DuocUC®

INFORMÁTICA Y TELECOMUNICACIONES

Manipulando Datos Almacenados en las Tablas

BDY1102: Noviembre, 2023

Objetivo de la clase

- Describir las características del lenguaje DML y Transacciones de Base de Datos.
- Cómo construir sentencias DML para Insertar Filas a las Tablas.
- Cómo construir sentencias DML para Actualizar Filas de las Tablas.
- Cómo construir sentencias DML para Eliminar Filas de las Tablas.
- Uso de sentencia TRUNCATE para eliminar filas de las Tablas.
- Uso de Subconsultas en sentencias DML.

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Conceptos del Lenguaje de Manipulación de Datos

BDY1102: Noviembre, 2023

Lenguaje de Manipulación de Datos (DML)

Está compuesto de comandos SQL que permiten modificar filas de las tablas de la de la base de datos

Todas las modificaciones de los datos se efectúan en la memoria de la Base de Datos

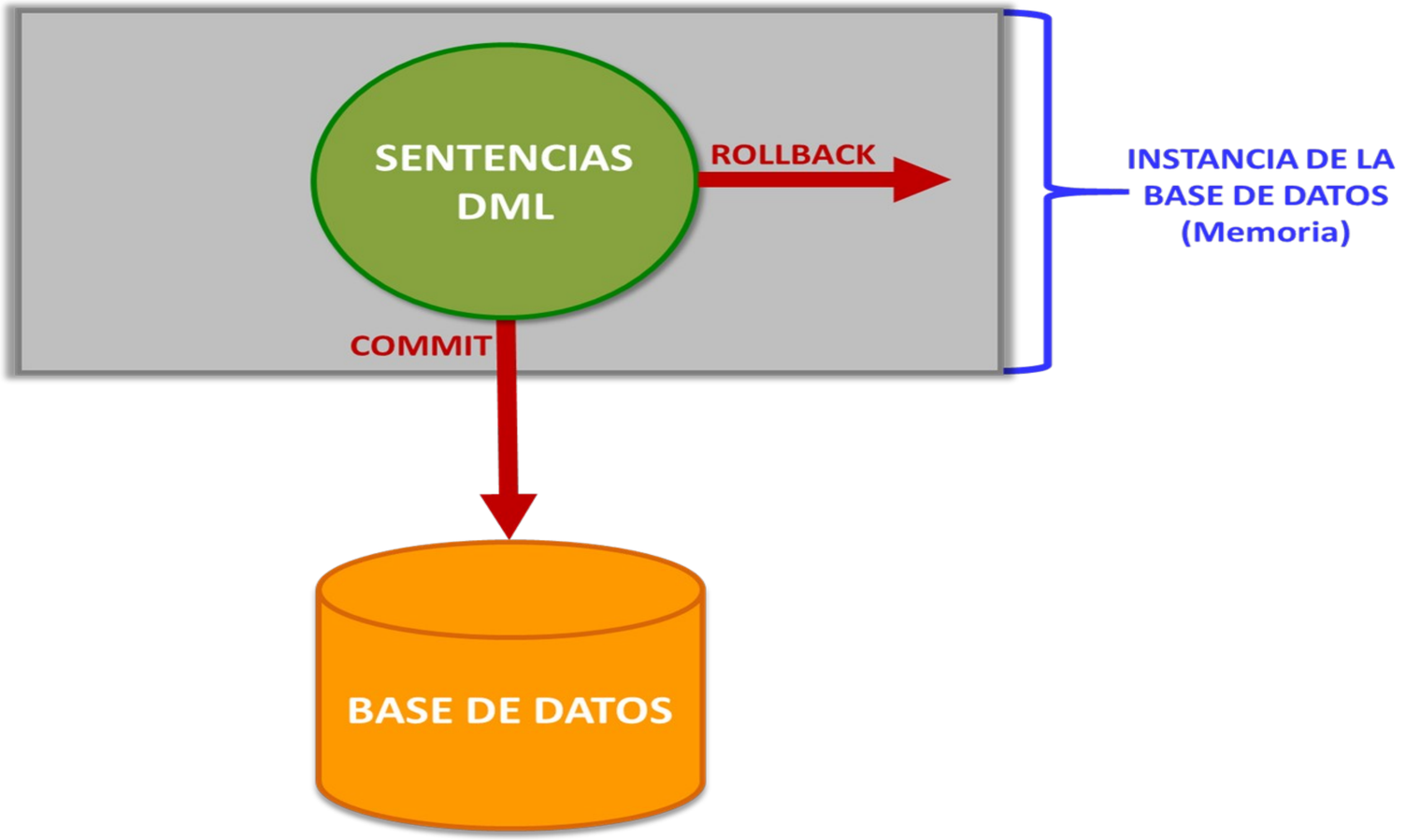
Mientras los cambios no sean confirmados, solo son visibles para el usuario que los efectuó

Para que las modificaciones se realicen físicamente en las tablas, se deben confirmar los cambios.

