

ASSIGNMENT - 1

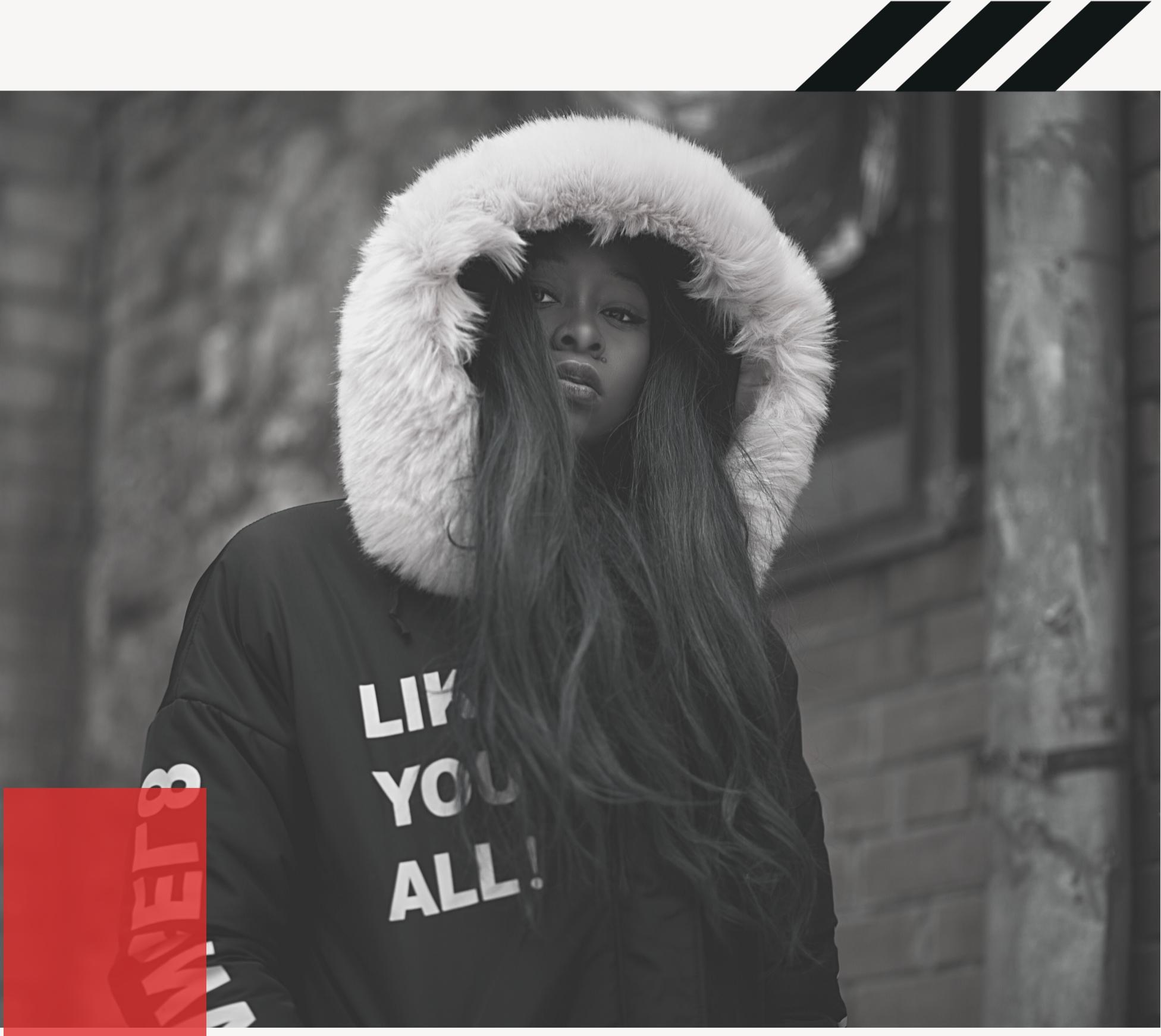
Study of SQL commands:

DDL, DML, DCL

**VIDUSHI YADAV
AU20B1013**

PROBLEM STATEMENT

In this assignment, students will be using the MySQL database for storing their data. Students will be provided with the database schema which they will have to design by creating the required tables in the MySQL database.





TASK TO BE PERFORMED

The tasks that you will accomplish in this assignment are as follows:

1. Create a database using DDL statements in the MySQL command-line tool.
2. Populate the database row-by-row from the provided raw data command line terminal of MySQL..
3. Write SQL queries to the database to be executed from the MySQL command-line tool.
4. Query and manipulate the database programmatically using the command line interface.
5. Generate reports based on some basic analysis.

