|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Database Management Systems Lab** | | 0 | 0 | 4 | 2 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** |  | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the concept of DBMS and ER Modelling.

2. To explain normalization, Query optimization and relational algebra.

3. To apply concurrency control, recovery, security and indexing for real time data.

**Course Outcomes**

**CO 1** Explain the terminologies, features and models of database systems.

**CO 2** Apply various disk storage, Indexing and hashing techniques for data storage.

**CO 3** Formulate SQL queries using relational algebra and relational calculus.

**CO 4** Apply normalization theory to database design.

**CO 5** Develop database application design and its implementation including integrity, constraints, transaction management and concurrent control algorithms.

**CO 6** Apply No SQL database concepts on real time data.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program**  Shape**Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** |  | 3 | 2 | 2 |  |  |  |  |  |  |  |  | 2 | 3 |  |
| **CO 2** |  | 3 | 2 | 2 |  |  |  |  |  |  |  |  | 2 | 3 |  |
| **CO 3** |  | 3 | 2 | 3 |  |  |  |  |  |  |  |  | 2 | 3 |  |
| **CO 4** |  | 3 | 2 | 3 |  |  |  |  |  |  |  |  | 2 | 3 |  |
| **CO 5** |  | 3 | 2 | 2 |  |  |  |  |  |  |  |  | 2 | 2 |  |
| **CO 6** |  | 2 | 1 | 1 |  |  |  |  |  |  |  |  | 2 | 1 |  |
| **Average** |  | 2.8 | 1.8 | 2.1 |  |  |  |  |  |  |  |  | 2 | 2.5 |  |

**1 – Weakly Mapped (Low)** **2 – Moderately Mapped (Medium)**

**3 – Strongly Mapped (High)                  “\_” means there is no correlation**

**List of Experiments**

**Experiment 1:**

**Title:** Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students’ transcripts.

1. The university keeps track of each student’s name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, gender, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student’s permanent address and to the student’s last name. Both Social Security number and student number have unique values for each student.

a. Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department.

b. Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course.

c. Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

d. A grade report has a student, section, letter grade, and numeric grade (0,1, 2, 3, or 4).

Design an Entity-Relationship diagram. Specify key attributes of each entity type, and structural constraints on each relationship type. Note any unspecified requirements and make appropriate assumptions to make the specification complete.

**Experiment 2**

Title. Consider the following set of requirements for a Company database that is used to keep track of employee.

The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

1. A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

1. We store each employee’s name, Social Security number,2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).

1. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.

Design an Entity-Relationship diagram for the company database.

**Experiment 3**

**Title: To understand DDL and DML Command**

**Objective:** To understand the concept of designing issue related to the database with creating, populating the tables. To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

1. **Create the tables for** Company database as per ER diagram of Exp 2.

TABLE 1: EMPLOYEE

[ Fname VARCHAR(15) NOT NULL,

Minit CHAR,

Lname VARCHAR(15) NOT NULL,

Ssn CHAR(9) NOT NULL,

Bdate DATE,

Address VARCHAR(30),

Sex CHAR,

Salary DECIMAL(10,2),

Super\_ssn CHAR(9),

Dno INT NOT NULL,

PRIMARY KEY (Ssn),

FOREIGN KEY (Super\_ssn) REFERENCES EMPLOYEE(Ssn),

FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)

]

TABLE 2: DEPARTMENT

[Dname VARCHAR(15) NOT NULL,

Dnumber INT NOT NULL,

Mgr\_ssn CHAR(9) NOT NULL,

Mgr\_start\_date DATE,

PRIMARY KEY (Dnumber),

UNIQUE (Dname),

FOREIGN KEY (Mgr\_ssn) REFERENCES EMPLOYEE(Ssn) );

]

 TABLE 3: DEPT\_LOCATIONS

( Dnumber INT NOT NULL,

Dlocation VARCHAR(15) NOT NULL,

PRIMARY KEY (Dnumber, Dlocation),

FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );

TABLE 4: PROJECT

( Pname VARCHAR(15) NOT NULL,

Pnumber INT NOT NULL,

Plocation VARCHAR(15),

Dnum INT NOT NULL,

PRIMARY KEY (Pnumber),

UNIQUE (Pname),

FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );

TABLE 5: WORKS\_ON

( Essn CHAR(9) NOT NULL,

Pno INT NOT NULL,

Hours DECIMAL(3,1) NOT NULL,

PRIMARY KEY (Essn, Pno),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),

FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );

TABLE 6:  DEPENDENT

( Essn CHAR(9) NOT NULL,

Dependent\_name VARCHAR(15) NOT NULL,

Sex CHAR,

Bdate DATE,

Relationship VARCHAR(8),

PRIMARY KEY (Essn, Dependent\_name),

FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );

1. **Insert the following data into their respective tables of Company database.**

|  |  |  |  |
| --- | --- | --- | --- |
| DEPARTMENT | | | |
| DNAME | DNUMBER | MGRSSN | MGRSTARTDATE |
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EMPLOYEE | | | | | | | | |
| FNAME | LNAME | SSN | BDATE | ADDRESS | SEX | SALARY | SUPERSSN | DNO |
| John | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston TX | M | 30000 | 333445555 | 5 |
| Franklin | Wong | 333445555 | 1965-12-08 | 638 Voss, Houston TX | M | 40000 | 888665555 | 5 |
| Alicia | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring TX | F | 25000 | 987654321 | 4 |
| Jennifer | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire TX | F | 43000 | 888665555 | 4 |
| Ramesh | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble TX | M | 38000 | 333445555 | 5 |
| Joyce | English | 453453453 | 1972-07-31 | 5631 Rice, Houston TX | F | 25000 | 333445555 | 5 |
| Ahmad | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston TX | M | 25000 | 987654321 | 4 |
| James | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston TX | M | 55000 | null | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| PROJECT | | | |
| PNAME | PNUMBER | PLOCATION | DNUM |
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

|  |  |  |
| --- | --- | --- |
| WORKS\_ON | | |
| ESSN | PNO | HOURS |
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DEPENDENT | | | | |
| ESSN | DEPENDENT\_NAME | SEX | BDATE | RELATIONSHIP |
| 333445555 | Alice | F | 1986-04-04 | Daughter |
| 333445555 | Theodore | M | 1983-10-25 | Son |
| 333445555 | Joy | F | 1958-05-03 | Spouse |
| 987654321 | Abner | M | 1942-02-28 | Spouse |
| 123456789 | Michael | M | 1988-01-04 | Son |
| 123456789 | Alice | F | 1988-12-30 | Daughter |
| 123456789 | Elizabeth | F | 1967-05-05 | Spouse |

|  |  |
| --- | --- |
| DEPT\_LOCATIONS | |
| DNUMBER | DLOCATION |
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Houston |
| 5 | Sugarland |

**Experiment 4**:

**Title:  To understand and apply the concept of Constraints.**

**Objective:** To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

**1. Create the tables described below:**

**Table name:   CLIENT\_MASTER**

**Description:** used to store client information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Constraints** |
| CLIENTNO | Varchar | 6 | Primary key / first letter must start with ‘C’ |
| NAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 |  |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 15 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 15 |  |
| BALDUE | Decimal | 10,2 |  |

**Table Name:   PRODUCT\_MASTER                                                                                                                                                                     Description:** used to store product information

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| PRODUCTNO | Varchar | 6 | Primary Key/ first letter must start with ‘P’ |
| DESCRIPTION | Varchar | 15 | Not Null |
| PROFITPERCENT | Decimal | 4,2 | Not Null |
| UNIT MEASURE | Varchar | 10 | Not Null |
| QTYONHAND | Integer | 8 | Not Null |
| REORDERL VL | Integer | 8 | Not Null |
| SELLPRICE | Decimal | 8,2 | Not Null |
| COSTPRICE | Decimal | 8,2 | Not Null |

