

Bases de Datos

Carrera Programador full-stack

Consultas

Lenguaje de manipulación de datos - DML

- Permite generar consultas para ordenar, filtrar y extraer datos de la base de datos:
 - **SELECT:** Consulta registros de la base de datos que satisfagan un criterio determinado
 - **INSERT:** Carga lotes de datos en la base de datos en una única operación
 - **UPDATE:** Modifica los valores de los campos y registros especificados
 - **DELETE:** Elimina registros de una tabla de una base de datos

SELECT

La sentencia ***SELECT*** sirve para recuperar datos de la BD

```
SELECT <lista de atributos>  
FROM <lista de tablas>;
```

- SELECT identifica las columnas a recuperar (el ***qué***)
- FROM identifica la tabla (de ***dónde*** obtener los datos)
- El resultado de una consulta es una **tabla** (si es un número, se considera como una tabla con una fila y una columna)
- <lista de atributos> es una lista de los atributos cuyos valores serán recuperados por la consulta
- <lista de tablas> es una lista de las tablas necesarias para procesar la consulta

SELECT

Selección sin Condición

apellido



```
SELECT  
  <nombre_columna>, <nombre_columna>  
FROM <nombre_tabla>;
```

Ejemplo: Seleccionar los nombres y apellidos de los clientes registrados en la base

```
SELECT nombre, apellido  
FROM e01_cliente;
```

Selección sin Condición

Ejemplo

Seleccionar los nombres y apellidos de los clientes registrados en la base

```
SELECT nombre, apellido  
FROM e01_cliente;
```

nombre	apellido
Xerxes	Hale
Brent	Leblanc
Kasper	Shannon
...	...

SELECT

Selección de Todas las Columnas

```
SELECT *  
FROM <nombre_tabla>;
```

Ejemplo: Obtener todos los datos de todos los clientes

```
SELECT *  
FROM e01_cliente;
```

Selección de Todas las Columnas

Ejemplo

```
SELECT *  
FROM e01_cliente;
```

nro_cliente	nombre	apellido	dirección	activo
1	Xerxes	Hale	129-5974 Suspendisse Ctra.	89
2	Brent	Leblanc	Apartado núm.: 372, 5244 Nibh.	58
3	Kasper	Shannon	Apdo.:304-6908 Class Ctra.	38
...

SELECT

Selección sin Valores Repetidos

```
SELECT DISTINCT <nombre_columna>,  
    <nombre_columna>  
FROM <nombre_tabla>;
```

Ejemplo: Obtener las distintas marcas de los productos

```
SELECT DISTINCT marca  
FROM e01_producto;
```


Selección sin Valores Repetidos

Ejemplo

Obtener las distintas marcas de los productos

```
SELECT DISTINCT  
  marca  
FROM  
  e01_producto;
```

marca

Dolor Sit Incorporated

A Felis Ullamcorper Company

Elit A Corp.

...

SELECT

Selección con Condición

La cláusula **WHERE** se usa para establecer una condición sobre los datos que retorna la consulta

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <condición>;
```

- <condición> es una expresión condicional (booleana) que se evalúa en las filas de la tabla
- Una cláusula WHERE contiene una condición lógica, la cual usa **operadores de comparación** u **operadores lógicos** (AND, OR, NOT)

Operadores de Comparación

Operador	Descripción
=	Igual
<>	Distinto
>	Mayor que
<	Menor que
<=	Mayor o igual que
>=	Menor o igual que

Operadores de Comparación

Operador	Descripción
BETWEEN	Entre (para comparar valores en un rango determinado)
LIKE	Parecido (se usa para buscar patrones)
IN	Para especificar múltiples posibles valores para una columna

Restricción de Consultas

Cláusula WHERE

- Los operadores de comparación se utilizan en la cláusula *WHERE* para comparar expresiones
- El resultado de la comparación puede ser:
 - Verdadero (T)
 - Falso (F)
 - Desconocido (U)

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <nombre_columna> <operador>  
      <valor>;
```

Restricción de Consultas

Ejemplo

Seleccionar los nombres y las marcas de los productos cuyo precio sea superior a 900\$

```
SELECT nombre,  
        marca, precio  
FROM e01_producto  
WHERE precio > 900;
```

nombre	marca	precio
fish	Aptent Taciti Incorporated	943.44
receipt	Amet Consectetuer Industries	908.77
half price	Auctor Corporation	946.39
cosmetics	Montes Nascetur Incorporated	926.6
ticket	Morbi Neque Tellus Incorporated	903.9

Restricción de Consultas

Ejemplo

Obtener la marca del producto cuyo código es 10.

```
SELECT marca, codigo_producto  
FROM e01_producto  
WHERE codigo_producto=10;
```

marca	codigo_producto
Lobortis Nisi Nibh Consulting	10

Operador BETWEEN

BETWEEN incluye los valores de los extremos dentro del rango

- BETWEEN x AND y es equivalente a $a \geq x$ AND $a \leq y$
- NOT BETWEEN x AND y es equivalente a $a < x$ OR $a > y$

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <nombre_columna> BETWEEN <x> AND <y>;
```


Operador BETWEEN

Ejemplo

Seleccionar código y nombre de los productos que tengan un stock de entre 60 y 100 unidades

```
SELECT codigo_producto,  
       nombre, stock  
FROM e01_producto  
WHERE stock BETWEEN 60  
       AND 100;
```

codigo_producto	nombre	stock
11	bakery	73
26	drugstore	73
28	drugstore	73
84	cheque	70
90	delicatessen	81
...

Operador LIKE

El operador **LIKE** busca en una columna según un patrón específico. Por ejemplo todos los nombres que empiezan con determinada letra. También se emplea en la forma negativa **NOT LIKE**.

- Comodines
 - %: cualquier secuencia de cero o más caracteres
 - _ : denota un solo carácter

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <nombre_columna> LIKE <er>;
```

Operador LIKE

Ejemplos

Seleccionar apellido y código del cliente con nombre “Jescie”

```
SELECT  
  nro_cliente, apellido, nombre  
FROM e01_cliente  
WHERE nombre LIKE 'Jescie';
```

nro_cliente	nombre	apellido
31	Jescie	Wong
49	Jescie	Daugherty

Operador LIKE

Ejemplos

Seleccionar código, nombre y apellido de clientes cuyo nombre empiece con la letra "F"

```
SELECT nro_cliente, nombre, apellido  
FROM e01_cliente  
WHERE nombre LIKE "F%";
```

nro_cliente	nombre	apellido
82	Faith	Sutton
88	Francis	Barker

Condiciones Compuestas

Un operador lógico combina los resultados de dos condiciones para producir un único resultado

- Los operadores **AND** y **OR** se pueden usar para componer expresiones lógicas
- El operador AND retorna VERDADERO si ambas condiciones evaluadas son VERDADERAS, mientras que el operador OR retorna VERDADERO si alguna de las condiciones es VERDADERA

Operadores Lógicos

OR

Se utiliza para pedir que se cumpla al menos una condición

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <condición> OR <condicion>;
```

OR

Ejemplo

Obtener el código, nombre y stock de los productos cuyo nombre sea “fish” o tengan un stock menor/igual a 26 unidades

```
SELECT codigo_producto,  
       nombre, stock  
FROM e01_producto  
WHERE (nombre LIKE 'fish')  
       OR (stock <= 26);
```

codigo_producto	nombre	stock
12	fish	498
30	fish	204
63	fish	100
99	cash only	12

Operadores Lógicos

AND

Se utiliza para pedir que se cumplan dos condiciones a la vez

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <condición> AND <condición>;
```


AND

Ejemplo

Seleccionar código y nombre de los productos que tengan un stock de entre 60 y 90 unidades

```
SELECT codigo_producto,  
        nombre, stock  
FROM e01_producto  
WHERE stock >= 60 AND  
        stock <= 90;
```

codigo_producto	nombre	stock
11	bakery	73
26	drugstore	73
28	drugstore	73
84	cheque	70
90	delicatessen	81

Bases de Datos

Carrera Programador full-stack

Consultas Práctica

Consultas Simples

SELECT

1. Obtener todos los datos de todos los clientes
2. Obtener solo los nombres y apellidos de todos los clientes
3. Obtener los nombres de los diferentes productos
4. Obtener los diferentes códigos de área de los teléfonos

Consultas Simples

Restricciones

5. Obtener el listado de todos los productos que tengan un stock mayor a 50 y menor a 200
6. Obtener los datos correspondientes al producto cuyo código es 50
7. Obtener los datos de las facturas cuyo total (con iva incluido) sea mayor a 40.000\$ y lo haya realizado el cliente número 8
8. Obtener los datos del cliente cuyo nombre es “Ivor” y el apellido “Saunders”
9. Todas las Facturas pertenecientes al cliente número 10
10. Todas las Facturas que superen los 50.000\$ (iva incluido)

