



UNIVERSIDAD
COOPERATIVA
DE COLOMBIA

Gestión de la Información

Gustavo Adolfo Gómez Gómez

**MSc. Gestión, aplicación y desarrollo de software
2024**

www.ucc.edu.co

Gestión de la Información

Unidad 1: Asociar diseños, plataformas y soportes informáticos a los modelos de gestión de información de la organización.

Tema: Lenguaje Estructurado de Consultas: SQL

Lenguaje de Manipulación de Datos - DML - Consultas

Estructura de consultas SQL

Cláusulas de consulta:

- **SELECT** (lista_atributos)
 - Obtiene relación de atributos deseados
- **FROM** ((relación))
 - Lista de relaciones (tablas) de donde tomar los datos
- **WHERE** (predicado ó condición lógica)
 - Condiciona la consulta



Estructura de consultas SQL

Estructura básica

select Attr1, Attr2,..., Attrn

from r1, r2, rm

where P

Attr: atributos

R: relaciones

P: Predicado o condición

Estructura de consultas SQL - SELECT

Cláusula **SELECT**

Ejemplo: consultar los nombres de los estudiantes:

Select *nombres*

From *estudiantes*

Estructura de consultas SQL - SELECT

Cláusula **SELECT**

Comodin * : todos los atributos de la relación

Consultar todos los campos de la tabla estudiantes:

Select *

From *estudiantes*

Estructura de consultas SQL - SELECT

Cláusula **SELECT** (operaciones de atributos)

- Expresiones aritméticas (+ - * /)

Ejp: Calcular el valor de las notas del curso si éstas valen un 30%

Select *(nota*30)/100*

From *cursos*

Estructura de consultas SQL - WHERE

Cláusula WHERE

Condición o predicado para limitar la consulta

Ejp: consultar los estudiantes de bucaramanga

Select *nombres*

From *estudiantes*

Where *ciudad = "Bucaramanga"*

Estructura de consultas SQL - WHERE

Cláusula WHERE (rango)

Permite condicionar un atributo a un rango de valores

Ejp: consultar los estudiantes nacidos entre 1990 y 2000

Select *nombres, apellidos*

From *estudiantes*

Where *fecha_nacimiento BETWEEN '1990' AND '2000'*

Estructura de consultas SQL - WHERE

Cláusula FROM

Identifica la relación/tabla (o relaciones/tablas) origen de los datos

Ejp: consultar los estudiantes y sus cursos

Select *estudiantes.nombres, cursos.nombre*

From *estudiantes, cursos*

Where *estudiantes.codigo_estudiante=cursos.codigo_estudiante*

Bases de datos relacionales

Datos Relación Sucursal - Cuenta

Sucursal		
nombre_sucursal	ciudad_sucursal	activos
SedeA	Bucaramanga	4000000
SedeB	Floridablanca	520000
SedeC	Lebrija	45000000
SedeD	Girón	95020000

Cuenta			
numero_cuenta	id_cliente	nombre_sucursal	saldo
c101	100026	SedeB	1560000
c102	100028	SedeA	10000
c104	100030	SedeA	35000
c105	100025	SedeC	300000
c106	100028	SedeD	200000

Bases de datos relacionales

Datos Relación Prestamo - Cliente

Prestamo			
numero_prestamo	id_cliente	nombre_sucursal	valor_prestado
p10006	100026	SedeB	10000
p10007	100028	SedeA	2510000
p10008	100027	SedeA	2611000
p10009	100029	SedeC	21540000
p10010	100028	SedeD	2541000
p10011	100026	SedeA	620000

Cliente			
id_cliente	nombre_cliente	calle_cliente	ciudad_cliente
100025	Pedro Perez	Americas	Bucaramanga
100026	Maria Suarez	Bulevard	Piedecuesta
100027	Antonio Alvarez	Bolivar	Floridablanca
100028	Marcela Gutierrez	Los angeles	Lebrija
100029	Adriana Martinez	Curazao	Girón
100030	Jose Perez	Americas	Bucaramanga

