



UNIVERSITY OF CHITTAGONG

Department of Computer Science & Engineering

Program: **B.Sc. (Engineering)**

Session: 2022-2023

4th Semester

Lab_Task_02

Topic: SQL Practice Exercises

Course Title: DataBase Systems Lab

Course Code: CSE - 414

Submitted To:

Dr. Rudra Pratap Deb Nath

Associate Professor

Department of Computer Science & Engineering

University of Chittagong

Submitted By:

Nilanjana Das Jui

ID: 23701011

Dept. of Computer Science & Engineering

Date of submission: May 31, 2025

Contents

• Chapter - 5: Reporting Aggregated Data	
Using the Group Functions	2
– Practice - 5	2
• Chapter - 6: Displaying Data from Multiple Tables	7
– Practice - 6	7
• Chapter-7: Using Subqueries to Solve Queries	12
– Practice - 7	12

Chapter - 5

Reporting Aggregated Data Using the Group Functions

Practice - 5

1. Group functions work across many rows to produce one result per group.

Answer: **True.**

2. Group functions include nulls in calculations.

Answer: **False.**

3. The WHERE clause restricts rows before inclusion in a group calculation.

Answer: **True.**

The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Save your SQL statement as lab_05_04.sql. Run the query.

Download

Execution time: 0.001 seconds

	MAXIMUM	MINIMUM	SUM	AVERAGE
1	24000	2100	691416	6462

Answer:

```
SELECT round(MAX(Salary)) "Maximum",  
       round(MIN(Salary)) "Minimum",  
       round(SUM(Salary)) "Sum",  
       round(AVG(Salary)) "Average"  
FROM HR.EMPLOYEES;
```

5. Modify the query in lab_05_04.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab_05_04.sql as lab_05_05.sql. Run the statement in lab_05_05.sql.

	JOB_ID	MAXIMUM	MINIMUM	SUM	AVERAGE
1	AD_PRES	24000	24000	24000	24000
2	AD_VP	17000	17000	34000	17000
3	IT_PROG	9000	4200	28800	5760
4	FI_MGR	12008	12008	12008	12008
5	FI_ACCOUNT	9000	6900	39600	7920
6	PU_MAN	11000	11000	11000	11000
7	PU_CLERK	3100	2500	13900	2780
8	ST_MAN	8200	5800	36400	7280
9	ST_CLERK	3600	2100	55700	2785
10	SA_MAN	14000	10500	61000	12200
11	SA_REP	11500	6100	250500	8350
12	SH_CLERK	4200	2500	64300	3215

Answer:

```
SELECT round(MAX(Salary)) "Maximum",
       round(MIN(Salary)) "Minimum",
       round(SUM(Salary)) "Sum",
       round(AVG(Salary)) "Average"
FROM HR.EMPLOYEES
GROUP BY job_id;
```

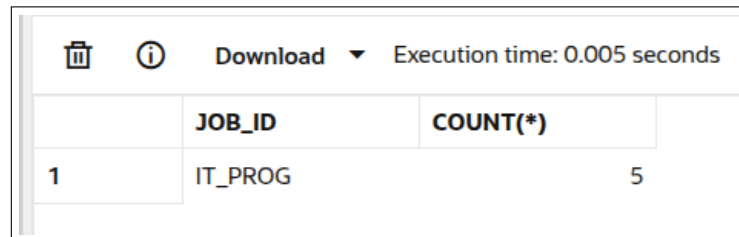
6. Write a query to display the number of people with the same job.

	JOB_ID	COUNT(*)
1	AC_ACCOUNT	1
2	AC_MGR	1
3	AD_ASST	1
4	AD_PRES	1
5	AD_VP	2
6	FI_ACCOUNT	5
7	FI_MGR	1
8	HR_REP	1
9	IT_PROG	5
10	MK_MAN	1
11	MK_REP	1
12	PR_REP	1

Answer:

```
SELECT job_id, count(*)  
FROM HR.EMPLOYEES  
GROUP BY job_id;
```

Generalize the query so that the user in the HR department is prompted for a job title. Save the script to a file named lab.05_06.sql. Run the query. Enter IT_PROG when prompted..

