

REVISION ESTRUCTURA DE TABLAS DE MYSQL

Introduccion

- Consulte la estructura general de organización de los archivos en la [siguiente URL](ftp://mirbase.org/pub/mirbase/CURRENT/database_files/)
`ftp://mirbase.org/pub/mirbase/CURRENT/database_files/`
- Visualización código fuente de music.sql. Está ubicado en `Data/`
- Uso de mysql. Se trabaja con un ejemplo del esquema de music.sql -> En la línea de comandos escriba por favor

Caracteres Utiles

- `\` Alt+92
- ``` Alt+96
- `~` Alt+126

```
mysql -u root -p
-p es el password asignado.
salabio
```

Una vez cargado el cliente monitor es posible realizar operaciones sobre la interface. Lo primero es identificar por ejemplo la versión de MySQL que se usa .

Una vez cargado el cliente monitor es posible realizar operaciones sobre la interface. Lo primero es identificar por ejemplo la versión de MySQL que se usa .

****1. Ejemplo: version ****

```
>SELECT VERSION();
```

****2. Ejemplo: ayuda ****

```
>HELP Contents ;
```

ayuda con búsqueda de una categoría especial por ejemplo

```
>HELP Data Manipulation;
```

3. Para salir

escriba `\q` y `\c` para borrar la afirmación que se está usando

```
>\q
>\c
```

TALLER

1. Ingrese de nuevo a la interface de comunicación y en el monitor cargue la base de datos `music.sql`. Está almacenada en `Data/`

```
>SOURCE ../Data/music.sql
```

2. Confirme que la base de datos fue importada

```
>SHOW DATABASES;
```

3. Escoja para uso la base de datos, en este caso es music

```
>USE music;
```

4. Active, su base de datos

```
>SELECT DATABASE();
```

5. Identifique que tablas conforman la base de datos

```
>SHOW TABLES;
```

6. Identifique en el modelo de **Entidad Relacion** ER (Entity relationships) Revise la imagen correspondiente en las diapositivas de la clase y utilizando un editor de texto consulte la estructura del código de las tablas del archivo `music.sql`

7. Identifique las columnas de cada una de las tablas

```
>SHOW COLUMNS FROM artist;
```

- Tiene dos columnas
- Tipos **Entero** de longitud 5 y **cadena de caracteres** de tamaño 128
- Si se permite que este vacía `Null`
- Si es parte de una **Llave Primaria** en este caso es una clave primaria para la tabla
- En algunos diseños se pueden utilizar más de 2 columnas como clave primaria, observe la tabla `album` por ejemplo. A veces también se usa
- **Unique** key para restringir que no pueden existir 2 valores repetidos. En ese sentido una Primary key es una unique key.

8. Seleccione todos los datos de la tabla `artist`

```
>SELECT * FROM artist;
```

9. **SELECCIONANDO COLUMNAS:** pueden seleccionarse varias de ellas

```
>SELECT artist_name FROM artist;  
>SELECT artist_name,artist_id FROM artist;
```

10. SELECCIONANDO FILAS con WHERE : en su forma mas sencilla coincide con un valor

```
>SELECT * FROM artist WHERE artist_name = "New Order";  
>SELECT * FROM artist WHERE artist_id = 4;
```

• Cual es la diferencia?

#----- UNION DE 2 TABLAS

#Dentro de los objetivos de modelos de bases relacionales nos interesa las relaciones entre las tablas y poder hacer consultas. Si utilizamos las relaciones podemos responder a preguntas mas interesantes. Por ejemplo usted quisiera saber que tracks conforman un album, que album pertenece a cada artista o que tan largo el album va a ser reproducido. Estos ejemplos muestran los casos uniendo 2 tablas, para ello en la sintaxis se usa el operador INNER JOIN

```
>SELECT artist_name, album_name FROM artist INNER JOIN album USING (artist_id) WHERE artist_name < 'M';
```

Esta compuesta la sintaxis como 2 nombres de tablas separados por INNER JOIN y segundo se tiene la palabra clave USING que indican en cual columna se tiene la relacion entre las dos tablas

#-----OPERACIONES DE BUSQUEDA MAS COMPLEJAS -----

#G

#-----

