First MidTerm of *Decision Support Systems / Decision Support Databases*

It is forbidden to consult any material during the test. Duration of written exam is 1.5h.

1. Let us consider the following database, without null values:

Products

PkP	Category
10	Tools
20	Tools
30	Garden
40	Garden

Sales

FkP	Qty	UnitPrice
10	2	15.00
20	3	20.00
10	1	12.00
30	4	8.00
20	2	18.00
30	1	10.00

   (a) **(1 point)** Give an SQL query to find the total revenue by product Category, for categories with at least two sales. Revenue is Qty times UnitPrice.

   (b) **(1 point)** Give the query logical tree, the type and the value of the result of the SQL query.

   (c) **(1 point)** Write an SQL query returning the PkP of products that have never been sold.

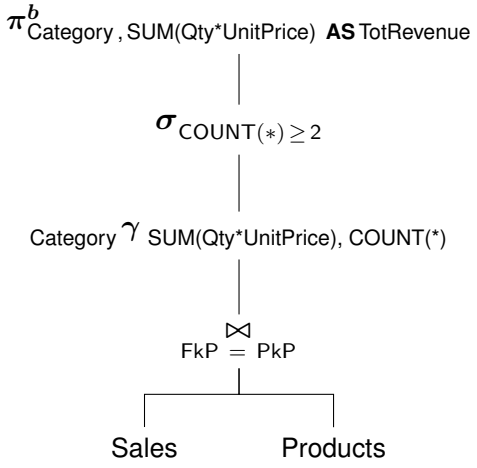
   (d) **(1 point)** A fact table `Stock(ProductFK, DateFK, ClosingQty)` records one end-of-day stock quantity for each product and day. Explain why `ClosingQty` is semi-additive, identifying the dimension along which `SUM` is inappropriate. Give one meaningful aggregation and one SQL aggregation that produces a misleading stock total.

Solution:

(a)

```
SELECT Category, SUM(Qty*UnitPrice) AS TotRevenue
FROM Sales, Products
WHERE FkP = PkP
GROUP BY Category
HAVING COUNT(*) >= 2
```

(b) The logical query plan:



The type of the result is {{`(Category:string, TotRevenue:float)`}}. The value of the result is:

Category	TotRevenue
Tools	138.00
Garden	42.00

(c)

```
(SELECT PkP
FROM Products)
EXCEPT
(SELECT DISTINCT FkP AS PkP
FROM Sales)
```

(d) `ClosingQty` is a *stock level* (periodic) measure: it records the quantity on hand at the end of a day. It is semi-additive with respect to `DateFK`: the closing quantities of two products on the *same* day can be summed, giving the stock of the two products together. However, this holds under the assumption that all products are measured with the same unit of measure. If this is not the case (e.g., we have products measured in liters and products measures in kilograms), then `ClosingQty` is non-additive. Under the assumptions, here it is a meaningful aggregation:

```
SELECT DateFK, SUM(ClosingQty) AS TotStock
FROM Stock
GROUP BY DateFK
```

and a misleading one:

```
SELECT ProductFK, SUM(ClosingQty) AS TotStock
FROM Stock
GROUP BY ProductFK
```

2. A company rents equipment through a network of branches. Each completed rental concerns exactly one physical equipment item and one customer. A customer renting several items generates a separate rental record for each item; each record has a unique rental identifier. The same item may be rented repeatedly.

   For each rental, the operational system records the pickup and return dates, the pickup and return branches (which may differ), the actual duration in days, and the total fee charged for that rental. Only completed rentals are analysed. All durations and fees are positive. Each equipment item has a code, a model, and one equipment type. Each model belongs to exactly one type. Each branch has a code and belongs to one city and region; each city belongs to one region.

   Each customer has a name, a segment (private or business), and a City and region of residence.

   Management is interested in the following business questions:

   (a) Number of rentals, total rental days, and total fees by equipment type and pickup month and year.

   (b) Total fees by customer segment and region of residence at pickup.

   (c) Average rental duration by equipment type and pickup branch.

   (d) Number of rentals by pickup branch, return branch, and return month and year.

   With respect to the above business scenario, answer the following questions:

   (a) **(6 points)** Design a conceptual schema for the data mart to support the business questions. Your schema should at least be able to satisfy the above mentioned analysis requirements. You may motivate other suitable attributes for the dimensions. Follow all the steps of the conceptual design methodology.

(b) **(3 points)** Give a logical data mart design, which includes the treatment of updates for dimensional attributes.

(c) **(1 point)** Write an SQL query to answer one business questions of your choice.

Solution:

(a) We follow the process of datawarehousing conceptual design. Tables 1 and 2 show the preliminary analysis.

ID	Dimensions	Measures	Metrics
(a)	Equipment (type), PickupDate (month, year)	Duration, Fee	No of rentals, TotalDuration, TotalFees
(b)	Customer (segment, region of residence)	Fee	TotalFees
(c)	Equipment (type), PickupBranch	Duration	AverageDuration
(d)	PickupBranch, ReturnBranch, ReturnDate (month, year)	–	No of rentals

Table 1: Business process requirements

Description	Preliminary Dimensions	Preliminary Measures
A fact *Rental* is a single completed rental record: one equipment item rented by one customer, from a pickup (date and branch) to a return (date and branch).	Equipment, Customer, PickupDate, ReturnDate, PickupBranch, ReturnBranch	Duration, Fee

Table 2: Fact description

In addition to the preliminary dimensions, from the context description, we also consider a degenerate dimension RentalsID to record the rentals made at a same time by a customer. Moreover, PickupDates and ReturnDates aree extended with the day attribute; and PickupBranch and ReturnBranch are extended with geographial attributes. Figure 1 shows the conceptual DFM schema.

