Universidad de O'Higgins

ICI INSTITUTO DE CIENCIAS DE LA INGENIERÍA

Vistas, indexación y optimización de consultas SQL.

Alex Di Genova

23/06/2022

Contenidos

- Notas y revisión control II
- Vistas
- Indexación
- Optimización de consultas SQL

Notas y revisión control

||

Revisión control II

Respuestas

1. **Base de datos Chinook**: Escriba una consulta qu estre los distintos tipos de empleados que trabajan en Canadá. Hint: Title. (**2 puntos**)

```
select distinct(Title) from Employee where country == "Canada";
```

2. **Base de datos Chinook**: Explique que tabla produce el siguiente código SQL (**3 punto**)

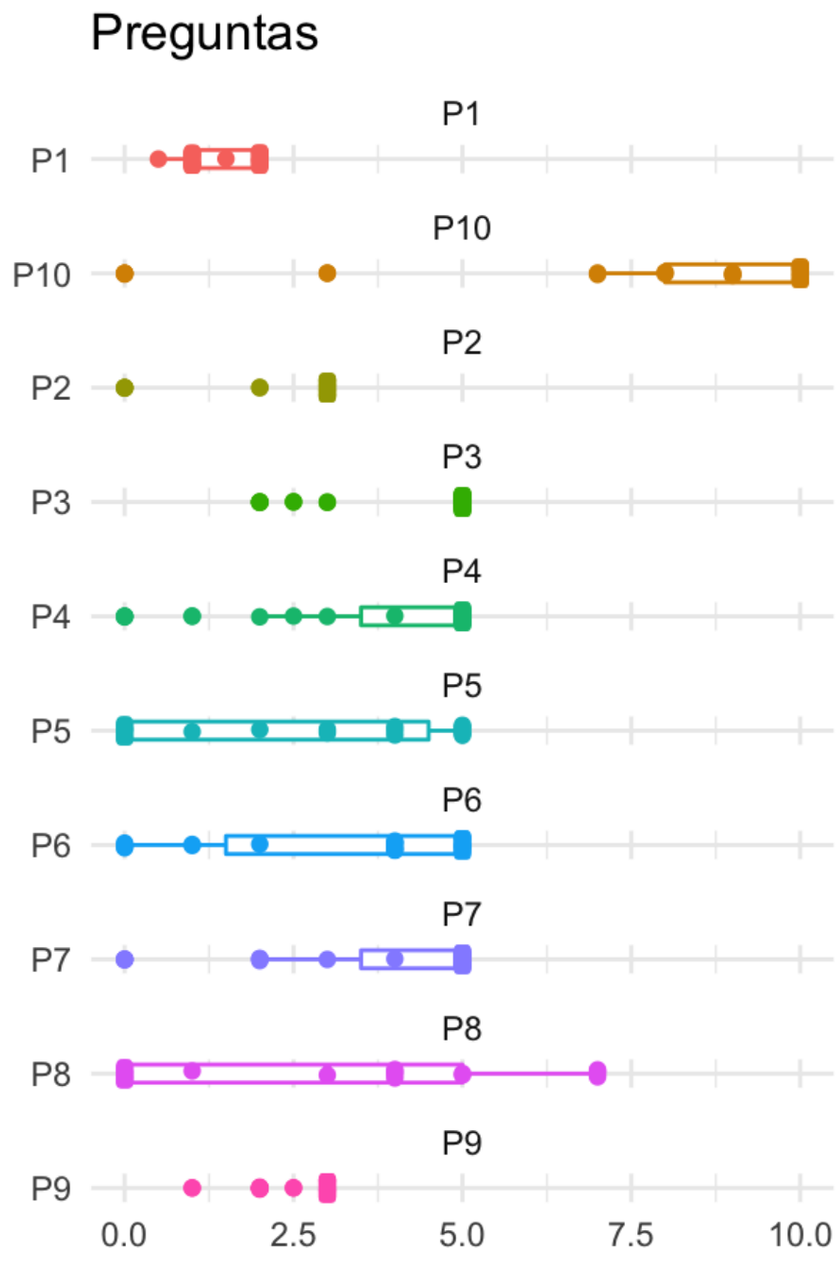
```
select i.billingcountry, sum(total) as 'TotalSales'
from invoice as i
group by billingcountry
order by TotalSales desc
```

