

# U P

### Tarea

# Evidencia de Desempeño Lenguaje Estructurado de Consulta Unidad 4

Axel David Pérez Hernández, Guadalupe Joan Anaya Cárdenas, Alejandro Yahir Mérida Cruz, Ángel Yahir Martínez Gómez, Alexis Yahir Villar Escorcia

Universidad Politécnica De Tulancingo, Calle Ingenierías #100, Huapalcalco, Tulancingo

Base de Datos

Nombre del profesor: Víctor Hugo Fernández Cruz

Tulancingo Hidalgo 11 de abril de 2024



EVIDENCIA DE DESEMPEÑO LENGUAJE ESTRUCTURADO DE CONSULTA UNIDAD 4	1
RESUMEN	3
INTRODUCCIÓN	4
CASO PRÁCTICO "NEGOCIO"	5
Construcción de consultas en álgebra relacional	6
Operadores del álgebra relacional	6
Consultas en álgebra relacional	6
FORMULAR SENTENCIAS DE MANIPULACIÓN DE DATOS EN LENGUAJE ESTRUCTURADO DE CONSULTA	8
Sentencias De Manipulación De Datos INSERT	8
Sentencias De Manipulación De Datos UPDATE	
Sentencias De Manipulación De Datos DELETE	14
FORMULAR SENTENCIAS DE CONSULTA DE DATOS EN LENGUAJE ESTRUCTURADO DE CONSULTA	
FORMULAR SENTENCIAS DE CONSULTA DE MÚLTIPLES TABLAS EN LENGUAJE ESTRUCTURADO DE CONSULTA	20
CONCLUSIONES	24
REFERENCIAS	25

### Resumen

El caso práctico del negocio presenta una base de datos que gestiona información crucial para un sistema de suministro, a través de las tablas "Pieza", "Proveedores" y "Suministra". Se ilustran consultas en álgebra relacional, así como sentencias de manipulación de datos en SQL, que permiten realizar operaciones como la inserción, actualización y eliminación de registros. Además, se presentan ejemplos de consultas de datos y consultas de múltiples tablas, demostrando la versatilidad y potencia de estas herramientas para optimizar la gestión empresarial.

# Introducción

El caso práctico del negocio presenta una base de datos vital para la gestión eficiente de un sistema de suministro. A través de las tablas "Pieza", "Proveedores" y "Suministra", se almacena y relaciona información esencial sobre los productos ofrecidos, los proveedores y los precios de suministro. Este conjunto de datos permite realizar consultas y manipulaciones que facilitan el seguimiento del inventario, la negociación de precios y la toma de decisiones estratégicas para garantizar el funcionamiento óptimo del negocio.

# Caso Práctico "Negocio"

La base de datos "Negocio" gestiona información crucial para un sistema de suministro.

La tabla "Pieza" almacena detalles sobre los productos ofrecidos, como sus códigos y nombres.

La tabla "Proveedores" contiene datos de los proveedores, identificándolos por un código único y su nombre. Finalmente, la tabla "Suministra" establece las relaciones entre las piezas y los proveedores, registrando los precios de suministro de cada pieza por parte de los proveedores.

En conjunto, esta base de datos facilita el seguimiento y la gestión de inventario, así como la negociación de precios con los proveedores para garantizar una operación eficiente del negocio.

Figura 1

Figura 2

Figura 3

Figura 4

### Construcción de consultas en álgebra relacional.

### Operadores del álgebra relacional

- Selección: Este operador filtra las tuplas de una relación según una condición específica, devolviendo solo aquellas que la cumplan. (σ)
- Proyección: Permite seleccionar solo ciertas columnas de una relación,
   descartando las demás, y eliminando posibles duplicados. (π)
- Unión: Combina dos relaciones compatibles en una sola, incluyendo todas las tuplas de ambas relaciones, eliminando duplicados automáticamente. (U)
- Intersección: Devuelve las tuplas que están presentes en ambas relaciones de entrada, descartando las que no son comunes. (∩)
- Diferencia: Este operador devuelve las tuplas que están en una relación de entrada, pero no en la otra. (-)
- Producto Cartesiano: Combina cada tupla de una relación con todas las tuplas de otra, creando una nueva relación con todas las combinaciones posibles. (x)
- Renombramiento: Cambia los nombres de las relaciones y de sus atributos,
   permitiendo un mejor entendimiento y manejo de las consultas. (ρ)

(Equipo de UAEH, 2024)

### Consultas en álgebra relacional

• Obtener todos los códigos y nombres de las piezas:

πcodigo, nombre(pieza)

• Obtener todos los lds y nombres de los proveedores:

 $\pi$ Id, nombre(proveedores)

• Obtener todos los códigos de pieza, lds de proveedor y precios:

πcodPieza, IdPieza, precio(suministra)

• Obtener el nombre de la pieza para cada código de pieza:

πcodPieza, nombre (pieza x suministra)

• Obtener el nombre del proveedor para cada Id de proveedor:

πldPieza, nombre (proveedores x suministra)

 Obtener el código de pieza, el nombre de la pieza, el ld del proveedor, el nombre del proveedor y el precio para cada registro en la tabla suministra:

πcodPieza, nombre (pieza x suministra), IdPieza, nombre (proveedores x suministra), precio(suministra)

 Obtener todos los códigos de pieza y sus precios, junto con el nombre del proveedor que los suministra:

πcodPieza, precio(suministra), nombre (proveedores x suministra)

 Obtener todos los Ids de proveedor y los nombres de las piezas que suministran, junto con el precio de cada pieza:

πIdPieza, nombre (pieza x suministra), precio(suministra)

• Obtener el precio promedio de cada pieza:

πcodPieza, AVG (precio)(suministra)

• Obtener el nombre del proveedor que suministra la pieza más cara:

πnombre (proveedores x (πcodPieza, MAX (precio)(suministra)))

### Formular sentencias de manipulación de datos en lenguaje estructurado de consulta.

Para hacer sentencias de manipulación de datos es necesario conocer que es lo que hace cada uno de los siguientes comandos SQL:

- INSERT: Añade nuevos registros a una tabla de una base de datos. Permite especificar los valores para cada columna del registro a insertar. Puede usarse para agregar múltiples registros en una sola operación separándolos por comas.
- UPDATE: Permite cambiar los valores de una o más columnas en una tabla existente. Al emplear esta sentencia, se especifica la tabla que se va a actualizar, se definen las columnas que se modificarán junto con los nuevos valores que se les asignarán, y se puede incluir una condición opcional para actualizar solo los registros que cumplan con ciertas condiciones específicas.
- DELETE: Este comando borra registros de una tabla en una base de datos,
  permitiendo especificar condiciones para seleccionar qué registros eliminar.
   Puede eliminar uno o varios registros según criterios como valores específicos en columnas o combinaciones de condiciones. También puede vaciar completamente una tabla si no se especifica ningún criterio, eliminando todos los datos.

(Hernández, 2024)

### Sentencias De Manipulación De Datos INSERT

 Supongamos que queremos registrar nuevos productos en la tabla de Pieza con los siguientes datos: 4," Paquete de cinchos"; 5," Paquete de rondanas".

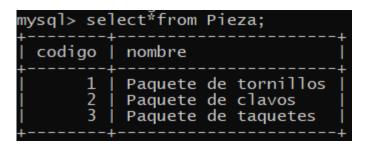


Figura 5

Figura 6

2. Ahora queremos agregar un proveedor en la tabla de Proveedores que se encargará de proporcionar las piezas que registramos en la tabla Pieza con los siguientes datos: 1005," Luis Manuel".

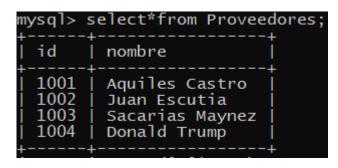


Figura 7

Figura 8

3. Y ya para finalizar solo hace falta agregar el registro que une al proveedor con las piezas que nos proporciona mediante un registro en la tabla suministra, en la que bien podemos agregar el precio si ya lo conocemos: 4, 1005, 34 y 5, 1005, null.

mysql> select*from Suministra;					
codPieza	idProveedor	precio			
1 1 1 1 2 1 2 3 3	1001 1002 1003 1001 1002 1002 1004	30     28     29     23     26     35     34			

Figura 9

```
mysql> insert into Suministra values (4,1005,34),(5,1005,null);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select*from Suministra;
+------+
| codPieza | idProveedor | precio |
+------+
| 1 | 1001 | 30 |
+------+
| 1 | 1002 | 28 |
| 1 | 1003 | 29 |
| 2 | 1001 | 23 |
| 2 | 1001 | 23 |
| 3 | 1002 | 35 |
| 3 | 1004 | 34 |
| 4 | 1005 | 34 |
| 5 | 1005 | NULL |
+------+
```

Figura 10

### Sentencias De Manipulación De Datos UPDATE

 Supongamos que queremos actualizar el precio de la pieza con código 1 suministrada por el proveedor con ID 1001 a un nuevo valor de 25.

