

CSE 180, Final Exam, Fall 2023, Shel Finkelstein

Multiple Choice Questions (Part I) Answered on Scantron Sheet

Test Form letter: A

This first Section (Part I) of the Fall 2023 CSE 180 Final is Multiple Choice and is double-sided. Answer all multiple choice questions on your Scantron sheet. You do not have to hand in this first Section of the Exam, but you must hand in the Scantron sheet, with your Name and Student ID filled in (including marking bubbles below) on that Scantron sheet. Please be sure to use a #2 pencil to mark your choices on this Section of the Final.

Name and Student ID must also be filled in by shading letters/numbers on the form. You must also mark the **version** ("Test Form letter **A**") of the Multiple Choice section that you took. The box for Test Form letter is at the top of the Scantron sheet, just to the left of the Multiple Choice questions.

The separate second Section (Parts II and III) of the Final is not Multiple Choice and is single-sided, so that you have extra space to write your answers. If you use that extra space, please be sure to write the number of the problem that you're solving next to your answer. Please write your Name, Email and Student ID on the second Section of the Exam, which you must hand in. You may use any writing implement on this Section of the Exam.

At the end of the Final, please be sure to hand in **both your Scantron sheet for this first Section of the Exam** and also the separate second Section of the Exam. You must also show your UCSC id when you hand them in. You do not have to hand in this Multiple Choice section of the Final, but you must show us the Test Form letter that's printed on your exam.

Part I: (40 points, 2 points each)

Answer the questions in Part I on your Scantron sheets, which should have your name, email and UCSC id on them. Select the **best answer** for each of the following. For some questions, a choice may be "**All of the answers are correct**", or "**None of the other answers is correct**", so read answer choices carefully.

Question 1: The relational model has logical independence. Which of the following best illustrates "logical independence" for the relational model?

- a) In relational database you write queries which describe that results that you want, not how to obtain those results.
- b) You can also write your queries using a Hierarchical Model, with trees.
- c) When you write queries, you don't have to know how the tables used are stored, or what indexes exist.
- d) If you change a table into a view that has the same output, then your queries on that table will still work correctly.
- e) None of the other answers is correct.

Question 2: What does Serializability mean?

- a) Transactions must be executed one at a time.
- b) Transactions don't have to be executed one at a time, but the effect of a sequence of transaction is as if they had been executed one at a time.
- c) A transaction that rolls back has no effect on the database.
- d) If a transaction commits, its changes are permanent, even if there are failures.
- e) Constraints are always maintained by the database system.

Question 3: $R(A,B,C,D,E)$ is a relation. A's domain has 3 values, B's domain has 10 values, C's domain has 2 values, D's domain has 9 values, and E's domain has 16 values. (A, B, C) is the Primary Key of R. D is UNIQUE, and it can be NULL. E is UNIQUE, and it can't be NULL. What is the maximum number of different tuples that there can be in an instance of R?

- a) 9
- b) 10
- c) 16
- d) 60
- e) 8640

Question 4: Here are two SQL queries on Union-Compatible relations R(A, B, C) and S(A, B, C).

Query1:

```
( SELECT DISTINCT *  
  FROM R  
  WHERE A > 123 )
```

UNION ALL

```
( SELECT DISTINCT *  
  FROM S  
  WHERE B < 456 );
```

Query2:

```
( SELECT *  
  FROM R  
  WHERE A > 123 )
```

UNION

```
( SELECT *  
  FROM S  
  WHERE B < 456 );
```

Which statement is true about the relationship of the results of Query1 and Query2?

- a) The same tuples will appear in both results, and tuples will appear with the same multiplicity in both.
- b) The same tuples will appear in both results, but some tuples may appear more times in the result of Query1 than in the result of Query2.
- c) The same tuples will appear in both results, but some tuples may appear more times in the result of Query2 than in the result of Query1.
- d) There may be some tuples that appear in the result of Query1 which don't appear in the result of Query2.
- e) There may be some tuples that appear in the result of Query2 which don't appear in the result of Query1.

Question 5: If an instance of relation R1(A, B, C) has 12 different tuples in it, and an instance of relation R2(C, D) has 6 different tuples in it, then how many tuples will there be in the result if the following SQL query is executed on those instances?

```
SELECT *  
FROM R1, R2  
WHERE R1.C = R2.C;
```

- a) 0
- b) Exactly 18
- c) Exactly 72
- d) Must be between 0 and 18. Could be 0 or 18, but could be values other than 0 and 18.
- e) Must be between 0 and 72. Could be 0 or 72, but could be values other than 0 and 72.

