Assignment 2: Relational Algebra and SQL

Due: Oct 28 (Sat), 11:59pm

Mark = 70 marks. Weight=10%

Submission instruction: (1) This assignment must be done by each student independently. (2) Handwriting is acceptable but the student is responsible for deduction of marks due to clarity issue. (3) Submission is through coursys.sfu.ca in a single pdf file with maximum file size of 10MB. Late submission will not be accepted. (4) The student is responsible for submitting the assignment successfully before the deadline (for example, by incremental submission multiple times well before the deadline, instead of one submission at the last minute). We only accept the submission that is in the system before the deadline.

This assignment will use two databases, "Department-Store Database" and "Drinker-Beer Database". Underlined fields are the attributes of primary keys. The records in the tables only serve as examples. The tables may contain more records.

Department Store Database

Employee relation:

eid	name	salary	dept
111	Jane	8000	Household
222	Anderson	8000	Toy
333	Morgan	10000	Cosmetics
444	Lewis	12000	Stationery
555	Nelson	6000	Toy
666	Hoffman	16000	Cosmetics

Sales relation:

dept	item
Stationery	pen
Cosmetics	lipstick
Toy	puzzle
Stationery	ink
Household	disk
Sports	skates
Toy	lipstick

Types relation:

item	color
pen	red
lipstick	red
pen	black
puzzle	black
ink	red
ink	blue

Drinker-beer Database

Frequents relation:

drinker	bar
Ullman	Manuel's
Ullman	Orchard Night
Ullman	Faculty Clue
Ullman	Dynasty
Graham	Dynasty
Sam	Manul's
Sam	Orchard Night
Smith	Dynasty

Serves relation:

bar	beer
Manuel's	Miller Lite
Manuel's	Tiger
Orchard Night	Busch
Manuel's	Qindao
Faculty Club	Tiger
Faculty Club	Miller Lite
Dynasty	Anchor

Likes relation:

drinker	beer
Ullman	Miller Lite
Ullman	Tiger
Ullman	Anchor
Jane	Anchor
Sam	Anchor

Question 1 (25 marks, 5 marks each). Express the following queries in Relational Algebra. Both the correctness and the simplicity count.

1. List items available in both "red" and "blue".

   $\prod_{item}(\sigma_{color=\text{“}red\text{”}}(Types)) \cap \prod_{item}(\sigma_{color=\text{“}blue\text{”}}(Types))$
   Wrong: $\prod_{item}(\sigma_{color=\text{“}red\text{”} \wedge color=\text{“}blue\text{”}}(Types))$

2. List the name of the employees making at least as much as "Jane". If there are several employees named "Jane", which Jane's salary is used in this comparison in your answer?

   $\rho(S, \prod_{salary}(\sigma_{name=\text{“}Jane\text{”}}(Employee)))$
   $\prod_{name}(\sigma_{Employee.salary \geq S.salary}(Employee \times S))$

   Note that the selection σ enforces "at least as much as Jane's salary". In case of multiple Janes, an employee is returned as long as his/her salary is at least as much as the least salary of these Janes.

3. Find the largest salary paid to any employees.

   $\rho(S, \prod_{salary}(Employee))$
   $\prod_{salary}(Employee) - \prod_{Employee.salary}(\sigma_{S.salary > Employee.salary}(S \times Employee))$

   The selection operator σ selects those employees whose salary is less than some employee, thus, is not the highest.

4. What departments sell every item with a red color.

   $Sales / \prod_{item}(\sigma_{color=\text{“}red\text{”}}(Types))$

   Note that the denominator contains all items with a red color. Then the division operator / returns every department that sells all such items.

5. What departments sell only items with only red color, in other words, what departments do not sell any item with a non-red color.

   $\prod_{dept}(Sales) - \prod_{dept}(\sigma_{color \neq \text{“}red\text{”}}(Sales \bowtie Types))$

   The first term returns all departments, the second term returns the departments that sell at least some item with a non-red color. Note that the natural join enforces the equality on item in both operands.

Question 2 (25 marks, 5 marks each). Express the queries in Question 1 in SQL.

1. List items available in both "red" and "blue".

   ```
   SELECT item
   FROM Types
   WHERE color="red"
   INTERSECT
   SELECT item
   FROM Types
   WHERE color="blue"
   ```

2. List the name of the employees making at least as much as "Jane". If there are several employees named "Jane", which Jane's salary is used in this comparison in your answer?

```
SELECT E1.name
FROM Employee E1, Employee E2
WHERE E1.salary ≥ E2.salary AND E2.name = "Jane"
```

3. Find the largest salary paid to any employees.

```
SELECT salary
FROM Employee
EXCEPT
SELECT E2.salary
FROM Employee E1, Employee E2
WHERE E1.salary > E2.salary
```

4. What departments sell every item with a red color.

```
SELECT S.dept
FROM Sales S
WHERE NOT EXISTS (
    SELECT item
    FROM Types
    WHERE Types.color="red"
    EXCEPT
    SELECT item
    FROM Sales
    WHERE S.dept=Sales.dept)
```

For each S.dept, the subquery within NOT EXIST() computes the red items not sold by S.dept. Then NOT EXIST() holds if the subquery returns empty result. This shows that S.dept sells every red item. See details on slide 28.

5. What departments sell only items with only red color, in other words, what departments do not sell any item with a non-red color.

```
SELECT dept
FROM Sales
EXCEPT
SELECT dept
FROM Sales S, Types T
WHERE S.item = T.item AND T.color ≠ "red"
```

Question 3 (20 marks, 5 marks each) This question refers to the queries in Question 1.

1. Express query 1 in SQL without using INTERSECT

```
SELECT T.item
FROM Types T
WHERE T.color="red" AND EXISTS (
    SELECT T2.item
    FROM Types T2
    WHERE T2.color="blue" AND T.item=T2.item)
```

2. Express query 2 in SQL using nested query

```
SELECT name
FROM Employee
WHERE salary ≥ ANY (
    SELECT E2.salary
    FROM Employee E2
    WHERE E2.name = "Jane")
```

Note that If ANY is replaced with ALL, the query returns the employees whose salary is ≥ the highest salary of any employee named Jane, instead of the lowest salary of any employee named Jane.

3. Express query 3 without using EXCEPT

```
SELECT salary
FROM Employee
WHERE salary ≥ ALL (
    SELECT E2.salary
    FROM Employee E2)
```

Note that ≥ ANY does not work. Alternatively,

```
SELECT Max(salary)
FROM Employee
```

4. Express query 5 without using EXCEPT

```
SELECT S.dept
FROM Sales S
WHERE NOT EXISTS (
    SELECT S2.item
    FROM Sales S2, Types T
    WHERE S.dept = S2.dept AND S2.item = T.item AND T.color ≠ "red")
```