IBM Information Management software

Column Functions and Grouping

Unit Objectives

After completing this unit, you should be able to:

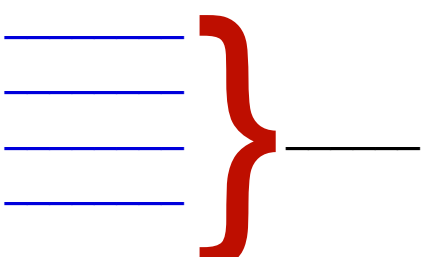
- Describe the difference between scalar and column functions
- List the more common DB2 column functions
- Group rows into sets based on one or more columns

SQL Functions

Function_name(f_argument)

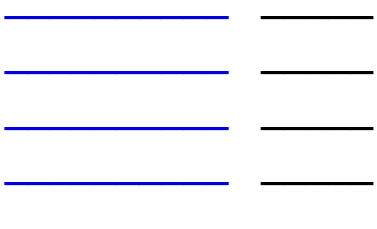
Column Functions

AVG(SALARY)



Scalar Functions

YEAR(HIREDATE)

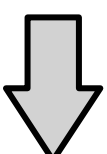


Column Functions

Compute Total	SUM(expression)
Calculate Average	AVG(expression)
Find Minimum Value	MIN(expression)
Find Maximum Value	MAX(expression)
Determine the number of rows meeting the search condition	COUNT(*) COUNT_BIG(*)
Determine the number of unique non-null values in a column	COUNT(DISTINCT col-name) COUNT_BIG(DISTINCT col-name)

Sample Column Functions

```
SELECT      SUM(SALARY) AS SUM,
            AVG(SALARY) AS AVG,
            MIN(SALARY ) AS MIN,
            MAX(SALARY) AS MAX,
            COUNT(*)       AS COUNT,
            COUNT (DISTINCT WORKDEPT) AS DEPT
FROM        EMPLOYEE
```



SUM	AVG	MIN	MAX	COUNT	DEPT
873715.00	27303.59375000	15340.00	52750.00	32	8

Beware of Nulls

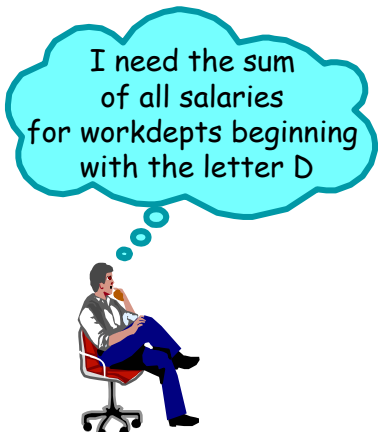
SUM(SALARY)+SUM(BONUS)+SUM(COMM)

VERSUS

SUM(SALARY+BONUS+COMM)

EMPNO	SALARY	BONUS	COMM	SALARY+BONUS+COMM
000010	1000	500	100	1600
000020	2000	NULL	300	NULL
000030	2500	400	NULL	NULL
000040	1500	100	400	2000
Sum:	7000	1000	800	--> 8800 3600

Column Functions Based on Subset

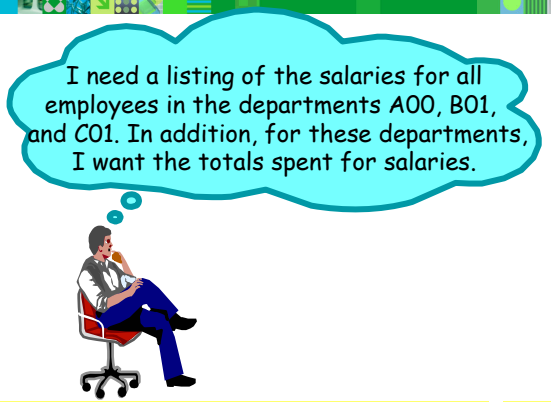



```
SELECT    SUM(SALARY) AS SUM
FROM      EMPLOYEE
WHERE     WORKDEPT LIKE 'D%'
```

SUM

373020.00

GROUP BY




```
SELECT      WORKDEPT, SALARY
FROM        EMPLOYEE
WHERE       WORKDEPT IN ('A00', 'B01', 'C01')
ORDER BY    WORKDEPT
```

WORKDEPT	SALARY
A00	52750.00
A00	46500.00
A00	29250.00
B01	41250.00
C01	38250.00
C01	23800.00
C01	28420.00

```
SELECT      WORKDEPT, SUM(SALARY) AS SUM
FROM        EMPLOYEE
WHERE       WORKDEPT IN ('A00', 'B01', 'C01')
GROUP BY    WORKDEPT
ORDER BY    WORKDEPT
```

WORKDEPT	SUM
A00	128500.00
B01	41250.00
C01	90470.00

GROUP BY more than one Column

Find out the average salary per education level for each department group (given by the first character of the department number), for education levels 18 and higher

```
SELECT SUBSTR(WORKDEPT,1,1) AS DEPT_GROUP,  EDLEVEL,
       DECIMAL(AVG(SALARY),9,2) AS AVGSAL
FROM EMPLOYEE
WHERE EDLEVEL >= 18
GROUP BY SUBSTR(WORKDEPT,1,1), EDLEVEL
```

DEPT_GROUP	EDLEVEL	AVGSAL
A	18	52750.00
A	19	46500.00
B	18	41250.00
C	18	28420.00
C	20	38250.00
D	18	29840.00

The Hardest to Remember Rule in All of SQL!

