

Lenguaje SQL V

Alex Di Genova

07/05/2022

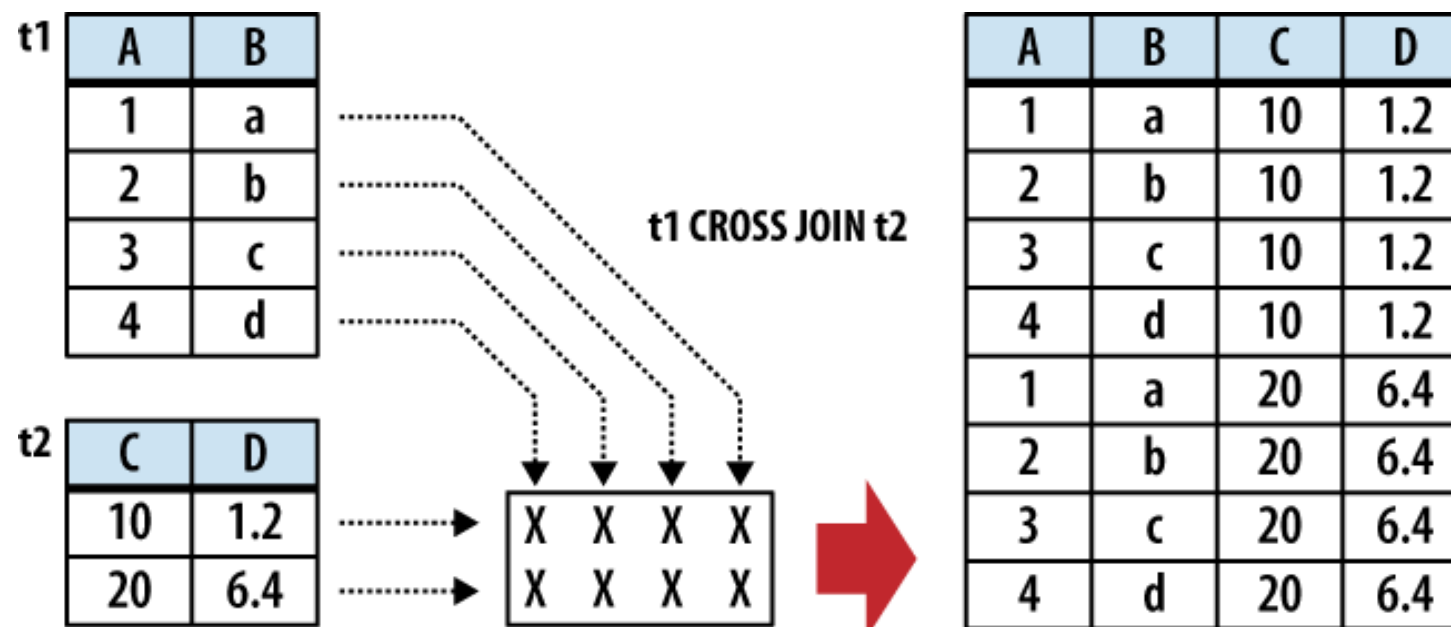
Contenidos

- Repaso
- Consultas avanzadas

SQL

CROSS JOIN

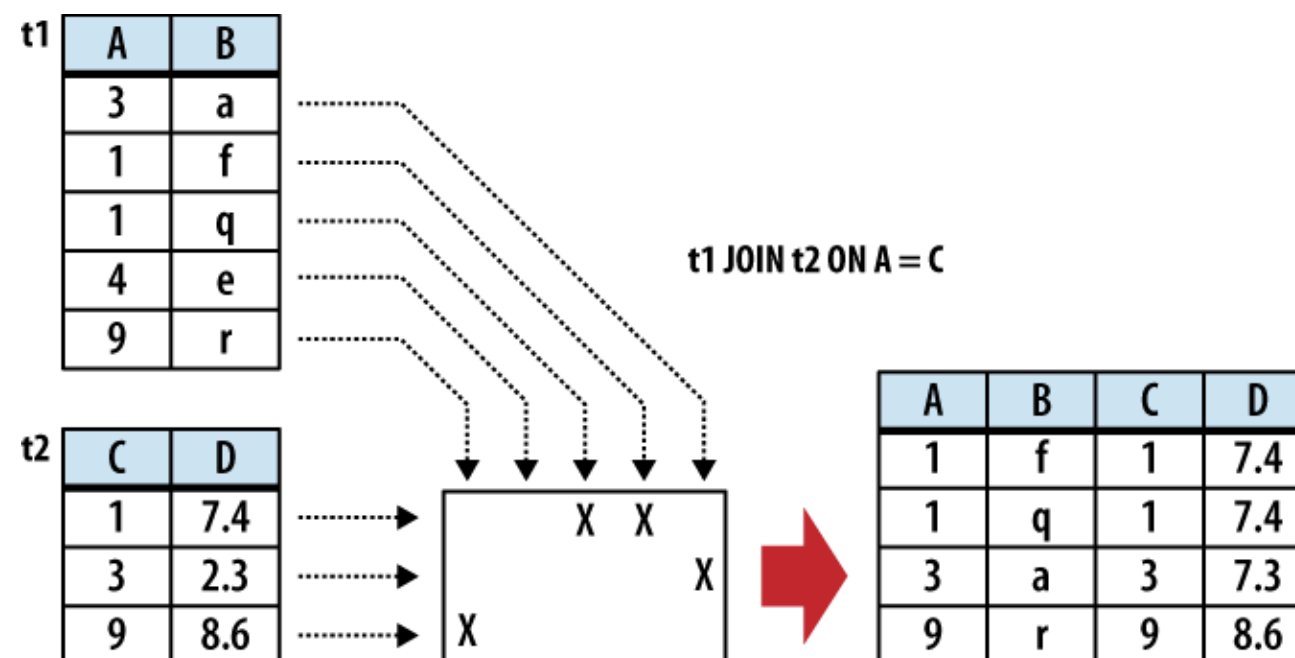
- **SELECT ... FROM t1 CROSS JOIN t2 ...**
- Hace coincidir cada fila de la primera tabla con cada fila de la segunda tabla.
- Si las tablas de entrada tienen x e y columnas, la tabla resultante tendrá x+y columnas.
- Si las tablas de entrada tienen n y m filas, la tabla resultante tendrá n x m filas (producto cartesiano)



