ALL LABS

By: Nasir Ali

Table of Contents

Lab 01: Foundation Statements of SQL

1. Write a query to display the names (first_name, last_name) using the alias name “First Name", "Last Name".

SELECT first_name as “First Name”, last_name as “Last Name” FROM employees;

First Name	Last Name
Steven	King
Neena	Kochhar
Lex	De Haan
Alexander	Hunold

2. Write a query to get a unique department ID from the employee table.

SELECT DISTINCT department_id FROM employees;

```
mysql> SELECT DISTINCT department_id FROM employees;
+---------------+
| department_id |
+---------------+
|          NULL |
|            10 |
|            20 |
|            30 |
|            40 |
|            50 |
|            60 |
|            70 |
|            80 |
|            90 |
|           100 |
|           110 |
+---------------+
```

3. Write a query to get all employee details from the employee table order by the first name, descending.

SELECT * FROM employees ORDER BY first_name DESC;

```
mysql> SELECT * FROM employees ORDER BY first_name DESC;
+-------------+------------+-----------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number       | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------------+------------+------------+----------+----------------+------------+---------------+
|         180 | Winston    | Taylor    | WTAYLOR  | 650.507.9876       | 1998-01-24 | SH_CLERK   |  3200.00 |           NULL |        120 |            50 |
|         171 | William    | Smith     | WSMITH   | 011.44.1343.629268 | 1999-02-23 | SA_REP     |  7400.00 |           0.15 |        148 |            80 |
|         206 | William    | Gietz     | WGIETZ   | 51hr5.123.8181     | 1994-06-07 | AC_ACCOUNT |  8300.00 |           NULL |        205 |           110 |
|         195 | Vance      | Jones     | VJONES   | 650.501.4876       | 1999-03-17 | SH_CLERK   |  2800.00 |           NULL |        123 |            50 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560       | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         141 | Trenna     | Rajs      | TRAJS    | 650.121.8009       | 1995-10-17 | ST_CLERK   |  3500.00 |           NULL |        124 |            50 |
|         132 | TJ         | Olson     | TJOLSON  | 650.124.8234       | 1999-04-10 | ST_CLERK   |  2100.00 |           NULL |        121 |            50 |
|         190 | Timothy    | Gates     | TGATES   | 650.505.3876       | 1998-07-11 | SH_CLERK   |  2900.00 |           NULL |        122 |            50 |
```

4. Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary.

SELECT employee_id, first_name, last_name, salary FROM employees ORDER BY salary asc;

```
mysql> SELECT employee_id, first_name, last_name, salary FROM employees ORDER BY salary ASC;
+-------------+-------------+-------------+----------+
| employee_id | first_name  | last_name   | salary   |
+-------------+-------------+-------------+----------+
|         132 | TJ          | Olson       |  2100.00 |
|         128 | Steven      | Markle      |  2200.00 |
|         136 | Hazel       | Philtanker  |  2200.00 |
|         127 | James       | Landry      |  2400.00 |
|         135 | Ki          | Gee         |  2400.00 |
|         119 | Karen       | Colmenares  |  2500.00 |
|         131 | James       | Marlow      |  2500.00 |
|         140 | Joshua      | Patel       |  2500.00 |
```

5. Write a query to get the total salaries payable to employee.

SELECT SUM(salary) as “Total Salaries Payable” FROM employees;

```
mysql> SELECT SUM(salary) as "Total Salaries Payable" FROM employees;
+--------------------------+
| Total Salaries Payable |
+--------------------------+
|              691400.00 |
+--------------------------+
```

6. Write a query to get the maximum and minimum salary from the employee's table.

SELECT MAX(salary), MIN(salary) FROM employees;

```
mysql> SELECT MAX(salary), MIN(salary) FROM employees;
+-------------+-------------+
| MAX(salary) | MIN(salary) |
+-------------+-------------+
|    24000.00 |     2100.00 |
+-------------+-------------+
```

7. Write a query to get the average salary and number of employees in the employees' table.

SELECT AVG(salary) as “Average Salary”, COUNT(*) as “Number of Employees” FROM employees;

```
mysql> SELECT AVG(salary) as "Average Salary", COUNT(*) as "Number of Employees" FROM employees;
+---------------+---------------------+
| Average Salary | Number of Employees |
+---------------+---------------------+
|    6461.682243 |                 107 |
+---------------+---------------------+
```

8. Write a query to get the number of jobs available in the employee's table.

SELECT COUNT(DISTINCT job_id) as “Number of Available Jobs” FROM employees;

```
mysql> SELECT COUNT(DISTINCT job_id) as "Number of Available Jobs" FROM employees;
+--------------------------+
| Number of Available Jobs |
+--------------------------+
|                       19 |
+--------------------------+
```

9. Write a query to get all first names from the employee's table in the upper case.

SELECT UPPER(first_name) as “First Name” FROM employees;

```
mysql> SELECT UPPER(first_name) as "First Name" FROM employees;
+-------------+
| First Name  |
+-------------+
| STEVEN      |
| NEENA       |
| LEX         |
| ALEXANDER   |
```

10. Write a query to select the first 10 records from a table.

SELECT * FROM employees LIMIT 0,10;

```
mysql> SELECT * FROM employees LIMIT 0,10;
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
|         100 | Steven     | King      | SKING    | 515.123.4567 | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            90 |
|         101 | Neena      | Kochhar   | NKOCHHAR | 515.123.4568 | 1989-09-21 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         102 | Lex        | De Haan   | LDEHAAN  | 515.123.4569 | 1993-01-13 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG    |  9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG    |  6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin    | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG    |  4200.00 |           NULL |        103 |            60 |
|         108 | Nancy      | Greenberg | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR     | 12000.00 |           NULL |        101 |           100 |
|         109 | Daniel     | Faviet    | DFAVIET  | 515.124.4169 | 1994-08-16 | FI_ACCOUNT |  9000.00 |           NULL |        108 |           100 |
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
```

11. Write a query to select the 3rd & 4th records of the employee's table.

SELECT * FROM employees LIMIT 3,2;

```
mysql> SELECT * FROM employees LIMIT 3,2;
+-------------+------------+-----------+---------+--------------+------------+---------+---------+----------------+------------+---------------+
| employee_id | first_name | last_name | email   | phone_number | hire_date  | job_id  | salary  | commission_pct | manager_id | department_id |
+-------------+------------+-----------+---------+--------------+------------+---------+---------+----------------+------------+---------------+
|         103 | Alexander  | Hunold    | AHUNOLD | 590.423.4567 | 1990-01-03 | IT_PROG | 9000.00 |           NULL |        102 |            60 |
|         104 | Bruce      | Ernst     | BERNST  | 590.423.4568 | 1991-05-21 | IT_PROG | 6000.00 |           NULL |        103 |            60 |
+-------------+------------+-----------+---------+--------------+------------+---------+---------+----------------+------------+---------------+
```

12. Write a query to select 2nd last record of the employee's table.

SELECT * FROM employees ORDER BY employee_id DESC LIMIT 1,1;

```
mysql> SELECT * FROM employees ORDER BY employee_id DESC LIMIT 1,1;
+-------------+------------+-----------+---------+--------------+------------+--------+---------+----------------+------------+---------------+
| employee_id | first_name | last_name | email   | phone_number | hire_date  | job_id | salary  | commission_pct | manager_id | department_id |
+-------------+------------+-----------+---------+--------------+------------+--------+---------+----------------+------------+---------------+
|         205 | Shelley    | Higgins   | SHIGGINS | 515.123.8080 | 1994-06-07 | AC_MGR | 12000.00 |           NULL |        101 |           110 |
+-------------+------------+-----------+---------+--------------+------------+--------+---------+----------------+------------+---------------+
```

Lab 02: Perform Arithmetic Operations & Querying Database Tables

1. Write a query to display EMPLOYEE_ID, FIRST_NAME, and SALARY of employees whose SALARY is less than $3000.

SELECT employee_id, first_name, salary FROM employees WHERE salary<3000;

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE salary<3000;
+-------------+------------+---------+
| employee_id | first_name | salary  |
+-------------+------------+---------+
|         116 | Shelli     | 2900.00 |
|         117 | Sigal      | 2800.00 |
|         118 | Guy        | 2600.00 |
|         119 | Karen      | 2500.00 |
|         126 | Irene      | 2700.00 |
|         127 | James      | 2400.00 |
|         128 | Steven     | 2200.00 |
|         130 | Mozhe      | 2800.00 |
|         131 | James      | 2500.00 |
|         132 | TJ         | 2100.00 |
|         134 | Michael    | 2900.00 |
|         135 | Ki         | 2400.00 |
|         136 | Hazel      | 2200.00 |
|         139 | John       | 2700.00 |
|         140 | Joshua     | 2500.00 |
|         143 | Randall    | 2600.00 |
|         144 | Peter      | 2500.00 |
|         182 | Martha     | 2500.00 |
|         183 | Girard     | 2800.00 |
|         190 | Timothy    | 2900.00 |
|         191 | Randall    | 2500.00 |
|         195 | Vance      | 2800.00 |
|         198 | Donald     | 2600.00 |
|         199 | Douglas    | 2600.00 |
+-------------+------------+---------+
```

2. Write a query to display FIRST_NAME, LASTNAME of all employees whose first name starts with letter 'A'.

SELECT * FROM employees WHERE first_name LIKE "a%";

```
mysql> SELECT first_name, last_name FROM employees WHERE first_name LIKE "A%";
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Alexander  | Hunold    |
| Alexander  | Khoo      |
| Adam       | Fripp     |
| Alberto    | Errazuriz |
| Allan      | McEwen    |
| Amit       | Banda     |
| Alyssa     | Hutton    |
| Alexis     | Bull      |
| Anthony    | Cabrio    |
| Alana      | Walsh     |
+------------+-----------+
```

3. Write a query to display FIRST_NAME, JOB_ID, DEPARTMENT_ID of employees who are either PU_CLERK or belongs to MANAGER_ID = 114.

SELECT first_name, job_id, department_id FROM employees WHERE job_id="PU_CLERK" OR manager_id=114;

