

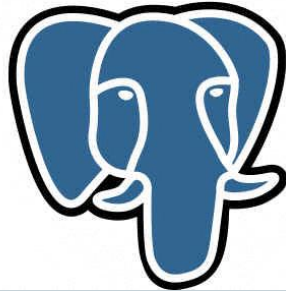
# **CURSO DE FUNDAMENTOS Y ADMINISTRACION DE DBMS**

**UBUNTU 10.4**



**SYBASE**

PostgreSQL



The **Apache**  
Software Foundation



**INSTRUCTOR: RUDY SALVATIERRA  
RODRIGUEZ**

# SQL: APLICACION LENGUAJE DML



- **INSERT** Utilizado para cargar lotes de datos en la base de datos en una única operación.
- **UPDATE** Utilizado para modificar los valores de los campos y registros especificados.
- **DELETE** Utilizado para eliminar registros de una tabla de una base de datos.
- **SELECT** Utilizado para poder realizar consultas a las tablas y generar reportes.

# SQL: APLICACION LENGUAJE DML

## INSERT



- Permite insertar registros (instancias) a una tabla con la siguiente sintaxis.

Sintaxis basico:

○ **insert into nombre\_tabla values('valores\_columna', ...);**

Ej.

**insert into** docente values (101,'alex','Rodriguez Rojas','Ing. De sistemas');

**insert into** materia values (1,'Introduccion a la programacion', 'primer semestre');

**insert into** dia values (1, 'lunes');

**insert into** Aula values( 1,'692 B','bloque nuevo',120 );

**insert into** Horario values( 1,'01-05-2008','15:45', '17:15', 'tarde');

# SQL: APLICACION LENGUAJE DML

## UPDATE



- Permite modificar registros en la tabla.

Sintaxis básico:

**update** nombre\_tabla set nombre\_columna = nuevo\_valor,...  
**where** condiciones;

Ej.

**update** docente set nombre\_doc = 'docente';

**update** docente set nombre\_doc = 'juan' **where** cod\_doc = 101;

**update** docente set nombre\_doc= 'jorge',  
apellido\_doc='valdez' **where** cod\_doc=1;

# SQL: APLICACION LENGUAJE DML

## DELETE



- Permite eliminar registros de una tabla

Sintaxis basico:

○ **delete from** nombre\_tabla where condiciones;

Ej.

**delete from** docente where codigo\_doc = '101';

**delete from** docente; OJO con este comando, elimina todos los registros de la tabla.

# SQL: APLICACION LENGUAJE DML

## SELECT



- Recuperando datos de una tabla

Sintaxis basico:

○ **select** columnas **from** nombre\_tabla, ... **where** condición;

Ej.

**select \* from** docente;

**select \* from** materia;

**select \* from** horario;

**select** cod\_doc, nombre\_doc **from** docente;

**select** cod\_doc, nombredoc **from** docente **where**  
docente.apellido\_pat='Rodriguez';

# SQL: APLICACION LENGUAJE DML

## SELECT



- Sentencia **SELECT** :  
Se usa para listar las columnas de las tablas que se desean ver en el resultado de una columna.
- Clausula **FROM**:  
Lista las tablas que deben ser analizadas en la evaluación .
- Clausula **WHERE**:  
Establece las condiciones y criterio de selección de ciertas filas en el resultado de dicha consulta.

**SELECT    DISTINCT    \*    FROM[TABLAS]    WHERE  
[CONDICION] [GROUP BY ][ORDER BY][HAVING];**

- Clausula **DISTINCT**: permite mostrar valores que no se repitan.

# SQL: APLICACION LENGUAJE DML

## SELECT



- Clausula **HAVING**:

Incluye a ciertos grupos de la clausula group by bajo una condición dada .

- Clausula **GROUP BY**:

Especifica una consulta sumaria. En ves de producir una fila de resultados por cada fila agrupa todas las filas similares.

- Clausula **ORDER BY**:

Permite mostrar el resultado de la consulta de forma ordenada.

```
postgres=# SELECT * FROM docente WHERE cod_doc > 5 ;
```

Columnas  
seleccionadas

Tablas  
participantes

Condiciones a  
tomar encuesta



# SQL: APLICACION LENGUAJE DML

## OPERADORES LOGICOS



- **AND** Es el “y” lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
- **OR** Es el “o” lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
- **NOT** Negación lógica. Devuelve el valor contrario de la expresión.