SQL

INNER JOIN

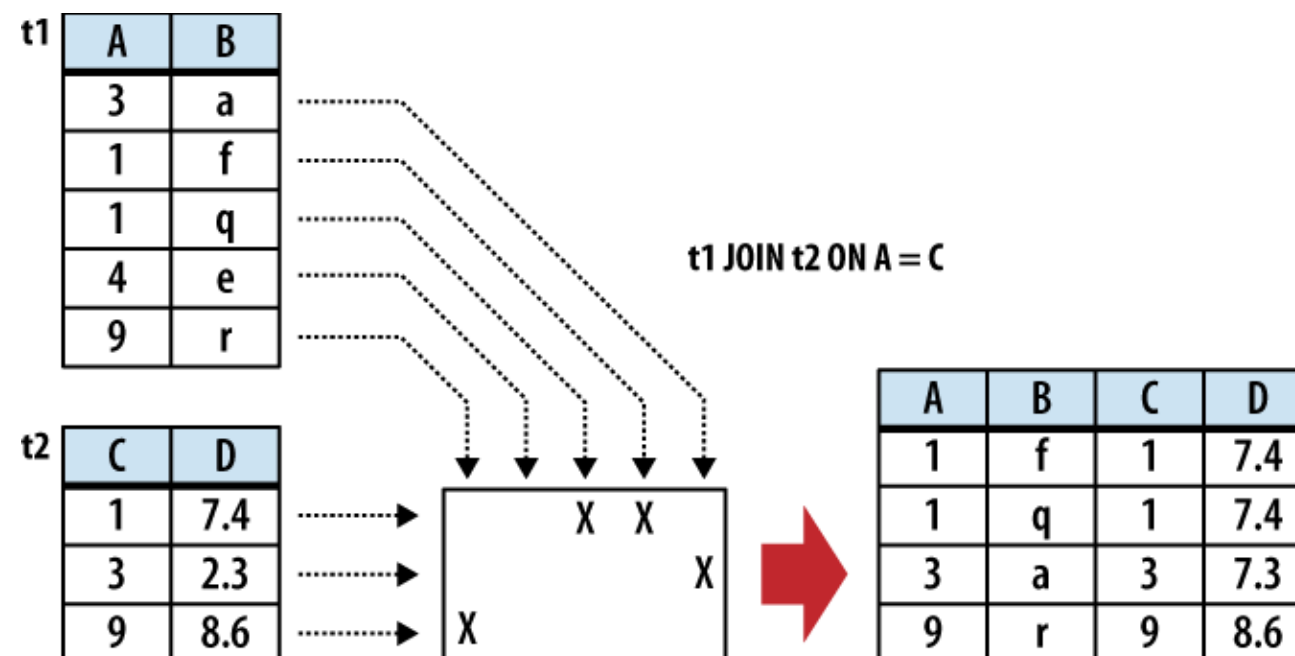
- **SELECT ... FROM t1 JOIN t2 ON expresion_de_condicion...**
- La expresion condición es usualmente utilizada para comparar dos columnas de dos tablas diferentes.
- Si las tablas de entrada tienen x e y columnas, la tabla resultante tendrá no más x+y columnas.
- Si las tablas de entrada tienen n y m filas, la tabla resultante tendrá [0,n x m] filas



SQL

INNER JOIN

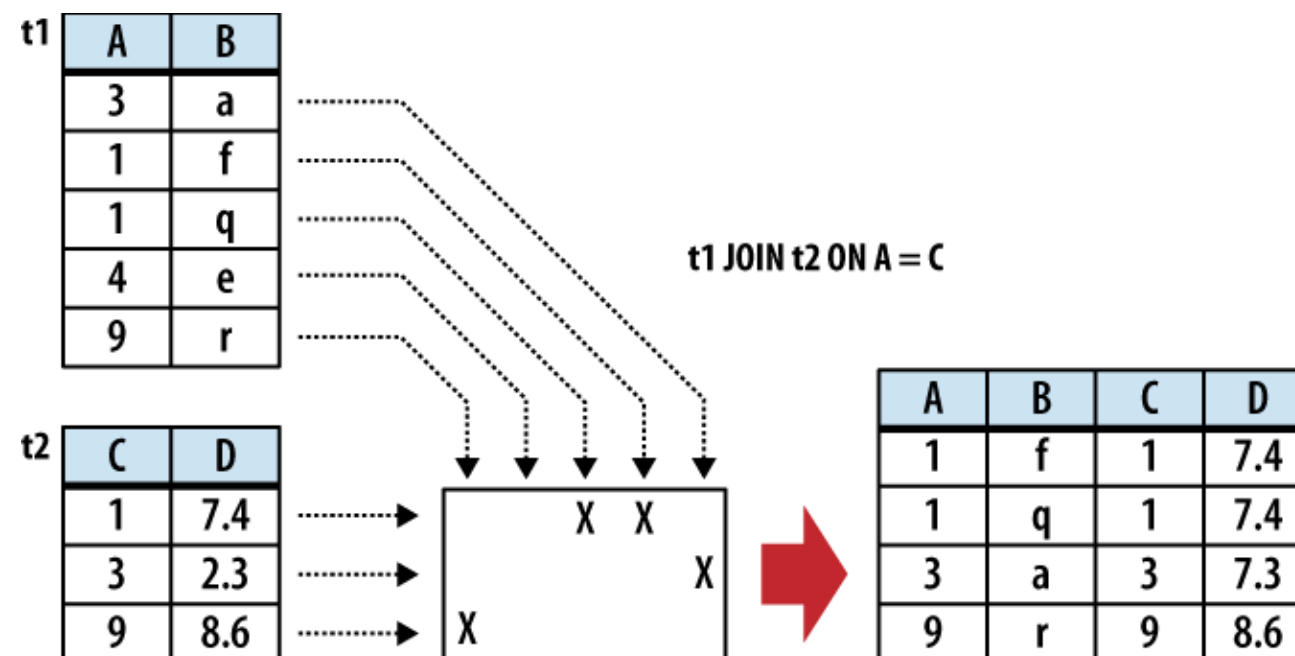
- **SELECT ... FROM t1 INNER JOIN t2 ON expresion_de_condicion...**
- **SELECT ... FROM t1 JOIN t2 ON expresion_de_condicion...**
- La expresion condición es usualmente utilizada para comparar dos columnas de dos tablas diferentes.
- Si las tablas de entrada tienen x e y columnas, la tabla resultante tendrá no más x+y columnas.
- Si las tablas de entrada tienen n y m filas, la tabla resultante tendrá [0,n x m] filas
- **SELECT ... FROM alumno JOIN curso ON alumno.id=curso.alum_id...**



