|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | |
| **Practical Examination** | | | | | | | |
| **Student Details** | | | | | | | |
| **Name of the Student:** | | |  | | **Roll No: -** | |  |
| **Signature of Student:** | | |  | | **Date:-** | |  |
| **General Instructions** | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | |
| **Sr.No** | **Experiment-1** | | | | | | **Remark** |
|  | Design a database for hospital with a set of patients and a set of medical doctors. Associate with each patient a log of various tests and examination conducted.  CREATE DATABASE HOSPITAL;  USE HOSPITAL;  CREATE TABLE DOCTOR  (Name VARCHAR(15) NOT NULL,  D\_Id INT NOT NULL,  DOB DATE,  Address VARCHAR(30),  Sex CHAR,  Grade VARCHAR(9),  Depart VARCHAR(25) NOT NULL,  DAYS VARCHAR(10));  INSERT INTO DOCTOR VALUES('RAM', 125, '1965-06-05', 'London', 'M', 'Head', 'Neuro', 'Friday');  INSERT INTO DOCTOR VALUES('SITA', 121, '1984-04-25', 'Paris', 'F', 'Surgeon', 'Neuro', 'Saturday');  INSERT INTO DOCTOR VALUES('AYUSHI', 126, '1975-01-18', 'Mexico', 'F', 'Head', 'Ortho', 'Monday');  INSERT INTO DOCTOR VALUES('FIROZ', 129, '1991-03-25', 'Texas', 'M', 'Surgeon', 'Ortho', 'Tuesday');  INSERT INTO DOCTOR VALUES('SNEHA', 124, '1989-09-15', 'Utah', 'F', 'Surgeon', 'Gynaec', 'Wednesday');  SELECT \* FROM DOCTOR;  DROP TABLE PATIENT;  CREATE TABLE PATIENT  (Name VARCHAR(15) NOT NULL,  P\_Id INT NOT NULL,  DOB DATE,  Sex CHAR  Address VARCHAR(30)  );  INSERT INTO PATIENT VALUES('SHYAM', 105, '1965-06-05','M', 'London');  INSERT INTO PATIENT VALUES('SNEHA', 103, '1965-06-05','F', 'Vashi');  INSERT INTO PATIENT VALUES('RIYA', 101, '1965-06-05','F', 'Sanpada');  INSERT INTO PATIENT VALUES('MIRAZ', 109, '1965-06-5','M','Lonavala');  INSERT INTO PATIENT VALUES('POOJA', 107, '1965-06-05','F', 'Dadar');  SELECT \* FROM PATIENT;  CREATE TABLE TEST  (Name VARCHAR(15) NOT NULL,  T\_Id INT NOT NULL,  Cost DECIMAL(10,2));  INSERT INTO TEST VALUES('BP', 115, 250);  INSERT INTO TEST VALUES('ECG', 111, 150);  INSERT INTO TEST VALUES('TSH', 116, 250);  SELECT \* FROM TEST;  CREATE TABLE Conducted\_On  (P\_Id INT NOT NULL,  T\_Id INT NOT NULL,  DOC DATE);  INSERT INTO Conducted\_On VALUES(103, 115, '2022-03-16');  INSERT INTO Conducted\_On VALUES(107, 111, '2022-02-26');  INSERT INTO Conducted\_On VALUES(109, 116, '2022-01-31');  INSERT INTO Conducted\_On VALUES(101, 111, '2022-03-21');  INSERT INTO Conducted\_On VALUES(105, 115, '2022-02-06');  SELECT \* FROM Conducted\_On;  CREATE TABLE EXAM  (P\_Id INT NOT NULL,  DOE DATE);  INSERT INTO EXAM VALUES(103, '2010-05-08');  INSERT INTO EXAM VALUES(107, '2022-02-26');  INSERT INTO EXAM VALUES(109, '2022-01-31');  INSERT INTO EXAM VALUES(101, '2022-03-21');  INSERT INTO EXAM VALUES(105, '2022-02-06'); | | | | | |  |
| **a.** | find the set of patients who live in “Vashi” and were examined on 8.05.10  SELECT \* FROM PATIENT P, EXAM E WHERE P.P\_Id= E.P\_Id AND P.Address='Vashi' AND DOE='2010-05-08'; | | | | | |  |
| **b.** | List the various tests and examination conducted on each patient  SELECT \* FROM Conducted\_On, EXAM WHERE C.P\_Id=E.P\_Id GROUP BY P\_Id; | | | | | |  |
| **c.** | Find the name of the doctors who visit only on Tuesday  SELECT \* FROM DOCTOR WHERE DAYS='Tuesday'; | | | | | |  |
| **d.** | Write any one trigger After Insert  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  ALTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | |  |
| **e.** | Write any one procedure or function to count no of patient from vashi  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | |  |
| **Marks Obtained** | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | **Oral Marks**  **(out of 10)** |  | **Total Marks**  **(out of 25)** |  |
|  | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-2** | | | | | | | | | | **Remark** | |
