### CONSULTAS INSERT ALL

Vamos a visualizar otro tipo de consultas de acción únicas de oracle

En Ansi SQL no existen consultas masivas de datos a no ser que

los recuperemos de otra tabla.

ANSI SQL: Un insert para cada registro.

5 inserciones = 5 insert

**Bulk insert**

Inserta mas rápido que el resto de consultas, porque el motor de base de datos sabe que tiene que ejecutar un conjunto.

1) Insertar varios registros en una tabla que exista

SINTAXIS:

     INSERT ALL

              INTO TABLA VALUES (VALOR1, VALOR2)

              INTO TABLA VALUES (VALOR1, VALOR2)

              INTO TABLA (CAMPO1, CAMPO2) VALUES (VALOR1, VALOR2)

     SELECT \* FROM DUAL;

Vamos a insertar dos departamentos en dept

INSERT ALL

       INTO DEPT VALUES (50, 'INFORMATICA', 'GETAFE')

       INTO DEPT VALUES (60, 'I+D', 'ALICANTE')

SELECT \* FROM DUAL;

SELECT \* FROM DEPT;

ROLLBACK;

Esta consulta es más rápida al insertar

Vamos a comparar dos consultas y cómo se comportan

**INSERT INTO CONTRA INSERT ALL**

**VAMOS A REALIZAR UN AUTOINCREMENTO. SUBCONSULTA CON MAX.**

INSERT INTO

INSERT INTO DEPT VALUES

((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '@@@@', '@@@@');

INSERT INTO DEPT VALUES

((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '????', '????');

INSERT INTO DEPT VALUES

((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '####', '####');

-----------------------------

**INSERT ALL**

INSERT ALL

       INTO DEPT VALUES ((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '@@@@', '@@@@')

       INTO DEPT VALUES ((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '####', '####')

       INTO DEPT VALUES ((SELECT MAX(DEPT\_NO) + 1 FROM DEPT), '????', '????')

SELECT \* FROM DUAL;

**CREACION DE TABLAS PARA INSERCIONES**

**CREATE..SELECT**

Esta instrucción lo que realiza es crear tablas a partir de un select

Solamente podemos utilizarla una vez.  Si la tabla ya está creada, devuelve error.

También inserta registros en la tabla si el select devuelve registros

Create permite crear objetos permanentes dentro de la bbdd.

Se utiliza para la creación de objetos que iremos viendo en clases posteriores

Ahora mismo lo que estamos viendo para una herramienta de inserción

**CREATE...SELECT** no solo crea, sino que además incluye los mismos tipos de dato que el origen.

**ROLLBACK NI COMMIT FUNCIONAN.  TODO QUEDA ALMACENADO.**

SINTAXIS:

     CREATE TABLE DESTINO

     AS

     SELECT \* FROM ORIGEN;

Hacer una copia de seguridad de la tabla departamentos.

CREATE TABLE DEPARTAMENTOS

AS

SELECT \* FROM DEPT;

-------------------

SELECT \* FROM DEPARTAMENTOS;

------------------

INSTRUCCION INSERT INTO SELECT

NOS PERMITE COPIAR DATOS DE UNA TABLA ORIGEN A UNA TABLA DESTINO

LA DIFERENCIA CON CREATE...SELECT ESTÁ EN QUE LA TABLA DEBE EXISTIR

SIN TABLA DE DESTINO NO PODEMOS EJECUTAR ESTA INSTRUCCION

LA TABLA DE DESTINO DEBE DE TENER LA MISMA ESTRUCTURA QUE LOS DATOS DEL SELECT DE ORIGEN

SINTAXIS:

INSERT INTO DESTINO

SELECT \* FROM ORIGEN;