

UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”

INTEGRANTES:

FECHA: 30/08/2021

- *Bonifaz Christian*
- *Calderon Mateo*
- *Guano Ariel*
- *Mera Erick*
- *Rosas Mario*

CARRERA:

Electrónica y Automatización

NRC:

3725

ASIGNATURA:

Programación Orientada a Objetos

TEMA:

Bases de Datos

OBJETIVOS:

- Manejar de manera correcta el programa de MySQL.
- Aplicar los debidos procesos para la extracción de información de la base de datos según lo pedido en informe.
- Conectar las tablas de bases de Datos y manejarlas de manera correcta.
- Implementar los conocimientos de clase para el desarrollo del informe.

MARCO TEÓRICO:

Generación de consultas en MySQL

Lo primero que debemos tener claro es que las consultas se refieren a la acción de obtener registros almacenados previamente en la Base de Datos. Para consultar registros de las tablas de nuestra base de datos debemos usar la instrucción select, este comando sql nos permite obtener registros de una o varias tablas. Para que las consultas nos devuelvan resultados debemos haber insertado datos a las tablas de nuestra base de datos y por supuesto haber creado dichas tablas.

- La sintaxis básica para consultar datos en Mysql es la siguiente:

SELECT ListaColumnas **FROM** Nombre_Tabla;

En donde:

- **SELECT:** es el comando que se utiliza para obtener registros de las tablas.
- **Lista Columnas:** hace referencia a los nombres de las columnas de las cuales queremos consultar registros.
- **From:** es el comando para especificar la tabla de la cual vamos a consultar los datos.
- **Nombre_Tabla:** hace referencia al nombre de la tabla de la cual queremos consultar los registros.

Cláusula Where en MySql

La cláusula WHERE nos permite condicionar las consultas con relación a los registros de una o varias columnas que se especifiquen en la sentencia, en tal sentido todos los registros que entran en el filtro del where se mostrarían en los resultados que obtendremos. La cláusula where es utilizada en los casos que no necesitamos que nos muestre todos los registros de una tabla, sino que únicamente los registros que cumplan ciertas condiciones. Las condiciones conllevan expresiones lógicas que se comprueban con la cláusula where, el valor que devuelven las comparaciones es un valor TRUE o FALSE, dependiendo del cumplimiento de la condición especificada. Podemos hacer uso de cualquier expresión lógica o algún operador como los siguientes:

> “Mayor”

>= “Mayor o igual”

< “Menor”

<= “Menor o igual”

= “Igual”

<> o != “Distinto”

IS [NOT] NULL “para validar si el valor de una columna no es nulo, es decir, si contiene o no contiene algún registro”

Acceso a una base de datos en MySQL

MySQL es el sistema gestor de bases de datos de código abierto más popular en la actualidad. MySQL está disponible para distintos sistemas operativos, como Linux, Mac OS X, Solaris, Windows y otros más. MySQL es muy popular en el desarrollo de aplicaciones web, ya que forma parte como sistema gestor de bases de datos de las plataformas LAMP, BAMP, MAMP y WAMP.

Habitualmente solemos conectar nuestra base de datos creada en el Cpanel a una aplicación subida en servidor, como puede ser una web o un gestor de contenidos. En otras ocasiones, necesitamos que la aplicación que se conecte a la base de datos de nuestro alojamiento de forma remota, esté en un servidor externo o en local, en nuestro propio ordenador.

Para dar acceso remoto a nuestra base de datos MySQL deberemos entrar en nuestro panel de control Cpanel, y en la sección Bases de Datos entrar en la opción Acceso Remoto a Bases de Datos MySQL. (MySQL remoto®)

Nos aparecerá un casillero para introducir el host deseado, Añadir Host de Acceso, y es en ese lugar donde deberemos introducir el host desde el cual queremos acceder remotamente. Tenemos varias opciones:

- Acceso desde una IP. Podemos introducir la IP de nuestro público de nuestro ordenador si queremos acceder desde una aplicación instalada en local, o la IP del servidor donde esté instalada esa misma aplicación si es un servidor diferente.

