Consultas en SQL (Parte II)

Clase sincrónica

{desafío} latam_

Generar consultas agrupadas sobre múltiples tablas en SQL para extraer información de bases de datos.

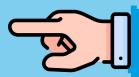
- Unidad 1: Primeros pasos del analista de datos
- Unidad 2: Consultas en SQL

  (Parte I)

  (Parte II)

  (Parte III)

  (Parte IV)



Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

- *Realizar consultas con condiciones en tablas.*
- *Realizar operaciones entre datos de una tabla.*
- *Combinar operaciones y condiciones en consultas.*
- *Realizar subconsultas.*

{desafío} latam_

¿Qué recordamos de las consultas en SQL?

{desafío} latam_

Comandos básicos de SQL

Recordemos

¿Qué realiza el comando SELECT * FROM TABLA?

- Indica cuál tabla se va a seleccionar.
- Selecciona todos los campos de TABLA.
- Selecciona todas las filas de TABLA.
- Selecciona todas las columnas de TABLA.

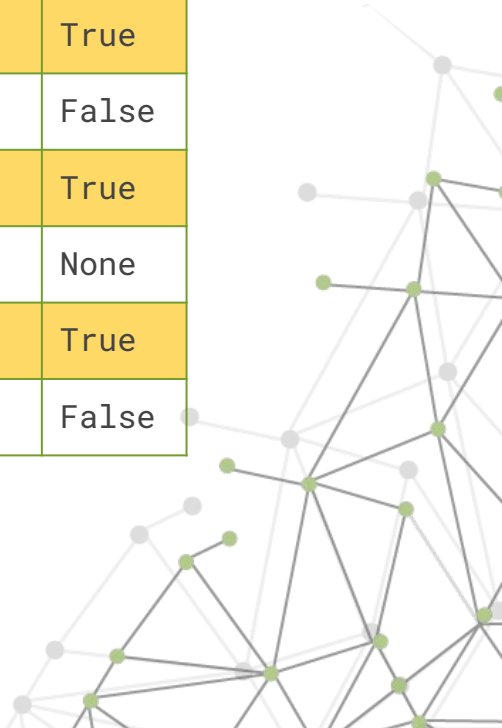
{desafío} latam_

Comandos básicos de SQL

Recordemos

- ¿Qué consulta seleccionaría todos los registros cuya columna C4 sea true?
- ¿Qué consulta seleccionaría los 2 primeros registros?
- ¿Qué consulta ordenaría todos los registros por orden decreciente en la columna c2?
- ¿Qué sucede si ejecutamos el comando UPDATE SET C2 = 0;?
- ¿Qué consulta borraría los 2 registros con c1 más altos de la tabla?

c1	c2	c3	c4
98	23	'a'	True
45	45	'b'	False
34	76	'c'	True
87	34	'd'	None
82	56	'e'	True
90	10	'f'	False



{desafío} latam_

/* Conteo de registros */

Contar

Contando la cantidad de registros

Tenemos una tabla con distintos tipos de monedas y algunas repetidas.

Nos piden contar la cantidad de elementos que hay en la tabla.

¿Cómo se realizaría?

nombre
Guarani
Rupiah
Ruble
Peso
Peso
Yuan Renminbi
Yuan Renminbi
Yuan Renminbi
Euro
Ariary
Denar
Rupiah
Ruble
Yuan Renminbi
Lek

{desafío} latam_

Consultas y reportes

Creemos una tabla para el ejercicio

```
create table monedas (
     nombre VARCHAR(50)
);
insert into monedas (nombre) values ('Guarani');
insert into monedas (nombre) values ('Rupiah');
insert into monedas (nombre) values ('Ruble');
insert into monedas (nombre) values ('Peso');
insert into monedas (nombre) values ('Peso');
insert into monedas (nombre) values ('Yuan Renminbi');
insert into monedas (nombre) values ('Yuan Renminbi');
insert into monedas (nombre) values ('Yuan Renminbi');
insert into monedas (nombre) values ('Euro');
insert into monedas (nombre) values ('Ariary');
insert into monedas (nombre) values ('Denar');
insert into monedas (nombre) values ('Rupiah');
insert into monedas (nombre) values ('Ruble');
insert into monedas (nombre) values ('Yuan Renminbi');
insert into monedas (nombre) values ('Lek');
```

Consultas y reportes

Contando los elementos

SELECT count(*) FROM monedas	Contamos todos los registros
SELECT count(*) FROM monedas where nombre = 'Peso'	Contamos todos los registros cuya nombre sea Peso
SELECT count(*) FROM monedas where nombre = 'Peso' or nombre = 'Euro'	Contamos todos los registros cuya nombre sea Peso o Euro

Count es una función de agregado.

Consultas y reportes

Seleccionar y contar distintos

Nos piden hacer una consulta para listar todos los tipos de moneda en la tabla sin repeticiones.

¡Distinct al rescate!

nombre
Guarani
Rupiah
Ruble
Peso
Peso
Yuan Renminbi
Yuan Renminbi
Yuan Renminbi
Euro
Ariary
Denar
Rupiah
Ruble
Yuan Renminbi
Lek

Consultas y reportes

Seleccionar y contar distintos

SELECT distinct(nombre) FROM monedas	Listamos todas las monedas con distinto nombre
SELECT count(distinct(nombre)) FROM monedas	Contamos todos los registros cuya nombre sea Peso

Ejercicio
"Contando en nuestra tabla"

Contando en nuestra tabla

1. Crea la siguiente tabla con sus respectivos registros en PostgreSQL

```
create table ventas (categoria VARCHAR(50), monto INT);
insert into ventas (categoria, monto) values ('Books', 214);
insert into ventas (categoria, monto) values ('Games', 293);
insert into ventas (categoria, monto) values ('Baby', 241);
insert into ventas (categoria, monto) values ('Tools', 719);
insert into ventas (categoria, monto) values ('Tools', 385);
insert into ventas (categoria, monto) values ('Movies', 882);
insert into ventas (categoria, monto) values ('Outdoors', 654);
insert into ventas (categoria, monto) values ('Baby', 332);
insert into ventas (categoria, monto) values ('Grocery', 332);
insert into ventas (categoria, monto) values ('Toys', 952);
insert into ventas (categoria, monto) values ('Games', 682);
insert into ventas (categoria, monto) values ('Books', 527);
insert into ventas (categoria, monto) values ('Kids', 980);
insert into ventas (categoria, monto) values ('Grocery', 300);
```

Contando en nuestra tabla

2. Crea consultas para:

- Contar la cantidad de registros.
- Contar la cantidad de registros con distinta categoría.
- Contar la cantidad de registros con distinto monto.
- Contar la cantidad de registros de tipo 'tool' o tipo ‘games’.

/* Registros agrupados */

Consultas y reportes

¿Cuántas veces aparece cada moneda?

Ahora nos piden contar cuántas veces aparece cada registro repetido.

Para lograrlo, contaremos los registros agrupados por nombre.

nombre
Guarani
Rupiah
Ruble
Peso
Peso
Yuan Renminbi
Yuan Renminbi
Yuan Renminbi
Euro
Ariary
Denar
Rupiah
Ruble
Yuan Renminbi
Lek

Consultas y reportes

¿Cuántas veces aparece cada moneda?

```
SELECT nombre, count(*) FROM monedas group by nombre;
```

nombre	count
Yuan Renminbi	4
Peso	2
Ruble	2
Guarani	1
Rupiah	2
Euro	1
Ariary	1
Lek	1
Denar	1

Ejercicio
"Conteo agrupado"

Conteo agrupado

En la tabla de ventas del ejercicio anterior, nos solicitan contar cuantas veces se repite cada categoría.

Pista: No selecciones los montos.

categoria	count
Kids	1
Movies	1
Grocery	2
Baby	2
Games	2
Toys	1
Tools	2
Books	2
Outdoors	1

La consulta arrojará el resultado que se muestra en la imagen.

{desafío} latam_

/* Suma y promedio de datos */

Consultas y reportes

El problema de seleccionar sobre un campo no agrupado

¿Cuál es el monto al agrupar por ropa? ¿100 o 200?

categoría	monto
ropa	100
ropa	200

¿Qué sucedería si agrupamos por categoría y monto?

Dado que en este caso no hay ninguna ocasión donde ambos se repiten simultáneamente, terminaremos con los mismos datos que al no agrupar.

Consultas y reportes

¿Cuánto se vendió en total de cada categoría?

Aquí nos piden el total de cada categoría.

Tenemos que agrupar por categoría y sumar los montos. Podemos hacerlo gracias a SQL.

departamento	monto
Books	214
Games	293
Baby	241
Tools	719
Tools	385
Movies	882
Outdoors	654
Baby	332
Grocery	332
Toys	952
Games	682
Books	527
Kids	980
Grocery	300

{desafío} latam_

Consultas y reportes

Agrupar y sumar

- SELECT categoria, sum(monto) FROM ventas group by categoria.
- También podemos mostrar el promedio con AVG
- SELECT categoria, sum(monto), avg(monto) FROM ventas group by categoria.

categoria	sum
Kids	980
Movies	882
Grocery	632
Baby	573
Games	975
Toys	952
Tools	1104
Books	741
Outdoors	654

SUM y AVG son funciones de agregado junto a COUNT.

