

CONCEPTOS

The background of the slide is a photograph of a serene landscape. In the foreground, a wooden dock made of planks extends from the bottom left towards the center. On the right side of the dock, there is a wooden Adirondack chair with a slatted back and seat, facing away from the viewer towards the lake. The middle ground is a calm body of water, likely a lake, which reflects the sky and the distant mountains. In the background, there are large, forested mountains under a sky with soft, white clouds. The overall mood is peaceful and natural.

**M. En C. Niels Henrik
nAvarrete Manzanilla**



SQL RDBMS

Relation Database Management System.

SQL Server, IBM DB2, Oracle, MySql y Microsoft Acces

SQL

- Significa Structured Query Language, en español **lenguaje de consulta estructurado**, es un lenguaje para el acceso a bases de datos relacionales, originado en IBM.
- Actualmente se ha convertido en un **estándar** de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores.

Proceso de SQL

Cuando se ejecuta una instrucción en SQL cualquier RDBMS, determina el mejor camino para realizar la solicitud generada.

Existen varios componentes que son incluidos en el proceso:

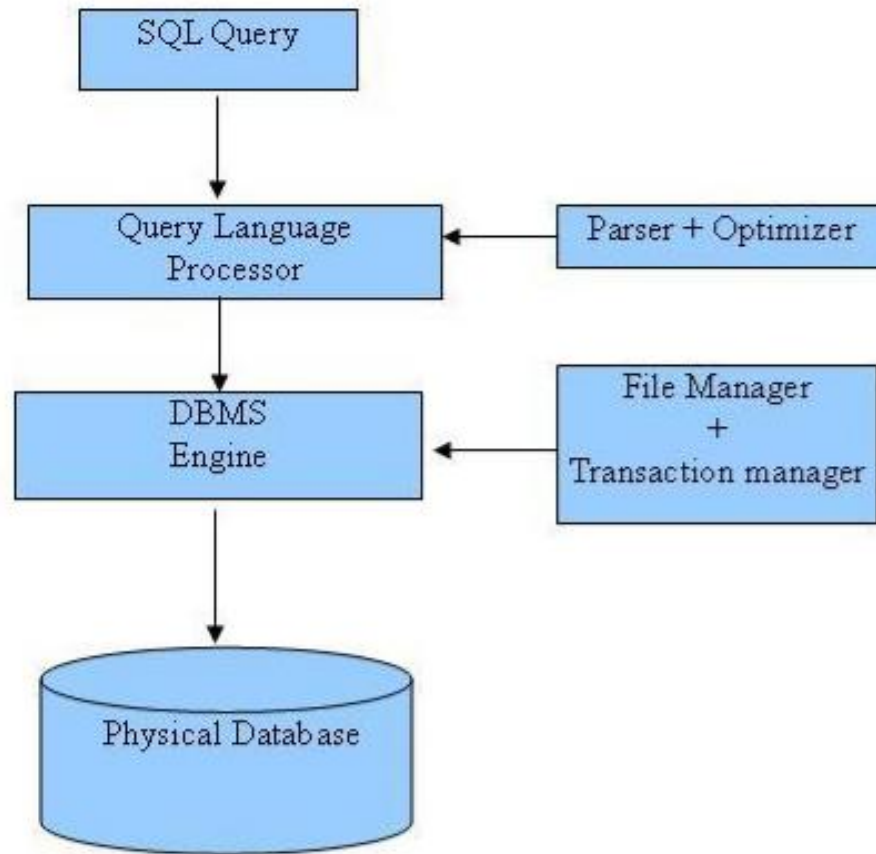
- Dispatcher:

Es responsable, de iniciar los procesos de trabajo y la distribución de la carga de transacciones a través de procesos de trabajo.

- Motor de Optimización

- Motor de consultas

Proceso de SQL





Comandos de SQL

Una sentencia de SQL es como una frase (escrita en inglés) con la que decimos lo que queremos obtener y de donde obtenerlo.

Estructurada con palabras reservadas, que solo el motor de base de datos reconoce, llamadas comandos.

Todas las sentencias empiezan con un verbo (palabra reservada que indica la acción a realizar), seguido de cláusulas, algunas obligatorias y otras opcionales que completan la frase.

Todas las sentencias siguen una sintaxis (orden en el que se escriben las palabras) para que se puedan ejecutar correctamente.