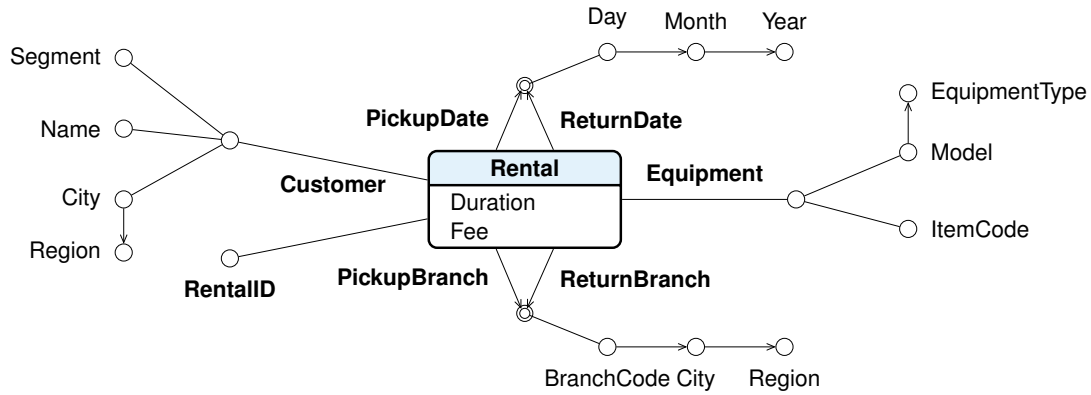


Figure 1: DFM schema.

(b) Figure 2 shows the logical schema. The following dimensional attributes are updated as Type 2: Segment, City and Region of the Customer dimension. All other attributes are updated as Type 1.

(c) We show next the SQL query for the first business question:

Number of rentals, total rental days, and total fees by equipment type and pickup month and year.

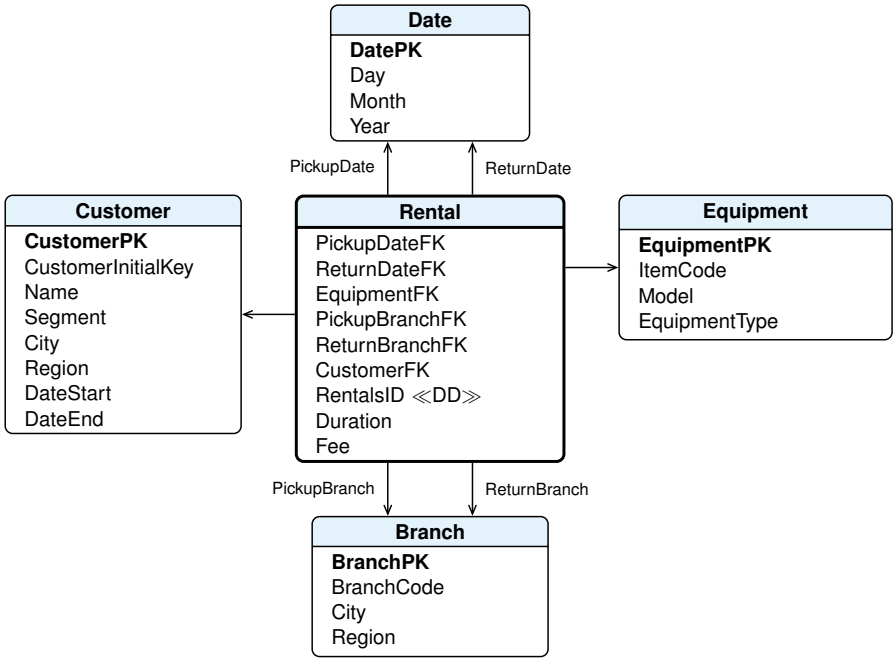


Figure 2: Logical schema.

```
SELECT EquipmentType, Month, Year,
       COUNT(*)      AS NoRentals,
       SUM(Duration) AS TotDuration,
       SUM(Fee)      AS TotFees
FROM Rental, Equipment, Date
WHERE EquipmentFK = EquipmentPK AND PickupDateFK = DatePK
GROUP BY EquipmentType, Month, Year
```

3. **(2 points)** A multidimensional cube has dimensions *EquipmentType*, *Branch*, and *Month*, and the additive measure *RentalFees*. Months include the year. Only the following two cuboids are available; the fact table cannot be accessed:

$$A(\text{EquipmentType}, \text{Branch}, *) \qquad B(*, \text{Branch}, \text{Month}).$$

Can the total rental fees for each branch, over all equipment types and months, be computed from A alone? From B alone? For each case, name the OLAP operation and the dimension on which it acts. Explain the relationship that must hold between the two resulting reports.

Solution: The requested report is the cuboid $C(*, \text{Branch}, *)$, which can be computed from both A and B as follows:

- from $A(\text{EquipmentType}, \text{Branch}, *)$: it is enough to **roll-up** along the *EquipmentType* dimension. In SQL: `SELECT Branch, SUM(RentalFees) FROM A GROUP BY Branch`.
- from $B(*, \text{Branch}, \text{Month})$: it is enough to **roll-up** along the *Month* dimension. In SQL: `SELECT Branch, SUM(RentalFees) FROM B GROUP BY Branch`.

The two resulting reports are the same, namely the cuboid $C(*, \text{Branch}, *)$. It can be computed from A or B only because such cuboids have finer (or same) granularity as C. It cannot be computed, for instance, from $D(*, *, *)$ nor from $E(*, \text{Branch}, \text{Nov2026})$.