

Decision support databases: Exercises

SALVATORE RUGGIERI



Example 1 (website): Ex 1.

Products		
PkP	UnitPrice	...
10	5	...
20	10	...
30	20	...

Sales		
FkP	Qty	...
10	50	...
20	10	...
30	20	...
10	30	...
20	100	...
30	10	...
10	30	...

- Write an SQL query to find the total sales revenue by product.
- Give a logical query plan for the SQL query, the type and the value of the result.
Modify the logical query plan to consider only products with $\text{UnitPrice} > 5$ sold each of them more than 5 times.
- Modify the SQL query to find also the rank of the product total quantity sold (the highest is first)).

Extra exercises

`Sales(Customer, Date, Store, Product, Amount)`

Write SQL query (possibly, with analytic functions) to find:

- Number of customers per store with a total amount less or equal than 100
- Number of customers per store with a total amount greater or equal than 100 in at least one single date
- Number of customers per store with a total amount lower or equal than 100 in all dates (with a sale for the customer in that store)
- Number of customers per store that spend more than 50% of their total in that store
- Number of customers per store that spend the largest amount of their total in that store

Example 1 (website): Ex 1.

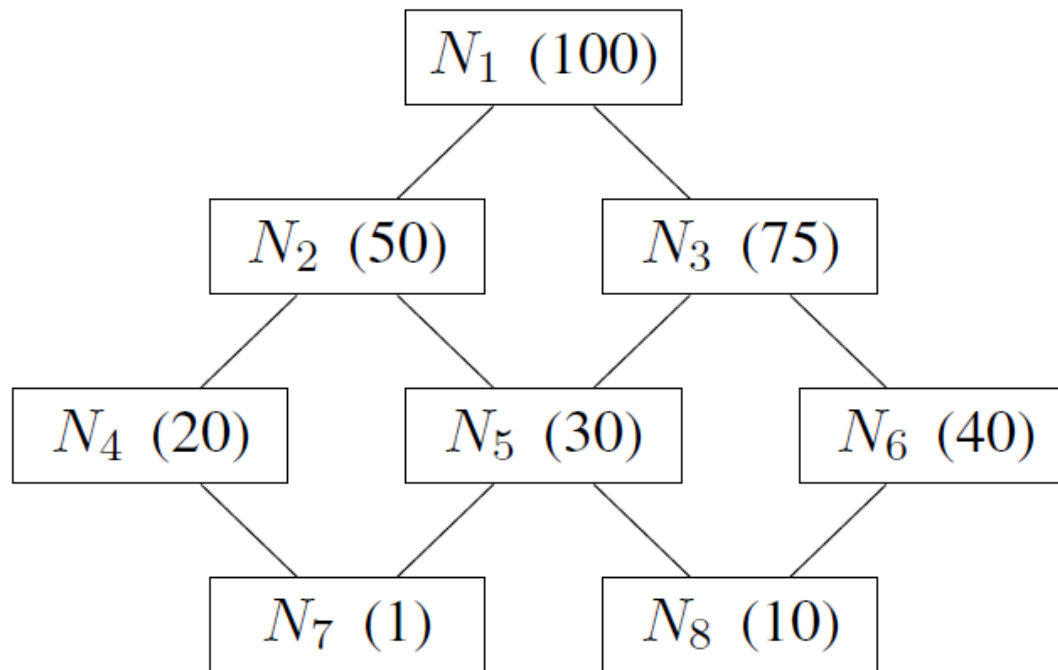
Products		
PkP	UnitPrice	...
10	5	...
20	10	...
30	20	...

Sales		
FkP	Qty	...
10	50	...
20	10	...
30	20	...
10	30	...
20	100	...
30	10	...
10	30	...

- (d) Show the instance of an index on the attribute FkP.
- (e) Show the instance of a *Foreign Column Join Index* on the attribute UnitPrice.

Example 1 (website): Ex 3.

Selects views to materialize in addition to N_1 .



Example 1 (website): Ex 4 a.

