



**DIPLOMA IN COMPUTER SCIENCE AND INFORMATION
TECHNOLOGY**

CENTRALIZED QUESTION BANK

1058234540- DATABASE MANAGEMENT SYSTEM

**DIRECTORATE OF TECHNICAL
EDUCATION GOVERNMENT OF
TAMILNADU**

DIPLOMA END SEMESTER / YEAR EXAMINATION – 2025

Course: Computer Science and Information Technology

Subject : Database Management System

QP Code : 1058234540

Time : 3 Hours

Date :

Session:

Max Marks: 100

Answer the Following Questions

1. A) Create a database named 'employee' and use the database 'employee' for doing the following operations

- Create 'department' Table with Primary Key as Department_id and 'emp' Table with Foreign Key
- Describe the table Schema.
- Show all the databases created by user.
- Show all the tables in employee database.

- B) Create a student data table for the following details,

Student Id	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	65	60	55
110	Jack	80	75	70

- Display students with marks in Maths above 70.
- Show students with marks in Chemistry not in pass range (45 and above).
- Show students with marks in Maths or Chemistry above 80. Display students with marks in all subjects below 50.

2. A) Create a database named 'employee' and use the database 'employee' for doing the following operations

- Create 'department' Table with Primary Key as Department _id and 'emp' Table with Foreign Key
- Describe the table Schema.
- Show all the databases created by user.
- Show all the tables in employee database.

B) Study the commands for views and execute the following queries using these commands:

- Create a view having e name and e city
- In the above view change the city to 'Chennai' where e name is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

3. A) Create a database named 'employee' and use the database 'employee' for doing the following operations

- Create 'department' Table with Primary Key as Department _id and 'emp' Table with Foreign Key
- Describe the table Schema.
- Show all the databases created by user.
- Show all the tables in employee database.

B) Create the following tables in the "Company DB" database:

- Employees(employee _id, name, role)
- Departments(department _id, name, manager _id)
- Use the GRANT statement to give User 1 SELECT permission on the Employee stable.
- Use the GRANT statement to give User 2 INSERT and UPDATE permissions on the Department stable.
- Use the REVOKE statement to revoke the SELECT permission from User 1 on the Department stable.

Use the REVOKE statement to revoke all permissions from User 2 on the Employee stable.

4. A) Create a database named 'employee' and use the database 'employee' for doing the following operations
- Create 'department' Table with Primary Key as Department _id and 'emp' Table with Foreign Key
 - Describe the table Schema.
 - Show all the databases created by user.
 - Show all the tables in employee database.
- B) Create the following tables in the "Company DB" database:
- Employees(employee _id, name, salary)
 - Departments(department _id, name, budget)
- Use the SAVE POINT statement to set a save point named "before _ update" in the transaction.
 - Update the salary of an employee in the Employees table.
 - Use the ROLL BACK TO SAVE POINT statement to roll back the transaction to the save point named "before _ update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.
5. A) Create a database named 'employee' and use the database 'employee' for doing the following operations
- Create 'department' Table with Primary Key as Department _id and 'emp' Table with Foreign Key
 - Describe the table Schema.
 - Show all the data bases created by user.
 - Show all the tables in employee database.
- B)**
- a) Create a Students table including three subject marks, date of birth. Write stored functions for calculating total, average, and age.
 - b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

6. A) Create a database named 'employee' and use the database 'employee' for doing the following operations

- Create 'department' Table with Primary Key as Department_id and 'emp' Table with Foreign Key
- Describe the table Schema.
- Show all the databases created by user., Show all the tables in employee database.

B) Implement a Cursor to Calculate Total Salary of Employees in a Department. Create a table called 'Inventory' containing information about products ,including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory' table when ever a product is sold.

7. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and execute the following queries using these comm. ands:

- Create a table 'emp' with attributes 'e number', 'ename', 'ecity', 'salary', 'eaddress', and 'deptname'.
- Create another table 'Company' with attributes 'cname', 'ccity',

- Emp number 'in the database 'employee'.
- Alter 'enumber' columnas'emp_no'in'emp'table.
- Delete all records in table 'Company'.

Student Id	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	65	60	55
110	Jack	80	75	70

‘emp_info’.

B) Create a student data table for the following details,

- Display students with marks in Maths above 70.
 - Show students with marks in Chemistry not in pass range (45 and above).
 - Show students with marks in Maths or Chemistry above 80. Display students with marks in all subjects below 50.

- Remove the ‘emp’ table from database.
- Re name the table name ‘emp’ as

8. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and execute the following queries using these commands:

- Create a table ‘emp’ with attributes ‘enumber’, ‘ename’, ‘ecity’, ‘salary’, ‘eaddress’,
- Create another table ‘Company’ with attributes ‘cname’, ‘ccity’,
- Emp number’ in the database ‘employee’.

- Alter 'enumber' column as 'emp_no' in 'emp' table.
- Delete all records in table 'Company'.
- Remove the 'emp' table from database.
- Rename the table name 'emp' as 'emp_info'.

B) Study the commands for views and execute the following queries using these commands:

- Create a view having ename and ecity
- In the above view change the ecity to 'Chennai' where ename is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

9. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and execute the following queries using these commands:

- Create a table 'emp' with attributes 'enumber', 'ename', 'ecity', 'salary', 'eaddress',
- Create another table 'Company' with attributes 'cname', 'ccity',
- 'empnumber' in the database 'employee'.
- Alter 'enumber' column as 'emp_no' in 'emp' table.
- Delete all records in table 'Company'.
- Remove the 'emp' table from database.
- Rename the table name 'emp' as 'emp_info'.

B) Create the following tables in the "CompanyDB" data base:

- Employees(employee_id, name, role)
- Departments(department_id, name, manager_id)
- Use the GRANT statement to give User 1 SELECT permission on the Employee stable.
- Use the GRANT statement to give User 2 INSERT and UPDATE permissions on the Department stable.
- Use the REVOKE statement to revoke the SELECT permission from User 1 on the Department stable.
- Use the REVOKE statement to revoke all permissions from User 2 on the Employee stable.

10. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and

execute the following queries using these commands:

- create table 'emp' with attributes 'enumber', 'ename', 'ecity', 'salary', 'eaddress',
- Create another table 'Company' with attributes 'cname', 'ccity',
- empnumber' in the database 'employee'.
- Alter 'enumber' column as 'emp_no' in 'emp' table.
- Delete all records in table 'Company'.
- Remove the 'emp' table from database.
- Rename the table name 'emp' as 'emp_info'.

B) Create the following tables in the "CompanyDB" database:

- Employees(employee_id, name, salary)
- Departments(department_id, name, budget)
- Use the SAVE POINT statement to set a save point named "before_update" in the transaction.
- Update the salary of an employee in the Employees table.
- Use the ROLL BACK TO SAVE POINT statement to rollback the transaction to the save point named "before_update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.

11. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and execute the following queries using these commands:

- Create a table 'emp' with attributes 'enumber', 'ename', 'ecity', 'salary', 'eaddress',
- Create another table 'Company' with attributes 'cname', 'ccity',
- Emp number' in the database 'employee'.
- Alter 'enumber' column as 'emp_no' in 'emp' table.
- Delete all records in table 'Company'.
- Remove the 'emp' table from database.
- Rename the table name 'emp' as 'emp_info'.

B) a) Create a Students table including three subject marks, date of birth. Write stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

12. A) Study Basic MySQL commands (create database, create table, use, drop, insert) and

execute the following queries using these commands:

- create table 'emp' with attributes 'enumber', 'ename', 'ecity', 'salary', 'eaddress',
- Create another table 'Company' with attributes 'cname', 'ccity',
- Emp number' in the database 'employee'.
- Alter 'enumber' column as 'emp_no' in 'emp' table.
- Delete all records in table 'Company'.
- Remove the 'emp' table from database.
- Rename the table name 'emp' as 'emp_info'.

