Create the following tables & insert the records in the table.

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

1. **Find the names of all branches in the *loan* relation**
2. **Find all loan number for loans made at the Perryridge branch with loan amounts greater than $1200.**
3. **Find all customers who have a loan, an account, or both**
4. **Find the number of tuples in the *customer* relation.**
5. **Delete all account tuples at the Perryridge branch**
6. **Modify the records of customer relation for the customer_city from Mumbai to Calcutta.**
7. **Drop all the tables from the database.**

```
mysql> select branch_name from loan;
+-------------+
| branch_name |
+-------------+
| Perryridge  |
| Perryridge  |
| xyz         |
| abc         |
| xyz         |
+-------------+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> select loan_number from loan
    -> where branch_name="Perryridge" and amount>1200;
+-------------+
| loan_number |
+-------------+
|        1234 |
|        5678 |
+-------------+
2 rows in set (0.00 sec)
```

```
mysql> select customer_name from depositor
    -> union
    -> select customer_name from borrower;
+---------------+
| customer_name |
+---------------+
| aksh          |
| sakshi        |
| dhruv         |
+---------------+
3 rows in set (0.00 sec)
```

```
mysql> select count(*) from customer;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)
```

```
mysql> delete from account
    -> where branch_name="Perryridge";
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> update customer
    -> set customer_city="Calcutta"
    -> where customer_city="Mumbai";
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

```
mysql> drop table customer,depositr;
ERROR 1051 (42S02): Unknown table 'djsce.depositr'
mysql>
```

B) Write a query in sql to create a table employee and department.

Employee(empno,ename,deptno,job,hiredate)

Department(deptno,dname,loc)

1. Include the following constraints on column of emp table.

a) to make the empno as primary key of the employee table and deptno as a primary key of department table

2) Perform the following complex queries

a) List dept no., Dept name for all the departments in which there are no employees in the department.

b) Count of number of employees in department wise.

```
mysql> create table employee (
    -> empno int,
    -> ename varchar(20),
    -> deptno int,
    -> job varchar(20),
    -> hiredate date
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table department (
    -> deptno int,
    -> dname varchar(20),
    -> loc varchar(20)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> insert into department values(101, "Comps", "Mumbai");
Query OK, 1 row affected (0.02 sec)

mysql> insert into department values(102, "IT", "Thane");
Query OK, 1 row affected (0.00 sec)

mysql> insert into department values(102, "AIML", "Andheri");
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(1, "aksh", "101", "SE", '2022-10-12');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to u
mysql> insert into employee values(1, "aksh", "101", "SE", '2022-10-12');
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(2, "saksh", "101", "SE", '2022-11-12');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(3, "dhruv", "102", "SE", '2022-9-12');
Query OK, 1 row affected (0.00 sec)

mysql> insert into employee values(4, "parth", "102", "ISE", '2022-9-11');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> alter table employee add primary key(empno);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> alter table department add primary key(deptno);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> select deptno, dname
    -> from department
    -> where deptno not in(select deptno from employee);
+--------+-------+
| deptno | dname |
+--------+-------+
|    103 | AIML  |
+--------+-------+
1 row in set (0.00 sec)
```

```
mysql> select deptno, count(*)
    -> from employee
    -> group by deptno;
+--------+----------+
| deptno | count(*) |
+--------+----------+
|    101 |        2 |
|    102 |        2 |
+--------+----------+
2 rows in set (0.00 sec)
```

A. **Create the following tables & insert the records in the table.**

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

1. Find the loan number of those loans with loan amounts between $90,000 and $100,000 (that is, ≥ $90,000 and ≤ $100,000)
2. Find the name, loan number and loan amount of all customers having a loan at the Perryridge branch; rename the column name *loan_number* as *loan_id.*
3. Find the names of all customers whose street includes the substring “Main”.
4. List in alphabetic order the names of all customers having a loan in Perryridge branch
5. Find all customers who have an account but no loan.
6. Find the average account balance at the Perryridge branch.
7. Drop all the tables from the database.

