|  |  |
| --- | --- |
| **Course Code:21AI44** | **Date:** |
| **Sem: IV** | **Duration:** 90 Minutes |
| **CIE-I**  **Data Base Management Systems** | |

**Answer all Questions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL. No** | | **Questions** | **M** | **BT** | **CO** |
| 1 | a) | Consider the following relations  Warehouse(Wcode,Location,Capacity)  Box (Bcode, Contents, Value, Wcode). Store the required data in the relation. Apply different forms of JOIN operation on the given relations.  Store the required data in the relation Warehouse and Box  Apply different forms of JOIN operation on the given relations.  Suitable Data should be stored in relation Warehouse and Box  (Wcode is primary key in Warehouse table and foreign key in Box table)  Bcode is primary key in Box table)  **Apply Different forms of Join operations**  *Inner Join:* Select W.\* ,B.\* from Warehouse W inner join Box B on  W.Wcode=B.Wcode  *Left Outer Join:* Select W.\* ,B.\* from Warehouse W Left outer join Box B  on W.Wcode=B.Wcode  *Right Outer Join :*Select W.\* ,B.\* from Warehouse W Right outer join Box  B on W.Wcode=B.Wcode  *Full Join:* Select W.\* ,B.\* from Warehouse W Full join Box B on  W.Wcode=B.Wcode | 6  4\*1.5 | 2 | 1 |
| b) | **EMPLOYEE**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Empid** | **Name** | **Address** | **Dnumber** | **Salary** | | E100 | Arun | Kengeri | D100 | 20000 | | E101 | Amir | R R Nagar | D200 | 30000 | | E103 | Bharathi | Kengeri | D300 | 20000 | | E104 | David | Indiranagar | D100 | 15000 | | E105 | Charan | R R Nagar | D200 | 2000 |   **DEPARTMENT**   |  |  | | --- | --- | | **Dnumber** | **Dname** | | D100 | Admin | | D200 | Sales | | D300 | Accounts |   Considering the relations and dataset given in question  i) Illustrate having by clause with a proper query and give its usage  ii) Illustrate group by and order by clause with proper query and give its usage.   1. Illustrate having by clause with a proper query and give its use   The HAVING clause is used when the WHERE keyword could not be used with aggregate functions..  *Select D.\*, count(Dnumber) from Employee E,Department D where E.Dnumber=D.Dnumber group by D.Dnumber having count(D.Dnumber)=2;*   1. Illustrate group by and order by clause with proper query and give its use   The ORDER BY keyword is used to sort the result-set in ascending or descending order.  The GROUP BY Clause is used to group rows with same values. The GROUP BY Clause is used together with the SQL SELECT statement. The SELECT statement used in the GROUP BY clause can only be used contain column names, aggregate functions, constants and expressions.  *Select D.\*, sum(salary) from Employee E, Department D where E.Dnumber=D.Dnumber group by D.Dnumber ,order by sum(salary) Desc* | 2+2 | 2 | 1 |
| 2 | a) | Consider the following relations  Customers(custid ,lastname, firstname)  Inventory(tapeid,movieid)  Movies(movieid,moviename)  Moviesupplier(supplierid,movieid,price)  Orders(orderid,supplierid,movieid,copies)  Rentals(custid,tapeid,ckoutdate,duration)  Suppliers(supplierid,suppliername)  Populate the data and formulate Queries for the following and write the results of the Query( Minimum one tuple should be there in the result set)   1. List the movies that have more than 4 copies been ordered? 2. Which customers rented movies for 5 days or more? 3. Which movies aren't in the inventory? 4. Which movie was rented for the longest duration (by any customer)?   i) List the movies that have more than 4 copies been ordered?  SELECT Movies.MovieName FROM Movies, Orders  WHERE Orders.MovieID = Movies.MovieID  GROUP BY Movies.MovieName  HAVING SUM(Copies) > 4  ii) Which customers rented movies for 5 days or more?  SELECT DISTINCT LastName  FROM Customers, Rentals  WHERE Customers.CustID = Rentals.CustID and Duration >= 5  iii)Which movies aren't in the inventory?  SELECT Movies.MovieName FROM Movies  WHERE MovieID NOT IN (SELECT MovieID FROM Inventory)  iv) Which movie was rented for the longest duration (by any customer)?  SELECT Movies.MovieName  FROM Rentals, Movies, Inventory  WHERE Movies.MovieID = Inventory.MovieID and Inventory.TapeID = Rentals.TapeID and Rentals.Duration >= ALL (SELECT Duration FROM Rentals) | 10  3  2  2  3 | 3 | 3 |
