

1. Create following tables with primary and foreign key and solve the queries given below  
Person (driver\_id, name, address)  
Car (license, model, year)  
Accident (report\_no, date\_acc, location)  
Owns (driver\_id, license)  
Participated (driver\_id, model, report\_no, damage\_amount)
  1. Insert records into each table
  2. Create a view named Driver\_Car\_View to show each driver's name and the car model they own.
  3. Display all records from the Driver\_Car\_View.
  4. Modify the existing view Driver\_Car\_View to include the car's year of manufacture.
  5. Create a sequence named seq\_report\_no starting from 6001 and increment by 5 for new accidents.
  6. Create a composite index on model and year columns of the Car table.
    - a. Create a view to show all drivers who participated in accidents with damage amount greater than ₹20,000.
  7. Create a sequence named seq\_driver\_id to auto-generate driver IDs starting from 201 and increment by 1.
  8. Insert a new record into Person using the Sequence.
  9. Create an index on the name column of the Person table for faster searching.
  10. Drop the index created on name column of Person.
  11. Create a view named Recent\_Accidents that shows accidents that occurred in the year 2025.
  12. Drop the view Recent\_Accidents.
  13. Create a synonym named acc\_info for the Accident table.
  14. Insert a new accident record using the acc\_info synonym.
  15. Create a unique index on license column of the Car table.
  16. Create a view to display all drivers who own cars manufactured after 2019.
  17. Create a sequence to generate accident report numbers automatically and use it in an INSERT statement.
  18. Create a synonym for the Owns table and use it to list all driver-car pairs.
  19. Create an index to improve the performance of queries filtering by damage\_amount.
2. Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory. Write a PL/SQL block of code for the following requirements:-Consider table Stud(Roll, Att, Status) Write a PL/SQL block for following requirement and handle the exceptions. Roll no. of student will be entered by user. Attendance of roll no. entered by user will be checked in Stud table. If attendance is less than 75% then display the message "Term not granted" and set the status in stud table as "D". Otherwise display message "Term granted" and set the status in stud table as "ND"

3. Create Customer and Account table and add rows shown below

**Customer Table**

C_Id	Cname	City
1	John	Nashik
2	Seema	Aurangabad
3	Amita	Nagar
4	Rakesh	Pune
5	Samata	Nashik
6	Ankita	Chandwad
7	Bhavika	Pune
8	Deepa	Mumbai
9	Nitin	Nagpur
10	Pooja	Pune

**Account Table**

C_Id	Acc_Type	Amount
1	Current	5000
2	Saving	20000
3	Saving	70000
4	Saving	50000
6	Current	35000
7	Loan	30000
8	Saving	50000
9	Saving	90000
10	Loan	8000
11	Current	45000

- 1) Show the cname, Acc\_Type, amount information of customer who is having an saving account.
2. Display the data using Natural, left and right join.
3. Display the information of customers living in the same city as of ‘pooja’.
4. Display the information of account, having less amount than average amount throughout the bank.
5. Display the C\_id having maximum amount in account.
6. Display the amount and acc\_type of those customers whose amount is the minimum amount of that Acc\_type.
7. Display the amount of those accounts whose amount is higher than amount of any saving account amount.

4.Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory.

Write a PL/SQL block of code for the following requirements:-

Schema:

1. Borrower(Rollin, Name, DateofIssue, NameofBook, Status)
2. Fine(Roll\_no,Date,Amt)

Accept roll\_no & name of book from user.

Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.

If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.

After submitting the book, status will change from I to R.

If condition of fine is true, then details will be stored into fine table.

5.Create following tables with primary and foreign key and solve the queries given below

student(S\_ID,name,dept\_name,tot\_cred)

instructor(T\_ID,name,dept\_name,salary)

course(course\_id,title,dept\_name,credits)

- i. Find the average salary of instructor in those departments where the average salary is more than Rs. 42000/-.
- ii. Increase the salary of each instructor in the computer department by 10%.
- iii. Find the names of instructors whose names are neither ‘Amol’ nor ‘Amit’.
- iv. Find the names of student which contains ‘am’ as its substring.
- v. Find the name of students from computer department that “DBMS” courses they take.