Estructura de consultas SQL

Estructura básica

Ejemplo: Listado de clientes de bucaramanga con su nombre y calle

select nombre_cliente, calle_cliente

from cliente

where ciudad_cliente= "Bucaramanga"

Estructura de consultas SQL

Estructura básica

Ejemplo: Listado de clientes con el respectivo valor prestado

select nombre_cliente, valor_prestado

from cliente, prestamo

where cliente.id_cliente=prestamo.id_cliente

Operación de renombramiento

Cláusula: **as**

- Renombrar Atributos y Relaciones
- Se utiliza tanto el **select** como el **from**

Nombre_antiguo as Nombre_nuevo

Operación de renombramiento

Consideremos la consulta anterior:

```
SELECT nombre_cliente, valor_prestado  
FROM cliente, prestamo  
WHERE cliente.id_cliente=prestamo.id_cliente
```

El resultado de esta consulta es una relación con los atributos siguientes:

```
nombre_cliente, valor_prestado
```

Operación de renombramiento

Dónde se utiliza:

- Atributos iguales entre relaciones
- Operaciones sobre atributos
- Cambiar el nombre de un atributo para la consulta

```
SELECT nombre_cliente AS 'Cliente', valor_prestado AS 'Valor'  
FROM cliente, prestamo  
WHERE cliente.id_cliente=prestamo.id_cliente
```

Operación de renombramiento

Renombrar la relación

(en algunos RDBMS la cláusula **as** es opcional)

```
select T.nombre_cliente, S.número_préstamo, S.impuesto  
from cliente as T, préstamo as S  
where T.cliente_id = S.cliente_id
```

Operación de renombramiento

Renombrar la relación: útil para comparar la relación consigo misma

Ejemplo: Determinar el nombre de todas las sucursales que tienen activos mayores que, al menos, una sucursal de Bucaramanga

```
select T.nombre_sucursal  
from sucursal as T, sucursal as S  
where T.activos > S.activos and S.ciudad_sucursal = "Bucaramanga"
```

Operaciones con cadenas de caracteres

SQL especifica caracteres entre comillas dobles o sencillas (“ ” - ‘ ’)

Diferencia entre mayúsculas y minúsculas

Comparación de caracteres (Cláusula: **Like** ó **Not Like**)

- **Coincidencia de texto (%):** Coincide con cualquier subcadena de caracteres
- **Coincidencia de caracteres (_):** El caracter _ coincide con cualquier caracter

Operaciones con cadenas de caracteres

Comparación de caracteres (Cláusula: **Like** ó **Not Like**)

Ejemplos:

- **"Perez%"**: coincide con cualquier cadena de caracteres que empiece con "Perez"
- **"%car%"**: Coincide con cualquier cadena de caracteres contenga "mar" como subcadena (ejp: cardenas, máscara)

Operaciones con cadenas de caracteres

Comparación de caracteres (Cláusula: **Like** ó **Not Like**)

Ejemplos:

- “_ _ _”: coincide con cualquier cadena de caracteres que tenga 3 caracteres
- “_ _ _%”: Coincide con cualquier cadena de caracteres que tenga al menos 3 caracteres

Operaciones con cadenas de caracteres

Comparación de caracteres (Cláusula: **Like** ó **Not Like**)

Ejercicio:

Determinar el nombre de todos los clientes cuya dirección empieza con “calle”

```
select nombre_cliente  
from cliente  
where calle_cliente like “Calle %”
```


Operaciones con cadenas de caracteres

Comparación de caracteres (Cláusula: **Like** ó **Not Like**)

Qué pasa si quiere buscar las direcciones que tenga el caracter % ó _ ?

Ejp: "Calle 105 # 60-64A Torre 5_105"

Se utiliza un caracter de escape: éste se utiliza antes de los caracteres especiales.caracter de escape: (\)

Ejp: like 'ab\%cd%': coincide con todas las cadenas que empiecen por "ab%cd".

