

RÉCAPITULATIF (POSTGRE)SQL

↪ et quelques points auxquels faire attention

Types de données (postgres):

smallint, int, bigint
numeric(nb_chiffres, nb_apres_virgule)
real, double precision
char(n), varchar(n), text
date, timestamp,
serial...

↪ varient selon SGBD

Créer une table:

```
CREATE TABLE nom (  
  nom1 int,  
  nom2 text);
```

↪ tables pourraient déjà exister

Changer le schéma:

```
DROP TABLE t  
ALTER TABLE t ADD COLUMN a int  
ALTER TABLE t DROP COLUMN a
```

↪ clés: DROP TABLE t CASCADE

Clés:

```
ALTER TABLE t ADD CONSTRAINT nom_cle1  
PRIMARY KEY (a);  
  
ALTER TABLE t2 ADD CONSTRAINT nom_cle2  
FOREIGN KEY b REFERENCES t(a);  
  
ALTER TABLE t DROP CONSTRAINT nom_cle1;
```

↪ impact sur mises à jour (données & schéma)

↪ ordre des déclarations & mises à jour

Mises à jour:

```
INSERT INTO t VALUES (10, 3, 'aa');  
INSERT INTO t(a,b) VALUES (10,3);  
UPDATE t SET a = a+2 WHERE a < 5;  
DELETE FROM t WHERE a != 5;
```

↪ clés

Chargement de masse:

```
\copy t FROM '/home/fichier.csv' CSV HEADER  
DELIMITER '|';
```

```
\copy (select * from t where a=2) TO  
'/home/fichier.csv' CSV HEADER DELIMITER '|';
```

↪ séparateur, en-tête

“Schémas” de postgresQL:

```
DROP SCHEMA nom CASCADE  
CREATE SCHEMA nom;  
SET SEARCH_PATH TO nom;  
SET SEARCH_PATH TO nom1, nom2;
```

↪ search_path réinitialisé à chaque session

Requêtes simples:

```
SELECT * FROM t;  
  
SELECT DISTINCT a, b  
FROM t  
WHERE a IS NOT NULL;
```

```
SELECT LOWER(a), b+c  
FROM t  
WHERE a = 'Donnée' AND b != 3;
```

↪ sensible à la casse

Requêtes: opérations ensemblistes

```
SELECT a FROM t1 EXCEPT SELECT a FROM t2;  
↪ ensembliste. Pour conserver les doublons:  
SELECT a FROM t1 UNION ALL SELECT a FROM t2
```

Requêtes: jointures, agrégats

```
SELECT a,b  
FROM t1  
JOIN t2 ON t1.c=t2.c  
JOIN t3 ON t1.d=t3.d AND t2.e=t3.e;
```

```
SELECT a, SUM(b) nom_alias,  
       COUNT(DISTINCT b) alias_bis  
FROM t1, t2  
WHERE t1.c = t2.c AND a < 4  
GROUP BY a  
HAVING sum(t2.d) > 3  
ORDER BY a DESC;
```

↪ colonnes ambiguës

↪ attributs non-agrégés doivent apparaître dans le groupe

Requêtes: sous-requêtes

```
SELECT t1.a,  
       CASE WHEN bb < 4 OR bb > 10  
       THEN c  
       ELSE 2*bb END n_alias  
FROM t t1, (SELECT a, b bb FROM t) t2  
WHERE t1.c = t2.c
```

Requêtes: jointures externes

```
SELECT * FROM t LEFT OUTER JOIN t2 ON t.a=t2.b;
```