Bases de Datos

Carrera Programador full-stack

Consultas

Consultas Simples

Repaso

- La sentencia ***SELECT*** sirve para recuperar datos de la BD
- Una sentencia **SELECT** devuelve una tabla
- Se utiliza * para recuperar todos los campos de la fila
- La cláusula ***WHERE*** se usa para realizar las **restricciones**

Sintaxis

Repaso

```
SELECT <lista de atributos>  
FROM <nombre_tabla>  
WHERE <condición>;
```

- <condición> es una expresión condicional (booleana) que se evalúa en las filas de la tabla
- <lista de atributos> es una lista de los atributos cuyos valores serán recuperados por la consulta
- <lista de tablas> es una lista de las tablas necesarias para procesar la consulta

Consultas Simples

Repaso

- La cláusula ***DISTINCT*** se utiliza para no recuperar valores repetidos.

```
SELECT DISTINCT <nombre_columna>,  
    <nombre_columna>  
FROM <nombre_tabla>;
```


Consultas Simples

Repaso

- Dentro de la *condición* del **WHERE** se pueden utilizar operadores aritméticos ($<$, $>$, $<=$, $>=$, $=$, $<>$) y lógicos para combinar más de una condición (**OR**, **AND** y **NOT**)
- Otros operadores que también se pueden utilizar son
 - **Between**
 - **Like**
 - **In**

Lenguaje SQL

Algunas funciones del estándar SQL son:

- **DDL**

- Definición de datos:
 - Creación de tablas (CREATE)
 - Modificación de tablas (ALTER)
 - Eliminación de tablas (DROP)

- **DML**

- **Consulta de datos**
 - **Selección (SELECT)**
- Actualización de los datos
 - Inserción (INSERT)
 - Actualización (UPDATE)
 - Eliminación (DELETE)

Bases de Datos

Carrera Programador full-stack

Consultas (Resolución Ejercicios)

Consultas Simples

SELECT

Problema: Obtener todos los datos de todos los clientes

Solución:

```
SELECT *  
FROM e01_cliente;
```

Consultas Simples

SELECT

Problema: Obtener solo los nombres y apellidos de todos los clientes

Solución:

```
SELECT nombre, apellido  
FROM e01_cliente;
```

Consultas Simples

SELECT

Problema: Obtener los nombres nombres de los diferentes productos

Solución:

```
SELECT DISTINCT nombre  
FROM e01_producto;
```

Consultas Simples

SELECT

Problema: Obtener los diferentes códigos de área de los teléfonos

Solución:

```
SELECT DISTINCT codigo_area  
FROM e01_telefono;
```

Consultas Simples

Restricciones

Problema: Obtener el listado de todos los productos que tengan un stock mayor a 50 y menor a 200

Solución:

```
SELECT codigo_producto, nombre  
FROM e01_producto  
WHERE stock BETWEEN 50 AND 200;
```


Consultas Simples

Restricciones

Problema: Obtener los datos correspondientes al producto cuyo código es 50

Solución:

```
SELECT *  
FROM e01_producto  
WHERE codigo_producto = 50;
```

Consultas Simples

Restricciones

Problema: Obtener los datos de las facturas cuyo total (con iva incluido) sea mayor a 400.000\$ y lo haya realizado el cliente número 8

Solución:

```
SELECT *  
FROM e01_factura  
WHERE total_con_iva > 400000  
AND nro_cliente = 8;
```

Consultas Simples

Restricciones

Problema: Obtener los datos del cliente cuyo nombre es “Ivor” y el apellido “Saunders”

Solución:

```
SELECT *  
FROM e01_cliente  
WHERE nombre LIKE 'Ivor' AND apellido LIKE  
'Saunders';
```

Consultas Simples

Restricciones

Problema: Todas las Facturas pertenecientes al cliente número 10

Solución:

```
SELECT *  
FROM e01_factura  
WHERE nro_cliente = 10;
```

Consultas Simples

Restricciones

Problema: Todas las Facturas que superen los 500.000\$

Solución:

```
SELECT *  
FROM e01_factura  
WHERE total_con_iva > 500000;
```