```
mysql> create table branch (
    -> branch_name varchar(20),
    -> branch_city varchar(20),
    -> assets int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table customer (
    -> customer_name varchar(20),
    -> customer_street varchar(20),
    -> customer_city varchar(20)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table loan (
    -> loan_number int,
    -> branch_name varchar(20),
    -> amount  int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table borrower (
    -> customer_name varchar(20),
    -> loan_number int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table account (
    -> account_number int,
    -> branch_name varchar(20),
    -> balance int
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table depositor (
    -> customer_name varchar(20),
    -> account_number int
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> insert into branch values("Perryridge", "Mumbai", 1000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into branch values("xyz", "Chennai", 2000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into branch values("pqr", "Jaipur", 3000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer (
    -> ^C
mysql> insert into customer values("aksh", "xyz main", "mumbai");
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values("sakshi", "abc main", "mumbai");
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values("dhruv, "abc", "jaipur");
    "> ;
    "> ^C
mysql> insert into customer values("dhruv, "abc", "jaipur");
    "> ^C
mysql> insert into customer values("dhruv", "abc", "jaipur");
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values("parth", "mno", "jaipur");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into loan values(1234, "Perryridge", 80000);
Query OK, 1 row affected (0.02 sec)

mysql> insert into loan values(5678, "Perryridge", 95000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into loan values(9876, "xyz", 96000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into loan values(9877, "abc", 100000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrower values("aksh", 1234);
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrower values("sakshi", 5678);
Query OK, 1 row affected (0.01 sec)

mysql> insert into borrower values("dhruv", 9876);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into account values(101, "Perryridge", 10000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into account values(102, "xyz", 20000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into account values(103, "abc", 30000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into depositor values("aksh", 101);
Query OK, 1 row affected (0.01 sec)

mysql> insert into depositor values("sakshi", 102);
Query OK, 1 row affected (0.00 sec)

mysql> insert into depositor values("dhruv", 103);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select loan_number
    -> from loan
    -> where amount between 90000 and 100000
    -> ;
+-------------+
| loan_number |
+-------------+
|        5678 |
|        9876 |
|        9877 |
+-------------+
3 rows in set (0.00 sec)
```

```
mysql> select customer_name from customer
    -> where customer_street like '%main%';
+---------------+
| customer_name |
+---------------+
| aksh          |
| sakshi        |
+---------------+
2 rows in set (0.00 sec)

mysql> select * from customer;
+---------------+-----------------+---------------+
| customer_name | customer_street | customer_city |
+---------------+-----------------+---------------+
| aksh          | xyz main        | mumbai        |
| sakshi        | abc main        | mumbai        |
| dhruv         | abc             | jaipur        |
| parth         | mno             | jaipur        |
+---------------+-----------------+---------------+
4 rows in set (0.00 sec)
```

```
mysql> select borrower.customer_name, borrower.loan_number as loan_id, loan.loan_number
    -> from borrower, loan
    -> where borrower.loan_number = loan.loan_number
    -> and loan.branch_name = "Perryridge";
+---------------+---------+-------------+
| customer_name | loan_id | loan_number |
+---------------+---------+-------------+
| aksh          |    1234 |        1234 |
| sakshi        |    5678 |        5678 |
+---------------+---------+-------------+
2 rows in set (0.00 sec)
```

```
mysql> select distinct customer_name
    -> from borrower, loan
    -> where borrower.loan_number = loan.loan_number and branch_
name="Perryridge"
    -> order by customer_name;
+---------------+
| customer_name |
+---------------+
| Parth         |
| Sakshi        |
+---------------+
2 rows in set (0.00 sec)
```

```
mysql> select customer_name from depositor
    -> where customer_name not in(select customer_name from borrower);
Empty set (0.00 sec)
```

```
mysql> select avg(balance) from account
    -> where branch_name = "Perryridge";
+--------------+
| avg(balance) |
+--------------+
|   10000.0000 |
+--------------+
1 row in set (0.00 sec)

mysql> drop table branch, customer, loan, borrower, account, depositorD
```

A. **Create the following tables & insert the records in the table.**

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

1. Find the number of depositors in the bank.
2. Find the names of all branches where the average account balance is more than $1,200.
3. Find all customers who have both an account and a loan at the bank.
4. Find the names of all branches that have greater assets than all branches located in Brooklyn.
5. Delete all accounts at every branch located in the city 'Needham'.
6. Increase all accounts with balances over $10,000 by 6%, all other accounts receive 5%.
7. Drop all the tables from the database.

```
mysql> select count(distinct customer_name) from depositor;
+--------------------------------+
| count(distinct customer_name) |
+--------------------------------+
|                              3 |
+--------------------------------+
1 row in set (0.00 sec)
```

```
mysql> select branch_name, avg(balance)
    -> from account
    -> group by branch_name
    -> having avg(balance)>1200;
+-------------+--------------+
| branch_name | avg(balance) |
+-------------+--------------+
| Perryridge  |   10000.0000 |
| xyz         |   20000.0000 |
| abc         |   30000.0000 |
+-------------+--------------+
3 rows in set (0.00 sec)
```

```
mysql> select distinct customer_name
    -> from borrower
    -> where customer_name in(select customer_name from depositor);
+---------------+
| customer_name |
+---------------+
| aksh          |
| sakshi        |
| dhruv         |
+---------------+
3 rows in set (0.00 sec)
```

```
mysql> select branch_name
    -> from branch
    -> where assets > all
    -> ( select assets from branch where branch_name="Brooklyn");
+-------------+
| branch_name |
+-------------+
| Perryridge  |
| xyz         |
| pqr         |
+-------------+
3 rows in set (0.00 sec)
```

```
mysql> delete from account where branch_name in
    -> (select branch_name from branch where branch_city="Needham");
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> update account
    -> set balance = balance + (balance * 0.06)
    -> where balance > 10000;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from account;
+----------------+-------------+---------+
| account_number | branch_name | balance |
+----------------+-------------+---------+
|            101 | Perryridge  |   10000 |
|            102 | xyz         |   21200 |
|            103 | abc         |   84800 |
+----------------+-------------+---------+
3 rows in set (0.00 sec)

mysql> update account
    -> set balance = balance + (balance * 0.05)
    -> where balance < 10000;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

B) Write a query in sql to create a table employee and department.

Employee(empno,ename,deptno,job,hiredate)

Department(deptno,dname,loc)

1. Include the following constraints on column of emp table.

a) to make the empno as primary key of the employee table and deptno as a primary key of department table

b) Select all records where ename starts with ‘S’ and its length is 6 char.

c) Select all records where ename may be any no of character but it should end with ‘R’.

d) Create a new user “abc” and give all privileges to it.

```
mysql> alter table employee add primary key(empno);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> alter table department add primary key(deptno);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> select * from employee
    -> where ename like 's_____';
+-------+--------+--------+------+------------+
| empno | ename  | deptno | job  | hiredate   |
+-------+--------+--------+------+------------+
|     2 | sakshi |    101 | SE   | 2022-11-12 |
+-------+--------+--------+------+------------+
1 row in set (0.00 sec)

mysql> select * from employee
    -> where ename like '%r';
Empty set (0.00 sec)

mysql> create user 'abc'@'localhost' identified by 'password';
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> grant all privileges on djsce to 'abc'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

A. Create the following tables & insert the records in the table.

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)

1. Find the number of depositors for each branch.
2. Find all customers who have a loan at the bank but do not have an account at the bank
3. Delete the record of all accounts with balances below the average at the bank.
4. Create a view of all loan data in the *loan* relation, hiding the *amount* attribute
5. Add a new tuple to *branch_loan*
6. Delete the view *branch_loan* from the database
7. Drop all the tables from the database.

```
mysql> select count(customer_name)
    -> from depositor, account
    -> where depositor.account_number=account.account_number
    -> group by branch_name;
+----------------------+
| count(customer_name) |
+----------------------+
|                    1 |
|                    1 |
|                    1 |
+----------------------+
3 rows in set (0.00 sec)
```

```
mysql> select customer_name
    -> from borrower
    -> where customer_name not in
    -> (select customer_name from depositor);
Empty set (0.00 sec)
```

```
mysql> delete from account
    -> where balance <
    -> (select avg(balance) from
    -> (select balance from account)
    -> as bal);
Query OK, 2 rows affected (0.01 sec)

mysql> select * from account;
+----------------+-------------+---------+
| account_number | branch_name | balance |
+----------------+-------------+---------+
|            103 | abc         |   84800 |
+----------------+-------------+---------+
1 row in set (0.00 sec)
```

```
mysql> create view branch_loan as
    -> select loan_number, branch_name
    -> from loan;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from branch_loan;
+-------------+-------------+
| loan_number | branch_name |
+-------------+-------------+
|        1234 | Perryridge  |
|        5678 | Perryridge  |
|        9876 | xyz         |
|        9877 | abc         |
+-------------+-------------+
4 rows in set (0.00 sec)
```

```
mysql> select * from branch_loan;
+-------------+-------------+
| loan_number | branch_name |
+-------------+-------------+
|        1234 | Perryridge  |
|        5678 | Perryridge  |
|        9876 | xyz         |
|        9877 | abc         |
+-------------+-------------+
4 rows in set (0.00 sec)

mysql> insert into branch_loan values(9999,
Query OK, 1 row affected (0.00 sec)

mysql> select * from branch_loan;
+-------------+-------------+
| loan_number | branch_name |
+-------------+-------------+
|        1234 | Perryridge  |
|        5678 | Perryridge  |
|        9876 | xyz         |
|        9877 | abc         |
|        9999 | xyz         |
+-------------+-------------+
5 rows in set (0.00 sec)
```

```
mysql>
mysql>
mysql> drop view branch_loan;
Query OK, 0 rows affected (0.02 sec)
```

B) Create a table

emp (eno, ename, hrs, pno, super_no) and

project (pname, pno, thrs, head_no)

where thrs is the total hours and is the derived attribute.

Its value is the sum of hrs of all employees working on that project.

eno and pno are primary keys, head_no is foreign key to emp relation.

a) **Insert 4 tuples and**

b) **Create a trigger to insert a new employee tuple and display the new total hours from project table.**

B) Database Schema for a Student Library scenario

Student(Stud_no : integer, Stud_name: string)
Book(book_no: integer, book_name:string, author: string)
Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer)

For the above schema, perform the following—

a) **Create the tables.**
b) **Insert around 2 records in each of the tables**
c) **In Student table Stud_No value should be auto incremented**
d) **Create a trigger to take automatic backup of Book table after insert**

```
mysql> create table student (
    -> stud_no int PRIMARY KEY AUTO_INCREMENT,
    -> stud_name varchar(20)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> create table book (
    -> book_no int,
    -> book_name varchar(20),
    -> author varchar(20)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table iss_rec (
    -> iss_no int,
    -> iss_date date,
    -> mem_no int,
    -> book_no int
    -> );
Query OK, 0 rows affected (0.03 sec)
```

B) Customer(Cust id : integer, cust_name: string)
Item(item_id: integer, item_name: string, price: integer)
Sale(bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty_sold: integer)

For the above schema, perform the following—

a) **Create the tables with the appropriate integrity constraints**
b) **Insert around 2 records in each of the tables**
c) **Display all cust_id and item_name for all customers who have ordered item X. Use Join.**
d) **Create a new user "abc" and give only select and insert privileges to it.**
e) **Sort all customers by cust_id in ascending order and cust_name in descending order.**

```
mysql> create table cust (
    -> cust_id int PRIMARY KEY,
    -> cust_name varchar(20)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table item(
    -> item_id int PRIMARY KEY,
    -> item_name varchar(20),
    -> price double
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table sale (
    -> bill_no int primary key,
    -> bill_date date,
    -> cust_id int,
    -> item_id int,
    -> quan_sold int,
    -> foreign key(cust_id) references cust(cust_id),
    -> foregin key(item_id) references item(item_id)
    -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the ma
L server version for the right syntax to use near 'key(item_id) refere
)' at line 8
mysql> create table sale (
    -> bill_no int primary key,
    -> bill_date date,
    -> cust_id int,
    -> item_id int,
    -> quan_sold int,
    -> foreign key(cust_id) references cust(cust_id),
    -> foreign key(item_id) references item(item_id)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> insert into cust values(101, "aksh");
Query OK, 1 row affected (0.01 sec)

mysql> insert into cust values(102, "sakshi");
Query OK, 1 row affected (0.00 sec)

mysql> insert into cust values(103, "dhruv");
Query OK, 1 row affected (0.00 sec)

mysql> insert into item values(1, "xyz", 1000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into item values(2, "abc", 2000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into item values(3, "pqr", 2300);
Query OK, 1 row affected (0.01 sec)

mysql> insert into sale values(210, "2022-05-12", 101, 1, 20);
Query OK, 1 row affected (0.01 sec)

mysql> insert into sale values(211, "2022-03-12", 102, 1, 10);
Query OK, 1 row affected (0.01 sec)

mysql> insert into sale values(212, "2022-03-11", 103, 2, 5);
Query OK, 1 row affected (0.01 sec)
```

```
    -> join sale c
mysql> select cust.cust_id, item.item_id
    -> from cust
    -> join sale
    -> on cust.cust_id=sale.cust_id
    -> join item
    -> on sale.item_id=item.item_id
    -> where item.item_id = 1;
+---------+---------+
| cust_id | item_id |
+---------+---------+
|     101 |       1 |
|     102 |       1 |
+---------+---------+
2 rows in set (0.00 sec)
```

```
mysql> create user 'abc'@'localhost' identified by 'password';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> grant select, insert on djsce.sale to 'abc'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from cust
    -> order by cust_id, cust_name desc;
+---------+-----------+
| cust_id | cust_name |
+---------+-----------+
|     101 | aksh      |
|     102 | sakshi    |
|     103 | dhruv     |
+---------+-----------+
3 rows in set (0.00 sec)

mysql> select * from cust
    -> order by cust_name desc, cust_id;
+---------+-----------+
| cust_id | cust_name |
+---------+-----------+
|     102 | sakshi    |
|     103 | dhruv     |
|     101 | aksh      |
+---------+-----------+
3 rows in set (0.00 sec)
```

A) Write a query in sql to create a table employee and department.
Employee(empno,ename,deptno,job,hiredate)
Department(deptno,dname,loc)
1. Include the following constraints on column of emp table.
a) to make the empno as primary key of the table and
b) to make primary key to increment automatically
b) to ensure that the ename column does not contain NULL values

2. Include the following constraints on column of dept table.
a) to make deptno as primary key.
b) to ensure dname,loc columns does not contain NULL values
Also enforce REFERENTIAL INTEGRITY, declare deptno field of dept table as primary key and deptno field of emp table as foreign key.

3. Modify the above table to add three more columns salary,mgr,comm to the emp table. add salary column with constraint greater than zero.

```
mysql> create table department (
    -> deptno int PRIMARY KEY,
    -> dname varchar(50) NOT NULL,
    -> loc varchar(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table employee (
    -> empno int PRIMARY KEY AUTO_INCREMENT,
    -> ename varchar(50) NOT NULL,
    -> deptno int,
    -> job varchar(50),
    -> hiredate date,
    -> foreign key(deptno) references department(deptno)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> alter table employee add salary int check(salary > 0);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table employee add column mgr varchar(50);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> alter table employee add comm varchar(50);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

B) Create the following constraints and put proper constraints wherever required:
Employee(eno,name,addr,qualification,course_id, dept, design,dob,doj)
Accounts(eno, basec_sal, DA,HRA,PF,gross_sal)
Training(course_id,course_name)

a) Select name of all the employees in 'system' department.
b) Select employee no. and qualification of all employees in marketing department and those whose name start with 'a'
c)
d) Select employee no. of all those employee whose gross pay is greater then 5000 but basic is less then 4000.

```
mysql> create table training (
    -> course_id int PRIMARY KEY,
    -> course_name varchar(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> create table employee (
    -> eno int PRIMARY KEY,
    -> name varchar(50) NOT NULL,
    -> addrs varchar(100),
    -> qualification varchar(50),
    -> course_id int,
    -> dept varchar(20),
    -> design varchar(20),
    -> dob date,
    -> doj date,
    -> foreign key(course_id) references training(course_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table accounts(
    -> eno int PRIMARY KEY,
    -> basic_salary double NOT NULL,
    -> DA double,
    -> HRA double,
    -> PF double,
    -> gross_salary double NOT NULL,
    -> foreign key(eno) references employee(eno)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select name from employee
    -> where dept='system';
Empty set (0.00 sec)

mysql> select eno, qualification from employee
    -> where dept='marketing' and name like 'a%';
Empty set (0.00 sec)

mysql> select employee.eno
    -> from employee
    -> join accounts
    -> on employee.eno=accounts.eno
    -> where gross_salary>5000 and basic_salary<5000;
Empty set (0.00 sec)

mysql>
```

A) Write a query in sql to create a table employee and department.