```
mysql> SELECT first_name, job_id, department_id FROM employees WHERE job_id="PU_CLERK" OR manager_id=114;
+------------+----------+---------------+
| first_name | job_id   | department_id |
+------------+----------+---------------+
| Alexander  | PU_CLERK |            30 |
| Shelli     | PU_CLERK |            30 |
| Sigal      | PU_CLERK |            30 |
| Guy        | PU_CLERK |            30 |
| Karen      | PU_CLERK |            30 |
+------------+----------+---------------+
```

4. Write a query to display EMPLOYEE_ID, FIRST_NAME, and SALARY of employees whose salaries lies in the range of $1500 to $3000.

SELECT employee_id, first_name, salary FROM employees WHERE salary BETWEEN 1500 AND 3000;

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE salary BETWEEN 1500 AND 3000;
+-------------+------------+---------+
| employee_id | first_name | salary  |
+-------------+------------+---------+
|         116 | Shelli     | 2900.00 |
|         117 | Sigal      | 2800.00 |
|         118 | Guy        | 2600.00 |
|         119 | Karen      | 2500.00 |
|         126 | Irene      | 2700.00 |
|         127 | James      | 2400.00 |
|         128 | Steven     | 2200.00 |
|         130 | Mozhe      | 2800.00 |
|         131 | James      | 2500.00 |
|         132 | TJ         | 2100.00 |
|         134 | Michael    | 2900.00 |
|         135 | Ki         | 2400.00 |
|         136 | Hazel      | 2200.00 |
|         139 | John       | 2700.00 |
|         140 | Joshua     | 2500.00 |
|         143 | Randall    | 2600.00 |
|         144 | Peter      | 2500.00 |
|         182 | Martha     | 2500.00 |
|         183 | Girard     | 2800.00 |
|         187 | Anthony    | 3000.00 |
|         190 | Timothy    | 2900.00 |
|         191 | Randall    | 2500.00 |
|         195 | Vance      | 2800.00 |
|         197 | Kevin      | 3000.00 |
|         198 | Donald     | 2600.00 |
|         199 | Douglas    | 2600.00 |
+-------------+------------+---------+
```

5. Write a query to display EMPLOYEE_ID, FIRST_NAME, and SALARY of employees whose commission is empty.

SELECT employee_id, first_name, salary FROM employees WHERE commission_pct is NULL;

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE commission_pct is NULL;
+-------------+-------------+----------+
| employee_id | first_name  | salary   |
+-------------+-------------+----------+
|         100 | Steven      | 24000.00 |
|         101 | Neena       | 17000.00 |
|         102 | Lex         | 17000.00 |
|         103 | Alexander   |  9000.00 |
|         104 | Bruce       |  6000.00 |
```

6. Write a query to display first names of all employees that end with alphabet ‘N’.

SELECT first_name FROM employees WHERE first_name LIKE “%N”;

```
mysql> SELECT first_name FROM employees WHERE first_name LIKE "%N";
+------------+
| first_name |
+------------+
| Steven     |
| John       |
| Den        |
| Karen      |
| Kevin      |
| Steven     |
| Jason      |
| Stephen    |
| John       |
| John       |
| Karen      |
| Allan      |
| Harrison   |
| Ellen      |
| Jonathon   |
| Winston    |
| Jean       |
| Kevin      |
| Susan      |
| Hermann    |
+------------+
```

7. Write a query to display FIRST_NAME, JOB_ID, DEPARTMENT_ID of employees who are not PU_CLERK.

SELECT first_name, job_id, department_id FROM employees WHERE job_id<>”PU_CLERK”;

```
mysql> SELECT first_name, job_id, department_id FROM employees WHERE job_id<>"PU_CLERK";
+-------------+------------+---------------+
| first_name  | job_id     | department_id |
+-------------+------------+---------------+
| Steven      | AD_PRES    |            90 |
| Neena       | AD_VP      |            90 |
| Lex         | AD_VP      |            90 |
| Alexander   | IT_PROG    |            60 |
| Bruce       | IT_PROG    |            60 |
| David       | IT_PROG    |            60 |
| Valli       | IT_PROG    |            60 |
| Diana       | IT_PROG    |            60 |
| Nancy       | FI_MGR     |           100 |
| Daniel      | FI_ACCOUNT |           100 |
| John        | FI_ACCOUNT |           100 |
| Ismael      | FI_ACCOUNT |           100 |
| Jose Manuel | FI_ACCOUNT |           100 |
| Luis        | FI_ACCOUNT |           100 |
| Den         | PU_MAN     |            30 |
| Matthew     | ST_MAN     |            50 |
| Adam        | ST_MAN     |            50 |
| Payam       | ST_MAN     |            50 |
```

8. Write a query to display EMPLOYEE_ID, FIRST_NAME, and SALARY of those employees who do not have salaries of $3300, $3200, $2200.

SELECT employee_id, first_name, salary FROM employees WHERE salary NOT IN(3300, 3200, 2200);

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE salary NOT IN(3300, 3200, 2200);
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         100 | Steven     | 24000.00 |
|         101 | Neena      | 17000.00 |
|         102 | Lex        | 17000.00 |
|         103 | Alexander  |  9000.00 |
|         104 | Bruce      |  6000.00 |
|         105 | David      |  4800.00 |
```

9. Write a query to display names of those employees whose first name starts with ‘A’ and ends with ‘N’.

SELECT first_name FROM employees WHERE first_name LIKE “A%N”;

```
mysql> SELECT first_name FROM employees WHERE first_name LIKE "A%N";
+------------+
| first_name |
+------------+
| Allan      |
+------------+
```

10. Write a query to display the list of employee names that have letters ‘LA’ in their names.

SELECT first_name FROM employees WHERE first_name LIKE “%LA%”;

```
mysql> SELECT first_name FROM employees WHERE first_name LIKE "%LA%";
+------------+
| first_name |
+------------+
| Laura      |
| Allan      |
| Clara      |
| Alana      |
| Douglas    |
+------------+
```

11. Write a query to display the EMPLOYEE_ID, FIRST_NAME, and SALARY of employees. In that, the highest paid employee should display first and lowest paid should display last.

SELECT employee_id, first_name, salary FROM employees WHERE

salary=(SELECT MAX(salary) FROM employees) OR

salary=(SELECT MIN(salary) FROM employees);

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE
    -> salary=(SELECT MAX(salary) FROM employees) OR
    -> salary=(SELECT MIN(salary) FROM employees);
+-------------+------------+----------+
| employee_id | first_name | salary   |
+-------------+------------+----------+
|         100 | Steven     | 24000.00 |
|         132 | TJ         |  2100.00 |
+-------------+------------+----------+
```

12. Write a query to display FIRST_NAME of employees that have "a" in the second position.

SELECT first_name FROM employees WHERE first_name LIKE "_a%";

```
mysql> SELECT first_name FROM employees WHERE first_name LIKE "_a%";
+------------+
| first_name |
+------------+
| David      |
| Valli      |
| Nancy      |
| Daniel     |
| Karen      |
| Matthew    |
| Payam      |
| James      |
| Laura      |
| James      |
| Jason      |
| Hazel      |
| Randall    |
| Karen      |
| David      |
| Nanette    |
| Janette    |
| Patrick    |
| Sarath     |
| Danielle   |
| Mattea     |
| David      |
| Harrison   |
| Tayler     |
| Jack       |
| Martha     |
| Nandita    |
| Randall    |
| Sarah      |
| Samuel     |
| Vance      |
| Pat        |
+------------+
```

13. Write a query to display EMPLOYEE_ID, FIRST_NAME, and SALARY of employees whose salaries do not lies in the range of $1500 to $3000;

SELECT employee_id, first_name, salary FROM employees WHERE

salary NOT BETWEEN 1500 AND 3000;

```
mysql> SELECT employee_id, first_name, salary FROM employees WHERE
    -> salary NOT BETWEEN 1500 AND 3000;
+-------------+-------------+----------+
| employee_id | first_name  | salary   |
+-------------+-------------+----------+
|         100 | Steven      | 24000.00 |
|         101 | Neena       | 17000.00 |
|         102 | Lex         | 17000.00 |
|         103 | Alexander   |  9000.00 |
|         104 | Bruce       |  6000.00 |
|         105 | David       |  4800.00 |
|         106 | Valli       |  4800.00 |
```

14. Write a query to display FIRST_NAME, LAST_NAME and DEPARTMENT_ID of all employees in departments 30 or 100 in ascending order.

SELECT first_name, last_name, department_id FROM employees WHERE

department_id WHERE department_id IN(30,100) ORDER BY department_id;

```
mysql> SELECT first_name, last_name, department_id FROM employees WHERE
    -> department_id IN(30,100) ORDER BY department_id;
+-------------+------------+---------------+
| first_name  | last_name  | department_id |
+-------------+------------+---------------+
| Den         | Raphaely   |            30 |
| Alexander   | Khoo       |            30 |
| Shelli      | Baida      |            30 |
| Sigal       | Tobias     |            30 |
| Guy         | Himuro     |            30 |
| Karen       | Colmenares |            30 |
| Nancy       | Greenberg  |           100 |
| Daniel      | Faviet     |           100 |
| John        | Chen       |           100 |
| Ismael      | Sciarra    |           100 |
| Jose Manuel | Urman      |           100 |
| Luis        | Popp       |           100 |
+-------------+------------+---------------+
```

15. Write a query to display FIRST_NAME, LAST_NAME and SALARY for all employees whose salary is not in the range $10,000 through $15,000 and are in department 30 or 100.

SELECT first_name, last_name, salary FROM employees WHERE

department_id IN(30,100) AND

salary NOT BETWEEN 10000 AND 15000;

