**Quiz 1 (30 minutes) [LIVINGSTON]**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ RUID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

QUERIES – write SQL (ask first, is this monotonic query?)

1. Find all drinkers who like Corona and Bud

SELECT l1.drinker

FROM likes l1, likes l2

WHERE l1.drinker = l2.drinker AND l1.beer = 'Corona' AND l2.beer = 'Bud';

Or

SELECT l1.drinker

FROM likes l1

WHERE l1.beer = 'Corona'

AND l1.drinker IN (SELECT l2.drinker FROM likes l2 WHERE l2.beer = 'Bud');

2. Find all bars which sell Corona but do not sell Bud

SELECT s.bar

FROM sells s

WHERE s.beer = 'Corona'

AND s.bar NOT IN (SELECT s1.bar FROM sells s1 WHERE s1.beer = 'Bud');

3. Find beers which are sold only by Old Tavern and by Cabana (both)

SELECT s1.beer

FROM sells s1, sells s2

WHERE s1.beer = s2.beer AND s1.bar = 'Old Tavern' AND s2.bar = 'Cabana'

AND s1.beer NOT IN (SELECT s3.beer FROM sells s3

WHERE s3.bar <> 'Old Tavern' AND s3.bar <> 'Cabana');

Or

SELECT s.beer

FROM sells s

WHERE s.bar = 'Old Tavern' OR s.bar = 'Cabana'

GROUP BY s.beer

HAVING COUNT(\*) = 2

AND s.beer NOT IN (SELECT s2.beer FROM sells s2

WHERE s2.bar <> 'Old Tavern' AND s2.bar <> 'Cabana');

4. Find drinkers who like all beers served by Cabana

SELECT d.name

From drinkers d

WHERE NOT EXISTS

(SELECT \* FROM sells s

WHERE s.bar= 'Cabana'

AND NOT EXISTS (SELECT \* FROM likes l

WHERE d.name = l.drinker AND s.beer = l.beer));

Or

SELECT l.drinker

FROM likes l, sells s

WHERE l.beer = s.beer AND s.bar = 'Cabana'

GROUP BY l.drinker

HAVING COUNT (\*) = (SELECT COUNT (\*) FROM sells s2 WHERE s2.bar = 'Cabana');

5. Assume database scheme R[Bar, Beer, Price, Day\_of\_the\_Week, Sold], where Sold is number of bottles of beer sold on a given day by a given bar.

Find bars which sell at least 100 bottles of beer (all) every Monday

**ANSWER**

(Since there is no DATE attribute, there will only be 7 days of the week, no possibility to distinguish between different Mondays)

SELECT Bar

FROM R

WHERE Day\_of\_the\_Week = 'Monday'

GROUP BY Bar

HAVING SUM (Sold) > 100

6. What does this SQL query return (write it in English)

Select distinct f.drinker

From frequents f

Where f.drinker NOT IN (

Select f2.drinker

From frequents f2

Where f2.bar NOT IN

(Select bar

From sells, likes

Where sells.beer = likes.beer AND likes.drinker = f.drinker));

**ANSWER**: Drinkers who only frequent bars which sell some beer they like (the drinker must frequent at least one bar).

7\*. What does this SQL query return (write it in English)

(\*This will not count but some extra credit will be given since I have missed one join condition here)

SELECT d.name

FROM drinkers d

WHERE NOT EXISTS

(Select s.bar

From likes l, sells s, frequents f

Where (l.beer = s.beer) and (l.drinker=d.name) and (f.drinker=l.drinker))

**ANSWER**: It would be “Drinkers who do not frequent any bar which serve a beer they like” if there was condition on bar (f.bar=s.bar).

8. What does this mysql query return for the database instance R of db scheme

B = [Beer, Manufacturer, Price, Alcohol]

R = {<Bud, Anheuser-Busch, NULL, 4.5>}

SELECT B.Beer

FROM B

WHERE (Alcohol < 6 AND Price > 5)

OR (Manufacturer = 'Anheuser-Busch' AND Price <= 5)

**ANSWER**: The result will be empty since due to the Price being NULL, mysql will evaluate both disjoints to MAYBE, consequently WHERE clause is not going to be true.

But this is incorrect since no matter what NULL is the WHERE condition will always be TRUE.