Name: Hitendra Sisodia
Sap id: 500091910

Lab 7

Q1- Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id

```
mysql> select * from employee;
+-------------+------------+------------+------------+--------+---------------+
| Employee_id | First_name | Last_name  | DOB        | Salary | Department_id |
+-------------+------------+------------+------------+--------+---------------+
| E1          | Eishan     | Pandey     | 2002-02-21 | 300000 | D1            |
| E2          | Priyanshu  | Tandon     | 2005-04-20 | 400000 | D2            |
| E3          | Aman       | Kumar      | 2006-06-25 | 100000 | D3            |
| E4          | Anirudh    | Srinivasan | 2001-05-12 | 700000 | D4            |
| E5          | Ujesh      | Sisodia    | 2008-09-26 | 800000 | D5            |
+-------------+------------+------------+------------+--------+---------------+
5 rows in set (0.00 sec)

mysql> CREATE INDEX employee_idx ON EMPLOYEES(Last_name,Department_id);
ERROR 1146 (42S02): Table 'ui.employees' doesn't exist
mysql> CREATE INDEX employee_idx ON employee(Last_name,Department_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q2- Find the ROWID for the above table and create a unique index on employee_id column of the EMPLOYEES.

```
mysql> CREATE UNIQUE INDEX employee_id_idx ON employee(Employee_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q3- Create a reverse index on employee_id column of the EMPLOYEES.

```
mysql> CREATE INDEX idx_employee_id ON employee(Employee_id DESC);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q4- Create a unique and composite index on employee_id and check whether there is duplicity of tuples or not.

```
mysql> CREATE UNIQUE INDEX idx2_employee_id ON employee(Employee_id);
Query OK, 0 rows affected, 1 warning (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> select Employee_id from employee group by Employee_id having count(*)>1;
```

Q5- Create Function-based indexes defined on the SQL functions UPPER(column_name) or LOWER(column_name) to facilitate case-insensitive searches(on column Last_Name).

```
mysql> select * from employee where upper(Last_name)=upper('Tandon');
+-------------+------------+-----------+------------+--------+---------------+
| Employee_id | First_name | Last_name | DOB        | Salary | Department_id |
+-------------+------------+-----------+------------+--------+---------------+
| E2          | Priyanshu  | Tandon    | 2005-04-20 | 400000 | D2            |
+-------------+------------+-----------+------------+--------+---------------+
1 row in set (0.01 sec)

mysql> select * from employee where lower(Last_name)=lower('Pandey');
+-------------+------------+-----------+------------+--------+---------------+
| Employee_id | First_name | Last_name | DOB        | Salary | Department_id |
+-------------+------------+-----------+------------+--------+---------------+
| E1          | Eishan     | Pandey    | 2002-02-21 | 300000 | D1            |
+-------------+------------+-----------+------------+--------+---------------+
```

Name: Hitendra Sisodia
Sap id: 500091910

Lab 7

Q6- Drop the function based index on column Last_Name.

```
mysql> DROP INDEX idx2_employee_id ON employee;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```