```
mysql> SELECT first_name, last_name, salary FROM employees WHERE
    -> department_id IN(30, 100) AND
    -> salary NOT BETWEEN 10000 AND 15000;
+-------------+------------+---------+
| first_name  | last_name  | salary  |
+-------------+------------+---------+
| Alexander   | Khoo       | 3100.00 |
| Shelli      | Baida      | 2900.00 |
| Sigal       | Tobias     | 2800.00 |
| Guy         | Himuro     | 2600.00 |
| Karen       | Colmenares | 2500.00 |
| Daniel      | Faviet     | 9000.00 |
| John        | Chen       | 8200.00 |
| Ismael      | Sciarra    | 7700.00 |
| Jose Manuel | Urman      | 7800.00 |
| Luis        | Popp       | 6900.00 |
+-------------+------------+---------+
```

16. Write a query to display FIRST_NAME, LAST_NAME and HIRE_DATE for all employees who were hired in 1987.

SELECT first_name, last_name, hire_date FROM employees WHERE

YEAR(hire_date)=1987;

```
mysql> SELECT first_name, last_name, hire_date FROM employees WHERE
    -> YEAR(hire_date)=1987;
+------------+-----------+------------+
| first_name | last_name | hire_date  |
+------------+-----------+------------+
| Steven     | King      | 1987-06-17 |
| Jennifer   | Whalen    | 1987-09-17 |
+------------+-----------+------------+
```

17. Write a query to display the LAST_NAME of employees whose LAST_NAME have exactly 6 characters.

SELECT last_name FROM employees WHERE LENGTH(last_name)=6;

```
mysql> SELECT last_name FROM employees WHERE LENGTH(last_name)=6;
+-----------+
| last_name |
+-----------+
| Hunold    |
| Austin    |
| Faviet    |
| Tobias    |
| Himuro    |
| Landry    |
| Markle    |
| Bissot    |
```

18. Write a query to display FIRST_NAME, SALARY and PF (15% of salary) of all employees.

SELECT first_name, salary, 0.15*salary as “PF(15% of salary)” FROM employees;

```
mysql> SELECT first_name, salary, 0.15*salary as "PF(15% of salary)" FROM employees;
+-------------+----------+-------------------+
| first_name  | salary   | PF(15% of salary) |
+-------------+----------+-------------------+
| Steven      | 24000.00 |         3600.0000 |
| Neena       | 17000.00 |         2550.0000 |
| Lex         | 17000.00 |         2550.0000 |
| Alexander   |  9000.00 |         1350.0000 |
| Bruce       |  6000.00 |          900.0000 |
| David       |  4800.00 |          720.0000 |
| Valli       |  4800.00 |          720.0000 |
| Diana       |  4200.00 |          630.0000 |
```

19. Write a query to display FIRST_NAME, SALARY and commission amount (% of salary) of all employees.

SELECT first_name, salary, salary*commission_pct as “commission amount” FROM employees

WHERE commission_pct IS NOT NULL;

```
mysql> SELECT first_name, salary, salary*commission_pct as "commission amount" FROM employees
    -> WHERE commission_pct IS NOT NULL;
+------------+----------+-------------------+
| first_name | salary   | commission amount |
+------------+----------+-------------------+
| John       | 14000.00 |         5600.0000 |
| Karen      | 13500.00 |         4050.0000 |
| Alberto    | 12000.00 |         3600.0000 |
| Gerald     | 11000.00 |         3300.0000 |
| Eleni      | 10500.00 |         2100.0000 |
| Peter      | 10000.00 |         3000.0000 |
| David      |  9500.00 |         2375.0000 |
| Peter      |  9000.00 |         2250.0000 |
| Christopher|  8000.00 |         1600.0000 |
| Nanette    |  7500.00 |         1500.0000 |
| Oliver     |  7000.00 |         1050.0000 |
| Janette    | 10000.00 |         3500.0000 |
| Patrick    |  9500.00 |         3325.0000 |
| Allan      |  9000.00 |         3150.0000 |
| Lindsey    |  8000.00 |         2400.0000 |
| Louise     |  7500.00 |         2250.0000 |
| Sarath     |  7000.00 |         1750.0000 |
| Clara      | 10500.00 |         2625.0000 |
| Danielle   |  9500.00 |         1425.0000 |
| Mattea     |  7200.00 |          720.0000 |
| David      |  6800.00 |          680.0000 |
| Sundar     |  6400.00 |          640.0000 |
| Amit       |  6200.00 |          620.0000 |
| Lisa       | 11500.00 |         2875.0000 |
| Harrison   | 10000.00 |         2000.0000 |
| Tayler     |  9600.00 |         1920.0000 |
| William    |  7400.00 |         1110.0000 |
| Elizabeth  |  7300.00 |         1095.0000 |
| Sundita    |  6100.00 |          610.0000 |
| Ellen      | 11000.00 |         3300.0000 |
| Alyssa     |  8800.00 |         2200.0000 |
| Jonathon   |  8600.00 |         1720.0000 |
| Jack       |  8400.00 |         1680.0000 |
| Kimberely  |  7000.00 |         1050.0000 |
| Charles    |  6200.00 |          620.0000 |
+------------+----------+-------------------+
```

20. Write a query to display FIRST_NAME, SALARY and NET_SALARY after 500 deduction from salary of all employees;

SELECT first_name, salary, salary-500 as “net_salary” FROM employees;

```
mysql> SELECT first_name, salary, salary-500 as "net_salary" FROM employees;
+-------------+----------+------------+
| first_name  | salary   | net_salary |
+-------------+----------+------------+
| Steven      | 24000.00 |   23500.00 |
| Neena       | 17000.00 |   16500.00 |
| Lex         | 17000.00 |   16500.00 |
| Alexander   |  9000.00 |    8500.00 |
| Bruce       |  6000.00 |    5500.00 |
| David       |  4800.00 |    4300.00 |
| Valli       |  4800.00 |    4300.00 |
```

Lab 03: SQL Functions

1. Write a query to generate new names of the employees by combining the first 3 characters of the First_Name and last 3 characters of the job.

SELECT CONCAT(LEFT(first_name,3), LOWER(RIGHT(job_id,3))) as new_names FROM employees;

```
mysql> SELECT CONCAT(LEFT(first_name,3), LOWER(RIGHT(job_id,3))) as new_names FROM employees;
+-----------+
| new_names |
+-----------+
| Steres    |
| Nee_vp    |
| Lex_vp    |
| Alerog    |
| Brurog    |
| Davrog    |
| Valrog    |
| Diarog    |
```

2. Generate new jobs of the employees by changing letter E with A in the existing jobs.

SELECT REPLACE(job_id,”E”,”A”) as new_jobs FROM employees;

```
mysql> SELECT REPLACE(job_id, "E","A") as new_job FROM employees;
+------------+
| new_job    |
+------------+
| AC_ACCOUNT |
| AC_MGR     |
| AD_ASST    |
| AD_PRAS    |
| AD_VP      |
| AD_VP      |
| FI_ACCOUNT |
| FI_ACCOUNT |
| FI_ACCOUNT |
```

3. Write a query to Display Names, hire date years in the department.

SELECT first_name, last_name, YEAR(hire_date) FROM employees;

```
mysql> SELECT first_name, last_name, YEAR(hire_date) FROM employees;
+-------------+-------------+-----------------+
| first_name  | last_name   | YEAR(hire_date) |
+-------------+-------------+-----------------+
| Steven      | King        |            1987 |
| Neena       | Kochhar     |            1989 |
| Lex         | De Haan     |            1993 |
| Alexander   | Hunold      |            1990 |
| Bruce       | Ernst       |            1991 |
```

4. Write a query to display names, hire date of all the employees who were hired before July 30, 1987. Keeping this in mind that hire dates should be displayed in the format

"MONTH DATE, YEAR". Also date in the where clause should be in the format "MONTH DATE, YEAR".

SELECT first_name, last_name, DATE_FORMAT(hire_date,"%M %d, %Y") FROM employees WHERE hire_date<"1987-6-30";

```
mysql> SELECT first_name, last_name, DATE_FORMAT(hire_date,"%M %d, %Y") as hire_date FROM employees WHERE hire_date<"1987-7-30";
+------------+-----------+---------------+
| first_name | last_name | hire_date     |
+------------+-----------+---------------+
| Steven     | King      | June 17, 1987 |
+------------+-----------+---------------+
```

5. Write a query to display the last day of the current month & three months before the current month.

SELECT LAST_DATE(CURRENT_DATE()), ADDDATE(CURRENT_DATE(), INTERVAL -3 MONTH);

```
mysql> SELECT LAST_DAY(CURRENT_DATE()), ADDDATE(CURRENT_DATE(), INTERVAL -3 MONTH);
+--------------------------+-------------------------------------------+
| LAST_DAY(CURRENT_DATE()) | ADDDATE(CURRENT_DATE(), INTERVAL -3 MONTH) |
+--------------------------+-------------------------------------------+
| 2022-10-31               | 2022-07-04                                |
+--------------------------+-------------------------------------------+
```

6. Write a query to get the a day of the current year

SELECT DAYOFYEAR(CURRENT_DATE());

```
mysql> SELECT DAYOFYEAR(CURRENT_DATE());
+---------------------------+
| DAYOFYEAR(CURRENT_DATE()) |
+---------------------------+
|                       277 |
+---------------------------+
```

7. Write a query to get the current date in the following format.

SELECT DATE_FORMAT(CURRENT_DATE(), "%M %d, %Y");

```
mysql> SELECT DATE_FORMAT(CURRENT_DATE(),"%M %d, %Y");
+-----------------------------------------+
| DATE_FORMAT(CURRENT_DATE(),"%M %d, %Y") |
+-----------------------------------------+
| October 04, 2022                        |
+-----------------------------------------+
```

8. Write a query to get the current date in Thursday September 2014 format.

SELECT DATE_FORMAT(CURRENT_DATE(),"%W %M %Y");