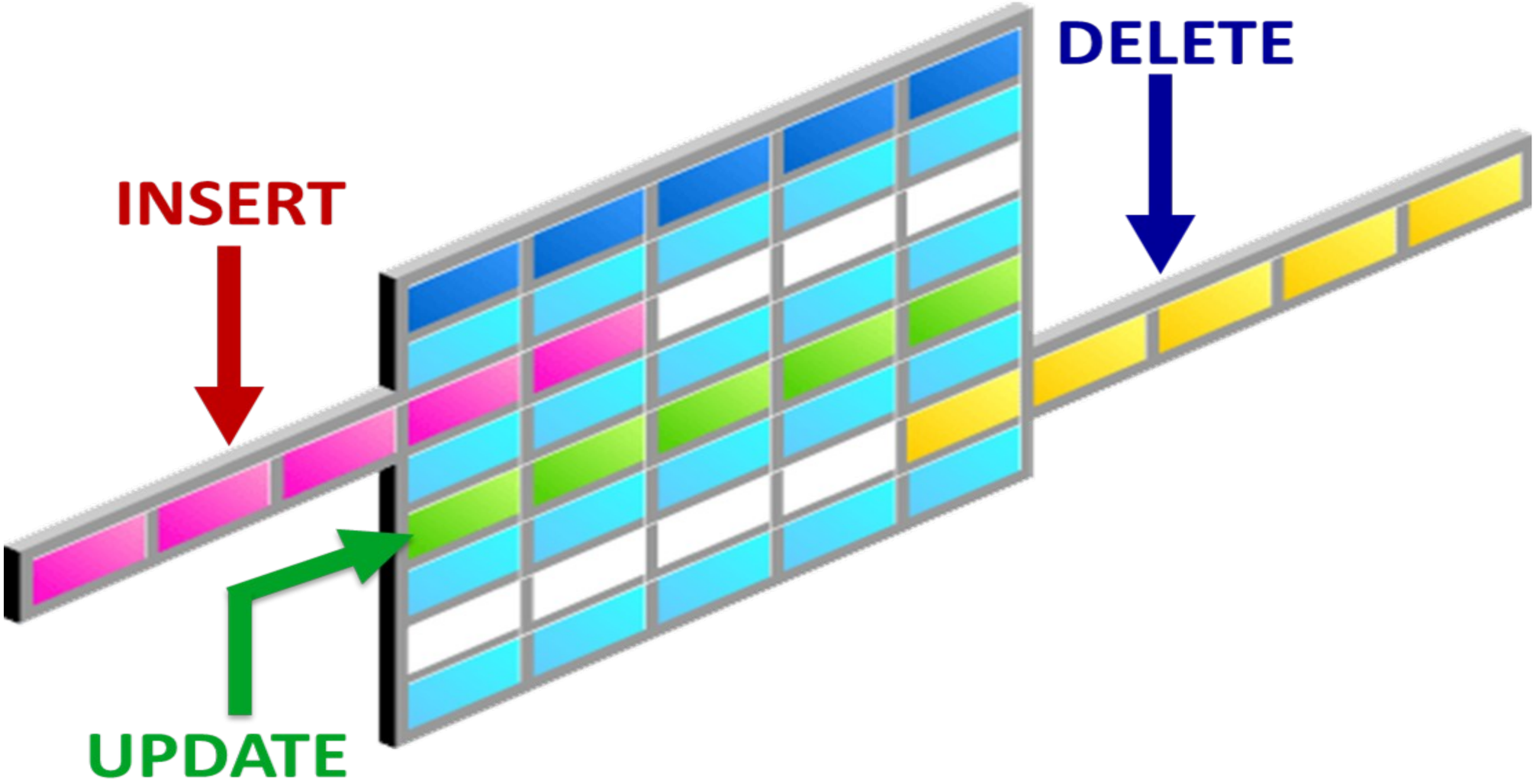
Para confirmar los cambios se debe usar el comando COMMIT

