

Lenguaje SQL VI

Alex Di Genova

09/05/2022

Contenidos

- Sentencias condicionales
- Consultas avanzadas II

Sentencias condicionales

SQL

SELECT

- **SELECT** pipeline

SELECT [DISTINCT] *select_header*

FROM *tablas*

WHERE *expresion_filtrado*

GROUP BY *expresion_de_agrupamiento*

HAVING *expresion_filtrado*

ORDER BY *expresion_orden*

LIMIT *contador*

OFFSET indice

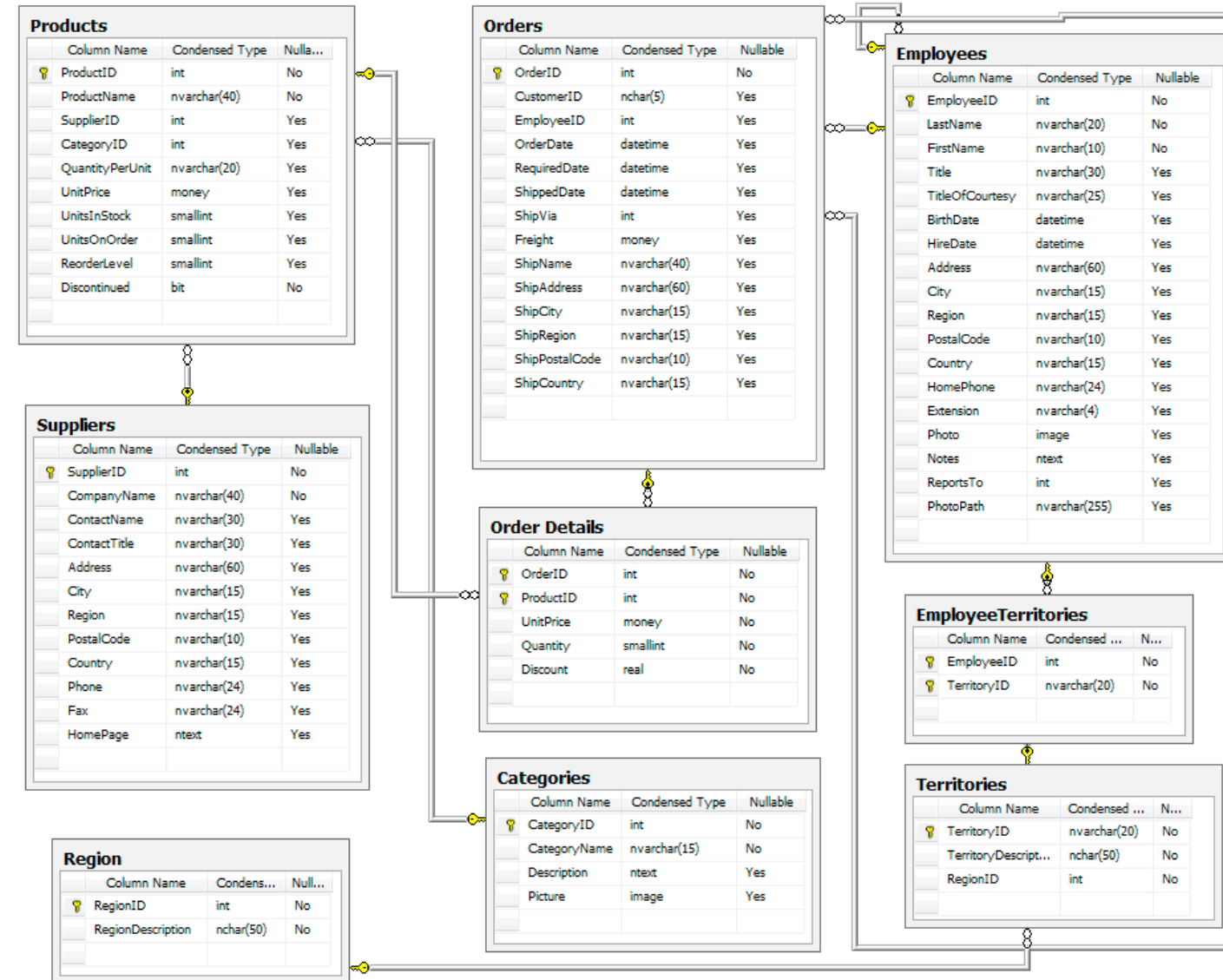
CASE

IF-THEN-ELSE

- Determine si los empleados son 'experimentados' o 'Junior' en función de la fecha de contratación

```
select LastName || " " || FirstName "Nombre", HireDate as
"Fecha Contrato",
CASE WHEN HireDate < '2025-01-01' THEN 'Experimentado'
WHEN HireDate >= '2025-01-01' THEN 'Junior'
ELSE 'Otro' END AS "Estado"
FROM Employee
```

