

BASES DE DATOS

Lenguaje SQL
DDL

Agenda



- ❑ Introducción
- ❑ DML Insert, Update, Delete.
- ❑ Truncate, Merge.
- ❑ DDL Create, Alter, Drop.
- ❑ Ejercicios

Structured Query Language

Codd, IBM, System R, Oracle

DDL: Create, Alter, Drop

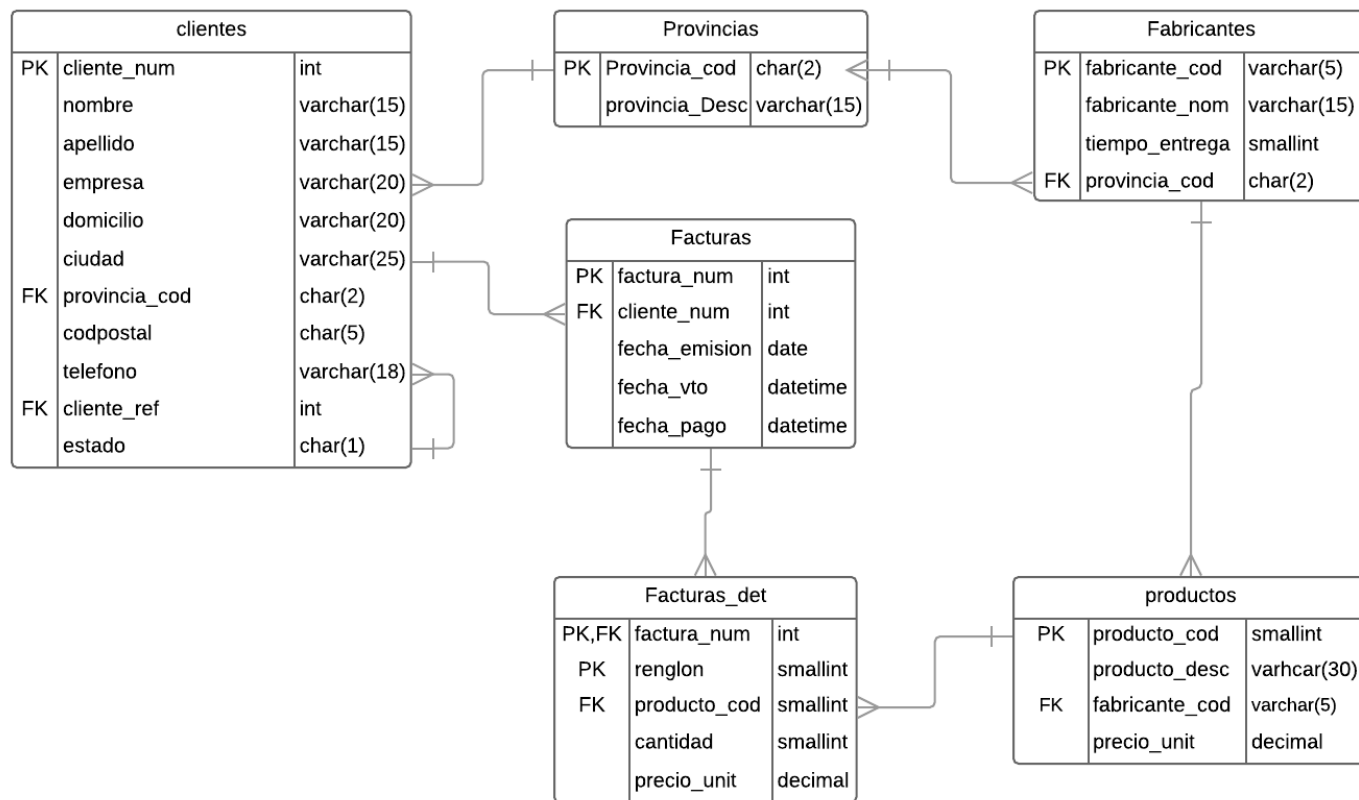
DML: Insert, Update, Delete, Select

DCL: Grant, Revoke

TCL: Begin Transaction, Commit, Rollback



DER



Sentencias DML

“

Estas sentencias se utilizan para modificar (insertar, actualizar, borrar, consultar) filas de las tablas (no objetos de la BD).

Operador INSERT

**INSERT INTO tabla [(lista de columnas)]
VALUES (lista de valores);**

Ej. **INSERT INTO provincias (provincia_cod, provincia_desc)
VALUES ('BA', 'Buenos Aires');**

**INSERT INTO tabla [(lista de columnas)]
VALUES (lista de valores), (lista de valores), (...);**

Ej. **INSERT INTO provincias (provincia_cod, provincia_desc)
VALUES ('JU', 'Jujuy'), ('CH', 'Chubut'), ('ST', 'Salta');**

Operador UPDATE

UPDATE tabla

SET columna=valor[,columna=valor]

[WHERE condición]

Ej. UPDATE clientes

SET empresa = 'UADE',

teléfono = '8888-7777'

WHERE cliente_num = 113;

Operador UPDATE

UPDATE (tabla1 | alias1)

SET alias1.c1 = t2.c2

[, alias1.c2 = expression, ...]

[FROM tabla1 alias1 [INNER | LEFT] JOIN tabla2 t2 ON ...]

[WHERE condición]

UPDATE c

SET c.address2 = 'Direccion 2'

FROM #clientes c JOIN #orders o

ON c.customer_num = o.customer_num

Operador DELETE

**DELETE [FROM] tabla
[WHERE condición]**

**Ej. DELETE FROM cliente
WHERE cliente_num = 114;**

Otro ej. DELETE FROM cliente;

INSERT - Subqueries

**INSERT INTO tabla [(lista de columnas)]
(lista de valores) Subquery**

**Ej. INSERT INTO provincias (provincia_cod, provincia_desc)
(SELECT codigo, desc FROM provincias_fuente);**

UPDATE - Subqueries

UPDATE tabla

SET columna= Subquery

[WHERE ... (=,!=, <=, >=, <, >, IN, Subquery
NOT IN)]

Ej. UPDATE clientes

SET Provincia_cod = (Select provincia_cod
from provincias

Where provincia_desc = 'Chubut')

WHERE cliente_num = 113

UPDATE - Subqueries

“

Ej. UPDATE clientes

SET ciudad = 'Rawson'

WHERE cod_provincia = (Select provincia_cod
from provincias
Where provincia_desc = 'Chubut')

UPDATE - Subqueries

“

UPDATE clientes

SET ciudad = 'Rawson'

WHERE cod_provincia = (Select provincia_cod
from provincias
Where provincia_desc IN ('Chubut', 'Salta'))

Esto funciona ?

DELETE - Subqueries

DELETE [FROM] tabla
[WHERE condición]

Subquery

Ej. DELETE FROM productos

WHERE producto_cod IN

(SELECT producto_cod

FROM productos

WHERE producto_desc LIKE '%borrar%')

Operador TRUNCATE

TRUNCATE TABLE tabla

Ej. TRUNCATE TABLE movimientos

- Borra **TODOS** los registros.
- Es más rápido que el *DELETE*.
- Dependiendo del motor de BD se puede *deshacer*
- **Bloquea** toda la tabla.
- No graba una entrada por cada registro borrado en los logs.
- Libera espacio
- Borra los datos pero **no la estructura**.

Operador CREATE

“

- Crea diferentes tipos de objetos en la BD.
- Objetos: Tablas, índices, vistas, sinónimos, triggers, store procedures, etc.

Operador CREATE

“

Tabla

Tipos de
datos

```
CREATE TABLE Provincias(  
    provincia_cod char(2) NOT NULL PRIMARY KEY,  
    provincia_desc varchar(15) NULL,  
);
```

Atributos o
columnas

Constraints

Operador CREATE

```
CREATE TABLE clientes(cliente_num int PRIMARY KEY,  
                        nombre varchar(15) NOT NULL,  
                        apellido varchar(15) NOT NULL,  
                        empresa varchar(20),  
                        domicilio varchar(20),  
                        provincia char(2) NOT NULL  
                        REFERENCES provincias(provincia_cod),  
                        cliente_ref int NULL,  
                        estado char(1) NULL CHECK(estado IN ('A', 'I')),  
                        FOREIGN KEY (cliente_ref) REFERENCES clientes(cliente_num))
```

Tabla

Atributos o
columnas

Constraints
referencial

Tipos de
datos

Constraints

Constraint
de Check

Operador CREATE

“

```
create table pedidos(  
  id int identity(5, 1) not null PRIMARY KEY,  
  fecha DATE not null,  
  ...  
)
```

Número
secuencial

IDENTITY(*comienzo*, *paso*)

Otras formas de crear una tabla (copia)

-- En Sql Server

*select **

into alumnosb

from alumnos

-- En Oracle

*create table alumnosb AS select * from alumnos*
(Oracle)

Solo copia la estructura. No Pks, ni Cheks, ni índices.

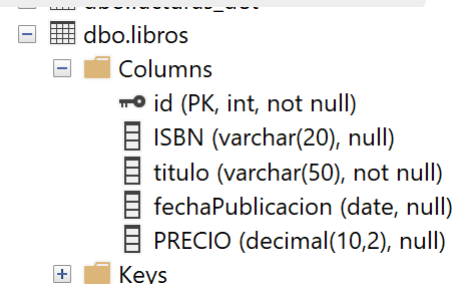
Operador ALTER

- Se utiliza para modificar objetos.
- En el caso de las tablas se puede utilizar para agregar, modificar o borrar columnas y/o constraints.

Operador ALTER

```
CREATE TABLE libros (id          int not null,  
                      ISBN       varchar(20),  
                      titulo      varchar(50),  
                      autor       varchar(50),  
                      fechaPublicacion date);
```

```
ALTER TABLE libros ADD PRIMARY KEY (id);  
ALTER TABLE libros ADD PRECIO DECIMAL(10,2) DEFAULT 0;  
ALTER TABLE libros ALTER COLUMN titulo varchar(50) not null;  
ALTER TABLE libros DROP COLUMN autor;
```



dbo.libros
Columns
id (PK, int, not null)
ISBN (varchar(20), null)
titulo (varchar(50), not null)
fechaPublicacion (date, null)
PRECIO (decimal(10,2), null)
Keys

Operador DROP

“

- Se utiliza para borrar objetos de la BD.
- Una vez borrados no se pueden recuperar.

DROP TABLE Libros;

Sentencia MERGE

Se utiliza esencialmente para realizar procesamiento batch (migraciones, carga de datos, apareos, etc.) de tablas.

```
--  
MERGE  <target_table>  
  USING <table_source>  
  ON <merge_search_condition>  
  [ WHEN MATCHED [ AND <clause_search_condition> ]  
    THEN <merge_matched> ]  
  [ WHEN NOT MATCHED [ BY TARGET ] [ AND <clause_search_condition> ]  
    THEN <merge_not_matched> ]  
  [ WHEN NOT MATCHED BY SOURCE [ AND <clause_search_condition> ]  
    THEN <merge_matched> ]  
  [ <output_clause> ] ;
```


Sentencia MERGE

Pasemos a un ejemplo para entender mejor su funcionamiento y supongamos tener los siguientes requerimientos:

Dada una tabla **Fuente** y una tabla **Destino** cuyas claves primarias en ambas es el atributo **código**:

- Si el **código** de la tabla **Fuente** existe en la tabla **Destino** y las **direcciones** son diferentes entonces actualizar la dirección en la tabla **Destino**
- Si el **código** de la tabla **Fuente** NO existe en la tabla **Destino** entonces insertar en la tabla **Destino** el registro de la tabla **Fuente**.
- Si el **código** de la tabla **Destino** NO existe en la tabla **Fuente** entonces borrar el registro de la tabla **Destino**

Sentencia MERGE

```
create table Fuente
(codigo smallint PRIMARY KEY,
 nombre varchar(30) not null,
 direccion varchar(50)
);
```

```
create table Destino
(codigo smallint PRIMARY KEY,
 nombre varchar(30) not null,
 direccion varchar(50),
 estado char(1) default 'A',
 observaciones varchar(50)
);
```

```
-- insertamos filas en la tabla fuente
insert into Fuente (codigo, nombre,
 direccion) values
(2, 'Ricardo Ruben', 'Paraguay 1888'),
(3, 'Juan Jose Jacinto', 'Terranova 765'),
(8, 'Carola Sampietro', 'Arenales 1265');
```

```
-- insertamos filas en la tabla destino
insert into Destino (codigo, nombre,
 direccion) values
(1, 'Pepe', 'Venezuela 3456'),
(2, 'Ricardo Ruben', 'Cucha Cucha 234'),
(3, 'Juan Jose Jacinto', 'Terranova 765'),
(4, 'Violeta Rivarola', 'Quito 2112');
```

Sentencia MERGE

SQLQuery2.sql - HE...(hernan\herni (52))*

```
select * from mergeFuente
```

100 %

Results Messages

	codigo	nombre	direccion
1	2	Ricardo Ruben	Paraguay 1888
2	3	Juan Jose Jacinto	Terranova 765
3	8	Carola Sampietro	Arenales 1265

SQLQuery2.sql - HE...(hernan\herni (52))* SQLQuery1.sql - HE...(hernan\h

```
select * from mergeDestino
```

100 %

Results Messages

	codigo	nombre	direccion	estado	observaciones
1	1	Pepe	Venezuela 3456	A	NULL
2	2	Ricardo Ruben	Cucha Cucha 234	A	NULL
3	3	Juan Jose Jacinto	Terranova 765	A	NULL
4	4	Violeta Rivarola	Quito 2112	A	NULL

Sentencia MERGE

La sentencia MERGE según los requerimientos:

```
MERGE Destino d
USING Fuente f
ON d.codigo = f.codigo
WHEN MATCHED AND d.direccion <> f.direccion THEN
    UPDATE
        SET d.direccion = f.direccion
WHEN NOT MATCHED BY TARGET THEN
    INSERT (codigo, nombre, direccion, estado, observaciones)
        VALUES (f.codigo, f.nombre, f.direccion, 'A', 'Nuevo')
WHEN NOT MATCHED BY SOURCE THEN
    DELETE;
```

Sentencia MERGE

```
Select * from Destino
```

sults

Messages

codigo	nombre	direccion	estado	observaciones
2	Ricardo Ruben	Paraguay 1888	A	NULL
3	Juan Jose Jacinto	Terranova 765	A	NULL
8	Carola Sanpietro	Arenales 1265	A	Nuevo

Según lo esperado.

Se actualizó la dirección de la fila de la tabla Destino con código = 2;

Se insertó la fila con código = 8 y

Se borraron las filas con códigos 1 y 4.



Preguntas?



Ejercicios