



UNIVERSIDAD

Nicolás Felipe Bernal Gallo

Juan Daniel Bogotá Fuetes

Modelos y bases de datos

MBDA AUTO ESTUDIO 1

Guía Auto Estudio #1 SQL Básico
05/02/2025

PROFESOR: María Irma Diaz Rozo y Laura
Catalina Herrera Correa

MODELOS Y BASES DE DATOS

SQL Básico

2025-1

Guía autoestudio 1/ 6

OBJETIVOS

Desarrollar competencias básicas para escribir consultas simples en SQL

- Consultas con proyecciones, restricciones y producto cruz
`SELECT .. FROM .. WHERE ..`
- Dar nuevos nombres
`AS`
- Resultados sin repeticiones
`DISTINCT`
- Orden en el resultado de consulta
`ORDER BY`
- Consultas que requieren agrupamiento
`GROUP BY ... HAVING ...`
- Operadores para expresiones
 - Numéricos: `ABS, ROUND, FLOOR, CEIL, DIV, MOD`
 - Lógicos: `AND, OR, NOT`
 - De comparación: `=, !=, <>, >, <, >=, <=, BETWEEN`
 - Cadenas: `LEN, INSTR, SUBSTR, REPLACE, TRIM, CONCAT, LIKE (% _)`
 - Tiempo: `CURRENT_DATE, CURRENT_TIMESTAMP, EXTRACT, TO_CHAR`
 - Agrupamiento: `AVG, COUNT, MAX, MIN, SUM`
 - Condicionales: `CASE`
 - Cambio de tipo: `CAST`

ENTREGA

Publicar las respuestas en el espacio correspondiente en un archivo **.zip**, el nombre de este archivo debe ser la concatenación en orden alfabético de los primeros apellidos de cada uno de los miembros.

INVESTIGACION

Para que tener conocimiento suficiente sobre la tecnología que vamos a usar en el laboratorio, investiguen las siguientes preguntas. No olviden incluir la bibliografía.

A. SQL

- ¿Qué es? ¿Para qué sirve?
- ¿Qué es DML, DDL, DCL, TCL?
- En este autoestudio, ¿en qué escribimos? ¿por qué?

B. Motor de bases de datos y bases de datos

- ¿Qué son?
- ¿Qué motores ofrece sqlzoo.net [<http://sqlzoo.net/>]?
- ¿Qué bases de datos ofrece sqlzoo?

SQL Select

SQL Select Distinct

SQL Where

SQL Order By

SQL Group By

SQL Having

SQL Count

SQL Sum

SQL Avg

PRACTICA

Usando w3schools SQL Tutorial [SQL Tutorial]

[En [auto01.doc](#)]

A. Estudien las secciones SQL SELECT, SQL SELECT DISTINCT, SQL WHERE, SQL ORDER BY, SQL GROUP BY, SQL HAVING y escriban expresiones las consultas ejemplo en cálculo o algebra.¹

[Escriban las sentencias en calculo y algebra [auto01.doc](#). Si no lograron escribir alguna sentencia indiquen la razón y el punto de problema]

Usando SQLzoo.net [<http://sqlzoo.net/>]

[En [auto01.doc](#)]

A. Estudien la section FUNCTIONS de la referencia, seleccionen 3 funciones y escriban 3 consultas que usen [event](#) de la base de datos MySQL [University Timetables](#) [Escriban las sentencias en [auto01.doc](#), ejecútenlas en sqlzoo y capturen el resultado. Si no lograron escribir alguna sentencia indiquen el punto de problema]

B.