Comandos de SQL

Los comandos estándares para interactuar con una base de datos relacional son:

- CREATE
- SELECT
- INSERT
- UPDATE
- DELETE
- DROP

Estos grupos se clasifican de acuerdo a su naturaleza

DDL

Lenguaje de definición de datos

COMANDO	DESCRIPCIÓN
CREATE	Crea: <ul style="list-style-type: none">• Tabla• Vista• Objetos
ALTER	Modifica un objeto existente como una tabla
DROP	Elimina completamente: <ul style="list-style-type: none">• Tabla• Vista• Objetos

DML

Lenguaje de manipulación de datos

COMANDO	DESCRIPCIÓN
INSERT	Crea/Agrega un registro
UPDATE	Modifica registros
DELETE	Elimina registros

DCL

Lenguaje de control de datos

COMANDO	DESCRIPCIÓN
GRANT	Asigna privilegios al usuario
REVOKE	Quita privilegios al usuario



DQL

Lenguaje de consulta de datos

COMANDO	DESCRIPCIÓN
SELECT	Regresa los registros consultados por una o varias tablas

Tabla

- Es una colección de entradas (datos), y estas se organizan en columnas (atributos) y filas (valores).
- Es la forma mas común y simple de almacenamiento de datos.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Campo

- Cada tabla se puede descomponer en pequeñas entidades que llamaremos campos.
- Un campo es una columna de una tabla, que esta diseñado para mantener información específica sobre cada registro de la tabla.

+	+	+	+	+	+	+
	ID		NAME		AGE	
					ADDRESS	
					SALARY	
+	+	+	+	+	+	+

Registro

- Un registro es cada entrada individual que existe en la tabla, también llamado fila de datos.

Ej.

Existe 7 registros

	1		Ramesh		32		Ahmedabad		2000.00	
	2		Khilan		25		Delhi		1500.00	
	3		kaushik		23		Kota		2000.00	
	4		Chaitali		25		Mumbai		6500.00	
	5		Hardik		27		Bhopal		8500.00	
	6		Komal		22		MP		4500.00	
	7		Muffy		24		Indore		10000.00	

Columna

- Es una entidad vertical perteneciente a una tabla y que contiene toda la información asociada a un campo específico.

ADDRESS
Ahmedabad
Delhi
Kota
Mumbai
Bhopal
MP
Indore

Valor NULL / Nulo

- Cuando se utiliza un valor nulo en algún campo de la tabla, este aparecerá en blanco, lo que significa que no tiene asignando ningún valor.
- Un valor NULL es diferente a un valor cero o un campo que contiene espacios.
- Un campo con un valor NULL es uno que se ha dejado vacío durante su creación.

Restricciones SQL

Son reglas establecidas en los datos de las columnas. Estas sirven para limitar el tipo de dato que puede almacenar la tabla, permitiendo asegurar la fiabilidad de los datos.

- Las restricciones pueden ser de dos tipos:
 - Nivel de columna:
Se aplican solo a las columnas
 - Nivel de tabla:
Se aplican a toda la tabla.

Restricciones comunes en SQL

► NOT NULL

Asegura que las columnas no contenga valores nulos

► DEFAULT

Asigna valores por omisión, cuando no se especifique algún dato.

► UNIQUE

Asegura que todos los valores en la columna sean diferentes.

► PRIMARY KEY

Identificador único de cada registro en la tabla.

Una llave primaria debe ser única y no nula, de manera que garantice que pueda ser siempre referenciada.

Restricciones comunes en SQL

► FOREIGN KEY

Identificador único de cada registro desde otra tabla

Llave foránea es cuando un registro hace referencia a un registro en otra tabla.

► CHECK

Asegura que los valores de la columna satisfaga ciertas condiciones.

► INDEX

Se utiliza para crear y recuperar los datos de manera rápida.

