DBMS LAB VIVA QUESTIONS 09-09-2024

1. A_____ is a query that retrieves rows from more than one table or view:
a) Start
b) End
c) Join
d) All of the mentioned

2. Which are the join types in join condition:
a) Cross join
b) Natural join
c) Join with USING clause
d) All of the mentioned

3. Which product is returned in a join query have no join condition:
a) Equijoins
b) Cartesian
c) Both Equijoins and Cartesian
d) None of the mentioned

4. Which is a join condition contains an equality operator:
a) Equijoins
b) Cartesian
c) Both Equijoins and Cartesian
d) None of the mentioned

5. Which join refers to join records from the write table that have no matching key in the left table are include in the result set:
a) Left outer join
b) Right outer join
c) Full outer join
d) Half outer join

6. Which view that contains more than one table in the top-level FROM clause of the SELECT statement:
a) Join view
b) Datable join view
c) Updatable join view
d) All of the mentioned

7. For the view Create view instructor_info as

SELECT ID, name, building

FROM instructor, department

WHERE instructor.dept name= department.dept name;

If we insert tuple into the view as insert into instructor info values (’69987’, ’White’, ’Taylor’);
What will be the values of the other attributes in instructor and department relations?
a) Default value
b) Null
c) Error statement
d) 0

8.CREATE VIEW faculty **AS**

SELECT ID, name, dept name

FROM instructor;

Find the error in this query.
a) Instructor
b) Select
c) View …as
d) None of the mentioned

9. If we want to retain all duplicates, we must write _____ in place of union.
a) Union all
b) Union some
c) Intersect all
d) Intersect some

10.(**SELECT** course id

FROM SECTION

WHERE semester = ’Fall’ **AND YEAR**= 2009)

EXCEPT

(**SELECT** course id

FROM SECTION

WHERE semester = ’Spring’ **AND YEAR**= 2010);

This query displays
a) Only tuples from second part
b) Only tuples from the first part which has the tuples from second part
c) Tuples from both the parts
d) Tuples from first part which do not have second part

Question:1

Consider the following relational schema for a database system of an online marketplace:

- **Customer**(*CustomerID*, Name, City, Age)
- **Order**(*OrderID*, CustomerID, OrderDate, TotalAmount)
- **Product**(*ProductID*, ProductName, Price, Category)
- **OrderDetails**(*OrderID*, *ProductID*, Quantity)
- **Supplier**(*SupplierID*, SupplierName, City)
- **Supply**(*SupplierID*, *ProductID*, Quantity)

Sub-questions:

a) Find the names of customers who have placed an order for all products in the category 'Electronics'.

b) List the names of suppliers who provide products that have never been ordered.

c) Write a query to find the names of customers who have spent more than the average total amount spent across all orders.

d) Find the names of customers who have placed orders for products supplied by suppliers located in the same city as the customer.

e) List the names of products that have been ordered by customers from all cities where the product's suppliers are located.

Question 2: Write a Pl/SQL Block to check whether a given number is prime or not?

Question:2

Consider the following relational schema for an online library system:

- Member(*MemberID*, Name, City, Age)
- Book(*BookID*, Title, Author, Price)
- Borrow(*MemberID*, *BookID*, BorrowDate, ReturnDate)
- Author(*AuthorID*, Name, Country)
- WrittenBy(*BookID*, *AuthorID*)

Sub-questions:

a) Find the names of members who have borrowed books written by the author "George Orwell".

b) List the titles of books that have never been borrowed.

c) Write a query to find the names of authors whose books are all priced above the average price of books in the library.

d) Find the names of members who have borrowed more books than the average number of books borrowed by all members.

e) List the titles of books borrowed by members from the same city as the author of the book.

Question 2: Write a Pl/SQL Block to check whether a given number is Armstrong or not?