SQL

INNER JOIN

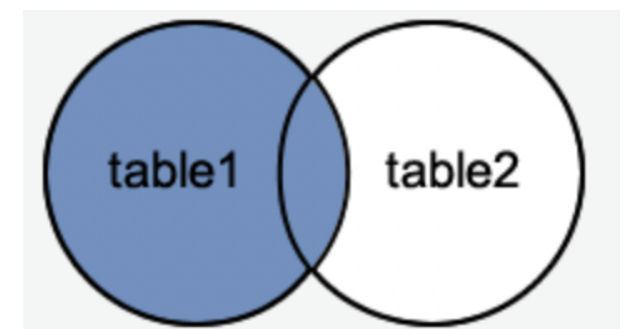
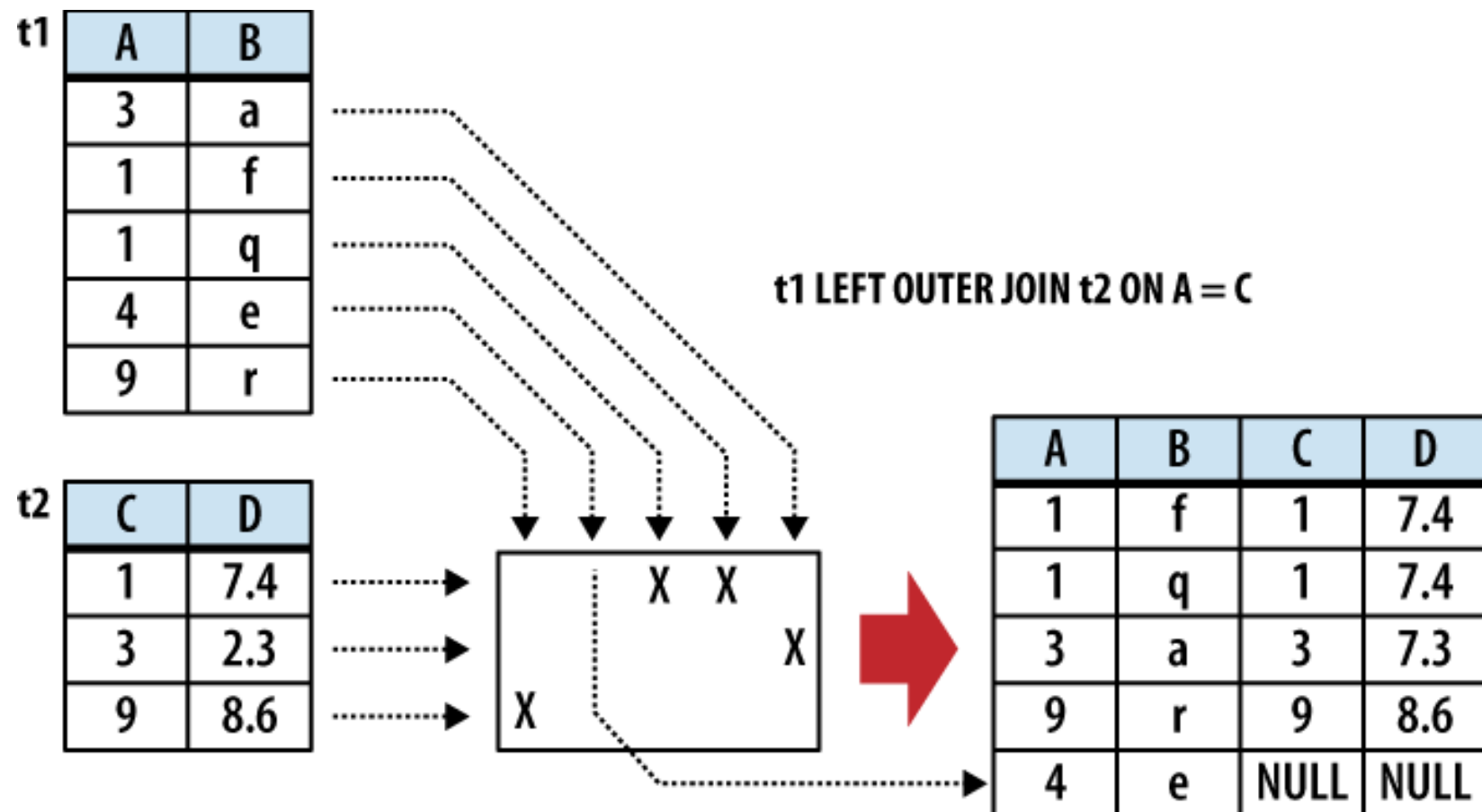
- **SELECT ... FROM** alumno **JOIN** curso **ON** alumno.id=curso.alum_id...
- **SELECT ... FROM** alumno **JOIN** curso **USING** (aid,...)
 - El atributo debe existir en ambas tablas
- **SELECT ... FROM** alumno **NATURAL JOIN** curso
 - Join automatico en aid



SQL

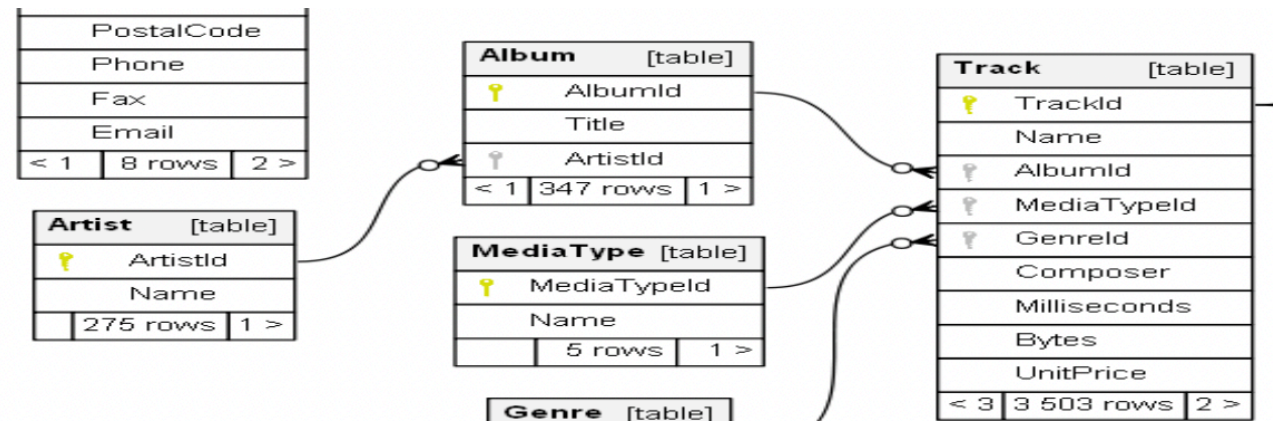
OUTER JOIN

- **SELECT ... FROM** alumno **LEFT OUTER JOIN** curso **ON** alumno.id=curso.alum_id...
- **Extension del INNER JOIN**
- **OUTER JOINS = LEFT, RIGHT, FULL (SQLite2 solo LEFT)**
- **Condiciones con ON, USING, NATURAL**



SQL

Multiples JOIN (JOIN & JOIN & JOIN...)



```
select * from album JOIN Artist USING (artistId) limit 5;
```

```
select * from album JOIN Artist USING (artistId) JOIN Track USING (albumid) limit 5;
```

```
[12] 1 %%sql
      2 select * from album JOIN Artist USING (artistId) limit 5;
```

```
* sqlite:///content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
```

AlbumId	Title	ArtistId	Name
1	For Those About To Rock We Salute You	1	AC/DC
2	Balls to the Wall	2	Accept
3	Restless and Wild	2	Accept
4	Let There Be Rock	1	AC/DC
5	Big Ones	3	Aerosmith

```
1 %%sql
2 select * from album JOIN Artist USING (artistId) JOIN Track USING (albumid) limit 5;
```

```
* sqlite:///content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
```

AlbumId	Title	ArtistId	Name	TrackId	Name_1	MediaTypeId	GenreId	Composer	Milliseconds
1	For Those About To Rock We Salute You	1	AC/DC	1	For Those About To Rock (We Salute You)	1	1	Angus Young, Malcolm Young, Brian Johnson	343719
2	Balls to the Wall	2	Accept	2	Balls to the Wall	2	1	None	342562
3	Restless and Wild	2	Accept	3	Fast As a Shark	2	1	F. Baltes, S. Kaufman, U. Dirkschneider & W. Hoffman	230619
3	Restless and Wild	2	Accept	4	Restless and Wild	2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirkschneider & W. Hoffman	252051
3	Restless and Wild	2	Accept	5	Princess of the Dawn	2	1	Deaffy & R.A. Smith-Diesel	375418

