Tecnicatura Universitaria en Desarrollo de Software

### Laboratorio 1

DDL, DQL, DML, DCL y TCL

#### Comandos DDL, DQL, DML, DCL y TCL

• El lenguaje de consulta estructurado (SQL), como todos sabemos, es el lenguaje de la base de datos mediante el cual podemos realizar ciertas operaciones en la base de datos existente y también podemos usar este lenguaje para crear una base de datos. SQL usa ciertos comandos como Create, Drop, Insert, etc. para llevar a cabo las tareas requeridas.

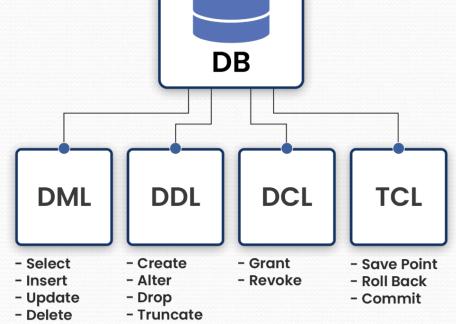
 Estos comandos <u>SQL</u> se clasifican principalmente en cuatro categorías como:

DDL: lenguaje de definición de datos

DQL: lenguaje de consulta de datos

DML: lenguaje de manipulación de datos

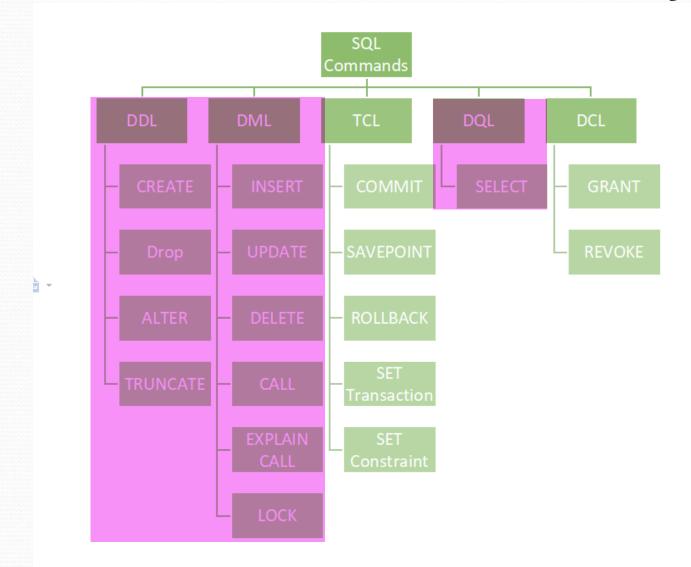
• DCL: lenguaje de control de datos



#### ¿Qué es la integridad referencial?

- La integridad referencial es un sistema compuesto por reglas que permiten la garantía de las relaciones entre las filas de la tabla relacionadas para que sean válidas y que estas relaciones no se eliminen ni se modifiquen por error.
- Las bases de datos son la clave de tener tantas aplicaciones robustas que puedan (en parte) abastecer nuestras necesidades, por ejemplo, en el caso de Facebook o Google.
- Los datos por sí solos no son nada hasta que se contextualizan.
- Para realizar operaciones en una base de datos relacional es importante conocer los siguientes conceptos: DDL, DML, DCL y TCL

#### Comandos DDL, DQL, DML, DCL y TCL



#### Que es el DDL?

- DDL significa Data Definition Language o Lenguaje de Definición de Datos, en español. Este lenguaje permite definir las tareas de las estructuras que almacenarán los datos.
- Comandos SQL que se pueden usar para definir el esquema de la base de datos. Se usa para crear y modificar la estructura
- Estos comandos normalmente no los utiliza un usuario general, que debería acceder a la base de datos a través de una aplicación.
- Sentencias de DDL (Data Definition Language)
- CREATE: Utilizado para crear nuevas tablas, campos e índices.
- ALTER: Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
- DROP: Empleado para eliminar tablas e índices.