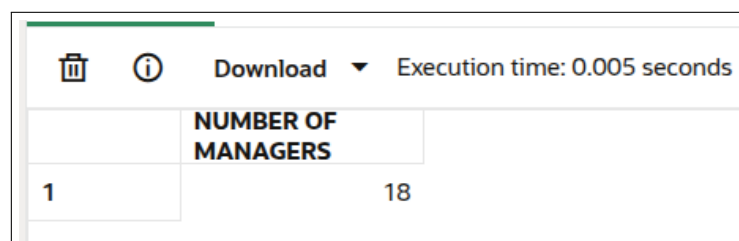


Download ▾ Execution time: 0.005 seconds		
	JOB_ID	COUNT(*)
1	IT_PROG	5

Answer:

```
SELECT job_id, count(*)  
FROM HR.EMPLOYEES  
where job_id = UPPER('&job_id')  
GROUP BY job_id;
```

7. Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.



Download ▾ Execution time: 0.005 seconds		
	NUMBER OF MANAGERS	
1	18	

Answer:

```
SELECT count(distinct manager_id) "Number of Managers"  
FROM HR.EMPLOYEES;
```

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

Download ▾ Execution time: 0.004 seconds	
	DIFFERENCE
1	21900

Answer:

```
SELECT MAX(salary) - MIN(salary) as DIFFERENCE
FROM HR.EMPLOYEES;
```

9. Create a report to 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.

Download ▾ Execution time: 0.005 seconds		
	MANAGER_ID	MIN(SALARY)
1	102	9000
2	205	8300
3	145	7000
4	146	7000
5	108	6900
6	147	6200
7	149	6200
8	148	6100
9	201	6000

Answer:

```
SELECT manager_id, MIN(Salary)
FROM HR.EMPLOYEES
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(Salary) >= 6000
ORDER BY MIN(Salary) DESC;
```

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

Download Execution time: 0.006 seconds					
	TOTAL	1995	1996	1997	1998
1	107	0	0	0	0

Answer:

```
SELECT COUNT (*)                                total,
       COUNT (decode (to_char (hire_date, 'YYYY'), 1995, '1')) "1995",
       COUNT (decode (to_char (hire_date, 'YYYY'), 1996, '1')) "1996",
       COUNT (decode (to_char (hire_date, 'YYYY'), 1997, '1')) "1997",
       COUNT (decode (to_char (hire_date, 'YYYY'), 1998, '1')) "1998"
FROM HR.EMPLOYEES;
```

11. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

Download Execution time: 0.006 seconds						
	JOB	DEPT 20	DEPT 50	DEPT 80	DEPT 90	TOTAL
1	AD_PRES	(null)	(null)	(null)	24000	24000
2	AD_VP	(null)	(null)	(null)	34000	34000
3	MK_MAN	13000	(null)	(null)	(null)	13000
4	MK_REP	6000	(null)	(null)	(null)	6000
5	SA_MAN	(null)	(null)	61000	(null)	61000
6	SA_REP	(null)	(null)	243500	(null)	243500
7	SH_CLERK	(null)	64300	(null)	(null)	64300
8	ST_CLERK	(null)	55700	(null)	(null)	55700
9	ST_MAN	(null)	36400	(null)	(null)	36400

Answer:

```
SELECT job_id                                "Job",
       SUM (decode (department_id, 20, salary)) "Dept 20",
       SUM (decode (department_id, 50, salary)) "Dept 50",
       SUM (decode (department_id, 80, salary)) "Dept 80",
       SUM (decode (department_id, 90, salary)) "Dept 90",
       SUM (salary)                           total
FROM HR.EMPLOYEES
WHERE department_id IN (20, 50, 80, 90)
GROUP BY job_id
ORDER BY job_id;
```

Chapter - 6

Displaying Data from Multiple Tables

Practice - 6

- Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	1000	1297 Via Cola di Rie	Roma	(null)	Italy
2	1100	93091 Calle della Tes	Venice	(null)	Italy
3	1200	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
4	1300	9450 Kamiya-cho	Hiroshima	(null)	Japan
5	1400	2014 Jabberwocky R	Southlake	Texas	United States of Ame
6	1500	2011 Interiors Blvd	South San Francisco	California	United States of Ame
7	1600	2007 Zagora St	South Brunswick	New Jersey	United States of Ame
8	1700	2004 Charade Rd	Seattle	Washington	United States of Ame

Answer:

```
SELECT location_id, street_address, city, state_province, country_name
FROM HR.LOCATIONS
NATURAL JOIN HR.COUNTRIES;
```

- The HR department needs a report of all employees. Write a query to display the last name, department number, and department name for all the employees.

