



UNIVERSIDAD FRANCISCO GAVIDIA
FACULTAD DE INGENIERIA Y SISTEMAS
PRACTICA
CICLO II – 2024

Asignatura: Administración de Bases de Datos

Profesor: Ing. Luis Enrique Reyes

Fecha de entrega: miércoles 21 de agosto de 2024

PRACTICA: FUNCIONES BASICAS DE SQL - GROUP BY / HAVING

OBJETIVO:

El objetivo de la práctica es conocer el uso de las INSTRUCCIONES GROUP BY y HAVING que son utilizadas en SQL para obtener resultados basados en el agrupamiento de los datos.

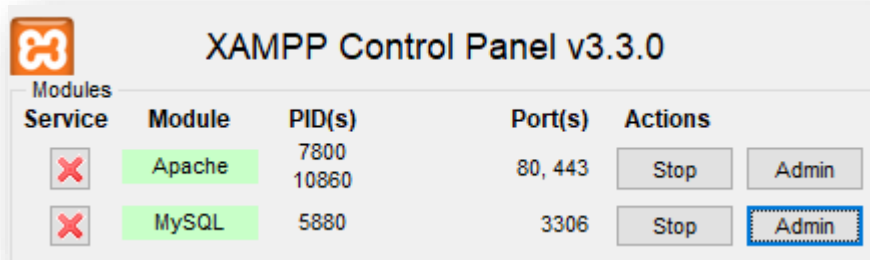
PREPARACIÓN Y EVIDENCIA A ENTREGAR (La entrega de la evidencia será evaluada)

Para iniciar la serie de ejercicios que estaremos realizando, utilizaremos la base de datos Northwind, la cual fue creada por Microsoft y se han generado diversas versiones para que sirva de referencia para la realización de pruebas basadas en el lenguaje SQL y otros propósitos.

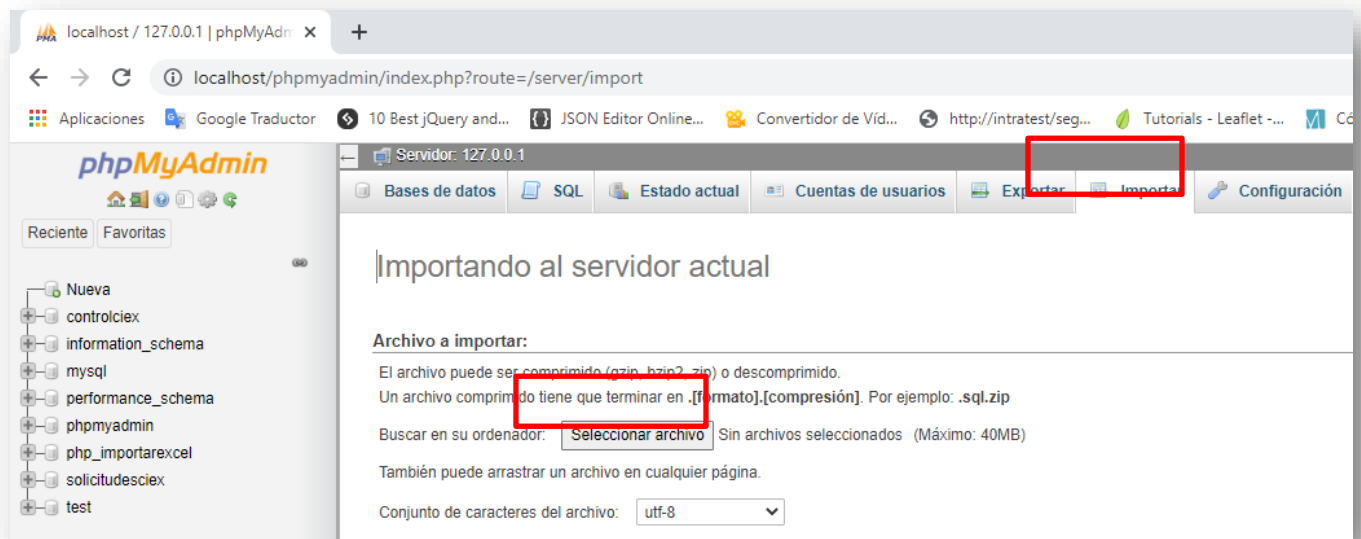
Northwind Traders es una organización ficticia que gestiona pedidos, productos, clientes, proveedores y muchos otros aspectos de una pequeña empresa.