Orden en la presentación de las tuplas

Control sobre el orden en el que se presentan los resultados de la consulta

Cláusula: **Order By**

Ejp: **Consultar en orden alfabético los clientes que tienen préstamo en la sucursal “Sede A”**

```
select c.nombre_cliente  
from cliente c, préstamo p  
where c.id_cliente = p.id_cliente and p.nombre_sucursal = “Sede A”  
order by c.nombre_cliente
```

Orden en la presentación de las tuplas

Cláusula: **Order By**

- Si el atributo es texto: Orden alfabético, Numérico: ascendente
- Por defecto: Ascendente (**asc**)
- En caso de necesitar Descendente se agrega la palabra **Desc**
- Se puede ordenar con respecto a 1 o más atributos

Ejp: **Consultar ordenado alfabéticamente por nombre de sucursal los préstamos de mayor a menor**

```
select p.*  
from préstamo p  
order by p.nombre_sucursal asc, p.valor_prestado desc
```

Orden en la presentación de las tuplas

Cláusula: **Order By**

Ejercicio: Consultar los clientes, la ciudad y sus cuentas ordenados por ciudad (alfabéticamente) y según el saldo de mayor a menor

```
select c.nombre_cliente, c.ciudad, cu.saldo  
from cliente c, cuenta cu  
where c.cliente_id = cu.cliente_id  
order by c.ciudad asc, c.saldo desc
```

Funciones de agregación

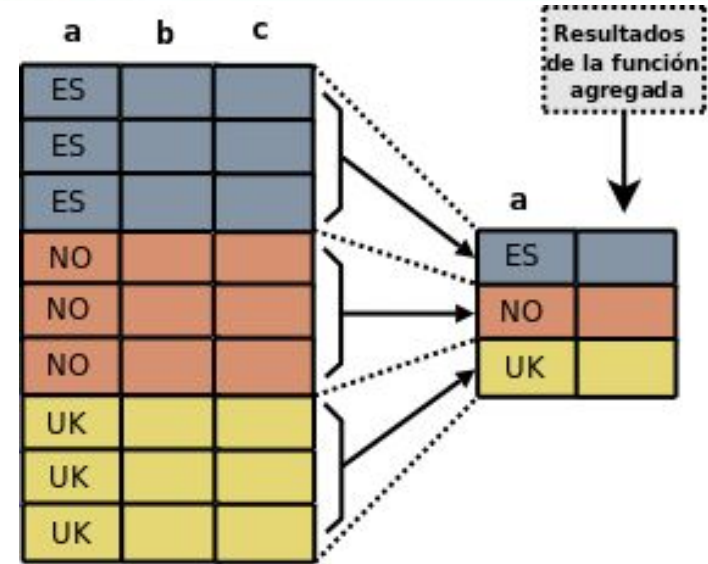
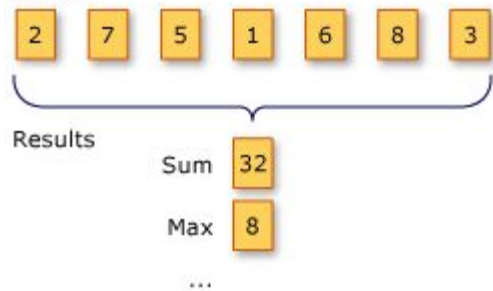
- Promedio: avg
- Mínimo: min
- Máximo: max
- Sumatoria: sum
- Recuento: count
- Adicionales (Group by, Distinct, Having)

R = (A	B)
1	3
3	4
3	2

SUM(A) = 7
COUNT(A) = 3
MAX(B) = 4
AVG(B) = 3

Funciones de agregación

Se aplican al conjunto total o a grupos



Funciones de agregación

Aplicado a un conjunto total:

Ejemplo: Consultar el promedio de los saldos de las cuentas de los clientes que abrieron cuenta en la sucursal SedeA

```
Select avg(saldo) as promedio  
From cuenta  
Where nombre_sucursal="SedeA"
```

Funciones de agregación

Aplicado a grupos

Ejemplo: consultar la sumatoria de los préstamos por sucursal

```
Select nombre_sucursal, sum(valor_prestado) as suma_sucursal  
From prestamo  
Group By nombre_sucursal
```


Funciones de agregación

No tener en cuenta duplicados (cláusula: **distinct**)

Ejemplo: contar los clientes con cuentas

```
Select count(distinct cliente_id)  
From cuenta
```

Funciones de agregación

Aplicar condición a la agrupación.