Por ejemplo: 82.194.73.160

- Acceso desde un rango de IPs. Si son varias IPs las que deben tener acceso a esa base de datos, y están dentro de un mismo rango podemos introducir un parámetro a modo de comodín para que con una sola orden podamos dar acceso a todas esas direcciones IP. El símbolo de porcentaje es el que funciona a modo de comodín.

Ejemplo: 82.194.%.%

- Acceso desde cualquier IP. Para dar acceso total a nuestra base de datos podemos utilizar el símbolo del porcentaje "%" a modo de comodín. Hay que utilizar esta opción con cuidado y debemos tener en cuenta que a pesar de permitir el acceso, se deberá conocer el nombre de la base de datos, y el usuario y contraseña de la misma para poder acceder.

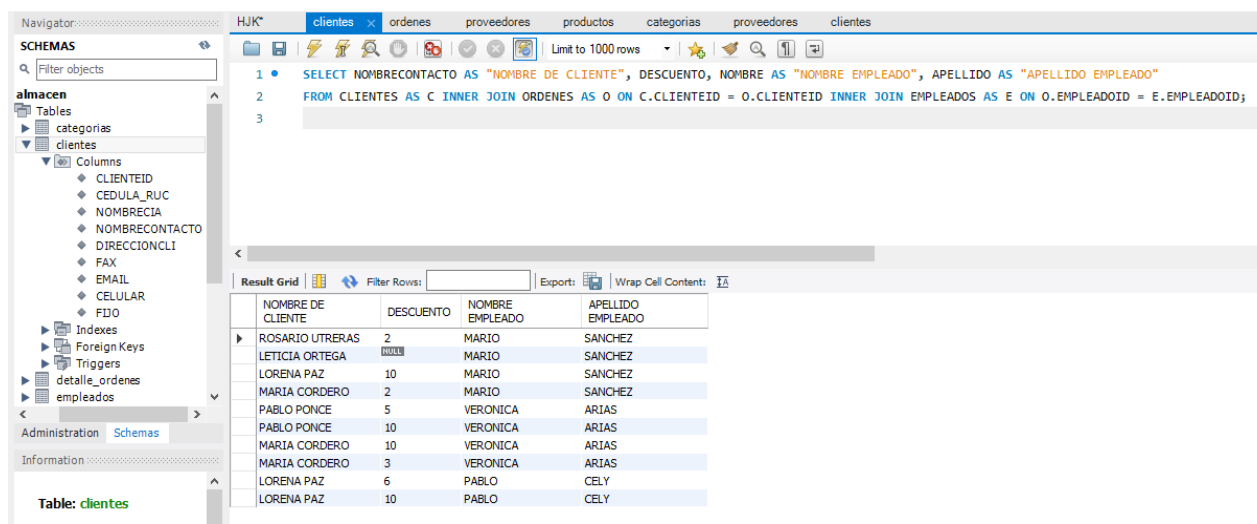
Ejemplo: %

DESARROLLO DEL EJERCICIO:

Con el script dado en clase para la creación de las diversas tablas de una base de datos; genere las instrucciones necesarias para extraer datos según los siguientes requerimientos:

- **Consulta para extraer el nombre de los clientes con el porcentaje de descuento obtenido en la compra y cual fue el vendedor que vendió o atendió el pedido.**

```
SELECT NOMBRECONTACTO AS "NOMBRE DE CLIENTE", DESCUENTO, NOMBRE AS "NOMBRE EMPLEADO", APELLIDO AS "APELLIDO EMPLEADO" FROM CLIENTES AS C INNER JOIN ORDENES AS O ON C.CLIENTEID = O.CLIENTEID INNER JOIN EMPLEADOS AS E ON O.EMPLEADOID = E.EMPLEADOID;
```

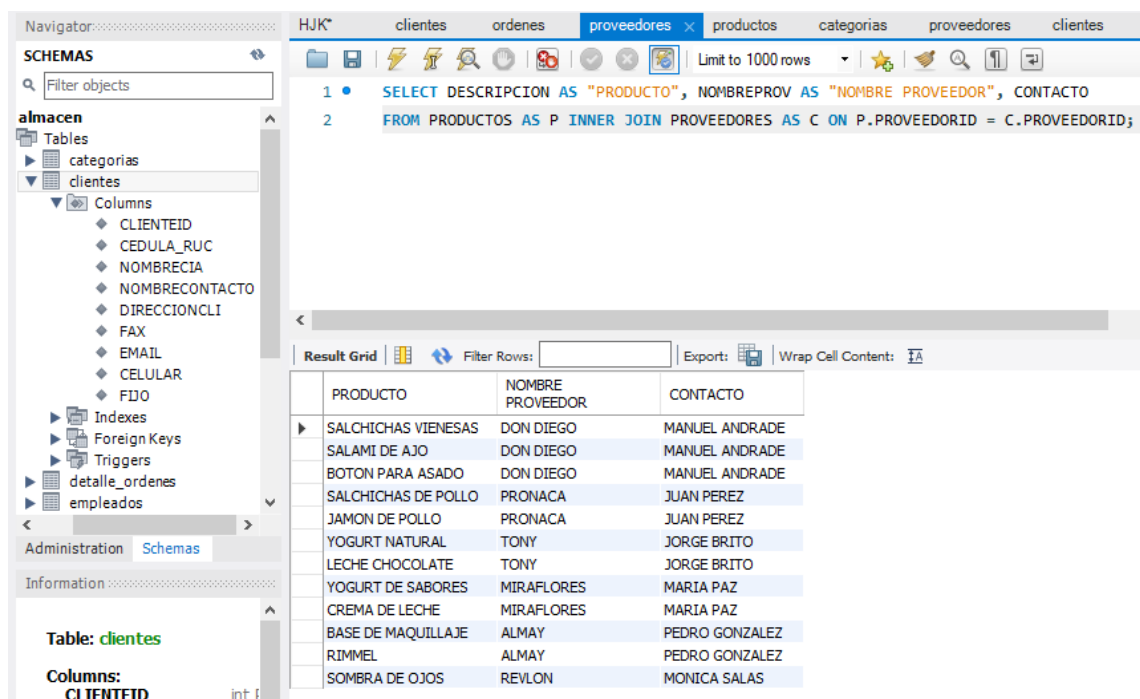


The screenshot shows a SQL query in the 'clientes' tab of a database tool. The query is:
 1 SELECT NOMBRECONTACTO AS "NOMBRE DE CLIENTE", DESCUENTO, NOMBRE AS "NOMBRE EMPLEADO", APELLIDO AS "APELLIDO EMPLEADO"
 2 FROM CLIENTES AS C INNER JOIN ORDENES AS O ON C.CLIENTEID = O.CLIENTEID INNER JOIN EMPLEADOS AS E ON O.EMPLEADOID = E.EMPLEADOID;
 3
 The result grid shows the following data:

NOMBRE DE CLIENTE	DESCUENTO	NOMBRE EMPLEADO	APELLIDO EMPLEADO
ROSARIO UTRERAS	2	MARIO	SANCHEZ
LETICIA ORTEGA	10	MARIO	SANCHEZ
LORENA PAZ	10	MARIO	SANCHEZ
MARIA CORDERO	2	MARIO	SANCHEZ
PABLO PONCE	5	VERONICA	ARIAS
PABLO PONCE	10	VERONICA	ARIAS
MARIA CORDERO	10	VERONICA	ARIAS
MARIA CORDERO	3	VERONICA	ARIAS
LORENA PAZ	6	PABLO	CELY
LORENA PAZ	10	PABLO	CELY

- **Consulta para extraer los productos con el nombre del proveedor y su contacto.**

```
SELECT DESCRIPCION AS "PRODUCTO", NOMBREPROV AS "NOMBRE PROVEEDOR", CONTACTO FROM PRODUCTOS AS P INNER JOIN PROVEEDORES AS C ON P.PROVEEDORID = C.PROVEEDORID;
```

