Introduction to SQL

Group Function

By Kanokwan Atchariyachanvanich

Faculty of Information Technology

KMITL

Database System Concepts

Objectives

- After completing this lesson, you should be able to do the following:
  - Describe the use of group by functions
  - Group data by using the GROUP BY clause
  - Include or exclude grouped rows by using the HAVING clause

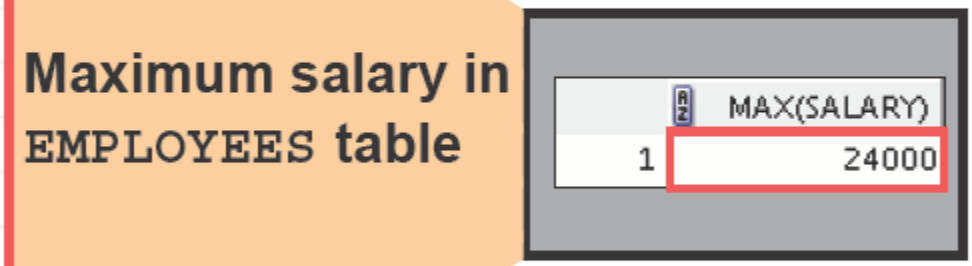
What Are Aggregate (Group By) Functions?

- Group functions operate on sets of rows to give one result per group.

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	110	12000
5	110	8300
6	90	24000
7	90	17000
8	90	17000
9	60	9000
10	60	6000
11	60	4200
12	50	5800
13	50	3500
14	50	3100
15	50	2600

...



Types of Aggregate (Group By) Functions

Function	Description
AVG([DISTINCT|ALL]*n*)	Average value of *n*, ignoring null values
COUNT({*|[DISTINCT|ALL]*expr* })	Number of rows, where *expr* evaluates to something other than null (count all selected rows using *, including duplicates and rows with nulls)
MAX([DISTINCT|ALL]*expr*)	Maximum value of *expr*, ignoring null values
MIN([DISTINCT|ALL]*expr*)	Minimum value of *expr*, ignoring null values
STDDEV([DISTINCT|ALL]*x*)	Standard deviation of *n*, ignoring null values
SUM([DISTINCT|ALL]*n*)	Sum values of *n*, ignoring null values
VARIANCE([DISTINCT|ALL]*x*)	Variance of *n*, ignoring null values

GROUP BY FUNCTIONS: SYNTAX

```
SELECT      [column,] aggregate_function(column), ...
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```

Note: All aggregate functions ignore null values. To substitute a value for null values, use the NVL, NVL2, or COALESCE functions. (Oracle)

USING THE AVG AND SUM FUNCTIONS

- You can use AVG and SUM for numeric data.
- ตัวอย่าง: แสดงค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด และผลรวม ของเงินเดือนของพนักงานตัวแทนขายทั้งหมด (sales representatives).

```
SELECT   AVG(salary), MAX(salary),
         MIN(salary), SUM(salary)
FROM     employees
WHERE    job_id LIKE '%REP%';
```

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8272.72727	11500	6000	273000

USING THE MIN AND MAX FUNCTIONS

- You can use MIN and MAX for numeric, character, and date data types.
- ตัวอย่าง: วันที่ที่พนักงานเข้าทำงานเริ่มแรกสุด และวันที่ที่พนักงานเข้าทำงานวันล่าสุด

```
SELECT   MIN(hire_date), MAX(hire_date)
FROM     employees;
```

MIN(HIRE_DATE)	MAX(HIRE_DATE)
17-JUN-87	21-APR-00

USING THE COUNT FUNCTION

1. COUNT(*) returns the number of rows in a table:
   - ตัวอย่าง: แสดงจำนวนพนักงานทั้งหมด

