ⓘ This quiz has been regraded; your score was not affected.

Quiz 5 Views and Triggers

Due Oct 4 at 10:30am **Points** 100 **Questions** 8
Available Oct 4 at 10am - Oct 4 at 10:30am 30 minutes **Time Limit** None

Instructions

This quiz is 30 minutes long and contains multiple choice questions.

Having an issue with the quiz? Please send an email to the course staff (rkheni@wpi.edu) (mailto:cvieira@wpi.edu) with "CS542 Quiz" included in the subject line any time during the quiz. If you require help through zoom then please join the zoom link https://wpi.zoom.us/j/2094237642 (https://wpi.zoom.us/j/2094237642) .

This quiz was locked Oct 4 at 10:30am.

Attempt History

	Attempt	Time	Score	Regraded
LATEST	Attempt 1	27 minutes	75 out of 100	75 out of 100

Score for this quiz: **75** out of 100
Submitted Oct 4 at 10:30am
This attempt took 27 minutes.

Question 1

5 / 5 pts

Which of the following is **NOT** a reason why views are useful?

○ Views offer a layer of independence from the base table(s) in case they change

Correct!

● Views can store the result of executions which can be used in later queries, increasing performance.

○ Database administrators can allow users access to views but not base tables to increase security.

○ Views can store complicated or frequently used queries for later

Question 2

0 / 7.5 pts

Given the following schema and view:

USER(login, ssn, dateOfBirth, firstName, lastName)

CREATE VIEW MyView AS SELECT ssn, dateOfBirth FROM USER;

Which of the following statements is true?

Note: Insertable means that you can insert new records into the view, and updatable means that you can change the values of records in a view.

ou Answered

● MyView is insertable but not updatable

orrect Answer

○ MyView is updatable but not insertable

○ MyView is updatable and insertable

○ MyView is neither insertable nor updatable

Question 3

20 / 20 pts

Given the following two tables:

Student(ID, name, address);

Grade(Sequence, StudentID, exam1, exam2, exam3, total);

Note: Each student has at most one record in Grade table.
Grade.StudentID references Student.ID

Create a view (named ExamAvg) that computes the average of each exam. The output from the view should be :

Test	Average
Exam1	---

Correct!

●

```
Create View ExamAvg As (

      Select ‘Exam 1’ As Test, Avg(exam1) As Average From Grade
Union

       Select ‘Exam 2’ As Test, Avg(exam2) As Average From Grade
Union

       Select ‘Exam 3’ As Test, Avg(exam3) As Average From Grade);
```

○

```
Create View ExamAvg As (

      Select Avg(exam1) As Average From Student
Union

       Select Avg(exam2) As Average From Student
Union

       Select Avg(exam3) As Average From Student);
```

○

```
Create View ExamAvg As (

      Select ‘Exam 1’ As Test, Avg(exam1) As Average From Student
Union

       Select ‘Exam 2’ As Test, Avg(exam2) As Average From Student
Union

       Select ‘Exam 3’ As Test, Avg(exam3) As Average From Student);
```

```
Create View ExamAvg As (
       Select ‘Exam 1’ As Test, Avg(exam3) As Average From Grade
Union
        Select ‘Exam 2’ As Test, Avg(exam1) As Average From Grade
Union
```

○

```
        Select ‘Exam 3’ As Test, Avg(exam2) As Average From Grade);
```

Question 4

15 / 15 pts

Given the following two tables:

Student(ID, name, address);

Grade(Sequence, StudentID, exam1, exam2, exam3, total);

Note: Each student has at most one record in Grade table. Grade.StudentID references Student.ID

Use the ExamAvg view (as well as the base tables if needed) to report the student Ids whose total grade is larger than the sum of the three exams’ averages, i.e., Grade.total > (Exam1’s avg + Exam2’s avg + Exam3’s avg)

○

```
Select total
From Student
Where total > (Select sum(Average) From ExamAvg);
```

○

```
Select StudentID
From Grade
Where total > (Select count(Average) From ExamAvg);
```

○

```
Select total
From Grade
Where total > (Select sum(Average) From ExamAvg);
```

Correct!

Select StudentID

From Grade

● Where total > (Select sum(Average) From ExamAvg);

Question 5

Original Score: 7.5 / 7.5 pts Regraded Score: 7.5 / 7.5 pts

ⓘ This question has been regraded.

Check all of the following situations in which the **:new** reference can be used within a trigger

□ Statement-level triggers

Correct!

☑ Row-level triggers

ou Answered

☑ INSERT triggers

ou Answered

☑ UPDATE triggers

□ DELETE triggers

Question 6

0 / 7.5 pts

Given the following relational schema, imagine you are trying to write a trigger that makes sure that the first and last names are automatically capitalized during inserts or updates, and if not, the trigger will automatically correct the error by capitalizing the supplied values for first name and last name. This trigger should not raise application errors.

User(id, firstName, lastName, email)

Which of the following granularities would make sense for this trigger?

orrect Answer

○ A BEFORE trigger

ou Answered

● An AFTER trigger

Question 7

7.5 / 7.5 pts

Given the same relational schema in the previous question, imagine you are trying to write a trigger that checks if a newly inserted row contains a valid email address (and throws an error if not)

Which of the following granularities would make sense for this trigger?

Correct!

● A BEFORE trigger

○ An AFTER trigger

Question 8

20 / 30 pts

Given the relational schema below, build a trigger that logs any change in an account balance into the ACCOUNTLOG table. This means that any time that any time someone **updates** the balance of an existing account, a new row in ACCOUNTLOG should be created with the "changeInBalance" field representing the change in the account balance, the "id" field being the account for which the balance changed.

ACCOUNT(id, balance)

ACCOUNTLOG(accountId, timeOfChange, changeInBalance)

FOREIGN KEY (accountId) REFERENCES ACCOUNT(id)

Select the right options for each missing part of this trigger to correctly construct a trigger that fulfills the above requirements.

```
CREATE OR REPLACE TRIGGER AccountLogTrigger
[ Select ]

FOR EACH ROW

BEGIN
    [ Select ]

END;
```

Answer 1:

orrect Answer

AFTER UPDATE OF balance ON ACCOUNT

ou Answered

AFTER INSERT OR UPDATE ON ACCOUNT

Answer 2:

Correct!

FOR EACH ROW

Answer 3:

Correct!

INSERT INTO ACCOUNTLOG VALUES (:new.id, SYSDATE, :new.balance - :old.balance);

Quiz Score: **75** out of 100