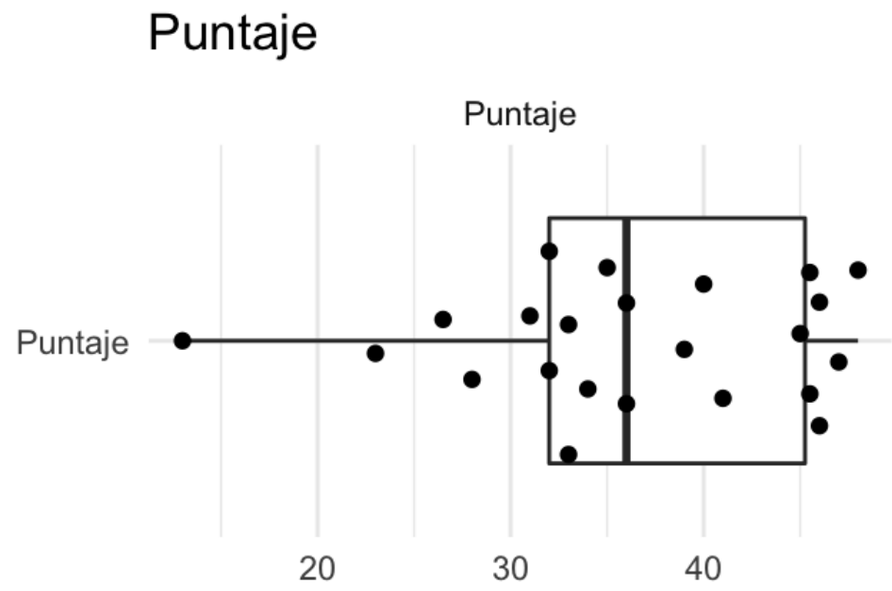
Muestra las ventas totales de canciones por país.

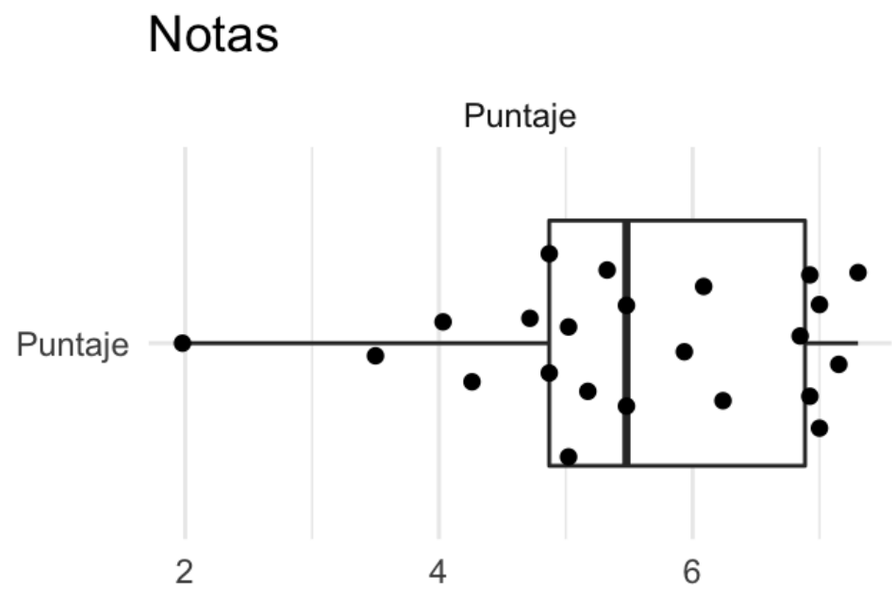
BillingCountry	TotalSales
USA	523.0600000000003
Canada	303.9599999999999
France	195.09999999999994
Brazil	190.09999999999997
Germany	156.48
United Kingdom	112.85999999999999
Czech Republic	90.24000000000001
Portugal	77.23999999999998
India	75.25999999999999
Chile	46.62

Notas

Control II







- Nota = Puntaje/46 * 7
- Promedio = 5,5
- En promedio incremento de 0.6 Decimas.

Vistas

Vistas

Consultas con nombre

- Las vistas proporcionan una forma de empaquetar consultas en un objeto predefinido.
- Una vez creadas son similares a tablas de solo lectura.
    - CREATE VIEW nombre_vista AS SELECT consulta
- Las vistas son dinamicas, cada vez que son invocadas se ejecuta el SELECT otra vez
- Alternativamente son llamadas consultas con nombre.
- Usos:
    - Para almacenar consultas frecuentes o complejas.
    - Para crear versiones de tablas mas amigables al usuario (tiempo y fechas).
- DROP VIEW nombre_vista; # que pasa con los datos?