4. Let us consider the logical schema of a data mart

Customer(PkCustPhoneNo, CustName, CustCity)

CallingPlans(PkPlanId, PlanName)

Calls(PkCustPhoneNo, FkPlanId, Day, Month, Year, Duration, Charge)

where PkPlanId e PlanName are two different keys, and the following query

```
Q:  SELECT      Year, PlanName, SUM(Charge) AS TC
     FROM      Calls, CallingPlans
     WHERE     FkPlanId = PkPlanId AND Year >= 2000 AND Year <=2005
     GROUP BY Year, PlanName
     HAVING    SUM(Charge) > 1000;
```

(a) **(3 points)** Show if and how the GROUP BY can be brought forward on the table Calls.

Example 2 (website): Ex 4 a,b.

4. Let us consider the database without null values:

Customer(PKCustomer, CName, CCity)

Order(PKOrder, FKCustomer, ODate)

Product(PKProduct, PName, PCost)

OrderLine(LineNo, FKOrder, FKProduct, Quantity, ExtendedPrice, Discount, Revenue)

and the query

Q: **SELECT** CCity, AVG(Revenue) **AS** avgR
FROM OrderLine, Order, Customer
WHERE FKOrder = PKOrder **AND** FKCustomer = PKCustomer
GROUP BY CCity, FKCustomer
HAVING SUM(Revenue) > 1000;

(a) **(2 points)** Show if and how the **GROUP BY** can be pushed on the join

(OrderLine $\bowtie_{FKOrder = PKOrder}$ Order).

(b) **(2 points)** Show if and how the **GROUP BY** can be pushed on the relation OrderLine.

Example 1 (website): Ex 4 b.

4. Let us consider the logical schema of a data mart

Customer(PkCustPhoneNo, CustName, CustCity)

CallingPlans(PkPlanId, PlanName)

Calls(PkCustPhoneNo, FkPlanId, Day, Month, Year, Duration, Charge)

where PkPlanId e PlanName are two different keys, and the following query

Q: **SELECT** Year, PlanName, SUM(Charge) AS TC
FROM Calls, CallingPlans
WHERE FkPlanId = PkPlanId **AND** Year >= 2000 **AND** Year <=2005
GROUP BY Year, PlanName
HAVING SUM(Charge) > 1000;

(b) (5 points) Show if and how the query can be rewritten using the materialized view

V1: **SELECT** FkPlanId, Month, Year, SUM(Charge) AS C
FROM Calls
WHERE Year >= 2000
GROUP BY FkPlanId, Month, Year;

Example 2 (website): Ex 4 c.

4. Let us consider the database without null values:

Customer(PKCustomer, CName, CCity)

Order(PKOrder, FKCustomer, ODate)

Product(PKProduct, PName, PCost)

OrderLine(LineNo, FKOrder, FKProduct, Quantity, ExtendedPrice, Discount, Revenue)

and the query

```
Q:  SELECT      CCity, AVG(Revenue) AS avgR
     FROM      OrderLine, Order, Customer
     WHERE     FKOrder = PKOrder AND FKCustomer = PKCustomer
     GROUP BY CCity, FKCustomer
     HAVING    SUM(Revenue) > 1000;
```

(c) (4 points) Show if and how the query Q can be rewritten using the materialized view V

```
V:  SELECT     FKCustomer, SUM(Revenue) AS TR, COUNT(*) AS Cnt
     FROM      OrderLine, Order
     WHERE     FKOrder = PKOrder
     GROUP BY FKCustomer;
```

Extra exercises

Let us consider the logical schema of a data mart, without null values,

Customers(Phone, CustName, CustCity)

CallingPlans(PlanId, PlanName)

Calls(Phone, PlanId, Day, Month, Year, Duration, Charge)

and the following query

```
Q:  SELECT      CustCity, SUM(Charge) AS SC
      FROM      Calls NATURAL JOIN Customers
      AND Year = 2005 AND CustCity IN ('Roma', 'Milano')
      GROUP BY CustCity;
```

1. Show if and how the GROUP BY can be brought forward on the table Calls.
2. Show if and how the query Q can be rewritten using the materialized view:

```
V:  SELECT      Phone, Year, SUM(Charge) AS SC
      FROM      Calls NATURAL JOIN Customers
      AND CustCity IN ('Roma', 'Milano', 'Firenze', 'Torino')
      GROUP BY Phone, Year;
```