

Práctica 1

Informe sobre el diseño

En esta primera práctica hemos realizado el modelo de base de datos de una librería con el fin de gestionar el catálogo de los libros, de los autores, las ventas, las ofertas, etc. a partir de algunos requisitos que nos daban.

Comenzamos creando la entidad principal, 'Libro': un libro normalmente se distingue por su título pero se puede dar el caso en el que haya dos o más libros distintos con este mismo título así que por ello añadimos un atributo llamado 'idLibro' que será la clave primaria de la entidad pues nos permitirá distinguir entre todos los libros.

Como un libro puede estar escrito por varios autores y un autor puede escribir más de un libro creamos una entidad 'Autor' que queda relacionada con 'Libro' de la forma n-m. Esta entidad es muy similar a la primera que hemos creado ya que nos encontramos con el mismo problema de que haya varios autores con el mismo nombre y lo solucionamos de la misma manera añadiendo el atributo 'idAutor'.

El primer requisito a la hora de hacer el modelo relacional también nos pide que tengamos en cuenta las múltiples ediciones de un mismo título. Entonces creamos una entidad llamada 'Edicion' relacionada de manera n-m con 'Libro' en la cual añadimos atributos como su idioma, precio, editor, si es de tapa dura, de bolsillo, etc. En esta entidad hay dos claves primarias que son el ISBN y el número de edición ya que un libro concreto mantendrá su ISBN aunque su edición varíe.

Una librería tiene ofertas, y por ello creamos otra entidad 'Oferta' relacionada con libro de la forma 1-n ya que varios libros pueden encontrarse en oferta pero no pueden tener más de una oferta a la vez. En esta entidad hemos añadido como atributos el nombre de la oferta, su fecha de inicio y de fin, y un 'Id' que hará de clave primaria a la par que de registro de todas las ofertas que se van haciendo.

Nuestro tercer requisito nos habla de usuarios fidelizados registrados cada uno con su identificador y una única tarjeta de crédito. A continuación creamos una nueva entidad denominada 'Usuario' en la que introducimos de momento estos atributos. Estos usuarios fidelizados reciben un descuento del 10% en cada libro que compren, incluidos los libros que están en oferta, y la librería quiere tener un registro de lo que gasta cada usuario así que añadimos un nuevo atributo llamado 'registro' y creamos nuestra última entidad 'Venta' relacionada de forma n-1 con 'Usuario'. En esta entidad guardamos todas las compras que han realizado los usuarios bajo un identificador que llamamos 'IdVenta' y el dinero gastado en cada compra, con los descuentos añadidos, bajo el nombre de 'importe'.

Otras personas que no son usuarios pueden comprar libros, así que para no tener que crear otra entidad añadimos un atributo 'UsuarioFidelizado' a 'Usuario' que nos dice si el comprador lo es o no. Finalmente tenemos que añadir nuevos atributos a la entidad venta como son la fecha y si se ha usado o no tarjeta de crédito, para tener toda la información necesaria sobre el comprador.

Cabeceras del modelo “tontorrón”:

Libro(título, idLibro)
Autor(Nombre, idAutor)
Edición(ISBN, num, Editor, Idioma, precio, tapaDura, deBolsillo)
Usuario(Id, ccard, registro, UsuarioFidelizado)
Venta(fecha, importe, pagoTarjeta, idVenta)
Oferta(inicio, fin, nombre, Id)
escritoPor(idLibro, idAutor)
tiene(idLibro, ISBN, numEdicion)
compra(idLibro, idUsuario)
efectua(idVenta, idUsuario)
seAplica(idOferta, idLibro)

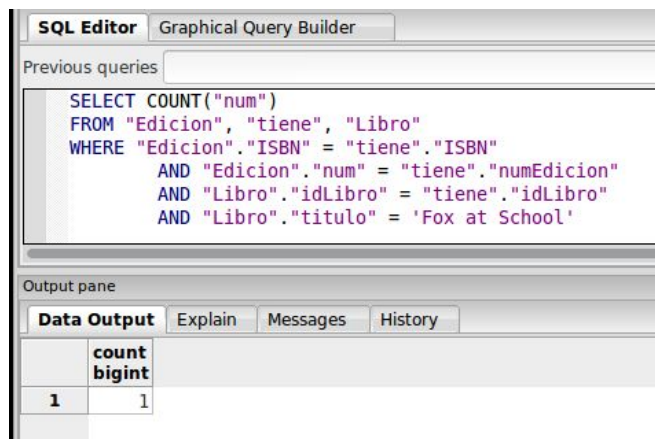
Cabeceras del modelo “optimizado”:

Libro(título, idLibro, idOferta, idUsuario)
Autor(Nombre, idAutor)
Edición(ISBN, num, Editor, Idioma, precio, tapaDura, deBolsillo)
Usuario(Id, ccard, registro, UsuarioFidelizado)
Venta(fecha, importe, pagoTarjeta, idVenta, idUsuario)
Oferta(inicio, fin, nombre, Id)
escritoPor(idLibro, idAutor)
tiene(idLibro, ISBN, numEdicion)

Consultas:

1. Dado un título, ¿Cuántas ediciones tiene? ¿En cuántos idiomas?

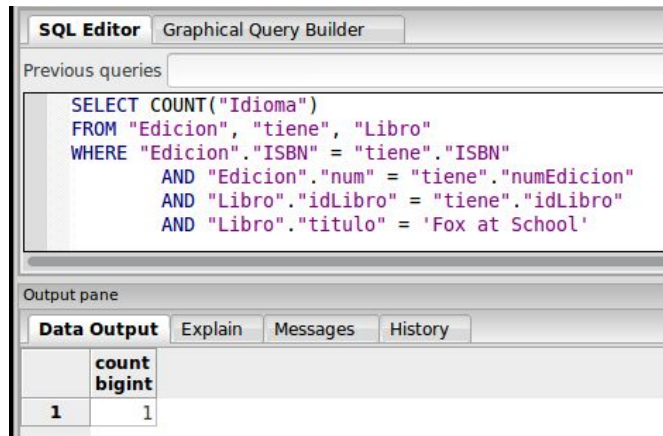
Para realizar estas primeras consultas hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y hemos utilizado la instrucción 'COUNT' de SQL para obtener el número de ediciones y de idiomas de un libro dado su título en vez de una tabla con todos ellos.



The screenshot shows the SQL Editor interface with a query that counts the number of editions for a book with the title 'Fox at School'. The query uses a JOIN between the 'Edicion' and 'Libro' tables via the 'tiene' relationship. The output pane shows a single result: 1 edition.

```
SELECT COUNT("num")
FROM "Edicion", "tiene", "Libro"
WHERE "Edicion"."ISBN" = "tiene"."ISBN"
      AND "Edicion"."num" = "tiene"."numEdicion"
      AND "Libro"."idLibro" = "tiene"."idLibro"
      AND "Libro"."titulo" = 'Fox at School'
```

| | count bigint |
|---|-----------------|
| 1 | 1 |



The screenshot shows the SQL Editor interface with a query that counts the number of languages for a book with the title 'Fox at School'. The query uses a JOIN between the 'Edicion' and 'Libro' tables via the 'tiene' relationship. The output pane shows a single result: 1 language.

```
SELECT COUNT("Idioma")
FROM "Edicion", "tiene", "Libro"
WHERE "Edicion"."ISBN" = "tiene"."ISBN"
      AND "Edicion"."num" = "tiene"."numEdicion"
      AND "Libro"."idLibro" = "tiene"."idLibro"
      AND "Libro"."titulo" = 'Fox at School'
```

| | count bigint |
|---|-----------------|
| 1 | 1 |

2. ¿Cuántos libros se han vendido de un autor dado?

Para realizar esta segunda consulta hemos cruzado las entidades 'Libro' y 'Autor' mediante la relación 'escritoPor' y hemos utilizado la instrucción 'COUNT' de SQL para obtener el número de libros dado un autor en vez de una tabla con todos ellos.

The screenshot shows the SQL Editor interface with a query that counts the number of books sold by a specific author. The query is as follows:

```
SELECT COUNT("titulo")
FROM "Autor", "escritoPor", "Libro"
WHERE "escritoPor"."idLibro" = "Libro"."idLibro"
AND "escritoPor"."idAutor" = "Autor"."Id"
AND "Autor"."Nombre" = 'Vargas Llosa'
```

The Output pane shows the results of the query:

| | count bigint |
|---|-----------------|
| 1 | 1 |

3. ¿Cuántos libros de un autor dado se han vendido en oferta?

Para realizar esta tercera consulta hemos cruzado las entidades 'Libro' y 'Autor' mediante la relación 'escritoPor' y las entidades 'Libro' y 'Oferta' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos utilizado la instrucción 'COUNT' de SQL para obtener el número de libros vendidos en oferta dado un título en vez de una tabla con todos ellos.

The screenshot shows the SQL Editor interface with a query that counts the number of books sold on offer by a specific author. The query is as follows:

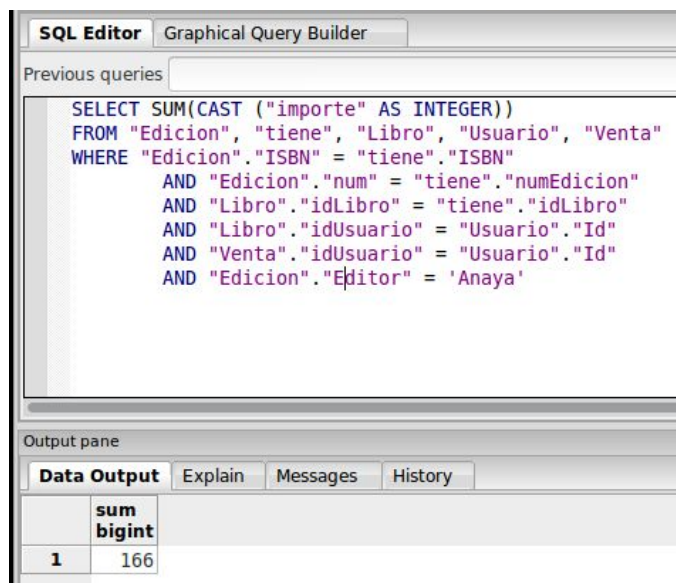
```
SELECT COUNT("titulo")
FROM "Autor", "escritoPor", "Libro", "Oferta"
WHERE "escritoPor"."idLibro" = "Libro"."idLibro"
AND "escritoPor"."idAutor" = "Autor"."Id"
AND "Libro"."idOferta" = "Oferta"."Id"
AND "Autor"."Nombre" = 'García Márquez'
```

The Output pane shows the results of the query:

| | count bigint |
|---|-----------------|
| 1 | 1 |

4. ¿Cuánto dinero se ha ganado vendiendo libros de un editor dado?

Para realizar esta cuarta consulta hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y las entidades 'Libro' y 'Usuario', y 'Usuario' y 'Venta' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos utilizado la instrucción 'SUM' de SQL para obtener el dinero total obtenido de vender libros dado su editor en vez de una tabla con todos ellos. También nos hemos visto en la necesidad de utilizar la instrucción 'CAST AS' de SQL pues habíamos definido el atributo 'importe' como un 'character varying' y la instrucción 'SUM' solo puede realizar la cuenta con enteros.



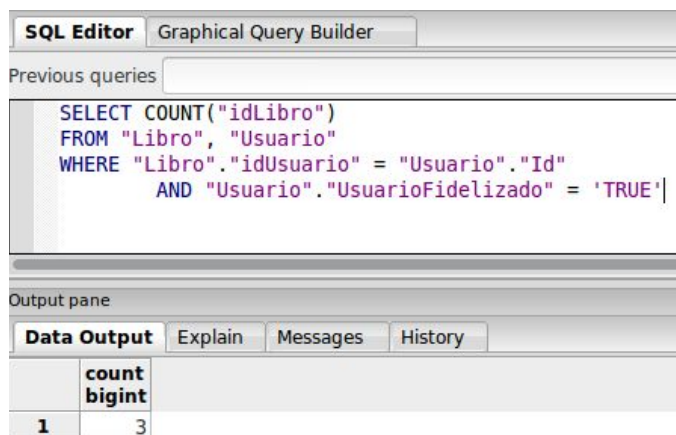
The screenshot shows the SQL Editor interface with a query to calculate the total revenue for the 'Anaya' publisher. The query uses the SUM function with a CAST to convert the 'importe' attribute to an integer. The output pane shows a single result row with a sum of 166.

```
SELECT SUM(CAST ("importe" AS INTEGER))
FROM "Edicion", "tiene", "Libro", "Usuario", "Venta"
WHERE "Edicion"."ISBN" = "tiene"."ISBN"
AND "Edicion"."num" = "tiene"."numEdicion"
AND "Libro"."idLibro" = "tiene"."idLibro"
AND "Libro"."idUsuario" = "Usuario"."Id"
AND "Venta"."idUsuario" = "Usuario"."Id"
AND "Edicion"."Editor" = 'Anaya'
```