Se le ha compartido el archivo "**northwind0.sql**", el cual contiene el script de creación de la base de datos "**northwind**", este archivo lo debe descargar para importarlo y realizar cada ejercicio.

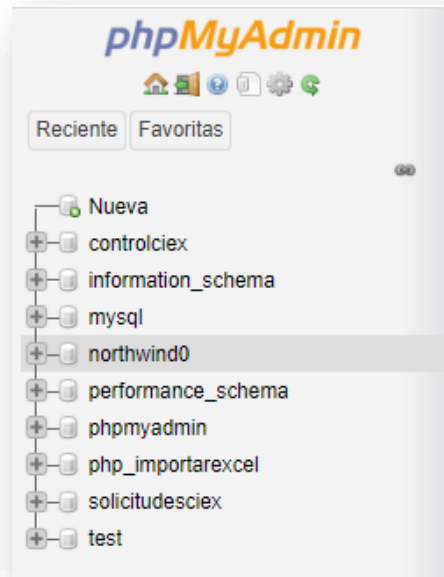
Para el desarrollo de esta guía, deberá acceder por medio de XAMPP, activando Apache y MySQL.



Al presionar el botón "Admin", accederá a phpMyAdmin y deberá importar la base de datos que se encuentra en el script "**northwind0.sql**", para lo cual deberá ubicarse en la pestaña "Importar", inmediatamente más abajo seleccionar el botón "Seleccionar archivo" y ubicar el archivo northwind0.sql, finalmente, más abajo presionar el botón "continuar" y comenzará el proceso de importación.



Concluido el proceso de importación deberá seleccionar en el panel de la izquierda la base de datos "northwind0".



Finalmente deberá seleccionar la opción del menú denominado SQL, en la cual, al seleccionarlo, le mostrará una ventana del editor SQL para que ingrese las instrucciones que se le indicarán y podrá verificar el resultado al presionar el botón “Continuar”.

A screenshot of the SQL editor window in phpMyAdmin. The window has a title bar with the text 'Ejecutar la(s) consulta(s) SQL en la base de datos northwind0:'. Below the title bar is a large text area for entering SQL queries. At the bottom of the window, there are several buttons and checkboxes: 'Limpiar', 'Formato', 'Obtener consulta almacenada automáticamente', 'Enlazar parámetros', 'Guardar esta consulta en favoritos:', 'Delimitador', 'Mostrar esta consulta otra vez', 'Mantener la caja de texto con la consulta', 'Deshacer («rollback») al finalizar', and 'Habilite la revisión de las claves foráneas'. A 'Continuar' button is located at the bottom right.

Si durante la realización de alguno de los ejercicios existe algún error en la instrucción, se le mostrará un recuadro indicando cual es el posible error existente, para lo cual deberá corregir el query respectivo y volver a ejecutarlo hasta obtener los resultados, por ejemplo:



Deberá proporcionar la evidencia de cada ejercicio, capturando un recorte de pantalla del ejercicio realizado, indicando el número del ejercicio y a continuación colocar la imagen al ejercicio respectivo.

Si en un ejercicio dar error, debe colocar el query con el error y el query corregido.

En los siguientes ejercicios veremos cómo poco a poco se va incrementando el grado de dificultad para obtener información de la base de datos **northwind0**.

Antes de realizar cada ejercicio lea las instrucciones, el query y las notas respectivas.

DESARROLLO

EJERCICIO 1. Obtendremos de la tabla customer, todos los puestos de trabajo (cargo) a los cuales pertenecen los clientes.

```
SELECT job_title, companyName
FROM customers
```

En este caso veremos que únicamente se muestran todos los registros existentes. El query es muy simple y fácil de identificar el resultado.

¿Cuántos registros de clientes hay por cada puesto?

NOTA: Este ejercicio generará un error, verifique cual es la instrucción correcta, corrijala y proceda con el ejercicio corregido.

EJERCICIO 2. Ahora, los ordenaremos por nombre del cargo, para que sepamos cuantos clientes hay por cada cargo.

```
SELECT job_title, company  
FROM customers  
ORDER BY job_title
```

Esta podría ser una forma de tratar de identificar cuantos clientes hay por cada cargo, si intentamos contarlos, el trabajo de conteo se complica, por lo que habrá que utilizar otra técnica para este propósito.

EJERCICIO 3. Si asumimos que no nos interesa quienes son los clientes y solo requerimos saber cuántos clientes hay por cargo, la acción a tomar sería la siguiente.

```
SELECT job_title, count(*)  
FROM customers  
GROUP BY job_title
```

Veremos en este caso, que mostrará la lista de todos los cargos existentes indicando el número de clientes por cada cargo.

Haciendo la aclaración que el listado aparecerá ordenado alfabéticamente.

EJERCICIO 4. Ahora, queremos que el listado se muestre, pero indicando primero cual es el cargo que posee más clientes hasta obtener el cargo que posee menos clientes.

```
SELECT job_title, count(*)  
FROM customers  
GROUP BY job_title  
ORDER BY count(*) DESC
```

En este caso veremos que la instrucción ORDER BY, hace que cambie el orden de los datos, aun cuando se esté utilizando la instrucción GROUP BY. Lo que hay que destacar es que se ordena sobre el campo de función de agrupamiento.

EJERCICIO 5. A continuación veremos el uso de la instrucción WHERE, que permitirá filtrar los registros que reúnan la condición de que solo algunos registros que cumplen el criterio de que el cargo inicie con la letra "A" deberán mostrarse.

```
SELECT job_title, count(*)  
FROM customers  
WHERE job_title like 'A%'  
GROUP BY job_title
```

En este caso veremos que solo dos cargos serán mostrados, ya que ellos inician con la letra "A".

EJERCICIO 6. En este ejercicio, lo que se espera es saber cuáles son los cargos que poseen menos de 5 clientes en sus respectivos cargos.

```
SELECT job_title, count(*)  
FROM customers  
GROUP BY job_title  
HAVING count(*) < 5  
ORDER BY count(*) DESC
```

Si observamos, hemos incluido la instrucción HAVING, que es la que se encarga en indicar que la función de agrupamiento es la que debe cumplir la condición. Esta es distinta a la que hace el WHERE, pues el WHERE funciona sobre los registros y el HAVING funciona sobre las funciones de agrupamiento.

EJERCICIO 7. En el siguiente ejercicio, veremos el uso combinado de la instrucción WHERE y el HAVING.

```
SELECT job_title, count(*)
```

```
FROM customers
WHERE job_title like 'S%' or job_title like 'A%'
GROUP BY job_title
HAVING count(*) <10
ORDER BY job_title
```

Lo que se busca en esta ocasión, es obtener todos aquellos cargos que tienen menos de 10 clientes, pero que además el cargo inicie con la letra "A" o con la letra "S"

Aquí se filtran primero los cargos que inician con las letras indicadas, y posteriormente determina cuantos clientes hay de los cargos que cumplen esa condición.

EJERCICIO 8. Ahora veremos cuantos cargos existen en la tabla customer.

```
SELECT count(job_title)
FROM customers
```

Si observamos, veremos que hemos caído en un error, pues el resultado que nos da es de 29, lo cual es incorrecto.