```
mysql> SELECT DATE_FORMAT(CURRENT_DATE(), "%W %M %Y");
+-----------------------------------------+
| DATE_FORMAT(CURRENT_DATE(), "%W %M %Y") |
+-----------------------------------------+
| Thursday October 2022                   |
+-----------------------------------------+
```

9. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'.

SELECT first_name, hire_date FROM employees WHERE hire_date BETWEEN “1987-06-01” AND “1987-07-30”;

```
mysql> SELECT first_name, hire_date FROM employees WHERE hire_date BETWEEN "1987-06-01" AND "1987-07-30";
+------------+------------+
| first_name | hire_date  |
+------------+------------+
| Steven     | 1987-06-17 |
+------------+------------+
```

10. Write a query to display the current date in the 05/09/2014 format.

SELECT DATE_FORMAT(CURRENT_DATE(), “%d/%m/%Y”);

```
mysql> SELECT DATE_FORMAT(CURRENT_DATE(), "%d/%m/%Y");
+-----------------------------------------+
| DATE_FORMAT(CURRENT_DATE(), "%d/%m/%Y") |
+-----------------------------------------+
| 06/10/2022                              |
+-----------------------------------------+
```

11. Write a query to get the firstname, lastname who joined in the month of June.

SELECT first_name, last_name FROM employees WHRER MONTH(hire_date)=6;

```
mysql> SELECT first_name, last_name FROM employees WHERE MONTH(hire_date)=6;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Steven     | King      |
| David      | Austin    |
| Jason      | Mallin    |
| Martha     | Sullivan  |
| Julia      | Dellinger |
| Kelly      | Chung     |
| Donald     | OConnell  |
| Susan      | Mavris    |
| Hermann    | Baer      |
| Shelley    | Higgins   |
| William    | Gietz     |
+------------+-----------+
```

12. Write a query to append '@iba-suk.edu.pk' to email field.

SELECT CONCAT(email, “@iba-suk.edu.pk”) FROM employees;

```
mysql> SELECT CONCAT(email,"@iba-suk.edu.pk") FROM employees;
+---------------------------------+
| CONCAT(email,"@iba-suk.edu.pk") |
+---------------------------------+
| SKING@iba-suk.edu.pk            |
| NKOCHHAR@iba-suk.edu.pk         |
| LDEHAAN@iba-suk.edu.pk          |
| AHUNOLD@iba-suk.edu.pk          |
| BERNST@iba-suk.edu.pk           |
| DAUSTIN@iba-suk.edu.pk          |
```

13. Write a query to get the employee id, first name and hire month using MID().

SELECT employee_id, first_name, MID(hire_date,6,2) FROM employees;

```
mysql> SELECT employee_id, first_name, MID(hire_date,6,2) FROM employees;
+-------------+-------------+--------------------+
| employee_id | first_name  | MID(hire_date,6,2) |
+-------------+-------------+--------------------+
|         100 | Steven      | 06                 |
|         101 | Neena       | 09                 |
|         102 | Lex         | 01                 |
|         103 | Alexander   | 01                 |
|         104 | Bruce       | 05                 |
```

14. Write a query to extract the last 4 character of phone numbers.

SELECT RIGHT(hire_date, 4) FROM employees;

```
mysql> SELECT RIGHT(phone_number,4) FROM employees;
+-----------------------+
| RIGHT(phone_number,4) |
+-----------------------+
| 4567                  |
| 4568                  |
| 4569                  |
| 4567                  |
| 4568                  |
| 4569                  |
```

15. Write a query to get the last word of the street address(from locations able).

SELECT SUBSTRING_INDEX(TRIM(street_address), " ", -1) FROM locations;

```
mysql> SELECT SUBSTRING_INDEX(TRIM(street_address), " ", -1) FROM locations;
+-----------------------------------------------+
| SUBSTRING_INDEX(TRIM(street_address), " ", -1) |
+-----------------------------------------------+
| Rie                                           |
| Testa                                         |
| Shinjuku-ku                                   |
| Kamiya-cho                                    |
| Rd                                            |
| Blvd                                          |
| St                                            |
| Rd                                            |
| Ave                                           |
| St                                            |
| Laogianggen                                   |
| (E)                                           |
| Street                                        |
| North                                         |
| St                                            |
| Park                                          |
| Road                                          |
| 7031                                          |
| 1360                                          |
| Corps-Saints                                  |
| 921                                           |
| 837                                           |
| 9991                                          |
+-----------------------------------------------+
```

16. Write a query to get the locations that have minimum street length.(locations tbl).

SELECT street_address FROM locations FROM employees

WHERE LENGTH(TRIM(street_address)) =

(SELECT MIN(LENGTH(street_address)) FROM locations);

```
mysql> SELECT street_address FROM locations WHERE LENGTH(TRIM(street_address))=(SELECT MIN(LENGTH(street_address)) FROM locations);
+----------------+
| street_address |
+----------------+
| 2007 Zagora St |
| 8204 Arthur St |
+----------------+
```

17. Write a query to display the first word from those job titles which contains more than one words.

SELECT MID(job_title, 1, INSTR(job_title,” ”)-1) FROM employees;

```
mysql> SELECT MID(job_title, 1,INSTR(job_title, " ")-1) FROM jobs;
+-----------------------------------------+
| MID(job_title, 1,INSTR(job_title, " ")-1) |
+-----------------------------------------+
| Public                                  |
| Accounting                              |
| Administration                          |
|                                         |
| Administration                          |
|                                         |
| Finance                                 |
| Human                                   |
|                                         |
| Marketing                               |
| Marketing                               |
| Public                                  |
| Purchasing                              |
| Purchasing                              |
| Sales                                   |
| Sales                                   |
| Shipping                                |
| Stock                                   |
| Stock                                   |
+-----------------------------------------+
```

18. Write a query to display the length of first name for employees where last name contain character 'c' after 2nd position.

SELECT LENGTH(first_name) FROM employees WHERE INSTR(last_name,”c”)>2;

