

# Bases de Datos I

## Unidad VIII

**Lenguaje de Definición de Datos  
SQL/DML**

**INSERT, UPDATE, DELETE y  
SELECT**



## **Dando vueltas sobre lo mismo...**

- **Si bien estuvimos trabajando hasta ahora con SQL, no todo lo que dijimos sobre las sentencias SQL/DML es cierto.**
- **En ocasiones ocultamos parte del verdadero poder de algunas sentencias, para ocultar su complejidad.**
- **Veamos el verdadero poder de las distintas sentencias, y como se pueden utilizar en conjunto.**

# Sentencia INSERT

- Si bien vimos que la sintaxis es:

**INSERT INTO** [tabla] [(cols)]

**VALUES** [(fila1), (fila2)];

- No es cierto que **VALUES**, sea parte de la sentencia **INSERT**.
- **VALUES** es una sentencia en sí misma.

# Sentencia **VALUES**

- **Calcula una fila o conjunto de filas, especificadas por las expresiones de valor de las mimas.**
- **Ejemplo:**  
**VALUES** (32111154, 'Ramos', 'Nadia Patricia'),  
(29458374, 'Perez', 'José Manuel');

# Sentencia **VALUES**

- Incluso podemos realizar el ordenamiento de las filas, de igual forma que con cualquier consulta **SELECT**.

- **Ejemplo:**

**VALUES** (32111154, 'Ramos', 'Nadia Patricia'),  
(29458374, 'Perez', 'José Manuel');

**ORDER BY 2 DESC;**

# Sentencia INSERT

- Entonces **INSERT**, espera un conjunto de filas que puede ser el resultado de una consulta **SELECT**, no sólo el resultado de una sentencia **VALUES**.

## **INSERT INTO**

cliente(documento, apellido, nombre)

**SELECT** documento, apellido, nombre

**FROM** empleado;

# Sentencia UPDATE

- Si bien vimos que la sentencia UPDATE se utiliza para modificar columnas de las filas de una tabla, también puede modificar expresiones de filas.

**UPDATE** empleado **SET**

(apellido, nombre) = ('Perez', 'Pedro')

**WHERE** documento = 11;

# Expresiones de fila

- Las expresiones de fila pueden utilizarse en la consultas también.

**SELECT \* FROM** empleado

**WHERE** (apellido, nombre) **IN** (**SELECT** apellido, nombre  
**FROM** cliente);



## Subconsultas para todos

- **No sólo en la sentencia SELECT pueden usarse subconsultas, también en las sentencias INSERT, UPDATE y DELETE.**
- **En la sentencia INSERT, para obtener valores o filas a insertar.**
- **En la sentencia UPDATE, para obtener valores a modificar o filas y en la clausula WHERE.**
- **En la sentencia DELETE, en la clausula WHERE.**

## Subconsultas en INSERT

```
INSERT INTO total_mensual(mes, total)
VALUES(1, (SELECT SUM(monto)
           FROM venta
           WHERE EXTRACT(MONTH FROM fecha)=1)
);
```

# Subconsultas en UPDATE

- Utilizamos una subconsulta en la clausula de asignación:

```
UPDATE total_mensual SET  
total = (SELECT SUM(monto)  
        FROM venta  
WHERE EXTRACT(MONTH FROM fecha) = 1) WHERE  
mes = 1;
```

## Subconsultas en UPDATE

- **Utilizamos una subconsulta en la clausula de selección:**

**UPDATE** empleado **SET**

retiro = CURRENT\_DATE

**WHERE** legajo **IN** (**SELECT** legajo  
**FROM** bajas);

- **Subconsultas en DELETE**

- **Utilizamos una subconsulta en la clausula de selección:**

**DELETE FROM** empleado

**WHERE** documento **IN** (**SELECT** documento  
**FROM** cliente);

# Tablas derivadas

- **En realidad también es posible utilizar subconsultas en la cláusula FROM.**
- **En este caso las subconsultas reciben el nombre de tablas derivadas.**

## Ejemplo

- Queremos obtener el número de factura y el monto total de la misma.

```
SELECT f.numero, f.fecha, m.monto
FROM factura AS f, (SELECT l.factura,
    SUM(l.cantidad * p.precio) AS monto
    FROM linea AS l, producto AS p
    WHERE l.producto = p.codigo
    GROUP BY l.factura) AS m
WHERE f.numero = m.factura;
```

## Ejemplo con sentencia **VALUES**

- Incluso podemos utilizar **VALUES** en una construcción de tabla derivada o en la cláusula **WITH**

- Ejemplo:

**SELECT** T.\*

**FROM** (

- **VALUES** (32111154, 'Ramos', 'Nadia Patricia'),  
(29458374, 'Perez', 'José Manuel')) **AS** T;



# Sentencia SELECT

- **Cláusulas de la sentencia SELECT:**

**WITH** → Cláusula de vistas temporales

**SELECT** → Cláusula de proyección

**FROM** → Cláusula de origen/reunión

**WHERE** → Cláusula de selección

**GROUP BY** → Cláusula de agrupamiento

**HAVING** → Cláusula de selección de grupos

**ORDER BY** → Cláusula de orden

**OFFSET FETCH** → Cláusula límite y paginado

# Sentencia SELECT

- **Orden de evaluación de las cláusulas:**

**1 → WITH**

**6 → SELECT**

**2 → FROM**

**3 → WHERE**

**4 → GROUP BY**

**5 → HAVING**

**7 → ORDER BY**

**8 → OFFSET FETCH**

# Expresiones comunes de tablas

- **El uso de subconsultas y tablas derivadas complejiza la escritura y lectura de consultas.**
- **Utilizando la cláusula WITH, podemos declarar consultas, proporcionarles un nombre y utilizarlas dentro de la consulta como una tabla más.**
- **La cláusula WITH también permite realizar consultas recursivas.**

# Índices

- **Aceleran la búsqueda y recuperación de datos,**
- **Mejoran el rendimiento actuando como un índice alfabético en un libro.**
- **Reducen el tiempo para encontrar información en grandes conjuntos de datos, mejora el rendimiento en consultas y búsquedas.**

**CREATE INDEX** idx\_factura\_fecha **ON** factura (fecha);

Se crea un índice llamado idx\_factura\_fecha en la tabla factura, sobre la columna fecha, que acelerará las búsquedas de datos en función de los valores de la columna fecha.

# Bibliografía

- **SQL-99 Complete, Really. 1999. Peter Gultzan y Trudy Pelzer.**
- **PostgreSQL Introduction and Concepts. 2001. Momjian, Bruce.**
- **PostgreSQL 9.6 Documentation. 2016. The PostgreSQL Global Development Group.**