1. **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.

**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.

1. **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.

**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.

1. **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.

**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.

1. 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.

**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.

1. **Insert 4 tuples and**
2. **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—

1. Create the tables.
2. Insert around 2 records in each of the tables
3. In Student table Stud\_No value should be auto incremented
4. Create a trigger to take automatic backup of Book table after insert

**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—

1. Create the tables.
2. Insert around 2 records in each of the tables
3. In Student table Stud\_No value should be auto incremented
4. Create a trigger to take automatic backup of Book table after insert

**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.

**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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. Display all cust\_id and item\_name for all customers who have ordered item X. Use Join.
4. Create a new user “abc” and give only select and insert privileges to it.
5. Sort all customers by cust\_id in ascending order and cust\_name in descending order.

**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.**

**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)



c) 

d)

1. **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

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

**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)



c) 

d)

**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**

1. Insert two rows in the table client master
2. Show all rows of client master
3. Add a new column in your table : AGE
4. Update the table client master
5. Delete a row from client master where age is greater than 60
6. Create table supplymaster (supplierno,suplliername,address1,address2,city,state,  
   pincode,remarks,baldue)
7. Insert values in table supplymaster by using clientmaster.
8. Drop table client master
9. Drop table supply master

**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)



c) 

d)

1. **. 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)



c) 

d)



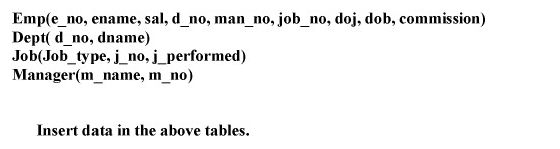
e)

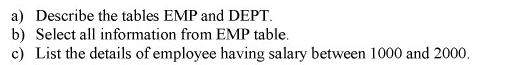
**B) Write a query in sql to create a table employee and department.**   
Employee(empno,ename,deptno,job,hiredate)  
Department(deptno,dname,loc)  
**1. Include the following constraints on column of emp table.**  
a) to make the empno as primary key of the employee table and deptno as a primary key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

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





d) 

e) 

**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.**

1. **.** 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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. List all the bills for the current date with the customer names and item numbers
4. List the details of the customer who have bought a product which has a price>200
5. Give a count of how many products have been bought by each customer
6. Give a list of products bought by a customer having cust\_id as 5
7. Create a view which lists out the bill\_no, bill\_date, cust\_id, item\_id, price, qty\_sold, amount

**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.

1. **Insert 4 tuples and**
2. **Create a trigger to insert a new employee tuple and display the new total hours from project table.**

**A). Database Schema for a Student Library scenario**

Student(**Stud\_no : integer,** Stud\_name: string)

Membership(**Mem\_no: integer**, **Stud\_no: integer**)

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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. List all the student names with their membership numbers
4. List the details of students who borrowed book whose author is CJDATE
5. Give a count of how many books have been bought by each student
6. Give a list of books taken by student with stud\_no as 5
7. List the book details which are issued as of today
8. Create a view which lists out the iss\_no, iss \_date, stud\_name, book name

**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.

1. **Insert 4 tuples and**
2. **Create a trigger to change the project of an employee and display the new total hours from project table.**
3. **. Database Schema for a Employee-pay scenario**

employee(**emp\_id : integer**, emp\_name: string)

department(**dept\_id: integer**,dept\_name:string)

paydetails(**emp\_id : integer**, **dept\_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date)

payroll**(emp\_id : integer**, pay\_date: date)

For the above schema, perform the following—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. List the employee details department wise
4. List all the employee names who joined after particular date
5. List the details of employees whose basic salary is between 10,000 and 20,000
6. Give a count of how many employees are working in each department
7. List the details for an employee\_id=5
8. Create a view which lists out the emp\_name, department, basic, dedeuctions,

**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.A) . Database Schema for a Video Library scenario**

Customer(**cust\_no: integer**,cust\_name: string)

Membership(**Mem\_no: integer**, **cust\_no: integer**)

Cassette(**cass\_no:integer**, cass\_name:string, Language: String)