```
mysql> SELECT LENGTH(first_name) FROM employees WHERE INSTR(last_name,"c")>2;
+--------------------+
| LENGTH(first_name) |
+--------------------+
|                  5 |
|                  5 |
|                  7 |
+--------------------+
```

Lab 04 : SQL Functions And Regular Expression

1. Write a query to lists the number of employees in each department.

SELECT department_id, COUNT(*) FROM employees GROUP BY department_id;

2. Write a query to display the department id where at least 5 employees should be in each department.

SELECT department_id FROM employees GROUP BY department_id HAVING count(*) >= 5;

3. Write a query to display all columns of those employees who has first name is unique.

SELECT * FROM employees group by first_name;

4. Write a SQL query to get name of employees containing exactly four characters.

SELECT first_name FROM employees WHERE length(first_name)=4;

5. Write a query to display the list of employee names that have letters ‘LA’ in their names.

SELECT first_name FROM employees WHERE first_name REGEXP 'LA+';

6. Write a query to display names of those employees whose first name starts with ‘A’ and ends with ‘N’.

SELECT first_name FROM employees WHERE first_name like 'A%n';

7. Write a query to display first names of all employees that end with alphabet ‘N’.

SELECT first_name FROM employees WHERE first_name REGEXP 'N$';

8. Write a query to display FIRST_NAME, LASTNAME of all employees whose first name starts with letter ‘A’.

SELECT first_name, last_name FROM employees WHERE first_name REGEXP '^[a]';

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

SELECT job_id, COUNT(*) FROM employees GROUP BY job_id;

10. Display the manager number and the salary of the lowest paid employee of that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is 2000. Sort the output is descending order of the salary.

SELECT phone_number, MIN(salary) FROM employees WHERE manager_id IS NOT NULL GROUP BY manager_id ORDER BY MIN(salary) DESC;

11.Display the total number of employees who have no commission.

SELECT commission_pct,COUNT(*) FROM employees WHERE commission_pct IS NULL;

12.Write a query to display FIRST_NAME, LASTNAME of all employees whose first name small ‘t’.

SELECT first_name, last_name FROM employees WHERE first_name REGEXP '[t]';

Lab 05: Working With Joins

1. Write a query in SQL to display the first name, last name, department number, and department name for each employee. (Sample tables: employees & departments).

Query: SELECT first_name, last_name, department_id, department_name FROM employees INNER JOIN departments USING (department_id);

first_name	last_name	department_id	department_name
Jennifer	Whalen	10	Administration
Michael	Hartstein	20	Marketing
Pat	Fay	20	Marketing
Den	Raphaely	30	Purchasing
Alexander	Khoo	30	Purchasing
Shelli	Baida	30	Purchasing

2. Write a query to find the name (first_name, last_name), job, department ID and name of the department who works in London. (Sample tables: employees , locations & departments)

Query: SELECT CONCAT(first_name, " ", last_name), job_id, department_id, department_name FROM employees INNER JOIN departments USING(department_id) INNER JOIN locations USING(location_id) WHERE city="London";

CONCAT(first_name, " ", last_name)	job_id	department_id	department_name
Susan Mavris	HR_REP	40	Human Resources

3. Write a query to find the employee id, name (last_name) along with their manager_id and name (last_name). (Sample tables: employees)

Query: SELECT e1.employee_id, e1.last_name, e2.employee_id, e2.last_name FROM employees as e1, employees as e2 WHERE e1.manager_id=e2.employee_id;

employee_id	last_name	employee_id	last_name
101	Kochhar	100	King
102	De Haan	100	King
103	Hunold	102	De Haan
104	Ernst	103	Hunold
105	Austin	103	Hunold

4. Write a query to find the name (first_name, last_name) and hire date of the employees who was hired after 'Jones'. (Sample tables: employees)

SELECT CONCAT(e1.first_name, " ", e1.last_name), e1.hire_date, CONCAT(e2.first_name, " ", e2.last_name), e2.hire_date FROM employees as e1, employees as e2 WHERE e1.hire_date > e2.hire_date AND e2.last_name="Jones";

	CONCAT(e1.first_name, " ", e1.last_name)	hire_date	CONCAT(e2.first_name, " ", e2.last_name)	hire_date
▶	Luis Popp	1999-12-07	Vance Jones	1999-03-17
	Karen Colmenares	1999-08-10	Vance Jones	1999-03-17
	Kevin Mourgos	1999-11-16	Vance Jones	1999-03-17
	Steven Markle	2000-03-08	Vance Jones	1999-03-17

5. Write a query to get the department name and number of employees in the department. (Sample tables: employees & departments)

Query: SELECT d.department_name as "Department Name", COUNT(d.department_name) as "Number of Employees" FROM employees as e INNER JOIN departments as d USING(department_id) GROUP BY d.department_name;

	Department Name	Number of Employees
▶	Administration	1
	Marketing	2
	Purchasing	6
	Human Resources	1

6. Write a query to display the department ID and name and first name of manager. (Sample tables: employees & departments)

SELECT d.department_id, d.department_name, e.first_name FROM employees as e INNER JOIN departments as d USING(department_id);

	department_id	department_name	first_name
▶	10	Administration	Jennifer
	20	Marketing	Michael
	20	Marketing	Pat
	30	Purchasing	Den
	30	Purchasing	Alexander

7. Write a query to display the department name, manager name, and city. (Sample tables: employees , locations & departments)

SELECT department_name, CONCAT(first_name, " ", last_name) as manager_name, city FROM employees as e INNER JOIN departments as d USING(department_id) INNER JOIN locations USING(location_id) GROUP BY city;

	department_name	manager_name	city
▶	Executive	Steven King	Seattle
	IT	Alexander Hunold	Southlake
	Shipping	Matthew Weiss	South San Francisco
	Sales	John Russell	Oxford
	Marketing	Michael Hartstein	Toronto

8. Write a query to display the job history that were done by any employee who is currently drawing more than 10000 of salary. (Sample tables: employees & job_history)

SELECT jh.employee_id, jh.start_date, jh.end_date, jh.job_id, jh.department_id FROM employees as e INNER JOIN job_history as jh USING(job_id) WHERE salary > 10000;

employee_id	start_date	end_date	job_id	department_id
101	1993-10-28	1997-03-15	AC_MGR	110
176	1998-03-24	1998-12-31	SA_REP	80
176	1998-03-24	1998-12-31	SA_REP	80
176	1998-03-24	1998-12-31	SA_REP	80
176	1999-01-01	1999-12-31	SA_MAN	80
176	1999-01-01	1999-12-31	SA_MAN	80
176	1999-01-01	1999-12-31	SA_MAN	80
176	1999-01-01	1999-12-31	SA_MAN	80
176	1999-01-01	1999-12-31	SA_MAN	80

9. Write a query to display the first name, last name, hire date, salary of the manager for all managers whose experience is more than 15 years. (Sample tables: employees & departments)

SELECT first_name, last_name, hire_date, salary, DATEDIFF(CURRENT_DATE(),hire_date)/365 as emperience FROM employees as e INNER JOIN departments as d ON (d.manager_id=e.employee_id) WHERE DATEDIFF(CURRENT_DATE(), hire_date)/365 > 15;

first_name	last_name	hire_date	salary	emperience
Steven	King	1987-06-17	24000.00	35.4959
Alexander	Hunold	1990-01-03	9000.00	32.9452
Nancy	Greenberg	1994-08-17	12000.00	28.3233
Den	Raphaely	1994-12-07	11000.00	28.0164
Adam	Fripp	1997-04-10	8200.00	25.6740

10. Write a query in SQL to display the name of the department, average salary and number of employees working in that department who got commission. (Sample tables: employees & departments)

SELECT d.department_name, AVG(e.salary), COUNT(*) as number_of_employees FROM employees as e INNER JOIN departments as d USING(department_id) WHERE commission_pct IS NOT NULL;

department_name	AVG(e.salary)	number_of_employees
Sales	8955.882353	34

11. Write a query in SQL to display the name of the country, city, and the departments which are running there. (Sample tables: countries , locations & departments)

SELECT country_name, city, department_name FROM countries as c INNER JOIN locations as l USING(country_id) INNER JOIN departments USING (location_id);

country_name	city	department_name
United States of America	Seattle	Administration
Canada	Toronto	Marketing
United States of America	Seattle	Purchasing
United Kingdom	London	Human Resources
United States of America	South San Francisco	Shipping

12. Write a query in SQL to display department name and the full name (first and last name) of the manager. (Sample tables: employees & departments)

SELECT department_name, CONCAT(first_name, " ", last_name) as manger_name FROM employees as e INNER JOIN departments as d ON (e.employee_id=d.manager_id);

department_name	manger_name
Administration	Jennifer Whalen
Marketing	Michael Hartstein
Purchasing	Den Raphaely
Human Resources	Susan Mavris
Shipping	Adam Fripp
IT	Alexander Hunold
Public Relations	Hermann Baer
Sales	John Russell
Executive	Steven King
Finance	Nancy Greenberg
Accounting	Shelley Higgins

13. Write a query in SQL to display the details of jobs which was done by any of the employees who is presently earning a salary on and above 12000. (Sample tables: employees & job_history)

SELECT jh.employee_id, jh.start_date, jh.end_date, jh.job_id, jh.department_id, e.salary FROM employees as e, job_history as jh WHERE e.job_id=jh.job_id AND e.salary >= 12000;

employee_id	start_date	end_date	job_id	department_id	salary
101	1993-10-28	1997-03-15	AC_MGR	110	12000.00
176	1999-01-01	1999-12-31	SA_MAN	80	14000.00
176	1999-01-01	1999-12-31	SA_MAN	80	13500.00
176	1999-01-01	1999-12-31	SA_MAN	80	12000.00

14. Write a query in SQL to display the full name (first and last name), and salary of those employees who working in any department located in London. (Sample tables: employees , locations & departments)

SELECT CONCAT(first_name, " ", last_name) as full_name, salary FROM employees as e, locations as l, departments as d WHERE e.department_id=d.department_id AND d.location_id=l.location_id AND city="London";

full_name	salary
Susan Mavris	6500.00

15. Write a query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job. (Sample tables: employees & jobs)

16. SELECT job_title, CONCAT(first_name, " ", last_name) as employee_name, MIN(salary) as minimum_salary FROM employees as e, jobs as j WHERE e.job_id=j.job_id GROUP BY e.job_id;

job_title	employee_name	minimum_salary
Public Accountant	William Gietz	8300.00
Accounting Manager	Shelley Higgins	12000.00
Administration Assistant	Jennifer Whalen	4400.00
President	Steven King	24000.00
Administration Vice President	Neena Kochhar	17000.00

17. Write a query to display the job title and average salary of employees. (Sample tables: employees & jobs)

SELECT j.job_title, avg(e.salary) average_salary FROM employees as e, jobs as j WHERE e.job_id=j.job_id GROUP BY j.job_title;

job_title	average_salary
Public Accountant	8300.000000
Accounting Manager	12000.000000
Administration Assistant	4400.000000
President	24000.000000
Administration Vice President	17000.000000

18. Write a query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90 from job history. (Sample tables: jobs & job_history)

SELECT jh.employee_id, j.job_title, DATEDIFF(jh.end_date, jh.start_date) as num_of_days FROM jobs as j, job_history as jh WHERE j.job_id=jh.job_id AND department_id=90;

employee_id	job_title	num_of_days
200	Administration Assistant	2100
200	Public Accountant	1644

Lab 06 : Working With Subqueries

1. Write a query in SQL to display details of those employees who have changed jobs at least once. (Sample tables: employees & job_history)

SELECT * FROM job_history as jh, employees as e WHERE jh.employee_id=e.employee_id AND (SELECT COUNT(*) FROM job_history as j WHERE e.employee_id=j.employee_id)>=2;

employee_id	start_date	end_date	job_id	department_id	employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
101	1989-09-21	1993-10-27	AC_ACCOUNT	110	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000.00	NULL	100	90
101	1993-10-28	1997-03-15	AC_MGR	110	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000.00	NULL	100	90
176	1998-03-24	1998-12-31	SA_REP	80	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	1998-03-24	SA_REP	8600.00	0.20	149	80
176	1999-01-01	1999-12-31	SA_MAN	80	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	1998-03-24	SA_REP	8600.00	0.20	149	80
200	1987-09-17	1993-06-17	AD_ASST	90	200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400.00	NULL	101	10
200	1994-07-01	1998-12-31	AC_ACCOUNT	90	200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400.00	NULL	101	10

2. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name='Bull'. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary > (SELECT salary FROM employees WHERE last_name="Bull");

name	salary
Steven King	24000.00
Neena Kochhar	17000.00
Lex De Haan	17000.00
Alexander Hunold	9000.00
Bruce Ernst	6000.00

3. Write a query to find the name (first_name, last_name) of all employees who works in the IT department. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name FROM employees WHERE department_id = (SELECT department_id FROM departments WHERE department_name="IT");

name
Alexander Hunold
Bruce Ernst
David Austin
Valli Pataballa
Diana Lorentz

4. Write a query to find the name (first_name, last_name) of the employees who have a manager and worked in a USA based department. (Sample tables: employees, departments & locations)

SELECT CONCAT(first_name, " ", last_name) as name FROM employees WHERE manager_id = ANY (SELECT d.manager_id FROM departments as d, locations as l WHERE d.location_id=l.location_id AND l.country_id="US");

	name
▶	Alexander Khoo
	Shelli Baida
	Sigal Tobias
	Guy Himuro
	Karen Colmenares

5. Write a query to find the name (first_name, last_name) of the employees who are managers. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name FROM employees WHERE employee_id = ANY (SELECT DISTINCT manager_id FROM employees);

	name
▶	Steven King
	Neena Kochhar
	Lex De Haan
	Alexander Hunold
	Nancy Greenberg

6. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary > ANY (SELECT AVG(salary) FROM employees);

	name	salary
▶	Steven King	24000.00
	Neena Kochhar	17000.00
	Lex De Haan	17000.00
	Alexander Hunold	9000.00
	Nancy Greenberg	12000.00

7. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is equal to the minimum salary for their job grade. (Sample tables: employees & jobs)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees as e WHERE salary = (SELECT min_salary FROM jobs as j WHERE e.job_id=j.job_id);

	name	salary
▶	Karen Colmenares	2500.00
	Martha Sullivan	2500.00
	Randall Perkins	2500.00

8. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the average salary and works in any of the IT departments. (Sample tables: employees & departments)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees as e, departments as d WHERE e.department_id=d.department_id AND d.department_name = ANY (SELECT department_name FROM departments WHERE department_name LIKE "%IT%") AND e.salary >

(SELECT AVG(salary) FROM employees as e1,departments d1 WHERE e1.department_id = d1.department_id AND department_name LIKE "%IT%");

name	salary
Alexander Hunold	9000.00
Bruce Ernst	6000.00

9. Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the earning of Mr. Bell. (Sample tables: employees & departments)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary > (SELECT salary FROM employees WHERE last_name LIKE "%Bell%");

name	salary
Steven King	24000.00
Neena Kochhar	17000.00
Lex De Haan	17000.00
Alexander Hunold	9000.00
Bruce Ernst	6000.00

10. Write a query to find the name (first_name, last_name), and salary of the employees who earn the same salary as the minimum salary for all departments. (Sample tables: employees & departments)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary = SOME (SELECT MIN(salary) FROM employees);

name	salary
TJ Olson	2100.00

11. Write a query to find the name (first_name, last_name), and salary of the employees whose salary is greater than the average salary of all departments. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary > ALL (SELECT AVG(salary) FROM employees GROUP BY department_id);

name	salary
Steven King	24000.00

12. Write a query to find the name (first_name, last_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB_ID = 'SH_CLERK'). Sort the results of the salary of the lowest to highest. . (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name, salary FROM employees WHERE salary > ALL (SELECT salary FROM employees WHERE job_id="SH_CLERK");

name	salary
Steven King	24000.00
Neena Kochhar	17000.00
Lex De Haan	17000.00
Alexander Hunold	9000.00
Bruce Ernst	6000.00

13. Write a query to find the name (first_name, last_name) of the employees who are not supervisors. (Sample tables: employees)

SELECT CONCAT(first_name, " ", last_name) as name FROM employees as e WHERE e.employee_id NOT IN (SELECT manager_id FROM employees);

name

14. Write a query to display the employee ID, first name, last name, and department names of all employees. (Sample tables: employees & departments)

SELECT employee_id, first_name, last_name, department_name FROM employees as e, departments as d WHERE e.department_id=d.department_id;

employee_id	first_name	last_name	department_name
200	Jennifer	Whalen	Administration
201	Michael	Hartstein	Marketing
202	Pat	Fay	Marketing
114	Den	Raphaely	Purchasing
115	Alexander	Khoo	Purchasing

15. Write a query to display the employee ID, first name, last name, salary of all employees whose salary is above average for their departments. (Sample tables: employees & departments)

SELECT employee_id, first_name, last_name, salary FROM employees as e1 WHERE e1.salary > (SELECT AVG(salary) FROM employees e2 GROUP BY department_id HAVING e1.department_id = e2.department_id);

employee_id	first_name	last_name	salary
100	Steven	King	24000.00
103	Alexander	Hunold	9000.00
104	Bruce	Ernst	6000.00
108	Nancy	Greenberg	12000.00
109	Daniel	Faviet	9000.00

Lab 07: DDL QUERIS

1. Create a replica of Employee table with all the records in it.

CREATE TABLE replicated_employees as SELECT * FROM employees;

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17000.00	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000.00	NULL	102	60

2. Add a column ‘Address’ in it.

ALTER TABLE replicated_employees ADD (address VARCHAR(200));

Field	Type	Null	Key	Default	Extra
employee_id	int unsigned	NO		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(25)	NO		NULL	
email	varchar(25)	NO		NULL	
phone_number	varchar(20)	YES		NULL	
hire_date	date	NO		NULL	
job_id	varchar(10)	NO		NULL	
salary	decimal(8,2)	NO		NULL	
commission_pct	decimal(2,2)	YES		NULL	
manager_id	int unsigned	YES		NULL	
department_id	int unsigned	YES		NULL	
address	varchar(200)	YES		NULL	

3. Drop column ‘Address’ from it.

ALTER TABLE replicated_employees DROP address;

Field	Type	Null	Key	Default	Extra
employee_id	int unsigned	NO		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(25)	NO		NULL	
email	varchar(25)	NO		NULL	
phone_number	varchar(20)	YES		NULL	
hire_date	date	NO		NULL	
job_id	varchar(10)	NO		NULL	
salary	decimal(8,2)	NO		NULL	
commission_pct	decimal(2,2)	YES		NULL	
manager_id	int unsigned	YES		NULL	
department_id	int unsigned	YES		NULL	

4. Add columns ‘House No’ character ,’Street No’ numeric, ’Area’ character ,’City’ character in it with the respective data types.

ALTER TABLE replicated_employees ADD(

house_number VARCHAR(15),

street_number VARCHAR(15),

area VARCHAR(25),

city VARCHAR(25)

);

Field	Type	Null	Key	Default	Extra
employee_id	int unsigned	NO		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(25)	NO		NULL	
email	varchar(25)	NO		NULL	
phone_number	varchar(20)	YES		NULL	
hire_date	date	NO		NULL	
job_id	varchar(10)	NO		NULL	
salary	decimal(8,2)	NO		NULL	
commission_pct	decimal(2,2)	YES		NULL	
manager_id	int unsigned	YES		NULL	
department_id	int unsigned	YES		NULL	
house_number	varchar(15)	YES		NULL	
street_number	varchar(15)	YES		NULL	
area	varchar(25)	YES		NULL	
city	varchar(25)	YES		NULL	

5. Change the data type of ‘House No’ from character to numeric.

ALTER TABLE replicated_employees MODIFY house_number INT(20);

Field	Type	Null	Key	Default	Extra
employee_id	int unsigned	NO		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(25)	NO		NULL	
email	varchar(25)	NO		NULL	
phone_number	varchar(20)	YES		NULL	
hire_date	date	NO		NULL	
job_id	varchar(10)	NO		NULL	
salary	decimal(8,2)	NO		NULL	
commission_pct	decimal(2,2)	YES		NULL	
manager_id	int unsigned	YES		NULL	
department_id	int unsigned	YES		NULL	
house_number	int	YES		NULL	
street_number	varchar(15)	YES		NULL	
area	varchar(25)	YES		NULL	
city	varchar(25)	YES		NULL	

6. Create the Data Definitions for each of the relations shown below, using SQL DDL. Assume the following attributes and data types:

CREATE DATABASE iba2;

USE iba2;