|  | Design a database of a university registrar’s office which maintains data about the course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom.  CREATE DATABASE UNI;  USE UNI;  CREATE TABLE COURSE\_OFFERING(  C\_NO INT NOT NULL,  SEC\_NO INT NOT NULL,  YEAR CHAR(4) NOT NULL,  SEMESTER VARCHAR(5) NOT NULL,  TIME CHAR(10) NOT NULL,  ROOM CHAR(15) NOT NULL);  SELECT \* FROM COURSE\_OFFERING;  INSERT INTO COURSE\_OFFERING VALUES(121, 301, 2009, 'IV', '11:00am', 'A1-230');  INSERT INTO COURSE\_OFFERING VALUES(123, 311, 2010, 'V', '2:30pm', 'A1-210');  INSERT INTO COURSE\_OFFERING VALUES(125, 421, 2008, 'VI', '4:00pm', 'A2-230');  INSERT INTO COURSE\_OFFERING VALUES(127, 356, 2007, 'III', '10:30am', 'A1-270');  INSERT INTO COURSE\_OFFERING VALUES(129, 247, 2010, 'II', '10:00am', 'A3-240');  CREATE TABLE INSTRUCTOR(  I\_ID INT NOT NULL,  Name VARCHAR(50) NOT NULL,  Dept VARCHAR(25) NOT NULL,  Title VARCHAR(30) NOT NULL)  SELECT \* FROM COURSE\_OFFERING;  SELECT \* FROM INSTRUCTOR;  INSERT INTO INSTRUCTOR VALUES(152, 'Jain', 'CS', 'HOD');  INSERT INTO INSTRUCTOR VALUES(143, 'Patil', 'IT', 'Assistant Professor');  INSERT INTO INSTRUCTOR VALUES(126, 'Pillai', 'BCA', 'HOD');  INSERT INTO INSTRUCTOR VALUES(225, 'Singh', 'MBA', 'Assistant Professor');  CREATE TABLE TEACHES(  C\_NO INT NOT NULL,  SEC\_NO INT NOT NULL,  SEM CHAR(5) NOT NULL,  YEAR CHAR(4) NOT NULL,  I\_ID INT NOT NULL);  SELECT \* FROM TEACHES;  INSERT INTO TEACHES VALUES(125, 421, 'VI', 2008, 152);  INSERT INTO TEACHES VALUES(121, 301, 'IV', 2009, 143);  INSERT INTO TEACHES VALUES(123, 311, 'V', 2010, 152);  INSERT INTO TEACHES VALUES(127, 356, 'III', 2007, 126);  INSERT INTO TEACHES VALUES(129, 247, 'II', 2010, 225); | | | | | | | | | |  | |
| **a.** | Retrieve the name of all courses taught by professor Jain in 2009 and 2010.  SELECT \* FROM COURSE\_OFFERING WHERE C\_NO IN (SELECT C\_NO FROM TEACHES WHERE I\_ID IN(SELECT I\_ID FROM INSTRUCTOR WHERE Name='Jain' )AND YEAR=2009 AND YEAR=2010); | | | | | | | | | |  | |
| **b.** | List all the course numbers taught by each instructor.  SELECT C\_NO FROM COURSE\_OFFERING WHERE EXISTS(SELECT \* FROM TEACHES WHERE C\_NO=C\_NO AND I\_ID IN (SELECT I\_ID FROM INSTRUCTOR GROUP BY I\_ID)); | | | | | | | | | |  | |
| **c.** | Update a course of your choice and delete the course which you don’t like.  UPDATE COURSE\_OFFERING SET YEAR=2009 WHERE C\_NO=125;  DELETE FROM COURSE\_OFFERING WHERE C\_NO=121; | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Instructor  ON INSTRUCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'instructor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-3** | | | | | | | | | | **Remark** | |
|  | Design a database of a university registrar’s office which maintains data about the students, including student-id, name, and program and their previous academic record. | | | | | | | | | |  | |
| **a.** | Retrieve the names of all senior students majoring in ‘CS’ (computer science).  SELECT Name FROM STUDENT WHERE YEAR=’Senior’ AND PROGRAM=’CS’ | | | | | | | | | |  | |
| **b.** | Create a view which gives the personal details of students along with their previous academic record  CREATE VIEW [STUDENT DETAILS]  AS  SELECT PID, PNAME, PREV\_MARS  FROM STUDENT;  SELECT \* FROM [STUDENT DETAILS] | | | | | | | | | |  | |
| **c.** | Find the name of student who has got first class in all his academic record.  SELECT Name FROM STUDENT WHERE GRADE=’FIRST’ | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-4** | | | | | | | | | | **Remark** | |
|  | Design a database of a university registrar’s office which maintains data about the instructors, including identification number, name, department, and title. | | | | | | | | | |  | |
| **a.** | List all the female instructors and PhD holders in each department with their identification number.  SELECT \* FROM INSTRUCTOR WHERE SEX=’F’ AND GRADE=’PHD’ GROUP BY I\_ID | | | | | | | | | |  | |
| **b.** | Find the name of the instructor who is working for more than 9 years in one department and then 5 years in the university.  SELECT Name FROM INSTRUCTOR WHERE EXP>9 GROUP BY Depart | | | | | | | | | |  | |
