

Universidad Especializada de las Américas

Licenciatura en Biomédica con Especialización en Electrónica Médica

Programación Avanzada

Taller de base de datos III

Ver:2.0 03/junio/2017

Importante: los prorgamas para instalar o ejecutar se deben hacer desde el modo Administrador, de lo contrario puede aparecer un mensaje de error por acceso denegado, esto ocurre mucho cuando los programas se instalan en Archivos de Programas o Programs Files y se intenta modificar algún archivo.

Prerrequisitos:

Leer , instalar y configurar los programas explicados en los documentos:

- Herramientas de Programación usadas en clase. Parte I.
- Herramientas de Programación usadas en clase. Parte II.
- Taller de base de datos I.
- Taller de base de datos II.

En el curso usaremos la base de datos FILM.FBD descargada de :

<https://github.com/damianquijano/PythonCurso3/tree/master/Data>

junto con otros archivos en formato CSV.

Utilizaremos los softwares:

- Firebird.
- SQL Manager Lite .

Índice.

1. Aprender actualizar registro desde la vista Form.
2. Aprender a eliminar registro desde la vista Form.
3. Aprender agregar nuevos registros desde la vista Form.
4. Aprender a:
 - Crear tablas
 - Editar tablas
 - Duplicar tablas
 - Eliminar tablas
5. Aprender a generar scripts con las instrucciones habituales
6. Aprender usar los parámetros para seleccionar datos.
7. Aprender a usar el constructor de Querys
8. Aprender a construir Joins más complicados con el constructor de Querys.
9. Crear y eliminar una base de datos.
10. Copiar y pegar los resultados a excell

1. Aprender actualizar registro desde la vista Form.

Ejecutamos un select para traer todos los registros y sus columnas de la tabla CATEGORY.

Recuerda que antes de usar el Form View, deben existir resultados producto de un query de consulta (select), el cual establece los campos que verás en el Form View.

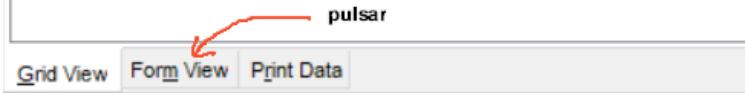
```
SELECT * FROM CATEGORY ;
```

The screenshot shows a database interface with a toolbar at the top containing buttons for Edit, Results, Performance Analysis, and Logs. The Results tab is highlighted with a red box. Below the toolbar is a row of navigation buttons for navigating through the results. The main area displays a grid of data with two columns: 'CATEG' and 'NAME'. The 'CATEG' column contains numerical values from 1 to 17, and the 'NAME' column contains category names like Action, Animation, Children, Classics, Comedy, Documentary, Drama, Family, Foreign, Games, Horror, Music, New, Sci-Fi, Sports, Travel, and Manga. The row for Category ID 17, 'Manga', is highlighted with a blue selection bar. At the bottom of the grid, there are three tabs: 'Grid View', 'Form View', and 'Print Data'. A red arrow points from the 'Form View' tab to the 'Form View' tab, indicating the current view mode.

CATEG	NAME
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-Fi
15	Sports
16	Travel
17	Manga

Al ejecutar el query, aparece la pestaña de Results al lado de Edit (ver arriba subrayado en rojo) y vemos los resultados. Abajo, resaltado también en rojo, tenemos tres pestañas: Grid View, Form View y Print Data. En estos momentos estamos en Grid View que nos muestra los resultados del query. Vamos a modificar el registro de la tabla Category cuyo campo Category_ID es 17, tenemos que pulsar sobre el registro hasta que quede resaltado en azul.

12	MUSIC
13	New
14	Sci-Fi
15	Sports
16	Travel
17	Manga



Pulsamos la pestaña Form View.

Abajo podemos ver el modo Form View que permite actuar sobre cada registro con las operaciones de Modificar, Eliminar o Crear un nuevo registro, detrás de esto están las acciones de Update, Delete e Insert.

CATEGORY_ID

NAME

Si pulsa en simbolo menos, eliminará el registro que aparece en el cuadro y que previamente has seleccionado.

17

Manga

Podemos escribir en ambos cuadros y esta es la información que cambiará los valores que estás viendo

Grid View Form View Print Data

Edit Results Performance Analysis Logs

Find:

CATEGORY_ID	<input type="checkbox"/> Null	<input type="text" value="17"/>
NAME	<input type="checkbox"/> Null	<input type="text" value="Cine Asiático"/>

Hemos cambiado el contenido de Name:Cine Asiático. Al momento de escribir en alguno de los recuadros, el programa habilita la opción Update o modificar mediante el gancho verde (toma el color verde al detectar posible modificación), es lo que haremos pulsando sobre el gancho.

Grid View Form View Print Data

CATEGORY_ID	<input type="checkbox"/> Null	<input type="text" value="17"/>
NAME	<input type="checkbox"/> Null	<input type="text" value="Cine Asiático"/>

Después de pulsar el gancho verde, vuelve a su estado de color gris, lo que nos indica que la modificación se realizó, pero la mejor manera de comprobarlo es volviendo al modo Grid View para ver los registros de la tabla.

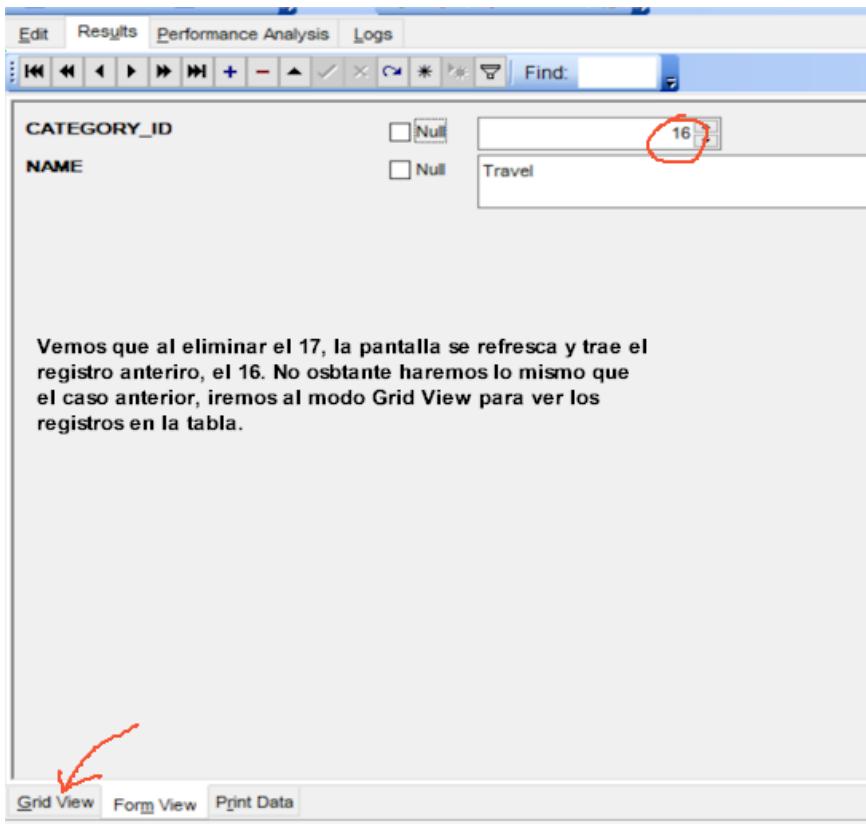
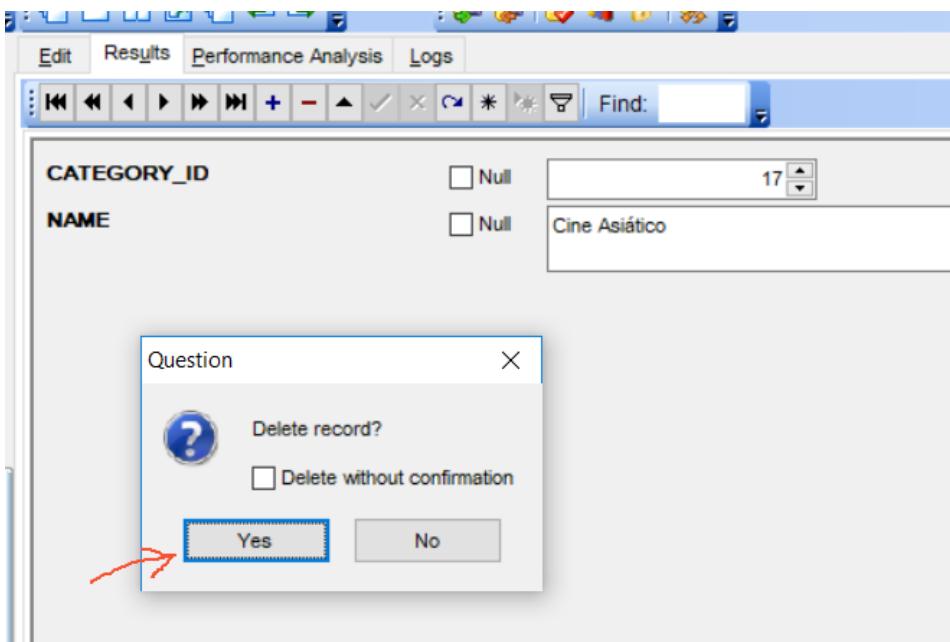
CATEG	NAME
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-Fi
15	Sports
16	Travel
17	Cine Asiático

Ejecuta el query de todas maneras para refrescar los datos.

2. Aprender a eliminar registro desde la vista Form.

Igual que antes, desde Grid View, seleccionamos un registro, en este caso el último cuyo CATEGORY_ID es 17 y pulsamos Form View.

CATEGORY_ID	NAME
17	Cine Asiático



Vemos que al eliminar el 17, la pantalla se refresca y trae el registro anterior, el 16. No obstante haremos lo mismo que el caso anterior, iremos al modo Grid View para ver los registros en la tabla.

Drag a column header here to group by that column

CATEG	NAME
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-Fi
15	Sports
16	Travel

Ya no existe el 17

Grid View Form View Print Data

Ejecuta el query de todas maneras para refrescar los datos.

3. Aprender agregar nuevos registros desde la vista Form.

Recuerda que antes de usar el Form View, deben existir resultados producto de un query de consulta (select), el cual establece los campos que verás en el Form View.

Es importante que selecciones el último registro, pues vamos a crear uno nuevo y debe quedar después del último registro actual; vamos directamente a Form View y pulsamos el símbolo en rojo más (+). Esto blanqueara los recuadros, los cuales debes llenar .

Edit Results Performance Analysis Logs

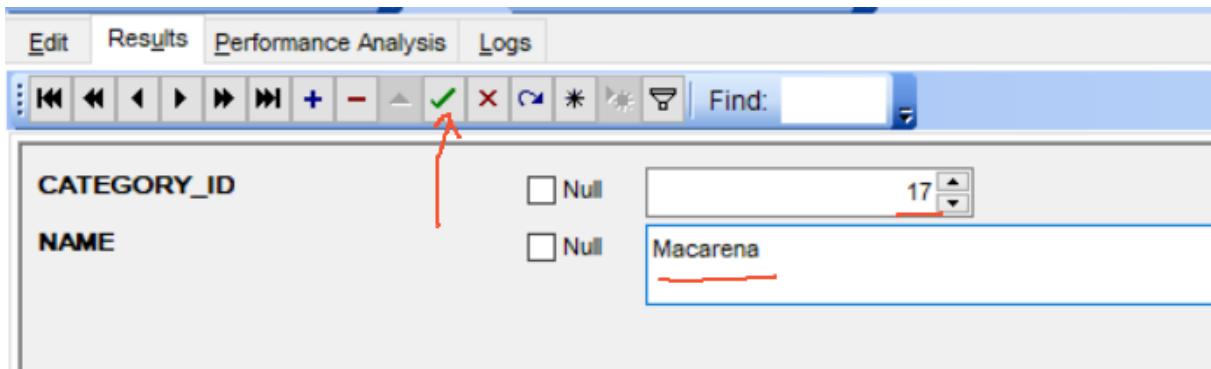
Insert record

CATEGORY_ID

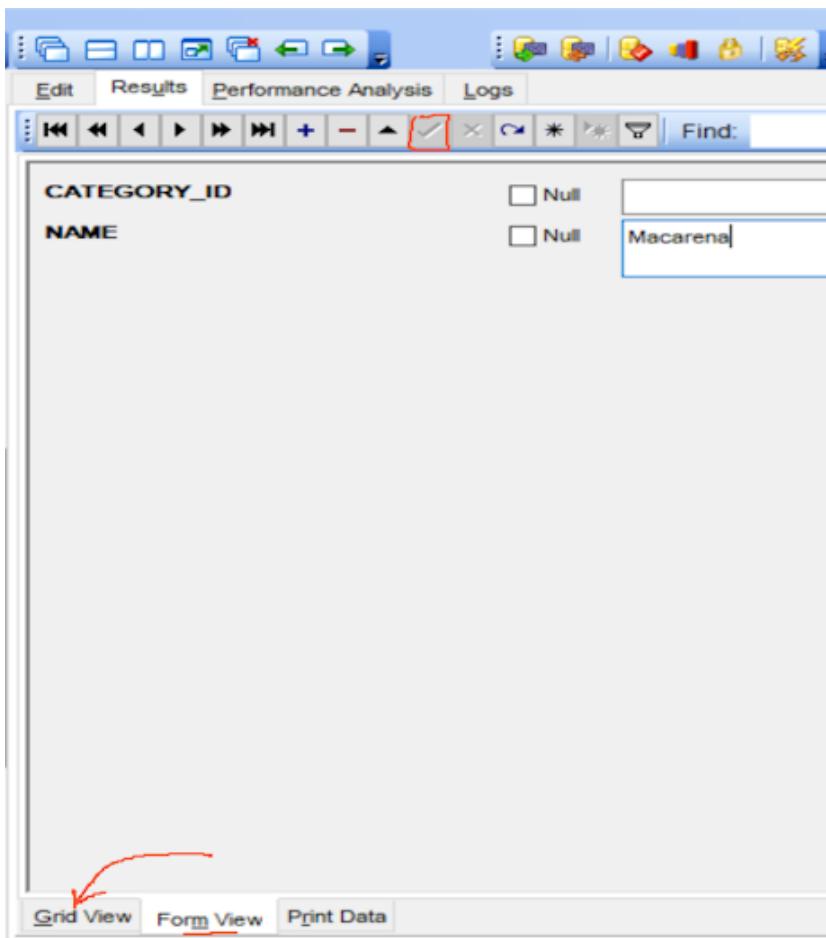
NAME

Vemos que se habilita el gancho verdad, que es el accionador del Insert, no lo pulses hasta que llenes los recuadros, de lo contrario agregará un registro con datos en blanco.