| | sum bigint |
|---|---------------|
| 1 | 166 |

5. ¿Cuántos libros han comprado los usuarios fidelizados?

Para realizar esta quinta consulta hemos cruzado las entidades 'Libro' y 'Usuario' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos utilizado la instrucción 'COUNT' de SQL para obtener el número de libros que han sido comprados por usuarios fidelizados en vez de una tabla con todos ellos.



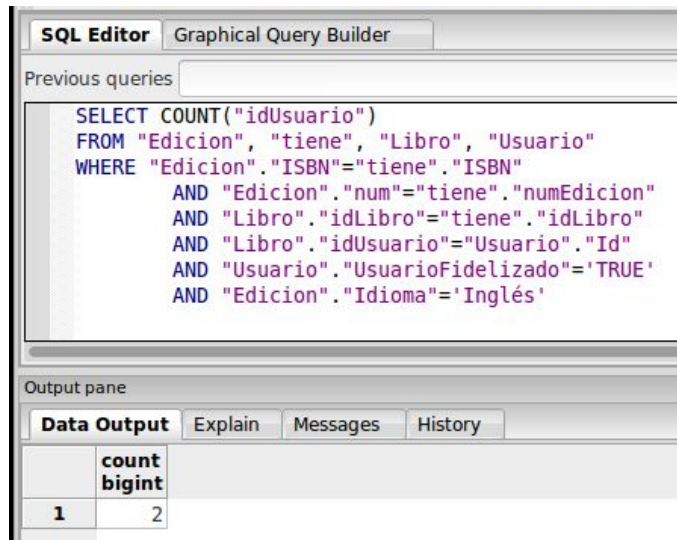
The screenshot shows the SQL Editor interface with a query to count the number of books purchased by loyalized users. The query uses the COUNT function to count the 'idLibro' attribute. The output pane shows a single result row with a count of 3.

```
SELECT COUNT("idLibro")
FROM "Libro", "Usuario"
WHERE "Libro"."idUsuario" = "Usuario"."Id"
AND "Usuario"."UsuarioFidelizado" = 'TRUE'
```

| | count bigint |
|---|-----------------|
| 1 | 3 |

6. ¿Cuántos usuarios fidelizados han comprado libros en inglés?

Para realizar esta sexta consulta hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y las entidades 'Libro' y 'Usuario' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos utilizado la instrucción 'COUNT' de SQL para obtener el número de libros que han sido comprados por usuarios fidelizados en inglés en vez de una tabla con todos ellos.



The screenshot shows the SQL Editor interface with a query that counts the number of users who have purchased books in English. The query uses a JOIN between the 'Edicion' and 'Libro' tables via the 'tiene' relationship, and filters for users who are 'fidelizados' (loyal) and books in 'Inglés'.

