LAB REPORT

COURSE TITLE	:	Database Systems Lab
COURSE CODE	:	CSE 208
LAB REPORT NO.	:	06
SUBMISSION DATE	:	03-09-2025

SUBMITTED TO

NAME	:	Farha Akhter Munmun
DEPT. OF	:	Computer Science and Engineering (CSE) Bangladesh University of Business & Technology (BUBT)

SUBMITTED BY

NAME	:	Shadman Shahriar
ID NO.	:	20245103408
INTAKE	:	53
SECTION	:	1
PROGRAM	:	B.Sc. Engg. in CSE

Database Systems Lab

Prerequisite

Create a database named **company**:

```
CREATE DATABASE company;
```

Create two tables: `employee` and `department`:

```
CREATE TABLE employee(
    emp_id int(6),
    emp_name varchar(12),
    job_name varchar(16),
    manager_id int(6),
    hire_date varchar(12),
    salary int(8),
    commission int(8),
    dep_id int(4)
);

CREATE TABLE department(
    dep_id int(4),
    dep_name varchar(16),
    dep_location varchar(24)
);
```

Inserting data into the tables:

```
-- Inserting data into `employee`
INSERT INTO employee VALUES(68319, "KAYLING", "PRESIDENT",
```

```
NULL, "1991-11-18", 600000, NULL, 1001);
INSERT INTO employee VALUES(66928, "BLAZE", "MANAGER",
68319, "1991-05-01", 275000, NULL, 3001);
INSERT INTO employee VALUES(67832, "CLARE", "MANAGER",
68319, "1991-06-09", 255000, NULL, 1001);
INSERT INTO employee VALUES(65646, "JONAS", "MANAGER",
68319, "1991-04-02", 295700, NULL, 2001);
INSERT INTO employee VALUES(67858, "SCARLET", "ANALYST",
65646, "1997-04-19", 310000, NULL, 2001);
INSERT INTO employee VALUES(69062, "FRANK", "ANALYST",
65646, "1991-12-03", 310000, NULL, 2001);
INSERT INTO employee VALUES(63679, "SANDRINE", "CLERK",
69062, "1990-12-18", 90000, NULL, 2001);
INSERT INTO employee VALUES(64989, "ADELYN", "SALESMAN",
66928, "1991-02-20", 170000, 40000, 3001);
INSERT INTO employee VALUES(65271, "WADE", "SALESMAN",
66928, "1991-02-22", 135000, 60000, 3001);

-- Inserting data into `department`
INSERT INTO department VALUES(1001, "FINANCE", "SYDNEY");
INSERT INTO department VALUES(2001, "AUDIT", "MELBOURNE");
INSERT INTO department VALUES(3001, "MARKETING", "PERTH");
INSERT INTO department VALUES(4001, "PRODUCTION",
"BRISBANE");
```

emp_id	emp_name	job_name	manager_id	hire_date	salary	commission	dep_id
68319	KAYLING	PRESIDENT	*NULL*	1991-11-18	600000	*NULL*	1001
66928	BLAZE	MANAGER	68319	1991-05-01	275000	*NULL*	3001
67832	CLARE	MANAGER	68319	1991-06-09	255000	*NULL*	1001
65646	JONAS	MANAGER	68319	1991-04-02	295700	*NULL*	2001
67858	SCARLET	ANALYST	65646	1997-04-19	310000	*NULL*	2001
69062	FRANK	ANALYST	65646	1991-12-03	310000	*NULL*	2001
63679	SANDRINE	CLERK	69062	1990-12-18	90000	*NULL*	2001
64989	ADELYN	SALESMAN	66928	1991-02-20	170000	40000	3001
65271	WADE	SALESMAN	66928	1991-02-22	135000	60000	3001

Figure - 1.1. Employee relation

dep_id	dep_name	dep_location
1001	FINANCE	SYDNEY
2001	AUDIT	MELBOURNE
3001	MARKETING	PERTH
4001	PRODUCTION	BRISBANE

Figure - 1.2. Department relation

Lab Tasks

1. Employee Details

From the following table, write a SQL query to find **employees** along with their **department details**.

Return `employee ID`, `employee name`, `job name`, `manager ID`, `hire date`, `salary`, `commission`, `department ID`, and `department name`.

```
SELECT * FROM employee NATURAL JOIN department;
```

dep_id	emp_id	emp_name	job_name	manager_id	hire_date	salary	commission	dep_name	dep_location
1001	68319	KAYLING	PRESIDENT	*NULL*	1991-11-18	600000	*NULL*	FINANCE	SYDNEY
3001	66928	BLAZE	MANAGER	68319	1991-05-01	275000	*NULL*	MARKETING	PERTH
1001	67832	CLARE	MANAGER	68319	1991-06-09	255000	*NULL*	FINANCE	SYDNEY
2001	65646	JONAS	MANAGER	68319	1991-04-02	295700	*NULL*	AUDIT	MELBOURNE
2001	67858	SCARLET	ANALYST	65646	1997-04-19	310000	*NULL*	AUDIT	MELBOURNE
2001	69062	FRANK	ANALYST	65646	1991-12-03	310000	*NULL*	AUDIT	MELBOURNE
2001	63679	SANDRINE	CLERK	69062	1990-12-18	90000	*NULL*	AUDIT	MELBOURNE
3001	64989	ADELYN	SALESMAN	66928	1991-02-20	170000	40000	MARKETING	PERTH
3001	65271	WADE	SALESMAN	66928	1991-02-22	135000	60000	MARKETING	PERTH

Figure - 1.3. Task 1

2. Well Paid Employees

From the following table, write a SQL query to identify those employees who earn **60000 or more** per year *or* **DO NOT work as ANALYST**.

```
SELECT * FROM employee
WHERE salary ⩾ 60000 OR job_name ≠ "ANALYST";
```

emp_id	emp_name	job_name	manager_id	hire_date	salary	commission	dep_id
68319	KAYLING	PRESIDENT	*NULL*	1991-11-18	600000	*NULL*	1001
66928	BLAZE	MANAGER	68319	1991-05-01	275000	*NULL*	3001
67832	CLARE	MANAGER	68319	1991-06-09	255000	*NULL*	1001
65646	JONAS	MANAGER	68319	1991-04-02	295700	*NULL*	2001
67858	SCARLET	ANALYST	65646	1997-04-19	310000	*NULL*	2001
69062	FRANK	ANALYST	65646	1991-12-03	310000	*NULL*	2001
63679	SANDRINE	CLERK	69062	1990-12-18	90000	*NULL*	2001
64989	ADELYN	SALESMAN	66928	1991-02-20	170000	40000	3001
65271	WADE	SALESMAN	66928	1991-02-22	135000	60000	3001

Figure - 1.4. Task 2

3. Third Highest Paid Employee

Find the **third highest salary** of the employee.

```
SELECT salary FROM employee
ORDER BY salary DESC LIMIT 2, 1
```

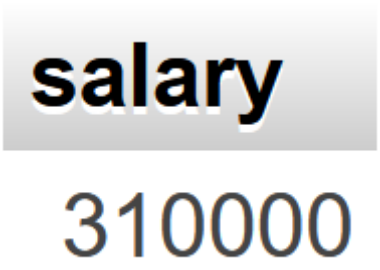

Figure - 1.5. Task 3