| **c.** | Find the name of the instructors whose appointment date is on or before 28.02.17  SELECT Name from INSTRUCTOR WHERE APP\_Date<2017-02-28 | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-5** | | | | | | | | | | **Remark** | |
|  | Design a database to record the marks that students get in different exams of different course offerings by the university. | | | | | | | | | |  | |
| **a.** | Find all failures of each course.  SELECT p.name  FROM Relevant\_subjects rs JOIN  Courses c  ON c.subject = rs.id JOIN  Course\_enrolments ce  ON ce.course = c.id JOIN  Students s  ON s.id = ce.student JOIN  People p  ON p.id = s.id  WHERE ce.mark < 50  GROUP BY p.id, p.name  HAVING COUNT(DISTINCT rs.id) = (SELECT COUNT(DISTINCT rs2.id) FROM relevant\_subjects rs2); | | | | | | | | | |  | |
| **b.** | List the name of topper of the of each course in the University.  SELECT Name, Course, Max(Marks)  FROM Student  GROUP by C\_Id; | | | | | | | | | |  | |
| **c.** | List all the students who have scored percentage between 70 and 100  SELECT \* FROM student WHERE PERCENTAGE BETWEEN 70 and 100 | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-6** | | | | | | | | | | **Remark** | |
|  | Design a database for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match, the players in each match and individual player statistics for each match. | | | | | | | | | |  | |
| **a.** | Find the highest scorer of all the matches  SELECT MAX(MATCH\_Score) AS MAX\_Score FROM TEAM GROUP BY Match\_No | | | | | | | | | |  | |
| **b.** | List number of matches played by the lowest scorer of the match.  SELECT No\_Match FROM TEAM WHERE MATCH\_Score=ALL(SELECT MN(MATCH\_Score) FROM TEAM) | | | | | | | | | |  | |
| **c.** | Find the total number of matches played in the year 2015-2019  SELECT COUNT(Match\_No) AS Total\_Count FROM TEAM WHERE YEAR BETWEEN 2015 AND 2019 | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-7** | | | | | | | | | | **Remark** | |
|  | Design a database which models an online bookstore. It should have the entities like book, author, publisher, shopping basket, customer and warehouse. | | | | | | | | | |  | |
| **a.** | List all the books which have highest demand in alphabetical order.  SELECT \* FROM BOOK WHERE MAX(Count) IN (SELECT COUNT(ORDER) FROM BOOK OREDER BY Name ASC) | | | | | | | | | |  | |
| **b.** | List the customer who has taken at least three books  SELECT \* FROM Customer WHERE COUNT(Order)>3 group by C\_Id | | | | | | | | | |  | |
| **c.** | Create a view which gives information on books available in the store and their author.  CREATE VIEW [BOOK DETAILS]  AS  SELECT BID, BNAME, AUTHOR\_ID, A\_Name  FROM BOOK AND AUTHOR;  SELECT \* FROM [STUDENT DETAILS] | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-8** | | | | | | | | | | **Remark** | |
|  | Design a database for a motor-vehicle sales company. The company sells motorcycles, passenger cars, vans, and buses. | | | | | | | | | |  | |
| **a.** | List all the vehicles which have highest demand in descending order  SELECT \* FROM VEHICLE WHERE MAX(ORDER) AS MAX ORDER BY DESC | | | | | | | | | |  | |
| **b.** | Find the total sale of the company  SELECT TOTAL(Amount) AS Total\_Sale FROM VEHICLE | | | | | | | | | |  | |
| **c.** | Find the number of vans sold in April 2019  SELECT COUNT(Order\_ID) AS NO FROM VEHICLE WHERE Type=’Van’ AND Date LIKE ‘2019-04-\_\_’ | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-9** | | | | | | | | | | **Remark** | |
|  | Design the relational database for given below:-  employee(person-name, street, city)  works(person-name, company-name, salary)  company(company-name, city)  manages (person-name, manager-name).  