NOT NULL Restricciones

Por omisión una columna puede almacenar valores nulos. Si uno no desea que las columnas almacenen este tipo de valor, entonces se tiene que especificar que columna no lo permita

```
CREATE TABLE CUSTOMERS (  
    ID      INT              NOT NULL,  
    NAME VARCHAR (20)       NOT NULL,  
    AGE     INT              NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY  DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```



NOT NULL Restricciones

Si deseamos modificar la tabla CUSTOMERS, asignando una restricción NOT NULL a la columna SALARY (Oracle y MySQL)

SQL NATIVO

```
ALTER TABLE CUSTOMERS  
  
    MODIFY SALARY    DECIMAL (18, 2) NOT NULL;
```

SQL SERVER

```
ALTER TABLE [Table] ALTER COLUMN [Column] INTEGER NOT NULL
```



Restricción DEFAULT

Asigna un valor por omisión a cierta columna, cuando se inserta un registro y no se proporciono ningún dato a dicha columna

```
CREATE TABLE CUSTOMERS(  
    ID    INT                NOT NULL,  
    NAME  VARCHAR (20)       NOT NULL,  
    AGE   INT                NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY DECIMAL (18, 2) DEFAULT 5000.00,  
    PRIMARY KEY (ID)  
);
```



Default Restricciones

Si deseamos modificar la tabla CUSTOMERS, asignando un Default a la columna SALARY con valor a 25,000

```
ALTER TABLE CUSTOMERS
```

```
MODIFY SALARY DECIMAL (18, 2) DEFAULT 5000.00;
```

```
ALTER TABLE Persons
```

```
ALTER COLUMN City SET DEFAULT 'SANDNES'
```

Si deseamos quitar la restricción Default a la columna SALARY

```
ALTER TABLE CUSTOMERS
```

```
ALTER COLUMN SALARY DROP DEFAULT;
```

```
ALTER TABLE Persons
```

```
ALTER COLUMN City DROP DEFAULT
```



Restricción UNIQUE

Asegura que todos los valores en la columna se han diferentes. Es decir, previene que en cierta columna dos registros tengan valores idénticos.

Ej. Deseamos que la edad se ha única.

```
CREATE TABLE CUSTOMERS (  
  
    ID      INT                NOT NULL,  
    NAME VARCHAR (20)          NOT NULL,  
    AGE     INT                NOT NULL UNIQUE,  
    ADDRESS CHAR (25) ,  
    SALARY  DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```



Restricción UNIQUE

Si deseamos modificar la tabla CUSTOMERS, asignando restricción UNIQUE a la columna edad

```
ALTER TABLE CUSTOMERS  
MODIFY AGE INT NOT NULL UNIQUE;
```

```
ALTER TABLE Persons  
ADD UNIQUE (P_Id)
```

Se puede asignar una restricción con nombre a múltiples columnas.

```
ALTER TABLE CUSTOMERS  
ADD CONSTRAINT myUniqueConstraint UNIQUE (AGE, SALARY);
```

Si deseamos quitar la restricción UNIQUE a la columna SALARY

```
ALTER TABLE CUSTOMERS  
DROP CONSTRAINT myUniqueConstraint;
```

```
ALTER TABLE CUSTOMERS
```

```
DROP INDEX myUniqueConstraint;
```




Restricción Primary Key

- Una llave primaria/clave primaria, es un campo único en la tabla y debe contener valores diferentes y no puede permitir valores nulos.
- Una tabla solo puede permitir una sola llave primaria compuesta por campos individuales o múltiples.
- Si una tabla tiene una clave principal definida en cualquier campo (s), entonces no se puede tener dos registros que tienen el mismo valor de ese campo (s)



Restricción Primary Key

```
CREATE TABLE CUSTOMERS (  
  
    ID      INT                NOT NULL,  
  
    NAME    VARCHAR (20)      NOT NULL,  
  
    AGE     INT                NOT NULL,  
  
    ADDRESS CHAR (25) ,  
  
    SALARY  DECIMAL (18, 2),  
  
    PRIMARY KEY (ID)  
  
);
```

Si deseamos agregar una Primary Key a una columna

```
ALTER TABLE CUSTOMER ADD PRIMARY KEY (ID) ;
```



Restricción Primary Key

* Si se usa ALTER TABLE para agregar una clave primaria, las columnas que van hacer candidatas deben ya estar definidas que no contengan valores nulos (cuando la tabla es creada).

Si queremos agregar múltiples claves primarias a diferentes columnas

```
CREATE TABLE CUSTOMERS (  
    ID      INT                NOT NULL,  
    NAME    VARCHAR (20)       NOT NULL,  
    AGE     INT                NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY  DECIMAL (18, 2),  
    PRIMARY KEY (ID, NAME)  
);
```



Restricción Primary Key

Cuando la tabla ya existe y deseamos agregar claves primarias múltiples.

```
ALTER TABLE CUSTOMERS  
  
ADD CONSTRAINT PK_CUSTID PRIMARY KEY (ID, NAME);
```

Si deseamos quitar las restricciones de clave primaria a toda la tabla.

```
ALTER TABLE CUSTOMERS DROP PRIMARY KEY ;
```

```
ALTER TABLE Persons  
DROP CONSTRAINT pk_PersonID
```



Restricción Foreign Key

Es una clave usada para unir (relacionar) dos tablas. A veces se le puede llamar clave de referencia.