B) Implement a Cursor to Calculate Total Salary of Employees in a Department.

Create a table called 'Inventory' containing information about products, including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory' table when ever a product is sold.

13. A) Study the viewing commands(select, update) and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
- Increase the salary of all employees by Rs.5,000.

Change the company city to Chennai where the company name is 'TCS'.

B) Create a student data table for the following details,

StudentId	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	65	60	55
110	Jack	80	75	70

- Display students with marks in Maths above 70.
- Show students with marks in Chemistry not in pass range (45 and above).
- Show students with marks in Maths or Chemistry above 80. Display students with marks in all subjects below 50.

14. A) Study the viewing commands(select, update)and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
 - Increase the salary of all employees by Rs.5,000.
- Change the company city to Chennai where the company name is 'TCS'.

B) Study the commands for views and execute the following queries using these commands:

- Create a view having e name and e city
- In the above view change the e city to 'Chennai 'where e name is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

15. A) Study the viewing commands (select, update)and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
 - Increase the salary of all employees by Rs.5,000.
- Change the company city to Chennai where the company name is 'TCS'.

B) a) Create a Student stable including three subject marks, date of birth. Write Stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

16. A) Study the viewing commands (select, update)and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
 - Increase the salary of all employees by Rs.5,000.
- Change the company city to Chennai where the company name is 'TCS'.

B) Create the following tables in the "Company DB" database:

- Employees(employee_id,name,salary)
- Departments(department_id,name,budget)

- Use the SAVE POINT statement to set a save point named "before_update" in the transaction.
- Update the salary of an employee in the Employees table.
- Use the ROLL BACK TO SAVE POINT statement to roll back the transaction to the save point named "before_update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.

17. A) Study the viewing commands (select, update) and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
- Increase the salary of all employees by Rs.5,000.

Change the company city to Chennai where the company name is 'TCS'.

B) a) Create a Student stable including three subject marks, date of birth. Write stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

18. A) Study the viewing commands (select, update) and execute the following queries using these commands:

- Find the names of all employees who live in Chennai.
- Increase the salary of all employees by Rs.5,000.

Change the company city to Chennai where the company name is 'TCS'.

B) Implement a Cursor to Calculate Total Salary of Employees in a Department.

Create a table called 'Inventory' containing information about products, including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory' table whenever a product is sold.

19. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Studentid	Student Name	Age	Grade	Subject ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	SubjectName	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

B) Create a student data table for the following details,

StudentId	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	6	60	5

		5		5
110	Jack	80	75	70

- Display students with marks in Maths above 70.
- Show students with marks in Chemistry not in pass range (45 and above).
- Show students with marks in Maths or Chemistry above 80. Display students with marks in all subjects below 50.

20. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Student id	Student Name	Age	Grade	Subject ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	Subject Name	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

B) Study the commands for views and execute the following queries using these commands:

- Create a view having e name and e city

- In the above view change the city to 'Chennai' where the name is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

21. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Student id	Student Name	Age	Grade	Subject ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	Subject Name	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

B) Create the following tables in the "Company DB" database:

- Employees(employee_id,name,role)
- Departments(department_id,name,manager_id)
- Use the GRANT statement to give User1 SELECT permission on the Employee table.
- Use the GRANT statement to give User2 INSERT and UPDATE

permissions on the Department stable.

- Use the REVOKE statement to revoke the SELECT permission from User1 on the Department stable.

Use the REVOKE statement to revoke all permissions from User2 on the Employee stable.

22. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Student id	Student Name	Age	Grade	Subject ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	Subject Name	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

- B) Create the following tables in the "Company DB" database:

- Employees(employee_id,name,salary)
- Departments(department_id,name,budget)
- Use the SAVE POINT statement to set a save point named "before_update" in the transaction.
- Update the salary of an employee in the Employees table.
- Use the ROLL BACK TO SAVE POINT statement to roll back the transaction

to the save point named "before_update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.

23. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Student id	Student Name	Age	Grade	Subject ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	Subject Name	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

B) a) Create a Students table including three subject marks, date of birth. Write stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

24. A) Create the following two Tables for executing the query given below,

Table1:Students_detail

Student	Student	Age	Grade	Subject
----------------	----------------	------------	--------------	----------------

id	Name			ID
1	John	18	A	101
2	Alice	17	B	102
3	Bob	16	C	103
4	Eve	17	A	105
5	Michael	18	B	104
6	Sarah	Null	Null	103
7	Rafi	19	B	Null
8	Ane	16	C	Null

Table2:Subjects_detail

Subject ID	Subject Name	Teacher
101	Mathematics	Mr.Smith
102	Science	Ms.Johnson
103	History	Mrs.Brown
104	English	Ms.Davis
105	Computer Science	Mr.Thompson
106	Geography	Null
107	Null	Mr.Stephan

B) Implement a Cursor to Calculate Total Salary of Employees in a Department.

Create a table called 'Inventory' containing information about products, including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory' table when ever a product is sold.

25. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers who made purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchase date in descending order, including the customer name, purchase date, and total amount spent.

B) Create a student data table for the following details,

Student Id	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	65	60	55
110	Jack	80	75	70

Display students with marks in Maths above 70.

- Show students with marks in Chemistry not in pass range (45 and above).
- Show students with marks in Maths or Chemistry above 80. Display students

with marks in all subjects below 50.

26. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers who made purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchase date in descending order, including the customer name, purchase date, and total amount spent.

B) Study the commands for views and execute the following queries using these commands:

- Create a view having e name and e city
- In the above view change the e city to 'Chennai' where e name is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

27. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers who made purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchase date in descending order, including the customer name, purchase date, and total amount spent.

B) Create the following tables in the "Company DB" database:

- Employees(employee_id,name,role)
- Departments(department_id,name,manager_id)
- Use the GRANT statement to give User1 SELECT permission on the Employee stable.
- Use the GRANT statement to give User2 INSERT and UPDATE permissions on the Department stable.
- Use the REVOKE statement to revoke the SELECT permission from User1 on the Department stable.

Use the REVOKE statement to revoke all permissions from User2 on the Employee stable.

28. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers who made purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchase date in descending order, including the customer name, purchase date, and total amount spent.

B) Create the following tables in the "Company DB" database:

- Employees(employee_id,name,salary)
- Departments(department_id,name,budget)
- Use the SAVE POINT statement to set a save point named "before_update" in the transaction.
- Update the salary of an employee in the Employees table.
- Use the ROLL BACK TO SAVE POINT statement to rollback the transaction to the save point named "before_update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.

29. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers who made purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchase date in descending order, including the

customer name, purchase date, and total amount spent.

B) a) Create a Student stable including three subject marks, date of birth. Write stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

30. A) Create a Purchases Table using the following data for executing the query given below,

Customer Id	Customer name	email	Purchase date	Purchase Amount	Purchase type
101	Alice	alice@gmail.com	15-02-24	50	Online
102	Bob	bob@gmail.com	16-02-24	25	In-store
103	Charlie	charlie@gmail.com	17-02-24	40	Online
101	Alice	alice@gmail.com	18-02-24	35	In-store
101	Alice	alice@gmail.com	19-02-24	20	Online

Union: Retrieve a list of all unique customers whom a de purchases either online or in-store, ordered alphabetically by their customer names.

Intersect: Find the customers who made purchases both online and in-store, displaying their names and email addresses.

Order by in My SQL: Display a list of all purchases made in-store, ordered by the purchased ate in descending order, including the customer name, purchase date, and total amount spent.