ABOUT

The assignment will be divided into different part as follows:

Part 1: Creating the database

Identify the keys and constraints for your system and design a database schema containing required tables having columns and their attributes.

Part 2: Populating the database

All the tables in the database should be populated using the SQL DDL statement. Students are expected to make extensive use of INSERT, UPDATE, DELETE, and ALTER commands of the SQL or Download the provided dataset and populate your database using respective commands.

Part 3: SQL queries on the database

Write the queries in SQL and run them on your MySQL database via the MySQL command-line tool. Prepare and submit them as separate files.



Database Schema

Schema Name: **Company**

Employee (Fname, Mname, Lname, Ssn, Bdate, Address, Gender, Salary, Super_ssn, Dno)

Department (Dname, DNumber, Mgr_SSN, Mgr_start_date)

Dept_Locations (Dnumber, Dlocation)

Project (Pname, Pnumber, Plocation, Dnum)

Works_ON (Essn, Pno, Hours)

Dependent (Essn, Dependent_name, Gender, Bdate, Relation)

Table Creation Commands

```
1 • CREATE DATABASE Company;
2 • USE Company;
3
4 • CREATE TABLE DEPARTMENT (
5     dname      varchar(50) not null,
6     dnumber    int not null,
7     mgrssn     char(9) not null,
8     mgrstartdate date,
9     primary key (dnumber),
10    UNIQUE (dname)
11 );
12
13 • CREATE TABLE EMPLOYEE (
14     fname      varchar(20) not null,
15     minit     varchar(1),
16     lname      varchar(17) not null,
17     ssn       char(9),
18     bdate     date,
```

```
17      ssn      char(9),
18      bdate    date,
19      address  varchar(50),
20      sex      char,
21      salary   decimal(10,2),
22      superssn char(9),
23      dno      int,
24      primary key (ssn),
25      foreign key (dno) references DEPARTMENT(dnumber)
26 );
27
28 • ⏺ CREATE TABLE DEPENDENT (
29      essn      char(9),
30      dependent_name varchar(15),
31      sex      char,
32      bdate    date,
33      relationship  varchar(8),
34      primary key (essn,dependent_name),
35      foreign key (essn) references EMPLOYEE(ssn)
36 );
37
38 • ⏺ CREATE TABLE DEPT_LOCATIONS (
39      dnumber  int,
40      dlocation varchar(15),
41      primary key (dnumber,dlocation),
42      foreign key (dnumber) references DEPARTMENT(dnumber)
43 );
44
45 • ⏺ CREATE TABLE PROJECT (
46      pname    varchar(25) not null,
47      pnumber  int,
48      plocation varchar(15),
49      dnum     int not null,
50      primary key (pnumber),
```

```
51      foreign key (dnum) references DEPARTMENT(dnumber),
52      foreign key (plocation) references LOCATIONS(location)
53 );
54
55 • ⏺ CREATE TABLE WORKS_ON (
56      empno   int not null,
57      pnumber int not null,
58      hours   int,
59      grade   char(2),
60      primary key (empno,pnumber),
61      foreign key (empno) references EMPLOYEE(empno),
62      foreign key (pnumber) references PROJECT(pnumber)
63 );
64
65 • ⏺ CREATE TABLE TAKES (
66      empno   int not null,
67      cwid    int not null,
68      grade   char(2),
69      primary key (empno,cwid),
70      foreign key (empno) references EMPLOYEE(empno),
71      foreign key (cwid) references COURSES(cwid)
72 );
73
74 • ⏺ CREATE TABLE GRADE (
75      cwid    int not null,
76      empno  int not null,
77      grade   char(2),
78      primary key (cwid,empno),
79      foreign key (cwid) references COURSES(cwid),
80      foreign key (empno) references EMPLOYEE(empno)
81 );
82
83 • ⏺ CREATE TABLE EMPLOYEE (
84      empno  int not null,
85      ssn    char(9),
86      bdate  date,
87      address  varchar(50),
88      sex      char,
89      salary   decimal(10,2),
90      manager  int,
91      department  int,
92      primary key (empno),
93      foreign key (manager) references EMPLOYEE(empno),
94      foreign key (department) references DEPARTMENT(dnumber)
95 );
96
97 • ⏺ CREATE TABLE DEPARTMENT (
98      dnumber  int not null,
99      dname   varchar(15),
100     location  varchar(15),
101     mngno   int,
102     budget   decimal(10,2),
103     primary key (dnumber),
104     foreign key (mngno) references EMPLOYEE(empno)
105 );
106
107 • ⏺ CREATE TABLE LOCATIONS (
108      location  varchar(15),
109     city     varchar(15),
110     state   varchar(2),
111     country  varchar(15),
112     zip      int,
113     primary key (location)
114 );
115
116 • ⏺ CREATE TABLE COURSES (
117      cwid    int not null,
118     cname   varchar(15),
119     credit  int,
120     primary key (cwid)
121 );
```

```

46     pname      varchar(25) not null,
47     pnumber    int,
48     plocation  varchar(15),
49     dnum       int not null,
50     primary key (pnumber),
51     unique (pname),
52     foreign key (dnum) references DEPARTMENT(dnumber)
53 );
54
55 • ⊖ CREATE TABLE WORKS_ON (
56     essn      char(9),
57     pno       int,
58     hours     decimal(4,1),
59     primary key (essn,pno),
60     foreign key (essn) references EMPLOYEE(ssn),
61     foreign key (pno) references PROJECT(pnumber)
62 );

