# Data Engineering

# Processing Sales Case Details

## Introduction

The Employee-Sales application (first version, normalized), suffers from product price being in the Product table. This design choice assumes price will remain fixed forever, which is impractical.

You are receiving the database when a considerable number of transactions have already been reported. You will have to bring the database in a new state. The following problems outline the sequence of steps you need to take.

## Tasks

### Price History

#### Create table

Create a table price\_history to store dates on which new prices are set in effect.

An entry

('23-MAR-2022', 'Pr12.G.0182', 312.85)

would indicate that, effective 23 March, 2022 and until a later date all sales of product Pr12.G.0182 were made at a price of $312.85.

Provide a CREATE TABLE statement that creates the appropriate table. Be sure to integrate it to the database so that data integrity is preserved.

#### Populate the table

Operations recording started on Jan 1, 2022. At that time a price was set for each product. On Jan 1, 2024, all products price was uniformly raised by 25 euros. This 2024 price has been in effect since and have not been changed.

Provide **INSERT SELECT** queries to populate the table with the previous prices, set on the correct date.

### Sales value

A record in the sales table was previous directly linked to price in product. Now you need a more elaborate query, which will correctly retrieve the price. This is reflected in the join conditions of sales and price\_history. You need to join on product and locate the one price that is in effect on the date found in sales.

This date is the max date found in price\_history for the product, that is less or equal than sales date. It is up to you to choose in which order to perform the following two tasks.

#### Build a view

Build a view which could be the base for several other queries involving sales value. This view essentially expands the Sales table with the proper price and should return

* Transaction date (sales date),
* Employee id,
* Product id,
* Customer id,
* Quantity,
* Price in effect on transaction date

## Some Analytics SQL

“Find all customers whose total orders last year exceeded 20% of the aggregate sales for their geographic region.”