```
SELECT   COUNT(*)
FROM     employees
```

COUNT(*)
107

Showing 1 to 1 of 1 entries

NOTE:

1. COUNT(*) returns the number of rows in a table that satisfy the criteria of the SELECT statement, including duplicate rows and rows containing null values in any of the columns.
2. If a WHERE clause is included in the SELECT statement, COUNT(*) returns the number of rows that satisfy the condition in the WHERE clause.

USING THE COUNT FUNCTION

1. COUNT(*) returns the number of rows in a table:
   - ตัวอย่าง: แสดงจำนวนพนักงานทั้งหมดที่อยู่ในแผนกรหัสที่ 60

```
SELECT    COUNT(*)
FROM      employees
WHERE     department_id = 60;
```

COUNT(*)
5

NOTE:

1. COUNT(*) returns the number of rows in a table that satisfy the criteria of the SELECT statement, including duplicate rows and rows containing null values in any of the columns.
2. If a WHERE clause is included in the SELECT statement, COUNT(*) returns the number of rows that satisfy the condition in the WHERE clause.

USING THE COUNT FUNCTION

2. COUNT(column_name) returns the number of rows in a specific column in a table:

   - ตัวอย่าง: แสดงจำนวนพนักงานทั้งหมดที่กำหนดให้รับค่าคอมมิชชัน

```
SELECT   COUNT(commission_pct)
FROM     employees
```

COUNT(commission_pct)
35

Showing 1 to 1 of 1 entries

NOTE:

1. COUNT(*) returns the number of rows in a table that satisfy the criteria of the SELECT statement, including duplicate rows and rows containing null values in any of the columns.
2. If a WHERE clause is included in the SELECT statement, COUNT(*) returns the number of rows that satisfy the condition in the WHERE clause.

USING THE COUNT FUNCTION

2. COUNT(expr) returns the number of rows with non-null values for expr

   - ตัวอย่าง: แสดงจำนวนพนักงานทั้งหมดที่อยู่ในแผนกรหัสที่ 80 ที่รับค่าคอมมิชชั่น

```
SELECT    COUNT(commission_pct)
FROM      employees
WHERE     department_id = 80;
```

NOTE: COUNT(*expr*) returns the number of non-null values that are in the column identified by *expr*.

```
SELECT    COUNT(department_id)
FROM      employees ;
```

USING THE DISTINCT FUNCTION

3. COUNT(DISTINCT expr) returns the number of distinct non-null values of *expr*.
   - ตัวอย่าง: แสดงจำนวนแผนกที่ไม่ซ้ำกันในตาราง employees

```
SELECT   COUNT(DISTINCT department_id)
FROM     employees;
```

COUNT(DISTINCTDEPARTMENT_ID)
11

PRACTICE 1: AGGREGATE FUNCTION

1. จงเขียน SQL statement แสดงจำนวนพนักงานที่เริ่มทำงานในปี 1999

```
SELECT      count(employee_id)
FROM        employees
where       hire_date between '1999-01-01' and '1999-12-31'
```

Aggregate Functions and Null Values

1. Aggregate functions ignore null values in the column

   - Example: The average is calculated as the total commission that is paid to all employees divided by the number of employees receiving a commission (มี 35 คน).

```
SELECT   AVG(commission_pct)
FROM     employees;
```

AVG(COMMISSION_PCT)
.222857143

The average is calculated based on *only* those rows in the table where a valid value is stored in the COMMISSION_PCT column.

AGGREGATE FUNCTIONS AND NULL VALUES

2. The IFNULL function : IFNULL(column_name, replace_with) forces group functions to include null values

   - Example: The average is calculated as the total commission that is paid to all employees divided by the total number of employees in the company (มี 107 คน).

MySQL