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*

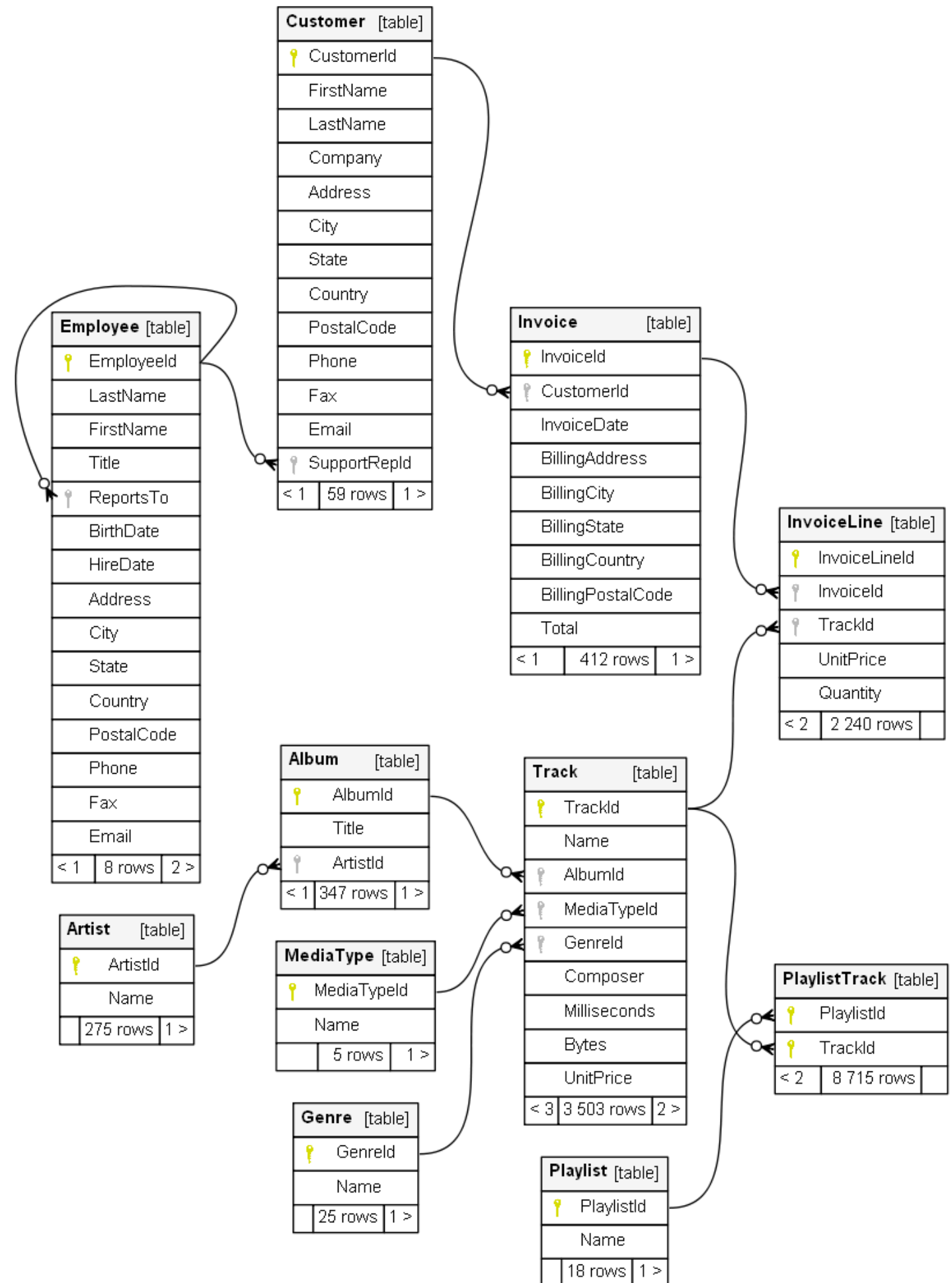
LIMIT *contador*

OFFSET indice

Consultas avanzadas

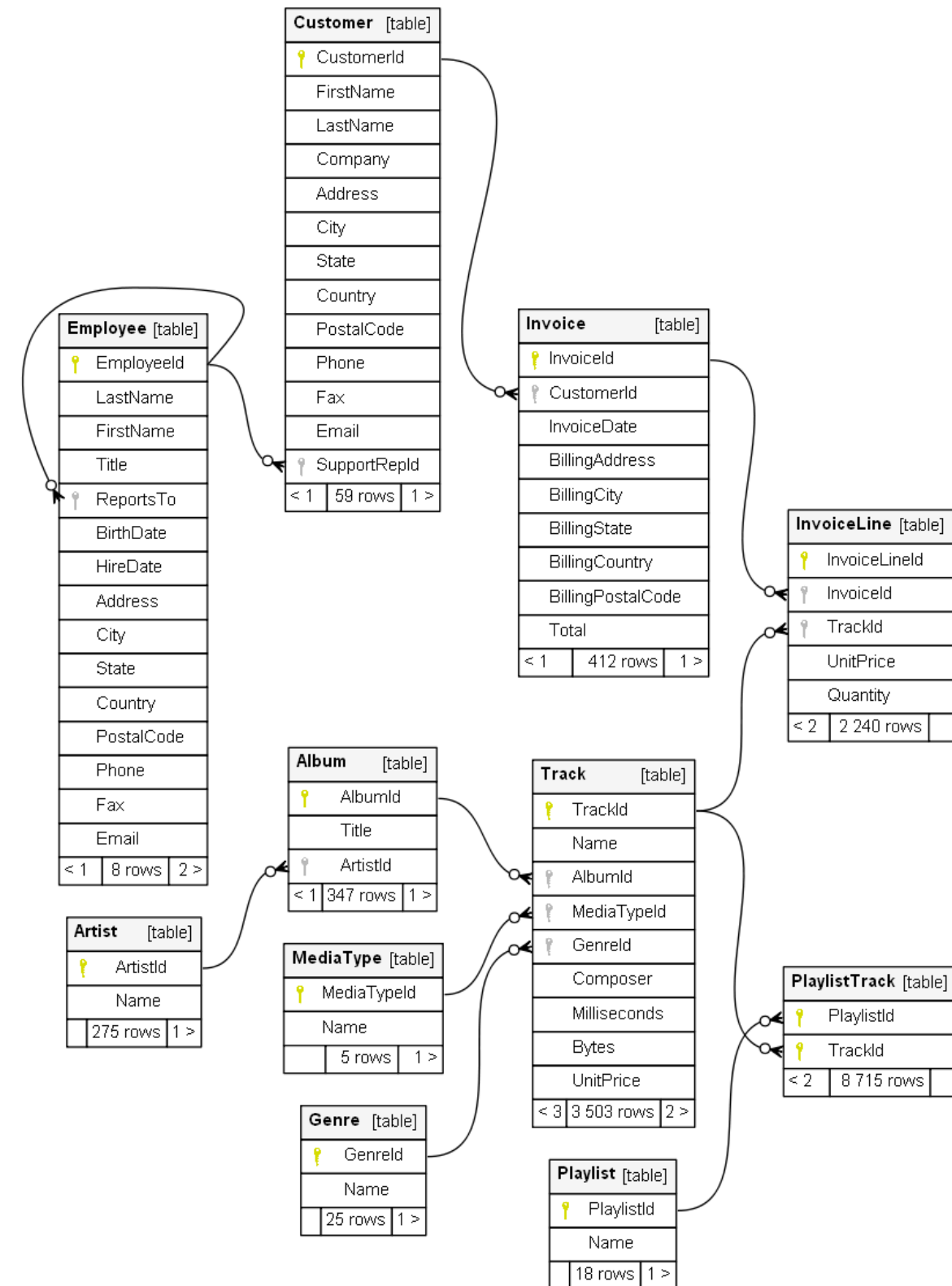
Consultas avanzadas

Chinook database



Cuales son los 5 artistas que venden más canciones?

```
SELECT a.Name 'Artista', sum(li.UnitPrice) 'Total vendido'
FROM InvoiceLine li, Track t, Album al, Artist a
WHERE li.TrackId = t.Trackid
      and al.AlbumId = t.AlbumId
      and a.ArtistId = al.ArtistId
GROUP BY a.Name
ORDER BY COUNT(a.ArtistId) DESC
limit 5;
```