Ejemplo: consultar las sucursales con promedio de préstamos mayor a 10000

```
Select nombre_sucursal, avg(valor_prestado) as promedio  
From prestamo  
Group By nombre_sucursal  
Having avg(valor_prestado)>10000
```

Funciones de agregación

Ejercicio: Teniendo en cuenta las siguientes relaciones:

Cliente(id, nombre, direccion, telefono)

compraCarro(id_carro, id_cliente, id_cliente_v, modelo, marca, valor, fecha)

rentaCarro(id_carro, id_cliente, dias, fecha)

1. Consultar el promedio del valor de los carros modelo 2010
2. Contar los clientes que han rentado carros
3. Contar cuántos carros se han comprado por marca
4. Retornar los clientes y la cantidad de carros de los clientes que hayan rentado por lo menos 2

Ejercicio

Ejercicio Clase parte 1:

Realizar las siguientes consultas SQL

- **Cuántos clientes tienen préstamos mayores de 500000**
- **Qué préstamos son del cliente con id=2**
- **Cuáles clientes mujeres tienen préstamos**
- **Promedio del valor de crédito entre los clientes hombres**
- **Sumatoria de los créditos de los clientes con fecha de préstamo del 31 de diciembre de 2018**

Ejercicio

Ejercicio Clase parte 2:

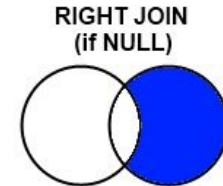
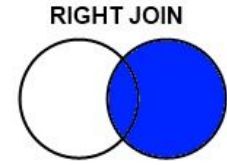
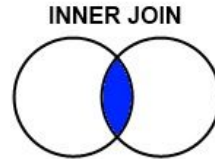
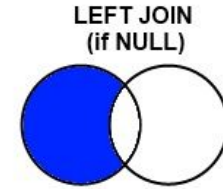
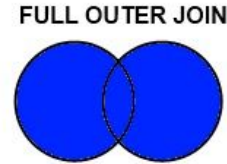
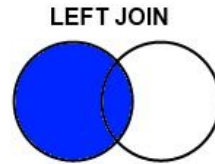
- Consultar cada sucursal por nombre y cantidad de activos
- Consultar el id y el nombre de los clientes del banco
- Consultar las sucursales en la ciudad de bucaramanga
- Recuperar las sucursales diferentes de lebrija y girón
- Consultar las sucursales con activos mayores a 10000
- Consultar los clientes de Girón y Bucaramanga
- Recuperar los préstamos mayores o iguales a 2500000, incluir el número y la sucursal donde se realizó el préstamo
- Consultar los clientes que viven en la calle “Los angeles” o en la calle “Americas” trayendo su nombre, id y ciudad

Structured Query Language - SQL

Avanzado

Uniones entre relaciones (avanzado)

- Inner Join
- Left Outer Join
- Left Join
- Right Join

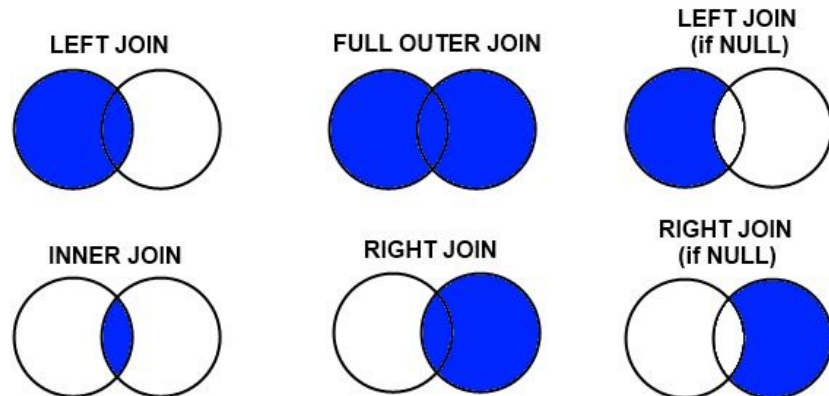


Consultas SQL

Uniones (avanzado)

