

Assignment No 3

Asg No	Problem statements
1.	<p>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 constraints, not null etc.</p> <ol style="list-style-type: none"> Find the names of all branches in loan relation. Find all loan numbers for loans made at Shivaji nagar Branch with loan amount > 12000. Find all customers who have a loan from bank. Find their names,loan_no and loan amount. List all customers in alphabetical order who have loan from Shivaji nagar branch. Find all customers who have an account or loan or both at bank. Find all customers who have both account and loan at bank. Find all customer who have account but no loan at the bank. Find average account balance at Shivaji nagar branch. Find the average account balance at each branch Find no. of depositors at each branch. Find the branches where average account balance > 12000. Find number of tuples in customer relation. Calculate total loan amount given by bank. Delete all loans with loan amount between 1300 and 1500. Delete all tuples at every branch located in Sharanpur road.
2.	<p>Consider the given relational table: employee(empno , empname, designation, city, salary, zipcode, county)</p> <ol style="list-style-type: none"> Creates a sequence used to generate employee numbers for the empno column of the emp table. Create an Index on county. Find the zipcode whose county = 071 and check whether the query uses the Index and write your observation. Create a view for employees having salary < 50000 and stays in 'Mumbai'
3.	<p>Consider the given database schema: Student (studentid , studentname,instructorid,studentcity) Instructor(instructorid,Instructorname,instructorcity,specialization) Use all types of Joins</p> <ol style="list-style-type: none"> Find the instructor of each student. Find the student who is not having any instructor. Find the student who is not having any instructor as well as instructor who is not having student. Find the students whose instructor's specialization is computer. Create a view containing total number of students whose instructor belongs to "Pune".
4.	<p>Create a database with following schemas Borrower(Rollin, Name, DateofIssue, NameofBook, Status) & Fine(Roll_no,Date,Amt)</p> <ol style="list-style-type: none"> Write a PL/SQL block to accept input for Borrower table. Write a PL/SQL block using control structures to calculate fine by using the following rules: <ol style="list-style-type: none"> 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

	<p>c. for days less than 30, Rs. 5 per day.</p> <p>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.</p>
5.	<p>Create two tables O_Roll(Rollno,Name,DOB,Phone,address) N_Roll(Rollno,Name,DOB,Phone,address)</p> <p>Write a PL/SQL 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.</p>
6.	<p>Create a Library database with the schema Books(AccNo,Title,Author,Publisher,Count).</p> <ol style="list-style-type: none"> Create a table Library_Audit with same fiels as of Books. Create a before trigger to insert records into Librry_Audit table if there is deletion in Books table. Create a after trigger to insert records into Librry_Audit table if there is updation in Books table.
7.	<p>Create a procedure called USER_QUERY_EMP that accepts three parameters. Parameter p_myeno is of IN parameter mode which provides the empno value. The other two parameters p_myjob and p_mysal are of OUT mode. The procedure retrieves the salary and job of an employee with the provided employee number and assigns those to the two OUT parameters respectively. The procedure should handle the error if the empno does not exist in the EMP table by displaying an appropriate message. Use bind variables for the two OUT Parameters. Compile the code, invoke the procedure, and display the salary and job title for employee number 7839. Do the same for employee number 7123.</p>
8.	<p>Create a function named USER_ANNUAL_COMP that has three parameters p_eno, p_sal and p_comm for passing on the values of an employee number, the current salary and commission of the employee respectively. The function calculates and returns the annual compensation of the employee by using the following formula. $\text{annual_compensation} = (\text{p_sal} + \text{p_comm}) * 12$ If the salary or commission value is NULL then zero should be substituted for it. Give a call to USER_ANNUAL_COMP from a SELECT statement, against the EMP table.</p>
9.	<p>Create a function named USER_VALID_DEPTNO that has a single parameter p_dno to accept a department number and returns a BOOLEAN value. The function returns TRUE if the department number exists in the DEPT table else it returns FALSE</p>
10.	<p>Create a table named salaryLog with the appropriate columns and insert the empno, new grade, old salary and new salary values in salaryLog table if the grade of an employee changes. Create a trigger named TR_CHECK_GRADE that will be fired when a user modifies the EMP table. It will check whether the grade has changed by making use of the SALGRADE table. (Grade is dependent on Salary.) If grade is changed, the trigger will log the corresponding employee number, old salary, new salary and new grade into salaryLog table. Test the working of the trigger by firing an appropriate DML query.</p>