**1:** Draw an ER for Bank database with atleast 5 entities and convert them into tables.

Perform DDL on above converted tables.

1. Create tables with all constraints

2. Create views on any two tables using join conditions

3. Create index called CustomerId. Entries should be in ascending order by customer name.

4. Create sequence on Acctno.

**2:** Draw an ER for Company database with atleast 4 entities and convert them into tables.

Perform DDL on Above converted tables.

1. Create tables with all constraints

2. create views on any two tables using conditions

3. create index called EmployeeId for the department table. Entries should be in ascending order by department id and then by employee id within each department.

4. create sequence on Employee id.

create view info as

-> select \* from account

-> order by b\_name,balance;

**3: w**rite a trigger for **Library (bid, bname, doi, status)** to update the number of copies (noc) according to ISSUE & RETURN status on update or insert query. Increase the noc if status is RETURN, Decrease noc if status is ISSUE in **Library\_Audit table(bid,bname,noc,timestampofquery).** Write a trigger after update on Library such that if doi is more than 20 days ago then status should be FINE and in the Library\_Audit table fine should be equal to no. of days \* 10.

4: 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.

**5** Create a collection sites(url,dateofaccess). Write a MapReduce function to find the no. of times a site was accessed in a month.

6 Create tables CitiesIndia(**pincode**,nameofcity,earliername,area,population,avgrainfall)

Categories(Type,pincode) ***Note:- Enter data only in CitiesIndia***

Write PL/SQL Procedure & function to find the population density of the cities. If the population density is above 3000 then Type of city must be entered as High Density in Category table. Between 2999 to 1000 as Moderate and below 999 as Low Density. Error must be displayed for population less than 10 or greater than 25718.

**7** Write PL/SQL Procedure & function to find class [Distinction (Total marks from 1499 to 990) ,First Class( 899 to 900) Higher Second (899 to 825) ,Second,Pass (824 to 750) ] of a student based on total marks from table **Student (rollno, name, Marks1, Marks2, Marks3, Marks4, Marks5).**

**Use exception handling when negative marks are entered by user(Marks<0) or Marks more than 100 are entered by user.**. Store the result into Result table recording RollNo,total marks, and class for each student .

**8.** Draw ER for Library database with atleast 5 entities and convert them into tables.

Perform DDL on above converted tables.

1. Create tables with all constraints (Based on ERD cardinalities)

2. Create views on any two tables using join condition

3. Create index called Lib\_Index1. Entries should be in ascending order by Author name.

4. Create sequence on Bookid.

9.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)

3. Library (bid, bname, doi, status,noc)

4. transaction (tid,bid, bname, status)

1. Accept roll\_no & name of book from user.
2. Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.
3. If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.
4. After submitting the book, status will change from I to R.
5. Update the noc in library according to the transaction made. Increase the noc if status is RETURN, Decrease noc if status is ISSUE.
6. If condition of fine is true, then details will be stored into fine table.

10. Implement SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym for following relational schema:

Borrower(Rollin, Name, DateofIssue, NameofBook, Status)

* **MySQL**, there's no true synonym feature — but you can mimic it with a view or just use a shorter alias in queries.
* MySQL doesn't support standalone CREATE SEQUENCE, but you can simulate it by making Rollin auto-increment.

11 Design at least 10 SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query and View.

SELECT \* FROM loac

WHERE c\_name IN (

SELECT c\_name FROM account

WHERE b\_name = 'pune'

);

Joins: Inner,left,right,full

MySQL **does not support FULL OUTER JOIN** directly, but you can simulate it using UNION on left and right join.

Subqueries – 3

View – 3(Normal , where , Group By,order by)

12 Implement Indexing and querying with MongoDB using following example.

Students(stud\_id, stud\_name,stud\_addr,stud\_marks)

**13** Create the instance of the COMPANY which consists of the following tables:

**EMPLOYEE(Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Dno)**

**DEPARTEMENT(Dname, Dno, Mgr\_ssn, Mgr\_start\_date)**

**DEPT\_LOCATIONS(Dnumber, Dlocation)**

**PROJECT(Pname, Pnumber, Plocation, Dno)**

**WORKS\_ON(Essn, Pno, Hours)**

**DEPENDENT(Essn, Dependent\_name, Sex, Bdate, Relationship)**

Perform following queries

1. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name,address, and birth date.
2. Make a list of all project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.
3. Retrieve all employees whose address is in Houston, Texas.

SELECT \*

FROM employee

WHERE address LIKE '%Houston, Texas%';

1. Show the resulting salaries if every employee working on the ‘ProductX’ project is given a 10 percent raise.

update q4

-> set salary = salary \* 1.1;

14 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Solve following query:

1. Find the average account balance at each branch
2. Find no. of depositors at each branch.
3. Find the branches where average account balance > 12000.
4. Find number of tuples in customer relation.

15 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.

Solve following query:

1. Find the names of all branches in loan relation.
2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.
3. Find all customers who have a loan from bank.
4. Find their names,loan\_no and loan amount.

16 Implement Map reduce operation with following example using MongoDB

Students(stud\_id, stud\_name,stud\_addr,stud\_marks)

**AND**

Write a PL/SQL code to calculate total and percentage of marks of the students in four subjects.

