

Transacciones

Las transacciones, tienen la capacidad para empaquetar varios pasos en una sola operación "todo o nada" y si ocurre que alguna falla que impida que se complete la transacción, entonces ninguno de los pasos se ejecuta y no se afecta la base de datos en absoluto.

- Las transacciones nos ayudan a llevar a cabo procesos complejos de manera segura, si una transacción falla, la base de datos no genera los cambios.
- Las funciones para las transacciones son: BEGIN, COMMIT y ROLLBACK
- BEGIN inicia la consulta con las condiciones, COMMIT guarda los cambios al final y ROLLBACK devuelve todo si algo quedo mal
- Debemos verificar en las opciones de ejecución que no esté activado el COMMIT
- Postgre por defecto corre el ROLLBACK cuando una transacción presenta problemas

SQL Transacción - Estructura Las transacciones tienen la siguiente estructura postgres. **Postgres en las operaciones normales usa de manera implícita el rollback.**

```
BEGIN;  
<br>Intrucciones<br>  
COMMIT|ROLLBACK
```

SQL Transacción - Ejemplo en PgAdmin

1. Desactivamos en la esquina superior de pg-admin el auto commit
2. Iniciamos la transacción

```
BEGIN;  
INSERT INTO public.estacion(nombre,direccion)
```

```
VALUES('Estación Transacción',' 1');

INSERT INTO public.tren(modelo, capacidad)
VALUES('Modelo Transacción','2');

COMMIT;
```

SQL Transacción - Ejemplo de un rollback implícito Como se puede visualizar en el ejemplo existe una inserción correcta en la tabla tren pero en la tabla estación se está haciendo un insert a un id que existe realmente.

```
BEGIN;

INSERT INTO public.tren(modelo, capacidad)
VALUES('Modelo Transacción 2','2');

INSERT INTO public.estacion(id, nombre, direccion)
VALUES(101, 'Estación Transacción 2', ' 1');

COMMIT;
```

- Si ocurriera que al hacer una transacción, hay algún error, automáticamente se hará un rollback y no insertará nada que se encuentre dentro de esta.