Para que no realicen los cambios en las tablas se debe usar el comando ROLLBACK

Sentencias COMMIT y ROLLBACK

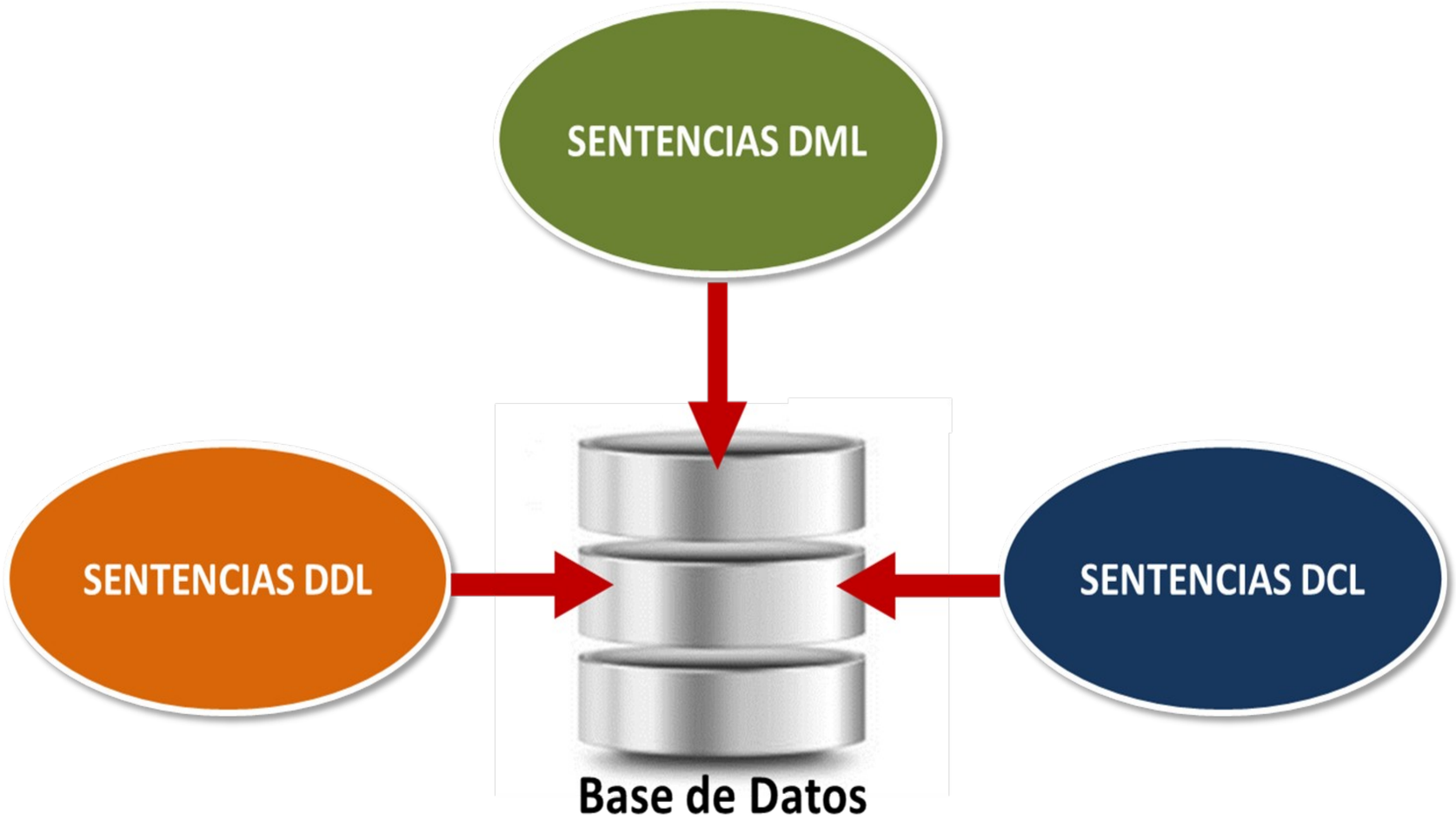


Sentencias DML



DuocUC®

Transacciones de Base de Datos



DuocUC®

Usando SAVEPOINT para marcar Transacciones

Un Savepoint es una marca que permite efectuar un Rollback parcial en la Transacción

- Ejemplo:

```
DELETE FROM employees
WHERE department_id IS NULL;
SAVEPOINT A;
UPDATE employees
SET salary = salary * 1.50;
SAVEPOINT B;
DELETE FROM employees
WHERE manager_id IS NULL;
ROLLBACK TO B;
COMMIT;
```

Usando SAVEPOINT para marcar Transacciones

- Ejemplo:

```
DELETE FROM employees
WHERE department_id IS NULL;
SAVEPOINT A;
UPDATE employees
SET salary = salary * 1.50;
SAVEPOINT B;
DELETE FROM employees
WHERE manager_id IS NULL;
TRUNCATE TABLE job_history;
ROLLBACK TO B;
COMMIT;
```

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Incorporando nuevas Filas en las Tablas

