CMPS 180, Final Exam, Winter 2017, Shel Finkelstein

Multiple Choice Questions (Part I)
Answered on Scantron Sheet

This first Section (Part I) of the Winter 2017 CMPS 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, email and student id on that Scantron sheet. Please be sure to use a #2 pencil to mark your choices on this Section of the Final.

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, and show your UCSC id when you hand them in.

Part I: (44 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 is **"All of the Above"**, so read answer choices carefully.

Question 1: If an instance of relation R(A,B) has 10 different tuples in it, and an instance of relation S(B,C,D) has 6 different tuples in it, then how many tuples are there in the result if the following SQL query is executed on those instances?

```
SELECT *
FROM R, S
WHERE R.B = S.B;
```

a) 0
b) Exactly 16
c) Exactly 60
d) Between 0 and 16
e) Between 0 and 60

Question 2: We discussed ACID properties of transactions. What does Atomicity (the "A" in ACID) refer to for transactions?

a) Transaction execution is as if they were executed one at a time.
b) Transactions happen completely or not-at-all.
c) If a transaction commits, its changes are permanent, even if there are failures.
d) Business rules are always maintained by the database system.
e) Uncommitted (dirty) values from one transaction are never read by any other transaction.

Question 3: An instance of relation R(A,B) has m tuples in it, all the same, and an instance of relation S(A,B) has n tuples in it, which are all the same as the tuples in R(A,B). If R and S are Union-Compatible, and m and n are both at least 1, then how many tuples are there in the result of the following query?

```
( SELECT * FROM R )
    INTERSECT
( SELECT * FROM S );
```

a) 1
b) m + n
c) m – n (but not less than zero)
d) min(m, n)
e) max(m, n)

Question 4: Why might there be a runtime error when the following statement is executed on relations Customers(cname, age) and Activities(cname, slopeid, date)?

```
SELECT  Customers.cname
FROM Customers
WHERE Customers.cname =
        ( SELECT Activities.cname
          FROM Activities );
```

a) Some customer may have participated in no activities.
b) Some customer may have participated in exactly one activity.
c) Some customer may have participated in more than one activity.
d) For some cname in Activities, there might be no customer with that cname in Customers.
e) For some cname in Activities, there might be more than one customer with that cname in Customers.

Question 5: For the relation Employees(name, age, salary), suppose that no employees who are 65 or older, but there are employees who are under 65. Which employee names will appear in the result to the following query:

```
SELECT e1.name
FROM Employees e1
WHERE e1.salary < ALL
        ( SELECT e2.salary
          FROM Employees e2
          WHERE e2.age >= 65 );
```

a) There will be no employee names in the result.
b) All of the employee names will be in the result.
c) The result will be NULL.
d) The result will be UNKNOWN.
e) The query will cause a runtime error.

Question 6: Which statement is true for Relational Algebra Operations, where R is a relation and C1 and C2 are conditions on the attributes of R?

a) σ is Commutative: $\sigma_{C1}(\sigma_{C2}(R)) = \sigma_{C2}(\sigma_{C1}(R))$
b) $\sigma_{C1}(\sigma_{C2}(R)) = \sigma_{C1 \text{ AND } C2}(R)$
c) Union is Commutative (for union-compatible relations): $R \cup S = S \cup R$
d) Union is Associative (for union-compatible relations): $(R \cup S) \cup T = R \cup (S \cup T)$
e) All of the Above

Question 7: Employees(<u>name</u>, age, salary) is a relation, and the salary for an employee named Smith is 5000. What is Smith's salary after the following, assuming that this is the only transaction executing?

```
BEGIN TRANSACTION;
UPDATE Employees SET salary = salary + 1000 WHERE name='Smith';
UPDATE Employees SET salary = 2 * salary WHERE name='Smith';
ROLLBACK TRANSACTION;
```

a) 5000
b) 6000
c) 12000
d) Could be 5000, 6000 or 12000
e) None of the Above

Question 8: Employees(name, age, salary) is a relation, and the salary for an employee named Smith is 5000. Two different transactions T1 and T2 are executed by different people at approximately the same time, with Isolation Level Serializable. Both transactions commit. T1 and T2 are the only transactions executing. What is Smith's salary afterwards?