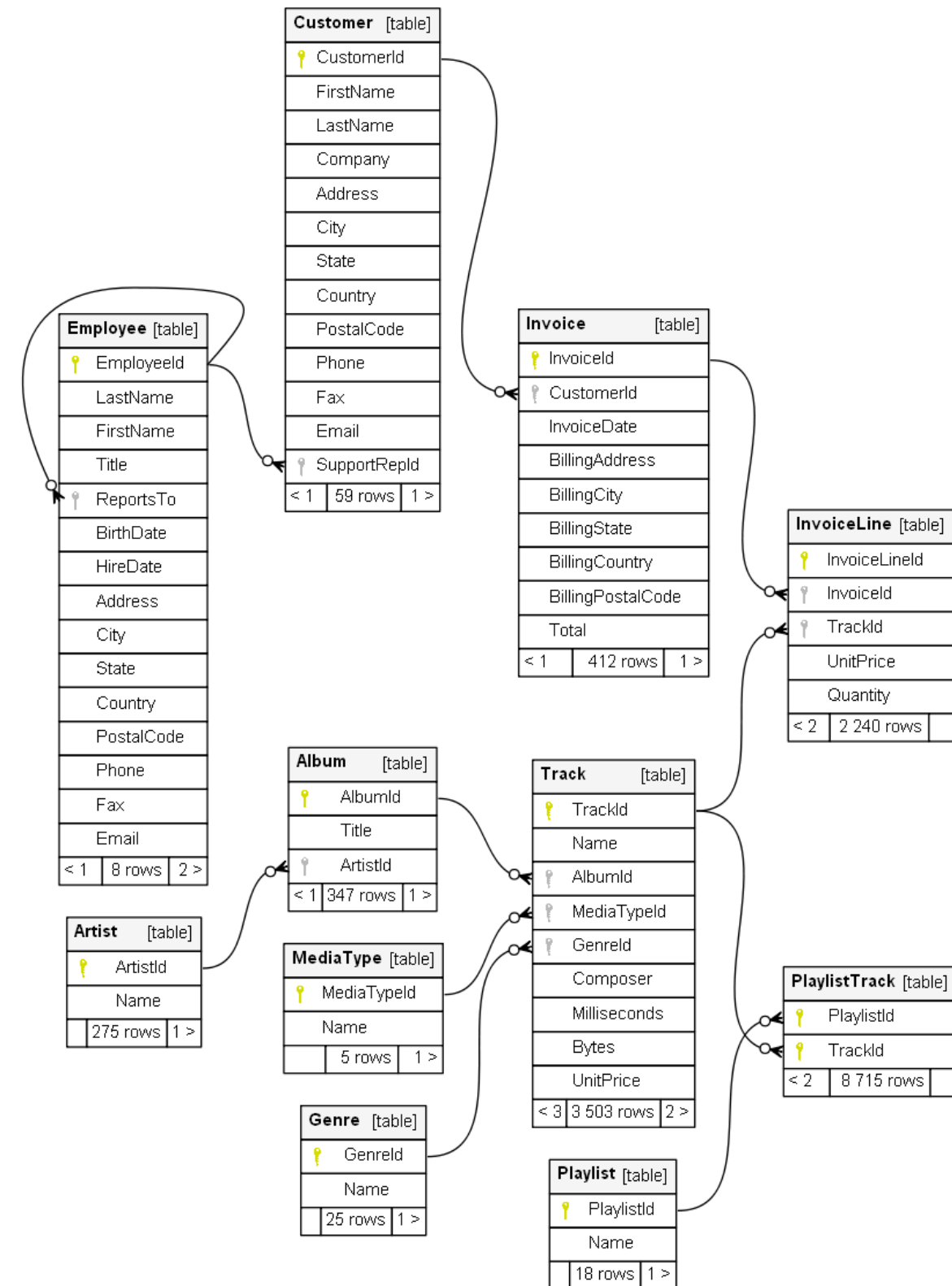


Cuales son las canciones más compradas del año 2012?

```
SELECT a.Name 'Artista', sum(li.UnitPrice) 'Total vendido'
FROM InvoiceLine li, Track t, Album al, Artist a
WHERE li.TrackId = t.Trackid
      and al.AlbumId = t.AlbumId
      and a.ArtistId = al.ArtistId
GROUP BY a.Name
ORDER BY COUNT(a.ArtistId) DESC
limit 5;
```

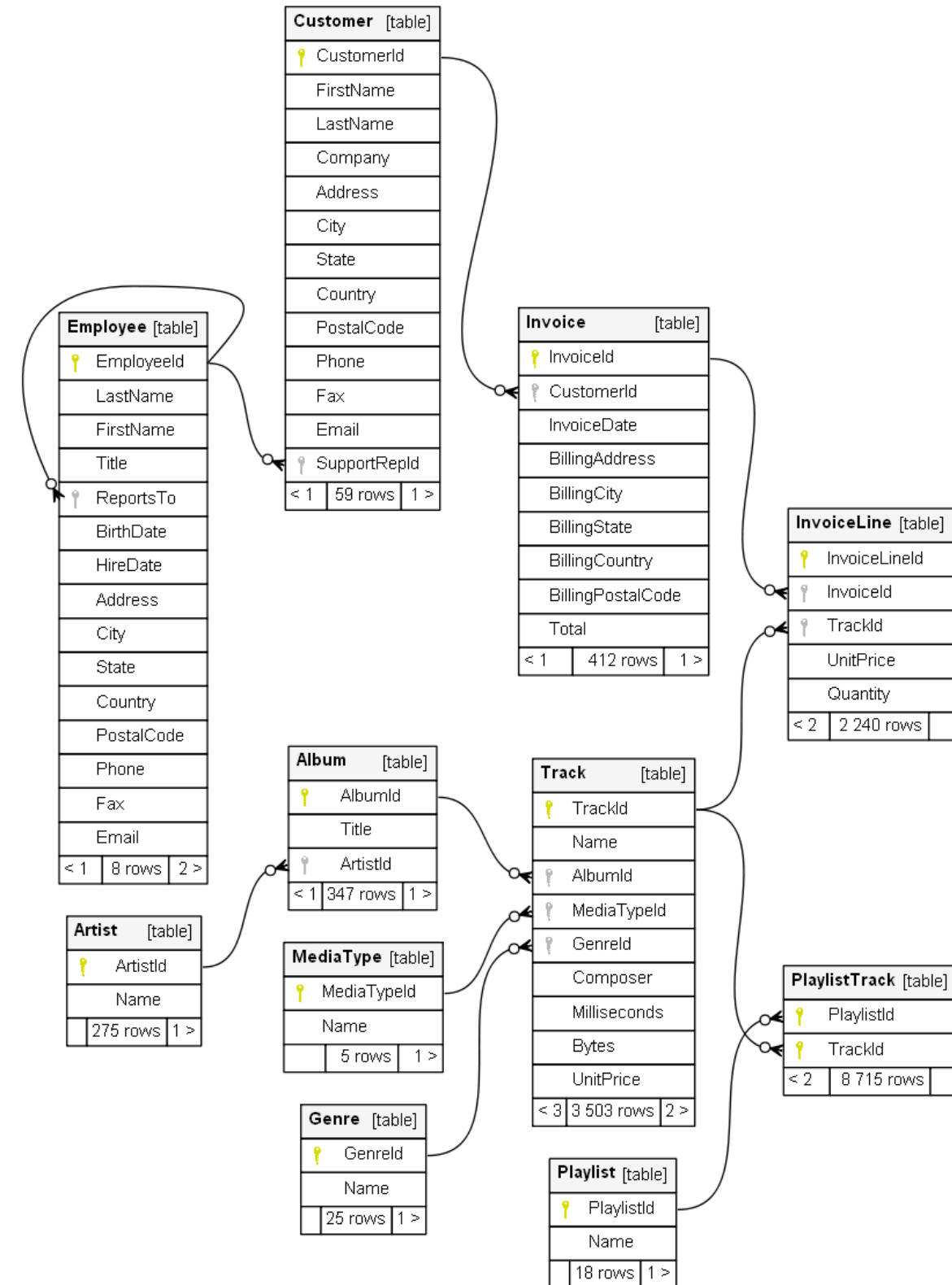
Usando Join?

```
SELECT a.Name 'Artista', sum(li.UnitPrice) 'Total vendido'
FROM InvoiceLine as li
join Track USING (TrackId)
join Album USING (AlbumId)
join Artist as a USING (ArtistId)
GROUP BY a.Name
ORDER BY COUNT(a.ArtistId) DESC
limit 5;
```