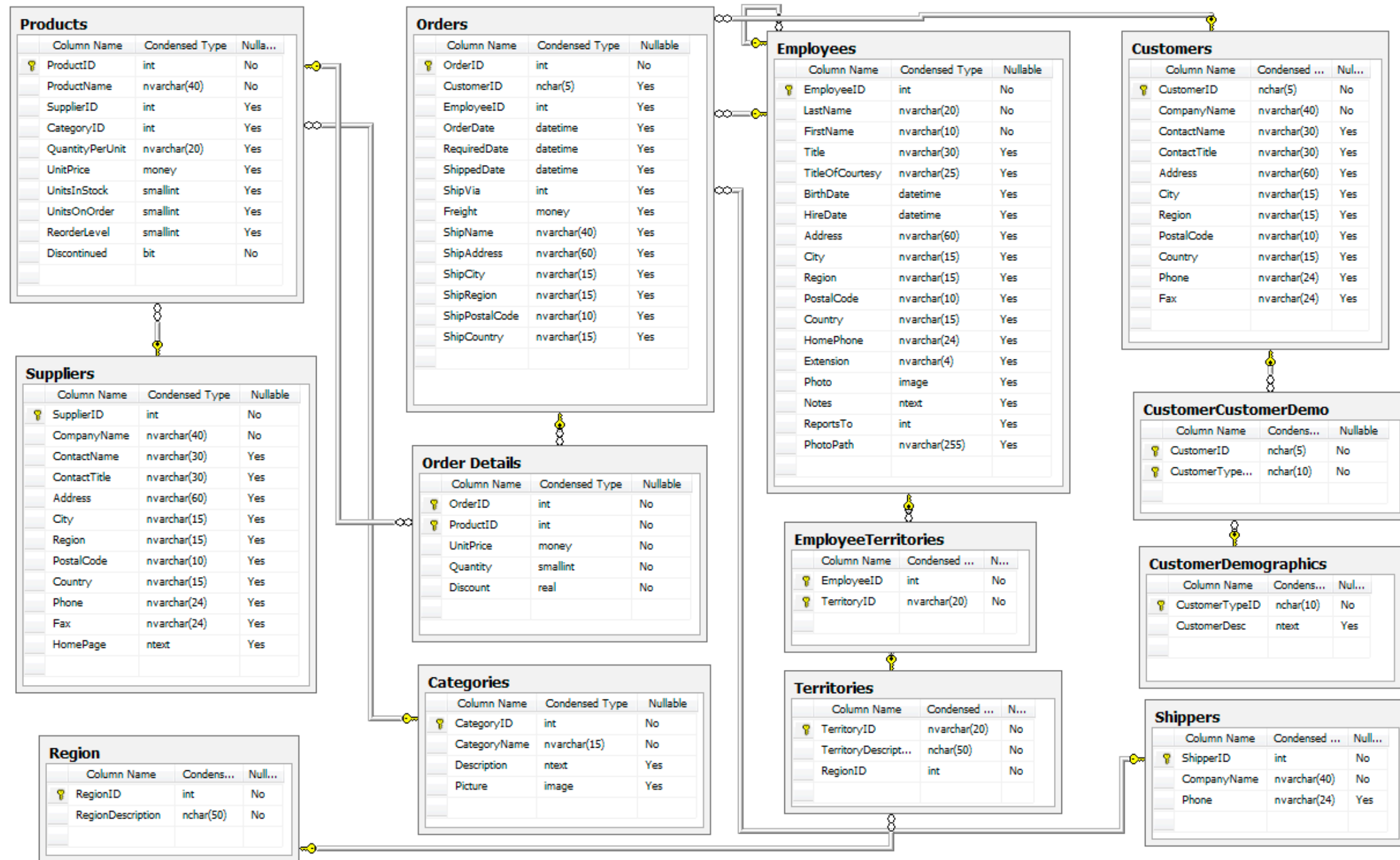
| Nombre | Fecha Contrato | Estado |
|------------------|----------------|---------------|
| Davolio Nancy | 2024-05-01 | Experimentado |
| Fuller Andrew | 2024-08-14 | Experimentado |
| Leverling Janet | 2024-04-01 | Experimentado |
| Peacock Margaret | 2025-05-03 | Junior |
| Buchanan Steven | 2025-10-17 | Junior |
| Suyama Michael | 2025-10-17 | Junior |
| King Robert | 2026-01-02 | Junior |
| Callahan Laura | 2026-03-05 | Junior |
| Dodsworth Anne | 2026-11-15 | Junior |