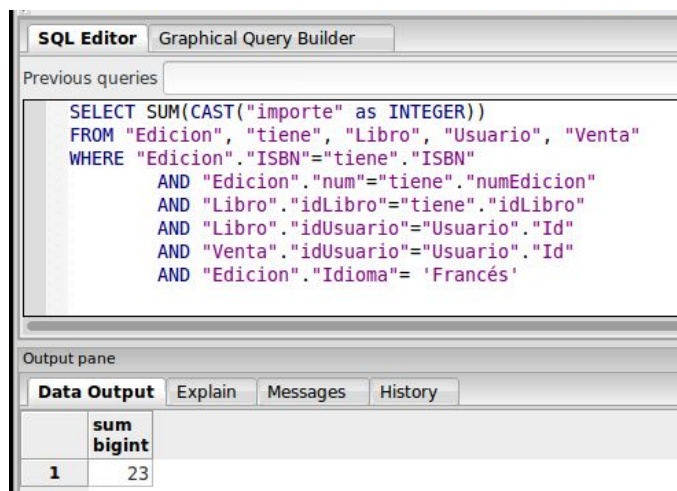
```
SELECT COUNT("idUsuario")
FROM "Edicion", "tiene", "Libro", "Usuario"
WHERE "Edicion"."ISBN"="tiene"."ISBN"
      AND "Edicion"."num"="tiene"."numEdicion"
      AND "Libro"."idLibro"="tiene"."idLibro"
      AND "Libro"."idUsuario"="Usuario"."Id"
      AND "Usuario"."UsuarioFidelizado"='TRUE'
      AND "Edicion"."Idioma"='Inglés'
```

The Output pane shows the result of the query:

| | count bigint |
|---|-----------------|
| 1 | 2 |

7. ¿Cuánto dinero se ha ganado vendiendo libros en Francés?

Para realizar esta séptima consulta hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y las entidades 'Libro' y 'Usuario', y 'Usuario' y 'Venta' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos utilizado la instrucción 'SUM' de SQL para obtener el dinero total obtenido de vender libros en francés en vez de una tabla con todos ellos. También nos hemos visto en la necesidad de utilizar la instrucción 'CAST AS' de SQL pues habíamos definido el atributo 'importe' como un 'character varying' y la instrucción 'SUM' solo puede realizar la cuenta con enteros.



The screenshot shows the SQL Editor interface with a query that sums the revenue from selling books in French. The query uses a JOIN between the 'Edicion', 'Libro', 'Usuario', and 'Venta' tables, and filters for books in 'Francés'.

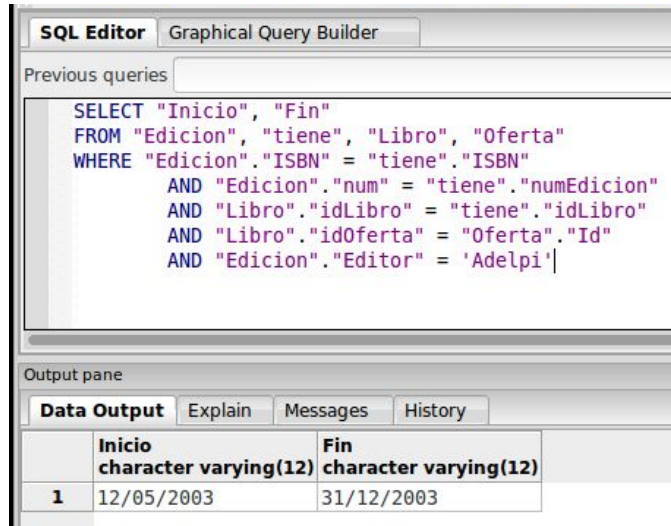
```
SELECT SUM(CAST("importe" as INTEGER))
FROM "Edicion", "tiene", "Libro", "Usuario", "Venta"
WHERE "Edicion"."ISBN"="tiene"."ISBN"
      AND "Edicion"."num"="tiene"."numEdicion"
      AND "Libro"."idLibro"="tiene"."idLibro"
      AND "Libro"."idUsuario"="Usuario"."Id"
      AND "Venta"."idUsuario"="Usuario"."Id"
      AND "Edicion"."Idioma"='Francés'
```

The Output pane shows the result of the query:

| | sum bigint |
|---|---------------|
| 1 | 23 |

8. ¿En qué días hubo ofertas de libros de la editorial Adelphi?

Para realizar esta octava consulta hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y las entidades 'Libro' y 'Oferta' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Hemos buscado las fechas de inicio y de fin en las que hubo una oferta que afectaba a los libros de la editorial Adelphi.