Cuales son los tipos de medio más vendido?

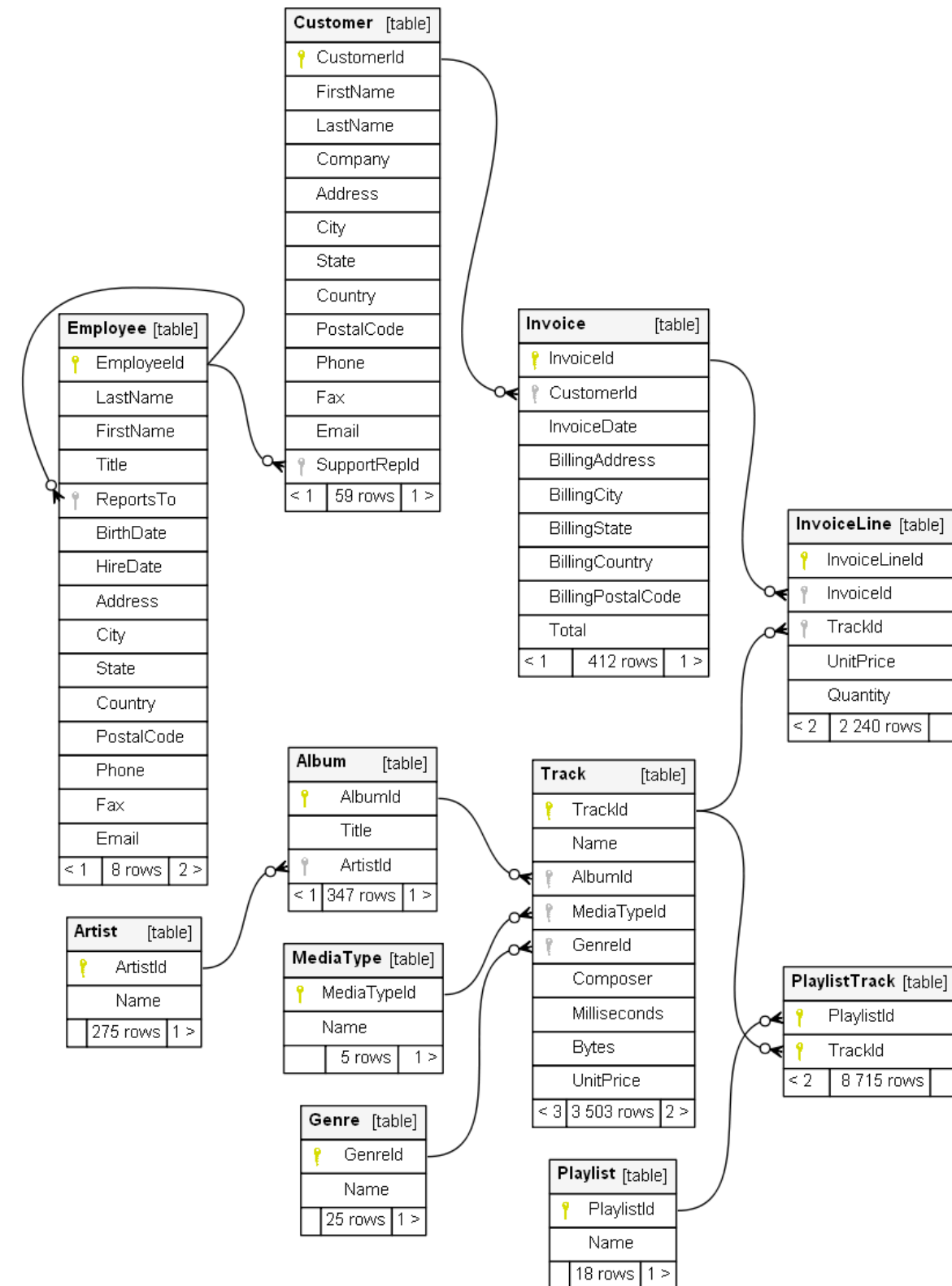
```
SELECT m.Name AS 'Tipo de Medio', COUNT(m.MediaTypeId)
'Total'
FROM InvoiceLine li, Track t, MediaType m
WHERE li.TrackId = t.TrackId
      and m.MediaTypeId = t.MediaTypeId
GROUP BY m.Name
ORDER BY m.MediaTypeId;
```



mejor vendedor del 2009?