```
>SELECT artist_name FROM artist WHERE artist_id < 5; puede usar <> o != no igual a
```

#----- Por ejemplo si desea buscar todos los albums que no tengan un album_id de 2 ... <> 2

```
>SELECT artist_name FROM artist WHERE artist_id <> 2;
```

#----- Funcionan los operadores para cadenas también pero aca se usa el orden lexicografico

```
>SELECT artist_name FROM artist WHERE artist_name < 'M';
```

#-----Otra acción que le interesaría es encontrar coincidencias usando algunos metacaracteres, se usa el operado LIKE

```
>SELECT album_name FROM album WHERE album_name LIKE "Retro%";
```

#-----O cuando desea ubicar un patrón repetido, por ejemplo "R__%" Se especifica una palabra de 3 caracteres iniciando en R.

```
>SELECT * FROM track WHERE track_name LIKE "R__%"
```

#-----Se pueden usar condiciones AND OR NOT XOR

#-----AND: se satisfacen 2 condiciones

#-----Desea encontrar un album con un titulo que comienza con un caracter mayor que C pero menor que M.

```
>SELECT album_name FROM album WHERE album_name > "C" AND album_name < "M";
```

#-----OR: se cumple una de las condiciones

```
>SELECT album_name FROM album WHERE album_name LIKE "L%" OR album_name LIKE "S" OR album_name LIKE "P%";
```

#-----NOT Suponga desea toda la lista de albums menos album_id 1 o 3.

```
>SELECT * FROM album WHERE NOT (album_id = 1 OR album_id =3);
```

XOR: exclusive operator X. Evalua como verdadero si solo si una de las expresiones es verdadera pero no ambas equivalente a (a AND (NOT b)) OR ((Not a) AND b)

#-----Ejemplo desea encontrar artistas cuyos nombres finalizan en "es" o comienzan con "The" pero no ambas

```
>SELECT artist_name FROM artist WHERE artist_name LIKE "The%" XOR artist_name LIKE "%es";
```

CONEXIÓN VIA REMOTA A LA BASE DE DATOS DEL UCSC

<https://genome.ucsc.edu/goldenpath/help/mysql.html>

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A

SHOW DATABASES;
USE dm1;
SELECT DATABASE();
SHOW TABLES;
SHOW COLUMNS FROM refGene;
SHOW COLUMNS FROM miRNA;
SHOW COLUMNS FROM seq;
SELECT chrom, cdsStart, cdsend, name FROM refGene;
SELECT name2 FROM refGene WHERE name2 LIKE "M%";
```

Extraer información de la base de datos de forma remota

Ejemplo 1

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -D dm1 -N -e "SELECT name, cdsStart, cdsEnd FROM refGene;" > dm1.bed
```

Ejemplo 2 : suponga que conoce el nombre del gen Myo95E (lo consultó en la base de datos en el paso SELECT name2 FROM refGene WHERE name2 LIKE "M%";)

Ahora de forma remota entonces:

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -D dm1 -N -e "SELECT name2, chrom, txStart, txEnd FROM refGene WHERE name2 LIKE 'Myo95E%';" > Myo95E.bed
```

Ejemplo 3: suponoga desea extraer las UTRs de los genes. Entonces

```
mysql -h genome-mysql.cse.ucsc.edu -u genome -D dm1 -N -A -e 'select chrom,strand, txStart,cdsStart from refGene w here  
txStart< cdsStart union select chrom,strand,cdsEnd,txEnd from refGene w here cdsEnd< txEnd ' > utrs.txt
```

Ejemplo 4: suponga desea bajar la anotación de miRNAs (recuerde estar conectado y cargada la base SELECT chrom,
chromStart, chromEnd, strand, name FROM miRNA;

Pero de forma remota

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -D dm1 -N -e "SELECT chrom, chromStart, chromEnd, strand, name  
FROM miRNA;" > miRNA Drosophila.bed
```

-----PARA MAS INFORMACION PUEDE CONSULTAR -----

Consulte el siguiente link para conocer el tipo de variables que maneja MySQL

http://www.sqlinfo.net/mysql/mysql_data_types.php

<http://www.mysqltutorial.org/>

Mucho mas sobre mas operaciones en sql

https://www.w3schools.com/sql/sql_join_inner.asp