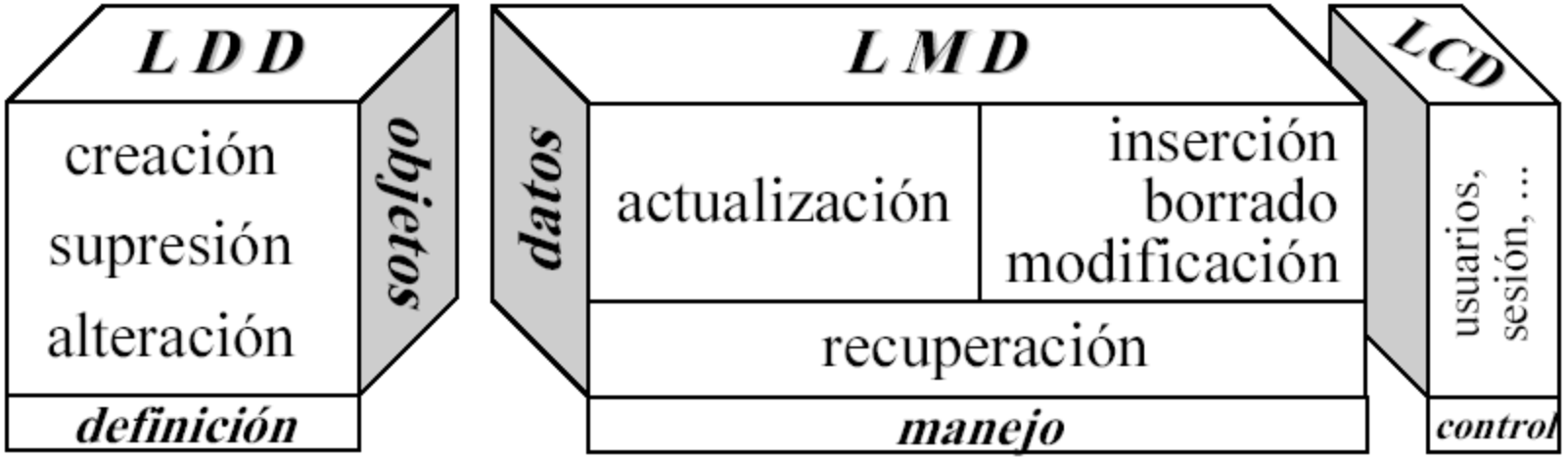
Clase Laboratorio 1 y 2: Introducción Lenguaje SQL

Prof. Ania Cravero

Lenguaje SQL (Structured Query Language)

- Instrucciones del lenguaje SQL: divididas en dos tipos
  - Lenguaje de Definición de Datos, conocido como DDL
  - Lenguaje de Manipulación de Datos, conocido como DML
- Ambos conjuntos son complementarios



Lenguaje SQL

- **Lenguaje de definición de datos (DDL)**
  - Las sentencias DDL se utilizan para crear y modificar la estructura de las tablas así como otros objetos de la base de datos.
  - CREATE - para crear objetos en la base de datos.
  - ALTER - modifica la estructura de la base de datos.
  - DROP - borra objetos de la base de datos.
  - TRUNCATE - elimina todos los registros de la tabla, incluyendo todos los espacios asignados a los registros.
- **Lenguaje de manipulación de datos (DML)**
  - Las sentencias de lenguaje de manipulación de datos (DML) son utilizadas para gestionar datos dentro de los schemas. Algunos ejemplos:
  - SELECT - para obtener datos de una base de datos.
  - INSERT - para insertar datos a una tabla.
  - UPDATE - para modificar datos existentes dentro de una tabla.
  - DELETE - elimina todos los registros de la tabla; no borra los espacios asignados a los registros.

SQL Constraints - Restricciones

- Se utilizan para especificar las reglas para los datos en una tabla.
- Si hay cualquier violación entre la restricción y la acción de datos, la acción se interrumpe por la restricción.

Algunas Restricciones

- NOT NULL - Indica que una columna no puede almacenar valor NULL
- UNIQUE - Asegura que cada fila de una columna debe tener un valor único
- PRIMARY KEY - Una combinación de un NOT NULL y UNIQUE. Asegura que una columna (o combinación de dos o más columnas) tienen una identidad única
- CHECK - Asegura que el valor de una columna cumple una condición específica
- DEFAULT - Especifica un valor por defecto

Lenguaje SQL: DDL

- Creación de Tablas

```
CREATE [TEMPORARY] TABLE <nombre>
      ( <elemento de tabla>
        [ {, <elemento de tabla>} ...]  )

<elemento de tabla> :=
      <def_columna> | <restricción_tabla>

<def_columna> :=
      <nombre> {<tipo_datos>|dominio}
      [SET DEFAULT <valor>]
      [CONSTRAINT <nombre>] <restricción>

<retricción tabla> :=
      [CONSTRAINT <nombre>] <restricción>
```

