Experiment – 11

```
create database exp11;

use exp11;
create table employees (
   employee_id char(10) primary key,
   first_name char(30) not null,
   last_name char(30) not null,
   dob date,
   salary decimal(25,2) not null,
   department_id char(10)
);


insert into employees (employee_id, first_name, last_name, dob, salary, department_id)
values
('E001', 'John', 'Doe', '1985-06-15', 55000.00, 'D001'),
('E002', 'Jane', 'Smith', '1990-04-20', 62000.00, 'D002'),
('E003', 'Robert', 'Johnson', '1982-12-05', 75000.00, 'D003'),
('E004', 'Emily', 'Davis', '1995-11-11', 50000.00, 'D001'),
('E005', 'Michael', 'Brown', '1988-09-22', 67000.00, 'D004'),
('E006', 'Sarah', 'Miller', '1992-02-17', 72000.00, 'D003');
```

```
mysql> select * from employees;
+-------------+------------+-----------+------------+----------+---------------+
| employee_id | first_name | last_name | dob        | salary   | department_id |
+-------------+------------+-----------+------------+----------+---------------+
| E001        | John       | Doe       | 1985-06-15 | 55000.00 | D001          |
| E002        | Jane       | Smith     | 1990-04-20 | 62000.00 | D002          |
| E003        | Robert     | Johnson   | 1982-12-05 | 75000.00 | D003          |
| E004        | Emily      | Davis     | 1995-11-11 | 50000.00 | D001          |
| E005        | Michael    | Brown     | 1988-09-22 | 67000.00 | D004          |
| E006        | Sarah      | Miller    | 1992-02-17 | 72000.00 | D003          |
+-------------+------------+-----------+------------+----------+---------------+
6 rows in set (0.00 sec)
```

1. Execute the following index related queries:

1. Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id

```
create index employee_idx on employees (last_name, department_id);
```

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

```
create unique index employee_id_unique_idx on employees (employee_id);
```

3. Create a reverse index on employee_id column of the EMPLOYEES.

```
create index employee_id_reverse_idx  on employees (employee_id);
```

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

```
create unique index employee_id_composite_unique_idx  on employees (employee_id);
```

5. Create Function-based indexes defined on the SQL functions

UPPER(column_name) or LOWER(column_name) to facilitate caseinsensitive

searches(on column Last_Name).

```
create index last_name_upper_idx on employees (upper(last_name));
```

6. Drop the function based index on column Last_Name.

```
drop index last_name_upper_idx on employees;
```