UCLA
Computer Science Department
Fall 2021

Instructor: J. Cho

CS143 Quiz 1: 2 hours

Student Name: ______________________________

Student ID: ______________________________

(IMPORTANT PLEASE READ **):**

- The exam is *closed book* and *closed notes*. You may use *one double-sided cheat-sheets*. You can use a calculator.
- *Simplicity and clarity of your solutions will count.* You may get as few as 0 point for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- If you need to make any assumption to solve a question, *please write down your assumptions*. To get partial credits, you may want to write down how you arrived at your answer step by step.
- If a question asks for a numeric answer, you don't have to calculate. You may just write down a numeric expression.
- Please, write your answers neatly.

Problem	Score	
1	20	
2	20	
3	10	
4	20	
5	30	
Total	100	

Problem 1: Relational Algebra (20 points)

1. Consider the relation

   Transaction(customer, item, price, date).

   The tuple ('John', 'Macbook Air', 999, 2021-10-01) in the relation means that John purchased a Macbook Air at the price of $999 on 10/01/2021.

   Write a relational algebra expression that returns all customers who purchased at least two different items of price $100 or above. You may assume that customer and item names are unique. You may also assume no NULL values in the table.

   Note that relational algebra does **NOT** have an aggregate function and simplicity and clarity of your answer count as well as correctness.

   ANSWER:
   $\pi_{T1.customer}(\sigma_C(\rho_{T1}(Transaction) \times \rho_{T2}(Transaction)))$ where C is T1.customer = T2.customer $\wedge$ T1.item $\neq$ T2.item $\wedge$ T1.price $\geq$ 100 $\wedge$ T2.price $\geq$ 100

2. Consider a relation $R(A, B, C)$. You may assume there are no NULL values or duplicate tuples in R.

   Consider the following relational query:

   $\sigma_{V \neq W}(\rho_{R_1(X,G,V)}(R) \bowtie \rho_{R_2(X,H,W)}(R))$

   Assume the result of the above query is always guaranteed to be empty. Using 30 words or less, state a nontrivial property/constraint/fact on R that is guaranteed to hold if and only if this assumption is true. Keep in mind that there is a very concise correct answer (e.g., "C is a key of R") corresponding to a concept covered in class. (10 points)

   ANSWER:
   There is a functional dependency $A \rightarrow C$. If the result is always empty, there exist no two tuples in R whose A values are the same, but C values are different.

Problem 2: Query Equivalence (20 points)

Two queries are considered equivalent if they return exactly the same results for all database instances.

Assume the following for the relation $R(\underline{A}, B)$ referenced in the queries.

- The attribute A is a key.
- The attribute B is not a key.
- The attribute A may not have NULL, but the attribute B may have NULL.

Be mindful of the above assumptions since they are critical to get the correct answers. Do not make any other assumptions.

For each of the pair of queries listed below, write "YES" if the two queries are equivalent and "NO" otherwise. If your answer is YES, briefly explain your answer. If your answer is NO, write down a simple database instance (e.g., $\{(1, 2), (3, 4)\}$) that returns different answers to the two queries.

Pay special attention to NULL values and duplicates. Note that you may get as low as zero point if your YES/NO answer is correct but your justification/example is not.

1. YES/NO

   1. `SELECT SUM(B)/COUNT(*) FROM R;`
   2. `SELECT AVG(B) FROM R;`

   ANSWER:
   No, consider {(1, NULL), (2, 2)}

2. YES/NO

   1. ```
      SELECT A
      FROM R
      WHERE A >= ALL (SELECT A FROM R);
      ```
   2. `SELECT MAX(A) FROM R;`

   ANSWER:
   No. When R is non-empty, both computes MAX(A) and there are no duplicate A values since it is a key. But when R is empty, Query 1 returns {} while Query 2 returns {NULL}.
      ```

3. YES/NO

   1. ```
      SELECT B
      FROM R R1
      WHERE B <= ALL (SELECT B FROM R R2 WHERE R1.A <> R2.A);
      ```

   2. ```
      SELECT MIN(B) FROM R;
      ```

   ANSWER:
   No, consider {(1, 1), (2, 1)} or {(1, 1), (2, NULL)}

4. YES/NO

   1. ```
      (SELECT B FROM R) EXCEPT (
          SELECT R1.B FROM R R1, R R2
          WHERE R1.B = R2.B AND R1.A <> R2.A);
      ```

   2. ```
      SELECT B FROM R GROUP BY B HAVING COUNT(*) = 1;
      ```

   ANSWER:
   No, consider {(1, NULL), (2, NULL)}

Problem 3: SQL Query (10 points)

Consider the Parent(parent, child) table that has parent-child relationship. For example, the tuple (John, Vanessa) in the table means that John is a parent of Vanessa. Assume that a person's name is unique.

Someone's descendants are the people in later generations who are born from them (directly or indirectly). Write a SQL query that returns the names of the people who have more descendants than Vanessa. Please write your query neatly and concisely. Significant points may get deducted if your solution is far more complicated than necessary, or if we cannot understand your solution.