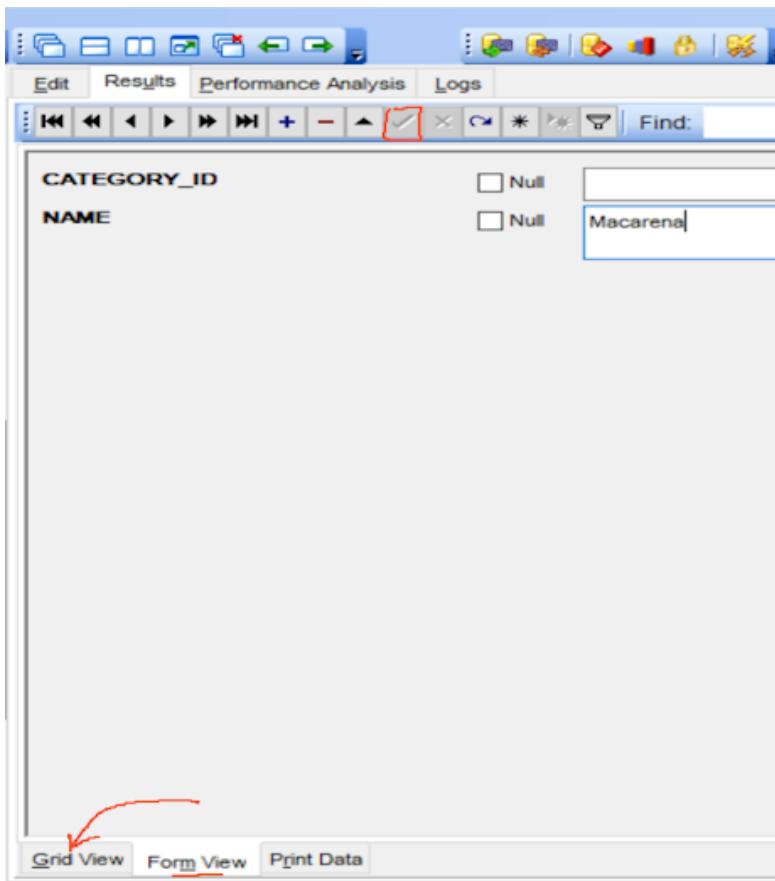
Para evitar eso, quita los checks de Null a ambos recuadros para evitar que agregues en blanco.



Arriba, hemos llenado los recuadros y entonces procederemos a pulsar el gancho verde de confirmación.



Vemos que el gancho ha quedado en color gris, lo cual significa que ha sido accionado el Insert y no hay más acciones que confirmar. Procedemos a pulsar sobre Grid View para ver la tabla de los registros.



Drag a column header here to group by that column	
CATEG	NAME
1	Action
2	Animation
3	Children
4	Classics
5	Comedy
6	Documentary
7	Drama
8	Family
9	Foreign
10	Games
11	Horror
12	Music
13	New
14	Sci-Fi
15	Sports
16	Travel
17	Macarena

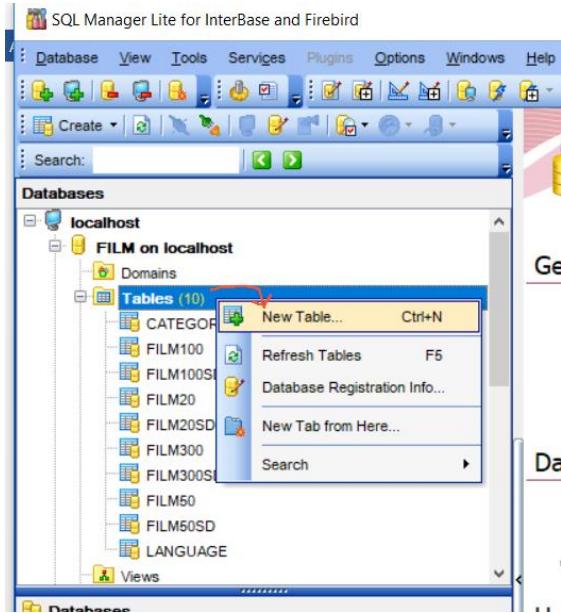
Ejecuta el query de todas maneras para refrescar los datos.

4. Aprender a:

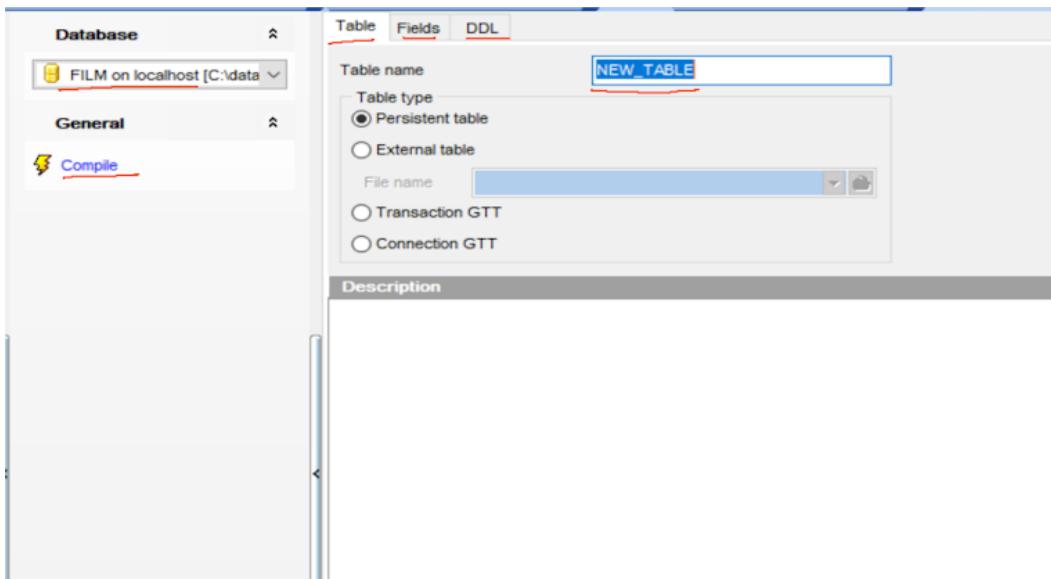
- Crear tablas

En la base de datos FILM, vamos a crear una nueva tabla llamada **Estudiantes**. En la vida real no se crean tablas con contenidos diferentes al objetivo de la base de datos. La base de datos FILM contiene tablas que contienen información sobre un negocio de alquiler de películas, por tanto una tabla con información de estudiantes no encaja en la base de datos, pero el objetivo es aprendizaje y no tomaremos en cuenta la advertencia.

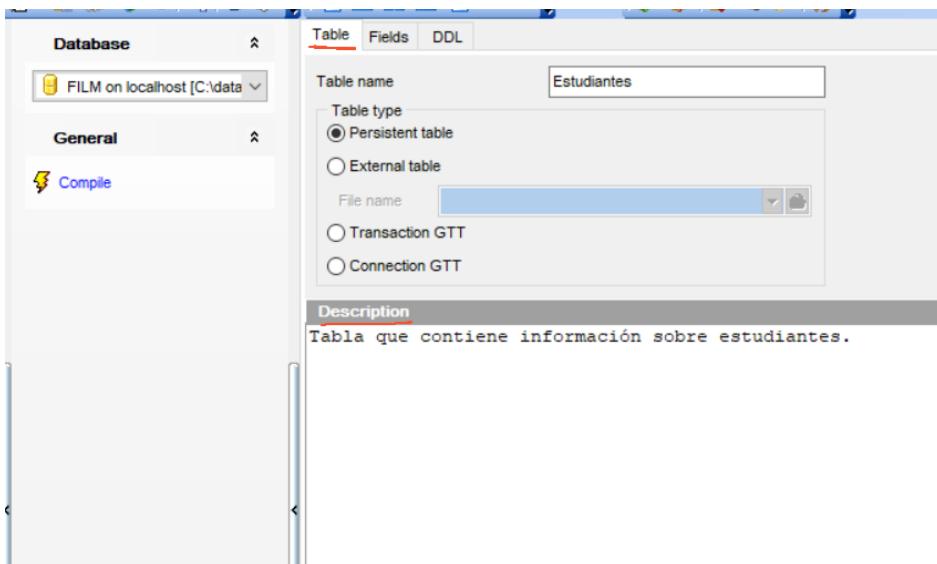
Seleccionamos la base de datos dentro de la cual deseamos crear una nueva tabla, y nos posicionamos encima de la sección de Tables y hacemos click derecho para que aparezca la ventana siguiente:



Seleccionamos la opción New Table.

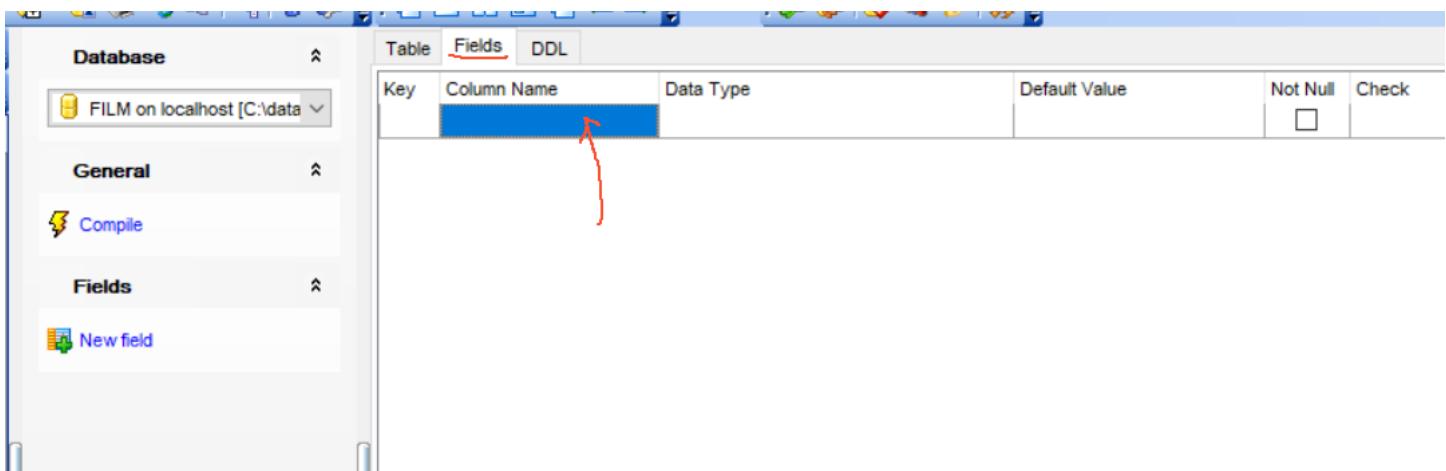


Arriba vemos que hay un nombre por defecto de la tabla, la que cambiaremos por el nombre Estudiantes. Nos damos cuenta que estamos en la pestaña (o tab) Table, y hay otras dos pestañas: Fields que permite agregar los campos a la tabla y DDL en el cual aparece las instrucciones SQL que permitan crear la tabla con sus campos.

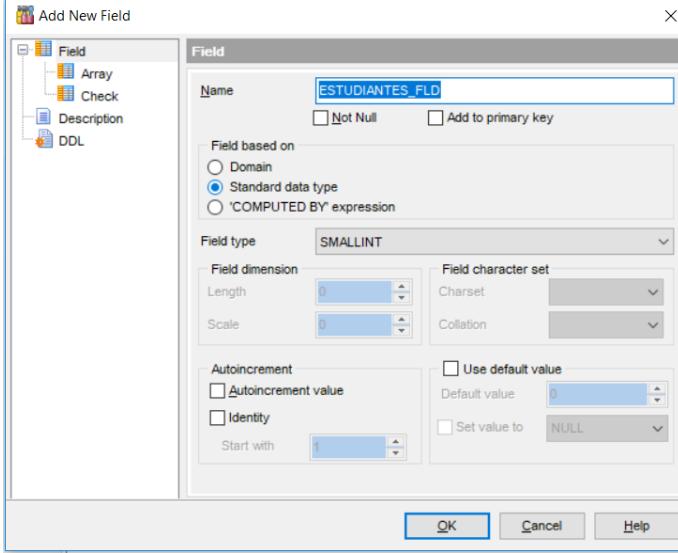


Terminado lo de arriba, pulsamos la pestaña Fields para agregar los campos.

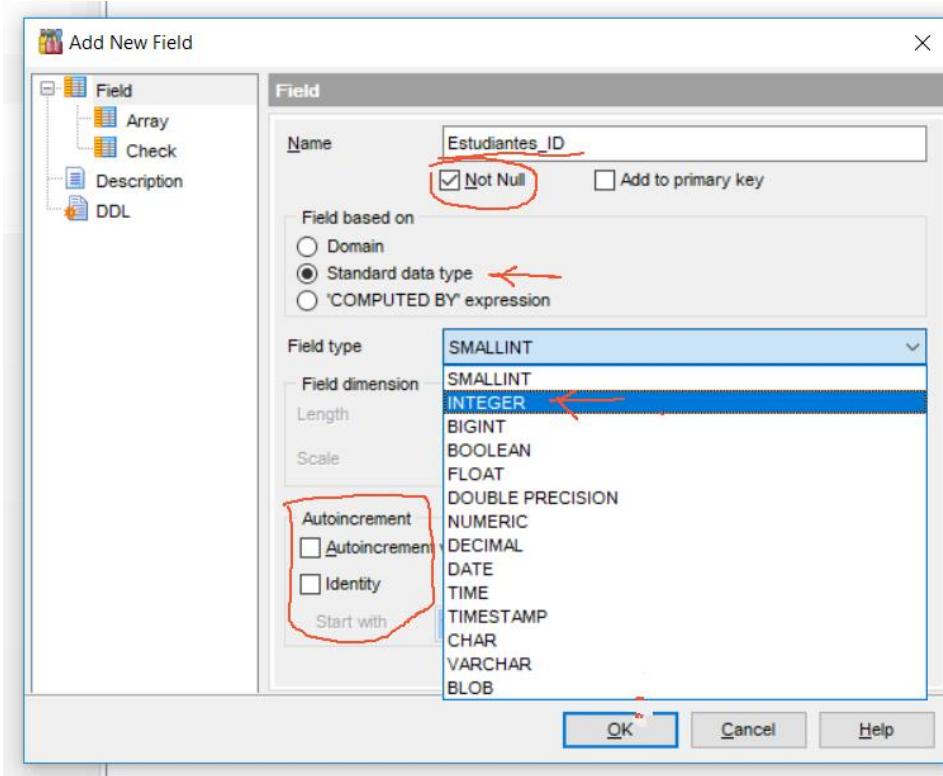
Abajo vemos la pantalla Fields. Para agregar un campo, hay que hacer doble clik sobre el espacio de Column Name para que aparezca la pantalla del Campo.



Abajo vemos la pantalla que construye el nuevo campo y que cambiaremos sus datos.



Abajo escribimos el nombre de nuestro Campo: Estudiantes_ID. Escogemos que no se permita que al momento de agregar una fila, el campo Estudiantes_ID quede en blanco(Nulo). Más adelante explicaremos el concepto de primary key . Dejaremos Standard data type, o sea, los tipos de datos que aparecen en la lista. Por defecto es SmallInt que se refiere a que el valor del campo es entero pequeño, o sea, acepta valores entre -255 y 255. Nosotros vamos a seleccionar el tipo de datos Integer, que permite un rango de números mucho mayor. Ojo, es entero, no permitirá valores decimales o letras.



Vemos en la ventana las opciones **Autoincrement** e **Identity**. **Autoincrement** indica al campo que sus valores los genera el motor de base de datos al momento de agregar un nuevo registro. Esto significa, que el usuario llenará los valores de todos los campos, excepto los valores de un campo marcado como Autoincrement. Al crear el primer registro, el campo con marca autoincrementado, automáticamente asignará el valor 1 . Al agregar otro registro, producirá el valor 2, y así sucesivamente y de forma secuencial y ascendente,

esto permite contar con una columna que se autonumera a medida que se crea un nuevo registro y evita que se tenga que hacer manualmente al momento de crear nuevos registros a la tabla. **Identity** marca al campo que vas a crear como un campo que contendrá valores únicos que no se pueden repetir, por tanto si al momento de crear una nuevo registro, metes un valor a dicho campo que ya existe en ese mismo campo pero en otro registro, te mostrará un error diciendo que ya hay otro registro que se repite en dicho campo y no permitirá que agregues el nuevo registro.

Continuamos y pulsamos OK.

Key	Column Name	Data Type	Default Value	Not Null	Check
	ESTUDIANTES_ID	INTEGER		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Vemos arriba que tenemos el primer campo creado en la tabla Estudiantes.

Abajo creamos el campo llamado Nombre, esta vez del tipo palabras o cadena llamado Varchar y de un tamaño de 50 caracteres, si se sobrepasa la información que se introduzca en este campo (un nombre enorme de 150 letras) aparecerá un mensaje de error o truncará la información recortando a 50 letras de las 150.

Name: NOMBRE
 Not Null Primary key

Field based on:
 Domain
 Standard data type
 'COMPUTED BY' expression

Field type: VARCHAR

Field dimension:
Length: 50 Scale: 0

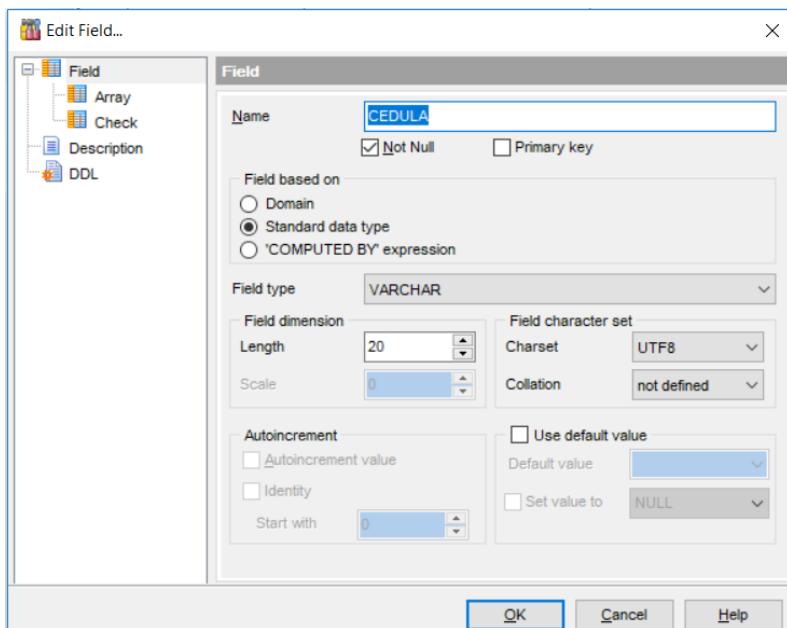
Field character set:
Charset: UTF8 Collation: not defined

Autoincrement:
 Autoincrement value
 Identity
Start with: 0

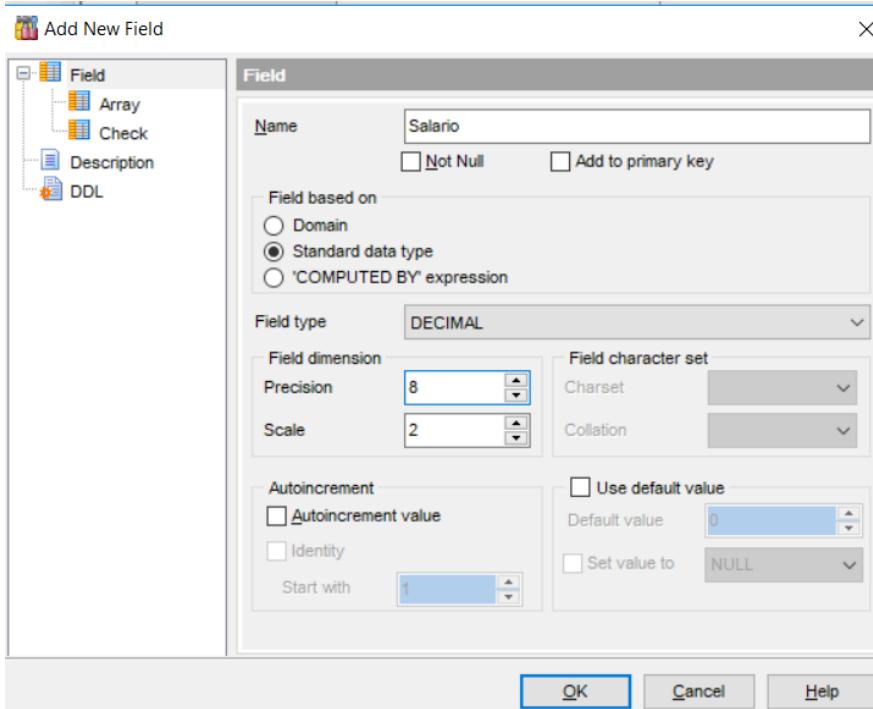
Use default value:
 Use default value
Default value:
 Set value to: NULL

Buttons: OK, Cancel, Help

Debemos seleccionar Charset UTF8 para que no existan problemas sobre la definición del juego de caracteres usados por la tabla. Esto solo aplica a campos cuyo tipo de datos es string o letras (Varchar)



Por último creamos el campo llamado Salario, es numérico decimal, por tanto seleccionamos su tipo de dato Decimal, entonces te pedirá precisión y Scala. Escribimos 8 y 2, lo cual significa que el número que podemos introducir tiene un máximo de 8 dígitos numéricos, de los cuales 2 son usados para posiciones decimales, ejem: 345,678.50 , vemos que son 8 dígitos numéricos de los cuales 2 son para decimales.



Terminados de construir todos los campos, vamos a la pestaña DDL, y veremos la construcción de la tabla con sus campos en versión código SQL.

```

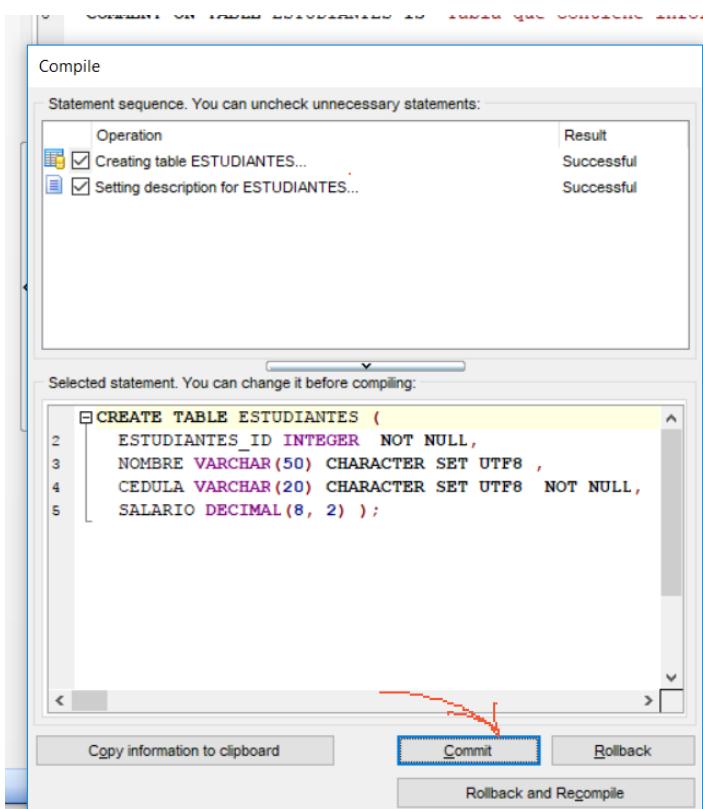
CREATE TABLE ESTUDIANTES (
    ESTUDIANTES_ID INTEGER NOT NULL,
    NOMBRE VARCHAR(50) ,
    CEDULA VARCHAR(20) NOT NULL,
    SALARIO DECIMAL(8, 2) ;
)

COMMENT ON TABLE ESTUDIANTES IS 'Tabla que contiene información sobre estudiantes.';

```

Debemos pulsar Compile para que mande ejecutar el código SQL y crear la tabla.

Después de compilar, aparece una pantalla que nos indica que la operación se ejecutó ok pero requiere confirmar mediante el Commit.



Abajo, vemos que la tabla Estudiantes aparece en el panel de las Tables junto con las demás, y a la derecha vemos la estructura de la tabla junto con otras nuevas pestañas que permitan agregar muchas nuevas funcionalidades que este curso básico no pretende aún cubrir.

Databases

- localhost
 - FILM on localhost
 - Tables (11)
 - CATEGORY
 - ESTUDIANTES
 - FILM100
 - FILM100SD
 - FILM20
 - FILM20SD
 - FILM300
 - FILM300SD
 - FILM50
 - FILM50SD
 - LANGUAGE

Object

Table Indices Triggers Description Data Dependencies Permissions DDL

Name: ESTUDIANTES

Fields Constraints

Field Name	Field Type	Domain	Not Null	Auto Inc	Identity	Default Source
ESTUDIANTES_ID	INTEGER	RDB\$212	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NOMBRE	VARCHAR(50)	RDB\$213	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CEDULA	VARCHAR(20)	RDB\$214	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SALARIO	DECIMAL(8, 2)	RDB\$215	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

General

Print Refresh table Options

Fields

New field Edit field Drop field

Tools

- Editar tablas

Seleccionamos la tabla de nuestro interés y click derecho, pulsamos la opción Edit Table.

Databases

- localhost
 - FILM on localhost
 - Tables (11)
 - CATEGORY
 - ESTUDIANTES
 - FILM100
 - FILM100SD
 - FILM20
 - FILM20SD
 - FILM300
 - FILM300SD
 - FILM50
 - FILM50SD
 - LANGUAGE

General

Print Refresh table

New Table... Ctrl+N

Edit Table 'ESTUDIANTES'... Ctrl+O

Drop Table 'ESTUDIANTES'... Shift+Del

Duplicate Table 'ESTUDIANTES'...

Tasks

Data Manipulation

Script to New SQL Editor

Script to Clipboard

Refresh Tables F5

Database Registration Info...

Search

Type

ESTUDIANTES_ID

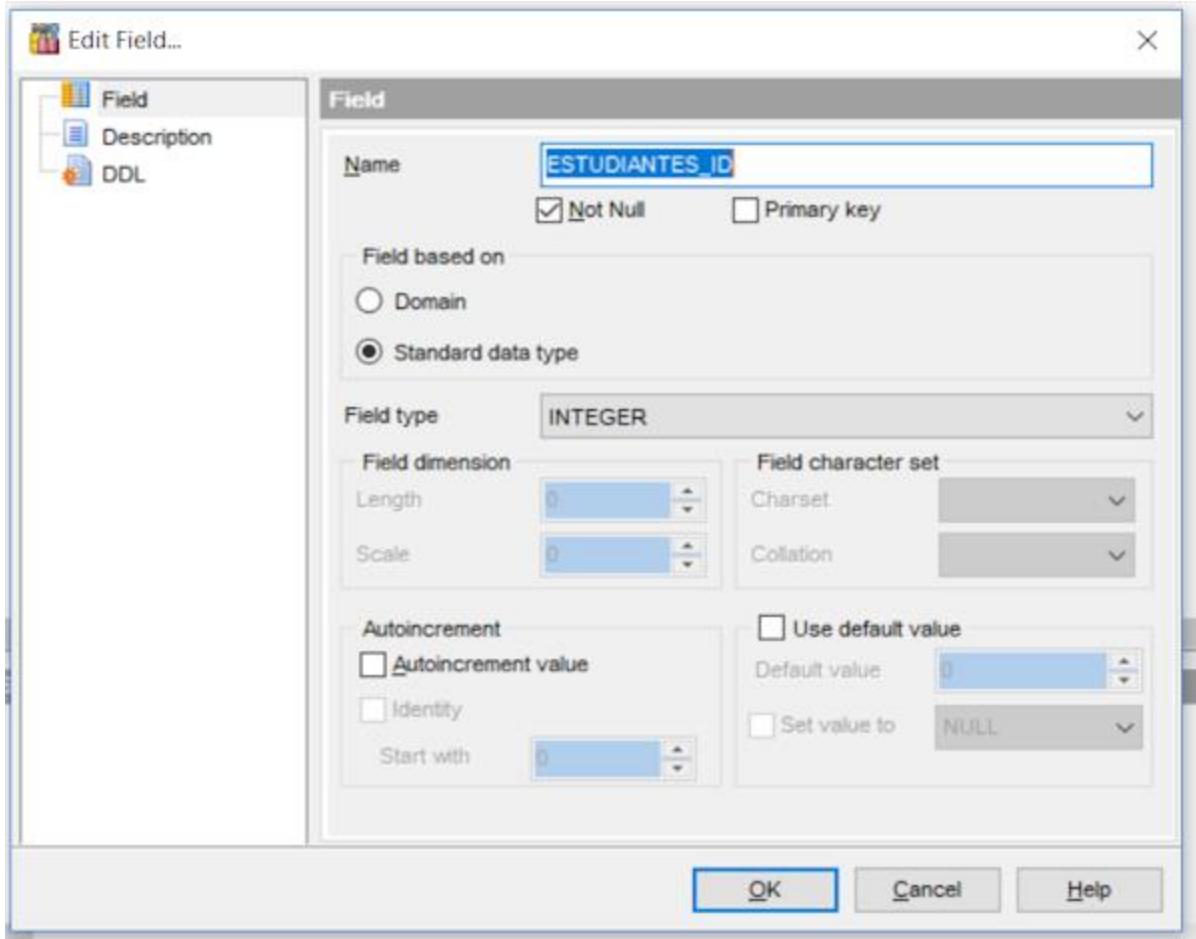
Explorer

Al seleccionar Edit aparece la pantalla que ya conocemos al momento de crear la tabla:

The screenshot shows the 'Fields' tab of the 'ESTUDIANTES' table configuration in Oracle SQL Developer. A context menu is open over the 'ESTUDIANTES_ID' field, which is defined as an INTEGER type. The menu includes options for creating a new field, editing the current one, renaming it, dropping it, duplicating it, reordering fields, copying field names to clipboard, and exporting the list.