# SQL: APLICACION LENGUAJE DML

## CONDICIONES DE BUSQUEDA



### Operador de comparación

- < Menor que
- > Mayor que
- = igual que
- <> Distinto de
- != Distinto de
- <= Menor ó Igual que
- >= Mayor ó Igual que

# SQL: APLICACION LENGUAJE DML

## CONDICIONES DE BUSQUEDA



- **BETWEEN** Utilizado para especificar un intervalo de valores.

```
SELECT tipo,modelo,precio FROM piezas  
WHERE precio BETWEEN 3 AND 8;
```

Saca piezas cuyos precios estén entre 3 y 8 (ambos incluidos).

- **IN** Permite obtener registros cuyos valores estén en una lista:

```
SELECT tipo,modelo,precio FROM piezas  
WHERE precio IN (3,5, 8);
```

- Obtiene piezas cuyos precios sea 3, 5 u 8, sólo uno de esos tres.

# SQL: APLICACION LENGUAJE DML

## CONDICIONES DE BUSQUEDA



- **LIKE** Se usa sobre todo con textos, permite obtener registros cuyo valor en un campo cumpla una condición textual. LIKE utiliza una cadena que puede contener estos símbolos:

Símbolo	Significado
%	Una serie cualquiera de caracteres
_	Un carácter cualquiera

```
/* Selecciona nombres que empiecen por S */  
SELECT nombre FROM personas WHERE nombre LIKE 'A%';  
  
/*Selecciona las personas cuyo apellido sea Sanchez, Senchez,  
Stnchez,...*/  
SELECT apellido1 FROM Personas WHERE apellido1 LIKE 'S_nchez';
```

# SQL: APLICACION LENGUAJE DML

## CONDICIONES DE BUSQUEDA



- **IS NULL** : Devuelve verdadero si una expresión contiene un nulo:

```
SELECT nombre,apellidos FROM personas  
WHERE telefono IS NULL
```

Esa instrucción selecciona a la gente que no tiene teléfono

# SQL: APLICACION LENGUAJE DML

## CONDICIONES DE BUSQUEDA



- **Precedencia de operadores**

Orden de precedencia	Operador
1	*(Multiplicar) / (dividir)
2	+ (Suma) - (Resta)
3	(Concatenación)
4	Comparaciones (>, <, !=, ...)
5	IS [NOT] NULL, [NOT ]LIKE, IN
6	NOT
7	AND
8	OR

# SQL: APLICACION LENGUAJE DML

## FUNCIONES DE AGREGACION



- **SUM:** Nos permite obtener la suma de cantidades de las filas de una columna dada.

```
SELECT SUM (cantidad) as 'Suma de las Cantidades' FROM detalle_ped
```

- **AVG:** Nos permite poder hallar la cantidad promedio de un conjunto de cantidades que se encuentran en cierta columna.

```
SELECT AVG (cantidad) as 'Cantidad Promedio Pedida' FROM detalle_ped
```

# SQL: APLICACION LENGUAJE DML

## FUNCIONES DE AGREGACION



- **MIN:** Obtiene la cantidad mínima de un conjunto de numero que estan en una columna.

```
SELECT MIN (cantidad) as 'Cantidad Mínimo Pedida' FROM detalle_ped
```

- **MAX :** Obtiene la cantidad máxima de un conjunto de numero que estan en una columna.

```
SELECT MAX (cantidad) as 'Cantidad Máxima Pedida'  
FROM detalle_ped
```

- **COUNT:** Obtiene la cantidad de tuplas que tiene una

```
SELECT COUNT (*) as 'Total de Clientes' FROM CLIENTE
```



# SQL: APLICACION LENGUAJE DML

## FUNCIONES DE AGREGACION



- **CONCAT** : Nos permite concatenar cadenas para mostrar resultados.

```
mysql> select productos.clave, concat(parte,' ',tipo,' ', especificación) as producto,  
proveedores.empresa , precio , pago from productos natural join precios natural join proveedores;
```

- **SELECT INTO:** MySQL Server no soporta la sintaxis de extensiones Sybase SQL: SELECT ... INTO TABLE .... En su lugar, MySQL Server soporta la sintaxis estándar SQL INSERT INTO ... SELECT ..., que básicamente es lo mismo.

```
INSERT INTO tbl_temp2 (fld_id)  
  SELECT tbl_temp1.fld_order_id  
  FROM tbl_temp1 WHERE tbl_temp1.fld_order_id > 100;
```