**Table Name:    SALESMAN\_MASTER**

**Description:** used to store salesman information working for the company.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column name** | **data type** | **Size** | **Attributes** |
| SALESMANNO | Varchar | 6 | Primary Key/ first letter must start with ‘S’ |
| SALESMANNAME | Varchar | 20 | Not Null |
| ADDRESS 1 | Varchar | 30 | Not Null |
| ADDRESS 2 | Varchar | 30 |  |
| CITY | Varchar | 20 |  |
| PINCODE | Integer | 8 |  |
| STATE | Varchar | 20 |  |
| SALAMT | Real | 8,2 | Not Null , Cannot be 0 |
| TGTTOGET | Decimal | 6,2 | Not Null , Cannot be 0 |
| YTDSALES | Double | 6,2 | Not Null |
| REMARKS | Varchar | 60 |  |

1. **Insert the following data into their respective tables:**
2. Data for **CLIENT\_MASTER**  table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Client no | Name | City | Pincode | State | BalDue |
| C00001 | Ivan bayross | Mumbai | 400054 | Maharashtra | 15000 |
| C00002 | Mamta muzumdar | Madras | 780001 | Tamil nadu | 0 |
| C00003 | Chhaya bankar | Mumbai | 400057 | Maharashtra | 5000 |
| C00004 | Ashwini  joshi | Bangalore | 560001 | Karnataka | 0 |
| C00005 | Hansel colaco | Mumbai | 400060 | Maharashtra | 2000 |
| C00006 | Deepak sharma | Mangalore | 560050 | Karnataka | 0 |

1. Data for  **PRODUCT**\_**MASTER** table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Product  No | Description | Profit percent | Unit measure | Quantity  On  hand | Recorder  Level | Sell  Price | Cost  Price |
| P00001 | T-Shirt | 5 | Piece | 200 | 50 | 350 | 250 |
| P0345 | Shirts | 6 | Piece | 150 | 50 | 500 | 350 |
| P06734 | Cotton jeans | 5 | Piece | 100 | 20 | 600 | 450 |
| P07865 | Jeans | 5 | Piece | 100 | 20 | 750 | 500 |
| P07868 | Trousers | 2 | Piece | 150 | 50 | 850 | 550 |
| P07885 | Pull Overs | 2.5 | Piece | 80 | 30 | 700 | 450 |
| P07965 | Denim jeans | 4 | Piece | 100 | 40 | 350 | 250 |
| P07975 | Lycra tops | 5 | Piece | 70 | 30 | 300 | 175 |
| P08865 | Skirts | 5 | Piece | 75 | 30 | 450 | 300 |

1. Data for **SALESMAN\_MASTER** table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Salesman No** | **Name** | **Address1** | **Address2** | **City** | **Pin Code** | **State** |
| S00001 | Aman | A/14 | Worli | Mumbai | 400002 | Maharashtra |
| S00002 | Omkar | 65 | Nariman | Mumbai | 400001 | Maharashtra |
| S00003 | Raj | P-7 | Bandra | Mumbai | 400032 | Maharashtra |
| S00004 | Ashish | A/5 | Juhu | Mumbai | 400044 | Maharashtr(a |

1. **Exercise on retrieving records from a table.**   
   a.    Find out the names of all the clients.   
   b.    Retrieve the entire contents of the Client\_Master table.   
   c.    Retrieve the list of names, city and the state of all the clients.   
   d.    List the various products available from the Product\_Master table.   
   e.    List all the clients who are located in Mumbai.   
   f.     Find the names of salesman who have a salary equal to Rs.3000.
2. **Exercise on updating records in a table**   
   a.     Change the city of ClientNo ‘C00005’ to ‘Bangalore’.   
   b.     Change the BalDue of ClientNo ‘C00001’ to Rs.1000.   
   c.     Change the cost price of ‘Trousers’ to rs.950.00.   
   d.     Change the city of the salesman to Pune.
3. **Exercise on deleting records in a table**   
   a.     Delete all salesman from the Salesman\_Master whose salaries are equal to Rs.3500.

b.     Delete all products from Product\_Master where the quantity on hand is equal to 100.   
c.     Delete from Client\_Master where the column state holds the value ‘Tamil Nadu’.

