S1) Create the following table STUDENT with Regd No as Primary Key.

REGD.NO	NAME	BRANCH
0001	Ram	CSE
0002	Hari	MECH
0003	Pradeep	EEE
0004	Deepak	ETC

1) Write down the SQL command which will show the Regd. No of Pradeep.

2) Write down the SQL command which will display the Name and Branch of Regd No 0002.

3) Write a SQL command which will count the number of rows existing in STUDENT table.

4) Add another column address in STUDENT table.

5) Change the branch of Ram from CSE to ETC.

S2) Create the following table STUDENT with Regd No as Primary Key.

REGD.No	Name	Branch
0001	Ram	CSE
0002	Hari	MECH
0003	Predeep	EEE
0004	Deepak	ETC

1) Write down the SQL command which will delete record of Ram.

2) write down the SQL command which will show all record whose name start with "R".

3) Write a SQL command which Update Name of "Pradeep" to "Pradeep Kumar".

4) Add another column address in STUDENT table and update all address.

5) Find the total registration count of each branch.

S3) Create a table Salesperson as follows with SID as a Primary Key and Table Product with SID as foreign Key and PID as a primary key.

Salesperson

SID	Name	Age	Salary
1	Abe	61	140000
2	Bob	34	44000
5	Chris	34	40000
7	Dan	41	52000
8	Ken	57	115000

Product

PID	SID	Name	City
1	1	Samsonic	pleasant
2	5	Panasonic	oaktown
3	7	Samony	jackson
4	8	Orange	Jackson

1. Find the name of all salespeople that have an order with Samsonic.
2. The names of all salespeople that do not have any order with Samsonic.
3. The names of salespeople that have 2 or more orders.
4. Display all salesperson with age less than 40.

5. Find the ID and name of sales person who is selling product Panasonic.

S4) Consider the following schema:

Suppliers (sid: integer, sname: varchar(50), address: varchar(60)),sid as a primary key.

Parts (pid: integer, pname: varchar(50), color: varchar(20)),pid as primary key.

Catalog (sid: integer, pid: integer, cost: real),sid and pid as a foreign key which refers Supplier and Parts table respectively.

Insert values in each table.

Write SQL command for each of the following queries.

1) Find the distinct pnames of all parts.
2) Alter the data types of sname as varchar(30).
3) Find out the supplier who is supplying part "Keyboard" whose cost is 5000.
4) Remove all parts whose name is "Mouse".

5) List all supplier whose name start with "S" in descending order.

S5) Create table as follows

dept (deptno, dname, mgreno).deptno as a primary key

emp (eno, ename, bdate, title, salary, deptno),eno as a primary key and deptno is foreign key.

proj (pno, pname, budget, deptno),pno as a primary key, deptno as a foreign refer dept

workson (eno, pno, responsibility, hours),eno and pno as a foreign key which references table emp and proj

Insert values in each table

1) Write an SQL query that returns the project number and name for projects with a budget greater than $100,000.
2) Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager'.
3) Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than $35,000.
4) Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary.
5) List manager Number whose department name is "Production".

S6) Create table as follows

dept (dno, dname, mgreno).dno as a primary key

emp (eno, ename, bdate, title, salary, dno),eno as a primary key

proj (pno, pname, budget, dno),pno as a primary key, dno as a foreign refer dept

workson (eno, pno, responsibility, hours),eno and pno as a foreign key which references table emp and proj

Insert values in each table

1) Write an SQL query that returns the departments (all fields) ordered by ascending department name.
2) Write an SQL query that returns the employee name, department name, and employee title.
3) Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
4) Write an SQL query that returns the project name, department name, and budget for all projects with a budget < $50,000.
5) Find the responsibility of the employee "Ramesh" who is working on project "Banking".

S6) Create table as follows

emp (eno, ename, bdate, title, salary, dno),eno as a primary key

proj (pno, pname, budget, dno),pno as a primary key, dno as a foreign refer dept

dept (dno, dname, mgreno).dno as a primary key

workson (eno, pno, resp, hours),eno and pno as a foreign key which references table emp and proj

Insert values in each table

1) Write an SQL query that returns the departments (all fields) ordered by ascending department name.
2) Write an SQL query that returns the employee name, department name, and employee title.
3) Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
4) Write an SQL query that returns the project name, department name, and budget for all projects with a budget < $50,000.
5) Find the responsibility of the employee "Ramesh" who is working on project "Banking".

S7) Create table as follows

emp (eno, ename, bdate, title, salary, dno),eno as a primary key

proj (pno, pname, budget, dno),pno as a primary key,dno as a foreign refer dept

dept (dno, dname, mgreno).dno as a primary key

workson (eno, pno, resp, hours),eno and pno as a foreign key which references table emp and proj

Insert values in each table

1) Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.

2) Write an SQL query that returns the employee name, project name, employee title, and hours for all works on records.

3) Find the entire employee whose salary in between 1000 and 8000.

4) List the entire projects name.

5) Find the employee who working on project "Banking" of 'Production" department with duration 120 hours.

S7) Create table as follows

emp (eno, ename, bdate, title, salary, dno),eno as a primary key

proj (pno, pname, budget, dno),pno as a primary key,dno as a foreign refer dept

dept (dno, dname, mgreno).dno as a primary key

workson (eno, pno, resp, hours),eno and pno as a foreign key which references table emp and proj.

Insert values in each table

1) Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.

2) Write an SQL query that returns the employee name, project name, employee title, and hours for all works on records.

