



Assignment 2

Adit (hjk708), Nikolaj (bxz911), Naomi (xng137)

DIS

11. februar 2025

Exercise 1

Consider the following three abstract relations, with the following schema:

$$P(x : \text{int}, y : \text{int}, z : \text{int}) \quad Q(x : \text{int}, y : \text{int}) \quad R(x : \text{int}, y : \text{int})$$

1.

For each of the next relational calculus formulas, give an equivalent relational algebra expression:

a)

$$P(x, y, z) \wedge Q(x, y)$$

$$P \bowtie_{x,y} Q$$

b)

$$Q(x, y) \wedge R(y, z)$$

$$Q \bowtie_y R$$

c)

$$P(x, y, z) \vee P(z, x, y)$$

$$P \cup P$$

d)

$$P(x, y, z) \wedge (\exists x. Q(x, z))$$

$$P \bowtie_z Q$$

e)

$$Q(x, y) \wedge \neg R(x, y)$$

$$Q - R$$

f)

$$P(x, y, z) \wedge \neg R(x, y)$$

$$\pi_{x,y} P - R$$

g)

$$Q(x, y) \wedge \neg R(x, x)$$

$$\pi_x Q - R$$

h)

$$(\exists x. \exists y. (P(x, y, z) \wedge Q(x, z))) \wedge R(x, y)$$

$$(\pi_z (P \bowtie_{x,z} Q)) \times R$$

2.

For each of the next relational algebra expressions, give an equivalent relational calculus formula. If needed, introduce new variable names in formulas to disambiguate different columns that have the same name in the result of the relational algebra expressions.

a)

$$Q \times R$$

$$Q(x_1, y_1) \wedge R(x_2, y_2)$$

b)

$$\pi_x(\sigma_{x=y \vee x=z}(P))$$

$$\exists y. \exists z. P(x, y, z) \wedge x = y \vee x = z$$

c)

$$Q \bowtie \rho_{S(y,z)}(\sigma_{x=y}(R))$$

$$Q(x, y) \wedge (S(x, y) = (R(x, y) \wedge x = y))$$

d)

$$P \cap (\pi_{x,y,x \mapsto s}(Q \cup R))$$

$$P(x, y, z) \wedge (Q(x, y) \vee R(x, y) \wedge \exists y. (Q(s, y) \vee R(s, y)))$$

Exercise 2

Consider the familiar database schema of a computer shop that consists of four relations:

```
Product(maker:string, model:int, type:string)
PC(model:int, speed:float, ram:int, hd:int, price:int)
Laptop(model:int, speed:float, ram:int, hd:int, screen:float, price:int)
Printer(model:int, color:bool, type:string, price:int)
```

Formulate (extended) relational algebra expressions to answer the following queries. Use the relational algebra calculator (<https://dbis-uibk.github.io/relax/calc/local/uibk/local/> 3) to evaluate your expressions. Draw the expression trees for the queries e) and i).

a)

Find the overpriced PCs. Those are PCs that are more expensive and at the same time worse than some other PC in at least two out of three categories (speed, ram, hd; bigger is better).

$$\begin{aligned} & \pi_{PC.model}(\sigma(PC.price > PC1.price \wedge \\ & (\\ & ((PC.speed < PC1.speed) \wedge (PC.ram < PC1.ram)) \vee \\ & ((PC.speed < PC1.speed) \wedge (PC.hd < PC1.hd)) \vee \\ & ((PC.ram < PC1.ram) \wedge (PC.hd < PC1.hd)) \\ &)) \\ & (\rho_{PC1}(PC) \times PC)) \end{aligned}$$

b)

Find those PCs that are faster (in terms of speed) than any laptop.

$\pi_{PC.model}(\sigma(PC.speed > Laptop.speed)(PC \times Laptop))$

c)

Find the makers, models, and prices of all offered devices.

$\pi_{maker, model, price}(Product \bowtie PC) \cup$
 $\pi_{maker, model, price}(Product \bowtie Laptop) \cup$
 $\pi_{maker, model, price}(\rho_{typeP \leftarrow type(Product)} \bowtie Printer)$

d)

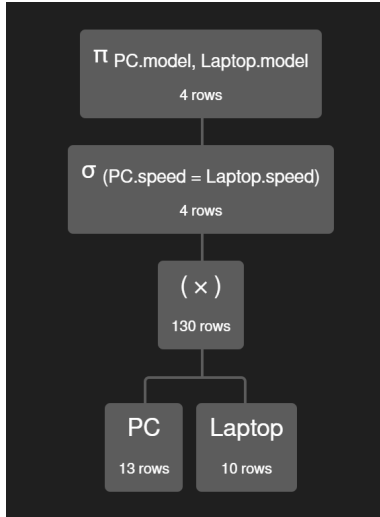
Find the most expensive device manufactured by each maker.

