Bases de Dados 2020/2021

Project Assignment - Part 3

1. Database

For the Part 3 of the project you can reuse and improve the table schema of the previous project. You should ensure that the database is adequately populated to support the development of the rest of the project.

2. Integrity Constraints

Write the PgPLSQL code to implement the following integrity constraints with the SQL procedural extensions (Stored Procedures and Triggers) in your database schema:

(A) For a given product, there can only be at most 3 secondary suppliers
(B) For a given product, a supplier cannot be simultaneously a primary and secondary supplier

3. View

Create a view **supplier_stats** that returns the number of primary and secondary suppliers for each product.

4. Application Development

Create a web application using Python CGI scripts and HTML pages that allows users to:

a) Insert and remove categories and sub-categories;
b) Insert and remove a new product and its respective suppliers (primary or secondary), ensuring that this operation is atomic;
c) List replenishment events for a given product, including the number of replenished units;
d) Change the designation of a product;
e) List all the sub-categories of a super-category, at all levels of depth.

The solution should prize security, preventing attacks by SQL INJECTION. Additionally, the **atomicity of related operations** in the database should be ensured.

5. Indexes

Present the SQL index(es) creating instructions to improve the querying times for each of the cases listed below.

Indicate, with proper justification, what type of index(es), over which attribute(s) and over which table(s) it would make sense to create, in order to speed up each query execution. Assume that the size of the tables exceeds the available memory by several orders of magnitude.

Assume that there are no indexes over the tables, aside from those implicit when declaring primary and foreign keys.

5.1 List the NIF and NAME of every primary supplier of products of the category 'Vegetables'.

```
SELECT DISTINCT S.nif, S.name
FROM supplier S INNER JOIN product P
      ON s.nif = P.nif_primary
WHERE P.category = 'Vegetables'
```

5.2 List the number of secondary suppliers of products with more than one (secondary supplier)

```
SELECT ean, COUNT(*)
FROM product P, secondary_supplier S
WHERE P.ean = S.ean
GROUP BY P.ean
HAVING COUNT(*) > 1
```

6. Multidimensional Model

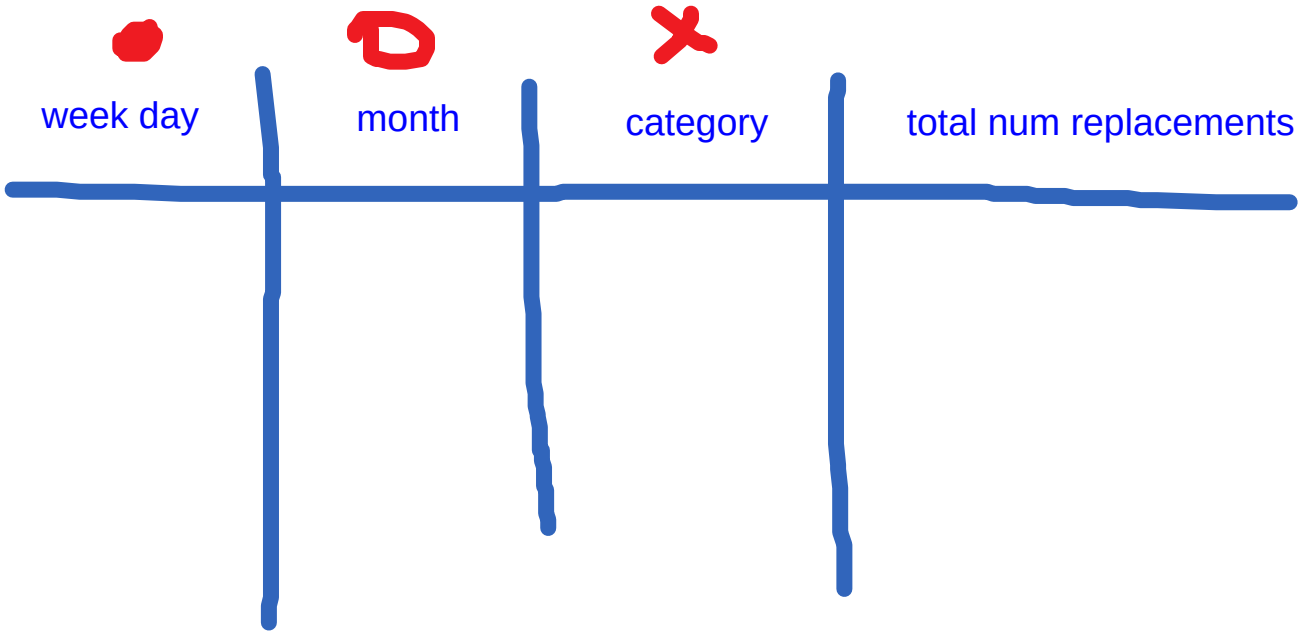
Create a star schema in the database with information concerning products and their corresponding replenishments, with at least the following dimensions:

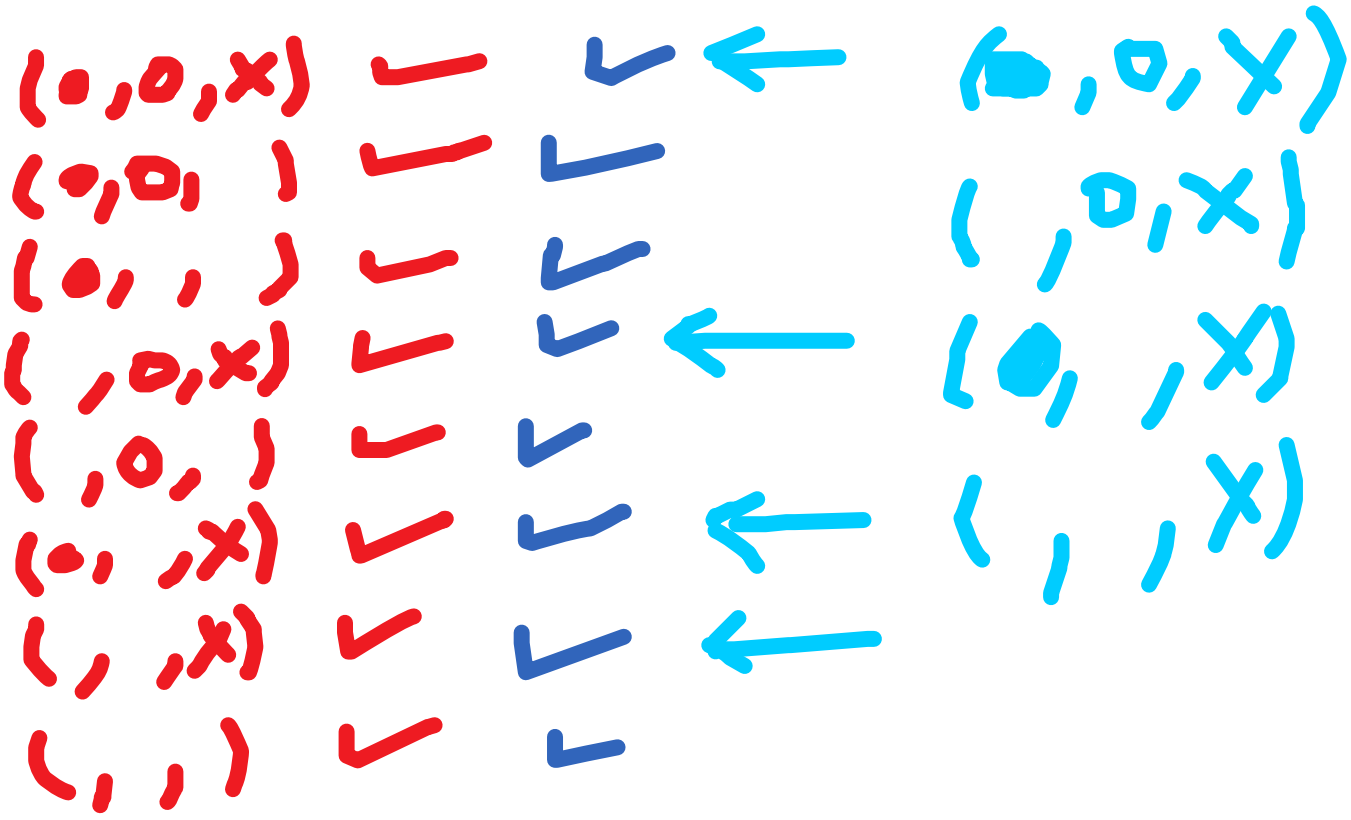
```
d_product(id_product, ean, category, nif_supplier)
d_date(id_date, day, week_day,  week, month, year)
```

Write the required SQL statements to create the dimension tables and the fact table. Write also the SQL statements to load the star schema from the existing tables, loading the dimension tables first, followed by the fact table. Attributes **id_product**, **id_date** are surrogate keys.

7. Data Analytics Queries

Within the context of the star schema created in the previous question, write an SQL query that allows you to analyze the total number of replenishments for each category by **week day** and **month**. The submitted solution must use ROLLUP, CUBE, GROUPING SETS instructions, or the UNION of GROUP BY clauses.





Report

The project will be graded based on the report submitted and the discussion. The report should contain all answers to the items requested above. The following table gives the breakdown of the points for each portion of the work.

Item	**Points**
Integrity Constraints	4
View	2
Application	6
Indexes	3
Multidimensional Model	3
ETL + Data analytics	2

The report should begin with a cover page with the title "**Database Project, Part 3**", the **name and number of the students, the contribution from each member in relative percentage, along with** the **effort (in hours)** that each member put into the project, the **group number**, the group **shift** and the lab teacher's name. Besides the cover page, the report should have, at most, **8 pages**.

Submission

The submission in Fénix should be a **zip** file with the following structure:

reportGG.pdf (where GG is the group number)	The report in **pdf** where **GG** is the group number, containing an simplified **explanation of the architecture of the web application with a link to a working version**, the **relations between the various files** and the **indexes**. You should not include the instructions used to populate the database. The report should not include the OLAP component. ⚠ Groups must make sure to have the application working online until after the discussion.
schema.sql	File with the SQL statements that create the database.
populate.sql	File to populate the database with the test data.

view.sql	File with the SQL of the view.
RI.sql	File to create the integrity constraints (triggers & stored procedures).
star_schema.sql	File to create the multidimensional schema.
etl.sql	File with the script to load the star schema.
olap.sql	File with OLAP queries
web/	Folder with the Python and HTML files.

The project must be delivered in ZIP formatted with the name `delivery-03-GG.zip`[1] (where **GG** is the group number), through Fénix until 23h59 of the delivery date.

21/05/2021 - 23h59

Note: Penalties apply to groups that do not follow the delivery instructions. Evaluation elements that are not found in the prescribed as above **will not be taken into account for grading purposes**.

[1] ⚠ The file format must be exclusively ZIP or GZ. Other archive formats (such as RAR) will not be accepted.