| 3 | a) | Consider the following given relations  EmployeeInfo(Empid,empfname,emplname,dept,project,address,dob,gender)  EmployeePosition(empid,empposition,dateofjoining,salary)  Consider suitable dataset ,formulate Queries for the following and write the results of the Query( Minimum one tuple should be there in the result set) i) Write query to find all the employees whose salary is between 50000 to100000 SELECT \* FROM EmployeePosition WHERE Salary BETWEEN '50000' AND '100000';  ii) Write a query to fetch all the records from the EmployeeInfo table ordered by EmpLname in descending order and Department in the ascending order.   |  |  | | --- | --- | |  | SELECT \* FROM EmployeeInfo ORDER BY EmpFname desc,  Department asc; |   iii) Write a query to fetch details of all employees excluding the employees with first names, “Sanjay” and “Sonia” from the EmployeeInfo table.   |  |  | | --- | --- | |  | SELECT \* FROM EmployeeInfo WHERE EmpFname NOT IN ('Sanjay','Sonia'); |   iv) Write a query to fetch all employees who also hold the managerial position.(Using Join)   |  |  | | --- | --- | |  | SELECT E.EmpFname, E.EmpLname, P.EmpPosition  FROM EmployeeInfo E INNER JOIN EmployeePosition P ON  E.EmpID = P.EmpID AND P.EmpPosition IN ('Manager'); |   v)Write a SQL query to retrieve employee details from EmployeeInfo table who have a date of joining in the EmployeePosition table.(Use Exists Clause)   |  |  | | --- | --- | |  | SELECT \* FROM EmployeeInfo E  WHERE EXISTS  (SELECT \* FROM EmployeePosition P WHERE E.EmpId = P.EmpId); | | 5\*2  =  10 | 3 | 3 |
| 4 | a) | The Library Management System database keeps track of readers with the following considerations –   * The system keeps track of the staff with a single point authentication system comprising login Id and password. * Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details. * A publisher has publisher Id, Year when the book was published, and name of the book. * Readers are registered with their user\_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers. * Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too. * Staff also generate reports that has readers id, registration no of report, book no and return/issue info.   Identification of Entities = 02  Identification of Attributes = 02  Identification of Relations = 02  Drawing the ER diagram with all constraints and cardinality = 02  Drawing the Relation model with proper references given = 02   * **Book Entity :** It has authno, isbn number, title, edition, category, price. ISBN is the Primary Key for Book Entity. * **Reader Entity :** It has UserId, Email, address, phone no, name. Name is composite attribute of firstname and lastname. Phone no is multi valued attribute. UserId is the Primary Key for Readers entity. * **Publisher Entity :** It has PublisherId, Year of publication, name. PublisherID is the Primary Key. * **Authentication System Entity :** It has LoginId and password with LoginID as Primary Key. * **Reports Entity :** It has UserId, Reg\_no, Book\_no, Issue/Return date. Reg\_no is the Primary Key of reports entity. * **Staff Entity :** It has name and staff\_id with staff\_id as Primary Key. * **Reserve/Return Relationship Set :** It has three attributes: Reserve date, Due date, Return date.   [**Relationships**](https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/) **between Entities –**   * A reader can reserve N books but one book can be reserved by only one reader. The relationship 1:N. * A publisher can publish many books but a book is published by only one publisher. The relationship 1:N. * Staff keeps track of readers. The relationship is M:N. * Staff maintains multiple reports. The relationship 1:N. * Staff maintains multiple Books. The relationship 1:N. * Authentication system provides login to multiple staffs. The relation is 1:N | 10 | 4 | 3 |
| 5 |  | A bank has many branches and a large number of customers. Bank is identified by its code. Other details like name, address and phone for each bank are also stored. Each branch is identified by its bank. Branch has name, address and phone. A customer can open different kinds of accounts with the branches. A branch can have many accounts. An account can belong to more than one customer. Customers are identified by their SSN, name, address and phone number. Age is used as a factor to check whether customer is a major. There are different types of loans, each identified by a loan number. A customer can take more than one type of loan and a loan can be given to more than one customer. Loans have a duration and interest rate.  ER diagram of Bank has the following description :    * Bank have Customer. * Banks are identified by a name, code, address of main office. * Banks have branches. * Branches are identified by a branch\_no., branch\_name, address. * Customers are identified by name, cust-id, phone number, address. * Customer can have one or more accounts. * Accounts are identified by account\_no., acc\_type, balance. * Customer can avail loans. * Loans are identified by loan\_id, loan\_type and amount. * Account and loans are related to bank’s branch.   **Entities** and their **Attributes** are :    * **Bank Entity :** Attributes of Bank Entity are Bank Name, Code and Address. Code is Primary Key for Bank Entity. * **Customer Entity :** Attributes of Customer Entity are Customer\_id, Name, Phone Number and Address. Customer\_id is Primary Key for Customer Entity. * **Branch Entity :** Attributes of Branch Entity are Branch\_id, Name and Address. Branch\_id is Primary Key for Branch Entity. * **Account Entity :** Attributes of Account Entity are Account\_number, Account\_Type and Balance. Account\_number is Primary Key for Account Entity. * **Loan Entity :** Attributes of Loan Entity are Loan\_id, Loan\_Type and Amount. Loan\_id is Primary Key for Loan Entity. * **Bank has Branches => 1 : N** One Bank can have many Branches but one Branch can not belong to many Banks, so the relationship between Bank and Branch is one to many relationship. * **Branch maintain Accounts => 1 : N**  One Branch can have many Accounts but one Account cannot belong to many Branches, so the relationship between Branch and Account is one to many relationship. * **Branch offer Loans => 1 : N**  One Branch can have many Loans but one Loan can not belong to many Branches, so the relationship between Branch and Loan is one to many relationship. * **Account held by Customers => M : N**  One Customer can have more than one Accounts and also One Account can be held by one or more Customers, so the relationship between Account and Customers is many to many relationship. * **Loan availed by Customer => M : N**  (Assume loan can be jointly held by many Customers).  One Customer can have more than one Loans and also One Loan can be availed by one or more Customers, so the relationship between Loan and Customers is many to many relationship.     Identification of Entities = 02  Identification of Attributes = 02  Identification of Relations = 02  Drawing the ER diagram with all constraints and cardinality = 02  Drawing the Relation model with proper references given = 02 | 10 | 4 | 2 |

|  |  |
| --- | --- |
| **Course Outcome** | |
| **CO1** | Understand and Apply Database Management Systems concepts to solve the given problem |
| **CO2** | Design solutions with societal and environmental concerns using modern tools to solve problems in Database Design domain |
| **CO3** | Analyze and develop Database Applications using SQL and NoSQL features by engaging in lifelong learning for emerging technology |
| **CO4** | Exhibit effective communication and engage in continuing professional development through experiential learning |
| **CO5** | Demonstrate skills like investigation, effective communication, working in team/Individual practices by implementing Database Design concepts and applications |

**M-Marks, BT-Blooms Taxonomy Levels, CO-Course Outcomes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Marks Distribution** | | **Particulars** | **CO1** | **CO2** | **CO3** | **CO4** | | **L1** | **L2** | **L3** | **L4** | | **L5** | | **L6** | |
| **Max Marks** | 10 | 20 | 20 | -- | | -- | 10 | 20 | 20 | | -- | | -- | |
| **Course Code:21AI4**  **Sem: IV** | | | | | | | **Date:**  **Duration: 20 Minutes** | | | | | | | | | | | |
| **Quiz - I** | | | | | | |  | | | | | | | | | | | |
| **Data Base Management Systems** | | | | | | | | | | | | | | | | | | |
| **SL. No** | **Question** | | | | | | | | | | | **M** | | **BT** | | **CO** | |
| **1** | Which symbol denote derived attributes in the ER Model?   1. Double ellipse (b) **Dashed ellipse**   (c) Squared ellipse (d) An ellipse with attribute name underlined. | | | | | | | | | | | 1 | | 1 | | 1 | |
| **2** | Consider a company ABC which stores the information of their employees and their dependents. The Empid, EName, DOB,AadharID is stored in Employee Table. In Dependent Table, DName, DOB, Gender and relationship is stored. Partial key is present in\_\_\_\_\_\_\_\_ Table and partial key is\_\_\_\_\_\_\_\_\_   1. Employee , AadharID 2. Employee .Dname 3. **Dependent, Dname**   Dependent, DOB. | | | | | | | | | | | 1 | | 2 | | 1 | |
| **3** | Given the list of Attributes of Entity Type Employee (EmpId, Name, DoB, Address, Designation, DOJ, YOE, Skillset), Identify the all possible attribute types.   1. Simple ,Stored ,Derived,Composite 2. **Simple, Composite, Derived, Stored,Multivalue** 3. Simple, Composite, Derived, Stored 4. Simple, Complex, Composite, Multivalue, Stored | | | | | | | | | | | 1 | | 1 | | 1 | |