```
CREATE TABLE faculty(
        faculty_id INT PRIMARY KEY,
        faculty_name VARCHAR(25)
);
```

Field	Type	Null	Key	Default	Extra
faculty_id	int	NO	PRI	NULL	
faculty_name	varchar(50)	YES		NULL	

```
CREATE TABLE course (
        course_id VARCHAR(8) PRIMARY KEY,
        course_name VARCHAR(15)
);
```

Field	Type	Null	Key	Default	Extra
course_id	char(8)	NO	PRI	NULL	
course_name	char(15)	YES		NULL	

```
CREATE TABLE class (
        class_id VARCHAR(8),
        course_id VARCHAR(8),
        section_number INT,
        semester VARCHAR(10),
        FOREIGN KEY(course_id) REFERENCES course(course_id)
);
```

Field	Type	Null	Key	Default	Extra
class_id	char(8)	YES		NULL	
course_id	char(8)	YES	MUL	NULL	
section_no	int	YES		NULL	
semester	char(10)	YES		NULL	

7. How would you add an attribute, CLASS, to the STUDENT table?

```
ALTER TABLE student ADD (class VARCHAR(20));
```

	Field	Type	Null	Key	Default	Extra
▶	std_NO	varchar(8)	NO	PRI	NULL	
	std_Name	varchar(30)	NO		NULL	
	Department	varchar(30)	NO		NULL	
	email	varchar(30)	NO		NULL	
	phone	varchar(30)	YES		NULL	
	project_id	int	YES	MUL	NULL	
	daysToComplete	int	YES		NULL	
	class	varchar(20)	YES		NULL	

8. Write a SQL statement to rename the table department to dept (with both methods).

One way)

RENAME TABLE department TO dep;

	Tables_in_hr
▶	class
	countries
	course
	dep

Second Way

ALTER TABLE dep RENAME TO department;

	Tables_in_hr
▶	class
	countries
	course
	department

9. Write a SQL statement to add a column regionId to the table locations.

ALTER TABLE locations ADD (region_id VARCHAR(15));

	Field	Type	Null	Key	Default	Extra
▶	location_id	int unsigned	NO	PRI	NULL	auto_increment
	street_address	varchar(40)	YES		NULL	
	postal_code	varchar(12)	YES		NULL	
	city	varchar(30)	NO		NULL	
	state	varchar(25)	YES		NULL	
	country_id	char(2)	NO	MUL	NULL	
	region_id	varchar(15)	YES		NULL	

10. Write a SQL statement to change the name of the column state_province to state in locations table, keeping the data type and size same.

ALTER TABLE locations CHANGE COLUMN state_province state VARCHAR(25);

	Field	Type	Null	Key	Default	Extra
▶	location_id	int unsigned	NO	PRI	NULL	auto_increment
	street_address	varchar(40)	YES		NULL	
	postal_code	varchar(12)	YES		NULL	
	city	varchar(30)	NO		NULL	
	state	varchar(25)	YES		NULL	
	country_id	char(2)	NO	MUL	NULL	
	region_id	varchar(15)	YES		NULL	

Lab 08: Constraints

1. Create a table DEPARTMENT with the following attributes:

DEPTNO number, DNAME varchar(10), LOC varchar(10).

PRIMARY KEY constraint on DEPTNO.