3) Find the entire employee whose salary in between 1000 and 8000.

4) List the entire projects name.

5) Find the employee who working on project "Banking" of 'Production" department with duration 120 hours.

S8) Create table as follows

Employee (employee-name, street, city) employee name as primary key.

Company (company-name, city) company-name as primary key.

Works (employee-name, company-name, salary)

Manages (employee-name, manager-name)

1) Count employees company wise where salary greater than 25000.
2) Delete column salary from Works.
3) Display the structure of manager table.
4) Update data type of employee-name in Manager from varchar(30) to varchar(50)
5) Find the employees whose salary ranges 25000 to 50000.

S8) Create table as follows

Employee (employee-name, street, city) employee name as primary key.

Company (company-name, city) company-name as primary key.

Works (employee-name, company-name, salary)

Manages (employee-name, manager-name)

1) Count employees company wise where salary greater than 25000.
2) Delete column salary from Works.
3) Display the structure of manager table.
4) Update data type of employee-name in Manager from varchar(30) to varchar(50)
5) Find the employees whose salary ranges 25000 to 50000.

S9) Create table as follows

1) ***Employee*** *(employee-name, street, city) employee name as primary key.*
2) ***Company*** *(company-name, city) company-name as primary key.*
3) ***Works*** *(employee-name, company-name, salary)*
4) ***Manages*** *(employee-name, manager-name)*

1) Find the names, street address, and cities of residence for all employees
2) Find the names of all employees in the database who live in the city "Pune"
3) Find the names of all employees in the database who do not work for 'First Bank Corporation'.
4) Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'.
5) Find all the managers.

S9) Create table as follows

5) *Employee (employee-name, street, city) employee name as primary key.*
6) *Company (company-name, city) company-name as primary key.*
7) *Works (employee-name, company-name, salary)*
8) *Manages (employee-name, manager-name)*

1) Find the names, street address, and cities of residence for all employees
2) Find the names of all employees in the database who live in the city "Pune"
3) Find the names of all employees in the database who do not work for 'First Bank Corporation'.
4) Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'.
5) Find all the managers.

S10) Create table

Employee(Employee_Id,Lastname,Firstname,Middlename,Job_Id,Manager_id,Hiredate,Salary,Department_id)

- Insert following records.

Employee_Id	Lastname	Firstname	Middlename	Job_Id	Manager_id	Hiredate	Salary	Department_id
7369	Smith	Jon	Q	667	7902	17-DEC-84	800	10
7499	Allen	Kevin	J	670	7698	20-FEB-85	1600	20
7505	Doyle	Jean	K	671	7839	04-APR-85	2850	20
7506	Dennis	Lynn	S	671	7839	15-MAY-85	2750	30
7507	Baker	Leslie	D	671	7839	10-JUN-85	2200	40
7521	wark	cynthia	D	670	7698	22-FEB-85	1250	10

1) Create a view for all column of Employee table.

2) Create a view of last name, firstname, middlename of Employee table.

3) Create a view of all employees whose last name start from "S" and middle name is "Q".

4) Create a view of all employees with salary incremented by 10 %.

5) Delete view for all column of Employee.

S11) Do the following

1) Create a table Animal (id, name) with auto increment id field and insert value in animal table.

2) Create table Location as follows

Location (Location_Id, Reginal_Group)

You will have to create an auto-increment field start with 100.

3) Rename Location table with "Location_of_india", and display Location_of_india table content.

4) Create a view of all location whose location_id =101;

5) Alter table Location to add column "Location Name"

S12) Create tables as follows

Location (Location_Id, Reginal_Group)

Department (Department_Id,Name,Location_Id)

Job (Job_Id,Function)

Employee (Employee_Id, Lastname, Firstname, Middlename, Job_Id, Manager_id, Hiredate, Salary, Department_id)

Insert the values

Execute following queries

1. List the details about "smith"
2. List out the employee whose job id is 671.
3. List out the employees who are earning salary between 3000 and 4500.
4. List out the employees who are working in department 10 or 20.
5. Find out the employees who are not working in department 10 or 30.

S12) Create tables as follows

Location (Location_Id, Reginal_Group)

Department (Department_Id,Name,Location_Id)

 Job (Job_Id,Function)

Employee (Employee_Id, Lastname, Firstname, Middlename, Job_Id, Manager_id, Hiredate, Salary, Department_id)

Insert the values

Execute following queries

1. List the details about "smith"
2. List out the employee whose job id is 671.
3. List out the employees who are earning salary between 3000 and 4500.
4. List out the employees who are working in department 10 or 20.
5. Find out the employees who are not working in department 10 or 30.

S13) Create the 'product ' table and 'product_price_history' table

- *CREATE TABLE product (product_id,,product_name,supplier_name, unit_price);*

- *CREATE TABLE product_price_history (product_id,,product_name,supplier_name, unit_price);*

create a trigger to update the 'product_price_history' table when the price of the product is updated in the 'product' table.

S14) *Create table account(accno int,amount int)*

Create a trigger on account table before update in new inserted amount is less than "0" then set amount "0" else if amount is greater than 100 then set amount 100

S15) Do the following

 Write a PL/SQL block to find the maximum number from given three numbers.

2) CREATE TABLE studentdata (mark1, mark2, mark3 mark4,name_of _student)

Insert some values.

Write a function to calculate average of marks.

List the average marks of each student.

S16) Write a PL/SQL block to calculate the grade of minimum 10 students. Using MYSQL procedure and function.