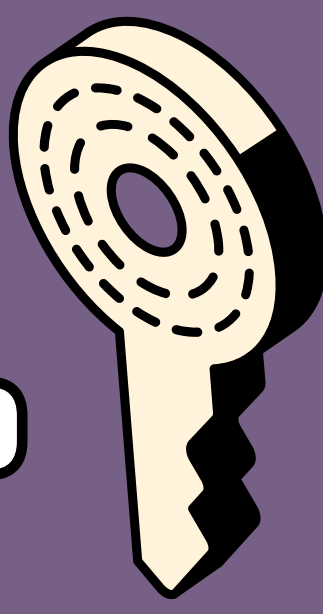
¿QUÉ ES RLS EN POSTGRESQL?



CONCEPTO

Row-Level Security permite definir políticas que controlan qué filas de una tabla puede ver o modificar un usuario determinado.

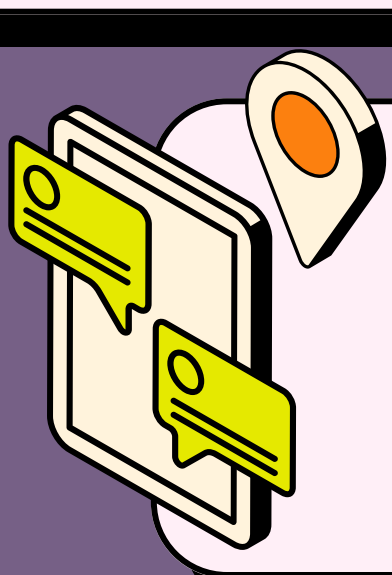
USOS

- Restringir datos por usuario o rol.
- Implementar reglas de privacidad y seguridad a nivel fila.
- Cumplir normativas como GDPR o HIPAA.

¿CÓMO SE APLICA RLS EN POSTGRESQL?

01 ACTIVAR RLS EN LA TABLA

ALTER TABLE empleados ENABLE **ROW** LEVEL SECURITY;

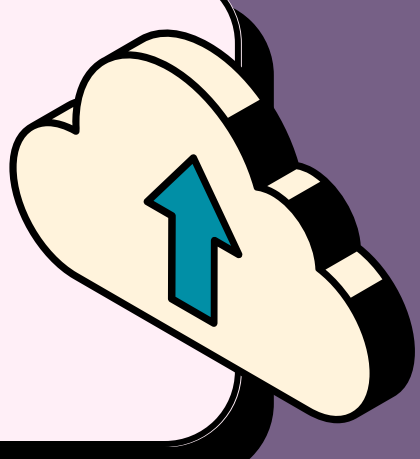


02 CREAR POLÍTICAS

CREATE POLICY solo_sus_datos
ON empleados
F**OR SELECT**
USING (usuario_id = current_user_id());

03 ASIGNAR POLÍTICAS

ALTER TABLE empleados ENABLE POLICY solo_sus_datos;