1. **Exercise on altering the table structure**   
   a.     Add a column called ‘Telephone’ of data type integer to the Client\_Master table.   
   b.    Change the size off SellPrice column in Product \_Master to 10, 2.
2. **Exercise on deleting the table structure along with the data**   
   a.    Destroy the table Client\_Master along with its data.

**EXPERIMENT-5**

**Title:  To understand and use SQL Sub-Query**

**Objective:** To understand the use of sql subquery.

**1. Create the following table.**

Supplier-(scode,sname,scity,turnover)

Part-(pcode,weigh,color,cost,sellingprice)

Supplier\_Part-(scode,pcode,qty)

**2. Populate the table**

**3. Write appropriate SQL Statement for the following:**

1. Get the supplier number and part number in ascending order of supplier number.

2. Get the details of supplier who operate from Bombay with turnover 50.

3. Get the total number of supplier.

4. Get the part number weighing between 25 and 35.

5. Get the supplier number whose turnover is null.

6. Get the part number that cost 20, 30 or 40 rupees.

7. Get the total quantity of part 2 that is supplied.

8. Get the name of supplier who supply part 2.

9. Get the part number whose cost is greater than the average cost.

10. Get the supplier number and turnover in descending order of turnover.

**EXPERIMENT-6**

**Title:  Use of Inbuilt functions and relational algebra operation**

**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

     Write and execute the following queries using the Relational Algebra on the COMPANY

          database schema.

1. Retrieve the names of all employees in department 5 who work more than 10 hours
2. per week on the ‘ProductX’ project.
3. List the names of all employees who have a dependent with the same first name as
4. themselves.
5. Find the names of employees who are directly supervised by ‘Franklin Wong’.
6. Retrieve the names of employees who work on every project.
7. Retrieve the names of employees who do not work on any project.
8. Retrieve the names and addresses of all employees who work on at least one project
9. located in Houston but whose department has no location in Houston.
10. Retrieve the last names of all department managers who have no dependents.

**EXPERIMENT-7**

**Title:  Use of Inbuilt functions and relational algebra operation**

**Objective:** To understand the use of inbuilt function and relational algebra with sql query.

1. **Create the following two tables (EMP and DEPT)**

**EMP TABLE**

     EMPNO      ENAME       JOB                MGR          HIREDATE          SAL         COMM           DEPTNO

       ----------      ----------      ---------            ----------        --------- ---------- ---------- ----------

      7369           SMITH       CLERK             7902            17-DEC-80            500           800                  20

      7499          ALLEN       SALESMAN     7698            20-FEB-81           1600           300                 30

      7521          WARD        SALESMAN      7698           22-FEB-81           1250          500                  30

      7566           JONES       MANAGER        7839           02-APR-81          2975                                  20

      7654          MARTIN     SALESMAN      7698           28-SEP-81           1250          1400                30

      7698          BLAKE       MANAGER        7839          01-MAY-81         2850                                  30

      7782          CLARK       MANAGER        7839          09-JUN-81          2450                                   10

       7788       SCOTT        ANALYST         7566              09-DEC-82       3000                                    20

       7839       KING           PRESIDENT                           17-NOV-81       5000                                    10

       7844       TURNER     SALESMAN       7698            08-SEP-81         1500            0                      30

       7876       ADAMS       CLERK               7788            12-JAN-83         1100                                   20

       7900       JAMES         CLERK               7698            03-DEC-81        950                                     30

       7902       FORD           ANALYST         7566            03-DEC-81        3000                                   20

       7934      MILLER        CLERK              7782             23-JAN-82        1300                                   10

**DEPT TABLE**

      DEPTNO                  DNAME                   LOC

       ----------                  --------------                -------------

         10                      ACCOUNTING            NEW YORK

         20                      RESEARCH                 DALLAS

         30                      SALES                          CHICAGO

         40                      OPERATIONS             BOSTON

Write the Queries for the following using In-built functions.

1. Retrieve average salary of all employees.
2. Retrieve the number of employees.
3. Retrieve distinct number of employee.
4. Retrieve total salary of employee group by job.
5. Display the employee information with maximum salary.
6. Find the highest paid employee in department 10.
7. List the emps whose sal is equal to the average of max and minimum.
8. List the emps who joined in the company on the same date.
9. Display the employee names in upper and lower case.
10. find the date of 3 days later from hiredate.