B) Implement a Cursor to Calculate Total Salary of Employees in a Department.

Create a table called 'Inventory' containing information about products, including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory 'table whenever a product is sold.

31. A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24
102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name
-

B) Create a student data table for the following details,

Student Id	Student Name	Maths	Physics	Chemistry
101	Alice	90	85	80
102	Bob	30	25	35
103	Charlie	50	45	55
104	David	70	65	60
105	Eve	40	35	30
106	Frank	45	40	35
107	Grace	55	50	45
108	Harry	20	15	10
109	Ivy	65	60	55
110	Jack	80	75	70

- Display students with marks in Maths above 70.
- Show students with marks in Chemistry not in pass range (45 and above).
- Show students with marks in Maths or Chemistry above 80. Display students with marks in all subjects below 50.

32.

A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24
102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name

B) Study the commands for views and execute the following queries using these commands:

- Create a view having e name and e city
- In the above view change the e city to 'Chennai' where e name is 'John'.
- Create a view having attributes from both the tables.

Update the above view and increase the salary of all employees of IT department by Rs.1000.

33.

A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24

102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name

B) Create the following tables in the "Company DB" database:

- Employees (employee_id,name,role)
- Departments (department_id,name,manager_id)
- Use the GRANT statement to give User1 SELECT permission on the Employee stable.
- Use the GRANT statement to give User2 INSERT and UPDATE permissions on the Department stable.
- Use the REVOKE statement to revoke the SELECT permission from User1 on the Department stable.
Use the REVOKE statement to revoke all permissions from User2 on the Employee stable.

34.

A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24

102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name

B) Create the following table sin the "Company DB" database:

- Employees (employee_id,name,salary)
- Departments (department_id,name,budget)
- Use the SAVE POINT statement to set a save point named "before_update" in the transaction.
- Update the salary o fan employee in the Employees table.
- Use the ROLL BACK TO SAVE POINT statement to rollback the transaction to the save point named "before_update" if the update was incorrect. If the update was correct, use the COMMIT statement to commit the transaction.

35.

A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24
102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name

B) a) Create a Students table including three subject marks, date of birth. Write stored functions for calculating total, average, and age.

b) Create a stored procedure that accepts a student ID as input and returns the name of the student.

36.

A) Create a Bank Operations Table using the following data,

Transaction ID	Customer name	Account No.	Operation Type	Amount	Transaction Date
101	Alice	ACC001	Deposit	1000	01-02-24
102	Bob	ACC002	Withdrawal	500	03-02-24
103	Alice	ACC001	Withdrawal	200	05-02-24
104	Charlie	ACC003	Deposit	1500	07-02-24
105	David	ACC004	Deposit	800	10-02-24
106	Eve	ACC005	Withdrawal	300	12-02-24

			1		
107	Alice	ACC001	Deposit	1200	15-02-24
108	Charlie	ACC003	Withdrawal	500	18-02-24
109	Bob	ACC002	Deposit	700	20-02-24
110	David	ACC004	Withdrawal	400	25-02-24

- Count the total number of operations (deposits and withdrawals) per account.
- Calculate Average Amount Deposited per Account with Customer Name
- Find Minimum Amount Deposited per Account with Customer Name Find Maximum Amount Withdrawn per Account with Customer Name

B) Implement a Cursor to Calculate Total Salary of Employees in a Department.

Create a table called 'Inventory' containing information about products, including 'Product ID', 'Product Name', and 'Quantity'. Design a trigger that decrements the 'Quantity' column in the 'Inventory' table whenever a product is sold.

Allocation Of Marks

Sl. No	Description	Marks
1	Aim(05), Program for the experiment from Part A (30)	35
2	Aim(05), Program for the experiment from Part B (30)	35
3	Execution of the experiment from (Part A OR Part B)	15
4	Output	10
5	Viva Voce	05
Total		100