Exercises Data Warehousing
Indexing

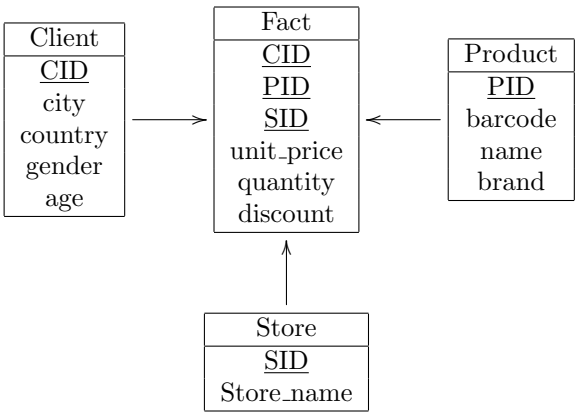
1. Consider the following relation `Cars`:

Brand	Type	Color	Risk
Opel	Corsa	Grey	Low
Opel	Corsa	Red	Medium
Peugeot	206	Black	Medium
BMW	A	Black	High

   (a) Construct a bitmap index for the attributes `Brand` and `Color` for this table.

   (b) Indicate how these two bitmap indices can be used to answer the query: *Give the total number of red Opel cars with a medium risk score.*

2. Consider the following Star schema for a ROLAP database. A fact (c, p, s, up, q, d) represents that customer c bought q units of product p in store s at unit price up and received a discount d.



   (a) Discuss the advantages and disadvantages of the following indices; for instance, describe under what circumstances and for which types of queries are these indices interesting.

      i. a bitmap join index for the attribute country from the table Client into the fact table (mapping countries to tuples in the fact table that are about this country);

      ii. a bitmap index on table Client for the attribute gender; and,

      iii. a bitmap index for the attribute unit_price in table Fact.

   (b) Consider the following query:

```
SELECT Client.Gender, SUM(Fact.quantity)
FROM Fact, Client
WHERE Fact.CID=Client.CID AND
      (Client.country = "The Netherlands"
       OR Client.country = "Belgium")
group by Client.Gender
```

   Select one of the indices from (a) and explain how this index may help to answer this query efficiently.

3. Deduplication is an important task in data cleaning. To do deduplication it is useful to have a measure of distance between string values such as names, addresses, phone numbers, etc. The edit distance is one such distance measure. Compute the edit distance between “Brussel” and “Bruges”.

4. Consider a cube with three dimensions A, B, and C, and one measure M. Suppose that the complete cube needs to be materialized for the aggregation function M. That is, given the base table B(IDA, IDB, IDC, M), we need to compute:

```
SELECT IDA, IDB, IDC, sum(M)
FROM B
GROUP BY CUBE(IDA, IDB, IDC)
```

Show how you could apply the pipe-sort algorithm to compute the cube. Assume that sorting a relation X takes SORT(X) time, and scanning the table takes FTS(X) (FTS stands for "full table scan"). Express the time needed by the query plan developed by the pipe-sort algorithm with the time needed by a brute-force solution that computes the aggregations for all grouping sets separately. What is the gain? (Express costs and gain in terms of SORT(X) and FTS(X) for X any of the aggregation tables constructed along the way)

5. Suppose that a cube $\mathsf{Sales}(\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{Amount})$ has to be fully materialized. The cube contains 64 tuples. Sorting takes the typical $n \log(n)$ time. Every GROUP BY with k attributes has 2^k tuples.

   (a) Compute the cube using the PipeSort algorithm.

   (b) Compute the gain of applying the PipeSort compared to the cost of computing all the views from scratch.