```
CREATE TABLE department (
    deptno INT,
    dname VARCHAR(10),
    loc VARCHAR(10),
    PRIMARY KEY(deptno)
);
```

Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	varchar(10)	YES		NULL	
loc	varchar(10)	YES		NULL	

2. Create a table EMPLOYEE with the following attributes:

EMPNO number, ENAME varchar(10),SAL number, DEPTNO number.

Apply FOREIGN KEY constraint on DEPTNO referencing the DEPARTMENT table created in question 1 and PRIMARY KEY constraint on EMPNO and DEPTNO.

```
CREATE TABLE employee (
    empno INT,
    ename VARCHAR(10),
    salary numeric(10,2),
    deptno INT,
    FOREIGN KEY(deptno) REFERENCES department(deptno),
    PRIMARY KEY(empno, deptno)
);
```

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(10)	YES		NULL	
salary	decimal(10,2)	YES		NULL	
deptno	int	NO	PRI	NULL	

3. ALTER table EMPLOYEE created in question 2 and apply the constraint CHECK on ENAME attribute such that ENAME should always be inserted in capital letters.

ALTER TABLE employee ADD CHECK (UCASE(ename)=TRUE);

Field	Type	Null	Key	Default	Extra
salary	decimal(10,2)	YES		NULL	
ename	varchar(10)	YES		NULL	
empno	int	NO	PRI	NULL	
deptno	int	NO	PRI	NULL	

4. ALTER table DEPARTMENT created in question 1 and apply constraint on DNAME such that DNAME should not be entered empty.

ALTER TABLE department CHANGE dname dname VARCHAR(10) NOT NULL;

Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
dname	varchar(10)	NO		NULL	
loc	varchar(10)	YES		NULL	

5. ALTER table EMPLOYEE created in question 2 and apply the constraint on SAL attribute such that no two salaries of the employees should be similar.

ALTER TABLE employee CHANGE salary salary numeric(10,2) UNIQUE;

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(10)	YES		NULL	
salary	decimal(10,2)	YES	UNI	NULL	
deptno	int	NO	PRI	NULL	

6. ALTER table EMPLOYEE created in question 2 and apply the constraint on DEPTNO attribute such that on update, update a child value and on delete set null value to a child.

ALTER TABLE employee ADD FOREIGN KEY(deptno) REFERENCES department(deptno) ON UPDATE CASCADE ON DELETE SET NULL;

Lab 09 : DML Queries

1. Write a SQL statement to insert 4 rows in project table and 4 rows in student table below by a single insert statement.

```
INSERT IGNORE INTO projects
        VALUES
                (1, "AI for Marketing",    "2019-08-01", "2019-12-31"),
                (2, "ML for Sales",        "2019-05-15", "2019-11-20"),
                (3, "CS for IT",           "2020-01-01", "2020-05-20"),
                (4, "SQL for input",       "2020-06-13", "2020-11-20");
```

project_id	name	start_date	end_date
1	AI for Marketing	2019-08-01	2019-12-31
2	ML for Sales	2019-05-15	2019-11-20
3	CS for IT	2020-01-01	2020-05-20
4	SQL for input	2020-06-13	2020-11-20
NULL	NULL	NULL	NULL

```
INSERT IGNORE INTO student
        VALUES
                ("S100", "Ali Mehmood", "Administration", "ali@iba-suk.edu.pk", "0333-895311", 3),
                ("S101", "Manisha Kataria", "Computer Science", "manisha@iba-suk.edu.pk", "0345-111333444", 2),
                ("S102", "Sagar Sanjay", "Engineering", "sagar@iba-suk.edu.pk", "0300-22224454", 2),
                ("S103", "Sara Shaikh", "IT", "sara@iba-suk.edu.pk", "0300-111110000", 3);
```

std_NO	std_Name	Department	email	phone	project_id
S100	Ali Mehmood	Administration	ali@iba-suk.edu.pk	0333-895311	3
S101	Manisha Kataria	Computer Science	manisha@iba-suk.edu.pk	0345-111333444	2
S102	Sagar Sanjay	Engineering	sagar@iba-suk.edu.pk	0300-22224454	2
S103	Sara Shaikh	IT	sara@iba-suk.edu.pk	0300-111110000	3

```
CREATE TABLE IF NOT EXISTS projects_copy SELECT * FROM projects;
```

project_id	name	start_date	end_date
1	AI for Marketing	2019-08-01	2019-12-31
2	ML for Sales	2019-05-15	2019-11-20
3	CS for IT	2020-01-01	2020-05-20
4	SQL for input	2020-06-13	2020-11-20

2. Write a SQL statement to delete all records from projects_copy table.

```
DELETE FROM projects_copy ORDER BY project_id DESC LIMIT 5;
```

	project_id	name	start_date	end_date

3. Write a SQL statement to insert rows from projects_copy table to projects_copy table.

INSERT IGNORE INTO projects_copy SELECT * FROM projects;

	project_id	name	start_date	end_date
▶	1	AI for Marketing	2019-08-01	2019-12-31
	2	ML for Sales	2019-05-15	2019-11-20
	3	CS for IT	2020-01-01	2020-05-20
	4	SQL for input	2020-06-13	2020-11-20

4. Write a SQL statement to update start_date to ‘2023-02-01’ of a project name CS for IT.

UPDATE projects SET start_date = "2023-02-01" WHERE name="CS for IT";

UPDATE projects SET start_date="2023-02-01" WHERE project_id=3;

	project_id	name	start_date	end_date
▶	1	AI for Marketing	2019-08-01	2019-12-31
	2	ML for Sales	2019-05-15	2019-11-20
	3	CS for IT	2023-02-01	2020-05-20
	4	SQL for input	2020-06-13	2020-11-20

ALTER TABLE projects ADD cost INT;

ALTER TABLE projects ADD daysToComplete INT;

	project_id	name	start_date	end_date	cost	daysToComplete
▶	1	AI for Marketing	2019-08-01	2019-12-31	NULL	NULL
	2	ML for Sales	2019-05-15	2019-11-20	NULL	NULL
	3	CS for IT	2023-02-01	2020-05-20	NULL	NULL
	4	SQL for input	2020-06-13	2020-11-20	NULL	NULL

5. Write a SQL statement to update cost of project to 90000 where cost are null.

UPDATE projects SET cost=90000 WHERE cost IS NULL;

UPDATE projects SET cost=90000 ORDER BY project_id DESC LIMIT 4;

	project_id	name	start_date	end_date	cost	daysToComplete
▶	1	AI for Marketing	2019-08-01	2019-12-31	90000	NULL
	2	ML for Sales	2019-05-15	2019-11-20	90000	NULL
	3	CS for IT	2023-02-01	2020-05-20	90000	NULL
	4	SQL for input	2020-06-13	2020-11-20	90000	NULL

6. Write a SQL statement to update daysToComplete column from student by calculating difference from project start_date and end_date.

UPDATE projects SET start_date="2020-01-01" WHERE project_id=3;

UPDATE projects SET daysToComplete=DATEDIFF(end_date,start_date) LIMIT 4;

project_id	name	start_date	end_date	cost	daysToComplete
1	AI for Marketing	2019-08-01	2019-12-31	90000	152
2	ML for Sales	2019-05-15	2019-11-20	90000	189
3	CS for IT	2020-01-01	2020-05-20	90000	140
4	SQL for input	2020-06-13	2020-11-20	90000	160

Lab 10: Stored Routines (Procedures & Functions)

1. Create a stored procedure DISPLAY without parameters. The procedure must display empno, ename and salary of all the employees of DEPTNO = 10.

```
DELIMITER $$

CREATE PROCEDURE DISPLAY()

BEGIN

        SELECT employee_id as empno, CONCAT(first_name, " ", last_name) as ename, salary FROM employees WHERE department_id=10;

END$$

DELIMITER ;

CALL DISPLAY();
```

	empno	ename	salary
▶	200	Jennifer Whalen	4400.00

2. Create a stored procedure DISPLAY2 with parameters. It must take DEPTNO as an input and must return the DNAME and TOTAL SALARY of the input department number.

```
CREATE PROCEDURE DISPLAY2(IN DEPTNO INT)

BEGIN

        SELECT d.department_name as DNAME, SUM(salary) as "TOTAL SALARY" FROM employees as e, departments as d

   WHERE e.department_id=d.department_id AND d.department_id=DEPTNO;

END$$

DELIMITER ;

CALL DISPLAY2(10);
```

	DNAME	TOTAL SALARY
▶	Administration	4400.00

3. Create a stored procedure DISPLAY3 with parameters. It must take DEPTNO as an input and must return the DNAME, SMALLEST and HIGHEST SALARIES of the input department number. DISPLAY3 must also display empno,ename,total salary (sal+comm) of all the employees of the input department number.

