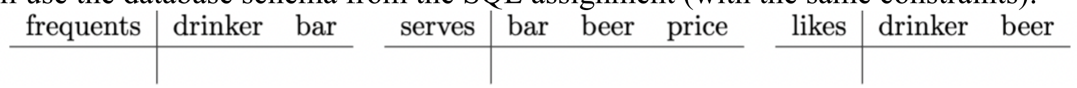
CSE 132A **Homework# 1** **Winter 2023**

Due February 2, 11:59pm PST (see instructions below)

This is an **individual** assignment.

Xiangyi Lin

We will use the database schema from the SQL assignment (with the same constraints):



frequents	drinker	bar

serves	bar	beer	price

likes	drinker	beer

Consider the following queries (the second is Q6 from the SQL assignment):

1. List the drinkers who like every beer served at 'The Swan'. The answer should have one attribute drinker. Note that, if 'The Swan' does not serve any beer, then all drinkers should be in the answer.
   (i) write the query in relational calculus using (at least one) universal quantification ∀

   $\{d: \text{drinker} \mid \exists a \in \text{likes}\ [d(\text{drinker})=a(\text{drinker}) \land \forall s \in \text{serves}\ [s(\text{bar})= \text{“The Swan”} \rightarrow \exists b \in \text{likes}\ (b(\text{drinker})= a(\text{drinker}) \land b(\text{beer})=s(\text{beer}))]]\}$

(ii) rewrite the query in (i) in using only existential quantification $\exists$

$$\{d: \text{drinker} \mid \exists a \in \text{likes}\ [d(\text{drinker})=a(\text{drinker}) \land \neg\exists s \in \text{serves}\ [s(\text{bar})= \text{“The Swan”} \land \neg\exists b \in \text{likes}\ (b(\text{drinker})= a(\text{drinker}) \land b(\text{beer})=s(\text{beer}))]]\}$$

(iii) write the SQL query corresponding directly to the relational calculus query in (ii), that uses only NOT EXISTS tests on nested queries.

```
SELECT DISTINCT a.drinker
FROM likes a
WHERE
NOT EXISTS
        (SELECT *
        FROM serves s
        WHERE s.bar=' The Swan ' AND
        NOT EXISTS
                (SELECT *
                FROM likes b
                WHERE b.drinker=a.drinker AND b.beer=s.beer));
```

2. Find the bars that serve every beer Joe likes at the lowest price. The answer should have a single attribute bar.

   (i) write the query in relational calculus using (at least one) universal quantification $\forall$

   $$\{x: \text{bar} \mid \exists a \in \text{serves}\ [x(\text{bar})=a(\text{bar}) \wedge \forall s \in \text{likes}\ [s(\text{drinker})= \text{“Joe”} \rightarrow \exists b \in \text{serves}\ [b(\text{bar})= a(\text{bar}) \wedge b(\text{beer})=s(\text{beer}) \wedge \neg\exists c \in \text{serves}(c(\text{beer})=b(\text{beer}) \wedge c(\text{price})<b(\text{price}))]]]\}$$

(ii) rewrite the query in (i) in using only existential quantification ∃

$$\{x: \text{bar} \mid \exists a \in \text{serves}\ [x(\text{bar})=a(\text{bar}) \wedge \neg\exists s \in \text{likes}\ [s(\text{drinker})= \text{“Joe”} \wedge \neg\exists b \in \text{serves}\ [b(\text{bar})= a(\text{bar}) \wedge b(\text{beer})=s(\text{beer}) \wedge \neg\exists c \in \text{serves}(c(\text{beer})=b(\text{beer}) \text{ AND } c(\text{price})<b(\text{price}))]]]\}$$

(iii) write the SQL query corresponding directly to the relational calculus query in (ii), that uses only NOT EXISTS tests on nested queries.

```
SELECT a.bar
FROM serves a
WHERE
NOT EXISTS
     (SELECT *
      FROM likes s
      WHERE s.drinker='Joe' AND
      NOT EXISTS
           (SELECT *
           FROM serves b
           WHERE b.bar=a.bar AND b.beer=s.beer AND
           NOT EXISTS
                 (SELECT *
                 FROM serves c
                 WHERE c.beer=b.beer AND c.price<b.price)));
```