAY)
```

DEPTNO	DEPT_TOT
D11	226600.00

Themis inc.

Solution #2: Use Nested Table Expressions

```
SELECT DEPTNO, DEPT_TOT
FROM    (SELECT DEPTNO,
                SUM(SALARY + BONUS) AS DEPT_TOT
                FROM EMP
            GROUP BY DEPTNO) DEPTPAY
WHERE DEPT_TOT =
        (SELECT MAX(TOT2)
          FROM (SELECT DEPTNO,
                       SUM(SALARY + BONUS) TOT2
                       FROM EMP
                   GROUP BY DEPTNO) DEPTPAY2
        )
```

Themis inc.

Solution #3: Use a Common Table Expression

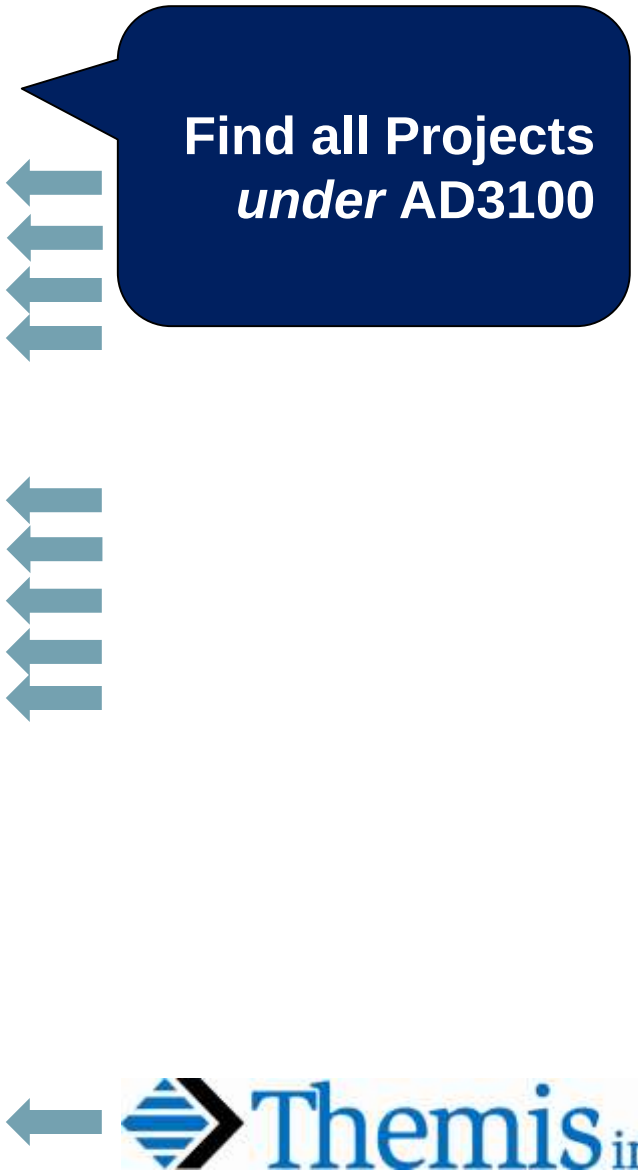
```
WITH DEPTPAY AS
        (SELECT DEPTNO,
                SUM(SALARY + BONUS) AS DEPT_TOT
                FROM EMP
          GROUP BY DEPTNO)

SELECT DEPTNO, DEPT_TOT
    FROM DEPTPAY
    WHERE DEPT_TOT = (SELECT MAX(DEPT_TOT)
                             FROM DEPTPAY)
```

Themis inc.

Recursive Data

PROJ Table

PROJNO	PROJNAME	MAJPROJ
AD3100	ADMIN SERVICES	
AD3110	GENERAL AD SYSTEMS	AD3100
AD3111	PAYROLL PROGRAMMING	AD3110
AD3112	PERSONNEL PROGRAMMG	AD3110
AD3113	ACCOUNT.PROGRAMMING	AD3110
IF1000	QUERY SERVICES	
IF2000	USER EDUCATION	
MA2100	WELD LINE AUTOMATION	AD3110
MA2110	W L PROGRAMMING	MA2100
MA2111	W L PROGRAM DESIGN	MA2110
MA2112	W L ROBOT DESIGN	MA2110
MA2113	W L PROD CONT PROGS	MA2110
OP1000	OPERATION SUPPORT	
OP1010	OPERATION	OP1000
OP2000	GEN SYSTEMS SERVICES	
OP2010	SYSTEMS SUPPORT	OP2000
OP2011	SCP SYSTEMS SUPPORT	OP2010
OP2012	APPLICATIONS SUPPORT	OP2010
OP2013	DB/DC SUPPORT	OP2010
PL2100	WELD LINE PLANNING	MA2100



Recursive SQL

```
WITH PROJECTS (PNO, PNAME)
AS (SELECT PROJNO, PROJNAME
      FROM PROJ
     WHERE PROJNO = 'AD3100'
     UNION ALL
     SELECT P.PROJNO, P.PROJNAME
    FROM PROJ P JOIN PROJECTS
      ON P.MAJPROJ = PROJECTS.PNO
   )
SELECT PNO, PNAME
FROM PROJECTS
ORDER BY PNO
```

Initial SELECT

Recursive SELECT

Final SELECT

Themis inc.

Avoiding Loops

```
WITH PROJECTS (PNO, PNAME, LVL)
AS (SELECT PROJNO, PROJNAME, 1 AS LVL
      FROM PROJ
     WHERE PROJNO = 'AD3100'
     UNION ALL
     SELECT P.PROJNO, P.PROJNAME,
                      PROJECTS.LVL + 1
    FROM PROJ P JOIN PROJECTS
      ON P.MAJPROJ = PROJECTS.PNO
    WHERE PROJECTS.LVL < 100
   )
SELECT PNO, PNAME, LVL
FROM PROJECTS
ORDER BY PNO
```

Themis inc.

David Simpson
Themis Training
dsimpson@themisinc.com

Themis inc.