BDY1102: Noviembre, 2023

Agregando Filas a una Tabla

TABLA DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	80	Sales	145	2500
...	...	...	...	...
22	230	IT Helpdesk	(null)	1700
23	240	Government Sales	(null)	1700
24	250	Retail Sales	(null)	1700
25	260	Recruiting	(null)	1700
26	270	Payroll	(null)	1700

Nueva Fila

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
70	Public Relations	204	2700

TABLA DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	70	Public Relations	204	2700
8	80	Sales	145	2500
...	...	...	...	...
23	230	IT Helpdesk	(null)	1700
24	240	Government Sales	(null)	1700
25	250	Retail Sales	(null)	1700
26	260	Recruiting	(null)	1700
27	270	Payroll	(null)	1700

DuocUC®

Agregando Filas a una Tabla

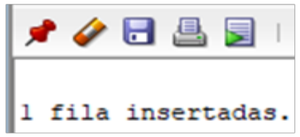
- Sintaxis: se inserta una fila a la vez.

```
INSERT INTO tabla [(columna [, columna...])]
VALUES (valor [, valor...]);
```

- Ejemplo:

```
INSERT INTO departments(department_id, department_name,
                        location_id, manager_id)
VALUES (70, 'Public Relations', 1700, 100);
```

```
INSERT INTO departments
VALUES (70, 'Public Relations', 100, 1700);
```



DuocUC®

Agregando Filas con Valores Nulos

- Ejemplo:

Implícitamente

```
INSERT INTO departments (department_id,  department_name)
VALUES (300, 'Purchasing');
```

- Ejemplo:

Explícitamente

```
INSERT INTO departments
VALUES (301, 'Finance', NULL, NULL);
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
301	Finance	(null)	(null)

DuocUC®

Agregando Filas con Valores Especiales

Agregando Filas con Valores Especiales

- Ejemplo:

```
INSERT INTO employees (employee_id, first_name, last_name, email, phone_number,
            hire_date,  job_id, salary, commission_pct, manager_id, department_id)
VALUES(300, 'Louis', 'Hamilton', 'LHAMILTON',  '515.124.4567', SYSDATE, 'AC_ACCOUNT',
          6900,  NULL, 205, 100);
COMMIT;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
300	Louis	Hamilton	LHAMILTON	515.124.4567	23/08/2018	AC_ACCOUNT	6900	(null)	205	100

- Ejemplo:

```
INSERT INTO employees (employee_id, first_name, last_name, email, phone_number,
            hire_date,  job_id, salary, commission_pct, manager_id, department_id)
VALUES(301, 'Michael', 'Santana', USER,  '515.124.4567', SYSDATE, 'AC_ACCOUNT',
          6900, NULL, 205, 100);
ROLLBACK;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
301	Michael	Santana	HR	515.124.4567	23/08/2018	AC_ACCOUNT	6900	(null)	205	100

DuocUC®

Agregando Filas con Valor obtenido de Subconsulta

- Ejemplo:

```
INSERT INTO employees (employee_id, first_name, last_name, email,
            phone_number, hire_date, job_id, salary, commission_pct, manager_id,
            department_id)
VALUES  (303, 'Juan', 'Soto', 'JSOTO' '  '515.124.4567', SYSDATE, 'AC_ACCOUNT',
            6900, NULL, 205,  (SELECT department_id FROM employees WHERE employee_id =100));
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
303	Juan	Soto	JSOTO	515.124.4567	23/08/2018	AC_ACCOUNT	6900	(null)	205	90

- Ejemplo:

```
INSERT INTO employees (employee_id, first_name, last_name, email,
            phone_number, hire_date, job_id, salary, commission_pct, manager_id,
            department_id)
VALUES  (304, 'Rosa', 'Garrido', 'RGARRIDO',  '245.412.4567', SYSDATE, 'AC_ACCOUNT',
            (SELECT MAX(salary) FROM employees), NULL, 205,  (SELECT department_id
                                                              FROM employees
                                                              WHERE employee_id =100));
```

DuocUC®

Agregando Múltiples Filas en diferentes Tablas

- Inserción de múltiples filas en múltiples tablas:

```
INSERT ALL
   INTO tabla_1(col1,col2,col3, ...) VALUES(valor1,valor2, valor3, ...)
   INTO tabla_2(col1,col2,col3, ...) VALUES(valor4,valor5, valor6, ...)
   INTO tabla_3(col1,col2,col3, ...) VALUES(valor7,valor8, valor9, ...)
Subconsulta;
```

- Inserción Condicional de múltiples filas en múltiples tablas:

```
INSERT ALL
   WHEN condición1 THEN
      INTO tabla_1(col1,col2,col3, ...) VALUES(valor1,valor2, valor3, ...)
   WHEN condición2 THEN
      INTO tabla_2(col1,col2,col3, ...) VALUES(valor4,valor5, valor6, ...)
   ELSE
      INTO tabla_3(col1,col2,col3, ...) VALUES(valor7,valor8, valor9, ...)
Subquery
```

Agregando Múltiples Filas en diferentes Tablas

- Ejemplo:

1

```
CREATE TABLE  TABLA_PRUEBA1
(ID NUMBER(10) GENERATED  ALWAYS AS IDENTITY MINVALUE 1
MAXVALUE 9999999999999999999999999999
INCREMENT BY 1 START WITH 1,
NOMBRE_DEPTO VARCHAR2(30) NOT NULL,
TOTAL_EMPLEADOS NUMBER(4));

CREATE TABLE  TABLA_PRUEBA2
(ID NUMBER(10) GENERATED  ALWAYS AS IDENTITY MINVALUE 10
MAXVALUE 9999999999999999999999999999
INCREMENT BY 10 START WITH 10,
NOMBRE_DEPTO VARCHAR2(30) NOT NULL,
TOTAL_EMPLEADOS NUMBER(4));
```

Agregando Múltiples Filas en diferentes Tablas

2

```
INSERT ALL
  INTO TABLA_PRUEBA1(NOMBRE_DEPTO,TOTAL_EMPLEADOS)
              VALUES(department_name,total_emp)
  INTO TABLA_PRUEBA2(NOMBRE_DEPTO,TOTAL_EMPLEADOS)
              VALUES(department_name,total_emp)
SELECT d.department_name, COUNT(e.employee_id) total_emp
FROM departments d LEFT OUTER JOIN employees e
ON d.department_id=e.department_id
GROUP BY d.department_name
ORDER BY d.department_name;
```

TABLA_PRUEBA1

	ID	NOMBRE_DEPTO	TOTAL_EMPLEADOS
1	1	Accounting	2
2	2	Administration	1
3	3	Benefits	0
4	4	Construction	0
5	5	Contracting	0
6	6	Control And Credit	0
7	7	Corporate Tax	0
8	8	Executive	3
9	9	Finance	6
10	10	Government Sales	0
24	24	Sales	34
25	25	Shareholder Services	0
26	26	Shipping	45
27	27	Treasury	0

TABLA_PRUEBA2

	ID	NOMBRE_DEPTO	TOTAL_EMPLEADOS
1	10	Accounting	2
2	20	Administration	1
3	30	Benefits	0
4	40	Construction	0
5	50	Contracting	0
6	60	Control And Credit	0
7	70	Corporate Tax	0
8	80	Executive	3
9	90	Finance	6
10	100	Government Sales	0
24	240	Sales	34
25	250	Shareholder Services	0
26	260	Shipping	45
27	270	Treasury	0

DuocUC®

Agregando Múltiples Filas en diferentes Tablas

- Ejemplo:

1

```
CREATE TABLE  DEPTOS_SIN_EMPLEADOS
(ID NUMBER(10) GENERATED  ALWAYS AS IDENTITY MINVALUE 1
MAXVALUE 9999999999999999999999999999
INCREMENT BY 1 START WITH 1,
NOMBRE_DEPTO VARCHAR2(30) NOT NULL);

CREATE TABLE DEPTOS_CON_EMPLEADOS
(ID NUMBER(10) GENERATED  ALWAYS AS IDENTITY MINVALUE 10
MAXVALUE 9999999999999999999999999999
INCREMENT BY 10 START WITH 10,
NOMBRE_DEPTO VARCHAR2(30) NOT NULL,
TOTAL_EMPLEADOS NUMBER(4));
```

Agregando Múltiples Filas en diferentes Tablas

2

```
INSERT ALL
WHEN total_emp = 0 THEN
   INTO DEPTOS_SIN_EMPLEADOS(NOMBRE_DEPTO)  VALUES(department_name)
ELSE
   INTO DEPTOS_CON_EMPLEADOS(NOMBRE_DEPTO,TOTAL_EMPLEADOS)
   VALUES(department_name,total_emp)
SELECT d.department_name, COUNT(e.employee_id) total_emp
FROM departments d LEFT OUTER JOIN employees e
ON d.department_id=e.department_id
GROUP BY d.department_name
ORDER BY d.department_name;
```

DEPTOS_SIN_EMPLEADOS

	ID	NOMBRE_DEPTO
1	3	Benefits
2	4	Construction
3	5	Contracting
4	6	Control And Credit
5	7	Corporate Tax
6	10	Government Sales
.....		
.....		
14	23	Retail Sales
15	25	Shareholder Services
16	27	Treasury

DEPTOS_CON_EMPLEADOS

	ID	NOMBRE_DEPTO	TOTAL_EMPLEADOS
1	10	Accounting	2
2	20	Administration	1
3	80	Executive	3
4	90	Finance	6
5	110	Human Resources	1
.....			
.....			
9	210	Purchasing	6
10	240	Sales	34
11	260	Shipping	45

DuocUC®

Errores Frecuentes al Insertar Filas

No insertar un valor a una columna definida como obligatoria

Valores duplicados para las Constraints Primary Key o Unique

Valores de Foreign Key que no existen

Tipos de datos inconsistente

Valor demasiado largo

DuocUC®

Errores Frecuentes al Insertar Filas

- Ejemplo:

```
INSERT INTO departments VALUES (300, 'Purchasing',NULL,NULL);
```

```
INSERT INTO departments VALUES (300, 'Purchasing',NULL,NULL)
Informe de error -
ORA-00001: unique constraint (HR.DEPT_ID_PK) violated
```

- Ejemplo:

```
INSERT INTO departments VALUES (340,'Informática y Soporte de Plataformas',NULL,NULL);
```

```
INSERT INTO departments VALUES (340, 'Informática y Soporte de Plataformas',NULL,NULL)
Informe de error -
ORA-12899: value too large for column "HR"."DEPARTMENTS"."DEPARTMENT_NAME" (actual: 37, maximum: 30)
```

- Ejemplo:

```
INSERT INTO departments VALUES (340,NULL, NULL,NULL);
```

```
INSERT INTO departments VALUES (340,NULL, NULL,NULL)
Informe de error -
ORA-01400: cannot insert NULL into ("HR"."DEPARTMENTS"."DEPARTMENT_NAME")
```

DuocUC®

Errores Frecuentes al Insertar Filas

- Ejemplo:

```
INSERT INTO departments VALUES (340, 'Informática','INFORM',NULL);
```

```
INSERT INTO departments VALUES (340, 'Informática','INFORM',NULL)
Informe de error -
ORA-01722: invalid number
```

- Ejemplo:

```
INSERT INTO employees VALUES(113, 'Louis', 'Popp', 'LPOPP', '515.124.4567', SYSDATE,
        'AC_ACCOUNT', 6900, NULL, 205, 999);
```

```
INSERT INTO employees VALUES(309, 'Jorge', 'Salinas', 'JSALINAS',  '515.124.4567', SYSDATE,'AC_ACCOUNT', 6900, NULL, 205, 999)
Informe de error -
ORA-02291: integrity constraint (HR.EMP_DEPT_FK) violated - parent key not found
```

- Ejemplo:

```
INSERT INTO employees VALUES(113, 'Louis', 'Popp', 'LPOPP',  '515.124.4567', SYSDATE,
        'AC_ACCOUNT', 6900, NULL, 205, 100);
```

```
INSERT INTO employees VALUES(113, 'Louis', 'Popp', 'LPOPP',  '515.124.4567', SYSDATE,'AC_ACCOUNT', 6900, NULL, 205, 100)
Informe de error -
ORA-00001: unique constraint (HR.EMP_EMAIL_UK) violated
```

DuocUC®

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Modificando Valores de las Tablas

BDY1102: Noviembre, 2023

Modificando Datos de las Tabla

Modificando Datos de las Tabla

TABLA EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	100	King	90
2	101	Kochhar	90
3	102	De Haan	90
4	103	Hunold	60
5	104	Ernst	60
6	105	Austin	60
7	106	Pataballa	60
.....			
.....			
103	202	Fay	20
104	203	Mavris	40
105	204	Baer	70
106	205	Higgins	110
107	206	Gietz	110

Actualizar en la tabla EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	100	King	90
2	101	Kochhar	90
3	102	De Haan	90
4	103	Hunold	110
5	104	Ernst	110
6	105	Austin	60
7	106	Pataballa	60
.....			
.....			
103	202	Fay	20
104	203	Mavris	40
105	204	Baer	70
106	205	Higgins	110
107	206	Gietz	110

Modificando Datos de las Tabla

- Sintaxis:

```
UPDATE tabla
      SET columna = valor [, columna = valor, ...]
[WHERE condición];
```

- Ejemplo:

```
UPDATE  employees
      SET department_id = 110
 WHERE employee_id IN (103, 104);
```

- Ejemplo:

```
UPDATE employees
      SET hire_date = SYSDATE – 2,
            salary = salary * 2;
```

DuocUC®

Modificando Datos de las Tabla

Modificando Datos de las Tabla

- Ejemplo:

```
UPDATE employees
      SET salary = CASE
      WHEN extract(year from hire_date)=2004 THEN ROUND(salary*1.15)
      WHEN extract(year from hire_date)=2005 THEN ROUND(salary*1.13)
      ELSE ROUND(salary*1.105)
      END
WHERE manager_id IN(145,147);
```