```

Data Insertion Commands

```

INSERT INTO DEPENDENT VALUES ('978653421','Justin','M','2000-11-12','Son');
INSERT INTO DEPENDENT VALUES ('968574312','Devil','M','1984-07-18','Brother');
INSERT INTO DEPENDENT VALUES ('986754231','Marylin','F','1950-10-14','Wife');
INSERT INTO DEPENDENT VALUES ('897654132','Kendrick','M','1969-03-14','Brother');
INSERT INTO DEPENDENT VALUES ('695489721','Shakespear','M','1994-01-05','Son');

INSERT INTO DEPARTMENT VALUES ('Quality','1','978653421','1979-05-03');
INSERT INTO DEPARTMENT VALUES ('HR','2','968574312','1984-07-17');
INSERT INTO DEPARTMENT VALUES ('Sales','3','986754231','1970-10-25');
INSERT INTO DEPARTMENT VALUES ('Management','4','897654132','1971-08-29');
INSERT INTO DEPARTMENT VALUES ('Accounting','5','695489721','1976-03-11');

INSERT INTO DEPT_LOCATIONS VALUES ('1','Pune');
INSERT INTO DEPT_LOCATIONS VALUES ('2','Alandi');
INSERT INTO DEPT_LOCATIONS VALUES ('3','Scranton');
INSERT INTO DEPT_LOCATIONS VALUES ('4','Stanford');
INSERT INTO DEPT_LOCATIONS VALUES ('5','Indore');

INSERT INTO EMPLOYEE VALUES ('John','L','Bratton','978653421','1954-02-13','Pune','M','30000',NULL,'1' );
INSERT INTO EMPLOYEE VALUES ('Angela','A','Smith','695489721','1961-05-18','Alandi','F','60000',NULL,'5' );
INSERT INTO EMPLOYEE VALUES ('Jim','C','Halpert','986754231','1947-01-22','Alandi','M','90000',NULL,'3' );
INSERT INTO EMPLOYEE VALUES ('Michael','B','Scott','897654132','1957-11-12','Pune','M','140000',NULL,'4' );
INSERT INTO EMPLOYEE VALUES ('Toby','K','Anderson','968574312','1965-12-11','Scranton','M','45000',NULL,'2' );

```

```

INSERT INTO WORKS_ON VALUES ('978653421','2','10');
INSERT INTO WORKS_ON VALUES ('968574312','3','2');
INSERT INTO WORKS_ON VALUES ('986754231','4','23');
INSERT INTO WORKS_ON VALUES ('897654132','1','14');
INSERT INTO WORKS_ON VALUES ('695489721','5','18');