$\gamma_{maker; MAX(price) \rightarrow maxPrice}(($
 $\pi_{maker, model, price}(Product \bowtie PC) \cup$
 $\pi_{maker, model, price}(Product \bowtie Laptop) \cup$
 $\pi_{maker, model, price}(\rho_{typeP \leftarrow type(Product)} \bowtie Printer)))$

e)

Find PC–laptop pairs that are equally fast.

$\pi_{PC.model, Laptop.model}(\sigma(PC.speed = Laptop.speed)(PC \times Laptop))$



f)

Find laptops that cost more than the average laptop price.

$\pi_{Laptop.model}(\sigma(Laptop.price > avp)$
 $(\gamma_{AVG(Laptop.price) \rightarrow avp}(Laptop) \times Laptop))$

g)

For every hard disk size (present in the PC table) compute the number of PCs of that size and their average price. Sort the result by the hard disk size (in ascending order).

```
 $\tau_{hd \text{ asc}} \gamma_{hd}; COUNT(hd) \rightarrow amount(PC) \bowtie$   

 $\gamma_{hd}; AVG(price) \rightarrow avgP(PC)$ 
```

h)

For each printer type, find the most and the least expensive model.

```
 $\tau_{newP \text{ desc}} ($   

 $\pi_{model, typeP, newP} \sigma (price = newP) ($   

 $($   

 $\gamma_{type}; MAX(price) \rightarrow newP (Printer) \cup$   

 $\gamma_{type}; MIN(price) \rightarrow newP (Printer)$   

 $)$   

 $\times \pi_{model, typeP, price} (\rho_{typeP \leftarrow type(Printer)}))$ 
```

i)

The body mass index (BMI) of a computer is its hard disk size divided by the square of its speed. Compute the BMI for each computer (PC and Laptop).

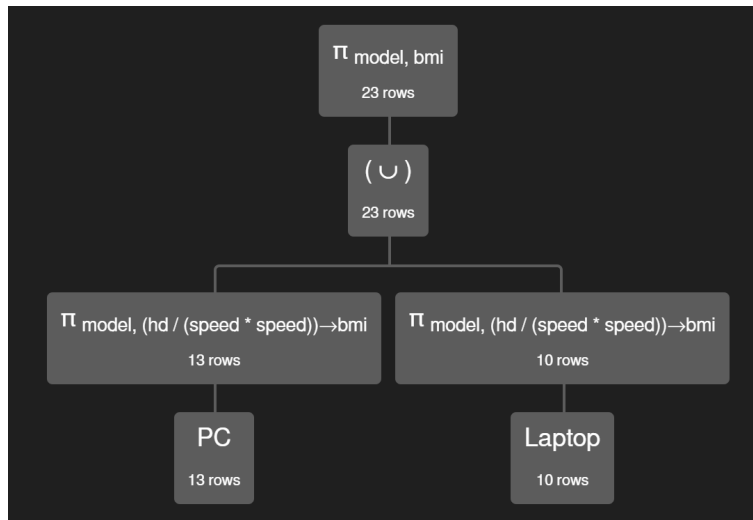
```
 $\pi_{model, bmi}$   

 $($   

 $\pi_{model, (hd / (speed * speed)) \rightarrow bmi} (PC) \cup$   

 $\pi_{model, (hd / (speed * speed)) \rightarrow bmi} (Laptop)$   

 $)$ 
```



Exercise 3

We continue the exploration of our favorite computer shop now using SQL. Write (and evaluate) SQL queries to answer the following queries. To evaluate your queries, download the schema and the example database from Absalon (file product.p.sql) and import it into PostgreSQL, e.g., by executing `psql < product.p.sql`.

a)

Find the model number, speed, and hard-disk size for all PCs whose price is under \$1000 where your rename the speed column to GHz and the hd column to GB.

```
SELECT PC.model, PC.speed AS GHz, PC.hd AS GB
FROM PC
WHERE PC.price < 1000;
```

b)

Find the manufacturers of printers.

```
SELECT DISTINCT Product.make
FROM Product
WHERE Product.type = 'printer';
```

c)

Find the manufacturer and speed of laptops with a hard disk of at least thirty gigabytes

```
SELECT DISTINCT Product.make, Laptop.speed
FROM Product
JOIN Laptop ON Product.model = Laptop.model
WHERE Laptop.hd >= 30;
```

d)

