

ASSIGNMENT 4

QUESTION 1

e) $A(a,b,c) \leftarrow (R(a,b,c) \text{ AND NOT } S(a,b,c)) \text{ AND } (R(a,b,c) \text{ AND NOT } T(a,b,c))$

f) $A(a,b) \leftarrow R(a,b,c)$

QUESTION 2

b) $R \bowtie_{x < y \text{ AND } y < z} S$

$I(x,y,z) \leftarrow R(x,y,z) \text{ AND } (x < y \text{ AND } z < y)$

$J(x,y,z) \leftarrow S(x,y,z) \text{ AND } (x < y \text{ AND } z < y)$

$A(xr,xs,yr,ys,zr,zs) \leftarrow I(xr,yr,zr) \text{ AND } J(xs,ys,zs)$

Or you can say

$A(xr,xs,yr,ys,zr,zs) \leftarrow R(xr,yr,zr) \text{ AND } S(xs,ys,zs) \text{ AND } (xr < yr \text{ AND } xs < ys) \text{ AND } (yr < zr \text{ AND } ys < zs)$

c) $R \bowtie_{x < y \text{ OR } y < z} S$

$A(xr,xs,yr,ys,zr,zs) \leftarrow R(xr,yr,zr) \text{ AND } S(xs,ys,zs) \text{ AND } ((xr < yr \text{ AND } xs < ys) \text{ OR } (yr < zr \text{ AND } ys < zs))$

QUESTION 3

d) Find the model number of item (PC, laptop, or printer) with the highest price.

First using ALL

```
SELECT model
FROM (
    (SELECT model, price FROM PC) UNION (SELECT model, price FROM
    Laptop) UNION (SELECT model, price FROM Printer)) AS AllPricesDevices
WHERE price >= ALL (
    SELECT price FROM AllPricesDevices
)
```

Second using IN and MAX

```
SELECT model
FROM (
    (SELECT model, price FROM PC WHERE (S) UNION (SELECT model, price
    FROM Laptop) UNION (SELECT model, price FROM Printer)) AS AllPricesDevices
)
WHERE price IN (
    SELECT MAX(AllDevicesMax.price)
    FROM (
        (SELECT model, price FROM PC) UNION (SELECT model, price FROM
        Laptop) UNION (SELECT model, price FROM Printer)) AS
        AllDevicesMax
    )
```

Noemi Lemonnier

#40001085

e) Find the marker of the color printer with the lowest price.

First using ALL and Distinct

```
SELECT DISTINCT marker
FROM Product, Printer
WHERE Product.model = Printer.model AND color = 'true' AND price <= ALL (
    SELECT price FROM Printer WHERE color = 'true')
```

Second using IN and MIN

```
SELECT DISTINCT marker
FROM Product, Printer
WHERE Product.model = Printer.model AND color = 'true' AND price IN(
    SELECT MIN(price) FROM Printer WHERE color = 'true')
```

QUESTION 4

b) Without using the Difference query of Example 6.17

```
SELECT name, address
FROM MovieStar, MovieExec
WHERE MovieStar.name NOT IN(SELECT name FROM MovieExec) AND
MovieStar.address NOT IN(SELECT address FROM MovieExec)
```

QUESTION 5

d)

```
DELETE FROM Laptop
WHERE model IN(
    SELECT Product.model
        FROM Product, Laptop
        WHERE Product.model = Laptop.model AND maker IN(
            (SELECT DISTINCT maker FROM Product)
            EXCEPT ALL (SELECT DISTINCT maker FROM Product
                WHERE type = 'printer')))
```

g)

```
UPDATE Laptop
SET price = price-100 AND screen = screen+1
WHERE model IN(
    SELECT model
        FROM Product, Laptop
        WHERE Product.marker = 'B' AND Product.model = Laptop.model)
```

QUESTION 6

d)

```
SELECT name
FROM Studio
WHERE name IN(
    SELECT studioName
```

```
FROM Movies
WHERE studioName = Studio.name
GROUP BY studioName
HAVING COUNT(*) > 1
)
```

QUESTION 7

a) No it is not updatable because it does not respect the rules indicated in the book.

From the Book page 345

SQL provides a formal definition of when modifications to a view are permitted. The SQL rules are complex, but roughly, they permit modifications on views that are defined by selecting (using **SELECT**, not **SELECT DISTINCT**) some attributes from one relation R (which may itself be an updatable view). Two important technical points:

- The **WHERE** clause must not involve R in a subquery.
- The **FROM** clause can only consist of one occurrence of R and no other relation.
- The list in the **SELECT** clause must include enough attributes that for every tuple inserted into the view, we can fill the other attributes out with **NULL** values or the proper default. For example, it is not permitted to project out an attribute that is declared **NOT NULL** and has no default.

An insertion on the view can be applied directly to the underlying relation R . The only nuance is that we need to specify that the attributes in the **SELECT** clause of the view are the only ones for which values are supplied.

b) Similar format from slide the course

```
CREATE TRIGGER insertPC
AFTER INSERT ON NewPC
REFERENCING
    OLD ROW AS oldTuple,
    NEW ROW AS newTuple
FOR EACH ROW
    INSERT INTO Product VALUES(newTuple.marker, newTuple.model,
    newTuple.type)
    INSERT INTO PC VALUES(newTuple.model, newTuple.speed,
    newTuple.ram, newTuple.hd, newTuple.price)
END
```