Iss\_rec**(iss\_no: integer**, iss\_date: date, **mem\_no: integer**, **cass\_no: integer**)

For the above schema, perform the following—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. List all the customer names with their membership numbers
4. List all the issues for the current date with the customer names and cassette names
5. List the details of the customer who has borrowed the cassette whose title is “ The Legend”
6. Give a count of how many cassettes have been borrowed by each customer
7. Give a list of book which has been taken by the student with mem\_no as 5
8. List the cassettes issues for today
9. Create a view which lists outs the iss\_no, iss\_date, cust\_name, cass\_name

**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—

1. Create the tables.
2. Insert around 2 records in each of the tables
3. In Student table Stud\_No value should be auto incremented
4. Create a trigger to take automatic backup of Book table after insert
5. **.** **Database Schema for a student-Lab scenario**

Student(**stud\_no: integer**, stud\_name: string, **class: string**)

Class(**class: string,** **descrip: string**)

Lab(**mach\_no: integer**, Lab\_no: integer, description: String)

Allotment(**Stud\_no: Integer, mach\_no: integer, dayof week: string)**

For the above schema, perform the following—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. List all the machine allotments with the student names, lab and machine numbers
4. List the total number of lab allotments day wise
5. Give a count of how many machines have been allocated to the ‘CSIT’ class
6. Give a machine allotment etails of the stud\_no 5 with his personal and class details
7. Count for how many machines have been allocatedin **Lab\_no 1**  for the day of the week as “Monday”
8. How many students class wise have allocated machines in the labs
9. Create a view which lists out the stud\_no, stud\_name, mach\_no, lab\_no, dayofweek

**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.

1. **Insert 4 tuples and**
2. **Create a trigger to insert a new employee tuple and display the new total hours from project table.**
3. **Database Schema for a Student Library scenario**

Student(**Stud\_no : integer,** Stud\_name: string)

Membership(**Mem\_no: integer**, **Stud\_no: integer**)

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—

1. Create the tables.
2. In Student table Stud\_No value should be auto incremented
3. Insert check constraint on “date”field of table Book. Date should not exceed today’s date.
4. Insert around 2 records in each of the tables
5. List all the student names with their membership numbers
6. List the details of students who borrowed book whose author is CJDATE
7. Create a trigger to take automatic backup of Book table after insert.

**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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. Perform outer join of customer and sale.
4. Create a new user “abc” and give only select and insert privileges to it.

**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. 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.**

**4. Count the number of employees in each department.**

**5. Find all employees working as “managers” in department of “computer engg”**

**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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. Display all cust\_id and item\_name for all customers who have ordered item X. Use Join.
4. Create a new user “abc” and give only select and insert privileges to it.
5. Sort all customers by cust\_id in ascending order and cust\_name in descending order.

| **EMPLOYEE\_ID** | **LAST\_NAME** | **SALARY** | **JOB\_ID** | **HIRE\_DATE** | **DEPT\_ID** |
| --- | --- | --- | --- | --- | --- |
| 100 | King | 10000 | AD\_PRES | 19-11-1999 | 90 |
| 101 | Kochhar | 20000 | AD\_VP | 01-10-1998 | 40 |
| 102 | De Haan | 15000 | AD\_VP | 20-12-1996 | 90 |
| 103 | Whalen | 4000 | AD\_PRES | 09-07-1999 | 45 |
| 104 | King | 15900 | AD\_VP | 05-06/1989 |  |

* + 1. Create query to display the last name and salary for all employees whose salary is not in the range of $5,000 and $12,000.
    2. Create a query to display the name and hire date of any employee hired after employee Kochhar.
    3. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the result in descending order of salary.
    4. Change the name of employee 101 to Drexler.
    5. Modify the EMP table. Add a COMMISSION column of type float. Add a constraint to the commission column that ensures that a commission value is greater than zero.
    6. Create a view called EMPLOYEE\_VU based on the employee\_id, last\_names, and dept\_id from the EMPLOYEE table. Changes the heading for the employee name to EMP\_NAME. And Display the contents of the EMPLOYEE\_VU view.
    7. Write compound query that lists the following:
       - Last names and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to any department or not.
       - Department ID and department name of all the departments from DEPARTMENTS table, regardless of whether or not they have employees working in them.