## ¿Qué es DML?

- DML significa Data Manipulation Language o Lenguaje de Manipulación de Datos, en español. Este lenguaje permite realizar diferentes acciones a los datos que se encuentran en una base de datos.
- Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una base de datos.
- Elementos del DML (Data Manipulation Language)
- **SELECT:** Utilizado para consultar registros de la **base de datos** que satisfagan un criterio determinado.
- INSERT: Utilizado para cargar de datos en la base de datos en una única operación.
- UPDATE: Utilizado para modificar los valores de los campos y registros especificados

## ¿Qué es DML?

- DML significa Data Manipulation Language o Lenguaje de Manipulación de Datos, en español. Este lenguaje permite realizar diferentes acciones a los datos que se encuentran en una base de datos.
- Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una base de datos.

#### Elementos del DML (Data Manipulation Language)

- SELECT: Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
- INSERT: Utilizado para cargar de datos en la base de datos en una única operación.
- UPDATE: Utilizado para modificar los valores de los campos y registros especificados
- > **DELETE:** Utilizado para eliminar registros de una tabla de una **base de datos**.

## ¿Qué es DQL?

- DQL (lenguaje de consulta de datos):
- Las declaraciones DQL se utilizan para realizar consultas sobre los datos dentro de los objetos del esquema.
- El propósito del comando DQL es obtener alguna relación de esquema basada en la consulta que se le pasó.
- Es un componente de la declaración SQL que permite obtener datos de la base de datos e imponerle un orden.
- Incluye la instrucción SELECT. Este comando permite sacar los datos de la base de datos para realizar operaciones con ella.
- Cuando se dispara un SELECT contra una tabla o tablas, el resultado se compila en una tabla temporal adicional, que se muestra

# ¿Qué es DCL y TCL?

■ Lenguaje de Control de Datos: (Data Control Language)

Permite crear roles, permisos e integridad referencial, así como el control al acceso a la base de datos. Elementos del DCL

- GRANT: Usado para otorgar privilegios de acceso de usuario a la base de datos.
- REVOKE: Utilizado para retirar privilegios de acceso otorgados con el comando GRANT.
- ☐ ¿Qué es TCL? Lenguaje de Control Transaccional
- Permite administrar diferentes transacciones que ocurren dentro de una base de datos.. Elementos del TCL\*\*(Transactional Control Language)\*\*
- COMMIT: Empleado para guardar el trabajo hecho. Confirma la Transacción
- ROLLBACK: Utilizado para deshacer la modificación que hice desde el último COMMIT. Revierte una transacción si ocurriera algún error.
- **SAVEPOINT:** crea puntos dentro del grupo de transacciones en los que hacer un ROLLBACK. Un SAVEPOINT es un punto en una transacción en el que puede revertir la transacción hasta cierto punto sin revertir toda la transacción. Establece un punto de guardado.

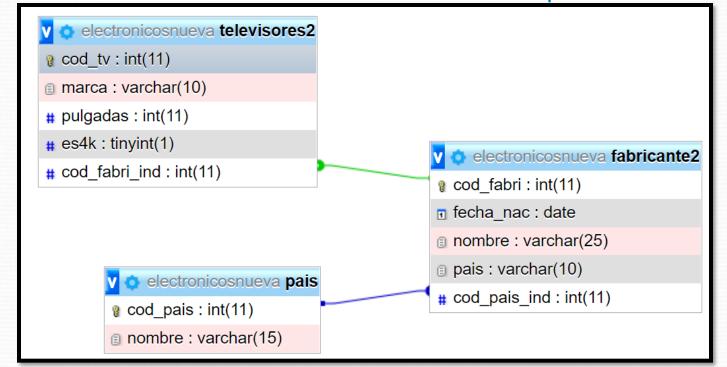
#### ¿Qué son las Transacciones?

- Agrupan un conjunto de tareas en una única unidad de ejecución.
- Cada transacción comienza con una tarea específica y finaliza cuando todas las tareas del grupo se completan con éxito.
- Si alguna de las tareas falla, la transacción falla. Por lo tanto, una transacción tiene solo dos resultados: éxito o fracaso.
- Los pasos incompletos resultan en el fracaso de la transacción. Una transacción de base de datos, por definición, debe ser atómica, consistente, aislada y duradera. Éstas se conocen popularmente como propiedades ACID.
  - COMMIT: Si todo está en orden con todas las declaraciones dentro de una sola transacción, todos los cambios se registran juntos en la base de datos desde el último comando COMMIT o ROLLBACK.
  - ROLLBACK: si ocurre algún error con cualquiera de las declaraciones agrupadas de SQL, todos los cambios deben cancelarse. El proceso de revertir los cambios se denomina reversión. Este comando solo se puede usar para deshacer transacciones desde que se emitió el último comando COMMIT o ROLLBACK.

