

# Bershka – Data Engineer

A continuación, se plantean una serie de cuestiones de SQL a responder. Para ello, se ha preparado un esquema en [sqlfiddle.com](http://sqlfiddle.com/#!17/4a785/1) (PostgreSQL) sobre el que poder apoyarse.

Las cuestiones se encuentran agrupadas en tres bloques: queries, índices e historificación.

→ Esquema de apoyo: <http://sqlfiddle.com/#!17/4a785/1>

## Bloque 1 - Queries:

Responda escribiendo la sentencia SQL acompañada de una explicación de lo que hace.

- I. Calcula el listado de los 10 productos con menos unidades vendidas (hint: incluye productos sin ventas)
- II. Calcula un listado de clientes que tienen cesta media (facturación dividida en número de pedidos) inferior a la cesta media de su género.
- III. Calcula la recency de compra (días desde la última compra) para todos los clientes con compra.
- IV. Usa la función ntile para asignar los productos a 3 tiers donde el tier más alto incluye los productos con mayor facturación.

## Bloque 2 – Índices:

Razone sobre la coherencia y el rendimiento de los siguientes índices, y explique cómo podrían mejorarse.

- I. 

```
CREATE INDEX tbl_idx ON tbl (date_column)
SELECT COUNT(*) FROM tbl WHERE DATEPART(YEAR, date_column) = 2017
```
- II. 

```
CREATE INDEX tbl_idx ON tbl (a, date_column)
SELECT TOP 1 * FROM tbl WHERE a = 12 ORDER BY date_column DESC
```
- III. 

```
CREATE INDEX tbl_idx ON tbl (a, b)
SELECT * FROM tbl WHERE a = 38 AND b = 1
SELECT * FROM tbl WHERE b = 1
```
- IV. 

```
CREATE INDEX tbl_idx ON tbl (text)
SELECT * FROM tbl WHERE text LIKE 'TJ%'
```

## Bloque 3 – Historificación:

Se cuenta con una tabla de *staging* que contiene información de precios por artículo, y que se actualiza diariamente con información de los artículos más recientes de forma que los

artículos más antiguos dejan de aparecer en esta tabla. Un artículo puede aparecer varias veces con precios diferentes (fechas diferentes también).

Debido al alto volumen de información diaria se desea aplicar una historificación a los datos que permita generar una tabla final con el registro de los cambios de estado.

Puede observarse la estructura de ambas tablas a continuación:

STG_PRECIOS	
CAMPO	TIPO
ID_FECHA	DATE
ID_ARTICULO	INT
PRECIO_COSTE	DECIMAL(10, 2)
PRECIO_VENTA	DECIMAL(10, 2)

FACT_PRECIOS	
CAMPO	TIPO
FECHA_INICIO_VIGENCIA	DATE
FECHA_FIN_VIGENCIA	DATE
ID_ARTICULO	INT
PRECIO_COSTE	DECIMAL(10, 2)
PRECIO_VENTA	DECIMAL(10, 2)

Implemente un *procedure* en T-SQL que genere dicha historificación.