T1:

```
BEGIN TRANSACTION;
UPDATE Employees SET salary = salary + 1000 WHERE name='Smith';
COMMIT TRANSACTION;
```

T2:

```
BEGIN TRANSACTION;
UPDATE Employees SET salary = 2 * salary WHERE name='Smith';
COMMIT TRANSACTION;
```

a) Must be 5000
b) Must be 11000
c) Must be 12000
d) Must be either 11000 or 12000
e) Could be something other than 11000 or 12000

Question 9: The Employees(name, age, salary) table has been created with name as the primary key, salary having a default value of 9000, and with age and salary both allowed to be NULL. What is in the tuple with name 'Chou' after the following statement is executed, assuming that there was no tuple for 'Chou' before the statement was executed?

```
INSERT INTO EMPLOYEES(name, age)
  VALUES ('Chou', 25);
```

a) 'Chou', 25
b) 'Chou', 25, NULL
c) 'Chou', 25, 9000
d) 'Chou', NULL, 9000
e) There will be no tuple for Chou because salary was not supplied in the INSERT.

Question 10: For the relations Customers(cname, age) and Activities(cname, slopeid, date), what does the following Relational Algebra query do?

$$\Pi_{\text{Customers.cname, Customers.age}}(\sigma_{\text{Customers.cname=Activities.cname}}(\text{Customer X Activities}))$$

a) Finds cname and age for the customers who participated in at least one activity.
b) Finds cname and age for the customers who didn't participate in any activity.
c) Finds cname and age for each customer.
d) For each customer, gives cname, age and counts the number of activities that the customer participated in.
e) None of the Above

Question 11: For a table Movies(title, year, length, studio), which index will probably help the most for executing the following query?

```
SELECT title, year, length
FROM Movies
WHERE year > 2000 AND studio = 'Disney";
```

a) An index on year
b) An index on studio
c) An index on (year, studio) in that order
d) An index on (studio, year) in that order
e) Indexes on (year, studio) and (studio, year) are both the most helpful, and they're equally good.

Question 12: Assume that name is the primary key in the Beers table. This use of a CHECK constraint is legal in many SQL systems (but not in PostgreSQL):

```
CREATE TABLE Sells (
        bar     CHAR(20),
        beer    CHAR(20)      CHECK ( beer IN
                                        (SELECT name FROM Beers) ),
        price   REAL
);
```

For SQL systems in which this use of CHECK is legal, which is the best answer?

a) If you INSERT a row into the Sells table whose beer attribute value doesn't appear as a name in the Beers table, then there will be an error.
b) If you DELETE a row from the Beers table, and there is a row in the Sells table whose beer attribute value equals the beer name in the deleted row, then there will be an error.
c) If you UPDATE a row in the Sells table, changing its beer attribute value to a different beer whose name appears in the Beers table, then there will be an error.
d) Answer a) and b) are both correct, but answer c) is not correct.
e) Answer a), b) and c) are all correct.

Question 13: Which operation can an ALTER statement do?

a) Create a table
b) Drop a table
c) Delete all rows from a table
d) Add additional columns to a table
e) None of the Above

Question 14: Which of these is/are advantages of Stored Procedures?

a) Can make performance better by enabling operations to be performed on data without moving that data from the database to the client.
b) Can make programming easier by allowing code to be written once in a procedure, and then reused by anyone authorized to execute the procedure.
c) Can improve security by allowing people to be authorized to execute a procedure without authorizing them to access all the data used by the procedure.
d) All of the Above
e) None of the Above

Question 15: 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 > 8.00)
INSERT INTO RipoffBars
        VALUES(new_row.bar);
```

a) If any row in Sells has a beer that costs more than 8 dollars, then the bar in that row is inserted into RipoffBars.
b) If a row is inserted into Sells that has a beer that costs more than 8 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 8 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 8 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 8 dollars higher than the old price, then the update returns an error.

Question 16: Sells(bar, beer, price) is a table, where bar and beer are CHAR(20) and price is FLOAT. Assuming that myCon is a connection, what error appears in the following JDBC, which is supposed to print the beers and prices for Joe''s Bar?