- Ejemplo

```
UPDATE employees
      SET salary =  (SELECT ROUND(AVG(salary))
                     FROM employees)
 WHERE salary = (SELECT MIN(salary)
                     FROM employees);
```

Errores Frecuentes al Actualizar Datos

Valores de Foreign Key que no existen

Valores duplicados para las Constraints Primary Key o Unique

Tipos de datos inconsistente

Valores demasiado largos

DuocUC®

Errores Frecuentes al Actualizar Datos

- Ejemplo:

```
UPDATE employees
      SET email = 'JCHEN'
 WHERE employee_id=100;
```

```
UPDATE employees
SET email = 'JCHEN'
WHERE employee_id=100
Informe de error -
ORA-00001: unique constraint (HR.EMP_EMAIL_UK) violated
```

- Ejemplo:

```
UPDATE employees
SET department_id = 999
WHERE employee_id=100;
```

```
UPDATE employees
SET department_id = 999
WHERE employee_id=100
Informe de error -
ORA-02291: integrity constraint (HR.EMP_DEPT_FK) violated - parent key not found
```

DuocUC®

Errores Frecuentes al Actualizar Datos

- Ejemplo:

```
UPDATE employees
SET hire_date = 'AC_ACOUNT'
WHERE employee_id=100;
```

```
UPDATE employees
SET hire_date = 'AC_ACOUNT'
WHERE employee_id=100
Informe de error -
ORA-01858: a non-numeric character was found where a numeric was expected
```

- Ejemplo:

```
UPDATE employees
SET salary = 'AC_ACOUNT'
WHERE employee_id=100;
```

```
UPDATE employees
SET salary = 'AC_ACOUNT'
WHERE employee_id=100
Informe de error -
ORA-01722: invalid number
```

Errores Frecuentes al Actualizar Datos

- Ejemplo:

```
UPDATE employees
SET salary = 2456738901
WHERE employee_id=100;
```

```
UPDATE employees
SET salary = 2456738901
WHERE employee_id=100
Informe de error -
ORA-01438: value larger than specified precision allowed for this column
```

- Ejemplo:

```
UPDATE employees
SET commission_pct = 12.08
WHERE employee_id=100;
```

```
UPDATE employees
SET commission_pct = 12.08
WHERE employee_id=100
Informe de error -
ORA-01438: value larger than specified precision allowed for this column
```

DuocUC®

Errores Frecuentes al Actualizar Datos

- Ejemplo:

```
UPDATE employees
SET salary = 2456738901
WHERE employee_id=100;
```

```
UPDATE employees
SET salary = 2456738901
WHERE employee_id=100
Informe de error -
ORA-01438: value larger than specified precision allowed for this column
```

- Ejemplo:

```
UPDATE employees
SET commission_pct = 12.08
WHERE employee_id=100;
```

```
UPDATE employees
SET commission_pct = 12.08
WHERE employee_id=100
Informe de error -
ORA-01438: value larger than specified precision allowed for this column
```

DuocUC®

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Eliminando Filas de las Tablas

BDY1102: Noviembre, 2023

Eliminando Filas de las Tablas

TABLA DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	70	Public Relations	204	2700
8	80	Sales	145	2500
...	...	...	...	...
22	230	IT Helpdesk	(null)	1700
23	240	Government Sales	(null)	1700
24	250	Retail Sales	(null)	1700
25	260	Recruiting	(null)	1700
26	270	Payroll	(null)	1700

Eliminar la fila de la tabla DEPARTMENTS

TABLA DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	70	Public Relations	204	2700
7	80	Sales	145	2500
...	...	...	...	...
23	230	IT Helpdesk	(null)	1700
24	240	Government Sales	(null)	1700
25	250	Retail Sales	(null)	1700
26	260	Recruiting	(null)	1700
27	270	Payroll	(null)	1700

DuocUC®

Eliminando Filas de las Tablas

- Sintaxis:

```
DELETE [FROM] tabla
[WHERE condición];
```

- Ejemplo:

```
DELETE employees;
```

- Ejemplo:

```
DELETE FROM departments
WHERE department_name = 'Finance';
```

Eliminando Filas de las Tablas

- Ejemplo:

```
DELETE FROM departments
WHERE department_id IN(30, 40);
```

- Ejemplo:

```
DELETE employees
WHERE salary between 2000 AND 5000;
```

- Ejemplo:

```
DELETE departments
WHERE department_id NOT IN (SELECT DISTINCT department_id
                            FROM employees;
```