Field Name	Field Type	Domain	Not Null	Auto Inc	Identity	Default Source	Computed Source
ESTUDIANTES_ID	INTEGER	NUMBER		<input type="checkbox"/>	<input type="checkbox"/>		
NOMBRE			Ins	<input type="checkbox"/>	<input type="checkbox"/>		
CEDULA			Edit	<input type="checkbox"/>	<input type="checkbox"/>		
SALARIO				<input type="checkbox"/>	<input type="checkbox"/>		

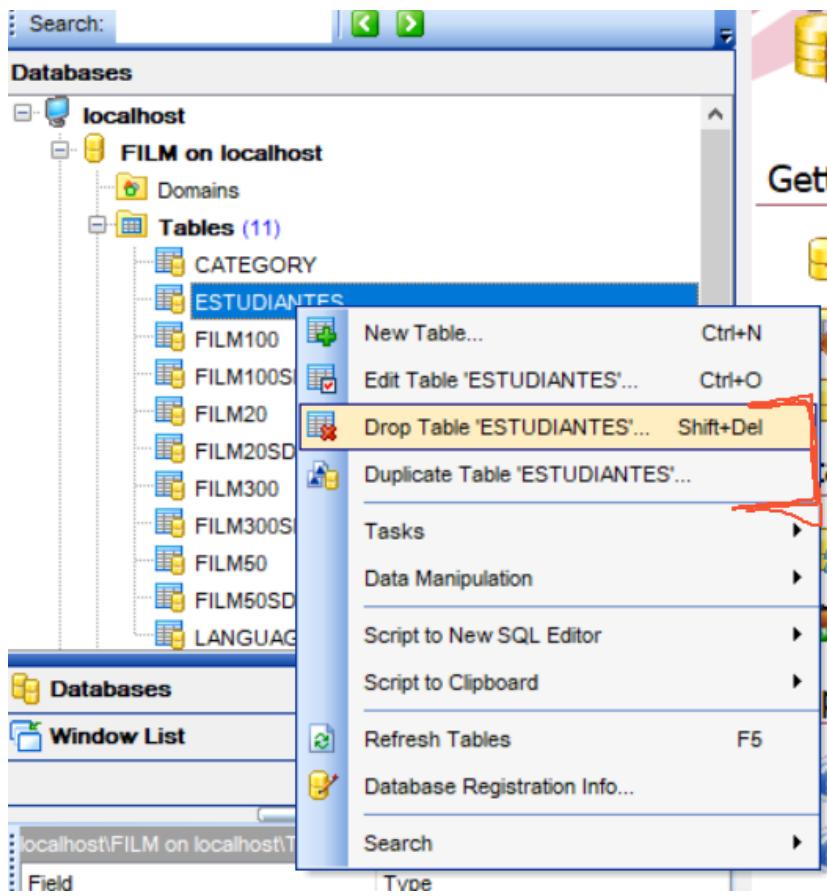
Arriba, seleccionamos un campo y hacemos click derecho. Tenemos la opción de cambiar el nombre del campo, eliminarlo, duplicarlo pero también modificarlo. Al pulsar Edit aparece la ventana ya conocida:



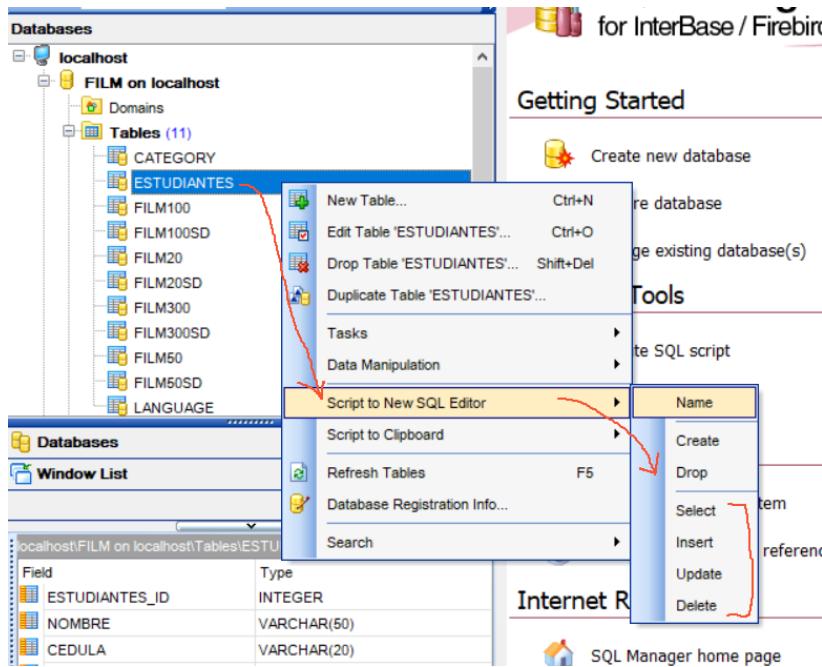
La pantalla permite cambiar el nombre, el tipo de datos (pero ojo, porque puede afectar los registros que ya cuenta la tabla con ese tipo de datos y puedes perderlos) y cualquier otra configuración previa.

- Duplicar tablas y eliminar tablas

Drop permite eliminar la tabla seleccionada y Duplicate permite generar un copia pero le pondrá otro nombre. Esto último es útil para tener una copia antes de modificar la tabla y que provoque borrados o modificaciones no deseadas.



5. Aprender a generar scripts de select, update, delete e insert de forma automática.



Al pulsar cada uno de esos botones, producirá lo siguiente:

```

1 SELECT
2   ESTUDIANTES_ID,
3   NOMBRE,
4   CEDULA,
5   SALARIO
6 FROM
7   ESTUDIANTES ;

```

```

1 INSERT INTO
2   ESTUDIANTES
3 (
4   ESTUDIANTES_ID,
5   NOMBRE,
6   CEDULA,
7   SALARIO
8 )
9 VALUES (
10   :ESTUDIANTES_ID,
11   :NOMBRE,
12   :CEDULA,
13   :SALARIO
14 );

```

```

1 UPDATE
2   ESTUDIANTES
3 SET
4   ESTUDIANTES_ID = :ESTUDIANTES_ID,
5   NOMBRE = :NOMBRE,
6   CEDULA = :CEDULA,
7   SALARIO = :SALARIO
8 WHERE
9 ;

```

```

1 DELETE FROM
2   ESTUDIANTES
3 WHERE
4 ;

```

Puedes modificarlos y ajustarlos a lo que deseas y ejecutarlos. Tal como aparecen, los valores son parámetros, por ejemplo: Nombre= :Nombre , esto último, con los dos puntos, se refiere a que :Nombre es un parámetro, y significa que al momento de ejecutar el query, aparecerá una pantalla que te pedirá que escribas el valor para el campo Name. Lo mismo para el resto de los querys de arriba.

6. Aprender usar los parámetros para seleccionar datos.

Esta técnica permite que se aporten valores de búsqueda al momento de ejecutar el query en vez de establecer los valores de búsqueda antes e incorporados en la construcción del query.

Veamos mediante un ejemplo lo que se desea enseñar.

Supongamos que deseamos conocer los valores de la información contenida en la tabla Category cuya columna Category_Id sea 1, o sea, deseamos ver todos los registros que pertenecen a la categoría 1. Esto sería así:

```
Select * from CATEGORY where CATEGORY_ID=1;
```

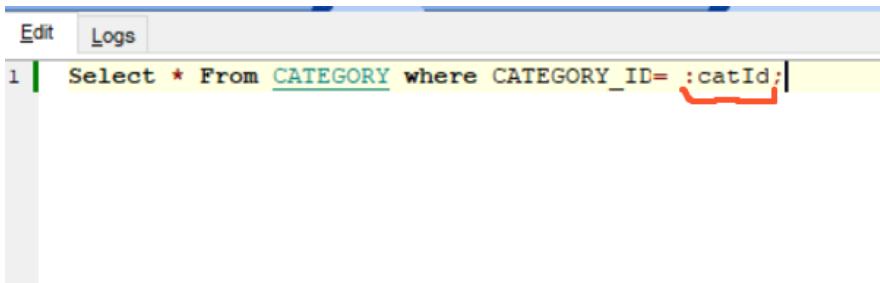
lo cual nos muestra:

CATEGORY_ID	NAME
1	Action

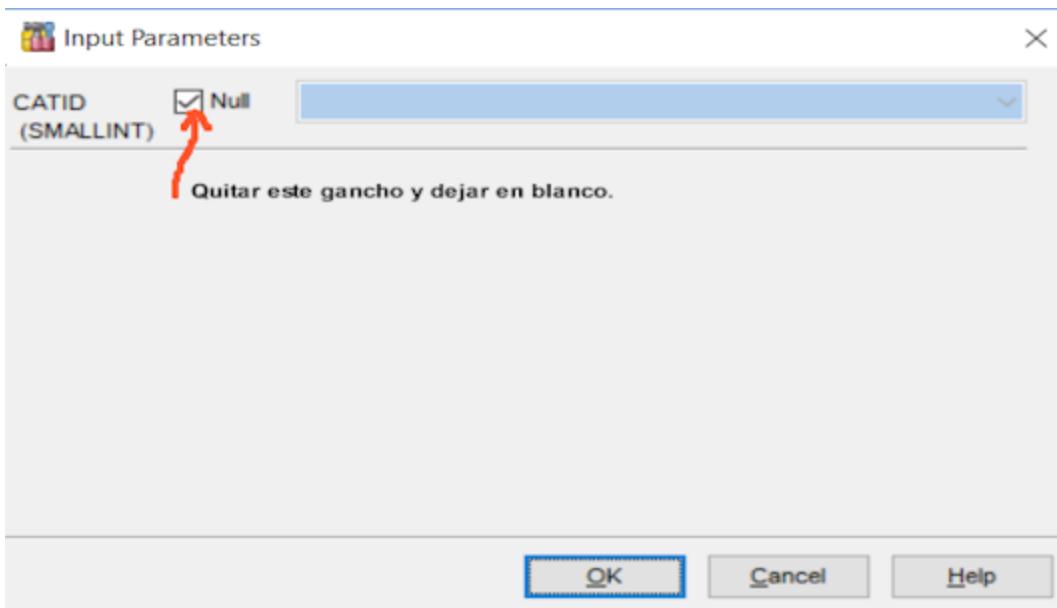
Ahora bien ¿y si deseamos que sea 2? tendríamos que escribir en la construcción del query el valor de 2 ¿y si queremos otro? ¿hay que escribir el query tantas veces como deseamos cambiar el valor de la categoría? . La solución es escribir una variable (parámetro) en vez de un valor fijo anteponiendo al nombre de la variable los dos puntos (:). Veamos la misma construcción del query anterior pero esta vez usando un parámetro:

```
Select * From CATEGORY where CATEGORY_ID= :catId;
```

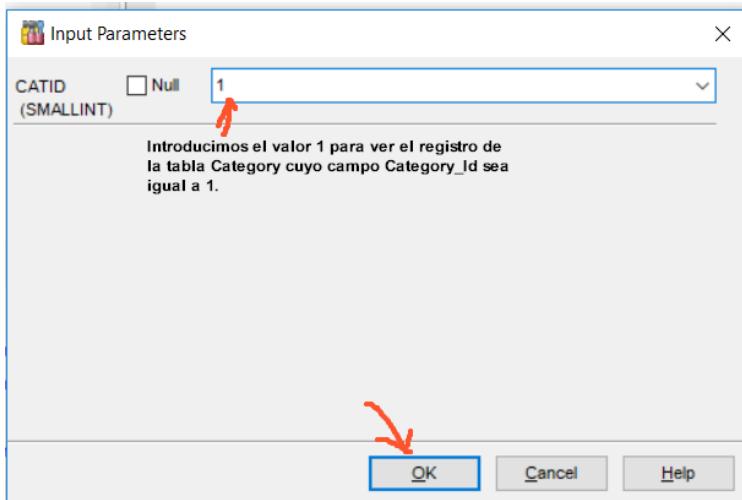
Vemos arriba que en vez de escribir por ejemplo 1, escribimos :catId, pero puede ser cualquier nombre, por ejemplo :categ o :Idcat , lo importante es que debe preceder de los dos puntos : .



Al momento de ejecutar el query de arriba, aparece la siguiente pantalla que nos pedirá el valor que deseamos consultar:



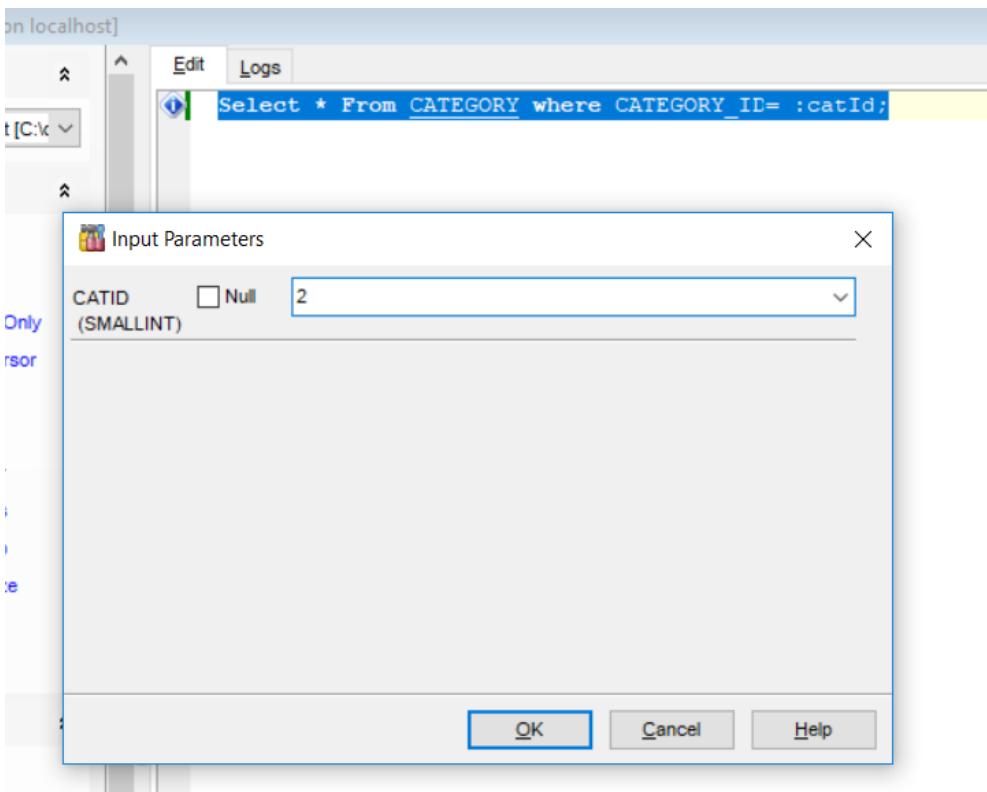
Arriba debemos dejar en blanco la opción Null para que habilite que podamos escribir un valor.



Produce el mismo resultado:

CATEGORY_ID	NAME
1	Action

Corremos el mismo query pero escribiremos otro valor, por ejemplo 2:

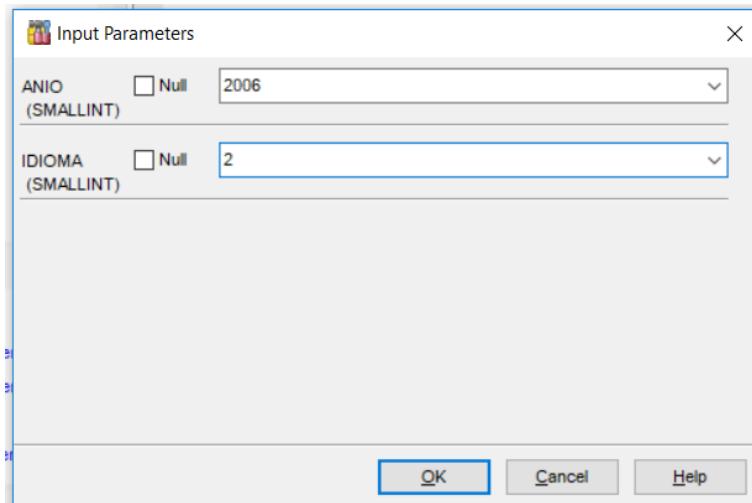


Resultado:

CATEGORY_ID	NAME
2	Animation

Incluso puede ser más complejo:

```
Select * From FILM20SD where RELEASE_YEAR= :anio and LANGUAGE_ID= :idioma;
```

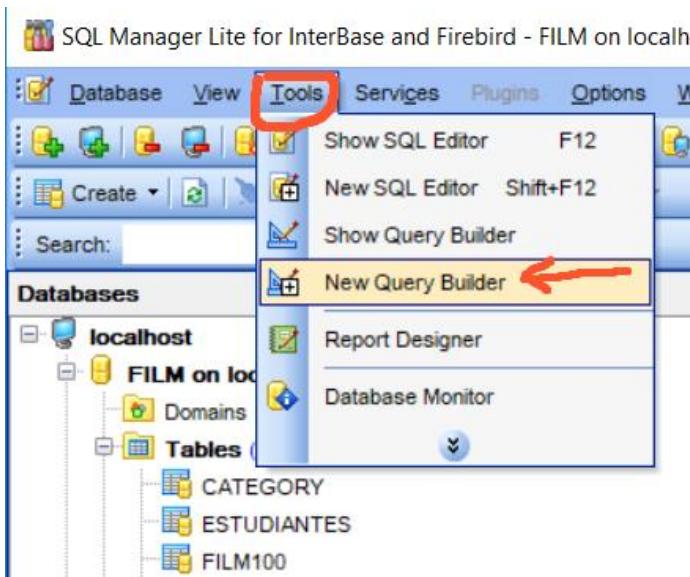


The screenshot shows a software interface for managing databases. At the top, there's a menu bar with 'Edit', 'Results', 'Performance Analysis', and 'Logs'. Below the menu is a toolbar with various icons. The main area contains a table titled 'FILM_JI' with columns: TITLE, RELEAS, LANGUA, RENTAL, RENTAL_RATI, LENGTH, REPLACEMEN, and RA. A single row is visible: 'CAROL TEXAS' with values 2,006, 2, 4, 2.99, 151, and 15.99 respectively.