Find the model number and price of all products (of any type) made by manufacturer B.

```
SELECT Product.model, COALESCE(PC.price, Laptop.price, Printer.price) AS price
FROM Product
LEFT JOIN PC ON Product.model = PC.model
LEFT JOIN Laptop ON Product.model = Laptop.model
LEFT JOIN Printer ON Product.model = Printer.model
WHERE Product.make = 'B';
```

e)

Find those manufacturers that sell Laptops, but not PCs.

```
SELECT DISTINCT Product.make
FROM Product
WHERE Product.type = 'laptop' AND Product.make NOT IN (
    SELECT Product.make
    FROM Product
    WHERE Product.type = 'pc'
);
```

f)

Find the laptops whose speed is slower than that of any PC.

```
SELECT Laptop.model
FROM Laptop
WHERE Laptop.speed < (
    SELECT MIN(PC.speed)
    FROM PC
);
```

g)

Find the maker of the color printer with the lowest price.

```
SELECT Product.maker
FROM Product, Printer
WHERE Printer.model = Product.model
      AND price = (SELECT MIN(price)
                   FROM Printer
                   WHERE color = 'true'
                  );
```

h)

Find the model numbers, and the price of the computers (PC or laptop) with the highest speed.

```
SELECT Product.model, COALESCE(PC.price, Laptop.price) AS price
FROM Product
LEFT JOIN PC ON Product.model = PC.model
LEFT JOIN Laptop ON Product.model = Laptop.model
WHERE COALESCE(PC.speed, Laptop.speed) = (
    SELECT MAX(speed)
    FROM (
        SELECT speed FROM PC
        UNION
        SELECT speed FROM Laptop
    ) AS computer_speeds
);
```

i)

Find for each manufacturer, the average screen size of its laptops.

```
SELECT Product.maker, AVG(Laptop.screen) AS average_screen
FROM Product
JOIN Laptop ON Product.model = Laptop.model
WHERE Product.type = 'laptop'
GROUP BY Product.maker;
```

j)

Find the manufacturers that make at least three different models of PC.

```
SELECT DISTINCT Product.maker
FROM Product
JOIN PC ON Product.model = PC.model
WHERE Product.type = 'pc'
GROUP BY Product.maker
HAVING COUNT(PC.model) >= 3;
```

k)

Find for each manufacturer who sells PCs the maximum price of a PC.

```
SELECT Product.maker, MAX(PC.price) AS maxPrice
FROM Product
JOIN PC ON Product.model = PC.model
WHERE Product.type = 'pc'
GROUP BY Product.maker;
```

l)

Find the average hard disk size of a PC for all those manufacturers that make printers.

We understand the query the following way: Find manufacturers that both produce printers and PCs, and then find the average hard disk size for the individual manufacturers and their PCs.

```
SELECT Product.maker, AVG(PC.hd) AS average_hd
FROM Product
JOIN PC ON Product.model = PC.model
WHERE Product.maker IN (
    SELECT maker
    FROM Product
    WHERE Product.type = 'printer'
)
GROUP BY Product.maker;
```

Exercise 4

For the same database schema (and example database), write (and evaluate) the following database modifications, based on the database schema. After executing each of your modifications, you should restore the original database state, e.g., by re-executing `psql < product p.sql`.

a)

Using two INSERT statements, store in the database the fact that PC model 1100 is made by manufacturer C, has speed 3.2, RAM 1024, hard disk 180, and sells for \$2499.

```
INSERT INTO Product values('C', 1100, 'pc');
INSERT INTO PC values(1100, 3.2, 1024, 180, 2499);
```

b)

Delete all PCs with less than 100 gigabytes of hard disk.

```
DELETE FROM PC
WHERE hd < 100;
```

c)

Delete all laptops made by a manufacturer that does not make printers.

```
DELETE FROM Laptop
WHERE model IN (
    SELECT model
    FROM Product
    WHERE type = 'laptop' AND maker NOT IN (
        SELECT maker
        FROM Product
        WHERE type = 'printer'
    )
);
```

d)

Manufacturer A acquires manufacturer B. Change all of B's products, so they are now made by A.

```
UPDATE Product
SET maker = 'A'
WHERE maker = 'B';
```


e)

For each PC, double the amount of RAM and add 60 gigabytes to the amount of hard disk.

```
UPDATE PC
SET ram = (ram * 2), hd = (hd + 60);
```

f)

For each laptop made by manufacturer B, add one inch to the screen size and subtract \$100 from the price.

```
UPDATE Laptop
SET screen = screen + 1, price = price - 100
WHERE model IN (
    SELECT model
    FROM Product
    WHERE type = 'laptop' AND maker = 'B'
);
```

g)

Enough! Delete all products!

```
DELETE FROM Product;
DELETE FROM PC;
DELETE FROM Laptop;
DELETE FROM Printer;
```