6.Cursors: (Implicit and Explicit Cursor). Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N\_RollCall with the data available in the table O\_RollCall. If the data in the first table already exist in the second table then that data should be skipped.

7. Write a Stored Procedure namely proc\_Grade for the categorization of student. If marks scored by students in examination is  $\leq 1500$  and  $\geq 990$  then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher Second Class Write a PL/SQL block for using procedure created with above requirement. Stud\_Marks(name, total\_marks) Result(Roll,Name, Class).

8. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius and area.

9. Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library\_Audit table. Frame the problem statement for writing Database Triggers of all types, in-line with above statement. The problem statement should clearly state the requirements.

10. create Collection “orderinfo“ which contains the documents given as below(Perform on Mongo Terminal)

```
{  
  cust_id:123  
  cust_name:"abc",  
  status:"A",  
  price:250  
}
```

- 1) find the average price for each customers having status 'A'
- 2) Display the status of the customers whose amount/price lie between 100 and 1000
- 3) Display the customers information without “\_id” .
- 4) create a simple index on onderinfo collection and fire the queries.

11. Create Collection “movies“ which contains the documents given as below(Perform on Mongo Terminal)

```
{  
  name: "Movie1",  
  type: "action",  
  budget:1000000  
  producer:{  
    name: "producer1",  
    address:"PUNE"  
  }  
}
```

- i. Find the name of the movie having budget greater than 1,00,000.
- ii. Find the name of producer who lives in Pune
- iii. Update the type of movie “action” to “horror”
- iv. Find all the documents produced by name “producer1” with their address

12. Consider following structure for Mongodb collection and write a query for following requirements in Mongodb

Teachers (Tname,dno,Experience,Salary,Date\_of\_Joining)

Department (Dno,Dname)

Students(Sname,Roll\_No,Class)

1. Write a query to create above collection insert some sample documents.
2. Find the information about all teachers of Dno=2 and having salary greater than or equal to 10,000/-
3. Find the student information having Roll\_no=2 or Sname='xyz'
4. Update student name whose Roll\_No=5
5. Delete all student whose Class is 'FE'

6. Apply index on Students Collection

13. Perform aggregation and Indexing using mongodb on below database

1. Create a database department

2. Create a collection as teacher with fields as name , department ,experience and salary

3. Display the department wise average salary.

4. Display the no. Of employees working in each department.

5. Display the department wise minimum salary.

6. Apply index and drop index

14. Collection “city” which contains the documents given as below(Perform on Mongo Terminal)

{

city:”pune”,

type:”urban”,

state:”MH”,

population:”5600000”

}

-using mapreduce, find statewise population

-using mapreduce, find citywise population

-using mapreduce, find typewise population.

15. Collection creation Student and insert following data in that:

Rollno:1,name:'Navin',subject:'DMSA',marks:78

Rollno:2,name:'anusha',subject:'OSD',marks:75

Rollno:3,name:'ravi',subject:'TOC',marks:69

Rollno:4,name:'veena',subject:'TOC',marks:70

Rollno:5,name:'Pravini',subject:'OSD',marks:80

Rollno:6,name: 'Reena',subject: 'DMSA',marks:50

Rollno:7,name:'Geeta',subject:'CN',marks:90

Rollno:8,name:'Akash',subject:'CN',marks:85

1. Write aggregate function to find Max marks of Each Subject.

2. Write aggregate function to find Min marks of Each Subject.

3. Write aggregate function to find Sum of marks of Each Subject.

4. Write aggregate function to find Avg marks of Each Subject.

5. Write aggregate function to find first record Each Subject.

6. Write aggregate function to find Last record of Each Subject.

7. Write aggregate function to find count number of records of each subject

8. Write aggregate function to find count number of records of each subject

16. Create database employee and create collection computer Define a map function to emit the "total" key and the "Salary" value for each document. Define a reduce function to sum the salaries associated with the "total" key. Execute the map-reduce operation on the "Computer" collection, specifying an output collection where the results will be stored.

