



Bases de datos relacionales

Consultando información de una tabla



Identificar las características, rol y elementos fundamentales de una base de datos relacional para la gestión de la información en una organización y utilizar el lenguaje estructurado de consultas SQL para la obtención de información que satisface los requerimientos planteados a partir de un modelo de datos dado

{desafío}
latam_

- Unidad 1:
Bases de datos relacionales
- Unidad 2:
Manipulación de datos y transaccionalidad en las operaciones
- Unidad 3:
Definición de tablas
- Unidad 4:
Modelos Entidad-Relación y Relacional



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Reconoce los elementos fundamentales del lenguaje SQL para la obtención de datos de una base de datos relacional.*
- *Utiliza sentencias SQL con condiciones de selección para resolver un problema planteado de selección condicional.*

¿Para qué utilizamos
la instrucción
INSERT INTO?



/* El Lenguaje Estructurado de Consultas SQL */

¿Qué es SQL?

Lenguaje de Consultas estructuradas (Structured Query Language)

- Está diseñado para el uso específico de administración y consulta de información en sistemas de gestión de bases de datos como PostgreSQL.
- Se utiliza para definir, manipular y controlar los datos.
- Con la sintaxis estructural del lenguaje podemos:
 - Crear registros.
 - Leer la información.
 - Actualizar los datos
 - Eliminar uno o un conjunto de datos.

Cargando el dataset

A continuación utilizaremos el dataset definido dentro de “**Material de apoyo - Consultando información de una tabla**”, el cual está disponible en la plataforma.

- **Paso 1:** Carga el archivo comprimido del material de apoyo en una nueva base de datos en tu computador, si tuviste problemas con la instalación puedes ocupar sqlliteonline
- **Paso 2:** Probemos nuestro dataset con `SELECT * FROM productos LIMIT 10.`



/* Utilización de funciones en las consultas */

Existen distintos tipos de funciones que podemos utilizar

Función	Descripción	Uso	Resultado
LEFT	Extrae un subconjunto de caracteres desde el inicio de una cadena.	<code>LEFT('PostgreSQL', 4)</code>	'Post'
COALESCE	Devuelve el primer valor no nulo de la lista de argumentos.	<code>COALESCE(NULL, NULL, 'Hola', 'Mundo')</code>	'Hola'
LOWER	Convierte todos los caracteres de una cadena a minúsculas.	<code>LOWER('PostgreSQL')</code>	'postgresql'
UPPER	Convierte todos los caracteres de una cadena a mayúsculas.	<code>UPPER('postgresql')</code>	'POSTGRESQL'
LENGTH	Retorna la longitud de una cadena de texto.	<code>LENGTH('Hola Mundo')</code>	10

Veamos un ejemplo

Queremos ordenar los productos por el largo del nombre. Esto lo podemos hacer con:

```
SELECT * FROM productos order by LENGTH(nombre_producto)
```

Incluso podríamos agregar una columna mostrando el largo para poder observar mejor los resultados

```
SELECT *, LENGTH(nombre_producto) FROM productos order by  
LENGTH(nombre_producto)
```



Utilizando lo aprendido,
selecciona todos los datos de
los productos junto a un código,
este código está compuesto por
las primeras 3 letras de la
categoría (revisa la diapositiva
de funciones)



Ejercicio guiado

"Implementar funciones de agregado sobre una tabla"



Funciones de agregado

Las **funciones de agregado** son un tipo especial de función en SQL que operan sobre un conjunto de valores y devuelven un único valor resumido o agregado. Veamos un ejemplo:

```
SELECT SUM (valor) FROM tabla_ejemplo;  
=> 210
```

Valor
10
20
30
40
50
60



Funciones de agregado

suma

```
SELECT SUM(campo) FROM tabla
```

promedio

```
SELECT AVG(campo) FROM tabla
```

Cuenta

```
SELECT COUNT(*) FROM tabla
```



SELECT COUNT(*), COUNT(campo) FROM
tabla puede dar dos resultados distintos si
hay valores nulos sobre campo

Máximo

```
SELECT MAX(campo) FROM tabla
```

Mínimo

```
SELECT MIN(campo) FROM tabla
```

Utilizando el dataset

1. Calcula el precio mínimo
2. Calcula el precio máximo
3. Calcula el precio promedio
4. Cuenta la cantidad de productos con stock
(pista: puedes combinar las funciones de agregado con where)



Utilizando el dataset

1. Cuenta la cantidad de precios sobre 7000
2. Calcula el promedio de los productos que tienen stock



¿Comprendiste el uso de funciones en consultas?



/* Group By */

A veces queremos agregar por grupo

Ejemplo nos piden el precio más alto de cada producto por categoría

```
SELECT max(precio) from productos GROUP BY categoria;
```

Podemos mostrar la categoría junto al precio.

```
SELECT categoria, max(precio) from productos GROUP BY categoria;
```



Regla importante !!

No podemos mostrar campos no agrupados salvo que estén en funciones de agregado.

```
SELECT nombre_producto, max(precio) from productos GROUP BY categoria;
```

nombre_producto es un campo no agrupado, al intentar esta consulta obtendremos el error:

ERROR: column "productos.nombre_producto" must appear in the GROUP BY clause or be used in an aggregate function



Utilizando el dataset de la tabla ventas

1. ¿Cuál es el promedio de ventas?
2. ¿Cuál es el promedio de ventas por producto?
3. ¿Cuál es el promedio de ventas por producto después del '2024-01-05'?



Utilizando el dataset de la tabla ventas

1. ¿Qué tipo de producto generó la venta más alta?
 - Pista: La venta se calcula como $\text{cantidad} * \text{precio_unitario}$
 - Ejemplo: `Select(cantidad * precio_unitario) as venta`



Utilizando el dataset de la tabla ventas

1. ¿Cuántos productos se vendieron en cada fecha?



¿Lograste aplicar la
agrupación de los datos?





Próxima sesión...

- *Utiliza sentencias SQL que requieren la consulta a varias tablas relacionadas a partir de un modelo de datos dado para resolver un problema planteado de selección.*
- *Utiliza sentencias SQL con funciones de agrupación para resolver un problema planteado que requiere la agrupación de datos.*

{desafío}
latam_

*Academia de
talentos digitales*