Consultas avanzadas II

Consultas avanzadas

Bases de datos Empresa



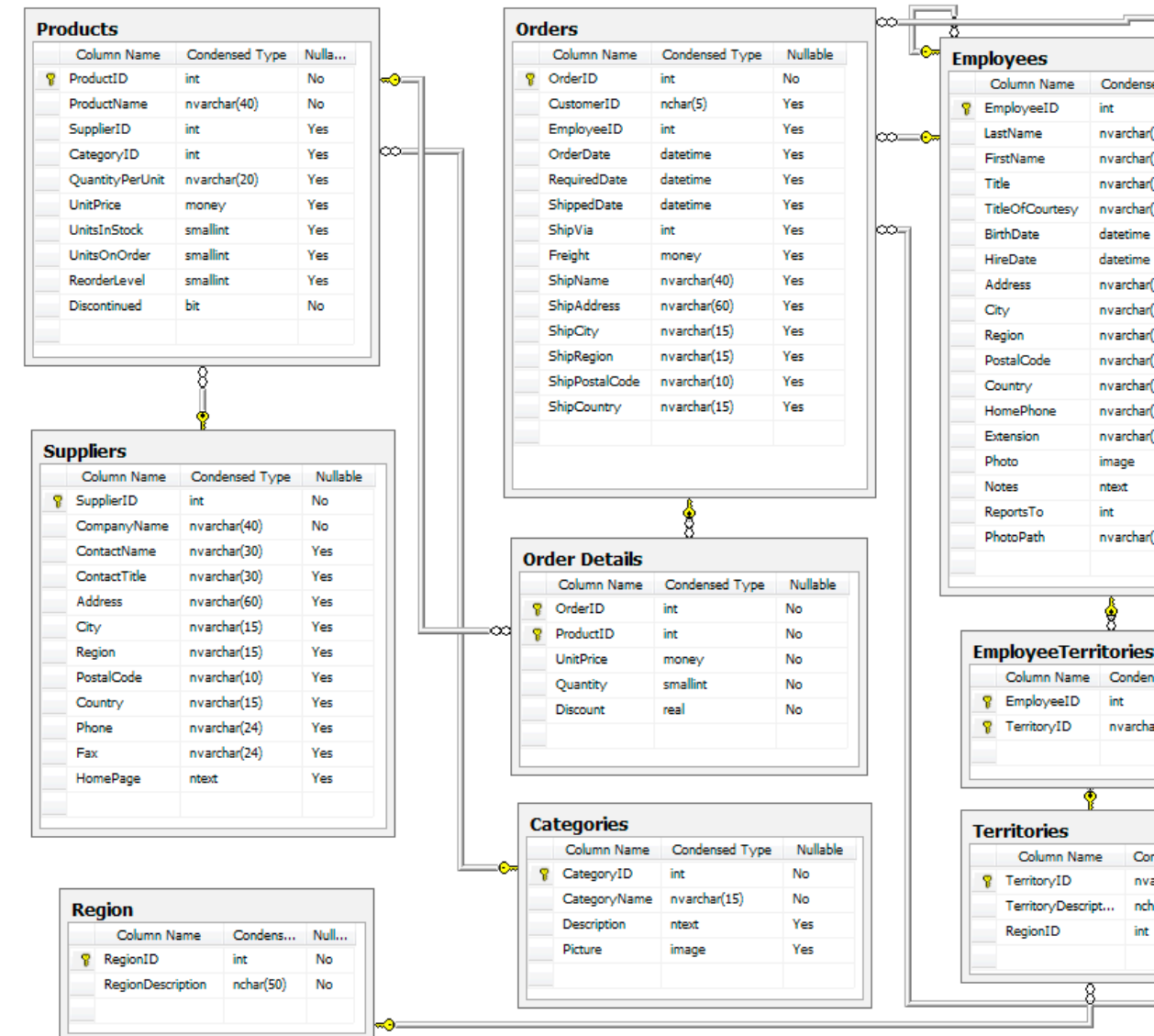
- La base de datos es para administrar clientes, pedidos, inventario, compras, proveedores, envíos y empleados de pequeñas empresas.

Ordenar resultados basado en una condición

- Ordenar los empleados por Ciudad si el cumpleaños es mayor a 1990 y por codigo postal en caso contrario.

```
SELECT FirstName "Nombre", LastName "Apellido",  
BirthDate "Fecha Nacimiento", City "Ciudad",  
PostalCode "Codigo Postal"  
FROM Employee  
ORDER BY  
CASE WHEN Birthdate > '1990-01-01'  
      THEN City  
      ELSE PostalCode  
END
```

| Nombre | Apellido | Fecha Nacimiento | Ciudad | Codigo Postal |
|----------|-----------|------------------|----------|---------------|
| Margaret | Peacock | 1969-09-19 | Redmond | 98052 |
| Nancy | Davolio | 1980-12-08 | Seattle | 98122 |
| Andrew | Fuller | 1984-02-19 | Tacoma | 98401 |
| Janet | Leverling | 1995-08-30 | Kirkland | 98033 |
| Michael | Suyama | 1995-07-02 | London | EC2 7JR |
| Robert | King | 1992-05-29 | London | RG1 9SP |
| Anne | Dodsworth | 1998-01-27 | London | WG2 7LT |
| Steven | Buchanan | 1987-03-04 | London | SW1 8JR |
| Laura | Callahan | 1990-01-09 | Seattle | 98105 |



Cual es la categoria de productos que vende más unidades?