Employee(empno,ename,deptno,job,hiredate)

Department(deptno,dname,loc)

Enforce REFERENTIAL INTEGRITY, declare deptno field of dept table as primary key and deptno field of emp table as foreign key.

Insert following values in employee table.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30

a. Give list of emp name & their job spec who are working in dept no 20 ?
b. Compile the details of emp working in dept no 30 ?
c. To find the total number of employees.
d. To find the total number of clerk hired after 13-jan-81.
e. Determine which departments have more than two people holding a particular job?
f. Find all department that have at least two clerk?
g. Retrieve emp name and job who have the same job as that of allen?
h. List all emp name and their job of those departments that are located at Chicago?

```
mysql> select ename, job from emp
    -> where deptno=20;
Empty set (0.01 sec)

mysql> select * from emp
    -> where dept=30;
ERROR 1054 (42S22): Unknown column 'dept' in 'where clause'
mysql> select * from emp
    -> where deptno=30;
Empty set (0.00 sec)

mysql> select count(*) from emp;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)

mysql> select count(*) from emp
    -> where job="clerk" and hiredate>"1981-01-13";
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)
```

```
mysql> select distinct deptno from emp
    -> where^C
mysql> select deptno, job
    -> from employee
    -> group by dept, job
    -> having count(*) > 2;
ERROR 1054 (42S22): Unknown column 'deptno' in 'field list'
mysql> select deptno, job
    -> from emp
    -> group by dept, job
    -> having count(*) > 2;
ERROR 1054 (42S22): Unknown column 'dept' in 'group statement'
mysql> select deptno, job
    -> from emp
    -> group by deptno, job
    -> having count(*) > 2;
Empty set (0.00 sec)

mysql> select deptno from emp
    -> where job = "clerk"
    -> group by deptno
    -> having count(*) >= 2;
Empty set (0.00 sec)

mysql> select ename, job from emp
    -> where job = (select job from emp where ename="allen");
Empty set (0.00 sec)

mysql> select ename, job from employee
    -> ^C
mysql> select enmae, job from emp
    -> join department
    -> on emp.deptno = department.deptno
    -> where department.loc = 'Chicago';
ERROR 1054 (42S22): Unknown column 'enmae' in 'field list'
mysql> select ename, job from emp
    -> join department
    -> on emp.deptno = department.deptno
    -> where department.loc = 'Chicago';
Empty set (0.00 sec)
```

A) Create a table client master with the following fields
client_no, name, address1, address2, city, state, pin_code, remarks, balance due

1) Add the following constraints

- **Create a primary key constraint on the column client_no**
- **to make primary key to increment automatically**
- **to ensure that the balance due column does not contain NULL values**

2) Perform the following

a. **Insert two rows in the table client master**
b. **Show all rows of client master**
c. **Add a new column in your table : AGE**
d. **Update the table client master**
e. **Delete a row from client master where age is greater than 60**
f. **Create table supplymaster (supplierno,supllliername,address1,address2,city,state, pincode,remarks,baldue)**
g. **Insert values in table supplymaster by using clientmaster.**
h. **Drop table client master**
i. **Drop table supply master**