- Inner Join
- Left Outer Join
- Left Join
- Right Join

Unión de conjuntos



Select *
From relacionA
 Join relacionB **on** atrA=atrB

Consultas SQL

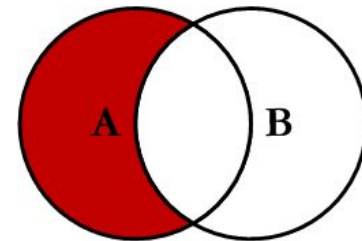
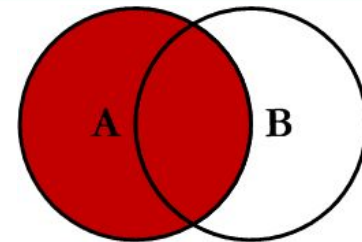
Uniones (avanzado)

Ejemplo Left Join: Consulta de los clientes con o sin cuentas

Select c.*

From clientes c

Left Join cuentas cu **on** c.cliente_id=cu.cliente_id



Modificación de base de datos

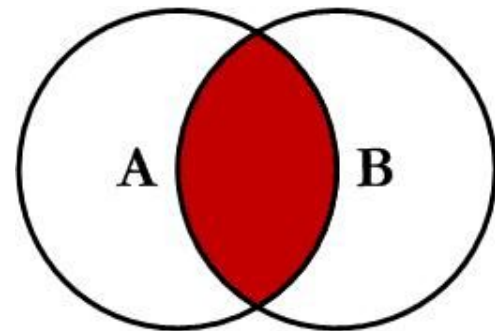
Uniones (avanzado)

Ejemplo inner Join: seleccionar solo los clientes con préstamo

Select c.*

From clientes c

Inner Join prestamos p **on** c c.cliente_id=p.cliente_id



Operaciones sobre conjuntos

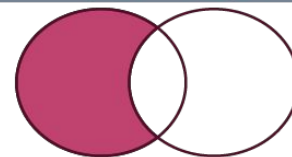
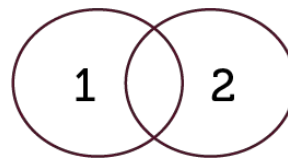
Ejercicio:

Teniendo en cuenta las siguientes relaciones:

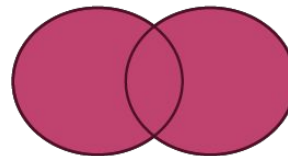
Cliente(id, nombre, direccion, telefono)

compraCarro(id_carro, id_cliente, id_cliente_v, modelo, marca, valor, fecha)

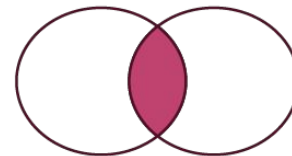
rentaCarro(id_carro, id_cliente, dias, fecha)



Diferencia



Union



Intersección

1. Consulta los clientes (con duplicados) que han comprado carros marca "Mazda" y han rentado carros por más de 1 día
2. Consulta los clientes que han comprado carros de modelos mayor al 2000 pero nunca han rentado

Ejercicio

Ejercicio Clase parte 3:

- Recuperar el nombre y la ciudad de los clientes que tienen préstamos con el banco agregando además el valor del préstamo
- Recuperar el nombre, ciudad y activos de las sucursales que han abierto cuentas con saldos mayores a 35000
- Consultar el nombre de los clientes que han abierto cuenta de una sucursal ubicada en la misma ciudad donde residen.
- Recuperar el número de préstamo y el nombre del cliente a quien se le aprobó préstamos mayores a 450000