Es una combinación de columnas, las cuales hacen referencias a claves primarias de otras tablas.

```
CREATE TABLE CUSTOMERS (  
  
    ID    INT                NOT NULL,  
  
    NAME  VARCHAR (20)       NOT NULL,  
  
    AGE   INT                NOT NULL,  
  
    ADDRESS CHAR (25) ,  
  
    SALARY DECIMAL (18, 2),  
  
    PRIMARY KEY (ID)  
  
);
```

```
CREATE TABLE Orders  
(  
    O_Id int NOT NULL PRIMARY KEY,  
    OrderNo int NOT NULL,  
    P_Id int FOREIGN KEY REFERENCES Persons(P_Id)  
)
```

```
CREATE TABLE ORDERS (  
  
    ID            INT            NOT NULL,  
  
    DATE          DATETIME,  
  
    CUSTOMER_ID INT references CUSTOMERS(ID),  
  
    AMOUNT        double,  
  
    PRIMARY KEY (ID)  
  
);
```



Restricción Foreign Key

Cuando la tabla ya existe Order y deseamos agregar una foreign key.

```
ALTER TABLE ORDERS
```

```
ADD FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS (ID);
```

```
ALTER TABLE Orders
```

```
ADD FOREIGN KEY (P_Id)
```

```
REFERENCES Persons(P_Id)
```

Si deseamos quitar las restricciones de foreign key.

```
ALTER TABLE ORDERS
```

```
DROP FOREIGN KEY;
```



Restricción Foreign Key

Al definir estas restricciones se debe considerar que el valor que esté en esta columna se debe encontrar en la tabla a la que se hace referencia.

A esto se le conoce como **Integridad Referencial**, la cuál consiste en que los datos que referencian otros (foreign key) deben ser correctos.



Restricción CHECK

Habilita una condición, para verificar si el valor cumple el requisito. Si no lo cumple el dato no va ser guardado en la tabla.

Ej.

Que la tabla no almacene un edad menor de 18 años

```
CREATE TABLE CUSTOMERS (  
    ID      INT              NOT NULL,  
    NAME    VARCHAR (20)     NOT NULL,  
    AGE     INT              NOT NULL CHECK (AGE >= 18),  
    ADDRESS CHAR (25) ,  
    SALARY  DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```




Restricción Check

Cuando la tabla ya existe y deseamos modificar una restricción Check

```
ALTER TABLE CUSTOMERS
```

```
MODIFY AGE INT NOT NULL CHECK (AGE >= 18 );
```

```
ALTER TABLE Persons
```

```
ADD CHECK (P_Id>0)
```

Se puede asignar un nombre a la restricción para que múltiples columnas tengan la condición

```
ALTER TABLE CUSTOMERS
```

```
ADD CONSTRAINT myCheckConstraint CHECK(AGE >= 18);
```

Si deseamos quitar la restricción Check

```
ALTER TABLE CUSTOMERS
```

```
DROP CONSTRAINT myCheckConstraint;
```



Restricción INDEX

- Permite realizar consultas rápidas
- Se puede crear usando un grupo de columnas
- Cuando se crea el index se le asigna un ROWID por cada fila antes de que los datos se ordenen
- Los índices son adecuados al realizar consultas que tengan una gran cantidad de datos, pero se necesita tener gran cuidado al crearlos

```
CREATE INDEX index_name  
  
ON table_name ( column1, column2.....);
```



Restricción INDEX

```
CREATE TABLE CUSTOMERS (  
    ID      INT              NOT NULL,  
    NAME    VARCHAR (20)     NOT NULL,  
    AGE     INT              NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY  DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```

Se desea crear un Index en la columna EDAD, para optimizar su búsqueda.

```
CREATE INDEX idx_age  
ON CUSTOMERS ( AGE );
```



Restricción INDEX

Se desea quitar el Index

```
ALTER TABLE CUSTOMERS  
DROP INDEX idx_age;
```