ONID:

CS540 Database Management Systems

Winter 2021

School of Electrical Engineering & Computer Science

Oregon State University

Midterm Examination

Time Limit: 80 minutes

- The exam is open book and notes.
- Any form of cheating on the examination will result in a zero grade.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Simplicity does count!*

Question:	1	2	3	Total
Points:	10	5	2	17
Score:				

1. **Relational Languages: SQL & Datalog**
   Consider the following schema:

   Coffee(<u>cbrand</u>, producer)
   CoffeeShop(<u>sname</u>, addr)
   Sells(<u>sname</u>,<u>cbrand</u>, price)

   Attributes *cbrand* and *producer* in relation *Coffee* are names and producers of coffee brands, respectively. Attributes *sname* and *addr* in relation *CoffeeShop* contain the names and addresses of coffee shops. The relation *Sells* stores the price at which coffee shops sell different brands of coffee. The underlined attributes are the keys for their relations.

   (a) (1 point) Write a SQL query that returns each producer that makes the most expensive brand(s) of coffee, i.e., coffee brand(s) sold at the highest price.

   Solution:

```
SELECT C.producer
FROM Coffee C, Sells S
WHERE C.cbrand = S.cbrand and S.price >= ALL
( Select price
  FROM Sells
)
```

   (b) (2 points) Write a SQL query that returns the address of each coffee shop that sells every brand in the *Coffee* relation.

   Solution:

```
SELECT C.addr
FROM CoffeeShop C, Sells S
WHERE C.sname = S.sname
Group By sname
Having count(cbrand) =
(SELECT count(cbrand) FROM Coffee)
```

(c) (2 points) Write a SQL query that returns the addresses of every pair of coffee shops that sell the same set of coffee brands.

Solution:

```
SELECT C1.addr, C2.addr
FROM CoffeeShop C1, CoffeeShop C2
WHERE NOT EXISTS
 ((SELECT cbrand
  FROM C1, Sells S1
  WHERE C1.sname = S1.sname)
  EXCEPT
  (SELECT cbrand
  FROM C2, Sells S2
  WHERE C2.sname = S2.sname))
```

Alternatively,

```
SELECT C1.addr, C2.addr
FROM CoffeeShop C1, CoffeeShop C2
WHERE NOT EXISTS
 (SELECT *
 FROM C1, Sells S1
 WHERE C1.sname = S1.sname AND
     cbrand NOT IN
       (SELECT cbrand
       FROM C2, Sells S2
       WHERE C2.sname = S2.sname))
```

(d) (2 points) Write a SQL query that returns the address of every coffee shop that sells only the brand 'Coava'. 'Coava' is a brand of coffee.

Solution:

```
SELECT C1.addr
FROM CoffeeShop C1, Sells S1
WHERE C1.sname = S1.sname AND S1.cbrand = 'Coava'
AND NOT EXISTS
( SELECT *
  FROM Sells S2
  WHERE C1.sname = S2.sname AND S2.cbrand <> 'Coava');
```

(e) (1 point) Consider the following SQL query.

```
SELECT C.cbrand
FROM Coffee C, Sells S
WHERE C.cbrand = S.cbrand
Group By C.cbrand
Having count(producer) > 10
```

Rewrite the above query without using the Group By and Having clauses so that the resulting query still produces the same result.

Solution:

```
SELECT C.cbrand
FROM Coffee C
WHERE 10 <
(SELECT count(producer)
 FROM C, Sells S
 WHERE C.cbrand = S.cbrand
)
```

(f) (2 points) Write a Datalog query that returns the address of every coffee shop that sells only the brand 'Coava'. 'Coava' is a brand of coffee.

Solution:

```
Q1(x) :- Sells(x, 'Coava', z)
Q2(x) :- Sells(x, y, z), not Q1(x)
Q(y) :- CoffeeShop(x, y), Q1(x), not Q2(x)
```

2. **Schema Normalization: BCNF & 3NF**
Consider relation $R(A, B, C, D)$ with functional dependencies $A \rightarrow B$ and $C \rightarrow D$.

(a) (3 points) Is relation R in BCNF? If it is not, convert it to a schema that is in BCNF.

Solution: The closure of the set of input FD is equal to the initial set of FDs. Thus, the key is A, C. It is not as the right hand side of $A \rightarrow B$ is not a key. Starting with $A \rightarrow B$ and applying the BCNF decomposition algorithm, we have $R_1(A, B), R_2(A, C, D)$. Using $C \rightarrow D$, we convert R_2 to $R_3(C, D)$ and $R_4(A, C)$.

(b) (2 points) Is relation R in 3NF? If it is not, convert it to a schema that is in 3NF.
Solution: It is not as the right hand side of $A \rightarrow B$ is not a key or B is not part of a key. The minimum basis for the original set of FDs is $A \rightarrow B$, $C \rightarrow D$. The 3NF synthesis of R will be $R_1(A, B)$, $R_2(C, D)$, $R_3(A, C)$

3. **Schema Normalization: Lossless Decomposition & Dependency Preservation** Consider relation $R(A, B, C, D, E, F)$ with the following functional dependencies: $A \rightarrow CDF$, $D \rightarrow A$ and $DF \rightarrow E$. We decompose R into two relations, $S(A, B, C, D)$ and $T(D, E, F)$.

   (a) (1 point) Is this decomposition a lossless decomposition? Briefly justify your answer.
   Solution: Since D is a key of T, the decomposition is a losless one.

   (b) (1 point) Is this decomposition a dependency-preserving decomposition? Briefly justify your answer.
   Solution: It is not as it does not preserve $A \rightarrow CDF$, i.e., A and D are in different relations in the decomposition.