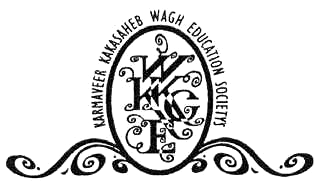
**K. K. Wagh Institute of Engineering Education and Research, Nashik.**

**Department of Computer Engineering**

**Academic Year:** 2019 – 2020  **Semester:** I

**Course Name:** Database Management System Lab  **Course Code:** 310247

**Class: T**E  **Division: A&**B

**Name of the Faculty:** A.V.Tavare & S. K. Gondhalekar

**Detail Assignment List**

|  |  |
| --- | --- |
| **Asg No** | **Problem statements** |
| 1. | **Install MYSQL on Ubuntu and List and Explain its features.** |
| 2. | **Consider following Database Schemas**  **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:**  Create above tables with appropriate constraints like primary key, foreign key, check constrains, not null etc.  1. Find the names of all branches in loan relation.  2. Find all loan numbers for loans made at Shivaji nagar Branch with loan amount > 12000.  3. Find all customers who have a loan from bank. Find their names,loan\_no and loan amount.  4. List all customers in alphabetical order who have loan from Shivaji nagar branch.  5. Find all customers who have an account or loan or both at bank.  6. Find all customers who have both account and loan at bank.  7. Find all customer who have account but no loan at the bank.  8. Find average account balance at Shivaji nagar branch.  9. Find the average account balance at each branch  10. Find no. of depositors at each branch.  11. Find the branches where average account balance > 12000.  12. Find number of tuples in customer relation.  13. Calculate total loan amount given by bank.  14. Delete all loans with loan amount between 1300 and 1500.  15. Delete all tuples at every branch located in Sharanpur road. |
| 3. | **Consider the given Database Schema:**  **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 all employees who work for First Bank Corporation.  2. Find the names and cities of residence of all employees who work for First Bank Corporation  3. Find the names, street addresses, and cities of residence of all employees  who work for First Bank Corporation and earn more than $10,000.  4. Find all employees in the database who live in the same cities as the companies for which they work.  5. Find all employees in the database who live in the same cities and on the same streets as do their managers.  6. Find all employees in the database who do not work for First Bank Corporation.  7. Find all employees in the database who earn more than each employee of Small Bank Corporation.  8. Assume that the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.  9. Find all employees who earn more than the average salary of all employees of their company.  10. Find the company that has the most employees.  11. Find the company that has the smallest payroll.  12. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation. |
| 4. | Consider the given relational table:  **employee(empno , empname, designation, city, salary, zipcode, county)**  1. Creates a sequence used to generate employee numbers for the empno column of the emp table.  2. Create an Index on county.  3. Find the zipcode whose county = 071 and check whether the query uses the Index and write your observation.  4. Create a view for employees having salary < 50000 and stays in ‘Mumbai’ |
| 5. | Consider the given database schema:  **Student (studentid , studentname,instructorid,studentcity)**  **Instructor(instructorid,Instructorname,instructorcity,specialization)**  **Use all types of Joins**   1. Find the instructor of each student. 2. Find the student who is not having any instructor. 3. Find the student who is not having any instructor as well as instructor who is not having student. 4. Find the students whose instructor’s specialization is computer. 5. Create a view containing total number of students whose instructor belongs to “Pune**”.** |
| 6. | Create a database with following schemas in mysql - **Group C**  Student(Roll,Name,Deptid)  Write a menu driven program in Java/Python to connect with a database using Jdbc-Odbc Connectivity  1. to display the student’s details, 2. to insert the student’s details  3. to Update students record 4. to delete a student record |
| 7. | Create a database with following schemas  **Borrower(Rollin, Name, DateofIssue, NameofBook, Status) & Fine(Roll\_no,Date,Amt)**  1. Write a PL/SQL block to accept input for Borrower table.  2. Write a PL/SQL block using control structures to calculate fine by using the following rules:  a. check the number of days (from date of issue), if days are between 15 to 30 then fine  amount will be Rs 5per day  b. If no. of days>30, per day fine will be Rs 50 per day  c. 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. |
| 8. | Create two tables O\_Roll(Rollno,Name,DOB,Phone,address)  N\_Roll(Rollno,Name,DOB,Phone,address)  Write a PLSQL block using various types of cursor(implicit,Explicit,For, Parameterized) to merge records from O\_Roll table with that of N\_Roll in such a way duplicate records are to be eliminated. |
| 9. | Create a Library database with the schema Books(AccNo,Title,Author,Publisher,Count).   1. Create a table Library\_Audit with same fiels as of Books. 2. Create a before trigger to insert records into Librry\_Audit table if there is deletion in Books table. 3. Create a after trigger to insert records into Librry\_Audit table if there is updation in Books table. |
|  | **Group B** |
| 10. | Install MongoDb on Ubuntu and List and Explain its features**.** |
| 11. | Implement the following MongoDb Query   1. Create a collection named books. 2. Insert 5 records with field TITLE,DESCRIPTION,BY,URL,TAGS AND LIKES 3. Insert 1 more document in collection with additional field of user name and comments. 4. Display all the documents whose title is 'mongodb'. 5. Display all the documents written by 'john' or whose title is 'mongodb'. 6. Display all the documents whose title is 'mongodb' and written by 'john'. 7. Display all the documents whose like is greater than 10. 8. Display all the documents whose like is greater than 100 and whose title is either 'mongodb' or written by 'john'. 9. Update the title of 'mongodb' document to 'mongodb overview' 10. Delete the document titled 'nosql overview'. 11. Display exactly two documents written by 'john'. 12. Display the second document published by 'john'. 13. Display all the books in the sorted fashion. 14. Insert a document using save method. |
| 12. | Create the collection Books having the following fields TITLE, DESCRIPTION, BY, URL, TAGS AND LIKES.  Implement the following Aggregation and Indexing Queries   1. Find the number of books published by john. 2. Find books which have minimum likes and maximum likes published by john. 3. Find the average number of likes of the books published by john. 4. Find the first and last book published by john.. 5. Create an index on author name.   Display the books published by john and check if it uses the index which we have created |
| 13. | Create the following schema  Order (id, amount ,status)   |  |  |  | | --- | --- | --- | | Cus id | Amount | Status | | A1 | 400 | P | | B1 | 300 | D | | A1 | 200 | F | | C1 | 200 | F | | B1 | 700 | P | | B1 | 800 | P |   Status: P=”Pending”, D= “Delivered”, F= “Failed”  Implement the following using Map Reduce function   1. Find the sum of amount of each customer whose status is P 2. Find the average amount of each customer 3. Find the min amount of each customer 4. Find the max amount of each customer whose status is F |
| 14. | Create employee array objects containing employee id , name, designation, salary using json and write a program in python to read and display the employee information. |
| 15. | Create the following collection in mongodb  Student(rollno, name, address, contact\_no, department)  Write a menu driven program in python to connect with the database and to perform the following operation   1. Insert details of the student 2. Update the address of John to Dwarka 3. Display the students information 4. Delete a student’s record |
| 16. | Mini Project. |