CSE 180, Midterm Exam, Winter 2020, Shel Finkelstein

Student Name: ____

Student ID: ____

UCSC Email: ____

Midterm Points

Part	Max Points	Points
I	30	
II	24	
III	20	
IV	30	
Total	104	

Please show your **UCSC ID** when you turn in your exam.

This exam is single sided, and you may write an answer on the blank side of a page, if you need more space. But if you do that, clearly identify question that you've answered.

Note: For questions that ask you to write SQL statement, points might be deducted if you write complicated queries, even if they are correct. You can lose credit if you use DISTINCT but it's not needed, and you will lose credit if you don't use DISTINCT and it is needed.

Part I: (30 points, 6 each):

Question 1: R1, R2 and R3 are relation instances. If R1 has 5 tuples in it, R2 has 4 tuples in it, and R3 has 6 tuples in it. How many tuples are there in the result of the following SQL query?

SELECT *
FROM R1, R2, R3;

Answer 1: __________

Question 2: Let R(A,B,C,D) be a relation, where (A, B, C) is the primary key for R, and attribute D can be NULL. Assume that attribute A's domain has 6 different values, B's domain has 2 different values, C's domain has 3 different values, and D's domain has 5 different values. What is the maximum number of tuples that can be in an instance of R?

Answer 2: __________

Question 3: NULL, UNKNOWN and the Empty Set have different meanings in SQL. Give an example of an expression that could have each of these values.

3a): NULL
Answer 3a):

3b): UNKNOWN
Answer 3b):

3c): Empty Set
Answer 3c):

Question 4: If a tuple appears 20 times in the result for Q1 and that exact same tuple appears 8 times in the result for Q2, then:

4a): How many times would that tuple appear in Q1 INTERSECT Q2?

Answer 4a): __________

4b): How many times would that tuple appear in Q1 INTERSECT ALL Q2?

Answer 4b): __________

4c): How many times would that tuple appear in Q1 UNION ALL Q2?

Answer 4c): __________

Question 5: Your answers to the following questions should provide a brief explanation, not just some buzzwords. Give just one answer for each.

5a): Explain one advantage of views.
Answer 5a):

5b): Explain one advantage of indexes.
Answer 5b):

5c): Explain one disadvantage of indexes.
Answer 5c):

Part II: (24 points, 8 each): The questions in Parts II and IV ask you to write SQL statements using the tables shown below, which are 3 of the tables in our Lab Assignments.

The primary key in each table is shown underlined. Assume that there aren't any NOT NULL or UNIQUE constraints specified for these tables. You may assume the same Referential Integrity constraints that we had in our Lab Assignment tables. Data types aren't shown to keep thing simple. There aren't any trick questions about data types.

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 6: Write a SQL statement outputs the movieID values for Movies whose year is before 2011 and whose length isn't NULL. No duplicates should appear in your result.

Answer 6:

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 7: Write a SQL statement that increases numSeats by 5 for each theater whose address ends with ‘Street’.

Answer 7:

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 8: Write a SQL statement that outputs the movieID and startTime for Showings whose showingDate is January 28, 2019. Your result tuples should appear sorted, with the biggest movieID first and the smallest movieID last. If movieID is the same for two result tuples, then the result tuple that has the earliest startTime should appear first. No duplicates should appear in your result.

Answer 8:

Part III: (20 points, 4 each)

The following **TRUE** or **FALSE** questions refer to the Customers table that is created by the following statement:

```
CREATE TABLE Customers (
        cid             INTEGER,
        name            VARCHAR(20) UNIQUE,
        type            VARCHAR(20),
        level           VARCHAR (20) DEFAULT('Beginner') NOT NULL,
        age             INTEGER,
        PRIMARY KEY (cid),
        CHECK (age >= 18 AND age <= 65)
        );
```

Answer each question TRUE or FALSE. No explanation is required, and no part credit will be given.

Question 9: TRUE or **FALSE:** The following two SQL queries are equivalent.

```
SELECT COUNT(name)                  SELECT COUNT(DISTINCT name)
FROM Customers;                     FROM Customers;
```

Answer 9: __________

Question 10: TRUE or **FALSE:** The following is a legal SQL query.

```
SELECT type, COUNT(*), AVG(age)
FROM Customers
WHERE level <> 'Advanced'
GROUP BY type;
```

Answer 10: __________

Question 11: TRUE or **FALSE**: Suppose that there are no tuples in Customers. Execution of the following statement will result in an error.

```
INSERT INTO Customers(cid, name, age)
        VALUES (36, ‘Cho’, 50);
```

Answer 11: __________

Question 12: TRUE or **FALSE** : The following two queries are equivalent.

```
SELECT C.cid
FROM Customers C
WHERE C.level = ANY ( SELECT C2.level
                      FROM Customers C2
                      WHERE C2.type = 'snowboard' );

SELECT C.cid
FROM Customers C
WHERE C.level IN ( SELECT C2.level
                   FROM Customers C2
                   WHERE C2.type = 'snowboard' );
```

Answer 12: __________

Question 13: TRUE or **FALSE** : The CHECK condition on age in the CREATE statement would be violated if the value of age was NULL, or if it was less than 18, or if it was greater than 65.

Answer 13: __________

Part IV: (30 points, 10 each): The questions in Part IV ask you to write SQL statements using the tables shown below, which are 3 of the tables in our Lab Assignments.

The primary key in each table is shown underlined. Assume that there aren't any NOT NULL or UNIQUE constraints specified for these tables. You may assume the same Referential Integrity constraints that we had in our Lab Assignment tables. Data types aren't shown to keep thing simple. There aren't any trick questions about data types.

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 14: Write a SQL statement that creates a view PoorMovies that finds the movieID and name for movies whose length is longer than 130, and which also have no showings in theaters that have 10 or more seats. No duplicates should appear in your result.

Answer 14:

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 15: totalEarned is an attribute of Movies. Write a SQL statement that for each showingDate finds the minimum totalEarned value for the movies that have Showings on that showingDate. But don't include a result tuple for a showingDate unless the number of Showings on that showingDate is more than 5. The attributes in your result should appear as theDate and minTotalEarned. No duplicates should appear in your result.

Answer 15:

Movies(movieID, name, year, rating, length, totalEarned)
Theaters(theaterID, address, numSeats)
Showings(theaterID, showingDate, startTime, movieID, priceCode)

Question 16: Write a SQL statement that finds the theaterID, showingDate and startTime for each showing that has an earlier startTime than some other showing of the same movie that's on the same showingDate. No duplicates should appear in your result.

Answer 16: