



# Ingeniería de Datos Base de Datos Relacionales DML y Consultas Básicas

Nadia Rodríguez

2013-1

# Agenda

---

- Lenguajes de Base de Datos Relacional
- Lenguaje de Manipulación de Datos (DML)
- Instrucciones DML
  - INSERT
  - UPDATE
  - DELETE
- SELECT

# Transacciones con la BDR

---

- Sentencias DML que realizan un cambio consistente en los datos
- Sentencias Lenguaje de Definición de Datos (DDL)
- Sentencias Lenguaje de Control de Datos (DCL)

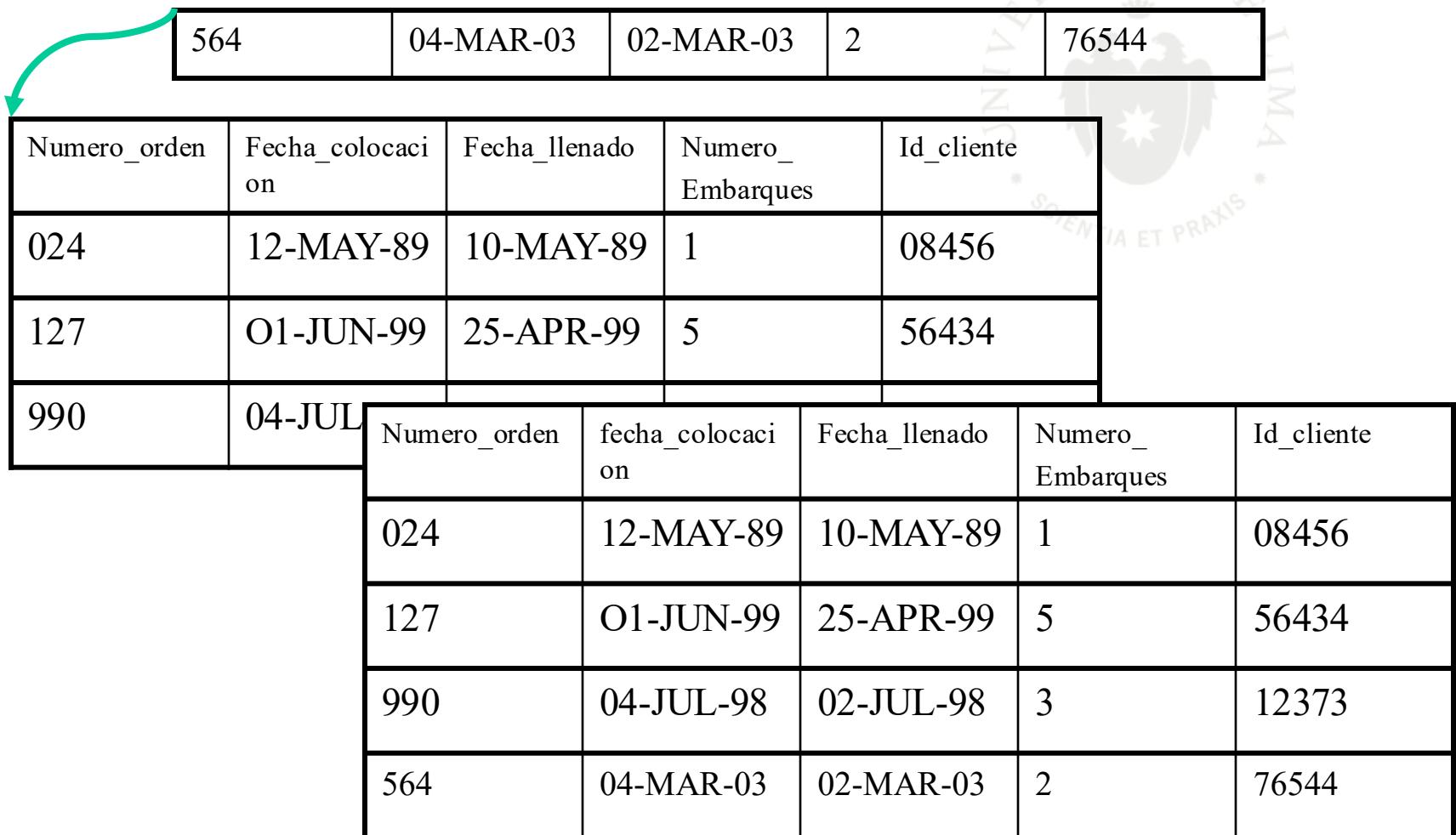
# Lenguajes de la BDR

---

- Recuperación de datos
  - SELECT
- Lenguaje de Manipulación de Datos (DML)
  - INSERT, UPDATE, DELETE
- Lenguaje de definición de Datos (DDL)
  - CREATE, ALTER, DROP, RENAME
- Lenguaje de Control de Datos (DCL)
  - GRANT, REVOKE
  - Control de transacción
    - COMMIT, ROLLBACK, SAVEPOINT

# INSERT

- Añade o inserta una nueva fila a una tabla



The diagram illustrates the process of inserting a new row into a database table. A green curved arrow points from a small table at the top to a larger table below it.

**Small Table (Top):**

564	04-MAR-03	02-MAR-03	2	76544
-----	-----------	-----------	---	-------

**Large Table (Bottom):**

Numero_orden	Fecha_colocacion	Fecha_llenado	Numero_Embarques	Id_cliente
024	12-MAY-89	10-MAY-89	1	08456
127	01-JUN-99	25-APR-99	5	56434
990	04-JUL-98			

Numero_orden	fecha_colocacion	Fecha_llenado	Numero_Embarques	Id_cliente
024	12-MAY-89	10-MAY-89	1	08456
127	01-JUN-99	25-APR-99	5	56434
990	04-JUL-98	02-JUL-98	3	12373
564	04-MAR-03	02-MAR-03	2	76544

# INSERT

```
INSERT INTO table [column, column...]  
VALUES (column, column ...);
```

- Todos los campos NOT NULL deben ingresarse
- Se puede especificar la elección de los campos a ingresar en cada registro, pero debe considerarse la restricción de dominio NOT NULL (obligatoria)
- El valor NULL para un campo es valido siempre y cuando tenga restricción de dominio NOT NULL

# Ejemplos de INSERT

---

***Insertar la orden de pedido # 555 para el cliente JOSE PARDO.***

```
CREATE TABLE cliente  
  (codigo_cliente NUMBER(5) PRIMARY KEY,  
   nombre        VARCHAR2(25) NOT NULL,  
   apellido      VARCHAR2(50) NOT NULL,  
   telefono      VARCHAR2(10) NOT NULL);
```

```
CREATE TABLE orden_pedido  
  (numero_orden      NUMBER(7) PRIMARY KEY,  
   fecha_colocación  DATE,  
   fecha_llenado     DATE,  
   numero_embarques  NUMBER(5),  
   id_cliente        NUMBER(7) REFERENCES CLIENTE );
```

```
INSERT INTO cliente  
  VALUES (1, 'JOSE', 'PARDO', '01-4414896');
```

```
INSERT INTO orden_pedido  
  VALUES (555, '25-OCT-07', '05-OCT-07', 5, 1);
```

# UPDATE

- Modifica el valor de un campo existente

Numerp_order	fecha_coloración	Fecha_llenado	Numero_Embarques	Id_cliente
024	12-MAY-89	10-MAY-89	1	08456
127	01-JUN-99	25-APR-99	5	56434
Numerp_order	fecha_coloración	Fecha_llenado	Numero_Embarques	Id_cliente
024	12-MAY-89	10-MAY-89	1	08456
127	01-JUN-99	25-APR-99	3	56434
990	04-JUL-98	02-JUL-98	3	12373
564	04-MAR-03	02-MAR-03	2	76544

# UPDATE

```
UPDATE table
SET column = new_value | calculation
[WHERE condition {AND|or} condition...]
```

- Actualiza en todas las filas de un campo dado perteneciente a una tabla, asignándole un valor predeterminado o el resultado de una operación aritmética sobre otro(s) campo(s)
- Puede elegirse solo algunas filas de una tabla para actualizarlas con la cláusula WHERE a través de una o mas condiciones enlazadas por los conectores lógicos AND|OR|NOT

# Ejemplos de UPDATE

---

*La orden de pedido # 555 va a tener 3 embarques en lugar de 5.*

```
UPDATE orden_pedido  
SET numero_embarques = 3  
WHERE numero_orden = 555;
```

*El nuevo número de teléfono del cliente JOSE PARDO es 01-4367788*

```
UPDATE cliente  
SET telefono = '01-4367788'  
WHERE codigo_cliente = 1;
```

# DELETE

- Elimina una o más filas de una tabla

		564	04-MAR-03	02-MAR-03	2	76544
Numerp_order	fecha_colocació	Fecha_llenado	Numero_Embarques	Id_cliente		
024	12-MAY-89	10-MAY-89	1	08456		
127	01-JUN-99	25-APR-99	5	56434		
990	04-JUL-98	02-JUL-98	3	12373		
564	04-MAR-03	Numerp_order	fecha_colocació	Fecha_llenado	Numero_Embarques	Id_cliente
		024	12-MAY-89	10-MAY-89	1	08456
		127	01-JUN-99	25-APR-99	5	56434
		990	04-JUL-98	02-JUL-98	3	12373

# DELETE

```
DELETE FROM table
[WHERE condition {AND|or} condition...]
```

- Elimina una o más filas de una tabla
- Puede elegirse solo algunas filas de una tabla para eliminarlas con la cláusula WHERE a través de una o mas condiciones enlazadas por los conectores lógicos AND|OR|NOT

# Ejemplo de DELETE

---

*El cliente ha decidido eliminar su orden de pedido # 555.*

```
DELETE FROM orden_pedido  
WHERE numero_orden = 555;
```

# SELECT

- Permite consultar la data contenida en una tabla

```
SELECT ALL|*|DISTINCT| (column, column...)  
FROM table  
[WHERE condition {AND|or} condition...]
```

- Muestra todas la filas de una tabla
- Puede elegirse mostrar solo algunas filas de una tabla con la cláusula WHERE a través de una o mas condiciones enlazadas por los conectores lógicos AND|OR|NOT
- Se utiliza ALL o \* para mostrar todos los campos o columnas de una tabla
- Se utiliza DISTINCT para mostrar valores diferentes, sin duplicados

# Ejemplos de SELECT

---

SELECT ALL FROM orden\_pedido;

SELECT \* FROM cliente;

SELECT \* FROM cliente  
WHERE codigo\_cliente = 1;

SELECT DISTINCT nombre FROM cliente;

SELECT nombre, apellido FROM cliente  
WHERE codigo\_cliente = 1;