```
DELIMITER $$

CREATE PROCEDURE DISPLAY3(IN DEPTNO INT)

BEGIN
```

```
SELECT d.department_name as DNAME, MIN(e.salary) as "SMALLEST SALARY",
MAX(e.salary) as "HIGHEST SALARY"

FROM employees as e, departments as d WHERE e.department_id=d.department_id AND
d.department_id=50;

SELECT e.employee_id as empno,  CONCAT(first_name, " ", last_name) as ename,
salary+(salary*commission_pct)  as "Total Salary"

FROM employees as e, departments as d WHERE e.department_id=d.department_id AND
d.department_id=90 AND commission_pct IS NOT NULL;

END$$

DELIMITER ;

CALL DISPLAY3(90);
```

	DNAME	SMALLEST SALARY	HIGHEST SALARY
▶	Shipping	2100.00	8200.00

	empno	ename	Total Salary

4. Create a stored function MANAGER without input parameters. It must return the total salary of all the managers in the EMP.

```
DELIMITER $$

CREATE FUNCTION MANAGER()

RETURNS NUMERIC(10,2)

DETERMINISTIC

BEGIN

    DECLARE total_salary NUMERIC(10,2) DEFAULT 0.0;

  SET total_salary = (SELECT SUM(e.salary) as "All Employees Salary" FROM employees as e,
departments as d WHERE e.manager_id=d.manager_id);

  RETURN (total_salary);

END$$

DELIMITER ;

SELECT MANAGER();
```

	MANAGER()
▶	319400.00

5. Create a stored function MANAGER2 with parameters. It must take empno as an input and must return its manager name. Write a SELECT statement to display all employees' names and their manager names. Manager names must be displayed using MANAGER2 stored function.

```
DELIMITER $$

CREATE FUNCTION MANAGER3(empno INT)

RETURNS VARCHAR(30)

DETERMINISTIC

BEGIN

        RETURN(SELECT CONCAT(e2.first_name, " ", e2.last_name) as manager_name

        FROM employees as e1, employees as e2 WHERE e1.manager_id=e2.employee_id AND
e1.employee_id=empno);

END$$

DROP FUNCTION MANAGER2;

DELIMITER ;

SELECT CONCAT(first_name, " ", last_name) as employee_name,  MANAGER2(employee_id) FROM
employees;
```

	employee_name	MANAGER2
▶	Steven King	NULL
	Neena Kochhar	Steven King
	Lex De Haan	Steven King
	Alexander Hunold	Lex De Haan
	Bruce Ernst	Alexander Hunold

6. Create a stored function MANAGER3 with parameters. It must take MANAGER NAME

as an input and must return average salary of its employees..

```
DELIMITER $$

CREATE FUNCTION MANAGER3(manager_name VARCHAR(30))

RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

        DECLARE avg_salary DECIMAL(10,2) DEFAULT 0.0;

   SET avg_salary = (SELECT AVG(e1.salary) as avg_salary FROM employees as e1, employees as e2

        WHERE e1.manager_id=e2.employee_id AND CONCAT(TRIM(e2.first_name), " ",
TRIM(e2.last_name)) = "Steven King");

   RETURN (avg_salary);
```

```
END$$

DELIMITER ;

SELECT MANAGER3("Steven King");
```

	MANAGER3("Steven King")
▶	11100.00

Lab 11: Control Structures

1. Create a procedure **countEven** that will sum the number from 2 to given particular Number passes through IN parameter using any Loop.

```
DELIMITER $$

CREATE PROCEDURE countEven(IN num INT, IN size_ INT)

BEGIN

        DECLARE str VARCHAR(255) DEFAULT "";

   DECLARE num_copy INT DEFAULT 0;

   SET str = "";

   SET num_copy = num;

   loop_label: LOOP

      IF MOD(num_copy,2)=0 THEN SET str = CONCAT(str, num_copy,",");

      ELSEIF num_copy>=size_ THEN LEAVE loop_label;

      END IF;

      SET num_copy = num_copy + 1;

   END LOOP loop_label;

   SELECT str;

END$$

DELIMITER ;

CALL countEven(1,10);
```

	str
▶	2,4,6,8,10,

2. Create a procedure **checkCustomer** that will take customerNumber through IN parameter and display Whether the customer exist in customer table or not.

```
CREATE PROCEDURE checkCustomer(IN empNo INT)

BEGIN

        DECLARE status_ VARCHAR(255) DEFAULT "";

   DECLARE temp BOOLEAN;

        SET temp = (SELECT empNo IN (SELECT employee_id FROM employees));

                IF temp=1 THEN SET status_ = "Yes, employee exists In The employees table.";

                ELSE SET status_="No, Employee does not exist in the employee table.";
```

```
    END IF;
    SELECT status_;

END$$
DELIMITER ;
CALL checkCustomer(102);
```

	status_
▶	Yes, employee exists In The employees table.

```
CALL checkCustomer(1020);
```

	status_
▶	No, Employee does not exist in the employee ta...

4. Create a procedure EvenOdd that takes number through IN parameter then check and display whether the given no is Odd or Even.

```
DELIMITER $$
CREATE PROCEDURE EvenOdd(IN num INT)
BEGIN
        DECLARE status_ VARCHAR(30) DEFAULT "";
    IF mod(num,2)=0 THEN SET status_="It is Even Number";
    ELSE SET status_="It is Odd Number";
    END IF;
    SELECT status_;
END$$
DELIMITER ;
CALL EvenOdd(10);
```

	status_
▶	It is Even Number

```
CALL EvenOdd(11);
```

	status_
▶	It is Odd Number

5. Create 3 procedures (with LOOP, WHILE, REPEAT UNTIL) to print following output.

```
+------------+
| output     |
+------------+
| 1,2,3,4,5, |
+------------+
```

--LOOP

```
DELIMITER $$

CREATE PROCEDURE LoopDemo()

BEGIN

        DECLARE x INT DEFAULT 0;

  DECLARE str VARCHAR(255) DEFAULT "";

  SET x = 1;

  loop_label: LOOP

                IF x>5 THEN LEAVE loop_label;

    END IF;

    SET str = CONCAT(str,x,",");

    SET x = x + 1;

    SELECT str as status_;

  END LOOP loop_label;

END$$

DELIMITER ;

CALL LoopDemo();
```

	status_
▶	1,2,3,4,5,

-- WHILE

```
DELIMITER $$

CREATE PROCEDURE WhileDemo()

BEGIN

        DECLARE x INT DEFAULT 0;

  DECLARE str VARCHAR(255) DEFAULT "";

  SET x = 1;

  SET str="";

  WHILE x<=5 DO
```

```
                SET str = CONCAT(str,x,",");
    SET x = x + 1;
  END WHILE;
  SELECT str as status_;
END$$
DELIMITER ;
CALL WhileDemo();
```

status_
1,2,3,4,5,

-- REPEAT UNTIl

```
DELIMITER $$
CREATE PROCEDURE RepeatUntilDemo()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 1;
  SET str="";
  REPEAT
                SET str = CONCAT(str,x,",");
    SET x = x + 1;
  UNTIL x>5 END REPEAT;
  SELECT str as status_;
END$$
DELIMITER ;
CALL RepeatUntilDemo();
```

status_
1,2,3,4,5,

6. Create 3 procedures (with LOOP, WHILE, REPEAT UNTIL) to print following output.

```
+------------+
| output     |
+------------+
| 5,4,3,2,1, |
+------------+
```

-- LOOP

```
DROP PROCEDURE  IF EXISTS LoopDemo2;
DELIMITER $$
CREATE PROCEDURE LoopDemo2()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  loop_label: LOOP
                IF(x<1) THEN LEAVE loop_label;
    END IF;
    SET str = CONCAT(str,x,",");
    SET x = x - 1;
  END LOOP loop_label;
  SELECT str as output;
END$$
DELIMITER ;
CALL LoopDemo2();
```

	output
▶	5,4,3,2,1,

-- WHILE DO

```
DELIMITER $$
CREATE PROCEDURE WhileDemo2()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  WHILE x>=1 DO
                SET str = CONCAT(str,x,",");
                SET x = x - 1;
  END WHILE;
  SELECT str as output;
END$$
DELIMITER ;
CALL WhileDemo2();

-- REPEAT UNTIL
DELIMITER $$
```

```
CREATE PROCEDURE RepeatUntilDemo2()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  REPEAT
                SET str = CONCAT(str,x,",");
    SET x = x - 1;
  UNTIL x<1
  END REPEAT;
  SELECT str as output;
END$$
DELIMITER ;
 CALL RepeatUntilDemo2();
```

	output
▶	5,4,3,2,1,

7. Create 3 procedures (with LOOP, WHILE, REPEAT UNTIL) to print following output.

```
+-------------------+
| output            |
+-------------------+
| -5,-4,-3,-2,-1,0, |
+-------------------+
```

-- LOOP

```
DELIMITER $$
CREATE PROCEDURE LoopDemo3()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  loop_label:LOOP
                IF x<0 THEN LEAVE loop_label;
    END IF;
    SET str = CONCAT(str,(-x),",");
    SET x = x - 1;
  END LOOP loop_label;
  SELECT str as output;
END$$
DELIMITER ;
CALL LoopDemo3();
```

	output
▶	-5,-4,-3,-2,-1,0,

-- WHILE

```
DELIMITER $$
CREATE PROCEDURE WhileDemo3()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  WHILE x>=0 DO
                SET str = CONCAT(str,x,",");
                SET x = x - 1;
  END WHILE;
  SELECT str as output;
END$$
DELIMITER ;
CALL WhileDemo3();
```

	output
▶	-5,-4,-3,-2,-1,0,

```
-- REPEAT UNTIL
DELIMITER $$
CREATE PROCEDURE RepeatUntilDemo3()
BEGIN
        DECLARE x INT DEFAULT 0;
  DECLARE str VARCHAR(255) DEFAULT "";
  SET x = 5;
  SET str="";
  REPEAT
                SET str = CONCAT(str,(-x),",");
    SET x = x - 1;
  UNTIL x<0 END REPEAT;
  SELECT str as output;
END$$
DELIMITER ;
CALL RepeatUntilDemo3();
```

	output
▶	-5,-4,-3,-2,-1,0,