CSE 311L(Database Management System)

LAB-Week 04 (Part A)

Instructor: Nazmul Alam Diptu

Aggregating Data Using Group Functions

Topics:

- Types of Group Functions
- Using the AVG and SUM Functions
- Using the MIN and MAX Functions
- Using the COUNT Function
- Using the GROUP BY Clause

Types of Group Functions

- **AVG**
- **COUNT**
- **MAX**
- **STDDEV**
- **MIN**
- **SUM**
- **VARIANCE**

Using the AVG and SUM Functions

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

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8150	11000	6000	32600

Using the MIN and MAX Functions

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

MIN(HIRE_	MAX(HIRE_
17-JUN-87	29-JAN-00

Using the COUNT Function

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

COUNT(DISTINCTDEPARTMENT_ID)
7

Using the GROUP BY Clause

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

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

Activity 01:

Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

Maximum	Minimum	Sum	Average
24000	2500	175500	8775

Activity 02:

Display the minimum, maximum, sum, and average salary for each job type

JOB_ID	Maximum	Minimum	Sum	Average
AC_ACCOUNT	8300	8300	8300	8300
AC_MGR	12000	12000	12000	12000
AD_ASST	4400	4400	4400	4400
AD_PRES	24000	24000	24000	24000
AD_VP	17000	17000	34000	17000
IT_PROG	9000	4200	19200	6400
MK_MAN	13000	13000	13000	13000
MK_REP	6000	6000	6000	6000
SA_MAN	10500	10500	10500	10500
SA_REP	11000	7000	26600	8867
ST_CLERK	3500	2500	11700	2925
ST_MAN	5800	5800	5800	5800

CSE 311L(Database Management System)

LAB-Week 04 (Part B)

Instructor: Nazmul Alam Diptu

Aggregating Data Using Group Functions

Topics:

- Using the GROUP BY Clause on Multiple Columns
- Illegal Queries Using Group Functions
- Excluding Group Results: The HAVING Clause
- Nesting Group Functions

Using the GROUP BY Clause on Multiple Columns

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

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

What is wrong with them?!!

```
>SELECT department_id, COUNT(last_name)
 FROM employees;

>SELECT department_id, AVG(salary)
 FROM employees
 WHERE AVG(salary) > 8000
 GROUP BY department_id;
```

Excluding Group Results: The HAVING Clause

```
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);
```

Nesting Group Functions

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

MAX(AVG(SALARY))
19333.3333

Activity 01:

Write a query to display the number of people with the same job.

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
IT_PROG	3
MK_MAN	1

Activity 02:

Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

MANAGER_ID	MIN(SALARY)
102	9000
205	8300
149	7000

Home Work:

Write a query to display each department's name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

Using the ROLLUP

```
select job_id, sum(salary) 'Total Salary'
from employees
group by job_id with rollup;

select dept_id, job_id, sum(salary) 'Total Salary'
from employees
group by dept_id, job_id with rollup;

select dept_id, job_id, sum(salary) 'Total Salary',
grouping(job_id)
from employees
group by dept_id, job_id with rollup;
```

Using GROUPING FUNCTION

```
select dept_id, job_id, sum(salary) 'Total Salary',
GROUPING (dept_id), GROUPING(job_id)
from employees
group by dept_id, job_id with rollup;

select dept_id,
      IF(GROUPING(job_id),'All Jobs',job_id) job_id,
      sum(salary) 'Total Salary'
from employees
group by dept_id, job_id with rollup;

select
      IF(GROUPING(dept_id), 'All Dept', dept_id),
      IF(GROUPING(job_id),'All Jobs',job_id) job_id,
    sum(salary) 'Total Salary'
from employees
group by dept_id, job_id with rollup;
```

CSE 311L(Database Management System)