create table Employee(  person\_name VARCHAR(20) PRIMARY KEY,  street VARCHAR(20),  city VARCHAR(20))  select \* from Employee  insert into Employee values ('Aaryan','Vashi','Navi Mumbai')  insert into Employee values ('Karishma','Panvel','Mumbai')  insert into Employee values ('Juhi','Thane','Thane')  insert into Employee values ('Nidhi','Sanpada','Navi Mumbai')  create table Company(  company\_name VARCHAR(30) PRIMARY KEY,  city VARCHAR(20))  select \* from Company  insert into Company values ('First Bank Corporation','Thane')  insert into Company values ('Small Bank Corporation','Vashi')  insert into Company values ('Second Bank Corporation','Panvel')  insert into Company values ('Third Bank Corporation', 'Vashi')  insert into Company values ('Fourth Bank Corporation', 'Thane')  insert into Company values ('Large Bank Corporation', 'Thane')  create table Works(  person\_name VARCHAR(20) FOREIGN KEY REFERENCES Employee(person\_name),  company\_name VARCHAR(30) FOREIGN KEY REFERENCES Company(company\_name),  salary int)  select \* from Works  insert into Works values ('Aaryan','First Bank Corporation',10000)  insert into Works values ('Juhi','Second Bank Corporation',20000)  insert into Works values ('Karishma','Small Bank Corporation',30000)  create table Manages(  person\_name VARCHAR(20) FOREIGN KEY REFERENCES Employee(person\_name),  manager\_name VARCHAR(20)FOREIGN KEY REFERENCES Employee(person\_name))  insert into Manages values('Aaryan','Karishma')  insert into Manages values('Karishma','Juhi')  select \* from Manages | | | | | | | | | |  | |
| **a.** | Find the employees who work for First Bank Corporation  select person\_name from Employee where person\_name in (select person\_name from Works where company\_name = 'First Bank Corporation') | | | | | | | | | |  | |
| **b.** | Assume the companies may be located in several cities. Find all companies located in the city in which Small Bank Corporation is located  select company\_name from Company where city in ( select city from Company where company\_name = 'Small Bank Corporation') | | | | | | | | | |  | |
| **c.** | Find the names of all employees in this database who do not work for First Bank Corporation  select person\_name from Employee where person\_name NOT IN ( select person\_name from Works where company\_name = 'First Bank Corporation') | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-10** | | | | | | | | | | **Remark** | |
|  | Design the database of schema given below: -  person(driver-id, name, address)  car(license, model, year)  accident(report-number, date, location)  owns(driver-id, license)  participated(driver-id, license, report-number, damage-amount)  create table Person  (driver\_id int primary key,  name varchar(20),  street\_address varchar(20),  )  create table Car  (license int primary key ,  model varchar(20),  Year int,  )  create table Accident  (report\_number int primary key,  Date date,  Location varchar(20),  )  create table Owns  (driver\_id int foreign key references person(driver\_id) ,  license int foreign key references car(license)  )  alter table owns  add foreign key(driver\_id) references person(driver\_id)  alter table owns  add foreign key(license) references car(license)  alter table participated  add foreign key(driver\_id) references person(driver\_id)  alter table participated  add foreign key(license) references car(license)  alter table participated  add foreign key(report\_number) references accident(report\_number)  create table Participated  (driver\_id int foreign key references person(driver\_id) ,  license int foreign key references car(license) ,  report\_number int foreign key references accident(report\_number),  damage\_amount int  )  Alter table  insert into person values(1,'Juhi','Panvel')  insert into person values(2,'Karishma','Vashi')  insert into person values(5,'Aaryan','sector-9a')  select\*from person  drop table person  drop table owns  insert into car values(123,'abc','2017')  insert into car values(9123,'def','2015')  insert into car values(23,'ghi','2013')  insert into car values(3,'jkl','2012')  select\*from accident  insert into Accident values(10,'1999-08-03','vashi')  insert into Accident values(11,'1999-09-02','kharghar')  insert into Accident values(12,'1999-07-01','sion')  insert into Accident values(13,'2018-07-01','khandeshwar')  insert into participated values(1,123,13,1234) | | | | | | | | | |  | |
| **a.** | Add a new accident to the database; assume any values for the required attributes  insert into participated values(1,123,13,1234) | | | | | | | | | |  | |
| **b.** | Find the names of all persons whose street address includes the substring ‘Sector-9a’  select name from person where street\_address like '%sector-9a%' | | | | | | | | | |  | |
| **c.** | List in alphabetical order all persons whose car met with accident in 2018.  select name from person where driver\_id in(select driver\_id from participated where report\_number in(select report\_number from accident where date like '2018%') ) | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-11** | | | | | | | | | | **Remark** | |
|  | A database is being constructed to keep track of the teams and games of a sports league. A team has a number of players, not all of whom participated in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of the game. Design a relational database for this application, stating any assumptions you make. Choose your favorite sport (e.g., soccer, football, baseball).  CREATE TABLE TEAM (  TEAM\_ID INT PRIMARY KEY,  TEAM\_NAME VARCHAR(50))  SELECT \* FROM TEAM  INSERT INTO TEAM VALUES (101,'TEAM RED')  INSERT INTO TEAM VALUES (102,'TEAM BLUE')  INSERT INTO TEAM VALUES (103,'TEAM GREEN')  INSERT INTO TEAM VALUES (104,'TEAM YELLOW')  CREATE TABLE PLAYER (  PID INT PRIMARY KEY,  PNAME VARCHAR(50),  TID INT FOREIGN KEY REFERENCES TEAM(TEAM\_ID))  INSERT INTO PLAYER VALUES (201,'ALEXANDER',101)  INSERT INTO PLAYER VALUES (202,'CAESER',102)  INSERT INTO PLAYER VALUES (203,'ROMA',103)  INSERT INTO PLAYER VALUES (204,'VICTORIA',104)  INSERT INTO PLAYER VALUES (205,'SIMA',101)  INSERT INTO PLAYER VALUES (206,'CLEOPATRA',102)  SELECT \* FROM PLAYER  CREATE TABLE SPORTS (  SPORTS\_ID INT PRIMARY KEY,  SNAME VARCHAR(50),  )  INSERT INTO SPORTS VALUES (301,'CRICKET')  INSERT INTO SPORTS VALUES (302,'SOCCER')  INSERT INTO SPORTS VALUES (303,'FOOTBALL')  INSERT INTO SPORTS VALUES (304,'BASKETBALL')  SELECT \* FROM SPORTS  CREATE TABLE PARTICIPATED (  TID INT FOREIGN KEY REFERENCES TEAM(TEAM\_ID),  SID INT FOREIGN KEY REFERENCES SPORTS(SPORTS\_ID))  INSERT INTO PARTICIPATED VALUES (101,301)  INSERT INTO PARTICIPATED VALUES (102,302)  INSERT INTO PARTICIPATED VALUES (102,303)  INSERT INTO PARTICIPATED VALUES (103,301)  INSERT INTO PARTICIPATED VALUES (103,303)  INSERT INTO PARTICIPATED VALUES (104,304)  INSERT INTO PARTICIPATED VALUES (104,301)  INSERT INTO PARTICIPATED VALUES (101,302)  INSERT INTO PARTICIPATED VALUES (101,303)  INSERT INTO PARTICIPATED VALUES (104,302)  INSERT INTO PARTICIPATED VALUES (102,301)  SELECT \* FROM PARTICIPATED  CREATE TABLE GAME(  GAME\_ID INT PRIMARY KEY,  GAME\_RESULT VARCHAR(50),  )  INSERT INTO GAME VALUES (401,'RED WIN')  INSERT INTO GAME VALUES (402,'GREEN WIN')  INSERT INTO GAME VALUES (403,'BLUE WIN')  SELECT \* FROM GAME  CREATE TABLE HAVE (  SID INT FOREIGN KEY REFERENCES SPORTS(SPORTS\_ID),  GID INT FOREIGN KEY REFERENCES GAME(GAME\_ID))  INSERT INTO HAVE VALUES (302,401)  INSERT INTO HAVE VALUES (301,402)  INSERT INTO HAVE VALUES (303,403)  SELECT \* FROM HAVE  CREATE TABLE PLAYS(  PID INT FOREIGN KEY REFERENCES PLAYER(PID),  GID INT FOREIGN KEY REFERENCES GAME(GAME\_ID),  PLAYER\_POSITION VARCHAR(50))  INSERT INTO PLAYS VALUES (201,401,'QUATERBACK')  INSERT INTO PLAYS VALUES (204,401,'SHOOTER')  INSERT INTO PLAYS VALUES (202,403,'DEFENDER')  INSERT INTO PLAYS VALUES (205,403,'GOALKEEPER')  INSERT INTO PLAYS VALUES (203,402,'BATSMAN')  INSERT INTO PLAYS VALUES (206,402,'BOWLER')  SELECT \* FROM PLAYS | | | | | | | | | |  | |
| **a.** | List all the games played in the sports league  SELECT \* FROM GAME | | | | | | | | | |  | |
| **b.** | Create a view showing detail of each player  CREATE VIEW [PLAYER DETAILS]  AS  SELECT PID, PNAME, TID  FROM PLAYER;  SELECT \* FROM [PLAYER DETAILS] | | | | | | | | | |  | |
| **c.** | Find the total number of teams participating in the sports league.  SELECT COUNT(DISTINCT TID) AS TEAM\_PARTICIPATED FROM PARTICIPATED | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-12** | | | | | | | | | | **Remark** | |
|  | Design a database to keep track of information for an art museum. The museum has a collection of ART\_ OBJECTS. Each ART\_ OBJECTS has a unique IdNo, an Artist (if known), a Year (when it was created, if known), a Title, and a Description. | | | | | | | | | |  | |
| **a.** | Create a view which gives description on artist their ART\_ OBJECTS.  CREATE VEW [artist\_details]  AS  SELECT AID, ANAME  FROM ARTIST;  SELECT \* FROM [artist\_details] | | | | | | | | | |  | |
| **b.** | Give description of all the ART\_OBJECTS which were created in the year 2001  SELECT \* FROM ART WHERE YEAR=2001 | | | | | | | | | |  | |
| **c.** | List all the female artists whose name starts with “Re”.  SELECT \* FROM ARTIST WHERE SEX=’F’ AND Name LIKE ‘Re%’ | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-13** | | | | | | | | | | **Remark** | |
|  | Design a database for a small private airport database that is used to keep track of airplanes, their owners, airport employees, and pilots. | | | | | | | | | |  | |
| **a.** | List all the pilots of the plane “Kingfisher”  SELECT \* FROM PILOTS WHERE PLANE=’Kingfisher’ | | | | | | | | | |  | |
| **b.** | Retrieve the list of all employees who were appointed between 01.01.2017 and 01.01.2019.  SELECT \* FROM EMPLOYEE WHERE App\_Date BETWEEN ‘2017-01-01’ AND ‘2019-01-01’ | | | | | | | | | |  | |
| **c.** | Update the database of your choice and delete which you don’t like. | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-14** | | | | | | | | | | **Remark** | |
|  | Consider the following relations for a database that keeps track of business trips of salespersons in a sales office:  SALESPERSON(Sid, Name, Start\_Year, Dept\_No)  TRIP(Sid, From\_City, To\_City, Departure\_Date, Return\_Date, Trip\_ID)  Expense(Trip\_ID, Account, Amount) | | | | | | | | | |  | |
| **a.** | Give the details (all attributes of Trip relation) for trips that exceeded Rs. 2000 in expenses  SELECT \* FROM TRIP WHERE Trip\_ID IN(SELECT Trip\_ID FROM Expense WHERE AMOUNT > 2000 | | | | | | | | | |  | |
| **b.** | Print the ENO of salesman who took trip to ‘Honolulu’  SELECT ENO FROM SALESPERSON WHERE Sid IN (SELECT Sid FROM TRIP WHERE To\_City=’Honlulu’) | | | | | | | | | |  | |
| **c.** | Print the total trip expenses incurred by the salesman with ENO = ‘234-56-7890’  SELECT SUM(Amount) AS TOTAL WHERE Trip\_ID IN (SELECT Trip\_ID FROM TRIP WHERE Sid IN ( SELECT Sid FROM SALESPERSON WHERE ENO=’234-56-7890’ | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-15** | | | | | | | | | | **Remark** | |
|  | Consider the following relations for the database that keeps track of student enrollment in courses and the books opted for each course:  Student(StuNo, Name, Major, Bdate)  Course(Course-id, Cname, Dept)  Enroll(StudNo, Course-id, Quarter, Grade)  Book\_Adoption(Course-id,Quarter,Book\_ISBN)  Text(Book\_ISBN, Book\_Title, Publisher, Author) | | | | | | | | | |  | |
| **a.** | List number of courses taken by aa students named ‘Ravi Shankar’ in Winter 2018  SELECT COUNT(Course-id) AS NO FROM Course WHERE Course-id IN(SELECT Course-id FROM Enroll WHERE Quarter=’Winter 2018’ AND StudNo IN (SELECT StudNo FROM Students WHERE Name=’ Ravi Shankar’ | | | | | | | | | |  | |
| **b.** | Produce a list of books for courses offered by the ‘CSE’ department that have used more than two books, | | | | | | | | | |  | |
| **c.** | List any department that has all its adopted books published by ‘TMH Publishing’. | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-16** | | | | | | | | | | **Remark** | |