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Martinez	20	Marketing
3	Davis	20	Marketing
4	Baida	30	Purchasing
5	Tobias	30	Purchasing
6	Li	30	Purchasing
7	Khoo	30	Purchasing
8	Himuro	30	Purchasing
9	Colmenares	30	Purchasing
10	Jacobs	40	Human Resources

Answer:

```
SELECT last_name, department_id, department_name
FROM HR.EMPLOYEES
JOIN HR.DEPARTMENTS using (department_id);
```

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

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Martinez	MK_MAN	20	Marketing
2	Davis	MK_REP	20	Marketing

Answer:

```
SELECT last_name, job_id, department_id, department_name
FROM HR.EMPLOYEES
JOIN HR.DEPARTMENTS using (department_id)
JOIN HR.LOCATIONS using (location_id)
WHERE city = 'Toronto';
```

4. Create a report to display employees' last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Save your SQL statement as lab_06_04.sql. Run the query.

	EMPLOYEE	EMP#	MANAGER	MGR#
1	Yang	101	King	100
2	Garcia	102	King	100
3	James	103	Garcia	102
4	Miller	104	James	103
5	Williams	105	James	103
6	Jackson	106	James	103
7	Nguyen	107	James	103
8	Gruenberg	108	Yang	101
9	Faviet	109	Gruenberg	108
10	Chen	110	Gruenberg	108

Answer:

```
SELECT e.last_name "Employee",
       e.employee_id "Emp#",
       m.last_name "Manager",
       m.employee_id "Mgr#"
FROM HR.EMPLOYEES e
JOIN HR.EMPLOYEES m ON (e.manager_id = m.employee_id)
ORDER BY e.employee_id;
```

5. Modify lab_06_04.sql to display all employees including King, who has no manager. Order the results by the employee number. Save your SQL statement as lab_06_05.sql. Run the query in lab_06_05.sql..

	EMPLOYEE	EMP#	MANAGER	MGR#
1	King	100	(null)	(null)
2	Yang	101	King	100
3	Garcia	102	King	100
4	James	103	Garcia	102
5	Miller	104	James	103
6	Williams	105	James	103
7	Jackson	106	James	103
8	Nguyen	107	James	103
9	Gruenberg	108	Yang	101
10	Faviet	109	Gruenberg	108

Answer:

```
SELECT e.last_name    "Employee",
       e.employee_id  "Emp#",
       m.last_name    "Manager",
       m.employee_id  "Mgr#"
FROM HR.EMPLOYEES e
LEFT JOIN HR.EMPLOYEES m ON (e.manager_id = m.employee_id)
ORDER BY e.employee_id;
```

6. Create a report for the HR department that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label. Save the script to a file named lab_06_06.sql.

	DEPARTMENT	EMPLOYEE	COLLEAGUE
1	20	Davis	Martinez
2	20	Martinez	Davis
3	30	Baida	Colmenares
4	30	Baida	Himuro
5	30	Baida	Khoo
6	30	Baida	Li
7	30	Baida	Tobias
8	30	Colmenares	Baida
9	30	Colmenares	Himuro
10	30	Colmenares	Khoo

Answer:

```
SELECT e.department_id department,
       e.last_name      employee,
       c.last_name      colleague
FROM HR.EMPLOYEES e
JOIN HR.EMPLOYEES c ON (e.department_id = c.department_id)
WHERE e.employee_id <> c.employee_id
ORDER BY department, employee, colleague;
```

7. The HR department needs a report on job grades and salaries. To familiarize yourself with the JOB_GRADES table, first show the structure of the JOB_GRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

Name	Null?	Type
JOB_ID	NOT NULL	VARCHAR2(10)
JOB_TITLE	NOT NULL	VARCHAR2(35)
MIN_SALARY		NUMBER(6)
MAX_SALARY		NUMBER(6)