| **4** | Consider a relationship between Strong Entity type and Weak entity type . The participation of Weak Entity type to Strong Entity \_\_\_\_\_\_  a) Always Partial Participation  b) **Always Total Participation**  c) Depends on participation of Strong Entity type with Weak Entity type  d) May be Total or Partial participation | | | | | | | | | | | 1 | | 2 | | 1 | |
| **5** | Consider the following sequence of Statements execution and identify which statement does not get executed and the reason.  **Statement 1:**  Create table Department(DeptID varchar(10) primary key, Dname varchar(10), ContactNumber bigint(10));  **Statement 2:**  Create table Employee(EmpidID varchar(10) primary key, Ename varchar(10),DeptId varchar(10), Foreign key(DeptID) references Department(DeptID));  **Statement 3:**  Insert into Employee Values(“emp100”,”raju”,”D100”);  **Statement 4:**  insert into Employee Values(“emp200”,”raju”,”NULL”);   1. Statement 1 does not get executed, Referential integrity 2. Statement 2 does not get executed, on delete cascade not specified 3. **Statement 3 does not get executed** ,Referential integrity 4. All the statements get executed | | | | | | | | | | | 1 | | 3 | | 3 | |
| **6** | Consider the relation Stud\_marks (Stdid, Marks\_test1, Marks\_ test2). The query  Select max(Marks\_test1) from Stud\_marks where Marks\_test1 not in (Select max(Marks\_test1) from Stud\_marks) will give the following result   1. Highest marks in test1 2. Highest marks in test2 3. **Second highest marks in test1** 4. Second highest marks in test2 | | | | | | | | | | | 1 | | 3 | | 2 | |
| **7** | For the relation, Employee(SSN, Ename, Address,DOB) , Select \* from Employee WHERE MOD (SSN, 2) <> 0 displays   1. All the attributes of the employee table 2. All the tuples of the employee table 3. All the tuples whose SSN is 2 4. **All the odd rows from the table** | | | | | | | | | | | 1 | | 3 | | 3 | |
| **8** | select empname from employee where salary >= select salary from employee where empid = 101;  a)Name of employee is retrieved who receives salary greater than salary  of the employee with empid=101  b) Name of employee is retrieved who receives salary greater than or  equal to salary of the employee with empid=101  c) The statement is not executed.  d)None  **Answer a** | | | | | | | | | | | 1 | | 2 | | 2 | |
| **9** | EMPLOYEE DEPARTMENT   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Empid | Name | DeptID |  | DeptID | Dname | | E100 | A | D101 |  | D101 | HR | | E200 | B | D101 |  | D102 | Finance | | E300 | C | D102 |  | D103 | Sales | | E400 | D | NULL |  |  |  |   select e.\* from employee e left outer join dept d on e.deptid=d.depti  d;  The sql statement will result in \_\_\_\_ tuples   1. 0 b)1 c) 3 **d) 4** | | | | | | | | | | | 1 | | 2 | | 1 | |
| **10** | Based on M:N cardinality, A New table is created with specifying the Key attributes of the participating entity types.Both the key attributes are specified individually as foreign keys.Composite Primary key is not  Specified during table creation.It is decided to add the composite primary key to the existing table.Which statement is appropriate  (Assume Table name is Works\_On, attributes are ssn,projno)   1. Update Works\_On set primary key=(ssn,projno); 2. Update table works set composite primary key==(ssn,projno); 3. **Alter table Works\_On add primary key(ssn, projno);** 4. Alter table Works\_On add composite primary key(ssn, projno); | | | | | | | | | | | 1 | | 1 | | 1 | |
| **11** | The Department of AIML, RVCE is organizing Talent Hunt Search Program for First year Students. Every Student has to participate in at least one of the events. More than one Student can participate in an event .Every event has certain students registered for it  Identify the cardinality and the participation constraint   1. 1: N , Both Side Partial participation 2. M:N , Partial and Total Participation 3. **M:N , Both side Total Participation** 4. M:N , Total and Partial Participation | | | | | | | | | | | 1 | | 3 | | 3 | |
| **12** | Let A(a,b,c,d) and B(w,x,y,z) be to relations in which a is foreign key of A that refers to the primary key of B. Consider the four options in sequence  (i)Insert into A (ii) Insert into B (iii) Delete from A (iv) Delete from B. Which of the following is correct about the referential integrity constraint with respect to above  (a) Operation (i) and (ii) will cause violation  (**b) Operation (iv) and (i) will cause violation**  **( c)** Operations (ii) and (iii) will cause violation.  (d) Operations (iii) and (iv) will cause violation. | | | | | | | | | | | 1 | | 2 | | 2 | |