- Cual es la categoria de productos que vende más unidades?

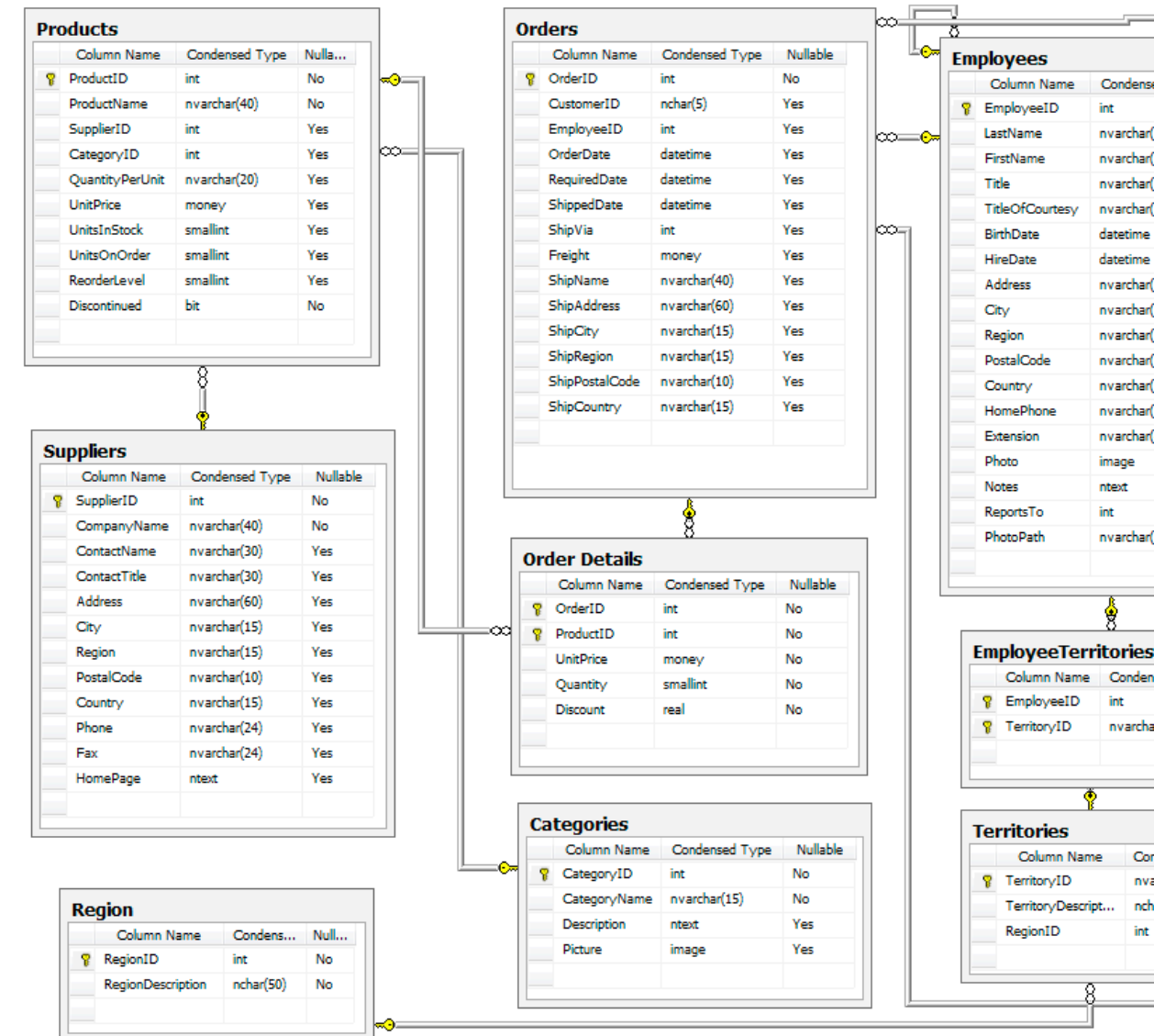
```
SELECT c.CategoryName "Categoria", SUM(o.Quantity) AS
"Total productos"
FROM OrderDetail o
JOIN Product p
ON o.ProductId = p.Id
JOIN Category c
ON p.Id = c.Id
GROUP BY 1
ORDER BY 2 Desc
```

| Categoria | Total productos |
|----------------|-----------------|
| Confections | 209901 |
| Condiments | 209418 |
| Grains/Cereals | 206906 |
| Meat/Poultry | 205879 |
| Produce | 205437 |
| Dairy Products | 204578 |
| Beverages | 203353 |
| Seafood | 203223 |