Question:3

Consider the following schema for a university database:

- **Student**(*StudentID*, Name, Age, DeptID)
- **Department**(*DeptID*, DeptName)
- **Course**(*CourseID*, CourseName, DeptID)
- **Enrollment**(*StudentID*, *CourseID*, Grade)
- **Professor**(*ProfID*, Name, DeptID)
- **Teaches**(*ProfID*, *CourseID*)

Sub-questions:

a) Find the average grade for each course.

b) List the names of students who have obtained the highest grade in any course.

c) Write a query to find the total number of students enrolled in each department.

d) Find the names of professors who teach more than 3 courses.

e) List the names of departments with an average student age greater than 21.

Question 2: Write a Pl/SQL Block to check create a calculator using ‘case’?

Question:4

Consider the following relational schema for a company's database:

- **Employee**(*EmpID*, Name, Age, DeptID, Salary)
- **Department**(*DeptID*, DeptName)
- **Project**(*ProjID*, ProjName, DeptID)
- **WorksOn**(*EmpID*, *ProjID*, HoursWorked)
- **Manager**(*EmpID*, DeptID)

Sub-questions:

a) Write a query to find the total number of employees in each department.

b) List the names of departments where the average salary of employees is greater than $50,000.

c) Find the names of employees who are working on more than 2 projects.

d) Write a query to display the total hours worked by each employee and include only those employees who have worked more than 100 hours in total.

e) List the department names and the maximum salary of employees in each department, but only include departments where the maximum salary is greater than $80,000.

Question 2: Write a Pl/SQL Block to check whether a given number is even or odd?

Question:5

Consider the following schema for a university's academic management system:

- **Student**(*StudentID*, Name, Age, DeptID)
- **Department**(*DeptID*, DeptName)
- **Course**(*CourseID*, CourseName, DeptID)
- **Enrollment**(*StudentID*, *CourseID*, Grade)
- **Professor**(*ProfID*, Name, DeptID)
- **Teaches**(*ProfID*, *CourseID*)
- **Alumni**(*StudentID*, GraduationYear)

Sub-questions:

a) Write a query to find the names of students who have never enrolled in any course.

(Hint: Use a **LEFT JOIN** or **NOT IN** operation.)

b) List the names of students who have enrolled in at least one course taught by professors from the same department as the student.

(Hint: Use a **JOIN** to match students and professors by department.)

c) Write a query to find the names of all students who are either current students or alumni, but not both.

(Hint: Use the **UNION** and **MINUS** (or **EXCEPT**) set operations.)

d) Find the names of professors who teach courses in departments other than their own.

(Hint: Use a **JOIN** and a condition to compare the professor's department with the course's department.)

****e) Write a query to list the names of students who have enrolled in all courses offered by the "Computer Science" department.**

(Hint: Use a **JOIN** and set operation or **NOT EXISTS**.)

Question:6

Consider the following schema for a hospital database system:

- **Doctor**(*DoctorID*, Name, Specialization, DeptID)
- **Patient**(*PatientID*, Name, Age, Gender)
- **Department**(*DeptID*, DeptName)
- **Appointment**(*AppointmentID*, *DoctorID*, *PatientID*, AppointmentDate, Diagnosis)
- **Prescription**(*PrescriptionID*, *AppointmentID*, Medication, Dosage)

Sub-questions:

a) Create a view that shows the names of all doctors along with their department names.

(Hint: Use a JOIN between the **Doctor** and **Department** tables.)

b) Write a query using the view created in part (a) to list the names of doctors from the "Cardiology" department.

c) Create a view that shows all patient information along with their doctor's name and the appointment date.

(Hint: Use a JOIN between **Doctor**, **Patient**, and **Appointment** tables.)

d) Write a query using the view created in part (c) to list the names of patients who have appointments scheduled with doctors specializing in "Pediatrics".

e) Write a query to update the view created in part (a) to change the department name of a doctor to "Neurology".

(Hint: Use the UPDATE statement on the view.)