EJEMPLO PRÁCTICO

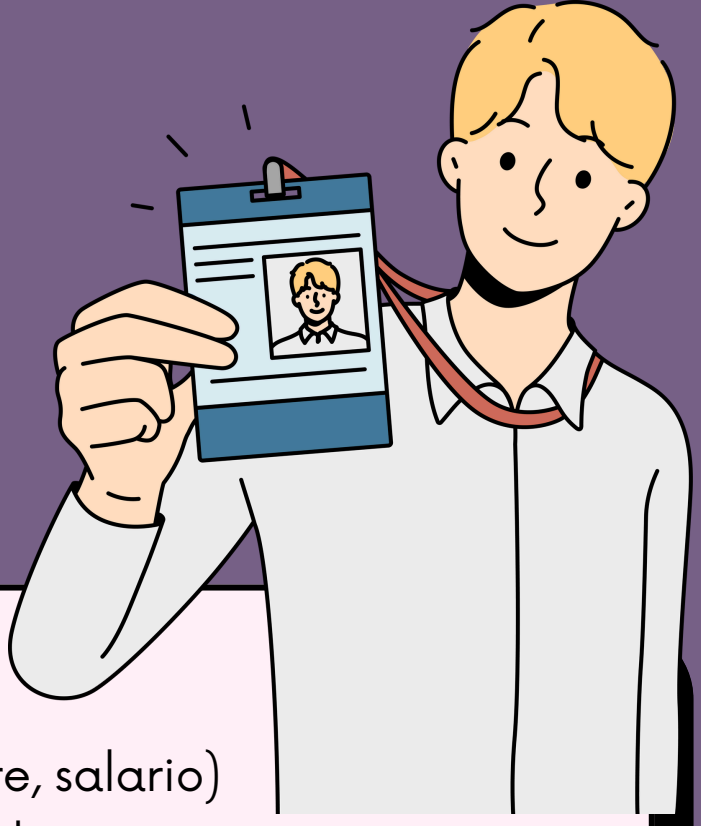
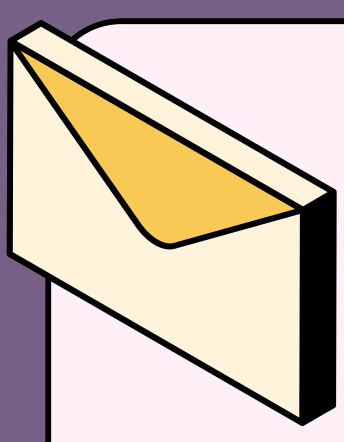


Tabla: empleados(usuario_id, nombre, salario)
Caso: Solo puedes ver tus propios datos.
👤 **Usuario:** usuario_1
🔎 **Consulta:** SELECT * FROM empleados;
✅ **Resultado:** Solo filas con usuario_id = usuario_1.



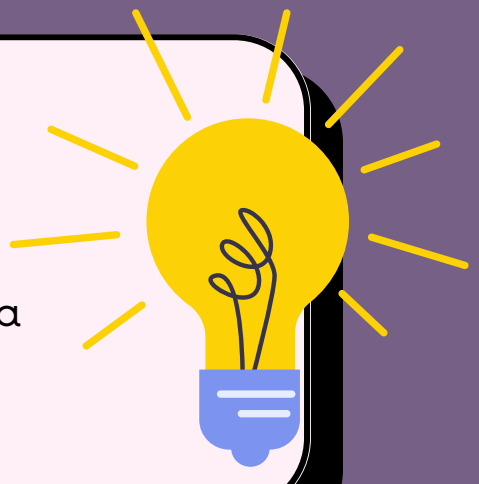
MODELO DE BASE DE DATOS

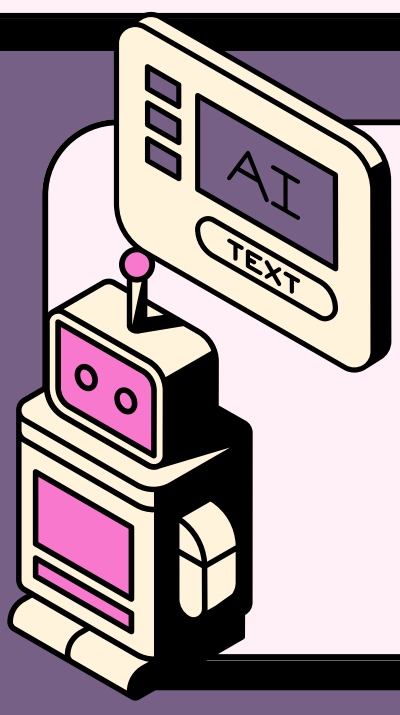
Incluye:

- Usuarios
- Roles
- Relaciones con las tablas protegidas por RLS

CONSEJO

Usar funciones como **current_user**, **current_setting**('my.app.userid') o **SET** para establecer contexto del usuario.





VENTAJAS DE USAR RLS

- Seguridad granular por fila
- Separación de responsabilidades
- Reglas centralizadas en la base de datos
- Más difícil de evadir que filtros a nivel de aplicación
- Compatible con vistas y funciones

FUNCIONES COMPLEMENTARIAS

- **pg_has_role(user, role, privilege)** – verificar permisos
- **set_config() / current_setting()** – establecer y obtener variables de sesión
- **pg_policies** – vista del catálogo con políticas activas