`event (id,mdle,kind,dow,tod,duration,room)`

C. Realicen todos los ejercicios propuestos en los siguientes tutoriales y presente los quices. Utilice el motor My SQL.

- SELECT basics, quiz,
- SELECT from world, quiz,
- SELECT from nobel, quiz
- SELECT in SELECT, quiz
- SUM and COUNT, quiz

[Tutoriales: escriban las respuestas a las preguntas que no la tienen respuesta en [auto01.doc](#), ejecútenlas en sqlzoo y capturen el resultado. Quices: escriban en [auto01.doc](#) el resultado obtenido de los quices]

D. Propongan consultas que cumplan los siguientes requerimientos. Usen [event](#) de la base de datos MySQL [University Timetables](#)

[Escriban la consulta en lenguaje natural y la sentencia en SQL en [auto01.doc](#) y ejecuten la sentencia SQL en sqlzoo. Si no lograron escribir alguna sentencia indiquen el punto de problema]

- 8 consultas: una para cada uno de los tipos de operadores para expresiones.
- 3 consultas anidadas que usen otra consulta: 1) (SELECT ...) en FROM, 2) SELECT en WHERE y 3) SELECT ... en SELECT
- 3 consultas con el siguiente esquema: 1) GROUP BY ... HAVING ... 2) ORDER BY 3) DISTINCT

¹ (1+1+3)

INVESTIGACIÓN

A. SQL:

- ¿Qué es? ¿Para qué sirve?

SQL (Structured Query Language) en español Lenguaje de consulta estructurado, es un lenguaje de programación que se usa en bases de datos relacionales para modificar, guardar y usar datos en dicha base.

- ¿Qué es DML, DDL, DCL, TCL?
- DML: Data Manipulation Language (DML) es una herramienta que permite gestionar y manipular datos en una base de datos relacional. Este es un subconjunto de SQL, los comandos de DML se utilizan para insertar, actualizar, eliminar y recuperar datos en una base de datos. Estos comandos permiten a los usuarios cambiar los datos almacenados en la base de datos sin afectar al esquema de la base de datos ni a la definición de los objetos de la base de datos.
- DDL: Data Definition Language (DDL) es un lenguaje de programación para definir estructuras de datos, proporcionando a los sistemas gestores de bases de datos. Define las estructuras que almacenan los datos así como los procedimientos o funciones que permiten consultarlos.
- DCL: Data Control Language(DCL) se centra en definir y regular la configuración de seguridad, los privilegios y los niveles de acceso de los usuarios dentro de un sistema de base de datos. Garantiza que los administradores de las bases de datos puedan proteger datos confidenciales, mantener la confidencialidad de los datos y hacer cumplir las reglas de autorización dentro del entorno de la base de datos.
- TCL: Transaction Control Language(TCL) es un aspecto importante de las bases de datos relacionales y desempeña un papel crucial en la administración de transacciones dentro de un sistema de gestión de bases de datos (DBMS). Proporciona las herramientas necesarias para mitigar tales problemas, asegurando la integridad de los datos mediante propiedades de atomicidad, consistencia, aislamiento, durabilidad (ACID), que son características esenciales de cualquier sistema de bases de datos transaccional confiable.
- En este autoestudio, ¿en qué escribimos? ¿por qué?

Escribimos en el lenguaje tradicional, en español. Porque es la forma en la que nos comunicamos.

B. Motor de bases de datos y bases de datos:

- ¿Qué son?
- Un motor de bases de datos es el software que gestiona el almacenamiento, recuperación y manipulación de datos dentro de la base de datos. Proporciona las herramientas necesarias para ejecutar las consultas, manejar transacciones y garantizar la integridad de los datos.

Ejemplos:

- MySQL
- Oracle
- SQL Server
- DB2
- Postgres
- Una base de datos es una colección organizada de datos estructurados que se almacenan y gestionan mediante un motor de bases de datos. Pueden llegar a contener varias tablas con la información relacionada.
- ¿Qué motores ofrece sqlzoo.net [<http://sqlzoo.net/>]?
 - SQL Zoo proporciona varios tutoriales y ejercicios prácticos en su pagina con respecto al aprendizaje de SQL utilizando diferentes motores de bases de datos, ya mencionados en el punto anterior como ejemplos.
- **Motores de bases de datos:**
 - MySQL
 - Oracle
 - SQL Server
 - DB2
 - PostgreSQL
 - Ingres

- ¿Qué bases de datos ofrece sqlzoo?