17. Collection “orderinfo” which contains the documents given as below(Perform on Mongo Terminal)

{

cust\_id:123

cust\_name:”abc”,

status:”A”,

price:250

}

i. Display the name of the customer having the price between 250 and 450

ii. Increment the price by 10 for cust\_id: 123 and decrement the price by 5 for cust\_id: 124

iii. Remove any one of the field from the orderinfo collection.

iv. Find the name of the customer whose status is either A or price is 250 or both.

**18.Emp(emp\_id,ename, street, city)**

works(emp\_id,c\_id,ename, cname, sal)

Company(c\_id,cname, city)

Manager(mgr\_id, mgrname)

i. Modify the database so that a particular company (eg. ABC) now in Pune

ii. Give all managers of Mbank a 10% raise. If salary is >20,000, give only 3% raise.

iii. Find out the names of all the employees who works in ‘Bosch’ company in city Pune

iv. Delete all records in the works table for employees of a particular company (Eg, SBC Company) whose salary>50,000.

19.Create database Employee.

2.Create collection emp1 using createCollection method.

3.Insert 4 to 5 documents in emp1 collection.(eno,ename,address,sal)

4.display all documents.

5.Display all employess having salary greater than 5000

6.Display all employess having salary less than 15000

7.Display all employess having salary greater than 10000 and less than 20000.

8.Update sal of all employee with 10%

9.Delete employee having salary less than 5000.

10.Rename collection emp1 with employee1.

11.Display employee whose name start with n.

12.sort name in ascending order.

13.Create a new database with name Employee1.

14.Drop employee1 database.

15.Create collection emp1 using insert method.

16.Drop collection emp1.

20.Empl(e\_no, e\_name, post, pay\_rate)

Position(pos\_no, post)

Duty-alloc (pos\_no, e\_no, month,year, shift)

Implement the following SQL queries

i. Get duty allocation details for e\_no 123 for the first shift in the month of April 2003

ii. Get the employees whose rate of pay is > or equal rate of pay of employees 'Sachin'.

iii. Create a view for displaying minimum, maximum and average salary for all the posts.

iv. Get count of different employees on each shift having post ‘manager’.

21.Employee(employee\_name,street,city)

Works(employee\_name,company\_name,salary)

Company(company\_name,city)

Manages(employee\_name,manager\_name)

1. Find the names of employees who work for First Bank Coorporation.

2. Find the names and cities of residence of all employees who work for First Bank Coorporation

3. Find the names, street addresses, and cities of residence of all employees who work for First Bank Coorporation and earn more than \$10000.

4.Find all employees in the database who lives in the same cities as tha companies for which they work.

5.Find all employees in the database who lives in the same cities and on the same streets as do their manager.

6.Find all employees in the database who do not work for First Bank Coorporation

7.Give all employees of First Bank Coorporation” a 10% raise

8.Delete all tuples in the “Works” relation for employees of “Small Bank Coorporation”.

22.Employee(employee\_name,street,city)  
Works(employee\_name,company\_name,salary)  
Company(company\_name,city)  
Manages(employee\_name,manager\_name)

- 1) Find all employees in the database who earn more than each employee of Small Bank Corporation
- 2) Assume that the company is may be located in several cities. Find all companies located in every city in which Small Bank Coorporation is located.
- 3) Find all employees who earn more than the average salary of all employees of their companies.
- 4) Find the company that has the most employees.
- 5) Find the company that has the smallest payroll.
- 6) Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Coorporation.
- 7) Modify the database so that “Jones” now lives in Newtown.

23.Create a table emp with following fields and constraints  
Eno -(Constraint:- primary key and apply sequence starts with 101) ,Ename -(Constraint :- not null)  
Address —(Constraint :-default ‘Nashik’) ,Joindate,  
After table creation add field - Post in the emp table.  
Insert some data in emp table.Create Index on Ename field of employee table.  
Create View on employee table to show only Ename and Salary.