ANSWER:

```
WITH RECURSIVE Descendant(name, descendant) AS (
    (SELECT * FROM Parent)
    UNION
    (SELECT D.name, P.child
        FROM Descendent D, Parent P
        WHERE D.descendent = P.parent)
) SELECT name
FROM Descendant
GROUP BY name
HAVING COUNT(*) > (SELECT COUNT(*) FROM Descendant WHERE name = 'Vanessa');
```

Here are some other possibilities

```
WITH RECURSIVE Descendant(name, descendant) AS (
    (SELECT * FROM Parent)
    UNION
    (SELECT D.name, P.child
        FROM Descendent D, Parent P
        WHERE D.descendent = P.parent)
) SELECT D2.name
FROM (SELECT name, COUNT(*) count FROM Descendant GROUP BY name) D1,
        (SELECT name, COUNT(*) count FROM Descendant GROUP BY name) D2
WHERE D1.name = 'Vanessa' AND D2.count > D1.count;
```

```
WITH RECURSIVE Descendant(name, descendant) AS (
    (SELECT * FROM Parent)
    UNION
    (SELECT D.name, P.child
        FROM Descendent D, Parent P
        WHERE D.descendent = P.parent)
), DescCount(name, count) AS (
    SELECT name, COUNT(*)
    FROM Descendant
    GROUP BY name
) SELECT D2.name
FROM DescCount D1, DescCount D2
WHERE D1.name = 'Vanessa' AND D2.count > D1.count;
```

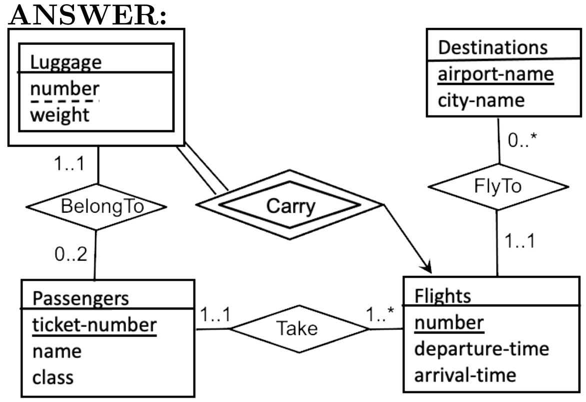
Problem 4: ER Model (20 Points)

Consider the following entity sets and their relationships:

- There are four entity sets: Passengers, Flights, Luggage and Destinations.
- Passengers have the attributes ticket-number, name, and class. Passenger ticket-numbers are unique.
- Flights have the attributes number, departure-time and arrival-time. Flight numbers are unique.
- Luggage have the attributes number and weight. Luggage numbers are numbers assigned to each piece of luggage within its corresponding flight, thus they are unique inside the flight, but can be the same across flights.
- Destinations have the attributes city-name and airport-name. Airport names are unique.
- Every passenger takes exactly one flight, and every flight contains at least one passenger.
- Every passenger carries 0, 1 or 2 pieces of luggage, and every luggage belongs to exactly one passenger and is carried by exactly one flight.
- Every flight has exactly one destination, and multiple flights can have the same destination.

1. Draw an E-R diagram using for the above scenario. In drawing the diagram use *only binary relationship sets*. Specify the cardinality and participation of relationship sets either using arrows and double lines or the general cardinality notation (like 1..3) whenever necessary. Note that some cardinality constraints may not be expressible using arrows. Also indicate whether an entity set is strong or weak, and the key or discriminator of every entity set. You can use any reasonable name for relationship sets and their attributes. (10 points)

ANSWER:



2. Convert the ER diagram into tables and write the schema of all tables. Underline the key of each table, like Movie(<u>mid</u>, title, year). (10 points)

ANSWER:

```
Passengers(_ticket-number_, name, class)
Flights(_number_, departure-time, arrival-time)
Destinations(_airport-name_, city-name)
Luggage(_flight-number_, _number_, weight)
Take(_ticket-number_, flight-number)
BelongTo(_flight-number_, _luggage-number_, ticket-number)
FlyTo(_flight-number_, airport-name)
```

Problem 5: Relational Design Theory (30 points)

1. Consider the relation $R(A, B, C, D)$, on which functional dependencies $A \rightarrow B$, $B \rightarrow C$, and $C \rightarrow D$ hold. If $|R| = 10$, $|\pi_B(R)| = 5$, and $|\pi_D(R)| = 4$, what are the possible value(s) of $|\pi_A(R) \times \pi_C(R)|$? Here, the notation $|R|$ means the number of tuples in R. Note that we are dealing with relational algebra operators that are based on the set semantics. (10 points)

   ANSWER:
   40, 50. Since A is the key, $|\pi_A(R)| = |R| = 10$. Because of the functional dependency, $|\pi_C(R)|$ can be either 4 or 5.

2. Suppose that we decompose the table $R(A, B, C, D, E)$ into $R1(A, B, C)$ and $R2(A, D, E)$. Assume the following functional dependencies hold on R.

   $A \rightarrow B$, $CD \rightarrow E$, $E \rightarrow A$.

   Is the decomposition lossless? YES/NO

   If your answer is YES, please explain why. If your answer is NO, please provide a simplest functional dependency (that involves the least number of attributes) whose addition will make the decomposition lossless. (10 points)

   ANSWER:
   NO. $A \rightarrow C$ or $B \rightarrow C$.

3. Consider a relation $R(A, B, C, D, E)$ and a set of functional dependencies:

   $A \to B, \quad CD \to E, \quad E \to D.$

   Is the table in BCNF? If yes, explain why the functional dependencies do not violate the BCNF definition. If no, normalize it into BCNF. If you normalize the table, underline the key in the normalized tables. (10 points)

   ANSWER:
   No. R1($\underline{\text{A}}$,B), R2($\underline{\text{A}}$, $\underline{\text{C}}$, $\underline{\text{D}}$), R3($\underline{\text{C}}$,$\underline{\text{E}}$), R4(D,$\underline{\text{E}}$)