|  | Consider the following relations for a database that keeps track of auto sales in a car dealership  Car(Serial\_No, Model, Manufacturer, Price)  Options(serial-No, Option-Name, Price)  Sales(Salesperson-id, Serial-No. Date, Sale-Price)  Salesperson(Salesperson-id,Name,Phone)  create table car(  serial\_no int primary key,  model varchar(20),  manufacturer varchar(20),  price int)  select \* from car  insert into car values(20345,'maruti','palak',50000)  insert into car values(20365,'maruti','nidhi',40000)  insert into car values(20375,'creta','aaryan',30000)  insert into car values(20385,'aulto','athrava',20000)  create table options(  Option\_serial\_no int foreign key references car(serial\_no),  option\_name varchar(20),  Option\_price int)  select \* from options  insert into options values(20345,'creta',50000)  insert into options values(20365,null,40000)  insert into options values(20375,null,30000)  insert into options values(20385,'maruti',20000)  create table sales(  salesperson\_id int foreign key references salesperson(salesperson\_id),  serial\_no int foreign key references car(serial\_no),  sdate date,  sale\_price int)  select \* from sales  insert into sales values(1020261,20345,'10-12-12',10000)  insert into sales values(1020248,20345,'10-12-12',10000)  insert into sales values(1020235,20345,'10-12-12',10000)  insert into sales values(1020251,20345,'10-12-12',10000)  create table salesperson(  salesperson\_id int primary key,  salesperson\_name varchar(20),  salesperson\_phone varchar(20))  select \* from salesperson  insert into salesperson values(1020261,'anna','yes')  insert into salesperson values(1020248,'jane doe','yes')  insert into salesperson values(1020235,'maria jonas',null)  insert into salesperson values(1020251,'peter',null) | | | | | | | | | |  | |
| **a.** | List the serial and Model of cars that have no options.  select serial\_no,model from car,options where car.serial\_no = options.Option\_serial\_no and option\_name is null | | | | | | | | | |  | |
| **b.** | For the salesperson named ‘Jane Doe’, list the following information for all the cars she sold  select \* from car,sales,salesperson where sales.salesperson\_id = salesperson.salesperson\_id  and car.serial\_no = sales.serial\_no  and salesperson\_name = 'jane doe' | | | | | | | | | |  | |
| **c.** | List the name of the salesperson who have no phone.  select \* from salesperson where salesperson\_phone is null | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-17** | | | | | | | | | | **Remark** | |
|  | Consider the following six relations for an order-processing database application in a company:  CUSTOMER(Cust#, Cname, City)  ORDER(Order, Odate,Cust#, Ord\_Amt)  ORDER\_ITEM(Order, Item, Qty)  ITEM(Item, Unit\_Price)  SHIPMENT(Order, Warehouse#, Ship\_date)  WAREHOUSE(Warehouse#, City) | | | | | | | | | |  | |
| **a.** | List the Order and Ship\_date for all orders shipped from Warehouse number ‘W122’  SELECT ORDER, Shp\_date FROM SHIPMENT WHERE Warehouse#=’W122’ | | | | | | | | | |  | |
| **b.** | List the orders that were not shipped within 30 days of ordering | | | | | | | | | |  | |
| **c.** | List the Order for orders that were shipped from all warehouses that the company has in New York. | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-18** | | | | | | | | | | **Remark** | |
|  | Design a COMPANY database which is specified as below  Employee(EID, Name, Bdate, Address, Salary, DeptID)  Department(DeptID, Dname, Office, Mng-EID)  Project(Code, Name, Budget, DeptID)  Join(EID, Pcode, StartDate)  Emp-Dependent(EID, Dependent-Name, Bdate, Relationship) | | | | | | | | | |  | |
| **a.** | Find the name of the employee who joined in ‘Green Green” project. | | | | | | | | | |  | |
| **b.** | Find the name of the employees who have no dependents | | | | | | | | | |  | |
| **c.** | Find the name of the employees who work for both project number 1 and project number 2. | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-19** | | | | | | | | | | **Remark** | |