- Cual es la categoria de productos que produce más dinero?

```
SELECT c.CategoryName "Categoria",
SUM(o.Quantity) "Total Productos",
SUM(o.Quantity * o.UnitPrice) "Total Vendido"
FROM OrderDetail o
JOIN Product p
ON o.ProductId = p.Id
JOIN Category c
ON p.Id = c.Id
GROUP BY 1
ORDER BY 3 Desc
```

| Categoria | Total Productos | Total Vendido |
|----------------|-----------------|-------------------|
| Seafood | 203223 | 8127800 |
| Produce | 205437 | 6162684 |
| Meat/Poultry | 205879 | 5146795 |
| Dairy Products | 204578 | 4500174.8 |
| Grains/Cereals | 206906 | 4416881.949999999 |
| Condiments | 209418 | 3977418.2 |
| Beverages | 203353 | 3659727.6 |
| Confections | 209901 | 2098810 |

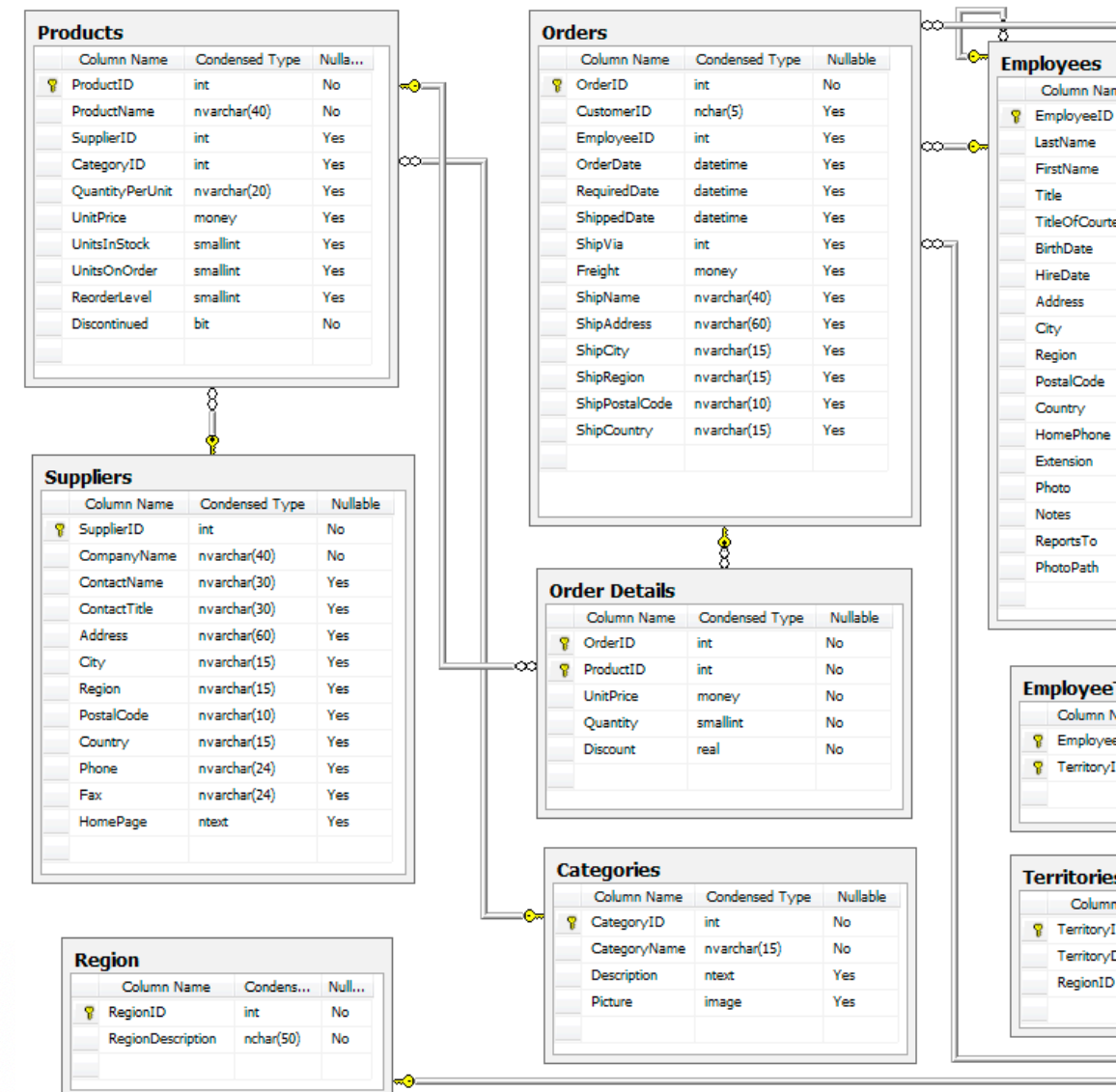


Determine los 10 productos más vendidos

- Determine los 10 productos más vendidos incluyendo precio promedio, descuento promedio y venta Total

```
SELECT p.Id "ID producto",
p.ProductName "Nombre",
c.CategoryName "Categoria",
    AVG(o.UnitPrice) AS "Precio Promedio",
    AVG(o.Discount) AS "Descuento Promedio",
    SUM(o.UnitPrice*o.Quantity*(1-o.Discount)) AS "Venta Total"
FROM Product p
JOIN OrderDetail o
    ON p.Id = o.ProductId
JOIN Category c
    ON p.CategoryId = c.Id
GROUP BY p.Id
ORDER BY "Venta Total" DESC limit 10;
```

| ID producto | Nombre | Categoria | Precio Promedio | Descuento Promedio | Venta Total |
|-------------|-------------------------|----------------|--------------------|------------------------|--------------------|
| 38 | Côte de Blaye | Beverages | 263.4479249011858 | 0.0001358695652173913 | 54009228.735 |
| 29 | Thüringer Rostbratwurst | Meat/Poultry | 123.7597387695445 | 0.000250244140625 | 25791923.041999985 |
| 9 | Mishi Kobe Niku | Meat/Poultry | 96.9975924547034 | 6.205013651030033e-05 | 19899792.5 |
| 20 | Sir Rodney's Marmalade | Confections | 80.98998763906057 | 9.147095179233625e-05 | 16816617.36 |
| 18 | Carnarvon Tigers | Seafood | 62.49071667285555 | 0.0002661220448075257 | 12940859.375 |
| 59 | Raclette Courdavault | Dairy Products | 54.97427058968361 | 0.00031392342730518274 | 11358970.7 |
| 51 | Manjimup Dried Apples | Produce | 52.988335982393934 | 0.00024452867098667317 | 11083309.65 |
| 62 | Tarte au sucre | Confections | 49.28288043477547 | 0.00032114624505928847 | 10185139.069999997 |
| 43 | Ipoh Coffee | Beverages | 45.9898180029513 | 0.00017215937038858828 | 9548562.7 |
| 28 | Rössle Sauerkraut | Produce | 45.58516681135679 | 0.0001550291454793501 | 9334526.240000056 |

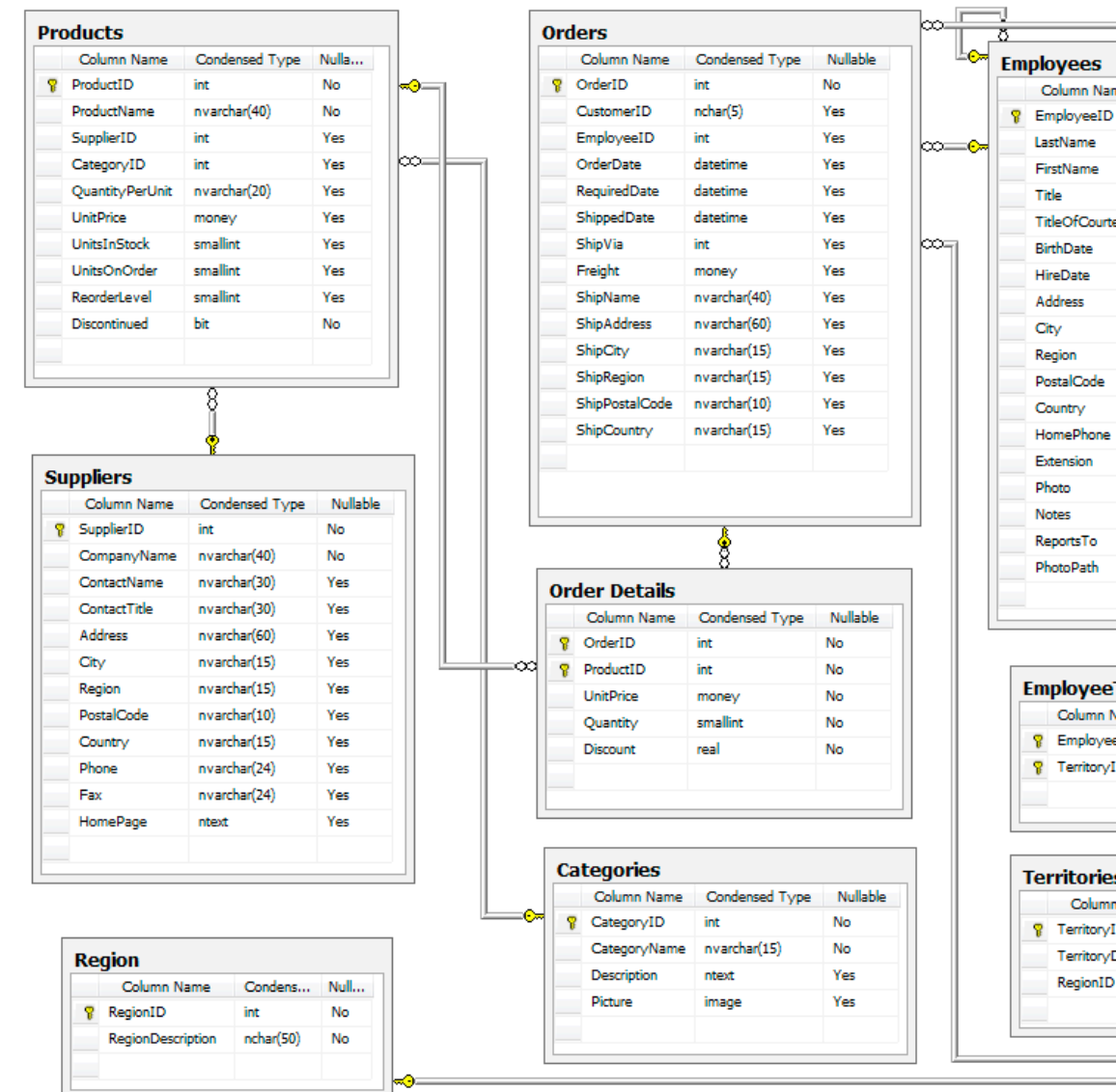


Cuales son los productos más caros por proveedor?

- Cuales son los productos más caros por proveedor?

```
SELECT s.companyname "Nombre Proveedor",  
       p.productname "Nombre Producto",  
       p.unitprice "Precio Unidad"  
FROM supplier as s  
JOIN product as p ON s.id = p.supplierid  
WHERE p.unitprice = (SELECT MAX(p2.unitprice) FROM product  
as p2 WHERE p2.supplierid = s.id)  
ORDER BY p.unitprice DESC, s.companyname ASC, p.productname  
ASC  
limit 10
```

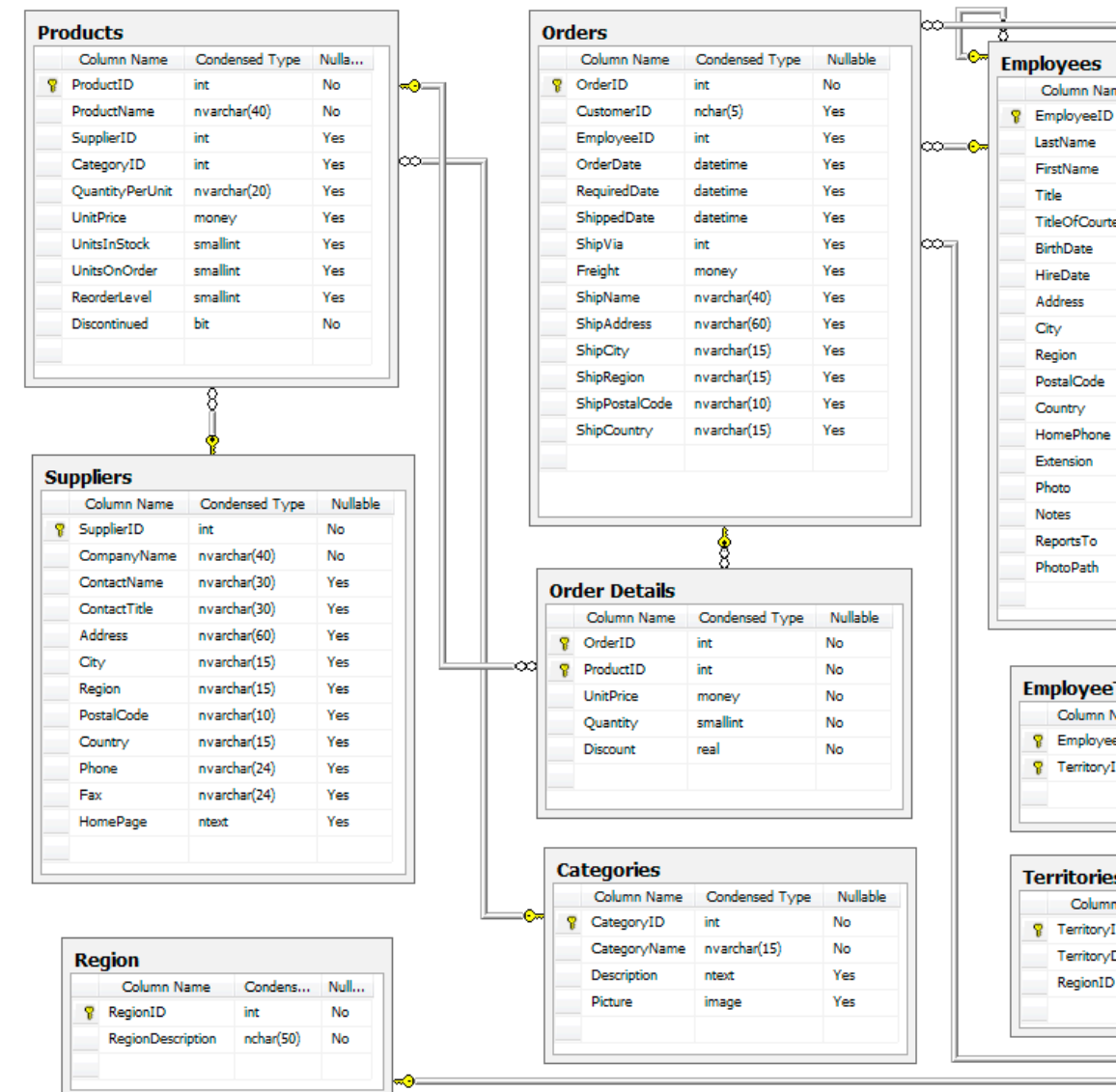
| Nombre Proveedor | Nombre Producto | Precio Unidad |
|-----------------------------------|-------------------------|---------------|
| Aux joyeux ecclésiastiques | Côte de Blaye | 263.5 |
| Plutzer Lebensmittelgroßmärkte AG | Thüringer Rostbratwurst | 123.79 |
| Tokyo Traders | Mishi Kobe Niku | 97 |
| Specialty Biscuits, Ltd. | Sir Rodney's Marmalade | 81 |
| Pavlova, Ltd. | Carnarvon Tigers | 62.5 |
| Gai pâturage | Raclette Courdavault | 55 |
| G'day, Mate | Manjimup Dried Apples | 53 |
| Forêts d'érables | Tarte au sucre | 49.3 |
| Leka Trading | Ipoh Coffee | 46 |
| Heli Süßwaren GmbH & Co. KG | Schoggi Schokolade | 43.9 |