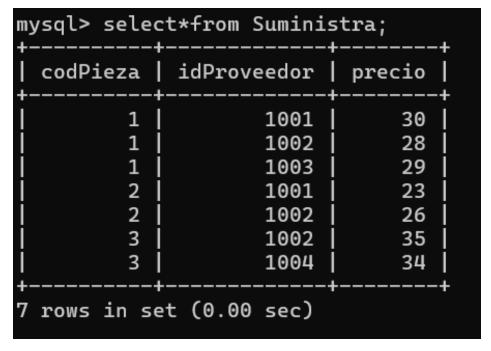


Figura 11

```
mysql> UPDATE Suministra
    -> SET precio = 25
    -> WHERE codPieza = 1 AND idProveedor = 1001;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select*from Suministra;
  codPieza
             idProveedor
                            precio
                                25
         1
                     1001
         1
                     1002
                                28
         1
                     1003
                                29
         2
                                23
                     1001
         2
                     1002
                                26
         3
                     1002
                                35
         3
                     1004
                                34
 rows in set (0.00 sec)
```

Figura 12

 Ahora deseamos aumentar en 5 unidades el precio de todas las piezas suministradas por el proveedor con ID 1002.

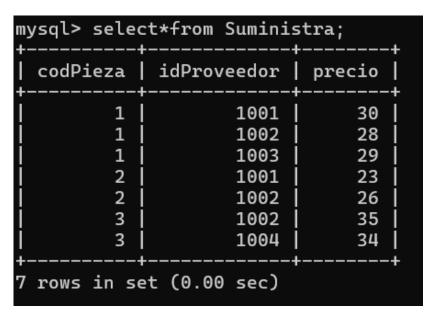


Figura 13

```
mysql> UPDATE Suministra
    -> SET precio = precio + 5
    -> WHERE idProveedor = 1002;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3 Changed: 3 Warnings: 0
mysql> ^C
mysql> select*from Suministra;
  codPieza
             idProveedor
                            precio
         1
                                25
                     1001
         1
                                33
                     1002
         1
                     1003
                                29
         2
                                23
                     1001
         2
                                31
                     1002
         3
                     1002
                                40
         3
                     1004
                                34
  rows in set (0.00 sec)
```

Figura 14

 Ahora requerimos cambiar el nombre de la pieza con código 3 de "Paquete de taquetes" a "Taquetes de madera".



Figura 15

Figura 16

### Sentencias De Manipulación De Datos DELETE

Si por alguna razón se ha registrado mal algún dato, la sentencia DELETE nos ayuda a eliminar el error según el caso, pero debemos tener cuidado con la sentencia, ya que podríamos eliminar todos los datos, por eso se requiere que seamos específicos en esta sentencia.

 Supongamos que queremos eliminar el registro donde la pieza con código 1 es suministrada por el proveedor con ID 1003.

mysql> select*from Suministra;							
codPieza   idProveedor   precio							
1	1001	25					
1	1002	33					
1   1003   29							
2							
2   1002   31							
3	1002	40					
3	1004	34					
++							
7 rows in set (0.00 sec)							

Figura 17

```
mysql> DELETE FROM Suministra
    -> WHERE codPieza = 1 AND idProveedor = 1003;
Query OK, 1 row affected (0.00 sec)
mysql> select*from Suministra;
  codPieza
                            precio
             idProveedor
         1
                     1001
                                25
         1
                     1002
                                33
         2
                     1001
                                23
         2
                     1002
                                31
         3
                     1002
                                40
         3
                     1004
                                34
  rows in set (0.00 sec)
```

Figura 18

 Supongamos que deseamos eliminar de la tabla Suministra todos los registros donde el proveedor es "Juan Escutia" (ID 1002) porque hace tiempo que ya no se tiene relación laboral con este proveedor.

mysql> select*from Suministra; +						
codPieza   idProveedor   precio						
1 1 1001 25						
1 1 1002 33						
1 1003 29						
2   1001   23						
2   1002   31						
3	1002	40				
3	1004	34				
++						
7 rows in set (0.00 sec)						

Figura 19

```
mysql> DELETE FROM Suministra
    -> WHERE idProveedor = 1002;
Query OK, 3 rows affected (0.01 sec)

mysql> select*from Suministra;
+-----+
| codPieza | idProveedor | precio |
+-----+
| 1 | 1001 | 25 |
| 2 | 1001 | 23 |
| 3 | 1004 | 34 |
+-----+
3 rows in set (0.00 sec)
```

Figura 20

 Digamos que queremos eliminar la pieza con código 2 de la tabla Pieza debido a que ya no fabrican esta pieza y se agotó la existencia en el inventario de Negocio.

```
mysql> select*from Pieza;
+-----+
| codigo | nombre
| +-----+
| 1 | Paquete de tornillos |
| 2 | Paquete de clavos |
| 3 | Taquetes de madera |
+-----+
3 rows in set (0.00 sec)
```

Figura 21

Figura 22

Formular sentencias de consulta de datos en lenguaje estructurado de consulta.