- Desde la vista diseñador podemos ver, todas las tablas y sus relaciones (líneas de unión) mediante claves.
- La tabla televisores, tiene relación entre su campo indexado (cod\_fabri\_ind) que es llave foránea (clave extranjera proveniente de tabla Fabricante).

La relación se crea al vincular la llave foránea con la llave primaria de

**Fabricante** 



```
CREATE TABLE `pais` (
   `cod_pais` int(11) NOT NULL,
   `nombre` varchar(15) NOT NULL
);

INSERT INTO `pais` (`cod_pais`, `nombre`)
VALUES (10, 'Buzekistan');
```

- CREAMOS la TABLA país,
- INGRESAMOS el registro del país Buzekistan
- MODIFICAR ESTRUCTURA
- Usamos ALTER TABLE con 3 comandos posibles: ADD, CHANGE, y DROP

ALTERAR ESTRUCTURA para agregar una Clave primaria, y otra clave foránea (KEY)

```
ALTER TABLE `fabricante`
ADD PRIMARY KEY (`cod_fabri`),
ADD KEY `cod_pais_ind` (`cod_pais_ind`);
```

```
ALTER TABLE `pais`

ADD PRIMARY KEY (`cod pais`);
```

ALTERAR ESTRUCTURA para MARCAR un CAMPO EXISTENTE como Clave primaria)

Para agregar registros a la tabla, usaremos:

INSERT INTO tabla (columnas entre comas)
VALUES (valores entre comas para cada columna)

```
INSERT INTO `fabricante2`
(`cod_fabri`, `fecha_nac`, `nombre`, `pais`, `cod_pais_ind`)
VALUES
(99, '2022-10-04', 'Abudabi', NULL, 10),
(111, '0000-00-00', 'Abdul Ali', 'Qatar', 10);
```

CREATE TABLE nombre (
 campo\_nombre tipo NOT NULL ,
 campo\_nombre tipo NOT NULL ,
 ...
[PRIMARY KEY (campo index)]

- CREAMOS la TABLA televisores, con su llave primaria cod\_tv
- A continuación INSERTAMOS, tres registros (englobando sus campos entre paréntesis) separados por coma (,)

```
INSERT INTO `televisores2` (`cod_tv`, `marca`, `pulgadas`, `es4k`, `cod_fabri_ind`)
VALUES (22, 'SAMSUN', 48, 1, 99),(33, 'Phinco', 32, 0, 111);
```

- El comando ALTER TABLE permite modificar la estructura de una tabla, agregando, modificando, o quitando columnas (campos) de la misma.
- También es posible utilizarlo para agregar la llave primaria, foránea, o las referencias entre estas claves.

```
Agregamos una relación
                                                                            (restricción) entre tabla
  - Filtros para la tabla `fabricante2`
                                                                            FABRICANTE y PAÍS,
                                                                            usando REFERENCES
ALTER TABLE `fabricante2`
  ADD CONSTRAINT `restriccion1`
  FOREIGN KEY (`cod_pais ind`) REFERENCES `pais`(`cod pais`);
  Filtros para la tabla `televisores2`
                                                   Marcamos en la restricción entre tablas que se deben
                                                   ACTUALIZAR O BORRAR en CASCADA. Para registros que
ALTER TABLE `televisores2`
                                                   mantienen relación, al borrarlo, se borran sus referencias
  ADD CONSTRAINT `restriccion2`
  FOREIGN KEY (`cod fabri ind`) REFERENCES `fabricante2`(`cod fabri`)
  ON DELETE CASCADE ON UPDATE CASCADE;
                       La instrucción COMMIT (TCL) aparecerá al final del script al exportar, e
COMMIT:
                       indica que se confirman los cambios de toda la SERIE de TRANSACCIONES
```

#### Sintaxis alternativa a CREATE TABLE

Otra sintaxis que nos permite crear una table y su clave primaria

CREATE TABLE demo2 (id INTEGER PRIMARY KEY AUTO\_INCREMENT, name VARCHAR(20), hint TEXT);

Podemos marcar la PRIMARY KEY al crear el campo entero id

Podemos marcar la PRIMARY KEY al crear el campo entero id

id: int(11)

iname: varchar(20)

ii hint: text