Que realiza la siguiente consulta?

• ?

```
select t.*, t."Venta Total"-t."Venta Descuento Total" as  
"Descuento Total" from (  
select o.ProductID "ID producto",  
sum(Quantity) "Unidades Total",  
sum(Quantity*UnitPrice) "Venta Total",  
sum(Quantity*UnitPrice*(1-Discunt)) "Venta Descuento  
Total" from OrderDetail as o  
Group by o.ProductId  
ORDER by 4 DESC ) as t  
limit 10;
```



Practicar en GoogleColab!!!

The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'The SQLite Query Optimizer', 'SQL_IV_chinook_db.ipynb', 'Home / Twitter', 'join sqlite - Google Search', and 'SQLite: Joins'. The notebook's address bar shows a Google Drive link. The notebook title is 'SQL_IV_chinook_db.ipynb' with a star icon. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', with a status 'All changes saved'. On the right, there are 'Comment', 'Share', and a user profile icon. Below the menu, there are '+ Code' and '+ Text' buttons, and a status bar showing 'RAM' and 'Disk' usage, and 'Editing' mode.

The notebook content includes:

- A small table with one row:

| Name |
|---------|
| 18 rows |
- Text: "Generated by SchemaSpy"
- Text: "In the above ER diagram, the tiny vertical key icon indicates a column is a primary key. A primary key is a column (or set of columns) whose values uniquely identify every row in a table. For example, `Employeeid` is the primary key in the `Employee` table."
- Text: "The relationship icon indicates a foreign key constraint and a one-to-many relationship."
- Section: "SQL con Chinook"
- Section: "Joins"
- Code cell 1:

```
1 # CROSS Join example
2 %%sql
3 select * from album CROSS JOIN Artist limit 5;
```
- Output for Code cell 1:

```
* sqlite:///content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
AlbumId      Title      ArtistId ArtistId_1  Name
1            For Those About To Rock We Salute You 1      1      AC/DC
1            For Those About To Rock We Salute You 1      2      Accept
1            For Those About To Rock We Salute You 1      3      Aerosmith
1            For Those About To Rock We Salute You 1      4      Alanis Morissette
1            For Those About To Rock We Salute You 1      5      Alice In Chains
```
- Code cell 2:

```
[ ] 1 #INNER Join
2 %%sql
3 select * from album JOIN Artist ON album.artistId=Artist.artistId limit 5;
```
- Output for Code cell 2:

```
* sqlite:///content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
AlbumId      Title      ArtistId ArtistId_1  Name
1            For Those About To Rock We Salute You 1      1      AC/DC
2            Balls to the Wall                2      2      Accept
3            Restless and Wild                  2      2      Accept
4            Let There Be Rock                      1      1      AC/DC
5            Big Ones                          3      3      Aerosmith
```
- Code cell 3:

```
[ ] 1 #SELECT ... FROM alumno JOIN curso USING (aid,...)
2 %%sql
```

The bottom status bar shows a green checkmark, '0s', and 'completed at 9:43 AM'.

Consultas?

Consultas o comentarios?

Muchas gracias