|  | Design the library database which is used to keep track of books, borrowers, and book loans  create table books(  book\_id int primary key,  title varchar(20),  borrower\_id int  )  create table borrower(  borrower\_id int primary key,  name varchar(20),  address varchar(20),  )  create table branch(  branch\_id int primary key,  location varchar(20),  )  DROP TABLE BRANCH  ALTER TABLE BOOKS  ADD FOREIGN KEY (borrower\_id) REFERENCES borrower(borrower\_id);  ALTER TABLE BOOKS ADD BRANCH\_ID INT  ALTER TABLE BOOKS  ADD FOREIGN KEY (BRANCH\_ID) REFERENCES BRANCH(BRANCH\_ID);  INSERT INTO BORROWER VALUES (1,'JUHI','VASHI')  INSERT INTO BORROWER VALUES (2,'AARYAN','SEC-1')  INSERT INTO BORROWER VALUES (4,'SHUBHOM','THANE')  INSERT INTO BORROWER VALUES (3,'KARISHMA','NAVI MUMBAI')  INSERT INTO books VALUES (1,'THE LOST TRIBE',3,2)  INSERT INTO books VALUES (3,'THE LOST TRIBE',3,1)  INSERT INTO books VALUES (2,'THE LOST TRIBE',1,1)  DELETE FROM BOOKS  INSERT INTO branch VALUES (1,'VASHI')  INSERT INTO branch VALUES (2,'MUMBAI')  SELECT \* FROM BRANCH  SELECT \* FROM BOOKS | | | | | | | | | |  | |
| **a.** | Retrieve the names of all borrowers who do not have any books checked out  SELECT NAME FROM BORROWER WHERE borrower\_id NOT IN (SELECT borrower\_id FROM BOOKS) | | | | | | | | | |  | |
| **b.** | How many copies of the book titled The Lost Tribe are owned by each library branch?  SELECT COUNT(TITLE) FROM BOOKS WHERE TITLE= 'THE LOST TRIBE' GROUP BY BRANCH\_ID | | | | | | | | | |  | |
| **c.** | Retrieve the names, address, and number of books checked out for all borrowers who have more than five books checked out.  SELECT NAME, ADDRESS,COUNT(BOOK\_ID) FROM BOOKS, BORROWER WHERE EXISTS(SELECT \*  FROM BOOKS GROUP BY BOOKS.BORROWER\_ID HAVING COUNT(BOOK\_ID)>1) | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course name: -** **CSL402- Database Management System Lab** | | | | | | | | | | | | |
| **Practical Examination** | | | | | | | | | | | | |
| **Student Details** | | | | | | | | | | | | |
| **Name of the Student:** | | |  | | | | **Roll No: -** | | | |  | |
| **Signature of Student:** | | |  | | | | **Date:-** | | | |  | |
| **General Instructions** | | | | | | | | | | | | |
| 1. | Read the problem Statement and identify entities, attributes and relationships | | | | | | | | | | | |
| 2. | Draw ER or EER diagram for the problem statement | | | | | | | | | | | |
| 3. | Convert ER/EER to relational model | | | | | | | | | | | |
| 4. | Apply suitable DDL and DML commands to implement it in SQL server | | | | | | | | | | | |
| 5. | Execute all the queries mentioned. | | | | | | | | | | | |
| 6. | Write down those query statement and their outputs | | | | | | | | | | | |
| **Sr.No** | **Experiment-20** | | | | | | | | | | **Remark** | |
|  | Design the library database which is used to keep track of books, borrowers, and book loans | | | | | | | | | |  | |
| **a.** | How many copies of the book titled The Lost Tribe are owned by the library branch whose name is ‘Sharps town’.  SELECT COUNT(BOOK\_ID) AS NO FROM BOOK WHERE Name=’ Sharps town’ GROUP BY BOOK\_ID | | | | | | | | | |  | |
| **b.** | For each library branch, retrieve the branch name and total number of books loaned out from that branch | | | | | | | | | |  | |
| **c.** | Update the database four new books. | | | | | | | | | |  | |
| **d.** | Write any one trigger  CREATE TABLE LOG\_FILE(  Sr\_No INT IDENTITY PRIMARY KEY,  Time\_of\_operation Datetime,  Trigger\_type Varchar(50),  operation Varchar(50),  table\_name Varchar(50));  SELECT \* FROM LOG\_FILE;  LTER TRIGGER After\_Insert\_Doctor  ON DOCTOR  After Insert  as  Insert into  LOG\_FILE(Time\_of\_operation, Trigger\_type, operation, table\_name) VALUES  (getdate(), 'After', 'insert', 'doctor'); | | | | | | | | | |  | |
| **e.** | Write any one procedure or function  Create procedure EXE  as  begin  Declare @A int, @B int, @SUM int;  set @A = 10;  set @B = 20;  set @SUM = @A+@B;  print'Sum of a and b is';  print @SUM;  end;  exec EXE; | | | | | | | | | |  | |
| **Marks Obtained** | | | | | | | | | | | | |
| **Practical Marks**  **(out of 15)** | |  | | | **Oral Marks**  **(out of 10)** | | |  | **Total Marks**  **(out of 25)** | | |  |
| **Examiners Details** | | | | | | | | | | | | |
| **External Examiner’s Name:-** | | | |  | | **Internal Examiner’s Name:-** | | | | **Mrs.Dakshayani.R** | | |
| **External Examiner’s Sign** | | | |  | | **Internal Examiner’s Sign** | | | |  | | |