ISME Name column daalna bhul gaya hu daal dena

```
mysql> create table client_master (
    -> client_no int,
    -> address1 varchar(20),
    -> address2 varchar(20),
    -> city varchar(20),
    -> state varchar(20),
    -> pincode int,
    -> remarks varchar(20),
    -> balance_due double
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
ysql> alter table client_master
    -> add primary key(client_no);
uery OK, 0 rows affected (0.07 sec)
ecords: 0  Duplicates: 0  Warnings: 0
```

```
mysql> alter table client_master
    -> modify client_no int auto_increment;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table client_master
    -> modify balance_due double not null;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> insert into client_master values(1, "xyz", "abc", "mumbai", "maharashtra", 400000, "good", 5000);

Query OK, 1 row affected (0.01 sec)

mysql> insert into client_master values(2, "xyz", "abc", "mumbai", "maharashtra", 400000, "good", 5000);

Query OK, 1 row affected (0.00 sec)

mysql> select * from client_master
    -> ;
+-----------+----------+----------+--------+-------------+---------+---------+-------------+
| client_no | address1 | address2 | city   | state       | pincode | remarks | balance_due |
+-----------+----------+----------+--------+-------------+---------+---------+-------------+
|         1 | xyz      | abc      | mumbai | maharashtra |  400000 | good    |        5000 |
|         2 | xyz      | abc      | mumbai | maharashtra |  400000 | good    |        5000 |
+-----------+----------+----------+--------+-------------+---------+---------+-------------+
2 rows in set (0.00 sec)
```

```
mysql> alter table client_master
    -> add column age int;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update client_master
    -> set age=61
    -> where client_no=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update client_master
    -> set age=59
    -> where client_no=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> delete from client_master
    -> where age>60;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> create table supplymaster(
    -> supplierno int,
    -> suppliername varchar(50),
    -> address1 varchar(50),
    -> address2 varchar(50),
    -> city varchar(50),
    -> state varchar(50),
    -> pincode int,
    -> remarks varchar(50),
    -> bal_due int
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> alter table supplymaster
    -> drop suppliername;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> insert into supplymaster select * from client_master;
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0
```

A) . Create the following constraints and put proper constraints wherever required:
Employee(eno,name,addr,qualification,course_id, dept, design,dob,doj)
Accounts(eno, basec_sal, DA,HRA,PF,gross_sal)
Training(course_id,course_name)

a) Select name of all the employees in 'system' department.
b) Select employee no. and qualification of all employees in marketing department and those whose name start with 'a'
c) Select employee no. of all those employee whose gross pay is greater then 5000 but basic is less then 4000.
d) All the employees have been given a phone. So to keep the records of phone no. a new field needs to get added modify structure of table.
e) Count the total number of employees, max salary received by an employee, total amount paid by the management for the employees' salary and avg salary of employee.

```
mysql> alter table employee
    -> add column phone int;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> select distinct count(employee.eno), max(basic_salary+gross_salary) from employee, accounts;
+---------------------+-------------------------------+
| count(employee.eno) | max(basic_salary+gross_salary) |
+---------------------+-------------------------------+
|                   0 |                          NULL |
+---------------------+-------------------------------+
1 row in set (0.00 sec)

mysql> select sum(basic_salary+gross_salary), avg(basic_salary+gross_salary) from accounts
    -> ;
+-------------------------------+-------------------------------+
| sum(basic_salary+gross_salary) | avg(basic_salary+gross_salary) |
+-------------------------------+-------------------------------+
|                          NULL |                          NULL |
+-------------------------------+-------------------------------+
1 row in set (0.00 sec)

mysql>
```

B) Write a query in sql to create a table employee and department.
Employee(empno,ename,deptno,job,hiredate)
Department(deptno,dname,loc)
1. Include the following constraints on column of emp table.
a) to make the empno as primary key of the employee table and deptno as a primary key of department table

2. Change the data type of "job"from string to integer.
3. Create a trigger such that if hiredate of employee is less than 31/12/1970, then remove that employee record.

```
mysql> alter table emp
    -> modify job int;
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

) Create the following constraints and put proper constraints wherever required:

Emp(e_no, ename, sal, d_no, man_no, job_no, doj, dob, commission)
Dept(d_no, dname)
Job(Job_type, j_no, j_performed)
Manager(m_name, m_no)

Insert data in the above tables.

a) Describe the tables EMP and DEPT.
b) Select all information from EMP table.
c) List the details of employee having salary between 1000 and 2000.

d) List the employees in dept 10 & 20 in alphabetical order,

e)

Display name , annual salary and commission of all employee whose monthly salary is greater then their commission.

B) Create a table

emp (eno, ename, hrs, pno, super_no) and

project (pname, pno, thrs, head_no)

where thrs is the total hours and is the derived attribute.

Its value is the sum of hrs of all employees working on that project.

eno and pno are primary keys, head_no is foreign key to emp relation.

a) **Insert 4 tuples and**

b) **Create a trigger to insert a new employee tuple and display the new total hours from project table.**