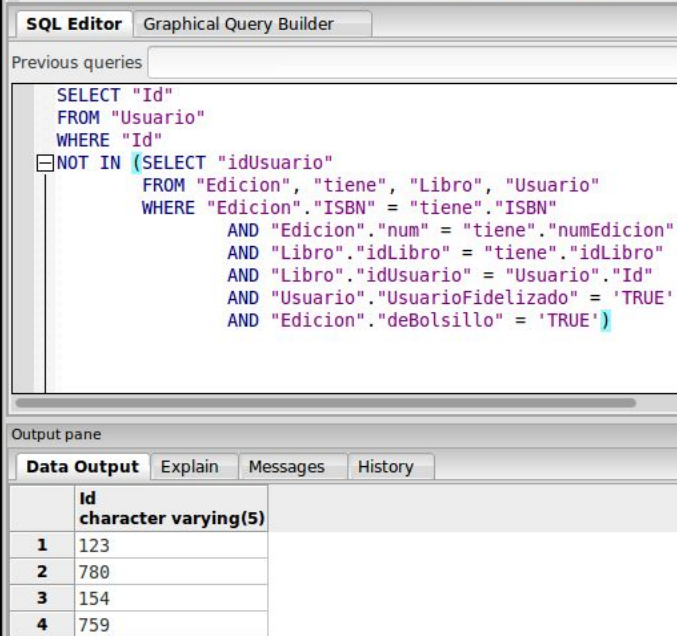
The screenshot shows an SQL Editor window with two tabs: 'SQL Editor' and 'Graphical Query Builder'. The 'SQL Editor' tab is active, displaying a SQL query. Below the query is an 'Output pane' with four sub-tabs: 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' sub-tab is active, showing a table with two columns: 'Inicio' and 'Fin'. The 'Inicio' column is labeled 'character varying(12)' and the 'Fin' column is labeled 'character varying(12)'. There is one row of data with the values '12/05/2003' and '31/12/2003'.

```
SELECT "Inicio", "Fin"
FROM "Edicion", "tiene", "Libro", "Oferta"
WHERE "Edicion"."ISBN" = "tiene"."ISBN"
      AND "Edicion"."num" = "tiene"."numEdicion"
      AND "Libro"."idLibro" = "tiene"."idLibro"
      AND "Libro"."idOferta" = "Oferta"."Id"
      AND "Edicion"."Editor" = 'Adelphi'
```

| | Inicio character varying(12) | Fin character varying(12) |
|---|---------------------------------|------------------------------|
| 1 | 12/05/2003 | 31/12/2003 |

9. ¿Qué usuarios fidelizados no han comprado nunca libros de bolsillo?

Para realizar esta última consulta hemos utilizado únicamente la entidad 'Usuario'. Hemos utilizado la instrucción 'NOT IN' de SQL para obtener una tabla con los usuarios fidelizados que no habían comprado nunca libros de bolsillo. Hemos realizado una subconsulta en la cual hemos cruzado las entidades 'Libro' y 'Edicion' mediante la relación 'tiene' y las entidades 'Libro' y 'Usuario' sin necesidad de establecer una relación pues ésta ha sido simplificada en el modelo optimizado. Aquí hemos buscado a aquellos usuarios fidelizados que sí habían comprado libros de bolsillo, y con el 'NOT IN' obtenemos justo lo que buscábamos.



The screenshot shows an SQL Editor window with two tabs: 'SQL Editor' and 'Graphical Query Builder'. The 'SQL Editor' tab is active, displaying the following SQL query:

```
SELECT "Id"
FROM "Usuario"
WHERE "Id"
NOT IN (SELECT "idUsuario"
FROM "Edicion", "tiene", "Libro", "Usuario"
WHERE "Edicion"."ISBN" = "tiene"."ISBN"
AND "Edicion"."num" = "tiene"."numEdicion"
AND "Libro"."idLibro" = "tiene"."idLibro"
AND "Libro"."idUsuario" = "Usuario"."Id"
AND "Usuario"."UsuarioFidelizado" = 'TRUE'
AND "Edicion"."deBolsillo" = 'TRUE')
```

Below the query editor is the 'Output pane' with four tabs: 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is active, showing a table with the results of the query:

| | Id character varying(5) |
|---|----------------------------|
| 1 | 123 |
| 2 | 780 |
| 3 | 154 |
| 4 | 759 |