Aiym B. Kabiden

18BD110317

Quiz

Variant 2

Part 1 (5 points)

Please answer the following MCQ:

1. Suppose relation R(A,C), S(B,C,D) has the following tuples:

Compute the natural join of R and S. Which of the following tuples is in the result? Assume each tuple has schema (A,B,C,D).

A.(3,4,6,9)

B.(5,6,4,6)

C.(1,2,4,6)

D.(5,6,7,8)

2. Suppose relation R(A,B,C) has the following tuples:

and relation S(A,B,C) has the following tuples:

Compute the intersection of the relations R and S. Which of the following tuples is in the result?

A.(2,2,6)

B.(2,5,4)

C.(4,2,3)

D.(2,5,3)

3. Consider the following query:

Select \* From Student, Apply, CollegeWhere Student.sID = Apply.sID and Apply.cName = College.cNameAnd Student.GPA > 1.5 And College.cName < 'Cornell'

Suppose we are allowed to create two indexes, and assume all indexes are tree-based. Which two indexes do you think would be most useful for speeding up query execution?

A.Student.sID, College.cName

B.Student.sID, Student.GPA

C.Apply.cName, College.cName

D.Apply.sID, Student.GPA

4. Consider a database containing two relations

Borrower(customer-name, loan-number) Loan(loan-number, amount)

We define a view loan-info as

CREATEVIEW loan-info as

SELECT customer-name,amount

FROM Borrower, Loan

WHERE Borrower.loan-number=Loan.loan-number

Consider the following insertions

1. INSERT INTO Borrower VALUES ('Johnson', null)

INSERT INTOLoan VALUES (null,1900)

1. INSERT INTO Borrower VALUES ('Johnson', 1209)

INSERT INTOLoan VALUES (1209,1900)

Which of the above operations will have the effect of inserting tuple ("Johnson",1900) into loan-info (assuming it is not there previously)?

A. I only

B. II only

C. I and II

D. None of the above

5. Consider the following SQL table declaration:

CREATE TABLE R (a INT, b INT, c INT, CHECK( [fill-in] ));

Currently R contains the tuples (1,4,14), (2,3,15), and (3,3,16).

Which of the following tuple-based CHECK constraints will cause the following insertion to be rejected?

INSERT INTO R VALUES (4,4,9);

A.b < (SELECT MIN(c) FROM R)

B.b > (SELECT AVG(a) FROM R)

C.c >= (SELECT SUM(b) FROM R)

D.a <= ALL (SELECT c - b FROM R)

Part 2 (5 points)Please write SQL queries for following tasks. Consider following schemas:

Таблица:

create table Movie(

title varchar(25),

mID serial primary key,

year integer,

director varchar(25)

);

create table Reviwer(

rID serial primary key,

name varchar(25)

);

create table Rating(

rID serial references Reviewer,

mID serial References movie,

stars integer,

rating integer,

ratingDate date

);

6. List movie titles and average ratings, from highest-rated to lowest-rated. If two or more movies have the same average rating, list them in alphabetical order.

select title, avg(stars) as averageStars from Movie join (Rating join Reviewer using (rID) using(mID)

group by title order by averageStars desc, title;

7. For each movie that has at least one rating, find the lowest number of stars that movie received. Return the movie title and number of stars. Sort by movie title.

select title, min(stars) from Movie join Rating using(mID) group by title order by title;

8. Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date.

select name from Reviewer, Rating where Reviewer.rID = Rating.rID and Rating.raatingDate is null;

9. Create materialized view for the next statement. Find the titles of all movies that have no ratings.

create materialized view view1 as select title from Movie join (Rating join Reviwer using (rID)) using (mID)

where stars is null group by title;

10. Create role with any name and give select and update privileges on the previous view.

create role SelectUpdateRole;

grant select, update on view1 to SelectUpdateRole;