

PREPARE

Totes les SQL que se llencen sobre la base de dades han de ser prèviament parsetjades i analitzades. El parseitg inclou validar que totes les taules de la consulta existeixen i són accessibles, que els camps existeixen, que la query està ben formada, etc. L'anàlisi inclou determinar quin és el millor pla d'execució: utilitzar o no un índex, si s'ha de crear una taula temporal en memòria, etc.

Evidentment, aquestes dues operacions tarden un temps. Si una consulta es llença moltes vegades, tot el temps invertit a partir de la segona vegada és temps que ens podríem estalviar si guardem el resultat d'aquestes dues passes.

La instrucció PREPARE fa exactament això, a més de re-escriure la consulta si és optimitzable. Se pot utilitzar per a les comandes SELECT i per a les DML (INSERT, UPDATE i DELETE).

És molt interessant que pot acceptar paràmetres. D'aquesta forma, la SQL no ha de ser exactament igual i tot continua funcionant. Exemples de PREPARE:

Per a un INSERT:

```
PREPARE fooplan (int, text, bool, numeric) AS
    INSERT INTO foo VALUES($1, $2, $3, $4);
EXECUTE fooplan(1, 'Hunter Valley', 't', 200.00);
```

Per a una SELECT:

```
PREPARE usrrptplan (int) AS
    SELECT * FROM users u, logs l WHERE u.usrid=$1 AND
    u.usrid=l.usrid
    AND l.date = $2;
EXECUTE usrrptplan(1, current_date);
```

Els paràmetres es quadren per ordre i s'especifiquen a la consulta i tenen la forma \$1, \$2, etc. Si no s'especifica el tipus (exemple del segon paràmetre de la SELECT), s'infereix de la pròpia base de dades.

Font:

<https://www.postgresql.org/docs/11/sql-prepare.html>

ON-COMMIT

Les taules temporals són taules on les dades poden persistir a dos nivells:

- Sessió. Quan es crea la sessió, la taula és buida. S'insereixen, eliminen, modifiquen i consulten dades independentment de les altres sessions. No es comparteixen les dades entre sessions. Quan la sessió finalitza, les dades desapareixen.
- Transacció. L'aïllament entre sessions és exactament el mateix que a nivell de sessió. La diferència és que cada vegada que una transacció finalitza (ja sigui amb un commit o amb un rollback), la taula es buida.

La clàusula ON COMMIT permet definir, a nivell de taula, si la persistència es produeix a nivell de sessió o de transacció. S'especifica quan es crea la taula i després es pot modificar mitjançant un alter.

Les opcions són:

- PRESERVE ROWS. Valor per defecte. Els registres no s'eliminen quan es fa un COMMIT o un ROLLBACK. Persistència a nivell de sessió.
- DELETE ROWS. Els registres s'eliminen quan es fa un COMMIT o un ROLLBACK. Persistència a nivell de transacció.
- DROP. No només s'eliminen els registres, sinó que s'elimina la taula sencera quan es fa un COMMIT o ROLLBACK. Persistència a nivell de transacció.

Exemple:

```
CREATE [ GLOBAL|LOCAL ] {TEMPORARY | TEMP } TABLE table_name
( column_name data_type,
  ...
  ...
  ... ) ON COMMIT DELETE ROWS;
```

Com a nota, dir que GLOBAL|LOCAL permet definir si la taula serà visible des de qualsevol sessió o només des de la sessió actual.

Font:

<https://www.postgresql.org/docs/11/sql-createtable.html>