

SQL

DDL

Índice

1. [Definición](#)
2. [Create Table](#)
3. [Alter table](#)
 - 3.1. [Sintaxis](#)
 - 3.2. [Restricciones](#)
4. [Drop](#)
5. [Truncate](#)
6. [Secuencias](#)
 - 6.1. [Sintaxis](#)
7. [Index](#)

Definición

Son sentencias que se usan para la definición de objetos de la base de datos. Entre estas sentencias están las de **creación (create)**, **modificación (alter, replace)** y **borrado (drop)** de objetos de la base de datos y las usadas para proporcionar integridad y consistencia a la base de datos (**constraints** y **checks**).

Todas estas instrucciones actualizan y mantienen lo que se denomina el diccionario de datos (tablas que contienen información sobre los objetos de la base de datos).

Create Table

```
CREATE TABLE nombre_tabla(
    COLUMNA TIPO
        [NOT NULL]
        [UNIQUE]
        [PRIMARY KEY]
        [DEFAULT valor]
        [REFERENCES Nomtabla][columna[,columna]][ON DELETE
            CASCADE]]
    [CHECK condicion],
    COLUMNA TIPO [NOT NULL],
    . . .
{CONSTRAINT nombre_clave_primaria PRIMARY KEY (columnas_clave)}
{CONSTRAINT nombre_clave_foránea
    FOREIGN KEY(columnas_clave) REFERENCES tabla_detalle(
        columnas_clave )
    {ON DELETE CASCADE} }
)
{TABLESPACE tablespace_de_creación}
{STORAGE( INITIAL XX{K|M} NEXT XX{K|M} )}
```

Ejemplo

```
CREATE TABLE FACTURA(  
    REFERENCIA VARCHAR2(10) NOT NULL,  
    DESCRIPCION VARCHAR2(50),  
    C_PAIS NUMBER(3),  
    C_CLIENTE NUMBER(5),  
    IMPORTE NUMBER(12),  
    CONSTRAINT PK_FACTURA PRIMARY KEY( REFERENCIA ),  
    CONSTRAINT FK_CLIENTE FOREIGN KEY(C_PAIS,C_CLIENTE) REFERENCES  
    CLIENTE( C_PAIS, C_CLIENTE)  
    ON DELETE CASCADE  
)  
TABLESPACE tab_facturas  
STORAGE( INITIAL 1M NEXT 500K );
```

Alter table

Sirve para modificar cualquier objeto existente en la base de datos.

Los objeto que más se modifican son las tablas, los índices y los objetos **PL/SQL**.

Sintaxis

```
ALTER TABLE nombretabla  
– [ADD (columna[,columna]...)  
– [MODIFY (columna[,columna]...)  
– [DROP COLUMN (columna[, columna]...)  
– [ADD CONSTRAINT restriccion]  
– [ADD CONSTRAINT restriccion];
```

Ejemplos

--añade dos columnas a una tabla

```
ALTER TABLE KTMP_VALORES  
ADD (ANYO_CONTABLE_VALO number(4),  
REF_CONTABLE_VALO number(20))
```

--modifica la definición de columnas de una tabla

```
ALTER TABLE reov_registro_op_valor  
MODIFY (fecha_operacion_reov NOT NULL,  
id_fiin number(12))
```

--añade una restricción a una tabla

ALTER TABLE ktmp_temporal

ADD CONSTRAINT FK_KTMP_EPER_EPER

FOREIGN KEY(id_eper)

REFERENCES

EPER_ELEMENTOS_PERSONA(id_eper)

Restricciones

La integridad referencial es una herramienta imprescindible de las bases de datos relacionales. Pero provoca varios problemas.

Por ejemplo, si borramos un registro en la tabla principal que está relacionado con uno o varios de la secundaria ocurrirá un error, ya que de permitírse nos borrar el registro ocurrirá fallo de integridad (habrá claves secundarios refiriéndose a una clave principal que ya no existe).

Por ello Oracle nos ofrece dos soluciones a añadir tras la cláusula **REFERENCES**:

ON DELETE SET NULL

Coloca nulos todas las claves secundarias relacionadas con la borrada.

ON DELETE CASCADE

Borra todos los registros cuya clave secundaria es igual que la clave del registro borrado.

Existe también la opción de **ON UPDATE CASCADE** o **ON UPDATE SET NULL**, pero **ORACLE** no las implementa.

Drop

Sirve para **ELIMINAR** objetos de la base de datos.

Los objetos más comúnmente eliminados son índices y tablas temporales.

Ejemplos

--elimina un índice

```
DROP INDEX IX_VALO_REF_EXTERNA
```

--elimina una tabla

```
DROP TABLE ktmp_temporal
```

--elimina un usuario o esquema

```
DROP USER ktmp_temporal
```

Junto con **DROP** se puede especificar la cláusula **CASCADE CONSTRAINTS**, lo haría que **eliminará todos los objetos que hicieran referencia al objeto eliminado**, convirtiendo esta instrucción en algo muy peligroso.

Truncate

Esta instrucción es un híbrido entre instrucciones **DDL** y **DML**. Sólo se ejecuta sobre tablas y no modifica la definición de las mismas en cuanto a estructura, pero elimina todos los datos de la tabla sobre la que se ejecuta. **TRUNCATE** emite automáticamente una instrucción **COMMIT**, y libera el espacio ocupado por la tabla en el sistema de ficheros reduciéndolo al inicial.

```
TRUNCATE TABLE TMP_FRACCIONES
```

Secuencias

Una secuencia (**sequence**) es un objeto de base de datos que genera números secuenciales.

Sintaxis

```
CREATE SEQUENCE nombre_secuencia  
{START WITH entero}  
{INCREMENT BY entero}  
{MAXVALUE entero | NOMAXVALUE}  
{MINVALUE entero | NOMINVALUE }  
{CYCLE | NOCYCLE};
```

Index

CREATE {UNIQUE} INDEX

nombre_índice

– **ON** tabla(columnas_indexadas)

– {**TABLESPACE** tab_indices}

– {**STORAGE**(**INITIAL** XX{K|M} **NEXT** XX{K|M}
)} }