```
Statement stmt = myCon.createStatement();
ResultSet Menu =
 stmt.executeQuery(" SELECT beer, price FROM Sells WHERE bar = 'Joe''s Bar' ");
while (Menu.fetch()) {
   // For each value in result, get values of beer and price, and print them
   System.out.println(Menu.getString(1), Menu.getFloat(2));
```

a) Should use *PreparedStatement*, not *Statement*
b) Should use *Menu.next*, not *Menu.fetch*
c) Should use *System.out.println(ResultSet.Float(1), ResultSet.getString(2));* not *System.out.println(Menu.getString(1), Menu.getFloat(2));*
d) Should use *stmt.updateQuery,* not *stmt.executeQuery*
e) In the SELECT statement, 'Joe''s Bar' has the single quote symbol appearing twice (between Joe and s) in *'Joe''s Bar'*, but it should only appear once, replaced by *'Joe's Bar'*

Question 17: R(A,B) and S(A,B) are union-compatible tables which may have duplicates. Here are two queries, Q1 (on the left) and Q2 (on the right) on those tables. Which statement is always correct for the results of those queries?

```
(SELECT DISTINCT *          (SELECT *
FROM R                      FROM R
WHERE A > 10)               WHERE A > 10)

UNION ALL                   UNION

(SELECT DISTINCT *          (SELECT *
FROM S                      FROM S
WHERE B < 300);             WHERE B < 300);
```

a) The results of Q1 and Q2 must always be the same.
b) Everything in the result of Q1 must always be in the result of Q2, but the results aren't always the same.
c) Everything in the result of Q2 must always be in the result of Q1, but the results aren't always the same.
d) The result of Q1 can never have duplicates.
e) None of the Above

Question 18: In On-Line Analytic Processing (OLAP), what does the term "roll-up" mean?

a) Separating attributes in a Fact table into Dimension attributes and Dependent attributes.
b) Joining a Fact table with its underlying Dimension tables.
c) Doing an Outer Join to obtain facts that are not in the Fact Table because values for those facts are 0 or NULL.
d) Aggregating values along one or more dimensions, e.g., taking a sum.
e) Disaggregating a value into its constituent parts, e.g., breaking a sum down into the values that led to the sum.

Question 19: For the following addressbook DTD:

```
<!DOCTYPE addressbook [
  <!ELEMENT addressbook (person*)>
  <!ELEMENT person
     (name, address+, ( homephone | ( workphone, mobile )* ), email?)>
  <!ELEMENT name          (#PCDATA)>
  <!ELEMENT address       (#PCDATA)>
  <!ELEMENT homephone     (#PCDATA)>
  <!ELEMENT workphone     (#PCDATA)>
  <!ELEMENT mobile        (#PCDATA)>
  <!ELEMENT email         (#PCDATA)>
]>
```

Does the following data correspond to that DTD?

```
<addressbook>
        <person>
                <name> Abraham Lincoln </name>
                <address> Washington D.C. 20500 </address>
                <workphone> (312) 555 9876 </workphone>
                <mobile> (773) 555 2543 </mobile>
        </person>
</addressbook>
```

a) Yes, the data corresponds to the DTD.
b) No, because there's only one person.
c) No, because there's no homephone.
d) No, because there's both a workphone and a mobile.
e) No, because there's no email.

Question 20: For the table R(A, B, C), which is a lossless join decomposition? R_1(A, B) is the projection of R on attributes A, B, and R_2(B,C) is the projection of R on attributes B, C.

a) R_1(A, B) and R_2(B, C), if there are no functional dependencies
b) R_1(A, B) and R_2(B, C), if we know that $A \rightarrow B$
c) R_1(A, B) and R_2(B, C), if we know that $B \rightarrow A$
d) R_1(A, B) and R_2(B, C), if we know that $A \rightarrow C$
e) R_1(A, B) and R_2(B, C), if we know that $C \rightarrow A$

Question 21: If you want to check whether attributes salary1 and salary2 in a relation are both NULL, which way (or ways) can you write a condition in a SQL WHERE clause to test for that?

a) salary1 = NULL AND salary2 = NULL
b) salary1 IS NULL AND salary1 = salary2
c) salary1 IS NULL AND salary2 IS NULL
d) All of the Above
e) None of the Above

Question 22: Suppose that R(A, B ,C, D) is a relation, r is an instance of R, and F is a non-trivial Functional Dependency on the attributes of R. Here are two statements:

i. If r satisfies F, then F must hold for R.
ii. If r does not satisfy F, then F does not hold for R.

You're asked to supply the correct answer about these two statements.

a) Both statements are True.
b) The first statement is True and the second statement is False.
c) The first statement is False and the second statement is True.
d) Both statements are False.
e) All of the Above