```

Data Manipulation Commands

```

/* TASK 2 */
select max(salary) as "Maximum Salary", min(salary) as "Minimum Salary", avg(salary) as "Average Salary", sum(salary) "Sum of the Salaries" from employee; /* Question-1 */
select count(*) from employee where dno = ( select dnumber from department where dname = "Analysis" ); /* Question-2 */
select dno, count(*), avg(salary) from employee group by dno; /* Question-3 */
select dname, avg(salary) from department, employee where dnumber=dno group by dname; /* Question-4 */
select d.dname from employee inner join department d on dno = dnumber group by d.dname having count(*) > 2; /* Question-5 */
select p.pnumber , p.pname, count(wo.essn) from project p inner join works_on wo on p.pnumber = wo.pno where wo.essn in
( select essn from works_on group by essn having count(essn) > 2 ) group by p.pnumber; /* Question-6 */

select * from employee order by salary desc; /* Question-7 */
select * from employee order by fname asc; /* Question-8 */
select fname, dayname(bdate) from employee; /* Question-9 */
select year(bdate) from employee; /* Question-10 */
select year(current_date()) - year(mgrstartdate) from department; /* Question-11 */
select fname, year(current_date())- year(bdate) as "Age" from employee; /* Question-12 */

/* TASK 1 */
select bdate, address from employee where fname = "John"; /* Question-1 */
select fname, lname, address from employee where dno = ( select dnumber from department where dname="Quality" ); /* Question-2 */
select concat(e.fname, " ", e.lname) as "Employee Name", concat(s.fname, " ", s.lname) as "Supervisor Name" from employee e, employee s where s.ssn=e.superssn; /* Question-3 */
select distinct(salary) from employee; /* Question-4 */
select * from employee where address like "%Alandi%"; /* Question-5 */
select * from employee where bdate like "195%"; /* Question-6 */
select fname, lname from employee where superssn is null; /* Question-7 */
select distinct e.salary + (0.1*e.salary) as "Increased Salary", e.fname, e.lname from employee e, works_on wo
where wo.pno = ( select pnumber from project where pname = "C" ); /* Question-8 */

select * from employee where dno = 5 and salary >= 50000 and salary <= 80000; /* Question-9 */
select * from employee where fname like "P%"; /* Question-10 */
select * from employee where address like "%Alandi%" or address like "%Pune%"; /* Question-11 */
select concat(fname, " ", lname) as "Fullname" from employee; /* Question-12 */

/* TASK 3 */
(select distinct pf.pnumber from project pf, department d, employee ef where ef.dno = d.dnumber and d.mgrssn = ef.ssn and ef.lname = "Smith") union
(select distinct pnumber from project, works_on, employee where pnumber = pno and essn = ssn and lname = "Smith"); /* Question-1 */

select e.* from employee e, dependent d where e.fname = d.dependent_name and e.sex = d.sex; /* Question-2 */
select fname, lname, salary from employee where salary > ( select max(salary) from employee where dno = 4 ); /* Question-3 */
select distinct essn from works_on where pno in (1,2,3); /* Question-4 */
select * from employee where not exists ( select * from dependent where ssn = essn ); /* Question-5 */
select * from employee where exists ( select * from department where ssn = mgrssn ) and exists ( select * from dependent where ssn = essn ); /* Question-6 */

```

Database Schema

Schema Name: **Bank**

Branch(branch_name, branch_city, assets)

Customer(customer_name, customer_street, customer_city)

Account(account_number, branch_name, balance)

Depositor(customer_name, account_number)

Loan(loan_number, branch_name, amount)

Borrower(customer_name, loan_number)

Table Creation Commands

```
CREATE TABLE IF NOT EXISTS BORROWER
(
CUSTOMER_NAME VARCHAR(250) NOT NULL,
LOAN_NUMBER INT NOT NULL,
PRIMARY KEY(CUSTOMER_NAME, LOAN_NUMBER),
FOREIGN KEY (CUSTOMER_NAME) REFERENCES CUSTOMER(CUSTOMER_NAME),
FOREIGN KEY (LOAN_NUMBER) REFERENCES LOAN(LOAN_NUMBER)
);
```

```
CREATE TABLE IF NOT EXISTS ACCOUNT
(
ACCOUNT_NUMBER INT NOT NULL,
BRANCH_NAME VARCHAR(250),
BALANCE DOUBLE,
PRIMARY KEY (ACCOUNT_NUMBER),
FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME)
);
```

```
CREATE TABLE IF NOT EXISTS DEPOSITOR
(
CUSTOMER_NAME VARCHAR(250) NOT NULL,
ACCOUNT_NUMBER INT NOT NULL,
PRIMARY KEY(CUSTOMER_NAME, ACCOUNT_NUMBER),
FOREIGN KEY (CUSTOMER_NAME) REFERENCES CUSTOMER(CUSTOMER_NAME),
FOREIGN KEY (ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER)
);
```

```
CREATE DATABASE BANK;
```

```
USE BANK;
```

```
CREATE TABLE IF NOT EXISTS BRANCH
```

```
(  
    BRANCH_NAME VARCHAR(250) NOT NULL,  
    BRANCH_CITY VARCHAR(250),  
    ASSETS DOUBLE,  
    PRIMARY KEY (BRANCH_NAME)  
) ;
```

```
CREATE TABLE IF NOT EXISTS LOAN
```

```
(  
    LOAN_NUMBER INT NOT NULL,  
    BRANCH_NAME VARCHAR(250),  
    AMOUNT DOUBLE,  
    PRIMARY KEY (LOAN_NUMBER),  
    FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME)  
) ;
```