El valor lo que significa es que hay 29 registros que poseen datos en la columna country.

EJERCICIO 9. La respuesta correcta para mostrar el número de cargos existentes en la tabla customer es el siguiente.

```
SELECT count(distinct job_title)
FROM customers
```

La razón es porque hemos incrustado la instrucción DISTINCT, el cual permite evitar que un mismo dato se repita (evitar duplicados de la columna indicada), esto no significa que los registros estén duplicados.

EJERCICIO 10. En este ejercicio veremos que podemos hacer uso de cálculos dentro de la instrucción sql partiendo de varias columnas, para lo cual tomaremos la tabla "order_details". En esta tabla conocemos el producto, el precio de venta y la cantidad vendida.

```
SELECT product_id, unit_price, quantity
FROM   order_details
ORDER BY product_id
```

El resultado proporciona conocer para cada producto, su precio unitario y la cantidad vendida, pero en la práctica ¿cuánto significa eso?

EJERCICIO 11. Alternativa mejorada para obtener las ventas de los productos.

```
SELECT product_id, unit_price, quantity, (unit_price * quantity) as
venta
FROM   order_details
ORDER BY product_id
```

Al observe el query, veremos que hemos realizado el producto del precio unitario y la cantidad vendida, y a ello le hemos creado un "ALIAS", para darle significado el resultado.

EJERCICIO 12. Ahora lo que tendremos que obtener es la venta de cada uno de los productos, sin mostrar el precio de venta y la cantidad vendida.

```
SELECT product_id, sum(unit_price * quantity) as venta
FROM   order_details
GROUP BY product_id
```

En este caso veremos que aparece la lista, pero solo indicando el identificador del producto y el valor total de la venta de cada producto.

¿Cómo sabremos cual es el nombre del producto?

EJERCICIO 13. Ahora lo que tendremos que obtener es la venta de cada uno de los productos, sin mostrar el precio de venta y la cantidad vendida.

```
SELECT products.id,  
       products.ProductName,  
       sum(order_details.unit_price * order_details.quantity) as venta  
FROM   products, order_details  
WHERE  products.id = order_details.product_id  
GROUP BY products.id, products.ProductName
```

Veremos en este query que hemos hecho varios cambios, por ejemplo hemos tenido que "calificar" los campos para saber de cual table proceden.

La instrucción WHERE permite "emparejar" la llave primaria y foránea de cada tabla.

En el GROUP BY, incluimos los dos campos que no tienen función de asociación.

Recuerde que, si aparece un error, debe corregirlo y luego incluir el resultado con el query con error y el ya corregido.

EJERCICIO 14. Finalmente, necesitamos saber cuáles productos se tuvieron ventas que superan los \$1,000.

```
SELECT products.id,  
       products.Product_Name,  
       sum(order_details.unit_price * order_details.quantity) as venta  
FROM   products, order_details  
WHERE  products.id = order_details.product_id  
GROUP BY products.id, products.Product_Name  
HAVING sum(order_details.unit_price * order_details.quantity) >1000
```

Para este ejercicio, se agregó la instrucción HAVING, incorporando la operación de la sumatoria que deberá cumplir que deber ser mayor a los 1,000.

EJERCICIO 15. Una alternativa al ejercicio anterior es el siguiente.

```
SELECT products.id,  
       products.Product_Name,  
       sum(order_details.unit_price * order_details.quantity) as venta  
FROM   products, order_details  
WHERE  products.id = order_details.product_id  
GROUP BY products.id, products.Product_Name  
HAVING venta >1000
```

La diferencia con el ejercicio anterior, es que solo agregamos la instrucción HAVING, para indicar que la condición se debe aplicar sobre la función de agrupamiento.

La otra característica es que utilizamos como filtro en el HAVING, el nombre asignado como ALIAS.