Question 6: StarsIn(starName, movieTitle, movieYear) is a table in our database in which (starName, movieTitle, movieYear) is the Primary Key. Assume that the tuples in StarsIn indicates that Harrison Ford starred in 5 movies in 2020, 4 movies in 2021, 1 movie in 2022, and no movies in any other years.

How many times does Harrison Ford appear in the answer to the following query?

```
SELECT s1.starName
FROM StarsIn s1
WHERE EXISTS ( SELECT *
                FROM StarsIn s2
                WHERE s1.movieTitle <> s2.movieTitle
                  AND s1.movieYear = s2.movieYear
                  AND s1.starName = s2.starName );
```

- a) 2
- b) 3
- c) 9
- d) 10
- e) 32

Question 7: Employees(name, age, salary) is a table in which name is the Primary Key. Which statement is true for the following query?

```
SELECT e1.name
FROM Employees e1
WHERE e1.salary =
      ( SELECT e2.salary
        FROM Employees e2
        WHERE e2.age = 21 );
```

- a) The query will always run successfully.
- b) The query will never run successfully.
- c) If there are two employees whose names are the same, there will be a runtime error.
- d) If there are two employees whose age is 21, there will be a runtime error.
- e) If there are no employees whose age is 21, there will be a runtime error.

Question 8: Students(student_id, name, address, age, major) is a table in which student_id is the Primary Key.

If (12345, 'Taylor Swift', 'Santa Cruz', 33, 'MUS') is a tuple in that table, and the following is executed, with no other work going on:

```
BEGIN TRANSACTION;
```

```
UPDATE Students  
SET age = age + 1  
WHERE name = 'Taylor Swift';
```

```
UPDATE Students  
SET major = 'CSE'  
WHERE age = 33;
```

```
ROLLBACK TRANSACTION;
```

then afterwards, what will be in the tuple that has student_id 12345?

- a) (12345, 'Taylor Swift', 'Santa Cruz', 33, 'MUS')
- b) (12345, 'Taylor Swift', 'Santa Cruz', 34, 'MUS')
- c) (12345, 'Taylor Swift', 'Santa Cruz', 33, 'CSE')
- d) (12345, 'Taylor Swift', 'Santa Cruz', 34, 'CSE')
- e) None of the other answers is correct.

Question 9: Employees(name, age, salary) is a table in which name is the Primary Key. Which statement correctly finds the names of all the employees who have the highest age?

- a) SELECT e.name
FROM Employees e
WHERE e.age = MAX(e.age);
- b) SELECT e.name
FROM Employees e
WHERE e.age = MAX(e2.age);
- c) SELECT e.name
FROM Employees e
WHERE e.age = (MAX(e2.age)
FROM Employees e2);
- d) SELECT e.name
FROM Employees e
WHERE e.age = MAX (SELECT e2.age
FROM Employees e2);
- e) None of the other answers is correct.

Question 10: Sailors(sid, sname, rating, age) is a table in which sid is the Primary Key. Which of the following would be an illegal HAVING clause for the SQL query below?

```
SELECT rating, MAX(age)
FROM Sailors
GROUP BY rating
HAVING ?????;
```

- a) HAVING sname = 'Zorba'
- b) HAVING MAX(age) <= 50
- c) HAVING EVERY(age >= 18)
- d) HAVING COUNT(DISTINCT age) >= 10
- e) All of these HAVING clauses would be illegal.

Question 11: Which statement about indexes is correct?

- a) SQL applications must choose which indexes to use when executing a SQL statement.
- b) SQL applications must handle updates on indexes, so that the indexes are always correct.
- c) Indexes always improve overall performance of the SQL statements in a workload.
- d) If A and B are two of the attributes of a table, then having an index on (A, B) in that order has the same benefits as having an index on (B, A).
- e) None of the other answers is correct.

Question 12: Assume that the Employees table has been created as follows:

```
CREATE TABLE Employees (  
    name          CHAR(20) PRIMARY KEY,  
    age           INTEGER NOT NULL DEFAULT 18,  
    salary        INTEGER,  
    CHECK (age < 21 OR salary >= 4000)  
);
```

Here are three INSERT statements. Which will result in an error?

- i) INSERT INTO Employees VALUES ('Sleepy', 30, 3500);
 - ii) INSERT INTO Employees(name, salary) VALUES ('Bashful', 2000);
 - iii) INSERT INTO Employees(name, age) VALUES ('Happy', 25)
- a) None of these statements will result in an error.
 - b) Only statement i) will result in an error.
 - c) Only statement ii) will result in an error.
 - d) Only statement iii) will result in an error.
 - e) Two of these statements will result in an error.