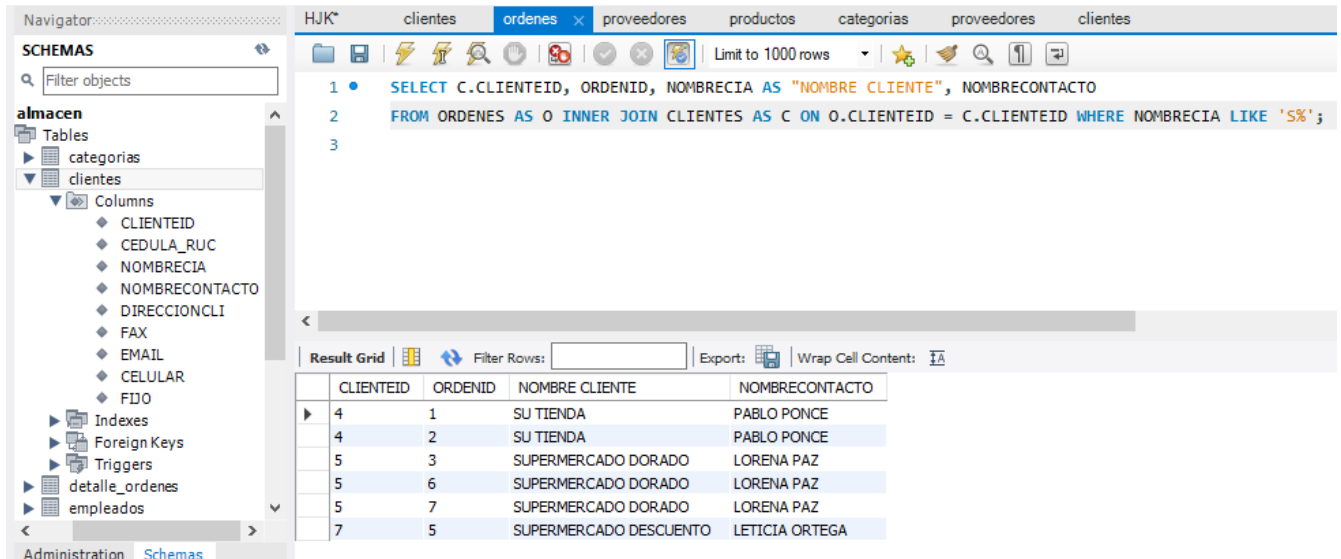


The screenshot shows a SQL query in the 'proveedores' tab of a database tool. The query is:
 1 SELECT DESCRIPCION AS "PRODUCTO", NOMBREPROV AS "NOMBRE PROVEEDOR", CONTACTO
 2 FROM PRODUCTOS AS P INNER JOIN PROVEEDORES AS C ON P.PROVEEDORID = C.PROVEEDORID;
 The result grid shows the following data:

PRODUCTO	NOMBRE PROVEEDOR	CONTACTO
SALCHICHAS VIENESAS	DON DIEGO	MANUEL ANDRADE
SALAMI DE AJO	DON DIEGO	MANUEL ANDRADE
BOTON PARA ASADO	DON DIEGO	MANUEL ANDRADE
SALCHICHAS DE POLLO	PRONACA	JUAN PEREZ
JAMON DE POLLO	PRONACA	JUAN PEREZ
YOGURT NATURAL	TONY	JORGE BRITO
LECHE CHOCOLATE	TONY	JORGE BRITO
YOGURT DE SABORES	MIRAFLORES	MARIA PAZ
CREMA DE LECHE	MIRAFLORES	MARIA PAZ
BASE DE MAQUILLAJE	ALMAY	PEDRO GONZALEZ
RIMMEL	ALMAY	PEDRO GONZALEZ
SOMBRA DE OJOS	REVLON	MONICA SALAS

- **Consulta para extraer de la tabla ordenes los clientes cuyo nombre empiece con la letra s.**

```
SELECT C.CLIENTEID, ORDENID, NOMBRECIA AS "NOMBRE CLIENTE", NOMBRECONTACTO FROM
ORDENES AS O INNER JOIN CLIENTES AS C ON O.CLIENTEID = C.CLIENTEID WHERE NOMBRECIA
LIKE 'S%';
```



The screenshot shows a database management tool interface. On the left, the 'Navigator' pane displays the 'almacen' schema with tables 'categorias', 'clientes', 'detalle_ordenes', and 'empleados'. The 'clientes' table is selected, showing its columns: CLIENTEID, CEDULA_RUC, NOMBRECIA, NOMBRECONTACTO, DIRECCIONCLI, FAX, EMAIL, CELULAR, and FIJO. The main query editor shows the following SQL query:

```
1 • SELECT C.CLIENTEID, ORDENID, NOMBRECIA AS "NOMBRE CLIENTE", NOMBRECONTACTO
2 FROM ORDENES AS O INNER JOIN CLIENTES AS C ON O.CLIENTEID = C.CLIENTEID WHERE NOMBRECIA LIKE 'S%';
3
```

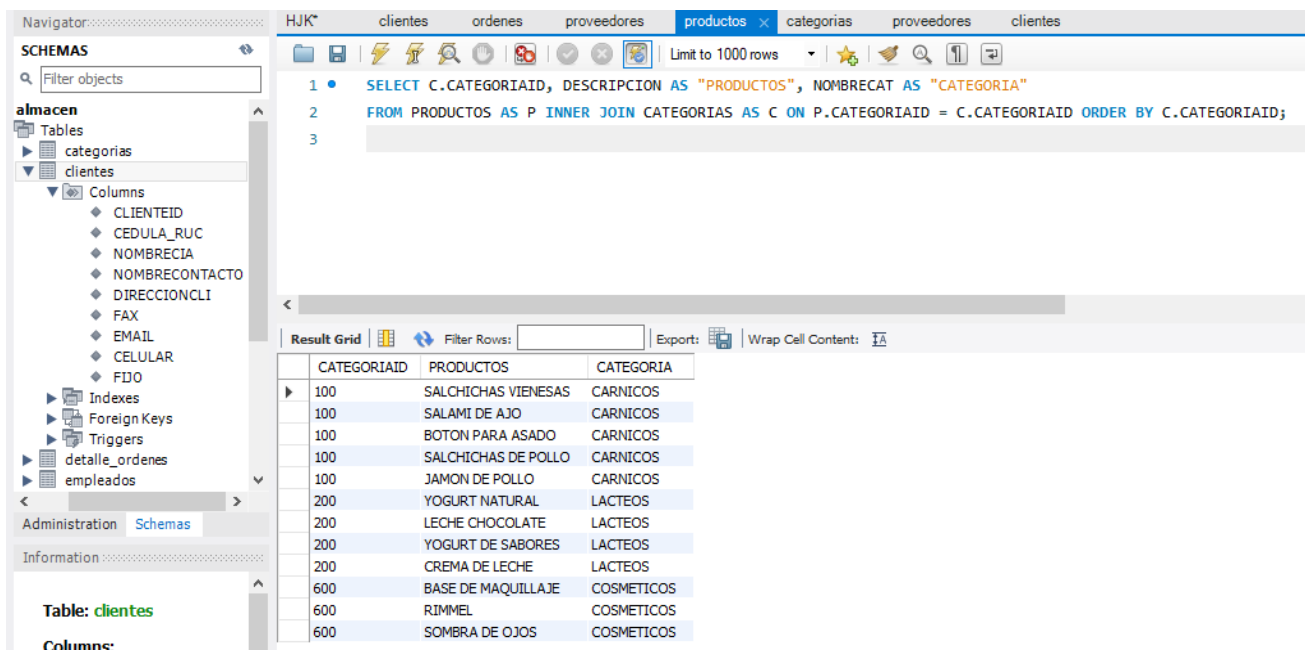
The 'Result Grid' shows the following data:

	CLIENTEID	ORDENID	NOMBRE CLIENTE	NOMBRECONTACTO
4	1		SU TIENDA	PABLO PONCE
4	2		SU TIENDA	PABLO PONCE
5	3		SUPERMERCADO DORADO	LORENA PAZ
5	6		SUPERMERCADO DORADO	LORENA PAZ
5	7		SUPERMERCADO DORADO	LORENA PAZ
7	5		SUPERMERCADO DESCUENTO	LETICIA ORTEGA

