

PostgreSQL Cheatsheet – Básico a Intermedio

Curso: SQL/PostgreSQL **Módulo:** Gestión de Bases de Datos **Tema:** Sintaxis, consultas y administración en PostgreSQL

Introducción

PostgreSQL es un sistema de gestión de bases de datos relacional y open-source. Principales características:

- Soporta SQL estándar y extensiones avanzadas
- Compatible con transacciones ACID
- Funciona con tipos de datos complejos y JSON
- Permite funciones, triggers y procedimientos almacenados

Cliente clave: **psql** (interfaz de línea de comandos)

Conexión y Bases de Datos

```
-- Conectar a PostgreSQL
psql -U usuario -d basedatos

-- Crear base de datos
CREATE DATABASE nombre_db;

-- Eliminar base de datos
DROP DATABASE nombre_db;

-- Listar bases de datos
\l
```

Gestión de Usuarios y Roles

```
-- Crear usuario con contraseña
CREATE USER dt_user WITH PASSWORD 'contraseña_segura';

-- Crear rol con permisos
CREATE ROLE admin_role;

-- Conceder privilegios
GRANT ALL PRIVILEGES ON DATABASE nombre_db TO dt_user;

-- Revocar privilegios
REVOKE ALL PRIVILEGES ON DATABASE nombre_db FROM dt_user;
```

❖ Estructura de Tablas

```
-- Crear tabla
CREATE TABLE clientes (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(50),
  email VARCHAR(100) UNIQUE,
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Modificar tabla
ALTER TABLE clientes ADD COLUMN telefono VARCHAR(20);

-- Eliminar tabla
DROP TABLE clientes;

-- Ver estructura de tabla
\d clientes
```

❖ Consultas Básicas

```
-- Seleccionar datos
SELECT * FROM clientes;
SELECT nombre, email FROM clientes WHERE id = 1;

-- Insertar datos
INSERT INTO clientes (nombre, email, telefono) VALUES
('Axel', 'axel@mail.com', '12345678');

-- Actualizar datos
UPDATE clientes SET email = 'nuevo@mail.com' WHERE id = 1;

-- Eliminar datos
DELETE FROM clientes WHERE id = 1;
```

❖ Operadores y Filtros

Operador	Uso	Ejemplo
=	Igualdad	WHERE id = 5
<> o !=	Diferente	WHERE nombre <> 'Axel'
> < >= <=	Comparaciones	WHERE id > 3
LIKE	Coincidencia de patrones	WHERE nombre LIKE 'A%'

Operador	Uso	Ejemplo
IN	Valores dentro de lista	WHERE id IN (1,2,3)
BETWEEN	Rango	WHERE id BETWEEN 1 AND 5
IS NULL / IS NOT NULL	Valores nulos	WHERE email IS NOT NULL

◆ Funciones Comunes

```
-- Agregación
SELECT COUNT(*), AVG(id), MAX(fecha_registro) FROM clientes;

-- Funciones de texto
SELECT UPPER(nombre), LOWER(nombre), LENGTH(email) FROM clientes;

-- Fechas
SELECT CURRENT_DATE, CURRENT_TIME, NOW();
```

◆ Join y Relaciones

```
-- INNER JOIN
SELECT o.id, c.nombre
FROM ordenes o
INNER JOIN clientes c ON o.cliente_id = c.id;

-- LEFT JOIN
SELECT o.id, c.nombre
FROM ordenes o
LEFT JOIN clientes c ON o.cliente_id = c.id;

-- RIGHT JOIN
SELECT o.id, c.nombre
FROM ordenes o
RIGHT JOIN clientes c ON o.cliente_id = c.id;
```

◆ Transacciones

```
-- Iniciar transacción
BEGIN;

-- Ejecutar operaciones
INSERT INTO clientes (nombre,email) VALUES ('Juan','juan@mail.com');

-- Confirmar cambios
```

```
COMMIT;  
  
-- Deshacer cambios  
ROLLBACK;
```

◆ Copia de Seguridad y Restauración

```
-- Exportar base de datos  
pg_dump -U usuario nombre_db > backup.sql  
  
-- Importar base de datos  
psql -U usuario -d nombre_db < backup.sql
```

◆ Buenas Prácticas

- Nombrar tablas y columnas de forma **clara y consistente**
- Usar **primary key** y **foreign key** para integridad
- Evitar duplicar información; normalizar tablas
- Usar transacciones para operaciones críticas
- Documentar consultas complejas con comentarios:

```
-- Esto es un comentario
```

◆ Otros casos

En muchas instalaciones modernas de PostgreSQL en Windows/WSL, el archivo pg_hba.conf no existe físicamente donde uno espera, porque el servicio se está ejecutando en un data directory dentro de WSL o incluso dentro de una instalación de Docker o de la app de PostgreSQL de Windows, y no es accesible directamente desde /etc/postgresql/....

Eso explicaría por qué:




Buscando pg_hba.conf manualmente te aparece vacío.

Incluso si abris la ruta típica /etc/postgresql/..., está vacío.

```
-- En WSL, lo más confiable es conectarse al servidor PostgreSQL usando TCP  
(localhost) y md5 en lugar de peer.  
-- Esto evita depender del archivo pg_hba.conf que no está accesible.  
  
-- Por ejemplo:  
  
psql -h localhost -U dt_user -d debt_tracker_db -W
```

- -h localhost → fuerza conexión TCP
- -U dt_user → usuario de la base
- -d debt_tracker_db → base de datos
- -W → fuerza que pida contraseña

Referencias oficiales

-  [Documentación PostgreSQL](#)
-  [PostgreSQL Tutorial](#)
-  [SQL Reference – W3Schools](#)