**EXPERIMENT-8**

**Title: Use of different SQL clauses and join**

**Objective:** To understand the use of group by and having clause and execute the SQL commands using JOIN

1.Consider the following schema:

**Student (sid, sname, age)**

**Match (mid, mname, venue)**

**Play (sid, mid, day(date))**

1. Populate all the tables.
2. nFind all information of students who have played match number B10.
3. Find the name of matches played by Amit.
4. Find the names of students who have played a match in Delhi.
5. Find the names of students who have played at least one match.
6. Find the ids and names of students who have played two different matches on the same day.
7. Find the ids of students who have played a match in Delhi or Mumbai.
8. Find the average age of students.

**EXPERIMENT-9**

**Title:  To understand the concepts of Views.**

**Objective:** Students will be able to implement the concept of views.

**1. Create table of table name: EMPLOYEES and add 6 rows**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| Salary | Number | 25 | NN |
| Department\_id | Character | 10 |  |

**2. Execute the following view related queries:**

1. Create View of name emp\_view and the column would be Employee\_id, Last\_Name, salary and department\_id only.
2. Insert values into view(remove the NOT NULL constraint and then insert values):
3. Modify, delete and drop operations are performed on view.
4. Creates a view  named salary\_view. The view shows the employees in department 20 and their annual salary.

**EXPERIMENT-10**

**Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.**

1. A view that has the department name, manager name, and manager salary for every department.
2. A view that has the employee name, supervisor name, and employee salary for each employee who works in the ‘Research’ department.
3. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
4. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

**EXPERIMENT-11**

**Title: To understand the concepts of Index.**

**Objective:** Students will be able to implement the concept of index.

**Create table of table name: EMPLOYEES and add 6 rows**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Width | Attributes |
| Employee\_id | Character | 10 | PK |
| First\_Name | Character | 30 | NN |
| Last\_Name | Character | 30 | NN |
| DOB | Date |  |  |
| Salary | Number | 25 | NN |
| Department\_id | Character | 10 |  |

**1. Execute the following index related queries:**

1. Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id
2. Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.
3. Create a reverse index on employee\_id column of the EMPLOYEES.
4. Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.
5. Create Function-based indexes defined on the SQL functions UPPER(column\_name) or LOWER(column\_name) to facilitate case-insensitive searches(on column Last\_Name).
6. Drop the function based index on column Last\_Name.

**EXPERIMENT-12**

**Title: To understand the concepts of Sequence.**

**Objective:** Students will be able to implement the concept of sequence.

1. Create a sequence by name EMPID\_SEQ starting with value 100 with an interval of 1.
2. Write a SQL command for finding the current and the next status of  EMPID\_SEQ.
3. Change the Cache value of the sequence EMPID\_SEQ to 20 and maxvalue to 1000.
4. Insert values in employees table using sequences for employee\_id column.
5. Drop sequence EMPID\_SEQ.
6. Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

**EXPERIMENT-13**

**Title: To understand the concepts of PL/SQL programming.**

**Objective:** Students will be able to implement the basic concepts of Pl/SQL.

1. Write a PL/SQL code to accept the value of A, B & C display which is greater.
2. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.
3. Write a PL/SQL code block to find the factorial of a number.
4. Write a PL/SQL program to generate Fibonacci series.
5. Write a PL/SQL code to fund the sum of first N numbers

**EXPERIMENT-14**

**Title:  To understand the concepts of function and procedure in PL/SQL.**

**Objective:** Students will be able to implement the Pl/SQL programs using function and procedure.

1. Implement the above experiments of PL/SQL using functions and procedures.

**EXPERIMENT-15**

**Title:  To understand the concepts of implicit and explicit cursor.**

**Objective:** Students will be able to implement the concept of implicit and explicit cursor.

1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message “**No Change**”.
2. Using explicit cursor fetch the employee name, employee\_id and salary of all the records from EMPLOYEES table.
3. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee\_id, Last\_Name and salary for those records whose salary exceeds 2500 into a new table TEMP\_EMP