Según lo visto en su pagina y luego de practicar un poco, pudimos encontrar que proporciona bases de datos de ejemplo para realizar consultas SQL. Las bases que encontramos fueron con respecto a datos de empleados que contienen información con respecto a los departamentos y los mismos trabajadores. Por otra parte también observamos que contienen bases de datos de ciudades que tienen información sobre las ciudades y su población.

PRACTICA

Usando w3schools SQL Tutorial [SQL Tutorial]

[En [auto01.doc](#)]

A. Estudien las secciones SQL SELECT, SQL SELECT DISTINCT, SQL WHERE, SQL ORDER BY, SQL GROUP BY, SQL HAVING y escriban expresiones las consultas ejemplo en cálculo o algebra. ¹

[Escriban las sentencias en calculo y algebra [auto01.doc](#). Si no lograron escribir alguna sentencia indiquen la razón y el punto de problema]

SQL SELECT:

Example

Return all the columns from the Customers table:

```
SELECT * FROM Customers;
```

Try it Yourself »

Álgebra Relacional:

$$\pi(\text{Customers})$$

Cálculo Relacional:

$$\{X \mid X \in \text{Customers}\}$$

Example

Get your own SQL Server

Return data from the Customers table:

```
SELECT CustomerName, City FROM Customers;
```

Try it Yourself »

Álgebra Relacional:

$$\pi_{\text{CustomerName}, \text{City}}(\text{Customers})$$

Cálculo Relacional:

$$\{X.\text{CustomerName}, X.\text{City} \mid X \in \text{Customers}\}$$

SQL SELECT DISTINCT:

Example

Select all the different countries from the "Customers" table:

```
SELECT DISTINCT Country FROM Customers;
```

Try it Yourself »

Get your own SQL Server

Álgebra Relacional:

$$\pi_{Country}(Customers)$$

Cálculo Relacional:

$$\{X.Country \mid \exists X \text{ Customers}\}$$

Example

```
SELECT Country FROM Customers;
```

Try it Yourself »

Álgebra Relacional:

$$\pi_{Country}(Customers)$$

Cálculo Relacional:

$$\{X.Country \mid \exists X \in \text{Customers}\}$$

Example

```
SELECT COUNT(DISTINCT Country) FROM Customers;
```

Álgebra Relacional:

$$\pi_{Country}(Customers)$$

Cálculo Relacional:

$$COUNT(\{X.Country \mid \exists X \in \text{Customers}\})$$

Example

```
SELECT Count(*) AS DistinctCountries  
FROM (SELECT DISTINCT Country FROM Customers);
```

Álgebra Relacional:

$$COUNT(\pi_{Country}(Customers))$$

Cálculo Relacional:

$$COUNT(\{X.Country \mid \exists X \in \text{Customers}\})$$

SQL WHERE:

Example

[Get your own SQL Server](#)

Select all customers from Mexico:

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\sigma_{Country='Mexico'}(Customers)$$

Cálculo Relacional:

$$\{X \mid X \in Customers \wedge X.Country = 'Mexico'\}$$

Example

```
SELECT * FROM Customers
WHERE CustomerID=1;
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\sigma_{CustomerID=1}(Customers)$$

Cálculo Relacional:

$$\{X \mid X \in Customers \wedge X.CustomerID = 1\}$$

Example

Select all customers with a CustomerID greater than 80:

```
SELECT * FROM Customers
WHERE CustomerID > 80;
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\sigma_{CustomerID>80}(Customers)$$

Cálculo Relacional:

$$\{X \mid X \in Customers \wedge X.CustomerID > 80\}$$

SQL ORDER BY:

Example[Get your own SQL Server](#)