Download ▾ Execution time: 0.002 seconds					
	LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE_LEVEL
1	Olson	ST_CLERK	Shipping	2100	A
2	Markle	ST_CLERK	Shipping	2200	A
3	Philtanker	ST_CLERK	Shipping	2200	A
4	Gee	ST_CLERK	Shipping	2400	A
5	Landry	ST_CLERK	Shipping	2400	A
6	Vargas	ST_CLERK	Shipping	2500	A
7	Perkins	SH_CLERK	Shipping	2500	A
8	Patel	ST_CLERK	Shipping	2500	A
9	Marlow	ST_CLERK	Shipping	2500	A
10	Sullivan	SH_CLERK	Shipping	2500	A

Answer:

```
DESC HR.JOBS;
SELECT e.last_name,
       e.job_id,
       d.department_name,
       e.salary,
       CASE
         WHEN e.salary BETWEEN 0 AND 3000 THEN 'A'
         WHEN e.salary BETWEEN 3001 AND 6000 THEN 'B'
         WHEN e.salary BETWEEN 6001 AND 9000 THEN 'C'
         WHEN e.salary BETWEEN 9001 AND 12000 THEN 'D' ELSE 'E'
       END AS grade_level
FROM HR.EMPLOYEES e
JOIN HR.DEPARTMENTS d ON e.department_id = d.department_id
ORDER BY e.salary;
```

8. The HR department wants to determine the names of all the employees who were hired after Davies. Create a query to display the name and hire date of any employee hired after employee Davies.

Download ▾ Execution time: 0.002 seconds		
	LAST_NAME	HIRE_DATE
1	Yang	21-SEP-15
2	James	03-JAN-16
3	Miller	21-MAY-17
4	Williams	25-JUN-15
5	Jackson	05-FEB-16
6	Nguyen	07-FEB-17
7	Chen	28-SEP-15
8	Sciarra	30-SEP-15

Answer:

```
SELECT e.last_name,
       to_char (e.hire_date, 'DD-MON-YY') hire_date
FROM HR.EMPLOYEES e
JOIN HR.EMPLOYEES davies ON (davies.last_name = 'Davies')
WHERE davies.hire_date < e.hire_date;
```

9. The HR department needs to find the names and hire dates of all the employees who were hired before their managers, along with their managers' names and hire dates. Save the script to a file named lab_06_09.sql.

Download ▾ Execution time: 0.009 seconds				
	LAST_NAME	HIRE_DATE_1	LAST_NAME_1	HIRE_DATE_2
1	Garcia	13-JAN-11	King	17-JUN-13
2	Jacobs	07-JUN-12	Yang	21-SEP-15
3	Higgins	07-JUN-12	Yang	21-SEP-15
4	Brown	07-JUN-12	Yang	21-SEP-15
5	Faviet	16-AUG-12	Gruenberg	17-AUG-12
6	Gruenberg	17-AUG-12	Yang	21-SEP-15
7	Li	07-DEC-12	King	17-JUN-13
8	Kauffling	01-MAY-13	King	17-JUN-13
9	Ladwig	14-JUL-13	Vollman	10-OCT-15
10	Whalen	17-SEP-13	Yang	21-SEP-15

Answer:

```
SELECT e.last_name AS last_name,
       TO_CHAR(e.hire_date, 'DD-MON-YY') AS hire_date_1,
       m.last_name AS last_name_1,
       TO_CHAR(m.hire_date, 'DD-MON-YY') AS hire_date_2
FROM HR.EMPLOYEES e
JOIN HR.EMPLOYEES m ON e.manager_id = m.employee_id
WHERE e.hire_date < m.hire_date
ORDER BY e.hire_date;
```

Chapter - 7

Using Subqueries to Solve Queries

Practice - 7

1. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

Download ▾ Execution time: 0.006 seconds			
	LAST_NAME	HIRE_DATE	
1	Singh	01-OCT-14	
2	Partners	05-JAN-15	
3	Errazuriz	10-MAR-15	
4	Cambault	15-OCT-17	
5	Tucker	30-JAN-15	

Answer:

```
SELECT last_name, to_char (hire_date, 'DD-MON-YY') hire_date
FROM HR.EMPLOYEES
WHERE department_id = ( SELECT department_id
                        FROM hr.employees
                        WHERE last_name = initcap ('&last_name'))
AND last_name <> initcap ('&last_name');
```

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

	EMPLOYEE_ID	LAST_NAME	SALARY
1	203	Jacobs	6500
2	123	Vollman	6500
3	165	Lee	6800
4	113	Popp	6900
5	155	Tuvault	7000
6	161	Sewall	7000
7	178	Grant	7000
8	164	Marvins	7200
9	172	Bates	7300
10	171	Smith	7400

Answer:

```
SELECT employee_id, last_name, salary
FROM HR.EMPLOYEES
WHERE salary > (SELECT AVG(salary)
                FROM HR.EMPLOYEES)
ORDER BY salary ASC;
```

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains the letter “u.” Save your SQL statement as lab_07.03.sql. Run your query. **Answer:**

	EMPLOYEE_ID	LAST_NAME
1	120	Weiss
2	121	Fripp
3	122	Kaufling
4	123	Vollman
5	124	Mourgos
6	125	Nayer
7	126	Mikkilineni
8	127	Landry
9	128	Markle
10	129	Bissot

```
SELECT employee_id, last_name
FROM HR.EMPLOYEES
WHERE department_id IN (SELECT department_id
                       FROM HR.EMPLOYEES
                       WHERE last_name LIKE '%u%');
```

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

Download ▾ Execution time: 0.006 seconds			
	LAST_NAME	DEPARTMENT_ID	JOB_ID
1	Whalen	10	AD_ASST
2	Himuro	30	PU_CLERK
3	Tobias	30	PU_CLERK
4	Baida	30	PU_CLERK
5	Li	30	PU_MAN
6	Colmenares	30	PU_CLERK
7	Khoo	30	PU_CLERK
8	Yang	90	AD_VP
9	Garcia	90	AD_VP
10	King	90	AD_PRES

Modify the query so that the user is prompted for a location ID. Save this to a file named lab_07.04.sql.

Answer:

```
SELECT last_name, department_id, job_id
FROM HR.EMPLOYEES
WHERE department_id IN (SELECT department_id
                        FROM HR.DEPARTMENTS
                        WHERE location_id = &location_id)
ORDER BY department_id;
```

5. Create a report for HR that displays the last name and salary of every employee who reports to King.

Download ▾ Execution time: 0.004 seconds		
	LAST_NAME	SALARY
1	Yang	17000
2	Garcia	17000
3	Li	11000
4	Weiss	8000
5	Fripp	8200

Answer:

```
SELECT last_name, salary
FROM HR.EMPLOYEES
WHERE manager_id IN (SELECT employee_id
                     FROM HR.EMPLOYEES
                     WHERE last_name = 'King');
```

6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

	DEPARTMENT_ID	LAST_NAME	JOB_ID
1	90	King	AD_PRES
2	90	Yang	AD_VP
3	90	Garcia	AD_VP

Answer:

```
SELECT department_id, last_name, job_id
FROM HR.EMPLOYEES
WHERE department_id IN (SELECT department_id
                        FROM HR.DEPARTMENTS
                        WHERE department_name = 'Executive');
```

7. Modify the query in lab_07_03.sql to display the employee number, last name, and salary of all employees who earn more than the average salary, and who work in a department with any employee whose last name contains a "u". Resave lab_07_03.sql as lab_07_07.sql. Run the statement in lab_07_07.sql.

	EMPLOYEE_ID	LAST_NAME	SALARY
1	120	Weiss	8000
2	121	Fripp	8200
3	122	Kaufling	7900
4	123	Vollman	6500
5	145	Singh	14000
6	146	Partners	13500
7	147	Errazuriz	12000

Answer:

```
SELECT employee_id, last_name, salary
FROM HR.EMPLOYEES
WHERE salary > (SELECT AVG(salary)
                FROM HR.EMPLOYEES)
AND department_id IN (SELECT department_id
                      FROM HR.EMPLOYEES
                      WHERE last_name LIKE '%u%');
```