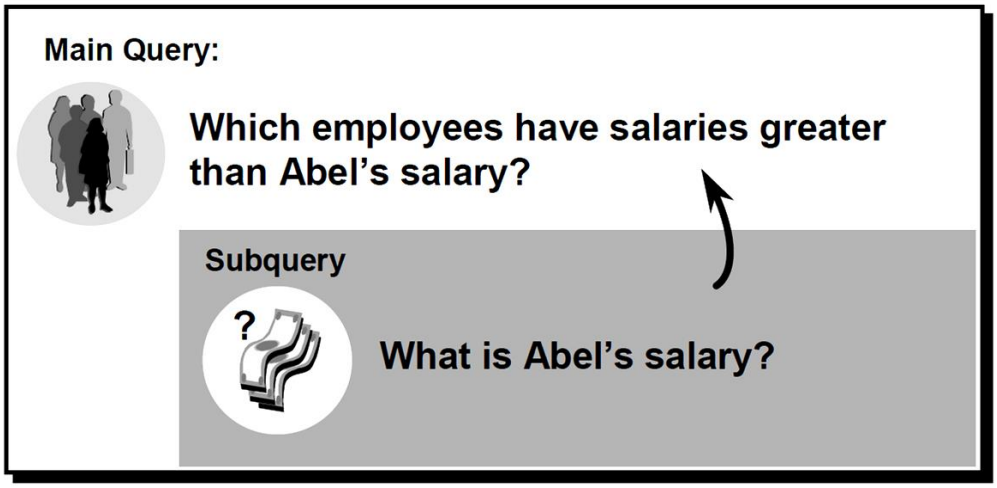
LAB-Week 04 (Part C) *Instructor: Nazmul Alam Diptu*

Subqueries

Topics:

- Using a Subquery to Solve a Problem
- Subquery Syntax
- Single-Row Subqueries
- Executing Single-Row Subqueries
- Using Group Functions in a Subquery

Who has a salary greater than Abel's?



Using a Subquery

```
SELECT last_name
FROM employees
WHERE salary >(SELECT salary
                FROM employees
                WHERE last_name = 'Abel');
```

LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

Single-Row Subqueries

- Return only one row
- Use single-row comparison operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

Executing Single-Row Subqueries

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =
            (SELECT job_id
             FROM employees
             WHERE employee_id = 141)
AND salary >
            (SELECT salary
             FROM employees
             WHERE employee_id = 143);
```

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100

Using Group Functions in a Subquery

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =
                  (SELECT MIN(salary)
                   FROM employees);
```

LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

Activity 01:

Write a query to display the last name and hire date of any employee in the same department as Zlotkey. Exclude Zlotkey.

LAST_NAME	HIRE_DATE
Abel	11-MAY-96
Taylor	24-MAR-98

Activity 02:

Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000
149	Zlotkey	10500
174	Abel	11000
205	Higgins	12000
201	Hartstein	13000
101	Kochhar	17000
102	De Haan	17000
100	King	24000

CSE 311L(Database Management System)

LAB-Week 04 (Part D)

Instructor: Nazmul Alam Diptu

Subqueries

Topics:

- Single-row operator with multiple-row subquery
- Multiple-Row Subqueries
- Using the ANY Operator
- Using the ALL Operator

What is Wrong with this Statement?

```
SELECT employee_id, last_name
FROM employees
WHERE salary =
                  (SELECT MIN(salary)
                   FROM employees
                   GROUP BY department_id);
```

Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison operators

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

Using the ANY Operator

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ANY
                         (SELECT salary
                          FROM employees
                          WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

Using the ALL Operator

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL
                        (SELECT salary
                         FROM employees
                         WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

Activity 01:

Display the last name and salary of every employee who reports to King.

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Mourgos	5800
Zlotkey	10500
Hartstein	13000

Activity 02:

Write a query to display the employee numbers, last names, and salaries of all employees who earn more than the average salary and who work in a department with any employee with a *u* in their name.

EMPLOYEE_ID	LAST_NAME	SALARY
103	Hunold	9000

Correlated Subquery

```
SELECT LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES AS e
WHERE SALARY > (
        SELECT FLOOR(AVG(SALARY))
        FROM EMPLOYEES
    WHERE JOB_ID = e. JOB_ID);
```

Subquery with `EXISTS`

select lname, dept_id, salary from employees

where exists

(select dept_id, avg(salary) name from employees e

join departments d on e.dept_id = d.id

group by dept_id having avg(salary) > 10000);

Finding Duplicates on Column Subset with Detail

```
WITH temp (id, fname, lname, hire_date, RowCnt)
as (
SELECT id, fname, lname, hire_date, SUM(1)
OVER (Partition By fname, lname, hire_date) as RowCnt
FROM employees
)
SELECT * from temp where RowCnt > 1
ORDER BY hire_date;
```