Sort the products by price:

```
SELECT * FROM Products
ORDER BY Price;
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\pi(Products)$$

Cálculo Relacional:

$$\{X \mid X \in Products\}$$

Example

Sort the products from highest to lowest price:

```
SELECT * FROM Products
ORDER BY Price DESC;
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\pi(Products)$$

Cálculo Relacional:

$$\{X \mid X \in Products\}$$

Example

Sort the products alphabetically by ProductName:

```
SELECT * FROM Products
ORDER BY ProductName;
```

[Try it Yourself »](#)

Álgebra Relacional:

$$\pi(Products)$$

Cálculo Relacional:

$$\{X \mid X \in Products\}$$

Example

Sort the products by ProductName in reverse order:

```
SELECT * FROM Products
ORDER BY ProductName DESC;
```

Try it Yourself »

Álgebra Relacional:

$$\pi(Products)$$

Cálculo Relacional:

$$\{X \mid X \in Products\}$$

Example

```
SELECT * FROM Customers
ORDER BY Country, CustomerName;
```

Try it Yourself »

Álgebra Relacional:

$$\pi(Customers)$$

Cálculo Relacional:

$$\{X \mid X \in Customers\}$$

Example

```
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```

Try it Yourself »

Álgebra Relacional:

$$\pi(Customers)$$

Cálculo Relacional:

$$\{X \mid X \in Customers\}$$

SQL GROUP BY:

Example

[Get your own SQL Server](#)

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

[Try it Yourself »](#)

Álgebra Relacional:

El álgebra relacional no tiene un operador directo para realizar agregaciones como **COUNT** o **GROUP BY**. Bajo nuestro conocimiento no pudimos realizar este punto.

Cálculo Relacional:

$$\{(X.Country, COUNT(C.CustomerID)) \mid X \in X \text{ Customers}\}$$

Example

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

[Try it Yourself »](#)

Álgebra Relacional:

El álgebra relacional no tiene un operador directo para realizar agregaciones como **COUNT** o **GROUP BY**. Bajo nuestro conocimiento no pudimos realizar este punto.

Cálculo Relacional:

$$\{(X.Country, COUNT(C.CustomerID)) \mid X \in X \text{ Customers}\}$$

Example

```
SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY ShipperName;
```

[Try it Yourself »](#)

Álgebra Relacional y Cálculo Relacional:

No pudimos realizar este punto puesto según lo que vimos en la teoría creemos que no es suficiente para poder desarrollar este tipo de ejercicio. No entendemos muy bien si sería proyección o restricción. A partir de esto no pudimos plantear las ecuaciones con respecto al Álgebra y el Cálculo relacional.

BIBLIOGRAFÍA:

1. AppMaster.io. (s.f.). *DML: Lenguaje de Manipulación de Datos*. Recuperado el [05/02/2025], de <https://appmaster.io/es/blog/dml-lenguaje-de-manipulacion-de-datos>
2. TodoPostgreSQL. (s.f.). *Diferencias entre DDL, DML y DCL*. Recuperado el [05/02/2025], de <https://www.todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/>
3. IBM. (s.f.). *Lenguaje de control de datos (DCL)*. Recuperado el [05/02/2025], de <https://www.ibm.com/docs/es/psfa/7.1.0?topic=categories-data-control-language>
4. AppMaster.io. (s.f.). *Lenguaje de control de datos (DCL)*. Recuperado el [05/02/2025], de <https://appmaster.io/es/glossary/lenguaje-de-control-de-datos-dcl>
5. AppMaster.io. (s.f.). *Lenguaje de control de transacciones (TCL)*. Recuperado el [05/02/2025], de <https://appmaster.io/es/glossary/lenguaje-de-control-de-transacciones-tcl>
6. Platzi. (s.f.). *¿Qué es DDL? Significado de DML, DCL y TCL*. Recuperado el [05/02/2025], de <https://platzi.com/tutoriales/50-sql-mysql-2016/1564-que-es-ddl-dml-dcl-y-tcl-integridad-referencial/>