• Seleccionar todos los registros de la tabla "Suministra":

SELECT \* FROM Suministra;

codPieza   idProveedor   precio	mysql> SELECT * FROM Suministra;					
1     1     1002     28       1     1003     29       2     1001     23       2     1002     26       3     1002     35       3     1004     34	codPieza	idProveedor	precio			
1 5 1 1005   NULL	j <u>2</u> j 3	1002 1003 1001 1002 1002 1004 1005	28   29   23   26   35   34			

Figura 23

• Seleccionar el precio de todas las piezas suministradas:

SELECT precio FROM Suministra;

Figura 24

• Contar cuántos registros hay en la tabla "Suministra":

SELECT COUNT(\*) AS total\_registros FROM Suministra;

Figura 25

Calcular el precio promedio de las piezas suministradas:

SELECT AVG(precio) AS precio\_promedio FROM Suministra;

Figura 26

• Encontrar el precio máximo de las piezas suministradas:

SELECT MAX(precio) AS precio\_maximo FROM Suministra;

Figura 27

Encontrar el precio mínimo de las piezas suministradas:

SELECT MIN(precio) AS precio\_minimo FROM Suministra;

Figura 28

• Seleccionar los registros de suministros con un precio mayor a \$30:

SELECT \* FROM Suministra WHERE precio > 30;

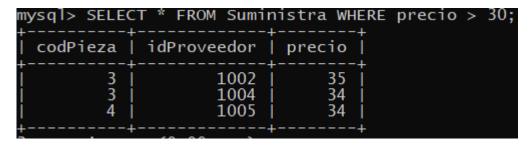


Figura 29

Contar cuántos registros de suministros hay para cada proveedor:

SELECT idProveedor, COUNT(\*) AS total\_registros FROM Suministra GROUP BY idProveedor;

Figura 30

• Seleccionar los proveedores que suministran piezas a un precio mayor a \$25:

SELECT DISTINCT idProveedor FROM Suministra WHERE precio > 25;

```
mysql> SELECT DISTINCT idProveedor FROM Suministra WHERE precio > 25;

+-----+

| idProveedor |

+----+

| 1001 |

| 1002 |

| 1003 |

| 1004 |

| 1005 |
```

Figura 31

• Calcular la suma total de precios de las piezas suministradas:

SELECT SUM(precio) AS total\_precios FROM Suministra;

Figura 32

Formular sentencias de consulta de múltiples tablas en lenguaje estructurado de consulta.

 INNER JOIN: Devuelve únicamente las filas que tienen coincidencias en ambas tablas.

 ${\tt SELECT\ Pieza.codigo,\ Pieza.nombre\ AS\ nombre\_pieza,\ Proveedores.id,}$ 

Proveedores.nombre AS nombre\_proveedor, Suministra.precio

- -> FROM Pieza
- -> INNER JOIN Suministra ON Pieza.codigo = Suministra.codPieza
- -> INNER JOIN Proveedores ON Suministra.idProveedor = Proveedores.id;
- Esta consulta devuelve solo las filas donde hay coincidencias entre las tablas
   Pieza, Suministra y Proveedores basadas en las claves primarias y foráneas.

+	nombre_pieza	id	nombre_proveedor	precio
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Paquete de tornillos Paquete de tornillos Paquete de tornillos Paquete de clavos Paquete de clavos Paquete de taquetes Paquete de taquetes	1002	Aquiles Castro Juan Escutia Sacarias Maynez Aquiles Castro Juan Escutia Juan Escutia Donald Trump	30   28   29   23   26   35   34
7 rows in	set (0.00 sec)	·	·	·+

Figura 33

 LEFT JOIN: Devuelve todas las filas de la tabla izquierda y las filas coincidentes de la tabla derecha. Si no hay coincidencias en la tabla derecha, se devuelven NULL para las columnas de la tabla derecha.