```
SELECT    AVG(IFNULL(commission_pct, 0))
FROM      employees;
```

ORACLE

```
SELECT    AVG(NVL(commission_pct, 0))
FROM      employees;
```

AVG(NVL(COMMISSION_PCT,0))
.072897196

The average is calculated based on *all* rows in the table, regardless of whether null values are stored in the COMMISSION_PCT column.

PRACTICE 2: AGGREGATE FUNCTIONS

2.จงเขียน SQL statement แสดงค่าเฉลี่ยของเงินเดือนขั้นต่ำ (average_salary), ค่าสูงสุดของเงินเดือนขั้นต่ำ (max_salary) และค่าต่ำสุดของเงินเดือนขั้นต่ำ (low_salary) จากตารางานทั้งหมด พร้อมทั้งแสดงจำนวนรหัสงานที่มีอยู่ทั้งหมด (count_job)

average_salary	max_salary	low_salary	count_job
7240.0000	20000	2000	20

Showing 1 to 1 of 1 entries Previous 1 Next

```
SELECT   AVG(min_salary) average_salary ,
         MAX(min_salary) max_salary,
        MIN(min_salary)  low_salary,
         COUNT(job_id) count_job

FROM      JOBS ;
```

Creating Groups of Data

EMPLOYEES

	DEPARTMENT_ID	SALARY	
1	10	4400	4400
2	20	13000	9500
3	20	6000	
4	50	2500	
5	50	2600	
6	50	3100	3500
7	50	3500	
8	50	5800	
9	60	9000	
10	60	6000	6400
11	60	4200	
12	80	11000	
13	80	8600	10033
14	80	10500	
15	90	17000	
16	90	24000	19333
17	90	17000	
18	110	8300	10150
19	110	12000	
20	(null)	7000	7000

Average salary in the EMPLOYEES table for each department

	DEPARTMENT_ID	AVG(SALARY)
1	10	4400
2	20	9500
3	50	3500
4	60	6400
5	80	10033.333333333333...
6	90	19333.333333333333...
7	110	10150
8	(null)	7000

you need to divide the table of information into smaller groups.
You can do this by using the GROUP BY clause

Creating Groups of Data: GROUP BY Clause Syntax

- สามารถแบ่งชุดแถวข้อมูลเป็นกลุ่มย่อยๆ โดยใช้ GROUP BY clause.

```
SELECT      column, group_function(column)
FROM        table
 [WHERE     condition]
 [GROUP BY  group_by_expression]
 [ORDER BY  column];
```

In the syntax:

group_by_expression specifies columns whose values determine the basis for grouping rows

Guidelines

- You must include the *columns* in the GROUP BY clause.
- You cannot use a column alias in the GROUP BY clause.

USING THE GROUP BY CLAUSE

- GROUP BY column ไม่จำเป็นต้องอยู่ในรายการ SELECT
- ตัวอย่าง: แสดงรหัสแผนก และค่าเฉลี่ยเงินเดือนของแต่ละแผนก

```
SELECT      department_id, AVG(salary)
FROM        employees
GROUP BY    department_id ;
```

department_id	AVG(salary)
NULL	7000.000000
10	4400.000000
20	9500.000000
30	4150.000000
40	6500.000000
50	3475.555556
60	5760.000000
70	10000.000000
80	8955.882353
90	19333.333333
100	8600.000000
110	10150.000000

Showing 1 to 12 of 12 entries

USING THE ORDER BY CLAUSE

- ตัวอย่าง: แสดงรหัสแผนกและค่าเฉลี่ยเงินเดือนสำหรับแต่ละแผนก พร้อมทั้งจัดลำดับค่าเฉลี่ยเงินเดือนจากน้อยไปมาก