(NOTA: Se realiza en base a "NOMBRECIA")

- **Consulta para mostrar los productos con sus categorías ordenados por categoría.**

```
SELECT C.CATEGORIAID, DESCRIPCION AS "PRODUCTOS", NOMBRECAT AS "CATEGORIA" FROM
PRODUCTOS AS P INNER JOIN CATEGORIAS AS C ON P.CATEGORIAID = C.CATEGORIAID ORDER BY
C.CATEGORIAID;
```



The screenshot shows the same database management tool interface. The 'categorias' table is selected in the 'Navigator' pane. The main query editor shows the following SQL query:

```
1 • SELECT C.CATEGORIAID, DESCRIPCION AS "PRODUCTOS", NOMBRECAT AS "CATEGORIA"
2 FROM PRODUCTOS AS P INNER JOIN CATEGORIAS AS C ON P.CATEGORIAID = C.CATEGORIAID ORDER BY C.CATEGORIAID;
3
```

The 'Result Grid' shows the following data:

	CATEGORIAID	PRODUCTOS	CATEGORIA
100	100	SALCHICHAS VIENESAS	CARNICOS
100	100	SALAMI DE AJO	CARNICOS
100	100	BOTON PARA ASADO	CARNICOS
100	100	SALCHICHAS DE POLLO	CARNICOS
100	100	JAMON DE POLLO	CARNICOS
200	200	YOGURT NATURAL	LACTEOS
200	200	LECHE CHOCOLATE	LACTEOS
200	200	YOGURT DE SABORES	LACTEOS
200	200	CREMA DE LECHE	LACTEOS
600	600	BASE DE MAQUILLAJE	COSMETICOS
600	600	RIMMEL	COSMETICOS
600	600	SOMBRA DE OJOS	COSMETICOS

(NOTA: Ordenado en base a "CATEGORIAID")

RECOMENDACIONES:

- Al momento de crear nuestras tablas tenemos que tener en cuenta el uso de los diferentes tipos de datos ya que de ser incorrectos nuestra base de datos almacenará datos distintos a los esperados, por otro lado, no es recomendable usar nombres complejos en las claves y campos de las tablas. Por ejemplo, en la tabla Clientes, es mejor utilizar nombre en vez de nombre_cliente para almacenar el nombre del cliente. Cuanto más sencillos sean los nombres, más rápido se ejecuta la consulta.
- Una vez creadas las tablas debemos tener una buena relación de índices entre tablas para que las búsquedas relacionales funcionen correctamente. Para ello es mejor agregar índices a las tablas y, sobre ellas utilizar las sentencias de consulta (SELECT, WHERE...). También resulta recomendable acostumbrarse a verificar periódicamente el registro de consultas para identificar aquellas que deben ser optimizadas.
- Al momento de realizar el planteamiento inicial de la base de datos, con el paso del tiempo, seguramente tengamos tablas en desuso es necesario borrarlas de esta manera se evita el desuso ya que es muy parecido a guardar cosas que no usas en un cajón. Sucede lo mismo en las bases de datos, los sistemas tienen que rastrear todas las tablas y elementos hasta encontrar la respuesta a la consulta.

CONCLUSIONES:

- Con la elaboración del proyecto aprendimos cómo generar y acceder a una base de datos en MySQL.
- Gracias a los conocimientos adquiridos durante las clases pudimos realizar de una manera más eficiente este laboratorio y ponerlo en práctica mediante MySQL.
- MySQL ha sido un gestor muy útil y fácil de usar lo que nos permitió realizar este laboratorio de una manera más rápida y eficiente.
- Con MySQL logramos acceder a bases de datos de una manera fácil y gracias a esto pudimos ir extrayendo y mostrando los datos solicitados en el laboratorio.

BIBLIOGRAFÍA:

Como hacer consultas en MySql Workbench. (s. f.). Código SQL. Recuperado 2021, de

<https://codigosql.top/mysql/consultas/>

Hostinet. (2003). *¿Cómo dar acceso remoto a una base de datos MySQL?*

<https://www.hostinet.com/formacion/panel-alojamiento/como-acceso-remoto-base-datos-mysql/>