Error Frecuente al Eliminar Filas

Valores de Primary Key con
Con Foreign Key asociadas

- Ejemplo:

```
DELETE FROM departments
WHERE department_id = 90;
```

```
DELETE FROM departments
WHERE department_id = 90
Informe de error -
ORA-02292: integrity constraint (HR.EMP_DEPT_FK) violated - child record found
```

DuocUC®

Uso de Sentencia TRUNCATE para todas las Filas

Es la forma más eficiente para eliminar todas las filas de una tabla

La sentencia TRUNCATE (DDL) en más rápida que la sentencia DELETE (DML)

TRUNCATE tiene un COMMIT implícito

- Sintaxis:

```
TRUNCATE TABLE nombre_tabla;
```

- Ejemplo

```
TRUNCATE TABLE employees;
```

DuocUC®

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Uso de Subconsultas en Sentencias DML

BDY1102: Noviembre, 2023

Usando Subconsultas en Sentencias DML

Se pueden insertar filas a partir de otras tablas. Esto es a través de Subqueries o Subconsultas

Cuando se insertan filas, el número de columnas y tipos de datos deben coincidir con el número de valores y tipos de datos que retorna la subconsulta

Se pueden actualizar valores de columnas de una tablas a partir del resultado que retorna una subconsulta

Se pueden eliminar filas de una tabla a partir de una condición basada en una subconsulta

Usando Subconsultas para Insertar Filas

- Sintaxis:

```
INSERT INTO tabla [ columna (, columna ) ]
SUBCONSULTA;
```

- Ejemplo:

```
INSERT INTO empleado_resp(codigo, apellido, salario, porc_comision)
SELECT employee_id, last_name, salary, commission_pct
  FROM employees
 WHERE job_id LIKE '%REP%';
```

- Ejemplo:

```
INSERT INTO copia_emp
SELECT *
 FROM employees;
```

Usando Subconsultas para Insertar Filas

- Ejemplo:

```
INSERT INTO emp_minus
SELECT e.employee_id, e.first_name || ' ' || e.last_name ,  e.department_id,
          d.department_name,  e.job_id, j.job_title
 FROM employees e JOIN departments d
ON(e.department_id=d.department_id)
JOIN jobs j
ON(e.job_id=j.job_id)
MINUS
SELECT e.employee_id, e.first_name || ' ' || e.last_name ,  e.department_id,
          d.department_name, e.job_id, j.job_title
 FROM employees e JOIN departments d
ON(e.department_id=d.department_id)
JOIN jobs j
ON(e.job_id=j.job_id)
WHERE salary < ALL (SELECT ROUND(AVG(salary))
              FROM employees
              GROUP BY department_id);
```

Usando Subconsultas para Actualizar Datos

- Sintaxis:

```
UPDATE tabla
      SET columna = (SUBCONSULTA) [, columna = valor, ...]
[WHERE condición];
```

- Ejemplo:

```
UPDATE employees
     SET  job_id = (SELECT  job_id
                    FROM  employees
                    WHERE employee_id = 205),
          salary = (SELECT salary
                    FROM employees
                    WHERE employee_id = 205)
WHERE employee_id = 114;
```

Usando Subconsultas para Actualizar Datos

- Ejemplo:

```
UPDATE employees
      SET salary = (SELECT ROUND(AVG(salary))
                    FROM employees)
 WHERE salary = (SELECT MIN(salary)
                    FROM employees);
```

- Ejemplo:

```
UPDATE employees
      SET salary = ROUND(salary * .125)
 WHERE department_id IN (SELECT DISTINCT department_id
                         FROM employees
                        WHERE salary <  (SELECT ROUND(AVG(salary))
                                         FROM employees));
```

DuocUC®

Usando Subconsultas para Eliminar Filas

- Sintaxis:

```
DELETE [FROM] tabla
WHERE columna operador_comparación (SUBCONSULTA)
```

- Ejemplo:

```
DELETE FROM employees
WHERE salary = (SELECT ROUND(AVG(salary))
                FROM employees);
```

- Ejemplo:

```
DELETE from employees_copia
WHERE department_id IN(SELECT department_id
                       FROM employees
                       HAVING COUNT(employee_id) > 10
                       GROUP BY department_id);
```

Resumen de la Clase

- Se describieron las características del lenguaje DML y Transacciones de Base de Datos
- Se explicó cómo construir sentencias DML para Insertar Filas a las Tablas.
- Se explicó cómo construir sentencias DML para Actualizar Filas de las Tablas.
- Se explicó cómo construir sentencias DML para Eliminar Filas de las Tablas.
- Se explicó el uso de la sentencia TRUNCATE para eliminar filas de las Tablas.
- Se explicó cómo usar Subconsultas en sentencias DML.