```
SELECT      department_id, AVG(salary)
FROM        employees
GROUP BY    department_id
ORDER BY    AVG(salary) ;
```

DEPARTMENT_ID	AVG(SALARY)
50	3475.55556
30	4150
10	4400
60	5760
40	6500
	7000
100	8600
80	8955.88235
20	9500
70	10000
110	10150
90	19333.3333

12 rows selected.

GROUPING BY MORE THAN ONE COLUMN

EMPLOYEES

	DEPARTMENT_ID	JOB_ID	SALARY
1	10	AD_ASST	4400
2	20	MK_MAN	13000
3	20	MK_REP	6000
4	50	ST_CLERK	2500
5	50	ST_CLERK	2600
6	50	ST_CLERK	3100
7	50	ST_CLERK	3500
8	50	ST_MAN	5800
9	60	IT_PROG	9000
10	60	IT_PROG	6000
11	60	IT_PROG	4200
12	80	SA_REP	11000
13	80	SA_REP	8600
14	80	SA_MAN	10500
15	90	AD_VP	17000
16	90	AD_PRES	24000
17	90	AD_VP	17000
18	110	AC_ACCOUNT	8300
19	110	AC_MGR	12000
20	(null)	SA_REP	7000

Add the salaries in the EMPLOYEES table for each job, grouped by department

	DEPARTMENT_ID	JOB_ID	SUM(SALARY)
1	10	AD_ASST	4400
2	20	MK_MAN	13000
3	20	MK_REP	6000
4	50	ST_CLERK	11700
5	50	ST_MAN	5800
6	60	IT_PROG	19200
7	80	SA_MAN	10500
8	80	SA_REP	19600
9	90	AD_PRES	24000
10	90	AD_VP	34000
11	110	AC_ACCOUNT	8300
12	110	AC_MGR	12000
13	(null)	SA_REP	7000

Sometimes you need to see results for groups within groups. The slide shows a report that displays the total salary that is paid to each job title in each department.

USING THE GROUP BY CLAUSE ON MULTIPLE COLUMNS

- return summary results for groups and subgroups by listing more than one GROUP BY column

```
SELECT      department_id dept_id, job_id, SUM(salary)
FROM        employees
GROUP BY    department_id, job_id ;
```

DEPT_ID	JOB_ID	SUM(SALARY)
	SA_REP	7000
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_MAN	11000
30	PU_CLERK	13900
40	HR_REP	6500
50	ST_MAN	36400
50	SH_CLERK	64300
50	ST_CLERK	55700
60	IT_PROG	28800
70	PR_REP	10000
80	SA_MAN	61000
80	SA_REP	243500
90	AD_VP	34000
90	AD_PRES	24000
100	FI_MGR	12000
100	FI_ACCOUNT	39600
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

PRACTICE 3: GROUPS OF DATA

3. แสดงรหัสแผนก รหัสงาน ผลรวมเงินเดือน (total_salary) ในรหัสงานเดียวกันที่อยู่ในแผนกเดียวกัน พร้อมทั้งแสดงผลลัพธ์เรียงด้วยรหัสแผนก จากน้อยไปมาก โดยในแผนกเดียวกันให้เรียงผลรวมเงินเดือนจากมากไปน้อย

department_id	job_id	total_salary
NULL	SA_REP	7000.00
10	AD_ASST	4400.00
20	MK_MAN	13000.00
20	MK_REP	6000.00
30	PU_CLERK	13900.00
30	PU_MAN	11000.00
40	HR_REP	6500.00
50	SH_CLERK	64300.00
50	ST_CLERK	53100.00
50	ST_MAN	36400.00