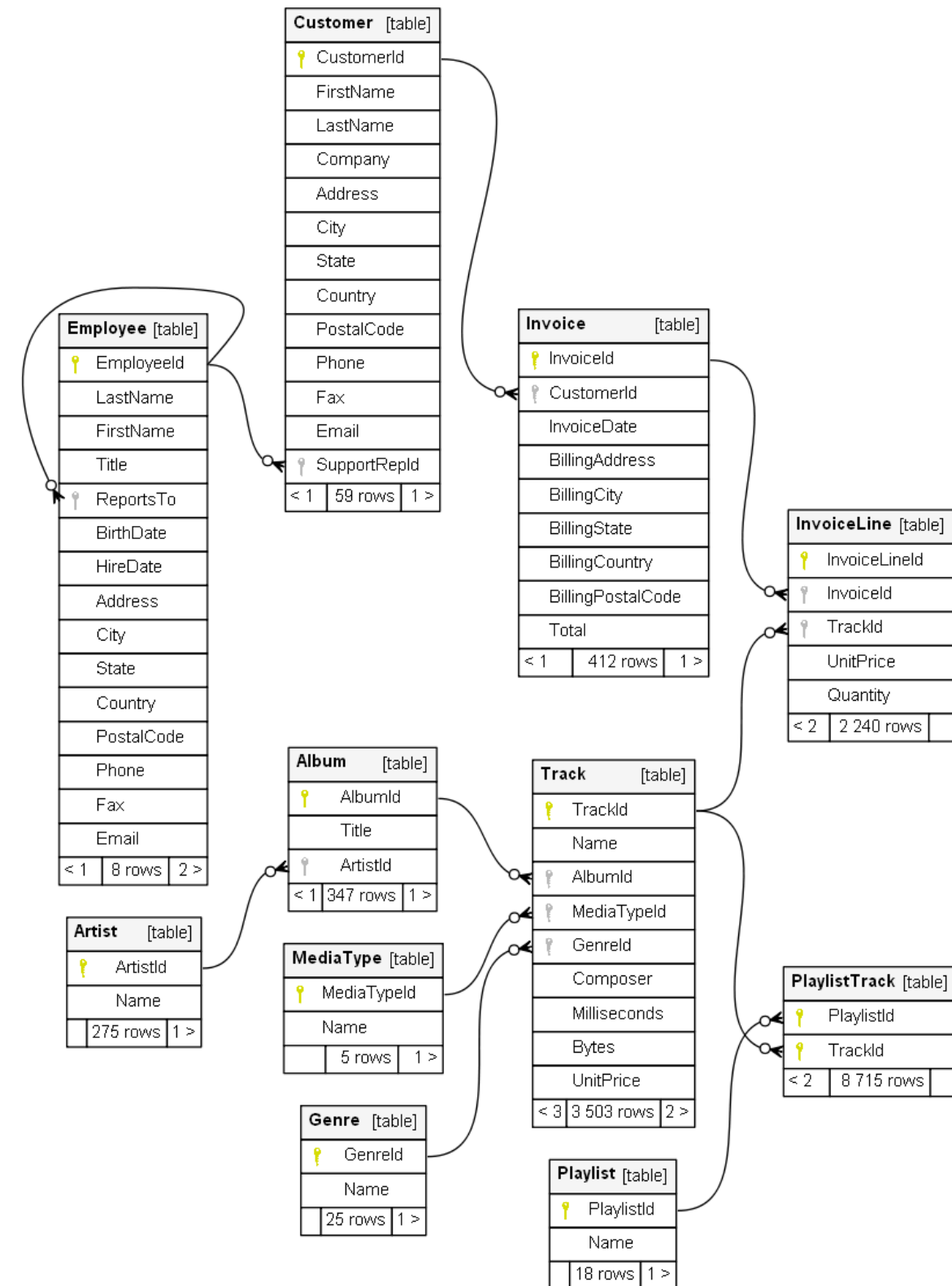
```
select e.firstName || " " || e.lastName as 'Mejor Vendedor 2009',
sum(i.total) as Total
from Invoice i
join customer c on c.customerid = i.customerid
join employee e on e.employeeid = c.supportrepid
where i.invoiceDate like '2009%'
group by e.Employeeid
order by sum(i.total) desc
limit 1
```

```
select "Mejor Vendedor 2009", max("Total") from (
select e.firstName || " " || e.lastName as "Mejor Vendedor 2009",
sum(i.total) as "Total" from Invoice i
join customer c on c.customerid = i.customerid
join employee e on e.employeeid = c.supportrepid
where i.invoiceDate like '2009%'
group by e.Employeeid
)
```



Muestre todas las canciones pero sin el ID. El resultado debe incluir el nombre del Album, el tipo de Medio y el Genero

```
select t.name as 'Nombre Cancion', t.composer, t.unitprice,  
a.title as 'Nombre Album', m.name as 'Tipo de Medio',  
g.name as 'Nombre de Genero' from Track t  
join album a on a.albumid = t.albumid  
join genre g on g.genreid = t.genreid  
join MediaType m on m.mediatypeid = t.mediatypeid  
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 the URL: `colab.research.google.com/drive/1hyhjRD2swHrT4y3acApBX6xXxCorLZ5b#scrollTo=uBTPUc-qyoCm`. The notebook title is 'SQL_IV_chinook_db.ipynb' with a star icon. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and 'All changes saved'. The left sidebar has icons for file explorer, search, and a table icon. The main area shows a table with 18 rows and 1 column, titled 'Name', generated by SchemaSpy. Below this, there is a text block explaining primary and foreign key constraints. The notebook content includes two sections: 'SQL con Chinook' and 'Joins'. The 'Joins' section contains two code blocks. The first code block shows a CROSS JOIN query between 'album' and 'Artist' tables, limited to 5 results. The second code block shows an INNER JOIN query between 'album' and 'Artist' tables, limited to 5 results. Both queries are executed successfully, as indicated by the 'Done.' status and the resulting tables.

Generated by SchemaSpy

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.

The relationship icon indicates a foreign key constraint and a one-to-many relationship.

SQL con Chinook

Joins

```
1 # CROSS Join example
2 %%sql
3 select * from album CROSS JOIN Artist limit 5;
```

* 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	2	2	Accept
1	For Those About To Rock We Salute You	3	3	Aerosmith
1	For Those About To Rock We Salute You	4	4	Alanis Morissette
1	For Those About To Rock We Salute You	5	5	Alice In Chains

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

* 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

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

0s completed at 9:43 AM

Consultas?

Consultas o comentarios?

Muchas gracias