

El comando **SELECT** se utiliza para extraer datos de una o más tablas.

Es el comando más básico y el **más utilizado** en SQL



Comando **SELECT**

Sintaxis básica

SELECT columna1, columna2, ...

FROM nombre_tabla;

columna1, columna2:

Las columnas que deseas recuperar.

nombre_tabla:

La tabla de la que se extraen los datos.



El comando **SELECT** es muy poderoso y tiene varias características avanzadas adicionales para profundizar.



Funciones **Agregadas**

Función agregada **SUM()**

SUM() es una función de agregación que se utiliza en ANSI SQL y es ampliamente soportada por casi todos los sistemas de bases de datos relacionales que implementan SQL. La función **SUM** se usa para **sumar** los valores de una columna numérica.



Actividad de aplicación

Ejemplo 1: selección de una columna para mostrar la suma de sus elementos.

Actividad: recuperar la suma de la columna edad de una tabla llamada clientes.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT SUM(edad)
FROM clientes;
```

Resultado:

SUM(edad)
77

Función agregada **AVG()**

AVG() es una función que se usa en ANSI SQL y es soportada por la mayoría de los sistemas de bases de datos relacionales. La función **AVG** se usa para calcular el **promedio** de los valores de una columna numérica.



Actividad de aplicación

Ejemplo 2: selección de una columna para mostrar el promedio de sus elementos.

Actividad: recuperar el promedio de la columna edad de una tabla llamada clientes.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT AVG(edad)
FROM clientes;
```

Resultado:

AVG(edad)
25.6666666668

Función agregada **MIN()**

MIN() es una función de agregación que se utiliza en ANSI SQL y es ampliamente soportada por casi todos los sistemas de bases de datos relacionales que implementan SQL. La función **MIN** se usa para obtener el valor mínimo en una columna numérica.



Actividad de aplicación

Ejemplo 3: selección de una columna para mostrar el valor mínimo de sus elementos.

Actividad: recuperar el valor mínimo de la columna edad de una tabla llamada clientes.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT MIN(edad)
FROM clientes;
```

Resultado:

MIN(edad)
22

Función agregada **MAX()**

MAX() es una función de agregación que se utiliza en ANSI SQL y es ampliamente soportada por casi todos los sistemas de bases de datos relacionales que implementan SQL. La función **MAX** se usa para obtener el valor máximo en una columna numérica.



Actividad de aplicación

Ejemplo 4: selección de una columna para mostrar el valor máximo de sus elementos.

Actividad: recuperar el valor máximo de la columna edad de una tabla llamada clientes.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT MAX(edad)
FROM clientes;
```

Resultado:

MIN(edad)
30

Características ALIAS AS

ALIAS se utiliza para renombrar columnas temporalmente en los resultados. Es útil cuando queremos que las columnas tengan nombres más amigables.



Actividad de aplicación

Ejemplo 5: selección de una columna para mostrar el valor de la misma empleando un alias.

Actividad: recuperar el nombre de los clientes de una tabla llamada clientes utilizando el alias Cliente.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT nombre AS Cliente  
FROM clientes;
```

Resultado:

Cliente
Juan
María
Andrés

Características CONCATENACION ||

(doble barra vertical)

De acuerdo con el estándar ANSI SQL se utiliza el operador **||** (doble barra vertical) de concatenación de cadenas para unir dos o más cadenas. Esta es la forma recomendada por ANSI SQL.

Aunque **CONCAT()** no es parte de ANSI SQL, se usa ampliamente en varias bases de datos. Ej.:

```
SELECT CONCAT(columna1,'-', columna2) as alias  
FROM nombre_tabla
```

En otros casos en donde tampoco se usa **||** para concatenación, se utiliza **+** (operador de adición) para concatenar cadenas. Ej.:

```
SELECT cast(columna1 as varchar) +  
cast(columna2 as varchar) as alias  
FROM nombre_tabla
```



Actividad de aplicación

Ejemplo 6: selección de dos columnas para mostrar el valor de estas empleando concatenación.

Actividad: recuperar el nombre y la edad de los clientes de una tabla llamada clientes utilizando concatenación.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT nombre || "-" || edad AS Cliente  
FROM clientes;
```

Resultado:

Cliente
Juan-25
María-30
Andrés-22

SubConsultas o Consultas anidadas (SubQueries o NestedQueries)

Una subconsulta consulta anidada es una consulta dentro de otra consulta. Es útil para recuperar datos de una consulta dependiente de otra.



Actividad de aplicación

Ejemplo 7: selección de columna nombre para mostrar los clientes cuya columna edad sea mayor que el promedio de la edad de los clientes.

Actividad: recuperar el nombre de los clientes cuya edad es mayor que el promedio de todas las edades de una tabla llamada clientes.

Tabla clientes:

id_cliente	nombre	edad	ciudad
1	Juan	25	Buenos Aires
2	María	30	Córdoba
3	Andrés	22	Rosario

Consulta:

```
SELECT nombre
FROM clientes
WHERE edad > (SELECT AVG(edad) FROM clientes);
```

Resultado:

nombre
María

Trabajo Práctico A22

