Challenge Mercado Libre

Autor:

- Parente Aun, Ruben Alfredo

Contenido

[Introducción. 2](#_Toc104895597)

[- Temática. 2](#_Toc104895598)

[- Presentación. 2](#_Toc104895599)

[Primera Parte - SQL 2](#_Toc104895600)

[Stack tecnológico. 2](#_Toc104895601)

[SQL Server 2](#_Toc104895602)

[Diagramas 3](#_Toc104895603)

[Tablas 4](#_Toc104895604)

[A resolver 9](#_Toc104895605)

[Segunda Parte - APIs (Deseable) 11](#_Toc104895606)

# Introducción.

## - Temática.

Interpretación de la consigna ● Orden y comentarios del código ● Queries simples y efectivas ● Calidad del entregable ● Si pensás en otro punto que pueda sumar, sentite libre de agregarlo y comentarnos cuál es y por qué lo sumaste

## - Presentación.

* Alfredo Parente: DATA SCIENTIST.Un dibujo de una persona

  Descripción generada automáticamente con confianza media

LinkedIn: https://www.linkedin.com/in/alfredo-parente-2b563851

# Primera Parte - SQL

## Stack tecnológico.

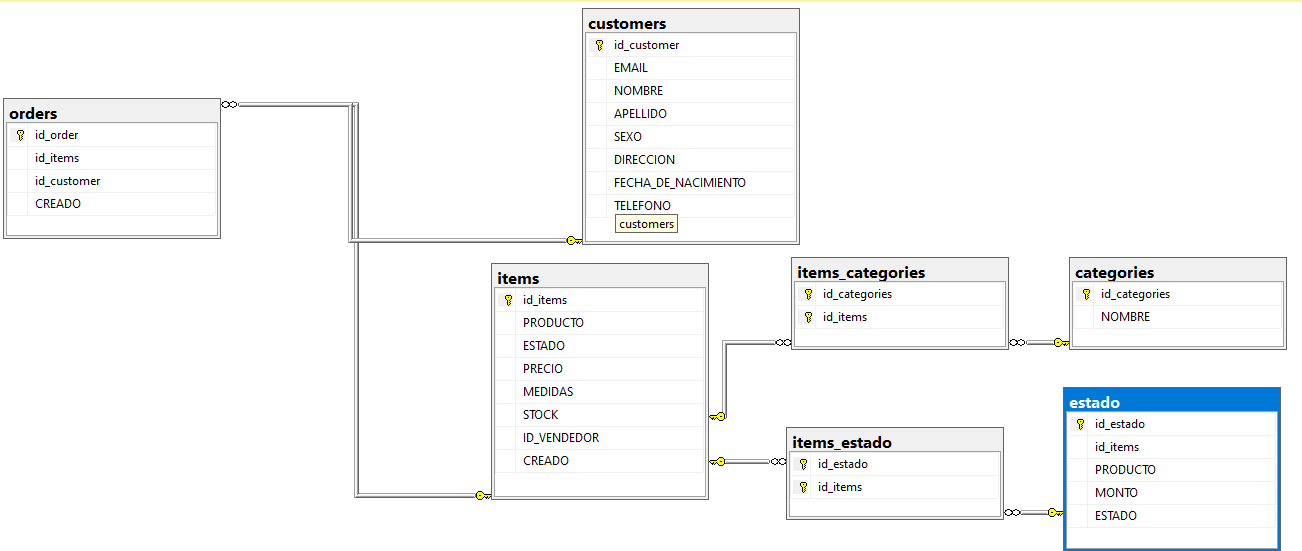
En este trabajo se utilizarán las siguientes herramientas:

## Logotipo, nombre de la empresa Descripción generada automáticamenteSQL Server

## Diagramas

* A nivel lógico:

En esta representación gráfica del DER se le suma columnas que brindan el detalle de las Primary y Foreign Keys, y su vez el tipo de dato en la última columna.



## Tablas

* Tabla Customers:

|  |  |  |  |
| --- | --- | --- | --- |
| customers | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_customer | Llave primaria | PK | NVARCHAR (50) |
| EMAIL | Dirección de correo electrónico del cliente |  | NVARCHAR (50) |
| NOMBRE | Primer y segundo nombre del cliente |  | NVARCHAR (50) |
| APELLIDO | Primer y segundo apellido del cliente |  | NVARCHAR (50) |
| SEXO | Género del cliente |  | NVARCHAR (1) |
| DIRECCION | Dirección particular del cliente |  | NVARCHAR (100) |
| FECHA\_DE\_NACIMIENTO | Fecha en la que nació el cliente |  | DATE |
| TELEFONO | Teléfono de contacto |  | NVARCHAR (14) |

* Creación de la tabla Customers:

CREATE TABLE customers (

id\_customer NVARCHAR(50)PRIMARY KEY,

EMAIL NVARCHAR(50) NOT NULL,

NOMBRE NVARCHAR(50) NOT NULL,

APELLIDO NVARCHAR(50) NOT NULL,

SEXO NVARCHAR(1) NOT NULL,

DIRECCION NVARCHAR(100) NULL,

FECHA\_DE\_NACIMIENTO DATE NULL,

TELEFONO NVARCHAR (14) NULL

);

|  |  |  |  |
| --- | --- | --- | --- |
| items | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_items | Llave primaria | PK | NVARCHAR (50) |
| PRODUCTO | Descripción del producto |  | NVARCHAR (100) |
| ESTADO | Estado actual del producto |  | NVARCHAR (50) |
| PRECIO | Precio actual del item |  | INT |
| MEDIDAS | Medidas del producto |  | INT |
| STOCK | Cantidad de producto disponible para la venta |  | INT |
| ID\_VENDEDOR | Identificación del vendedor en el sistema |  | INT |
| CREADO | Fecha de la publicación |  | DATE |

* Tabla items:
* Creación de la tabla items:

CREATE TABLE items (

id\_items NVARCHAR (50) PRIMARY KEY,

PRODUCTO NVARCHAR(100) NOT NULL,

ESTADO NVARCHAR (50) NOT NULL,

PRECIO INT NOT NULL,

MEDIDAS INT NOT NULL,

STOCK INT NOT NULL,

ID\_VENDEDOR INT NOT NULL,

CREADO DATE NULL

);

* Tabla categories:

|  |  |  |  |
| --- | --- | --- | --- |
| categories | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_categories | Llave primaria | PK | NVARCHAR (50) |
| NOMBRE | Nombre de la Categoria para su identificación |  | NVARCHAR (50) |

* Creación de la tabla categories:

CREATE TABLE categories (

id\_categories NVARCHAR (50) PRIMARY KEY,

NOMBRE VARCHAR(50) NOT NULL

);

* Tabla orders:

|  |  |  |  |
| --- | --- | --- | --- |
| orders | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_order | Llave primaria | PK | NVARCHAR (50) |
| Id\_items | Nombre de la Categoria para su identificación | FK | NVARCHAR (50) |
| id\_customer | Llave primaria | FK | NVARCHAR (50) |
| CREADO | Fecha de concretación de la venta. |  | DATE |

* Creación de la tabla orders:

CREATE TABLE orders (

id\_order NVARCHAR (50) PRIMARY KEY,

id\_items NVARCHAR (50) foreign key references items (id\_items),

id\_customer NVARCHAR (50) foreign key references customers (id\_customer),

CREADO DATE NULL

);

* Tabla order\_items\_customer:

|  |  |  |  |
| --- | --- | --- | --- |
| orders | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_order | Llave primaria | PK | NVARCHAR (50) |
| Id\_items | Nombre de la Categoria para su identificación | FK | NVARCHAR (50) |
| id\_customer | Llave primaria | FK | NVARCHAR (50) |
| CREADO | Fecha de concretación de la venta. |  | DATE |

|  |  |  |  |
| --- | --- | --- | --- |
| order\_items\_customer | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_items | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |
| Id\_customer | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |

|  |  |  |  |
| --- | --- | --- | --- |
| orders | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_order | Llave primaria | PK | NVARCHAR (50) |
| Id\_items | Nombre de la Categoria para su identificación | FK | NVARCHAR (50) |
| id\_customer | Llave primaria | FK | NVARCHAR (50) |
| CREADO | Fecha de concretación de la venta. |  | DATE |

* Creación de la tabla order\_items\_customer:

CREATE TABLE order\_items\_customer (

id\_items NVARCHAR (50),

id\_customer NVARCHAR (50),

primary key(id\_items, id\_customer),

foreign key (id\_items) references items (id\_items),

foreign key (id\_customer) references customers (id\_customer)

)

* Tabla items\_categories:

|  |  |  |  |
| --- | --- | --- | --- |
| order\_items\_customer | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_categories | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |
| Id\_items | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |

* Creación de la tabla ítems\_categories:

CREATE TABLE items\_categories (

id\_categories NVARCHAR (50),

id\_items NVARCHAR (50),

primary key(id\_categories, id\_items),

foreign key (id\_categories) references categories (id\_categories),

foreign key (id\_items) references items (id\_items)

)

* Tabla estado:

|  |  |  |  |
| --- | --- | --- | --- |
| estado | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_estado | Llave primaria | PK | NVARCHAR (50) |
| Id\_items | Llave primaria de tabla items | FK | NVARCHAR (50) |
| PRODUCTO | Nombre del producto |  | NVARCHAR (50) |
| MONTO | Monto del producto. |  | DATE |
| ESTADO | Estado de la venta |  | DATE |

* Creación de la tabla estado:

CREATE TABLE estado (

id\_estado NVARCHAR (50) PRIMARY KEY,

id\_items NVARCHAR (50),

PRODUCTO NVARCHAR (50),

MONTO INT,

ESTADO NVARCHAR (50)

);

|  |  |  |  |
| --- | --- | --- | --- |
| items\_estado | | | |
| CAMPO | DESCRIPCION | TIPO DE CLAVE | TIPO DE CAMPO |
| id\_estado | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |
| Id\_items | Llave primaria - Llave foránea | PK-FK | NVARCHAR (50) |

* Tabla items\_estado:
* Creación de la tabla ítems\_estado:

CREATE TABLE items\_estado (

id\_estado NVARCHAR (50),

id\_items NVARCHAR (50),

primary key(id\_estado, id\_items),

foreign key (id\_items) references items (id\_items),

foreign key (id\_estado) references estado (id\_estado)

);

## A resolver

1. Listar los usuarios que cumplan años el día de hoy cuya cantidad de ventas realizadas en enero 2020 sea superior a 1500.

SELECT \*

FROM customers C

JOIN orders O ON C.id\_customer = O.id\_customer

WHERE C.FECHA\_DE\_NACIMIENTO = GETDATE()

AND C.id\_customer IN ( SELECT Tabla\_Sumarizada.id\_customer

FROM ( SELECT SUM (O.id\_customer) as Suma\_Usuario,

O.id\_customer AS ID\_CUSTOMER

FROM customers C

JOIN orders O ON C.id\_customer = O.id\_customer

WHERE CAST (YEAR (O.CREADO) AS VARCHAR) + CAST(MONTH(O.CREADO) AS VARCHAR) = '20201'

GROUP BY O.id\_customer

ORDER BY Suma\_Usuario ASC

) Tabla\_Sumarizada

WHERE Tabla\_Sumarizada.Suma\_Usuario > 1500

)

1. Por cada mes del 2020, se solicita el top 5 de usuarios que más vendieron ($) en la categoría Celulares. Se requiere el mes y año de análisis, nombre y apellido del vendedor, cantidad de ventas realizadas, cantidad de productos vendidos y el monto total transaccionado.

SELECT MONTH (O.CREADO) MES,

YEAR (O.CREADO) AÑO,

C.NOMBRE NOMBRE,

C.APELLIDO AS APELLIDO,

TOP\_USUARIOS.TOTAL\_VENTAS AS TOTAL\_VENTAS,

TOP\_USUARIOS.TOTAL\_PRECIO AS TOTAL\_PRECIO

FROM customers C

JOIN orders O ON C.id\_customer = O.id\_customer

JOIN (

SELECT TOP 5 COUNT(items.id\_items) AS TOTAL\_VENTAS,

SUM(items.PRECIO) AS TOTAL\_PRECIO,

C.id\_customer AS ID\_CUSTOMER

FROM customers C

JOIN orders O ON O.id\_customer = C.id\_customer

JOIN items ON items.id\_items = O.id\_items

JOIN categories CAT ON CAT.id\_categories = items.id\_items

WHERE CAT.NOMBRE LIKE 'CELULARES'

GROUP BY CAST(YEAR(O.CREADO) AS VARCHAR)+CAST(MONTH(O.CREADO) AS VARCHAR), C.id\_customer

ORDER BY C.id\_customer ASC

) TOP\_USUARIOS

ON C.id\_customer = TOP\_USUARIOS.ID\_CUSTOMER

;

3. Se solicita poblar una nueva tabla con el precio y estado de los Ítems a fin del día. Tener en cuenta que debe ser reprocesable. Vale resaltar que en la tabla Item, vamos a tener únicamente el último estado informado por la PK definida. (Se puede resolver a través de StoredProcedure)

GO

CREATE PROCEDURE TABLA\_PRECIO\_ESTADO\_ITEMS

@DIA DATETIME

AS

INSERT INTO estado

SELECT estado.id\_estado,

items.id\_items,

items.PRODUCTO,

estado.MONTO AS MONTO,

'CERRADO' AS ESTADO

FROM items

JOIN orders ON orders.id\_items = orders.id\_items

JOIN estado ON estado.id\_items = orders.id\_items

JOIN items\_estado ON items.id\_items = items\_estado.id\_items

WHERE orders.CREADO = @DIA

UNION

SELECT estado.id\_estado,

items.id\_items,

items.PRODUCTO,

estado.MONTO AS MONTO,

'CERRADO' AS ESTADO

FROM items

JOIN orders ON items.id\_items = orders.id\_items

JOIN estado ON estado.id\_items = orders.id\_items

JOIN items\_estado ON items.id\_items = items\_estado.id\_items

WHERE items\_estado.id\_items IS NULL

AND orders.id\_items IS NULL

# Segunda Parte - APIs (Deseable)

Se adjunta Notebook con la respuesta de los tres puntos de este challenge.