FILM_JI	CATEG	TITLE	RELEAS	LANGUA	RENTAL	RENTAL_RATI	LENGTH	REPLACEMEN	RA
8	2	CAROL TEXAS	2,006	2	4	2.99	151	15.99	PC

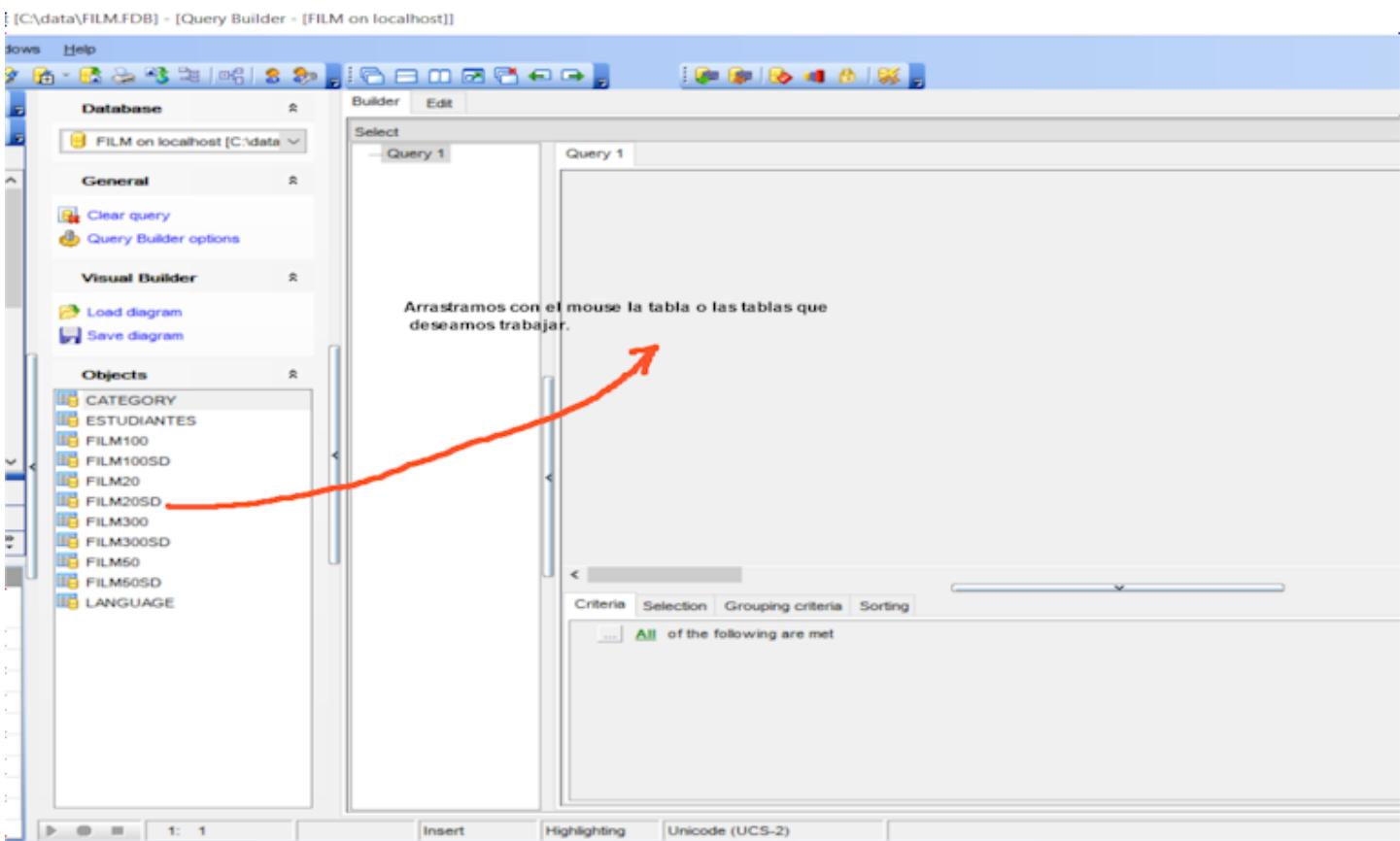
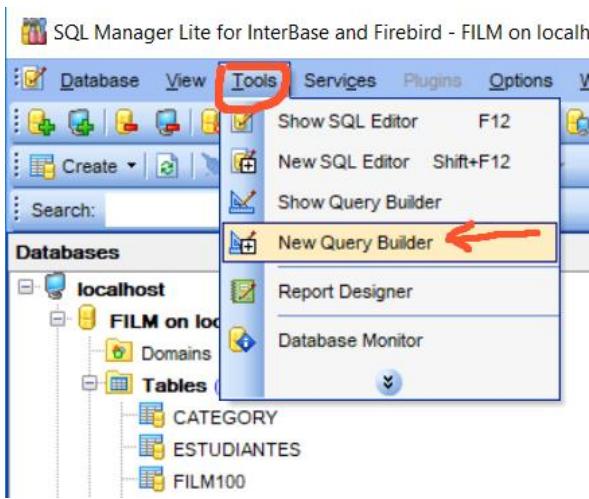
7. Aprender a usar el constructor de Querys

El constructor

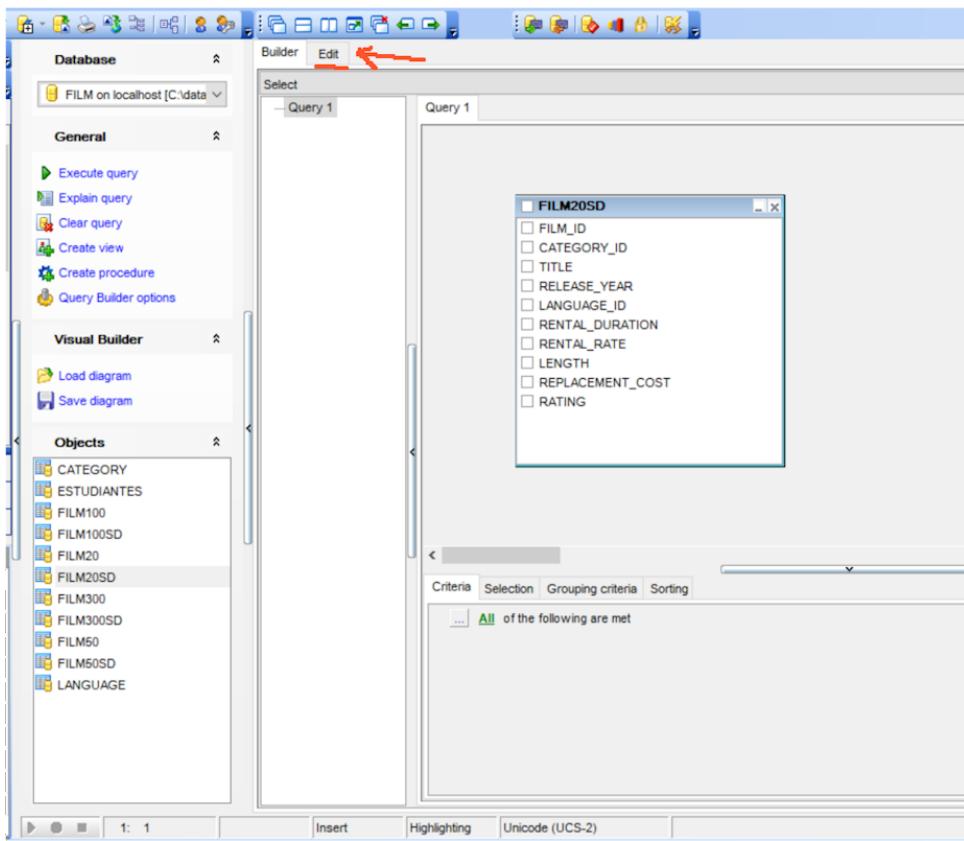


es una forma de construir querys de forma visual , rápida y sencilla. No obstante es sumamente importante que el estudiante pase por todo el material anterior construyendo los querys mediante el Editor de SQL primeramente, es más artesanal y esforzado, pero garantiza al estudiante el entendimiento de lo que hace y debe aprender, de tal modo, que al adquirir la destreza y dominio necesario, pueda transitar a herramientas que facilitan mucho más el trabajo de construir querys, lo contrario implica arriesgarse en caer en un proceso de repetición sin entender el trasfondo del origen y funcionamiento de la construcción de los querys y sus diferentes tipos(select, update, delete e insert).

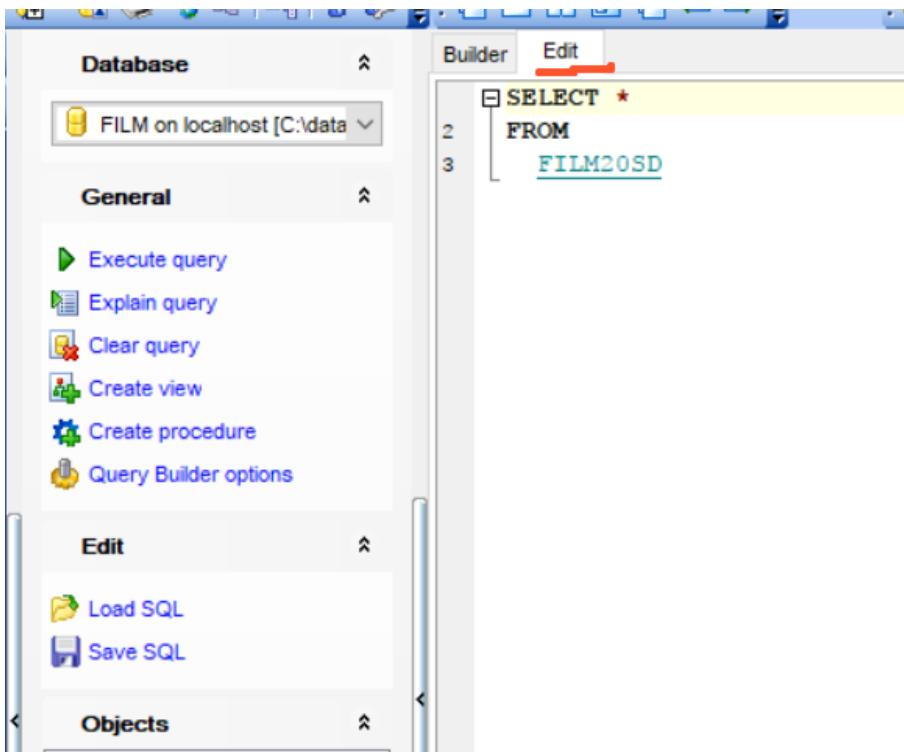
Veamos un ejemplo de construcción de un select con where con el Query Builder(constructor de querys) .



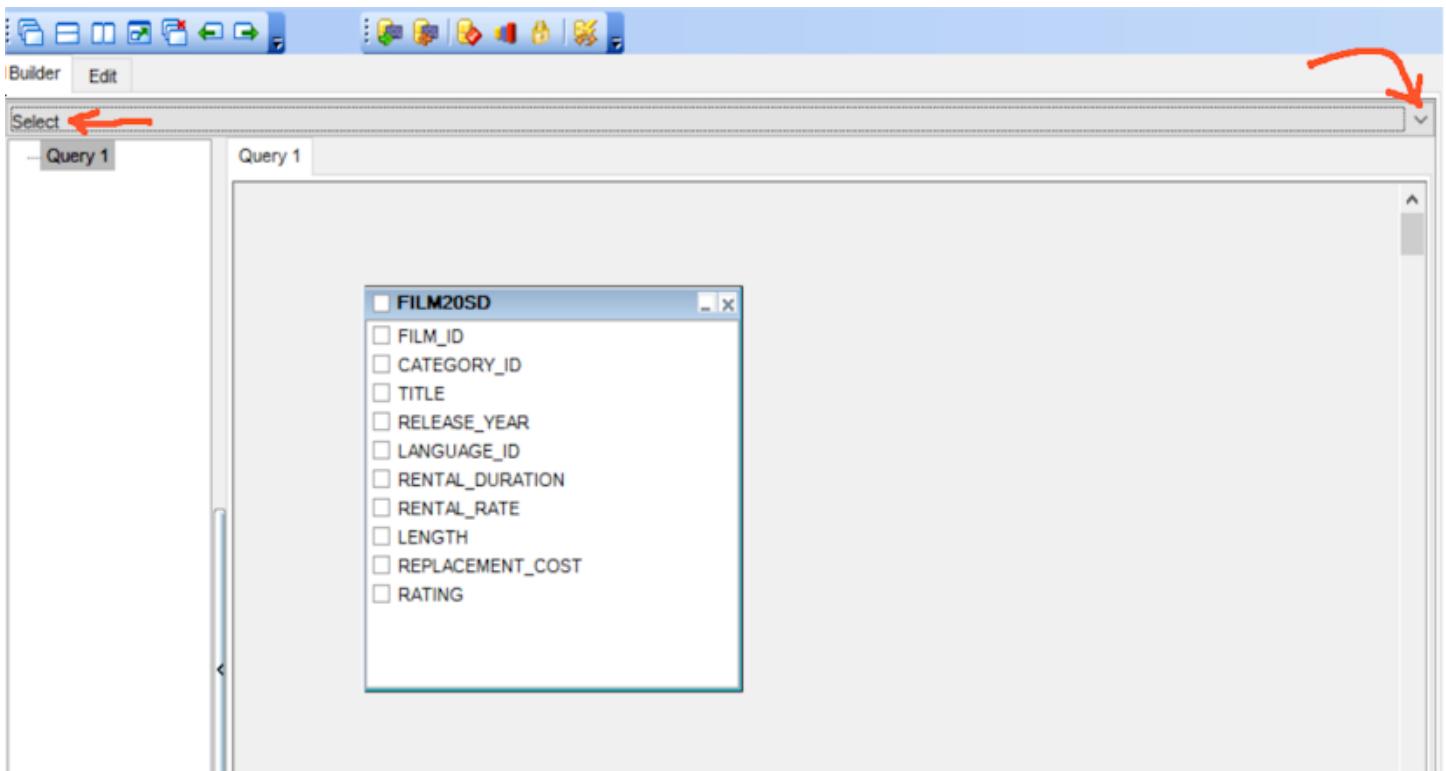
Notemos lo siguiente, ya con el hecho de agregar la tabla a la ventana del constructor, se genera el query select automáticamente, el cual podemos ver al entrar a la pestaña Edit señalada con la flecha roja:



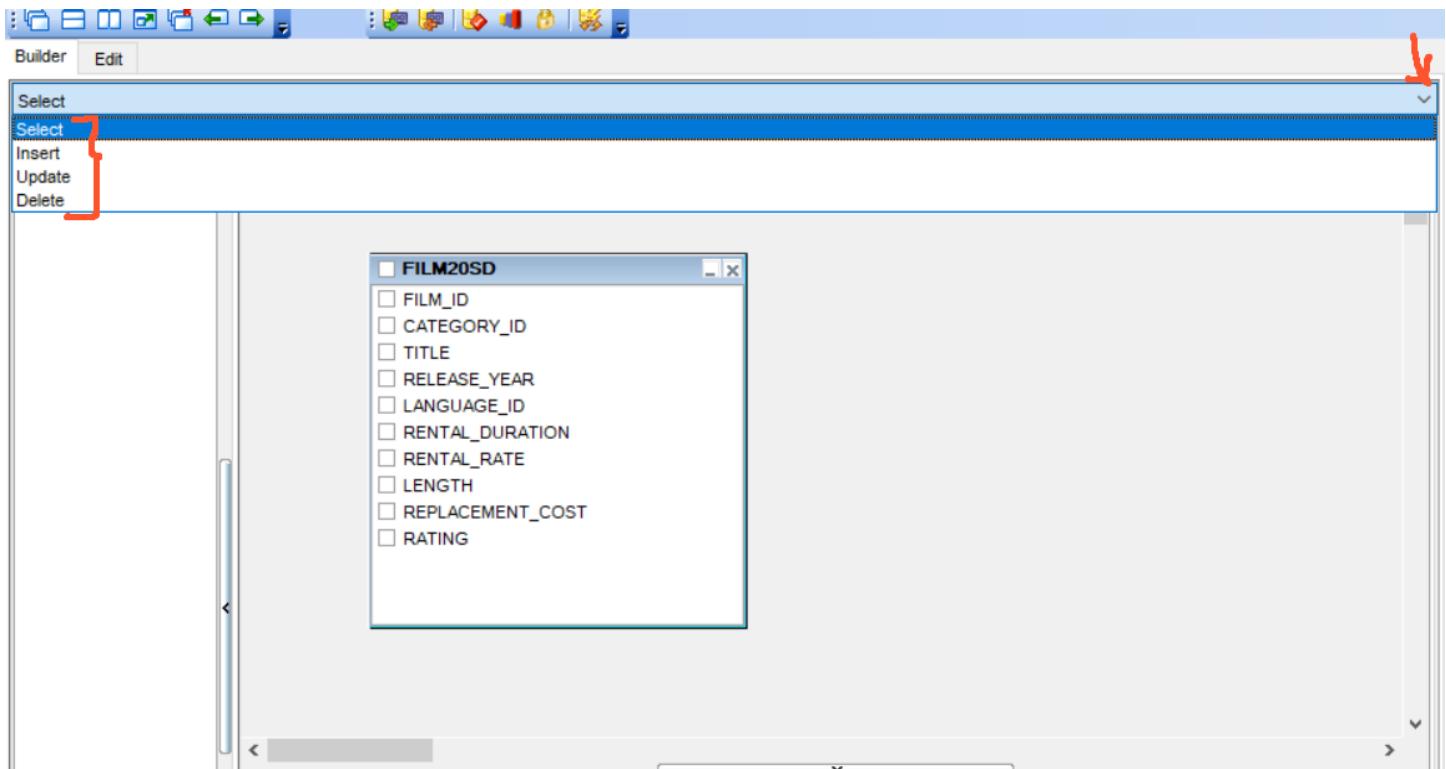
En la pantalla Edit vemos el select construido.



Regresamos a la pantalla Builder: importante, observe la flecha que señala a Select, pero esto puede cambiar a Update, Delete o Insert al presionar en la esquina señalada por la otras flecha:



Aparece una lista de los tipos de querys que puedes construir: select, insert, update y delete, y esto permitirá generar la sintaxis SQL correspondiente. En nuestro caso vamos a trabajar con un query del tipo select (consulta).



Vamos a proceder a seleccionar los campos que deseo incorporar a mi query, pues no deseo que aparezcan todos. Pulsamos la opción Selection, desde el cual iremos seleccionando los campos.

A screenshot of the query builder interface showing the 'Query 1' tab. On the left, there's a 'FILM20SD' field list with the same set of fields as the previous window. Below it, a toolbar has tabs for 'Criteria', 'Selection' (which is highlighted with a red box), 'Grouping criteria', and 'Sorting'. Underneath the toolbar, there's a checkbox for 'Select only unique records'. A table below has columns for 'Source field name', 'Name of output field', 'Aggregate', and 'Grouping'. The 'Source field name' column is currently empty.

Abajo, a medida que hacemos chek en los campos de la tabla, irán apareciendo los campos en la ventana debajo (Source field name) .

The screenshot shows the Oracle SQL Developer interface with the 'Builder' tab selected. In the main pane, a 'Query 1' window is open, displaying a list of fields from the 'FILM20SD' table. Several fields are checked: FILM_ID, CATEGORY_ID, TITLE, RELEASE_YEAR, LANGUAGE_ID, and RENTAL_DURATION. Below this list is a table mapping source field names to output field names:

Source field name	Name of output field	Aggregate	Grouping
FILM20SD.FILM_ID	FILM_ID		
FILM20SD.CATEGORY_ID	CATEGORY_ID		
FILM20SD.TITLE	TITLE		
FILM20SD.LANGUAGE_ID	LANGUAGE_ID		
FILM20SD.RENTAL_DURATION	RENTAL_DURATION		
FILM20SD.RELEASE_YEAR	RELEASE_YEAR		

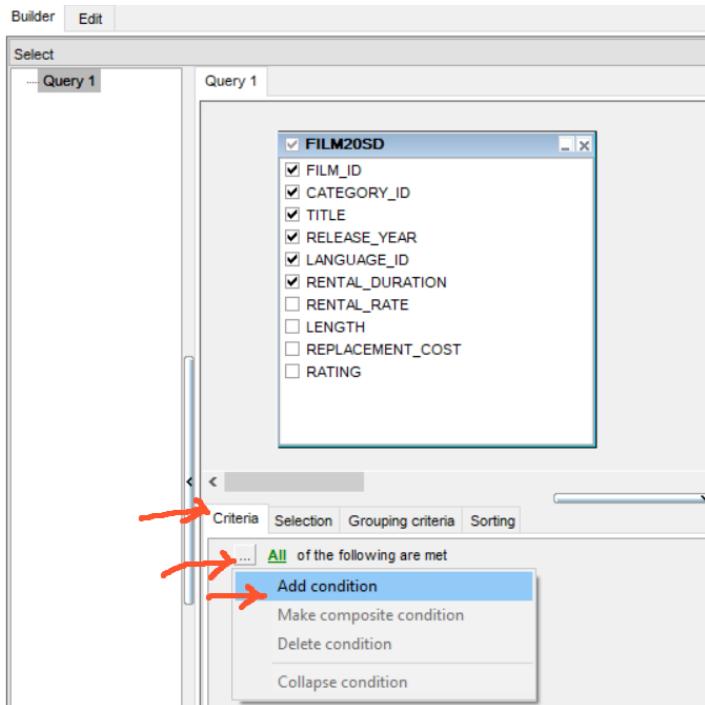
En este caso hemos seleccionado 6 campos, para ver el efecto en la construcción del query vamos nuevamente a la pestaña Edit:

The screenshot shows the Oracle SQL Developer interface with the 'Edit' tab selected. The query editor displays the following SQL code:

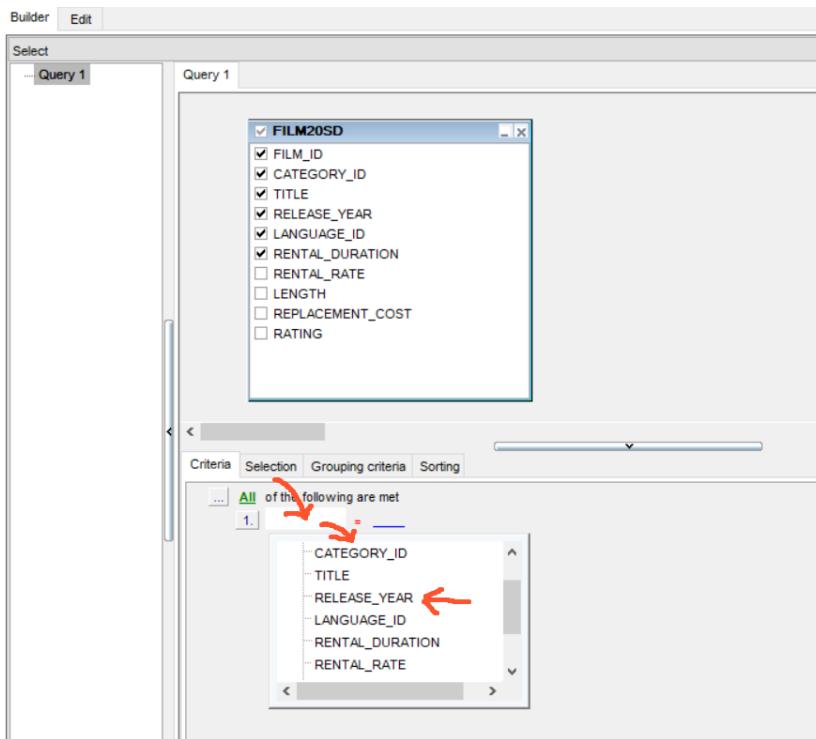
```
1 SELECT
2   FILM20SD.FILM_ID,
3   FILM20SD.CATEGORY_ID,
4   FILM20SD.TITLE,
5   FILM20SD.LANGUAGE_ID,
6   FILM20SD.RENTAL_DURATION,
7   FILM20SD.RELEASE_YEAR
8 FROM
9   FILM20SD
```

Arriba vemos que ha quedado variado el query, se incluye solamente los campos que hemos seleccionado en la pantalla Builder.

Regresamos a Builder. Vamos a crear nuestro WHERE, o sea, agregar condiciones de búsqueda:



Arriba, nos posicionamos en la pestaña Criteria (abajo) dentro de la ventana Builder. Aparecen unos puntos seguidos y pulsamos. Aparece la opción "Add condition" y pulsamos para agregar la primera condición.



Aparece una línea enumerada 1. con 3 partes, en una permite incluir el campo usado que vamos condicionar, luego la operación que vamos aplicar (=,<,>,etc) y luego el cuadro que permite escribir un valor o elegir otro campo. Arriba hemos seleccionado el campo RELEASE_YEAR.

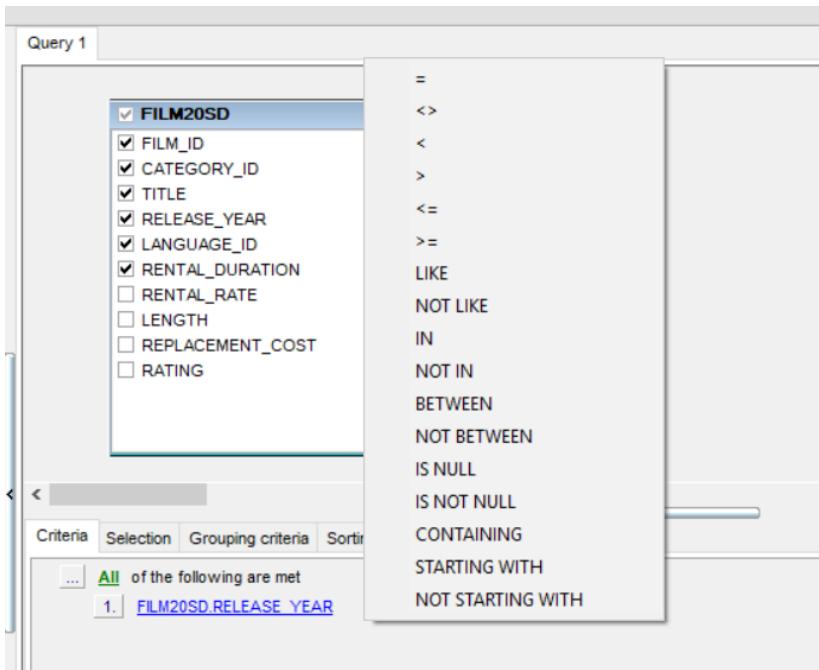
A la derecha hemos optado por escribir el año 200

The screenshot shows the MySQL Workbench Query Builder interface. On the left, there's a tree view with 'Select' and 'Query 1'. The main area contains a table definition for 'FILM20SD' with columns: FILM_ID, CATEGORY_ID, TITLE, RELEASE_YEAR, LANGUAGE_ID, RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, and RATING. Most columns have checkboxes checked. Below the table definition is a 'Criteria' tab with a dropdown menu showing 'All of the following are met'. Under this, there is a condition: '1. FILM20SD.RELEASE_YEAR = 2006'. A red arrow points to the '=' sign.

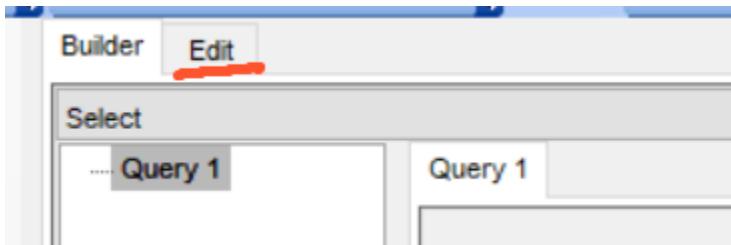
Además podemos cambiar el signo (=) por otra operador de comparación.

This screenshot shows the same MySQL Workbench interface as above, but with a red arrow pointing to the '=' sign in the condition '1. FILM20SD.RELEASE_YEAR = 2006'. The '=' sign is highlighted in blue, indicating it is being selected from a dropdown menu.

Al pulsar sobre el operador, aparece una lista de operadores posibles que podemos seleccionar.



Si vamos a la pestaña Edit, veremos el resultado final



```

1 SELECT
2     FILM20SD.FILM_ID,
3     FILM20SD.CATEGORY_ID,
4     FILM20SD.TITLE,
5     FILM20SD.LANGUAGE_ID,
6     FILM20SD.RENTAL_DURATION,
7     FILM20SD.RELEASE_YEAR
8 FROM
9     FILM20SD
10 WHERE
11     FILM20SD.RELEASE_YEAR = 2006

```

Podemos agregar más condiciones WHERE:

Query 1

FILM20SD

- FILM_ID
- CATEGORY_ID
- TITLE
- RELEASE_YEAR
- LANGUAGE_ID
- RENTAL_DURATION
- RENTAL_RATE
- LENGTH
- REPLACEMENT_COST
- RATING

Criteria Selection Grouping criteria Sorting

... All of the following are met

1. FILM20SD.RELEASE_YEAR = 2006
2. FILM20SD.CATEGORY_ID = 2

```

SELECT
    FILM20SD.FILM_ID,
    FILM20SD.CATEGORY_ID,
    FILM20SD.TITLE,
    FILM20SD.LANGUAGE_ID,
    FILM20SD.RENTAL_DURATION,
    FILM20SD.RELEASE_YEAR
FROM
    FILM20SD
WHERE
    FILM20SD.RELEASE_YEAR = 2006 AND
    FILM20SD.CATEGORY_ID = 2
  
```

Arriba podemos modificar el select y quitar o poner o agregar. Por ejemplo, podemos cambiar AND por OR en el WHERE.

También podemos ordenar:

Criteria Selection Grouping criteria **Sorting**

Up Down

Output fields	Sorted fields	Sort order
FILM20SD.FILM_ID	FILM20SD.CATEGORY_ID	Ascending
FILM20SD.TITLE		
FILM20SD.LANGUAGE_ID		
FILM20SD.RENTAL_DURATION		
FILM20SD.RELEASE_YEAR		
FILM20SD.RENTAL_RATE		
FILM20SD.LENGTH		
FILM20SD.REPLACEMENT_COST		
FILM20SD.RATING		

El query queda:

The screenshot shows the SQL Manager Lite interface for InterBase and Firebird. The main window is divided into several panes:

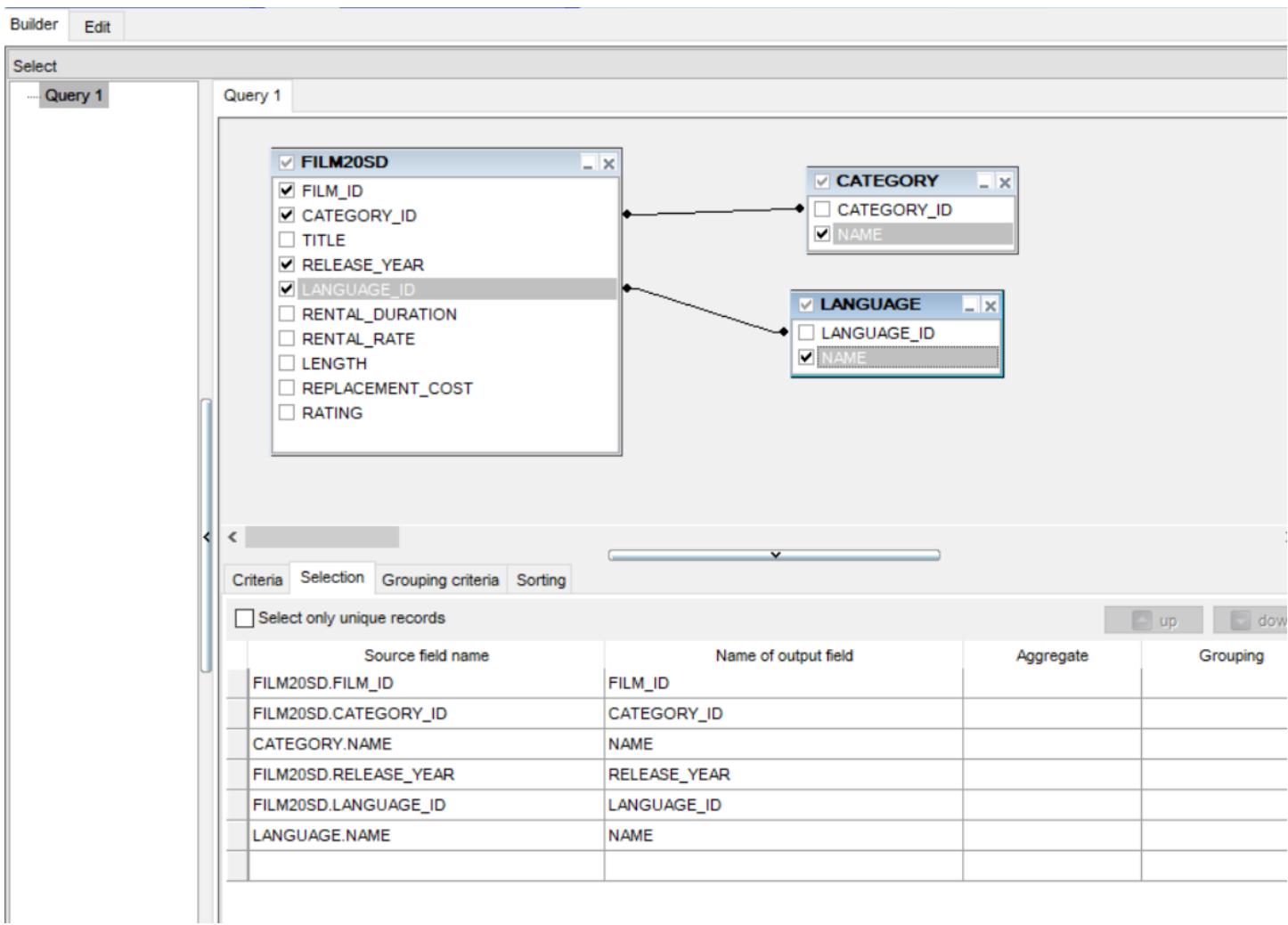
- Left pane (Databases):** Shows the database structure under "localhost". Under "FILM on localhost", there are "Domains", "Tables (11)", and other objects like "CATEGORY", "ESTUDIANTES", etc.
- Bottom-left pane (Table structure):** Displays the structure of the "FILM20SD" table, listing fields such as FILM_ID, CATEGORY_ID, TITLE, RELEASE_YEAR, LANGUAGE_ID, RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, and RATING.
- Center-right pane (Builder):** Contains a "SELECT" query builder. The query is:

```
1 SELECT
2   FILM20SD.FILM_ID,
3   FILM20SD.CATEGORY_ID,
4   FILM20SD.TITLE,
5   FILM20SD.LANGUAGE_ID,
6   FILM20SD.RENTAL_DURATION,
7   FILM20SD.RELEASE_YEAR
8 FROM
9   FILM20SD
10 WHERE
11   FILM20SD.RELEASE_YEAR = 2006 OR
12   FILM20SD.CATEGORY_ID = 2
13 ORDER BY
14   FILM20SD.CATEGORY_ID
```
- Bottom-right pane (Objects):** Shows a list of objects including "CATEGORY", "ESTUDIANTES", "FILM100", "FILM100SD", "FILM20", "FILM20SD", "FILM300", "FILM300SD", "FILM50", "FILM50SD", and "LANGUAGE".

The status bar at the bottom right shows the date and time: 05/30/2017 2:15 p.m.

Se pueden también construir agrupaciones y las operaciones de agrupaciones.

Veamos la construcción de un JOIN.



Arriba hemos arrastrado tres tablas a la pantalla de Builder. Seleccionamos el campo CATEGORY_ID de FILM20SD y lo arrastramos al campo CATEGORY_ID de la tabla CATEGORY. Estamos estableciendo una relación entre las dos tablas. Hacemos lo mismo con los campos LANGUAGE_ID de las dos tablas FILM20SD y LANGUAGE.

Luego, procedemos a seleccionar los campos de cada tabla que deseamos que aparezcan en los resultados. Se observa en la imagen (arriba) el nombre de la tabla y el campo.

Vamos a ver el query que se construye:

```

Builder Edit
SELECT
 1 FILM20SD.FILM_ID,
 2 FILM20SD.CATEGORY_ID,
 3 CATEGORY.NAME,
 4 FILM20SD.RELEASE_YEAR,
 5 FILM20SD.LANGUAGE_ID,
 6 LANGUAGE.NAME
FROM
 7 FILM20SD
 8 INNER JOIN CATEGORY ON (FILM20SD.CATEGORY_ID = CATEGORY.CATEGORY_ID)
 9 INNER JOIN LANGUAGE ON (FILM20SD.LANGUAGE_ID = LANGUAGE.LANGUAGE_ID)
10
11

```

Veamos el caso de un agrupamiento sencillo: entre al tab o pestaña (abajo) Selection

The screenshot shows the MySQL Workbench Query Builder interface. The 'Selection' tab is active. A 'Select' dialog box is open, titled 'Query 1'. It contains a table named 'FILM20SD' with several columns listed. The 'RELEASE_YEAR' column is checked, indicating it is the field being grouped by. Below the table, there are tabs for 'Criteria', 'Selection', 'Grouping criteria', and 'Sorting'. Under 'Grouping criteria', there is a checkbox for 'Select only unique records'. A table below shows the mapping of source fields to output fields, including aggregate functions like SUM and AVG.

Source field name	Name of output field	Aggregate	Grouping
FILM20SD.RELEASE_YEAR	RELEASE_YEAR		Yes
FILM20SD.RENTAL_DURATION	SUMANIO	SUM	
FILM20SD.RENTAL_RATE	PROMRATE	Avg	

Se incluye primero los campos que se van agrupar, en este caso solo RELEASE_YEAR. Luego se incluyen los campos que se les aplicará las operaciones de agregamiento, en esto caso RENTAL_DURATION que es SUM() y RENTAL_RATE que es AVG().

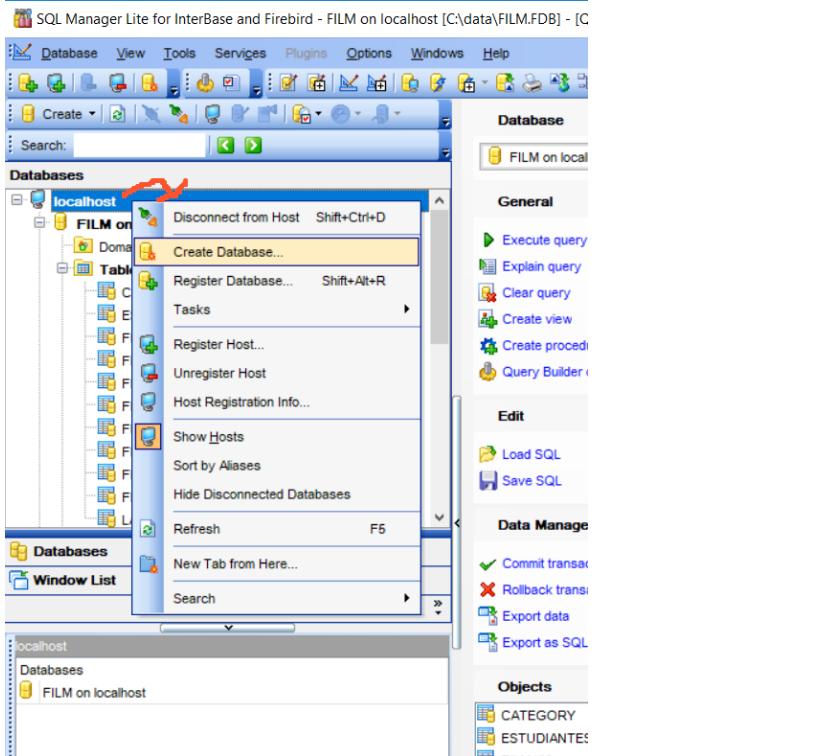
Veamos el query que produce:

The screenshot shows the MySQL Workbench Result tab with the generated SQL query. The query performs a SELECT operation on the 'FILM20SD' table, grouping by 'RELEASE_YEAR' and calculating the sum of 'RENTAL_DURATION' and the average of 'RENTAL_RATE' for each group.

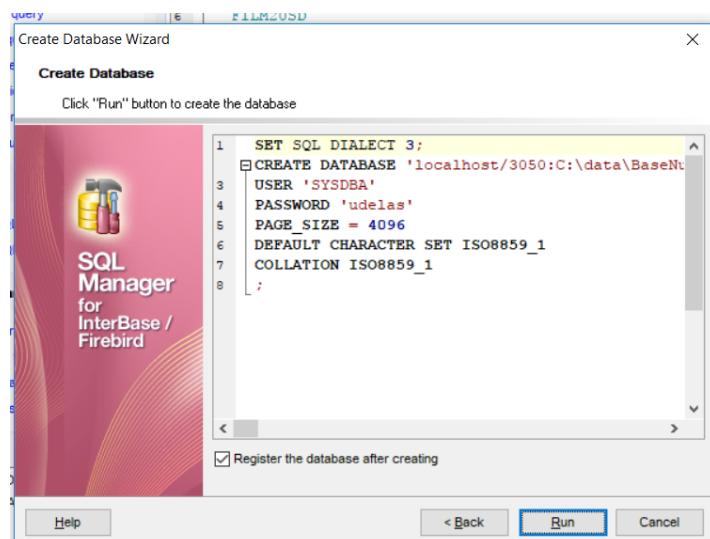
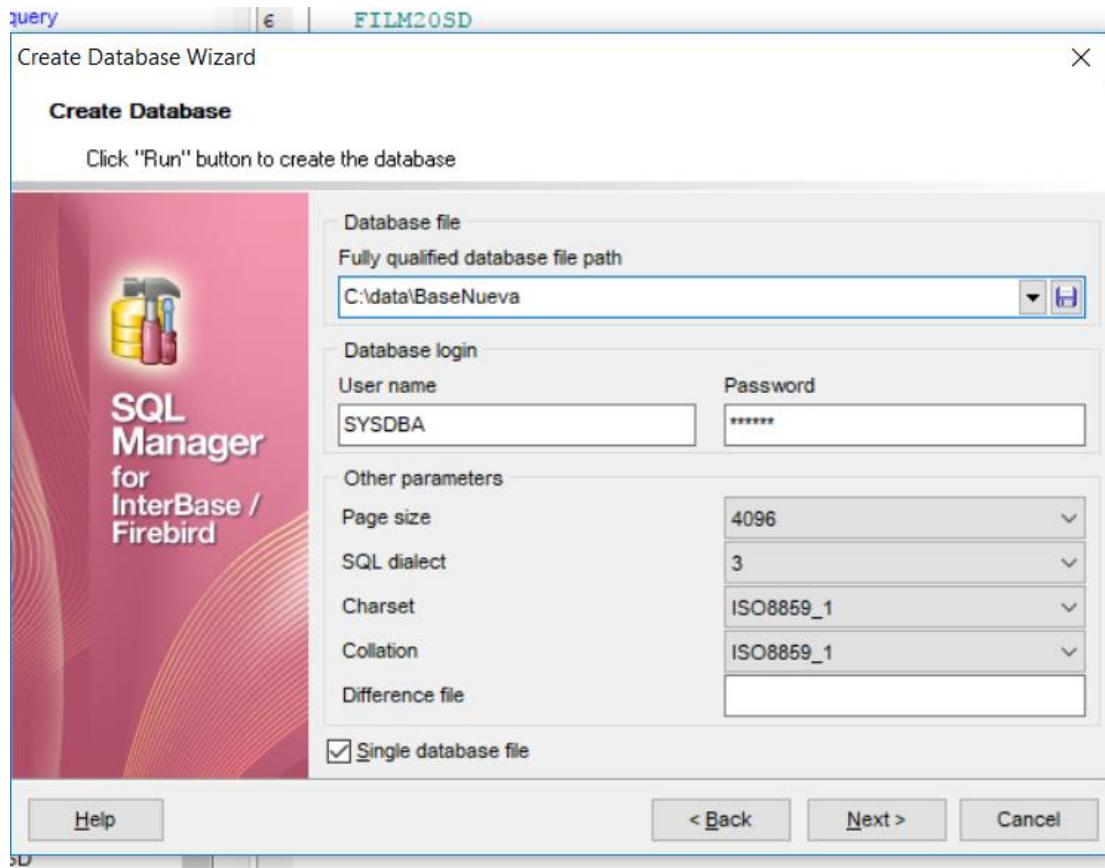
```
1 SELECT
2     FILM20SD.RELEASE_YEAR,
3     SUM(FILM20SD.RENTAL_DURATION) AS SUMANIO,
4     AVG(FILM20SD.RENTAL_RATE) AS PROMRATE
5 FROM
6     FILM20SD
7 GROUP BY
8     FILM20SD.RELEASE_YEAR
```

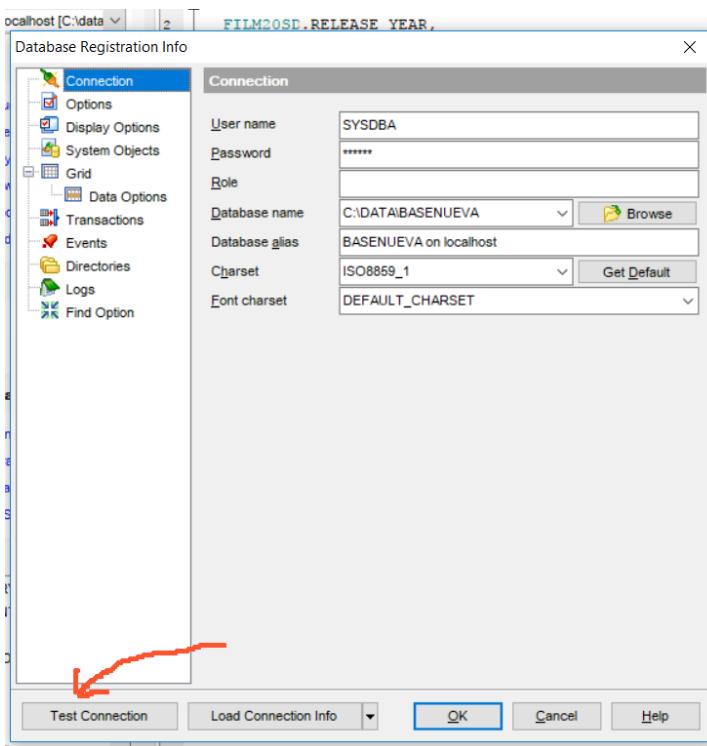
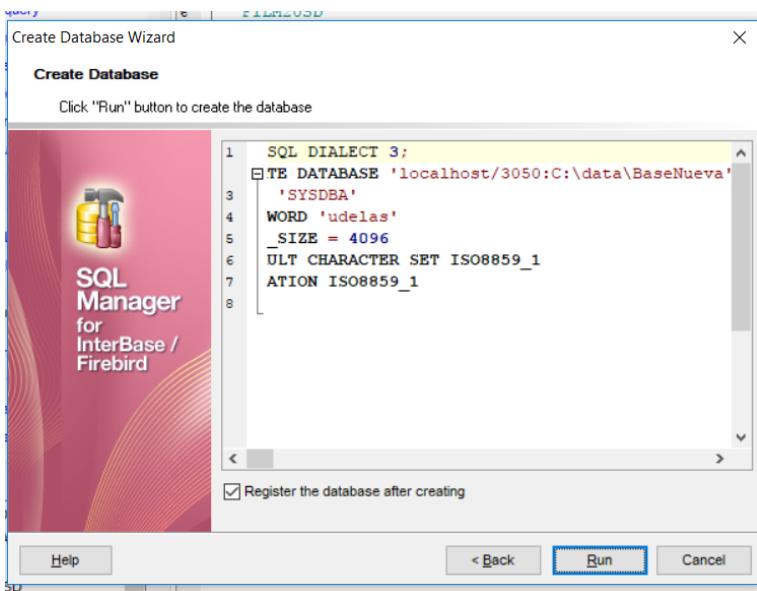
9. Crear y eliminar una base de datos.

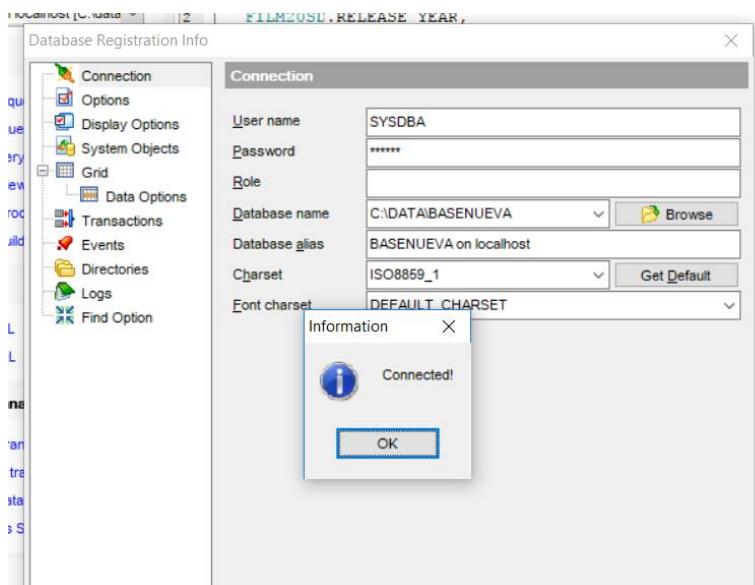
Nos colocamos encima del servidor Host que es localhost, hacemos click derecho y seleccionamos crear Database.



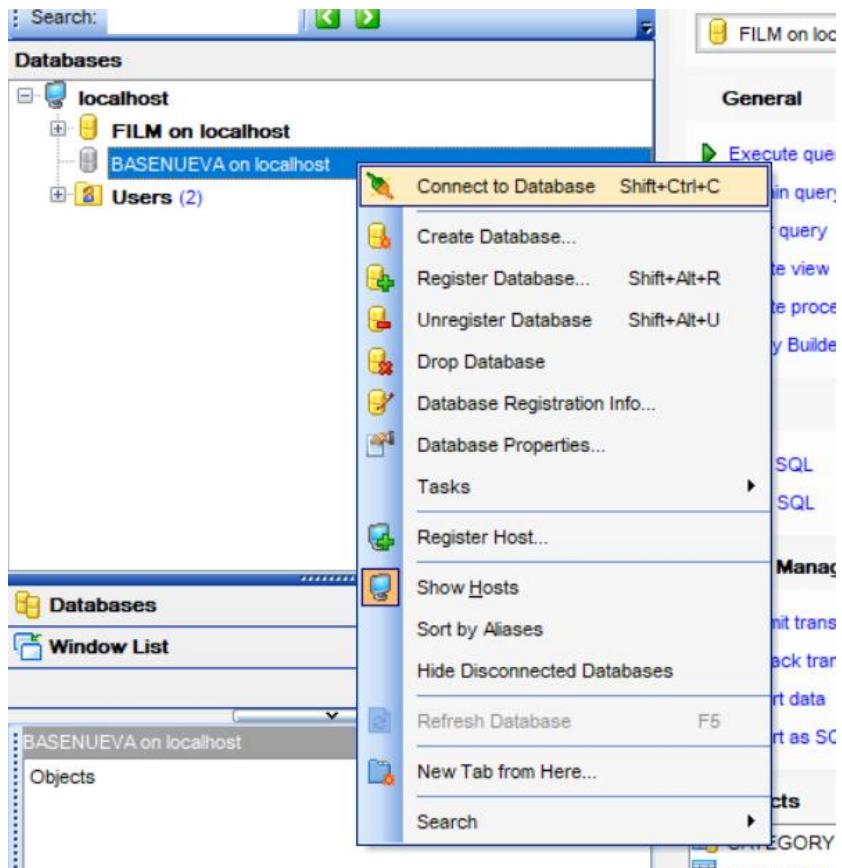
Abajo, escribimos el path y el nombre de la base de datos nueva, el programa crear el path si no existe y crea la base de datos nueva con el nombre que escribimos:BaseNueva

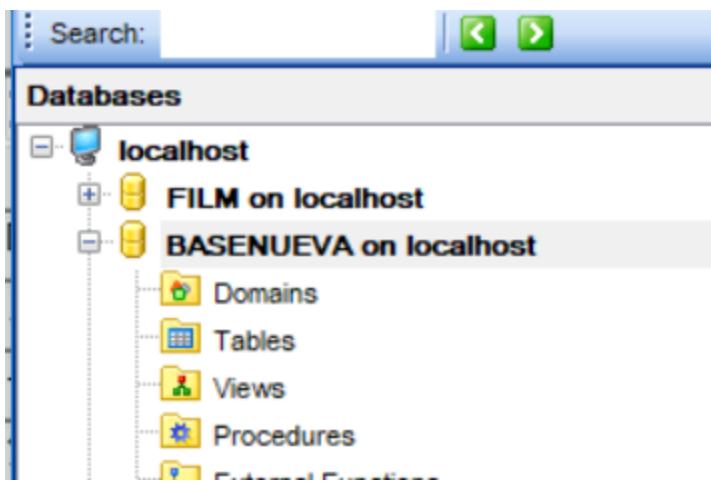






Ya la hemos creado, tenemos que Conectar.

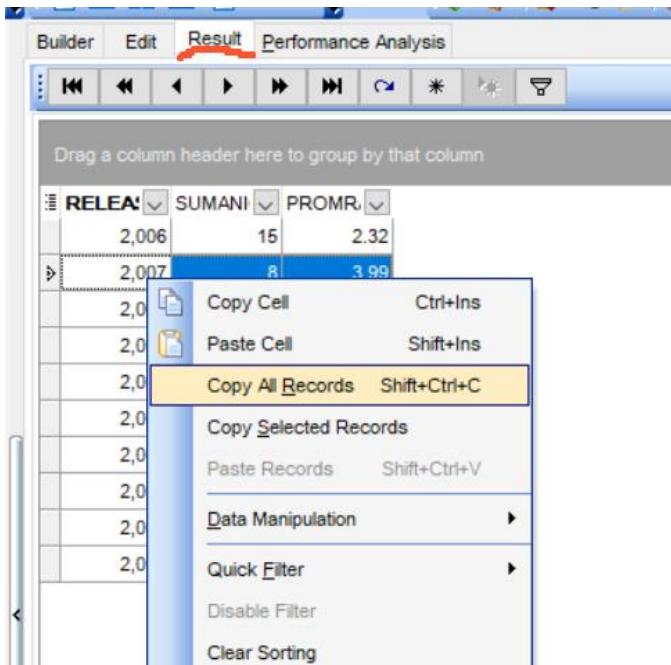




Para eliminar una base de datos, solamente hay que seleccionar la base de datos, click derecho , luego tenemos dos opciones: desregistrar, lo que permite que la base de datos salga de nuestra lista de bases de datos pero no la elimina, y la otra opción es Drop o eliminar la base de datos físicamente.

10. Copiar y pegar los resultados a excell.

Esto se hace desde la ventana de Resultados, nos ponemos encima de cualquier registro de la tabla de resultados y hacemos click derecho:



Al hacer Copy All Records el programa copia al Clipboard. Lo pegamos directamente en excell:

	A	B	C	D	E	F
1						
2						
3						
4			RELEASE_YEAR	SUMANIO	PROMRATE	
5			2006	15	2.32	
6			2007	8	3.99	
7			2008	9	2.99	
8			2010	5	2.99	
9			2011	3	2.99	
10			2012	4	2.99	
11			2013	20	2.99	
12			2014	5	2.99	
13			2015	3	4.99	
14			2016	20	2.99	
15						(Ctrl)
16						
17						

Si pegamos en Notepad plus:

```

*new 3 - Notepad++
Archivo Editar Buscar Vista Codificación idioma Configuración
+ new 2 x new 1 x new 3 x
1
2
3
4
5 "RELEASE_YEAR"    "SUMANIO"      "PROMRATE"
6 "2006"    "15"        "2.32"
7 "2007"    "8"         "3.99"
8 "2008"    "9"         "2.99"
9 "2010"    "5"         "2.99"
10 "2011"   "3"         "2.99"
11 "2012"   "4"         "2.99"
12 "2013"   "20"        "2.99"
13 "2014"   "5"         "2.99"
14 "2015"   "3"         "4.99"
15 "2016"   "20"        "2.99"
16
17

```

Significa que los campos están delimitado por la comilla, esto es importante conocerlo para que se pueda delimitar los campos en otros programas.

Referencias de estudio:

Blog Firebird SQL:

Teoría y Práctica sobre Firebird

<https://firebird21.wordpress.com>

Autor: Walter R. Ojeda Valiente

<https://firebird21.wordpress.com/about/>

Sitio completo:

<https://firebird21.wordpress.com/2013/06/16/el-indice-del-blog-firebird21/>

En este sitio, se muestra con mucho detalles y muchos ejemplos en español, operaciones con Firebird. Las secciones que deben leer al menos por encima y tener en cuenta como una fuente de consulta y aprendizaje autónomo, son:

Ejemplos de SELECTs para principiantes

<https://firebird21.wordpress.com/category/principiantes/ejemplos-de-selects-para-principiantes/>

Ejemplos de SELECTs para intermedios

<https://firebird21.wordpress.com/category/ejemplos-de-selects-para-intermedios/>

Funciones agregadas

<https://firebird21.wordpress.com/category/funciones/funciones-agregadas/>

INSERTs y UPDATEs

<https://firebird21.wordpress.com/category/inserts-y-updates/>

Sobre Join:

<https://firebird21.wordpress.com/2013/07/21/entendiendo-a-los-join/>

<https://firebird21.wordpress.com/2013/12/02/inner-join-y-outer-join/>

<https://firebird21.wordpress.com/2014/04/13/join-implicito-y-join-explicito/>

Otros ejemplos para prácticas:

Ejemplo N° 005 – Usando LEFT JOIN

<https://firebird21.wordpress.com/2013/05/08/ejemplo-no-005-usando-left-join/>

Ejemplo N° 006 – Usando LEFT JOIN e INNER JOIN

<https://firebird21.wordpress.com/2013/05/08/ejemplo-no-006-usando-left-join-e-inner-join/>

Los predicados de comparación

<https://firebird21.wordpress.com/2014/04/27/los-predicados-de-comparacion/>

Usando SIMILAR TO

<https://firebird21.wordpress.com/2014/04/27/usando-similar-to/>

Optimizando los JOIN

<https://firebird21.wordpress.com/2014/05/14/optimizando-los-join/>

Paginando un SELECT

<https://firebird21.wordpress.com/2014/05/29/paginando-un-select/>

Un error de concepto en la cláusula WHERE

<https://firebird21.wordpress.com/2014/09/14/un-error-de-concepto-en-la-clausula-where/>

Poniendo los JOIN en el orden correcto

<https://firebird21.wordpress.com/2015/07/31/poniendo-los-join-en-el-orden-correcto/>

Consultando datos que NO EXISTEN en una tabla

<https://firebird21.wordpress.com/2013/11/29/consultando-datos-que-no-existen-en-una-tabla/>

Entendiendo la cláusula GROUP BY: agrupando datos

<https://firebird21.wordpress.com/2015/09/01/entendiendo-la-clausula-group-by-agrupando-datos/>

La cláusula GROUP BY requiere estar ordenada

<https://firebird21.wordpress.com/2015/09/02/la-clausula-group-by-requiere-estar-ordenada/>

La cláusula HAVING: filtrando las filas agrupadas

<https://firebird21.wordpress.com/2015/09/03/la-clausula-having-filtrando-las-filas-agrupadas/>

Obteniendo la primera fila de cada grupo

<https://firebird21.wordpress.com/2015/12/27/obteniendo-la-primera-fila-de-cada-grupo/>

Hallando la primera palabra

<https://firebird21.wordpress.com/2016/05/12/hallando-la-primera-palabra/>

Hallando la última palabra

<https://firebird21.wordpress.com/2016/05/13/hallando-la-ultima-palabra/>