```
SELECT department_id,job_id, Sum(salary) total_salary
FROM   employees
GROUP BY  department_id, job_id
ORDER BY  department_id,  Sum(salary) DESC;
```

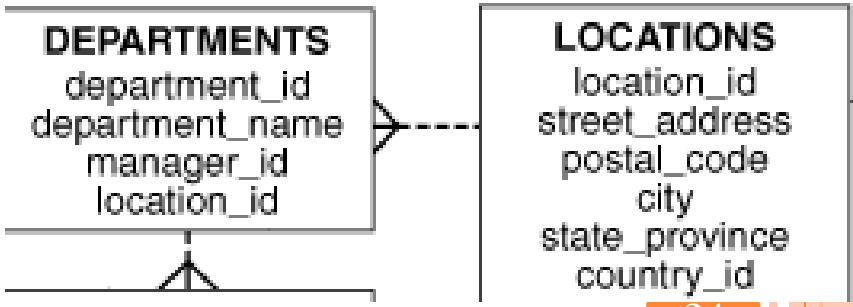
PRACTICE 4: GROUPS OF DATA

4. จงเขียน SQL statement แสดงชื่อเมือง (city) เฉพาะที่มีแผนก และจำนวนแผนกที่อยู่ในแต่ละเมือง เมือง ตั้งชื่อคอลัมน์คือ number_of_dep

city	number_of_dep
London	1
Munich	1
Oxford	1
Seattle	21
South San Francisco	1
Southlake	1
Toronto	1

Showing 1 to 7 of 7 entries Previous 1 Next

```
SELECT   City, count(department_id) number_of_dep
FROM     departments
JOIN     locations
USING    (location_id)
GROUP BY City ;
```



RESTRICTING GROUP RESULTS

EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	110	12000
5	110	8300
6	90	24000
7	90	17000
8	90	17000
9	60	9000
10	60	6000
11	60	4200
12	50	5800
13	50	3500
14	50	3100
15	50	2600

...

The maximum salary per department when it is greater than $10,000

	DEPARTMENT_ID	MAX(SALARY)
1	20	13000
2	80	11000
3	90	24000
4	110	12000

ใช้ HAVING clause เพื่อจำกัดกลุ่ม. To find the maximum salary in each of the departments that have a maximum salary greater than $10,000, you need to do the following:

1. หาค่าสูงสุดของเงินเดือนของแต่ละแผนกโดยการจัดกลุ่มด้วย department_id
2. จำกัดเฉพาะกลุ่มของแผนกที่มีเงินเดือนสูงสุดมากกว่า $10,000

Restricting Group Results with the HAVING Clause

- When you use the HAVING clause, the Oracle server restricts groups as follows:
  - จัดกลุ่มแถว Rows are grouped.
  - ใช้ group function
  - กลุ่มที่ตรงกับเงื่อนไขใน HAVING clause ถูกแสดงออกมา

```
SELECT      column, group_function
FROM        table
 [WHERE     condition]
 [GROUP BY  group_by_expression]
 [HAVING    group_condition]
 [ORDER BY  column];
```

USING THE HAVING CLAUSE

- จงแสดงรหัสแผนกและ*ค่าสูงสุด*ของเงินเดือนของแต่ละแผนกโดยการจัดกลุ่ม และจำกัดเฉพาะกลุ่มของแผนกที่มีเงินเดือนสูงสุดมากกว่า $10,000

```
SELECT      department_id, MAX(salary)
FROM        employees
GROUP BY    department_id
HAVING      MAX(salary)>10000 ;
```

DEPARTMENT_ID	MAX(SALARY)
20	13000
30	11000
80	14000
90	24000
100	12000
110	12000

- ใช้ GROUP BY clause โดยไม่จำเป็นต้องใช้ group function ใน SELECT
- ถ้าจำกัดแถวตามผลลัพธ์ของ Group function คุณต้องมี GROUP BY clause และ HAVING clause. ใน sql statement.

USING THE HAVING CLAUSE

- จงแสดงรหัสแผนกและ*ค่าเฉลี่ย*ของเงินเดือนของแต่ละแผนกโดยการจัดกลุ่ม และจำกัดเฉพาะกลุ่มของแผนกที่มีเงินเดือนสูงสุดมากกว่า $10,000

```
SELECT      department_id, AVG (salary)
FROM        employees
GROUP BY    department_id
HAVING      MAX(salary)>10000 ;
```

DEPARTMENT_ID	AVG(SALARY)
20	9500
30	4150
80	8955.88235
90	19333.3333
100	8600
110	10150

USING THE HAVING CLAUSE

- แสดงรหัสงานและผลรวมเงินเดือนของแต่ละรหัสงานที่มีผลรวมเกิน $13,000 โดยไม่รวมตัวแทนขาย และเรียงลำดับผลตามผลรวมเงินเดือนจากน้อยไปมาก