17 Create following collection and using MongoDB implement all CRUD operations.

Orders( cust\_id, amount, status)

18 Implement all SQL DML opeartions with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.Solve following query:

1. Find all customers who have an account or loan or both at bank.
2. Find all customers who have both account and loan at bank.
3. Find all customer who have account but no loan at the bank.
4. Find average account balance at Akurdi branch.

19 Implement all SQL DML operations with operators, functions, and set operator for given schema:

Account(Acc\_no, branch\_name,balance)

branch(branch\_name,branch\_city,assets)

customer(cust\_name,cust\_street,cust\_city)

Depositor(cust\_name,acc\_no)

Loan(loan\_no,branch\_name,amount)

Borrower(cust\_name,loan\_no)

Solve following query:

1. Calculate total loan amount given by bank.
2. Delete all loans with loan amount between 1300 and 1500.
3. Delete all tuples at every branch located in Nigdi.

20 Create the following tables.

1. Deposit (actno,cname,bname,amount,adate)
2. Branch (bname,city)
3. Customers (cname, city)
4. Borrow(loanno,cname,bname, amount)

Add primary key and foreign key wherever applicable.Insert data into the above created tables.

1. Display account date of customers “ABC”.
2. Modify the size of attribute of amount in deposit
3. Display names of customers living in city pune.
4. Display name of the city where branch “OBC” is located.
5. Find the number of tuples in the *customer* relation

21 Create following tables:

1. Deposit (actno,cname,bname,amount,adate)
2. Branch (bname,city)
3. Customers (cname, city)
4. Borrow(loanno,cname,bname, amount)

Add primary key and foreign key wherever applicable. Insert data into the above created tables.

1. Display customer name having living city Bombay and branch city Nagpur
2. Display customer name having same living city as their branch city
3. Display customer name who are borrowers as well as depositors and having living city Nagpur.

22 Create the following tables.

1. Deposit (actno,cname,bname,amount,adate)
2. Branch (bname,city)
3. Customers (cname, city)
4. Borrow(loanno,cname,bname, amount)

Add primary key and foreign key wherever applicable.

Insert data into the above created tables.

1. Display loan no and loan amount of borrowers having the same branch as that of sunil.
2. Display deposit and loan details of customers in the city where pramod is living.
3. Display borrower names having deposit amount greater than 1000 and having the same living city as pramod.
4. Display branch and living city of ‘ABC’

23 Implement all Aggregation operations and types of indexing with following collection using MongoDB.

Employee(emp\_id, emp\_name,emp\_dept,salary)

24 Create the following tables.

1. Deposit (actno,cname,bname,amount,adate)
2. Branch (bname,city)
3. Customers (cname, city)
4. Borrow(loanno,cname,bname, amount)

Add primary key and foreign key wherever applicable. Insert data into the above created tables.

1. Display amount for depositors living in the city where Anil is living.
2. Display total loan and maximum loan taken from KAROLBAGH branch.
3. Display total deposit of customers having account date later than ‘1-jan-98’.
4. Display maximum deposit of customers living in PUNE.

25 Design and Implement any 5 query using MongoDB

1. Create a collection called ‘games’.
2. Add 5 games to the database. Give each document the following properties: name, gametype, score (out of 100), achievements
3. Write a query that returns all the games
4. Write a query that returns the 3 highest scored games.
5. Write a query that returns all the games that have both the ‘Game Maser’ and

the ‘Speed Demon’ achievements.

26 Write a PL/SQL code to calculate tax for an employee of an organization ABC and to display his/her name & tax, by creating a table under employee database as below:

Employee\_salary(emp\_no,basic,HRA,DA,Total\_deduction,net\_salary,gross\_Salary)

27 Create PL/SQL code block: Write a PL/SQL block of code for the following schema:

Borrower(Rollin, Name, DateofIssue, NameofBook, Status)

Fine(Roll\_no,Date,Amt)

Solve following queries:

1. Accept roll\_no & name of book from user.
2. Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.
3. If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.
4. After submitting the book, status will change from I to R.
5. If condition of fine is true, then details will be stored into fine table.

Use of Control structure and Exception handling is mandatory.

28 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.

29 Writ a PL/SQL procedure to find the number of students ranging from 100-70%, 69-60%, 59-50% & below 49% in each course from the student\_course table given by the procedure as parameter.

Schema: Student (ROLL\_NO ,COURSE, COURSE\_COD ,SEM ,TOTAL\_MARKS, PERCENTAGE)

30 Write a Stored Procedure namely proc\_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class .

Consider Schema as Stud\_Marks(name, total\_marks) and Result(Roll,Name, Class)

31.Create database :**Citydetails(\_id,name,area,population(total,Adults,seniorcitizens,sexratio), geography(avgtemp, avgrainfall, longitude, latitude))**

1. Find the total population in pune.

2. returns all city with total population greater than 10 million

3. returns the average populations for each city.

4. returns the minimum and maximum cities by population for each city.

32.Create database :**Citydetails(\_id,name,area,population(total,Adults,seniorcitizens,sexratio), geography (avgtemp, avgrainfall, longitude, latitude))**

1. Find area wise total population and sort them in increasing order.
2. Retrieve name and area where average rain fall is greater than 60
3. Create index on city and area find the max population in Mumbai
4. Create index on name.