Question 13: Which of these statements is true for Relational Algebra?

- a) Union is Commutative:
$$R \cup S = S \cup R$$
- b) If C1 and C2 are conditions on the attributes of R, then:
$$\sigma_{C1} (\sigma_{C2} (R)) = \sigma_{C2 \text{ AND } C1} (R)$$
- c) Product is Associative:
$$(R \times S) \times T = R \times (S \times T)$$
- d) Intersection can be expressed using Difference as follows:
$$R \cap S = R - (R - S)$$
- e) All of the answers are correct.

Question 14: Relations R(A,B,C,D) and S(A,B,C,D) have all their attributes in common. For this R and S, which of the following always equals $R \bowtie S$, the Natural Join of R(A,B,C,D) and S(A,B,C,D)?

- a) $R \cup S$
- b) $R \cap S$
- c) $R \times S$
- d) $R - S$
- e) None of the other answers is correct.

Question 15: Which is/are advantages of Stored Procedures (and Functions)?

- a) After a Stored Procedure has been written, it can be used and re-used by anyone who has the authority to execute it.
- b) A Stored Procedure can encapsulate complex processing whose inner workings might be hidden from users of that Stored Procedure.
- c) A user may be authorized to execute a Stored Procedure that accesses certain data, even though the user is not authorized to access that data directly.
- d) With a Stored Procedure, processing can be performed inside the database system, instead of moving data back and forth between the client and the database.
- e) All of the answers are correct.

Question 16: Sells(bar, beer, price) is a table, and RipoffBars(bar) is another table. What does this Trigger do, assuming that price represents a dollar amount?

```
CREATE TRIGGER PriceTrig
AFTER UPDATE OF price ON Sells
REFERENCING
    OLD ROW AS old_row
    NEW ROW AS new_row
FOR EACH ROW
WHEN(new_row.price > old_row.price + 5.00)
INSERT INTO RipoffBars VALUES (new_row.bar);
```

- a) If any row in Sells has a beer whose price is more than 5 dollars, then the bar in that row is inserted into RipoffBars.
- b) If a row is inserted into Sells that has a beer whose price is more than 5 dollars, then the bar in that inserted row is inserted into RipoffBars.
- c) If price is updated for a row in Sells, and the new price is more than 5 dollars higher than the old price, then the bar in that updated row is inserted into RipoffBars.
- d) If price is updated for a row in Sells, and the old price is more than 5 dollars higher than the new price, then the bar in that updated row is inserted into RipoffBars.
- e) If price is updated for a row in Sells, and the new price is more than 5 dollars higher than the old price, then the update returns an error.

Question 17: In OLAP, what is a dimension attribute?

- a) An attribute in a Fact table that appears in a key of a Dimension table.
- b) An attribute in a Fact table that is not in a key of one of the Dimension tables.
- c) The result of doing Roll-Up on a table.
- d) The result of doing Drill-Down on a table.
- e) None of the other answers is correct.

Question 18: Which of the following statements about keys and superkeys is true?

- a) Every superkey must be a key.
- b) There can't be more than one key for a relation.
- c) All keys for a relation must have the same number of attributes.
- d) Every attribute of a relation is part of some superkey.
- e) None of the other answers is correct.

Question 19: $R(A,B,C,D)$ is a relation, and the Functional Dependency $AB \rightarrow CD$ holds for that relation. Determine which of the following 4 Functional Dependencies must also hold for $R(A,B,C,D)$, and answer based on the choices below.

- i. $A \rightarrow CD$
- ii. $B \rightarrow CD$
- iii. $AB \rightarrow C$
- iv. $AB \rightarrow D$

- a) All of them must hold.
- b) Only i and ii must hold.
- c) Only iii and iv must hold.
- d) Only i and iii must hold.
- e) Only ii and iv must hold.

Question 20: R is a relation, and F is a Functional Dependency $X \rightarrow A$, where X is a set of attributes of R and A is an attribute of R . You want to determine if F holds for R . You inspect 10 million instances of R , and $FD F$ holds for all of them. But there are other instances of R . Which statement is correct?

- a) There cannot be a relation R and a Functional Dependency F that holds for 10 million instances of R .
- b) $FD F$ must hold for all instances of R .
- c) There must be an instance of R for which F does not hold.
- d) It's possible that $FD F$ holds for all instances of R , but it's also possible that there is an instance of R for which $FD F$ does not hold.
- e) None of the other answers is correct.