

### **Problem Statement:-1**

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

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

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

Depositor(cust\_name,acc\_no)

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

Borrower(cust\_name,loan\_no)

Solve following query:

1. Create above tables with appropriate constraints like primary key, foreign key, check constraints, not null etc.
2. Find the names of all branches in loan relation.
3. Find all loan numbers for loans made at Pimpri Branch with loan amount > 12000.
4. Find all customers who have a loan from bank. Find their names, loan\_no and loan amount.
5. List all customers in alphabetical order who have loan from Akurdi branch.
6. Find all customers who have an account or loan or both at bank.
7. Find all customers who have both account and loan at bank.
8. Find average account balance at Pimpri branch.
9. Find the average account balance at each branch
10. Find the branches where average account balance > 12000.
11. Calculate total loan amount given by bank.

### **Problem Statement:-2**

1. 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  
Write a PL/SQL block for using procedure created with above requirement.

Stud\_Marks(name, total\_marks)

Result(Roll,Name, Class)

### **Problem Statement:-3**

1. Write a PL/SQL block of code using parameterized Cursor that will merge the data available in the newly created table N\_Roll\_Call with the data available in the table O\_Roll\_Call. If the data in the first table already exists in the second table then that data should be skipped

### **Problem Statement: -4**

#### **Create following collections and Perform MongoDB CRUD Operations.**

Teachers (Tname, dno, dname, experience, salary, date\_of\_joining )

Students(Sname, roll\_no, class)

1. Find the information about all teachers alphabetically.
2. Find the information about all teachers of the computer department
3. Find the information about all teachers of computer,IT,and e&TC department
4. Find the information about all teachers of computer,IT,and E&TC department having salary greater than or equal to 10000/-
6. Find the student information having roll\_no = 2 or Sname=xyz
7. Update the experience of teacher-praveen to 10 years, if the entry is not available in database consider the entry as new entry.
9. Update the department of all the teachers working in IT department to COMP
10. find the teacher's name and their experience from teachers' collection
11. Delete all the documents from teacher's collection having IT dept.
12. display with pretty() method, the first 3 documents in teacher's collection in ascending order.

### **Problem Statement: -5**

#### **MongoDB Aggregation**

You have been given a dataset containing details about different books. Each book has the following fields:

- title: The title of the book
- author: The author of the book
- genre: The genre of the book (e.g., Fiction, Non-Fiction, Mystery, Sci-Fi)
- price: The price of the book
- published\_date: The date the book was published.

The data has been stored in a MongoDB collection named books.

Using the MongoDB aggregation framework, perform the following tasks:

1. Find the average price of all books.
2. Find the count of books in each genre.
3. For each genre, find the most expensive book.
4. Find the authors who have written maximum books.
5. Sort the books by published\_date in descending order.
6. Sort the price in ascending order.
7. create an index on title of the book and describe the index details

### **Problem Statement: -6**

A retail company maintains a MongoDB collection named customer. Each document in this collection represents a purchase and contains fields such as cid (Customer ID), cname (Customer Name), amount (Amount spent on product purchase), and product\_name (Product Name).

Implement a MapReduce function in MongoDB to analyze the customer collection and produce a summarized report that displays **the total amount spent by each customer on product** purchases.

### **Problem Statement: -7**

**Unnamed PL/SQLcode block: Use of Control structure and Exception handling is mandatory.**

Suggested Problem statement:

Consider Tables:

1. Borrower (Roll\_no, Name, Date\_of\_Issue, Name\_of\_Book, Status)

2. Fine (Roll\_no, Date, Amt)

- Accept Roll\_no and 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 and 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.

### **Problem Statement: -8**

Develop a student database to efficiently manage and retrieve student records (Student id, Student Name, Class, address, grades, and enrolment details, subject name, attendance.

- Create Views to provide summarized insights into student performance and attendance. (Consider the attributes which shows attendance of students while creating view)
- Create Sequences to generate unique student IDs.
- Create an index on a table using student name.