Ejemplos

Datos no pueden ser Nulos
Son Obligatorios

```
CREATE TABLE Producto
(  ProdNum INTEGER NOT NULL,
   Nombre CHAR(100) NOT NULL,
   Stock Integer NOT NULL,
   PrecioUnit Float NOT NULL,
   PRIMARY  KEY(ProdNum),
   UNIQUE(Nombre),
   CHECK (Stock >= 0)
);
```

Restricciones

Más ejemplos

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
UNIQUE (P_Id)
)
```

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT uc_PersonID UNIQUE (P_Id,LastName)
)
```

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
PRIMARY KEY (P_Id)
)
```

```
CREATE TABLE Persons
(
P_Id int NOT NULL PRIMARY KEY,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)
```

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT pk_PersonID PRIMARY KEY (P_Id,LastName)
)
```

Más ejemplos

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255) DEFAULT 'Sandnes'
)
```

```
CREATE TABLE Orders
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
OrderDate date DEFAULT GETDATE()
)
```

Tipos de datos en Mysql

- Tipos numéricos

- **SmallInt:** número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535.
- **MediumInt:** número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8.388.607. Sin signo el rango va desde 0 a16777215.
- **Integer, Int:** número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295
- **Float:** número pequeño en coma flotante de precisión simple. Los valores válidos van desde -3.402823466E+38 a -1.175494351E-38, 0 y desde 1.175494351E-38 a 3.402823466E+38.

Tipo de Campo	Tamaño de Almacenamiento
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 ú 8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D	M+2 bytes sí D > 0, M+1 bytes sí D = 0
NUMERIC(M,D)	M+2 bytes if D > 0, M+1 bytes if D = 0

Tipos de datos en Mysql

- **Tipos fecha:**

- **Date:** tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-dia
- **DateTime:** Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-dia horas:minutos:segundos
- **TimeStamp:** Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo:.
- **Time:** almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS‘
- **Year:** almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.

Tipo de Campo	Tamaño de Almacenamiento
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

Tipos de datos en Mysql

- **Tipos de cadena:**

- **Char(n):** almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 255 caracteres.
- **VarChar(n):** almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.
- **Blob y Text:** un texto con un máximo de 65535 caracteres.

Tipo de campo	Tamaño de Almacenamiento
CHAR(n)	n bytes
VARCHAR(n)	n +1 bytes
TINYBLOB, TINYTEXT	Longitud+1 bytes
BLOB, TEXT	Longitud +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitud +3 bytes
LONGBLOB, LONGTEXT	Longitud +4 bytes
ENUM('value1','value2',...)	1 ó dos bytes dependiendo del número de valores
SET('value1','value2',...)	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

Lenguaje SQL: DDL

- Supresión de elementos dentro del esquema
- DROP: instrucciones para el borrado de un elemento.

```
DROP <elemento> <nombre>
```

💣 Atención: es peligrosa, ya que no tiene marcha atrás.

• Ejemplos:

```
DROP DOMAIN tipo_turno

DROP TABLE Gente

DROP ASSERTION buenos_clientes

DROP SCHEMA mi_esquema
```

Lenguaje SQL: DDL

- Modificaciones dentro de un esquema
- ALTER: instrucción para la modificación de un elemento.

```
ALTER <elemento> <nombre>
  {ADD|ALTER|DROP} <elemento> [<definición>]
```

Ejemplos:

```
ALTER TABLE Gente
    ADD COLUMN Edad NUMERIC(2,0)

ALTER TABLE Clientes
    ALTER COLUMN Capital NUMERIC(30,2)

ALTER TABLE Clientes
    DROP CONSTRAINT Solvente

ALTER DOMAIN tipo_edad
    ADD CONSTRAINT Matusalen CHECK (VALUE<100)
```

Más ejemplos

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

```
ALTER TABLE Persons
ADD CONSTRAINT uc_PersonID UNIQUE (P_Id,LastName)
```

```
ALTER TABLE Persons
DROP CONSTRAINT uc_PersonID
```

```
ALTER TABLE Persons
ALTER City SET DEFAULT 'SANDNES'
```

```
ALTER TABLE Persons
ALTER City DROP DEFAULT
```

Lenguaje SQL: DDL

- Creación de Dominios

```
CREATE DOMAIN <nombre> [AS] <tipo_datos>
        [DEFAULT <valor por defecto>]
        [<cláusula de restricción de dominio>]

<cláusula de restricción de dominio> :=
[CONSTRAINT <nombre_restricción>]
 CHECK (<condición>)
[INITIALLY {DEFERRED|INMEDIATE} [[NOT] DEFERRED] ]
```

Ejemplos

CREATE DOMAIN TIPO_RUT **AS** VARCHAR(10);

```
CREATE DOMAIN NATURAL AS NUMERIC(40,0)

CREATE DOMAIN tipo_turno AS CHARACTER(1)
      CHECK (VALUE IN ('M', 'T'))

CREATE DOMAIN tipo_edad AS NUMERIC(40,0)
      DEFAULT (18)
      CHECK (VALUE IN (0..100)
      INITIALLY DEFERRED
```