SELECT Pieza.codigo, Pieza.nombre AS nombre\_pieza, Proveedores.id, Proveedores.nombre AS nombre\_proveedor, Suministra.precio

FROM Pieza

LEFT JOIN Suministra ON Pieza.codigo = Suministra.codPieza

LEFT JOIN Proveedores ON Suministra.idProveedor = Proveedores.id;

i. Esta consulta devuelve todas las filas de la tabla Pieza y las filas coincidentes de las tablas Suministra y Proveedores. Si no hay coincidencias en las tablas Suministra y Proveedores, se devuelven NULL para las columnas de estas tablas. (Esto considerando que se eliminó la pieza con código 2.)

+	nombre_pieza	id	nombre_proveedor	precio
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		1002 1003 1002 1002	Juan Escutia	30   28   29   26   35   34
6 rows in	set (0.00 sec)	H		+

Figura 34

3. RIGHT JOIN: Es similar al LEFT JOIN, pero devuelve todas las filas de la tabla derecha y las filas coincidentes de la tabla izquierda. Si no hay coincidencias en la tabla izquierda, se devuelven NULL para las columnas de la tabla izquierda. SELECT Pieza.codigo, Pieza.nombre AS nombre\_pieza, Proveedores.id,

Proveedores.nombre AS nombre\_proveedor, Suministra.precio

**FROM Pieza** 

RIGHT JOIN Suministra ON Pieza.codigo = Suministra.codPieza

RIGHT JOIN Proveedores ON Suministra.idProveedor = Proveedores.id;

Esta consulta devuelve todas las filas de las tablas Suministra y
Proveedores y las filas coincidentes de la tabla Pieza. Si no hay
coincidencias en la tabla Pieza, se devuelven NULL para las columnas
de esta tabla. (Ahora tomando en cuenta que se eliminó la pieza con
codigo 3 y 2).

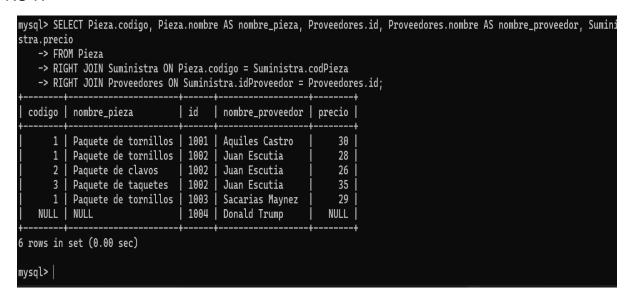


Figura 35

 FULL JOIN: Devuelve todas las filas de ambas tablas, combinando filas cuando hay coincidencias y devolviendo NULL en las columnas donde no hay coincidencias.

SELECT Pieza.codigo, Pieza.nombre AS Pieza, Proveedores.id AS idProveedor, Proveedores.nombre AS Proveedor, Suministra.precio

FROM Pieza

JOIN Suministra ON Pieza.codigo = Suministra.codPieza

JOIN Proveedores ON Suministra.idProveedor = Proveedores.id;

Esta consulta devuelve todas las filas de las tablas Pieza, Suministra y
 Proveedores, combinando las filas cuando hay coincidencias y
 devolviendo NULL donde no hay coincidencias.

1   Paquete de tornillos   1001   Aquiles Castro   1   Paquete de tornillos   1002   Juan Escutia   1   Paquete de tornillos   1003   Sacarias Maynez   2   Paquete de clavos   1002   Juan Escutia   3   Paquete de taquetes   1002   Juan Escutia	+   codigo	Pieza	idProveedor	Proveedor	precio
1 3   radacec de cadacecs   1002   odan Escacia	1 1 1 2	Paquete de tornillos Paquete de tornillos	1002 1003 1002	Juan Escutia Sacarias Maynez Juan Escutia	30   28   29   26   35

Figura 36

# **Conclusiones**

El análisis detallado de este caso práctico destaca la importancia de las consultas en álgebra relacional, así como de las sentencias de manipulación de datos en SQL, para la gestión eficiente de una base de datos empresarial. Estas herramientas proporcionan la capacidad de realizar consultas complejas y manipulaciones precisas de los datos, lo que resulta fundamental para optimizar la gestión de inventario, negociar precios con proveedores y obtener insights valiosos para la toma de decisiones estratégicas en el negocio.

# Referencias

- Equipo de UAEH. (11 de 04 de 2024). 3.3 Álgebra Relacional. Obtenido de CIDECAME: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/33\_algebra\_relacion al.html
- Hernández, J. J. (11 de 04 de 2024). *Unidad 11 .Manipulación de datos*. Obtenido de josejuansanchez.org: https://josejuansanchez.org/bd/unidad-11-teoria/index.html