Vista de un Join anidado

Ejemplo

- Crear una vista de un join anidado

```
create view join_album_artista_track as select
* from artist natural join album join track
using (albumid)
```

- Como desplegamos los elementos de la lista?

```
select * from join_album_artista_track limit
10;
```

- Como eliminamos la vista?

```
drop view join_album_artista_track;
```

```
[7]  1 %%sql
     2 create view join_album_artista_track as select * from artist natural join album join track using (albumid)
```

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

```
[8]  1 %%sql
     2 select * from join_album_artista_track limit 10;
```

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

ArtistId	Name	AlbumId	Title	TrackId	Name:1	MediaTypeId	GenreId	Composer
1	AC/DC	1	For Those About To Rock We Salute You	1	For Those About To Rock (We Salute You)	1	1	Angus Young, Malcolm Young,
2	Accept	2	Balls to the Wall	2	Balls to the Wall	2	1	None
2	Accept	3	Restless and Wild	3	Fast As a Shark	2	1	F. Baltes, S. Kaufman, U. Dirks Hoffman
2	Accept	3	Restless and Wild	4	Restless and Wild	2	1	F. Baltes, R.A. Smith-Diesel, S. Dirkscneider & W. Hoffman
2	Accept	3	Restless and Wild	5	Princess of the Dawn	2	1	Deaffy & R.A. Smith-Diesel
1	AC/DC	1	For Those About To Rock We Salute You	6	Put The Finger On You	1	1	Angus Young, Malcolm Young,
1	AC/DC	1	For Those About To Rock We Salute You	7	Let's Get It Up	1	1	Angus Young, Malcolm Young,
1	AC/DC	1	For Those About To Rock We Salute You	8	Inject The Venom	1	1	Angus Young, Malcolm Young,
1	AC/DC	1	For Those About To Rock We Salute You	9	Snowballed	1	1	Angus Young, Malcolm Young,
1	AC/DC	1	For Those About To Rock We Salute You	10	Evil Walks	1	1	Angus Young, Malcolm Young,

```
#drop view
%%sql
drop view join_album_artista_track;
```

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

Indices

Indices

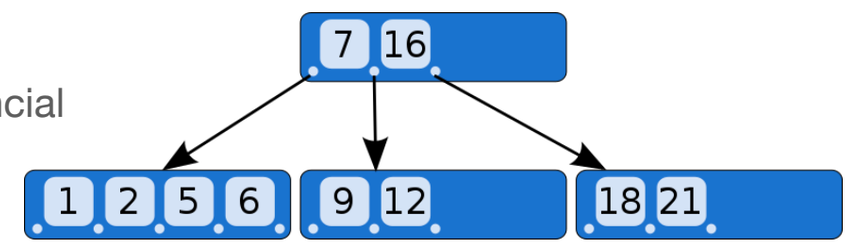
- Los indices sirven para optimizar consultas mediante el ordenamiento o indexación de uno o más columnas de una tabla.
    - La idea es encontrar filas sin la necesidad de revisar toda la tabla.
- Los indices requieren ser actualizados una vez que las tablas son modificadas.
    - CREATE [UNIQUE] INDEX nombre_index ON nombre_tabla (atributo1, atributo2…)
        - Los indices permiten valores duplicados pero la sentencia UNIQUE prohibe repeticiones.
- Existen algunos estandar para nombrar indices:
    - CREATE INDEX idx_alumno_nombre ON Alumno (nombre);
- DROP INDEX idx_alumno_nombre; # que pasa con la tabla alumno?
- Las claves primarias?

Indices

Optimización de consultas

- Sin indices, el motor de base de datos esta obligado de revisar las tablas completas en cada consulta.
    - Los JOINs son costosos coomputacionalmente.
    - La idea es encontrar filas sin la necesidad de revisar toda la tabla.
- Cada indice agrega una carga adicional a INSERT, UPDATE y DELETE.
    - Debemos evaluar donde colocar los indices.
- Los indices no se pueden crear en Vistas.
- Como almacena los datos el motor SQL?
- Por defecto, cada tabla es almacenada utilizando una estructura indexada (B-Tree SQLite).
    - A medida que se insertan filas en el B-Tree, las filas se ordenan, organizan y optimizan, de modo que una fila con un ROWID específico y conocido se puede recuperar de manera relativamente directa y rápida.

Permite: busquedas, inserciones, deleciones y accesso secuencial
Time(n)=O(log n)



- Cuando creamos un índice, el sistema de base de datos crea otro B-Tree para almacenar los datos del índice.
- El nuevo B-Tree se ordena y organiza usando la columna o columnas que se especifican en la definición del índice (! ROWID).