Only applies to queries with COLUMN FUNCTIONS in the SELECT Clause

If a SELECT clause has COLUMN functions
AND
columns not in COLUMN functions

ALL

columns not in COLUMN functions must be included in the GROUP BY clause

GROUP BY, ORDER BY

```
SELECT MAJPROJ, DEPTNO,
       AVG(PRSTAFF)  AS "AVG(PRSTAFF)",
       COUNT(*) AS "COUNT(*)"
FROM PROJECT
GROUP BY MAJPROJ, DEPTNO
ORDER BY MAJPROJ DESC
```

MAJPROJ	DEPTNO	AVG(PRSTAFF)	COUNT(*)
-	C01	1.50000000000000000000000000000	2
-	D01	9.25000000000000000000000000000	2
-	E01	5.50000000000000090000000000000	2
OP2010	E21	1.00000000000000000000000000000	3
OP2000	E21	4.00000000000000000000000000000	1
OP1000	E11	5.00000000000000000000000000000	1
MA2110	D11	2.66666666666666610000000000000	3
MA2100	B01	1.00000000000000000000000000000	1
MA2100	D11	8.99999999999999820000000000000	1
AD3110	D21	1.66666666666666630000000000000	3
AD3100	D21	6.00000000000000000000000000000	1

GROUP BY, HAVING (1 of 2)


```
SELECT    WORKDEPT, SUM(SALARY) AS SUM
FROM      EMPLOYEE
WHERE     WORKDEPT IN ('A00', 'B01', 'C01')
GROUP BY  WORKDEPT
ORDER BY  WORKDEPT
```

WORKDEPT	SUM
A00	128500.00
B01	41250.00
C01	90470.00

```
SELECT    WORKDEPT, SUM(SALARY) AS SUM
FROM      EMPLOYEE
WHERE     WORKDEPT IN ('A00', 'B01', 'C01')
GROUP BY  WORKDEPT
HAVING    SUM(SALARY)  > 50000
ORDER BY  WORKDEPT
```

WORKDEPT	SUM
A00	128500.00
C01	90470.00

GROUP BY, HAVING (2 of 2)

By department, I need a listing of jobs, excluding managers, designer, and field representative, with an average salary higher than $25,000.

```
SELECT      WORKDEPT, JOB, AVG(SALARY) AS AVG
FROM        EMPLOYEE
WHERE       JOB NOT IN ('MANAGER', 'DESIGNER', 'FIELDREP')
GROUP BY    WORKDEPT, JOB
HAVING      AVG(SALARY) > 25000
ORDER BY    WORKDEPT, JOB
```

WORKDEPT	JOB	AVG
A00	CLERK	29250.00000000
A00	PRES	52750.00000000
A00	SALESREP	46500.00000000
C01	ANALYST	26110.00000000

Examples with HAVING


```
SELECT    WORKDEPT,
          COUNT(* ) AS NUMB
FROM      EMPLOYEE
GROUP BY  WORKDEPT
ORDER BY  NUMB, WORKDEPT
```

WORKDEPT	NUMB
B01	1
E01	1
A00	3
C01	3
E21	4
E11	5
D21	6
D11	9

```
SELECT    WORKDEPT,
          COUNT(* ) AS NUMB
FROM      EMPLOYEE
GROUP BY  WORKDEPT
HAVING    COUNT(*) > 1
ORDER BY  NUMB, WORKDEPT
```

WORKDEPT	NUMB
A00	3
C01	3
E21	4
E11	5
D21	6
D11	9

Restrictions

- Column functions may be specified only in
  - SELECT
  - HAVING
- SELECT may specify only
  - Column functions
  - Columns specified in 'GROUP BY'
- HAVING may specify
  - *Any* column function on any column in a table being queried. This column need not be in the SELECT.
- Column functions may not be nested

SELECT Statement - Six Clauses

```
SELECT DEP, JOB, AVG(SAL)

FROM EMPL

WHERE JOB <> 'M'

GROUP BY DEP, JOB

HAVING AVG(SAL) > 28000

ORDER BY 3 DESC
```

Conceptual Execution of a SELECT (1 of 2)

BASE TABLE

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

WHERE

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

GROUP BY

JOB	SAL	DEP
C	27000	BLU
S	31000	BLU
S	29000	BLU
S	30000	BLU
C	27000	GRE
C	28000	GRE
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

Conceptual Execution of a SELECT (2 of 2)

HAVING

JOB	SAL	DEP
S	31000	BLU
S	29000	BLU
S	30000	BLU
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

SELECT

DEP	JOB	AVG(SAL)
BLU	S	30000
GRE	S	34000
RED	S	30500

ORDER BY

DEP	JOB	AVG(SAL)
GRE	S	34000
RED	S	30500
BLU	S	30000

Checkpoint

1. True or False? A scalar function produces a summary row for a set of rows.
2. A SELECT statement whose SELECT list includes a column function (SUM, AVG, MIN, MAX, COUNT, and so forth) and three columns not in column functions does not require a GROUP BY clause.
3. Which clause qualifies groups for further processing?
   a. SELECT
   b. FROM
   c. WHERE
   d. GROUP BY
   e. HAVING
   f. ORDER BY
4. True or False? The following query is syntactically correct.

```
SELECT WORKDEPT, AVG(SALARY)
FROM EMPLOYEE
WHERE AVG(SALARY) > 20000
GROUP BY WORKDEPT
HAVING COUNT(*) > 3
ORDER BY 2 DESC
```

Checkpoint Solutions

1. False. A column function produces a summary row for a set of rows.
2. False. A GROUP BY is required and all three columns not in column functions must be listed in the GROUP BY clause.
3. e
4. False. Column functions may not be used in a WHERE clause.

Unit Summary

Having completed this unit, you should be able to:

- Describe the difference between scalar and column functions
- List the more common DB2 column functions
- Group rows into sets based on one or more columns