```
SELECT      job_id, SUM(salary) PAYROLL
FROM        employees
WHERE       job_id NOT LIKE '%REP%'
GROUP BY    job_id
HAVING      SUM(salary) > 13000
ORDER BY    SUM(salary);
```

JOB_ID	PAYROLL
PU_CLERK	13900
AD_PRES	24000
IT_PROG	28800
AD_VP	34000
ST_MAN	36400
FI_ACCOUNT	39600
ST_CLERK	55700
SA_MAN	61000
SH_CLERK	64300

9 rows selected.

NESTING GROUP FUNCTIONS

- Group functions can be nested to a depth of two.
- To display the maximum average salary. แสดงเงินเดือนเฉลี่ยที่สูงสุดจากทุกแผนก โดยคำนวณจากค่าเฉลี่ยเงินเดือนของพนักงานในแผนกเดียวกัน

ORACLE

```
SELECT      MAX(AVG(salary))
FROM        employees
GROUP BY    department_id;
```

MySQL

```
SELECT   MAX(avsal.sal)
FROM     (select avg(e.salary) sal
          from employees e
          group by department_id) avsal ;
```

Column alias

Table alias

MAX(AVG(SALARY))
19333.3333

PRACTICE 5: HAVING CLAUSE

5. จงเขียน SQL statement แสดงชื่อเมือง (city) เฉพาะที่มีแผนก และจำนวนแผนกที่มีมากกว่า 1 แผนกในแต่ละเมือง ตั้งชื่อคอลัมน์คือ number_of_dep

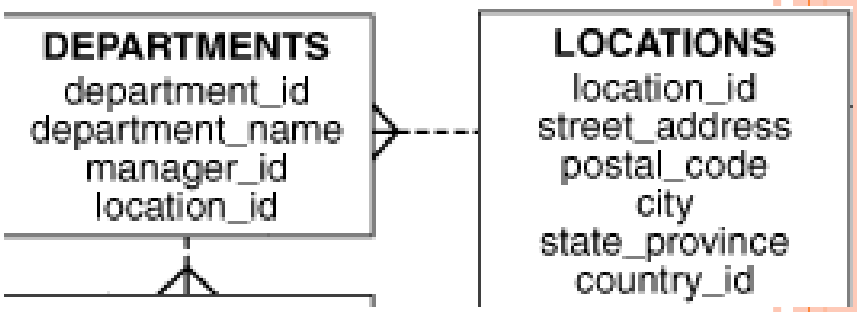
city	number_of_dep
Seattle	21

Showing 1 to 1 of 1 entries

```
SELECT    City, count(department_id) number_of_dep
FROM      departments
JOIN      locations
USING     (location_id)
GROUP BY  City
HAVING    Count(department_id) > 1;
```

หรือ

```
HAVING    number_of_dep > 1 ;
```



Summary

- In this lesson, you should have learned how to:
  - Use the group functions COUNT, MAX, MIN, and AVG
  - Write queries that use the GROUP BY clause and HAVING clause
- You can create subgroups by using the GROUP BY clause.
- Groups can be restricted using the HAVING clause.
- Place the HAVING and GROUP BY clauses after the WHERE clause in a statement.
- The order of the HAVING and GROUP clauses following the WHERE clause is not important.
- Place the ORDER BY clause last.
- The Oracle server evaluates the clauses in the following order:
  1. If the statement contains a WHERE clause, the server establishes the candidate rows.
  2. The server identifies the groups that are specified in the GROUP BY clause.
  3. The HAVING clause further restricts result groups that do not meet the group criteria in the HAVING clause.