A) **Create following tables and put proper constraints wherever required:**

Employee(e\_no,name,addr,qualification,course\_id,dept,design,dob,date\_join)

Accounts(e\_no,basic\_sal,DA,HRA,PF,gross\_sal)

Training(Course\_id,course\_name)

**Insert data in the above tables.**

**Solve following queries using SQL**

1. Select name of all the employees in ‘system’department.
2. Select e\_no and qualification of all employees in marketing department and those whose name starts with ‘a’.
3. Select e\_no of all employees who do not belong to ‘account’ department.
4. Extract the names, departments, designation of all employees sorted according to names in descending order within departments in ascending order.
5. All the employees have been given a phone. So keep the records of phone\_no a new field needs to get added. Modify structure of table.
6. Employee relation needs to be renamed as ‘personal’. Alter the name of table.
7. Count The total number ofemployee ,max salary received by an employee,total salary paid by management for employees’salary and avg salary of employee according to department wise.

**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—

1. Create the tables with the appropriate integrity constraints
2. Insert around 2 records in each of the tables
3. Display all cust\_id and item\_name for all customers who have ordered item X. Use Join.
4. Create a new user “abc” and give only select and insert privileges to it.
5. Sort all customers by cust\_id in ascending order and cust\_name in descending order.

**For the following relation schema:**

**employee(employee-name, street, city)**

**works(employee-name, company-name, salary)**

**company(company-name, city)**

**manages(employee-name, manager-name)**

1. Create tables for the above mentioned schema
2. Insert five records in each table
3. Insert appropriate integrity constraint for each table
4. Describe the tables.
5. Create a view “employee\_company” which will have the records of all employees along with the company they are working for.
6. Display all records of “ employee\_company”
7. Drop view.
8. Perform the following complex queries.
   1. Find the names of all employees in the database who live in the same cities as the companies for which they work.
   2. Find the names of all employees in the database who live in the same cities and on the same streets as do their managers.
   3. Find the names, street address, and cities of residence for all employees who work for 'First Bank Corporation' and earn more than $10,000.
   4. Find the names of all employees in the database who do not work for 'First Bank Corporation'. Assume that all people work for exactly one company.
   5. Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company

**For the following relation schema:**

**employee(employee-name, street, city)**

**works(employee-name, company-name, salary)**

**company(company-name, city)**

**manages(employee-name, manager-name)**

1. Create tables for the above mentioned schema
2. Insert five records in each table
3. Insert appropriate integrity constraint for each table
4. Describe the tables.
5. Rename the table “works” as “ works\_for”
6. Create outer join of “ employee” and “manager”
7. Perform the following complex queries.
   1. Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.
   2. Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'.
   3. Find the names of all employees in the database who live in the same cities and on the same streets as do their managers.
   4. Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company
   5. Find the names, street address, and cities of residence for all employees who work for 'First Bank Corporation' and earn more than $10,000.

**Write a query in sql to create a table employee and department.   
Employee(empno,ename,deptno,job,hiredate)  
Department(deptno,dname,loc)**

Include the following constraints on column of emp table.  
a) to make the empno as primary key of the table and  
b) to ensure that the ename column does not contain NULL values and  
c) to put the current date as default date in hire date column in case data is not supplied for the column.

Include the following constraints on column of dept table.  
a) to make deptno as primary key.  
b) to ensure dname,loc coloumns 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.

Perform

1. List the job that have average annual salary greater than all manager?
2. Find the no of employees and job group by deptno?
3. List all emp names and their salaries who are working in dept no 10 ?
4. To find the minimum salary paid to any employee
5. To list the employee who heads the list alphabetically.
6. List all emp name, dept names and location whose deptno is common in both dept table and emp table?
7. List all jobs in dept 10 and 20 or in both?
8. Change the data type of “job”from string to integer.
9. Create a trigger such that if hiredate of employee is less than 31/12/1970, then remove that employee record.
10. Drop trigger
11. Create user “xyz” and add all priviledges to it.