Indices

Ejemplo

- Creamos una tabla con 4 x 1000 numeros.

```
create table tbl (a,b,c,d);
INSERT INTO tbl (a, b,c,d)
VALUES
(84,39,78,79),
(182,39,67,153),
(83,166,143,188),
(145,205,380,366),
(317,358,70,303), …
```

```
1 %%time
2 %%sql
3 select * from tbl where a=29238;
```

```
 * sqlite:////content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
CPU times: user 5.2 ms, sys: 0 ns, total: 5.2 ms
Wall time: 7.23 ms
```

a	b	c	d
29238	3171	23996	48794
29238	297097	765679	213680

```
[12]  1 %%time
      2 %%sql
      3 create index idx_tbl_a_b ON tbl(a,b)
```

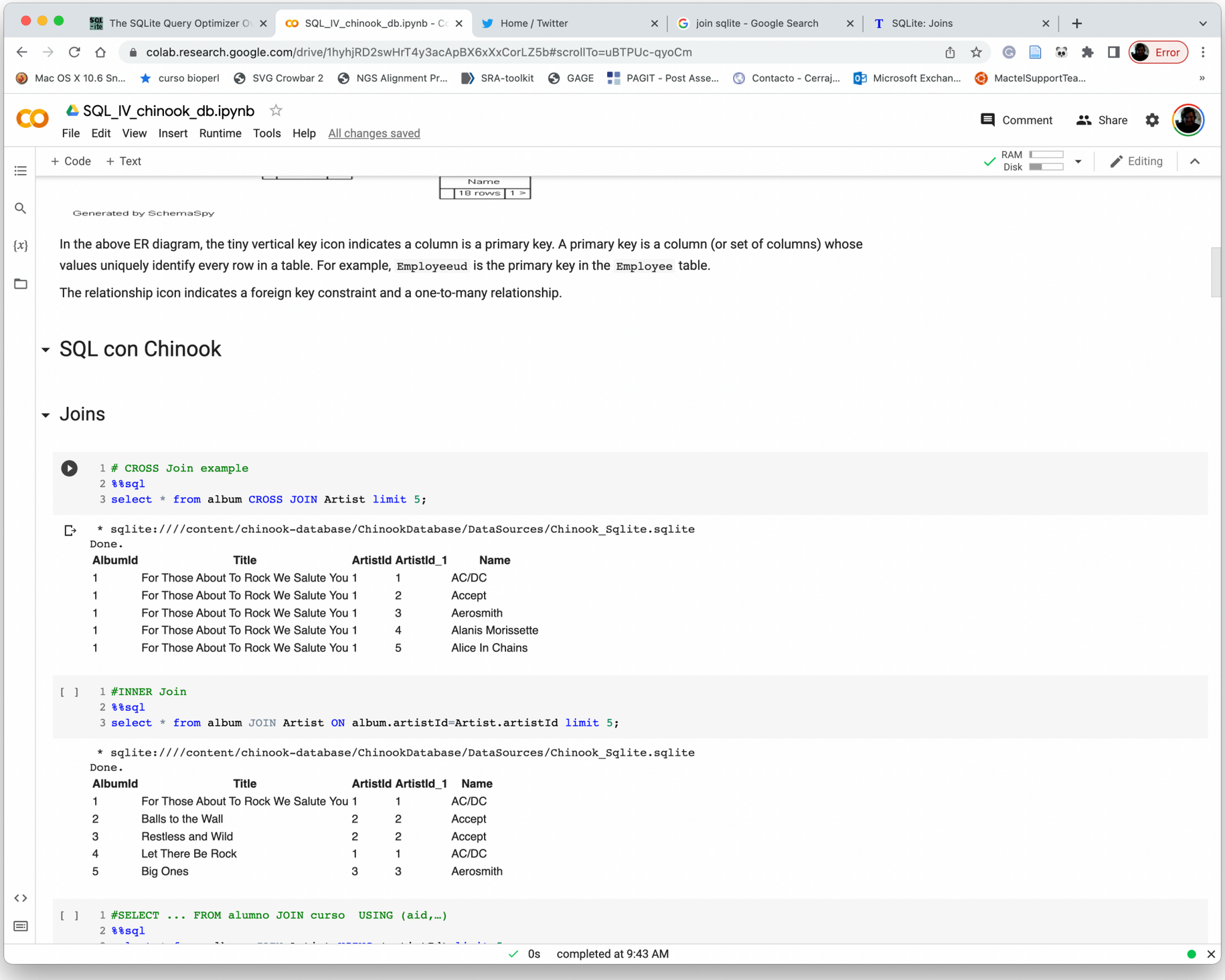
```
 * sqlite:////content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
CPU times: user 9.09 ms, sys: 32 µs, total: 9.13 ms
Wall time: 22.4 ms
[]
```

```
1 %%time
2 %%sql
3 select * from tbl where a=29238;
```

```
 * sqlite:////content/chinook-database/ChinookDatabase/DataSources/Chinook_Sqlite.sqlite
Done.
CPU times: user 3.88 ms, sys: 0 ns, total: 3.88 ms
Wall time: 3.76 ms
```

a	b	c	d
29238	3171	23996	48794
29238	297097	765679	213680

Practicar en GoogleColab!!!



https://github.com/adigenova/uohdb/blob/main/code/View_index_chinook_db.ipynb

Consultas?

Consultas o comentarios?

Muchas gracias