Data Insertion Commands

```
insert into ACCOUNT values('101', 'SapnaSangeeta', 25000);  
insert into ACCOUNT values('102', 'VijayNagar', 25);  
insert into ACCOUNT values('103', 'Nanakheda', 10000);  
insert into ACCOUNT values('104', 'SiliconCity', 100000);  
insert into ACCOUNT values('105', 'Freeganj', 22000);
```

```
insert into CUSTOMER values('Bhuvan', 'MGRoad', 'Indore');  
insert into CUSTOMER values('Shankar', 'Freeganj', 'Ujjain');  
insert into CUSTOMER values('Zakir', 'Nanakheda', 'Ujjain');  
insert into CUSTOMER values('Ajay', 'Navlakha', 'Indore');  
insert into CUSTOMER values('Vipul', 'GPO', 'Indore');
```

```
insert into DEPOSITOR values('Bhuvan', '101');  
insert into DEPOSITOR values('Zakir', '103');  
insert into DEPOSITOR values('Vipul', '104');
```

```
insert into BRANCH values('SapnaSangeeta', 'Indore', '9000000');  
insert into BRANCH values('VijayNagar', 'Indore', '8000000');  
insert into BRANCH values('Nanakheda', 'Ujjain', '100000');  
insert into BRANCH values('SiliconCity', 'Indore', '25000');  
insert into BRANCH values('Freeganj', 'Ujjain', '8000000');
```

```
insert into BORROWER values('Bhuvan', '203')  
insert into BORROWER values('Zakir', '205');
```

Data Manipulation Commands

```
/*task-1*/  
  
delete from customer where customer_city in('Mumbai');/*Question-01*/  
update branch set assets = 0.1*assets where branch_name='Alandi';/*Question-02*/  
select distinct branch_name from loan; /*Question-03*/  
select loan_number from loan where branch_name = 2; /*Question-04*/  
select loan_number from loan where amount between 8 and 10; /*Question-05*/  
select customer_name from customer where customer_city like "%Road%"; /*Question-06*/  
SELECT UPPER(customer_name) FROM customer; /*Question-07*/  
SELECT LOWER(customer_name) FROM customer; /*Question-08*/  
select * from customer where customer_name like 'S%'; /*Question-09*/  
select * from customer where customer_name like '%h'; /*Question-10*/  
SELECT * FROM customer WHERE customer_name like '_a%' ; /*Question-11*/  
select * from branch where branch_city='Pune' and assets>'25'; /*Question-12*/  
select * from branch where branch_city='Pune' or branch_city='Mumbai'; /*Question-13*/  
update account set Balance= 0.05*Balance where branch_name='2'; /*Question-14*/  
SELECT CEIL(10) from account where Balance='10'; /*Question-15*/  
  
/*task-2*/  
  
select avg(Balance) from account where branch_name='2'; /*Question-01*/  
select count(customer_name) from depositor; /*Question-02*/  
select avg(Balance),branch_name from account where Balance > 1200 group by branch_name; /*Question-03*/  
select loan_number from loan where amount is null; /*Question-04*/  
select avg(balance),branch_name from account natural join depositor group by branch_name; /*Question-05*/  
select branch_name from account group by branch_name having avg(balance) >= all( select avg(balance) from account having branch_name); /*Question-06*/  
select distinct customer_name from borrower, loan where borrower.loan_number = loan.loan_number and branch_name = "2" order by customer_name asc; /*Question-07*/  
select * from loan order by Amount desc; /*Question-08*/  
  
/*task-3*/  
  
(select customer_name from borrower)union(select customer_name from depositor); /*Question-01*/  
select distinct customer_name from borrower,loan where borrower.loan_number=loan.loan_number and branch_name='2' and (branch_name, customer_name)  
in(select branch_name, customer_name from depositor,account where depositor.customer_name=account.customer_name); /*Question-02*/  
  
select loan_number from loan natural join borrower where loan_number not in (select customer_name from account natural join depositor); /*Question-03*/  
select branch_name from branch where assets > some(select assets from branch having branch_city='Alandi'); /*Question-04*/  
select branch_name from branch where assets > some(select assets from branch where branch_city= 'Alandi'); /*Question-05*/  
Select customer_name from depositor union all select customer_name from borrower; /*Question-07*/  
select branch_name from account group by branch_name having max(Balance); /*Question-06*/  
select distinct customer_name from borrower as b1 where exists (select customer_name from depositor as b2 where b1.customer_name = b2.customer_name); /*Question-07*/
```