/* Filtros por condiciones */

Consultas y reportes

Condiciones sobre campos agrupados

¿Qué tendríamos que hacer si queremos seleccionar todas las categorías que, al agruparlas, sumen montos mayores a 900?

categoria	sum
Kids	980
Movies	882
Grocery	632
Baby	573
Games	975
Toys	952
Tools	1104
Books	741
Outdoors	654

Consultas y reportes

Where V.S Having

Con **where** podemos filtrar basándonos en una condición. Si filtramos categoría con monto mayor a 900, solo los registros que cumplan esta condición entrarán en el grupo. Aquí serían kids y toys.

```
SELECT categoria,
sum(monto) FROM ventas
WHERE monto > 900
GROUP BY categoria
```

departamento	monto
Books	214
Games	293
Baby	241
Tools	719
Tools	385
Movies	882
Outdoors	654
Baby	332
Grocery	332
Toys	952
Games	682
Books	527
Kids	980
Grocery	300

categoria	sum
Kids	980
Toys	952

Esto **no** es lo que nos pidieron

{desafío} latam_

Consultas y reportes

Where V.S Having

Para sumar y luego filtrar por la suma, ocuparemos una cláusula nueva llamada **having**.

```
SELECT categoria,
sum(monto)
FROM ventas
GROUP BY categoria HAVING
sum(monto) > 900;
```

departamento	monto
Books	214
Games	293
Baby	241
Tools	719
Tools	385
Movies	882
Outdoors	654
Baby	332
Grocery	332
Toys	952
Games	682
Books	527
Kids	980
Grocery	300

categoria	sum
Kids	980
Games	975
Toys	952
Tools	1104

Esto **sí** es lo que nos pidieron

{desafío} latam_

Ejercicio
"Consultas y cláusulas"

{desafío}
latam_

Consultas y cláusulas

Tenemos un listado de transacciones de distintos usuarios, donde el id lo representa. Los montos están definidos como datos de tipo numéricos (number) y representa unidades monetarias (UM):

```
create table transacciones (id INT, monto INT);

insert into transacciones (id, monto) values (4, 490);
insert into transacciones (id, monto) values (4, 159);
insert into transacciones (id, monto) values (2, 1000);
insert into transacciones (id, monto) values (2, 578);
insert into transacciones (id, monto) values (1, 613);
insert into transacciones (id, monto) values (3, 366);
insert into transacciones (id, monto) values (2, 546);
insert into transacciones (id, monto) values (5, 265);
insert into transacciones (id, monto) values (1, 163);
insert into transacciones (id, monto) values (2, 135);
insert into transacciones (id, monto) values (2, 404);
insert into transacciones (id, monto) values (4, 885);
insert into transacciones (id, monto) values (2, 309);
insert into transacciones (id, monto) values (5, 78);
```

{desafío} latam_

Con lo anterior, realiza el ejercicio de la siguiente diapositiva.

Consultas y cláusulas

Utilizando la tabla de transacciones, crea las siguientes consultas:

1. Listar el monto total en transacciones por usuario.
2. Listar el monto total en transacciones por usuario filtrando aquellas inferiores a 500 Unidades Monetarias.
3. Listar el monto total de transacciones solo si como monto total se superan las 1000 Unidades Monetarias.

/* Subconsultas */

Consultas y reportes

Subconsultas

En algunos casos vamos a querer seleccionar datos con base en el resultado de otra consulta. Por ejemplo, obtener todos los registros mayores que el promedio de los registros

Para esto utilizaremos una consulta dentro de otra.

```
SELECT *
FROM ventas
WHERE monto >
  (SELECT avg(monto)
  FROM ventas)
```

departamento	monto
Books	214
Games	293
Baby	241
Tools	719
Tools	385
Movies	882
Outdoors	654
Baby	332
Grocery	332
Toys	952
Games	682
Books	527
Kids	980
Grocery	300

categoria	monto
Tools	719
Movies	882
Outdoors	654
Toys	952
Games	682
Kids	980

Consultas y reportes

Subconsultas

```
SELECT * FROM ventas
WHERE monto > (SELECT AVG(monto) FROM ventas);
```

- Las subconsulta debe ir entre paréntesis.
- El punto y coma (;) solo se utiliza al final de la consulta para terminarla.

Ejercicio "Subconsultas"

{desafío} latam_

Subconsultas

Utilizando la tabla de transacciones anteriormente creada, lista todas las transacciones que superen el monto de transacción promedio.
Toma en consideración que el promedio en la columna monto es de 427,93.

{desafío} latam_

Desafío - Consultas agrupadas

{desafío}
latam_

Desafío

"Consultas agrupadas"

- Descarga el archivo "Desafío".
- Tiempo de desarrollo asincrónico: desde 4 horas.
- Tipo de desafío: individual

¡AHORA TE TOCA A TI! 💪

{desafío} latam_

Ideas fuerza

SQL permite realizar ***conteos de registros****, poniendo* ***condiciones específicas*** *sobre los datos.*

Podemos realizar ***operaciones*** *sobre los datos de una tabla y obtener su* ***resultado.***

Una ***condición*** *para una consulta puede ser, precisamente, el resultado de una operación.*

Es posible realizar ***subconsultas****, es decir, consultas con condiciones* ***dentro de conjuntos de datos*** *que a su vez cumplen condiciones.*

¿Por qué es importante poder realizar consultas agrupadas en un negocio?

Recursos asincrónicos

¡No olvides revisarlos!

Esta semana contarás con los siguientes recursos:

- Guía de estudio.
- Desafío “Consultas agrupadas”.

{desafío} latam_

Próxima sesión...

- *Realizar consultas en múltiples tablas en PostgreSQL*

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales