

Introducción a SQL

Laboratorio de Base de datos

Cristian E. Sánchez

Agenda

- Drop Table
- SQL como lenguaje de manipulación de datos
 - Eliminar registros
 - Obtener datos
 - Insertar registros
 - Modificar registros

Eliminar una tabla – Instrucción DROP TABLE

DROP TABLE schema_name.table_name
[CASCADE CONSTRAINTS | PURGE];

- **schema**: Si no se especifica el esquema, se asume que se elimina una tabla del esquema propio.
- **cascade constraints**: Elimina todas las restricciones de integridad referencial. Si existe una referencia de integridad y no se usa esta cláusula, entonces Oracle genera un error y no elimina la tabla.
- **purge**: Se puede usar para liberar el espacio asociado al objeto eliminado (tabla). En este caso, Oracle no deja la tabla en el Recycle bin.

Tarea



DROP TABLE – Ejemplos

```
CREATE TABLE estudiantes (  
    estudiante_id NUMBER,  
    nombre VARCHAR2(50) NOT NULL,  
    apellido VARCHAR2(50) NOT NULL,  
    PRIMARY KEY(estudiante_id)  
);
```

```
DROP TABLE estudiantes;
```

DROP TABLE CASCADE CONSTRAINTS – Ejemplos

```
CREATE TABLE marcas(  
    marca_id NUMBER PRIMARY KEY,  
    nombre_marca varchar2(50)  
);
```

```
CREATE TABLE autos(  
    auto_id NUMBER PRIMARY KEY,  
    auto_modelo VARCHAR(50) NOT NULL,  
    anio_modelo NUMBER NOT NULL,  
    placa VARCHAR(25),  
    marca_id NUMBER NOT NULL,  
  
    CONSTRAINT fk_marca  
    FOREIGN KEY (marca_id)  
    REFERENCES marcas(marca_id) ON DELETE CASCADE  
);
```

```
DROP TABLE marcas;
```

Para eliminar la tabla MARCAS se debe usar la cláusula *CASCADE CONSTRAINTS*

```
DROP TABLE marcas CASCADE CONSTRAINTS;
```

```
DROP TABLE marcas CASCADE CONSTRAINTS;
```

Esta instrucción no solo elimina la tabla MARCAS sino también la llave foránea que existía en la tabla AUTOS

Tarea de la clase anterior:
¿En qué consisten las
expresiones Numeric y
Date?

SQL – Expresiones numéricas

- Son usadas para llevar a cabo operaciones matemáticas en consultas SQL.

```
SELECT NUMERICAL_EXPRESSION as OPERATION_NAME  
FROM table_name
```

EmpId	EmpName	EmpAge	EmpSalary
1	John	32	2000
2	Smith	25	2500
3	Henry	29	3000

```
SELECT count(*) FROM Employee; -- 3
```

SQL – Expresiones numéricas

- Son usadas para llevar a cabo operaciones matemáticas en consultas SQL.

```
SELECT NUMERICAL_EXPRESSION as OPERATION_NAME  
FROM table_name
```

EmpId	EmpName	EmpAge	EmpSalary
1	John	32	2000
2	Smith	25	2500
3	Henry	29	3000

```
select min(EmpSalary) from Employee; -- 2000  
select max(EmpSalary) from Employee; -- 3000  
select sum(EmpSalary) from Employee; -- 7500  
select avg(EmpSalary) from Employee; -- 2500.0000
```

SQL – Date Expression

- Son usadas para generar valores datetime

SELECT CURRENT_TIMESTAMP; -- 2018-01-20 10:32:37

-- MYSQL

SELECT now(); -- 2018-01-20 10:32:57

-- SQL Server

SELECT GetDate();

-- Oracle DB

select sysdate from Dual; -- 20-JAN-21

Oracle – DELETE

- Esta sentencia permite eliminar uno o más registros de una tabla

```
DELETE  
FROM  
    table_name  
WHERE  
    condition;
```

```
DELETE  
FROM autos  
WHERE anio_modelo = 2017;
```

1 row deleted.

```
DELETE  
FROM autos  
WHERE anio_modelo > 2015  
AND marca_id = 2;
```

2 rows deleted.

```
DELETE  
FROM autos;
```

5 rows deleted.

Oracle – DELETE CASCADE

- En la práctica, generalmente es necesario borrar registros de una tabla que tienen relaciones de llave foránea con registros de otras tablas.
- En nuestro ejemplo, podríamos eliminar registros de la tabla Marcas con marca_id 1 y también borrar todos los autos asociados a esa marca de la tabla Autos.
- Esto puede pensarse de dos formas: La primera es usar dos instrucciones **DELETE** por cada tabla. La segunda es SABER que se ajustaron correctamente las restricciones de la tabla

Oracle – DELETE CASCADE

- OPCIÓN 1

```
DELETE  
FROM marcas  
WHERE marca_id = 1;
```

```
DELETE  
FROM autos  
WHERE marca_id = 1;
```

```
COMMIT WORK;
```

COMMIT WORK asegura que ambas instrucciones se ejecuten completas o ninguna. Esto evita que en la tabla Autos queden registros huérfanos en caso que la segunda instrucción falle.

- OPCIÓN 2

```
CREATE TABLE autos(  
    auto_id NUMBER PRIMARY KEY,  
    -- otras columnas
```

```
    CONSTRAINT fk_marca  
    FOREIGN KEY (marca_id)  
    REFERENCES marcas(marca_id)
```

```
ON DELETE CASCADE  
);
```

```
DELETE  
FROM marcas  
WHERE marca_id = 1;
```

ORACLE – INSERT

- Para insertar un registro en una tabla

```
INSERT INTO table_name (column_list)  
VALUES( value_list);
```

- Si la lista de valores a insertar tiene el mismo orden que las columnas de la tabla

```
INSERT INTO table_name  
VALUES( value_list);
```

```
Insert into AUTOS (AUTO_ID, AUTO_MODELO, ANIO_MODELO, PLACA, MARCA_ID)  
values (1, 'A1', 2017, '007', 1);
```

ORACLE – SELECT

- Permite obtener la información de una tabla

```
SELECT  
column_1,  
column_2,  
...  
FROM  
table_name;
```

```
Select anio_modelo, placa  
From Autos;
```


ORACLE – UPDATE

- Permite modificar los registros de una tabla

UPDATE

table_name

SET

column1 = value1,

column2 = value2,

column3 = value3,

...

WHERE

condition;

Update autos

Set anio_modelo = 2018

Where auto_id = 2;

SQL como lenguaje de manipulación de datos

Observaciones importantes

- La sentencia UPDATE sin cláusula WHERE, actualiza todos los registros de la tabla.
- La sentencia DELETE sin cláusula WHERE, borra todos los registros de la tabla.
- La sentencia TRUNCATE elimina todos los registros de una tabla. No se puede deshacer con rollback. No se disparan triggers.

SQL como lenguaje de manipulación de datos

Transacciones

- Simboliza una unidad de trabajo realizada dentro del sistema de administración de base de datos contra una BD en particular. Puede estar compuesta por varias operaciones (e.g, sentencias SQL)
- Su objetivo es mantener la base de datos en un estado consistente

Ejemplo: Transferencia de dinero de una cuenta bancaria a otra

- La transacción completa requiere extraer la cantidad de dinero a transferir desde una cuenta y adicionar la misma cantidad de dinero en la otra cuenta.

Transacciones – Propiedades ACID

- **Atomicity** (Atomicidad): Es la propiedad que asegura que la operación se realice o no se realice, y que no quede a medias.
- **Consistency** (Consistencia): Una transacción nunca debe dejar datos inconsistentes. Debe llevar la base de datos de un estado válido a otro también válido.
- **Aislamiento** (Isolation): Cada transacción es independiente. Si dos o más transacciones actúan concurrentemente sobre los mismos datos, no se debe generar error y se debe garantizar que los cambios sean visibles para las demás transacciones.
- **Durabilidad** (Durability): Asegura la persistencia de la información, es decir, una vez que termine la transacción, los cambios permanecerán y no se podrán deshacer aunque falle el sistema.

Transacciones – Oracle

Las sentencias básicas para control de las transacciones en Oracle son:

- **SAVEPOINT:** Marca un punto dentro de la transacción hasta el cual se puede deshacer (rollback).
- **COMMIT:** Finaliza una transacción, hace los cambios permanentes, borra los SAVEPOINT y libera los bloqueos para que otras transacciones puedan operar.
- **ROLLBACK :** Deshace los cambios realizados por toda la transacción o hasta el último SAVEPOINT especificado.

En SQL Developer, están disponibles los íconos para realizar COMMIT o ROLLBACK.

Transacciones – Oracle

```
UPDATE employees
  SET salary = 7000
  WHERE last_name = 'Banda';
SAVEPOINT banda_sal;

UPDATE employees
  SET salary = 12000
  WHERE last_name = 'Greene';
SAVEPOINT greene_sal;

SELECT SUM(salary) FROM employees;

ROLLBACK TO SAVEPOINT banda_sal;

UPDATE employees
  SET salary = 11000
  WHERE last_name = 'Greene';

COMMIT;
```