**EXPERIMENT-16**

**Title:  To understand the concepts of Trigger.**

**Objective:** Students will be able to implement the concept of trigger.

**CUSTOMER Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | NAME | AGE | ADDRESS | SALARY |
| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |
| 2 | Khilan | 25 | Delhi | 1500.00 |
| 3 | Kaushik | 23 | Kota | 2000.00 |
| 4 | Chaitali | 25 | Mumbai | 6500.00 |
| 5 | Hardik | 27 | Bhopal | 8500.00 |
| 6 | Komal | 22 | MP | 4500.00 |

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.

**EXPERIMENT-17**

**Title:  To understand the concepts of Trigger.**

**Objective:** Students will be able to implement the concept of trigger.

1. CREATE TRIGGER SALARY\_VIOLATION BEFORE INSERT OR UPDATE OF SALARY, SUPERVISOR\_SSN ON EMPLOYEE of experiment 3

**EXPERIMENT-18**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to display all the documents in the collection hotel.
2. Write a MongoDB query to display the fields hotel\_id, name, Borough and cuisine for all the documents in the collection hotel.
3. Write a MongoDB query to display the fields hotel\_id, name, Borough and cuisine, but exclude the field \_id for all the documents in the collection hotel.
4. Write a MongoDB query to display the fields hotel\_id, name, Borough and zip code, but exclude the field \_id for all the documents in the collection hotel. hotel
5. Write a MongoDB query to display all the hotel which is in the Borough Bronx.

**EXPERIMENT-19**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to display the next 5 hotels after skipping first 5 which are in the Borough Bronx.
2. Write a MongoDB query to find the hotels that achieved a score, more than 80 but less than 100.
3. Write a MongoDB query to find the hotels which locate in latitude value less than -95.75
4. Write a MongoDB query to find the hotels that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

**EXPERIMENT-20**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to arrange the name of the cuisine in ascending order and for that same cuisine Borough should be in descending order.
2. Write a MongoDB query to know whether all the addresses contains the street or not.
3. Write a MongoDB query which will select all documents in the hotels collection where the coord field value is Double.
4. Write a MongoDB query which will select the hotel Id, name and grades for those hotels which returns 0 as a remainder after dividing the score by 7.
5. Write a MongoDB query to find the hotel name, Borough, longitude and attitude and cuisine for those hotels which contains 'mon' as three letters somewhere in its name.
6. Write a MongoDB query to find the hotel name, Borough, longitude and latitude and cuisine for those hotels which contain 'Mad' as first three letters of its name.

**EXPERIMENT-21**

**Title:  To understand the concepts of NoSQL Database**

**Objective:** Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.   
2. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the Borough Brooklyn. The document must be displayed according to the cuisine in descending order.

3. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'ces' as last three letters for its name.

 4. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'Reg' as three letters somewhere in its name.

5. Write a MongoDB query to find the hotels which belong to the Borough Bronx and prepared either American or Chinese dish.

6. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which belong to the Borough Staten Island or Queens or Hyatt.

7. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which are not belonging to the Borough New Delhi or Queens or Hyatt.

8. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which achieved a score which is not more than 10.

 9. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which prepared dish except 'American' and 'Chinees' or hotel's name begins with letter 'Wil'.

 10. Write a MongoDB query to find the hotel Id, name, and grades for those hotels which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.

**EXPERIMENT-22**

Mini Project – On SQL

**EXPERIMENT-23**

Mini Project- On NoSQL

**Total Lab hours 30**

**Textbooks**

1. Ivan Bayross, "SQL, PL/SQL – The Programming Language of Oracle", 4th Revised Edition, 2010.

2. Kristina Chodorow, and Michael Dirolf, "MongoDB: The Definitive Guide", O'Reilly Media Inc., 2010.

**Reference Books**

**Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination**

**Examination Scheme:** Continuous Assessment

|  |  |  |
| --- | --- | --- |
| **Components** | **Quiz & Viva** | **Performance & Lab Report** |
| Weightage (%) | 50 | 50 |