| **13** | Consider an relation EMP(Eid , ename, address) where Eid is the primary key. Following operations are performed on the relation  (i)Insert into EMP values (001, ‘Raju’,’ NULL’);  (ii)Insert into EMP values (‘NULL’, ‘Raju’, “Bangalore”);  Following is true with respect to the operations performed   1. Only i violates entity integrity constraint 2. **Only ii violates entity integrity constraint** 3. Both i and ii violates entity integrity constraint   None of the above | | | | | | | | | | | 1 | | 3 | | 3 | |
| **14** | An Entity has E has a,b,c as single value attribute and d as multivalued attribute with a as primarykey. There is another Entity F with w,x,y,z as single valued attributes . There is relation R1 between entity E and F. When designing the relation schema for the above scenario,which of the following hold good   1. Create a relation with R having a as attribute and d as foreign key in R 2. Create a relation with R having a as primary key and d as foreign key in R 3. Create a relation with R having a as foreign key and d as primary key in R   **Create a relation with R having d as attribute and a as foreign key in R** | | | | | | | | | | |  | | 2 | | 2 | |
| **15** | In the query if DISTINCT key is used it will   1. Not eliminate duplicate values 2. Retain duplicate values 3. **Eliminate duplicate values**   None of the above | | | | | | | | | | | 1 | | 2 | | 1 | |
| **16** | Given the Sql Create statement, identify which of the following statements are true.  Create table ACT\_IN(actid varchar(10),movieid varchar(10),  Constraint Fk\_actid\_ACT\_IN foreign key(actid) references actor(actid) on delete cascade on update cascade, Consraint fk\_movieid\_ACT\_IN foreign key(movieid) references movie(movieid) on delete cascade on update cascade, primary key(actid,movieid))   1. Single/Multiple related records can be updated or deleted from Actor and Movie Tables. 2. Duplicate Records will be allowed. 3. Both the foreign key constraint can be dropped using alter statement. 4. NULL value can be inserted for the actid and movieid attribute. 5. i and ii b) i and iii c) i and iv d) ii and iv   **Answer b** | | | | | | | | | | | 1 | | 2 | | 2 | |
| **17** | Consider the following SQL statements, Which of the following is true.   1. select DISTINCT (dept\_name) from Employee e inner join Departmentd on e.dept\_id = d.dept\_id; 2. select DISTINCT (dept\_name) from Employee e, Department d where e.dept\_id = d.dept\_id; 3. select DISTINCT (dept\_name) from Employee e join Department   d on e.dept\_id = d.dept\_id;   1. Output of all the statements will be same 2. Output of statement 1 and 2 will be same 3. Output of statement 1 and 3 will be same 4. All the Statements will result in different output   **Answer a** | | | | | | | | | | |  | | 2 | | 2 | |
| **18** | Which of the following statement is true about the SQL query   1. Not All attributes which appear in group by clause need not appear in select statement 2. SQL query can have having clause with group by clause 3. All attributes which appear in group by clause should appear in select statement 4. SQL query can have having clause without group by clause 5. (i) and (ii) (b) (i) and (iii) (c) **(ii) and (iii)** (d) (ii) and (iv) | | | | | | | | | | |  | | 1 | | 1 | |
| **19** | Let P and Q be two entities in an E-R diagram with simple single-valued attributes. R11 and R12 are two relationships between P and Q, where R1 is one-to-many and R2 is one-to-one. R11 and R12 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model? **(A) 2** (B) 3 (C) 4 (D) 5 | | | | | | | | | | | 1 | | 2 | | 2 | |
| **20** | Basic operations for changing the database are  Insert (b) Delete (c) Modify **(d) All** | | | | | | | | | | | 1 | | 2 | | 1 | |

|  |  |
| --- | --- |
| **Course Outcome** | |
| **CO1** | Understand and Apply Database Management Systems concepts to solve the given problem |
| **CO2** | Design solutions with societal and environmental concerns using modern tools to solve problems in Database Design domain |
| **CO3** | Analyze and develop Database Applications using SQL and NoSQL features by engaging in lifelong learning for emerging technology |
| **CO4** | Exhibit effective communication and engage in continuing professional development through experiential learning |
| **CO5** | Demonstrate skills like investigation, effective communication, working in team/Individual practices by implementing Database Design concepts and applications |

**M-Marks, BT-Blooms Taxonomy Levels, CO-Course Outcomes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Marks Distribution** | **Particulars** | **CO1** | **CO2** | **CO3** | **CO4** | **CO5** | **L1** | **L2** | **L3** | **L4** | **L5** | **L6** |
| **Max Marks** | 6 | 2 | 2 | -- | -- | 3 | 4 | 3 | -- | -- | -- |