

TP PostgreSQL (1)

Ouvrir une session postgresQL :

- Interface web : <https://dbserver.emi.u-bordeaux1.fr>
- En ligne de commande :

```
psql -h dbserver.emi.u-bordeaux1.fr -U <login> <nom de la base>
```

- Commandes accessibles avec \<commande>

```
\? : aide, \q : quitter
```

```
\i script.sql : exécuter le fichier script script.sql
```

```
\d : lister les tables, \d nomtable : afficher sa structure
```

Exercice 1 (gestion d'un annuaire d'amis)

Ouvrir une session postgresQL et exécuter les scripts copains_creation.sql et copains_insertion.sql suivants :

copains_creation.sql :

```
CREATE TABLE annu(  
    Num INTEGER,  
    nom CHAR(15),  
    prenom CHAR(20),  
    cp INTEGER,  
    ville CHAR(20)  
);
```

```
CREATE TABLE films(  
    fNum INTEGER,  
    fNom CHAR(15),  
    notes INTEGER  
);
```

```
CREATE TABLE vues(  
    Num INTEGER,  
    fNum INTEGER  
);
```

copains_insertion.sql :

```
INSERT INTO annu VALUES (  
    1,  
    'Dupont',  
    'Eric',  
    75016,  
    'Paris'  
);
```

```
INSERT INTO annu VALUES (2,'Durant','',' 0,'Brest');
INSERT INTO annu VALUES (3,'Dulac','Michel',
68000,'Colmar');
INSERT INTO annu VALUES (4,'Legrand','',' 75001,'Paris');
INSERT INTO annu VALUES (5,'Lavigne','Eric', 6000,'Nice');
INSERT INTO annu VALUES (6,'Sage','Martine',
95800,'Cergy');

INSERT INTO films VALUES (1,'Moulin rouge',10);
INSERT INTO films VALUES (2,'IA',5);
INSERT INTO films VALUES (3,'Titanic',7);

INSERT INTO vues VALUES (1,1);
INSERT INTO vues VALUES (1,3);
INSERT INTO vues VALUES (2,1);
INSERT INTO vues VALUES (4,3);
```

Saisir les requêtes suivantes et analyser le résultat :

```
SELECT
  SELECT * FROM annu;

  SELECT nom, prenom, ville FROM annu;

  SELECT nom, prenom, ville FROM annu WHERE cp>68000;
```

Jointures

```
SELECT annu.Nom, films.fNom
FROM annu, films, vues
WHERE vues.num=annu.num and vues.fnum=films.fnum;
```

Agrégats

```
SELECT AVG(Note) AS note_moyenne FROM films;

SELECT C.nom, F.fnom
FROM annu C, films F, vues V
WHERE V.num=C.num and V.num=F.num;
```

Groupes

```
SELECT F.fnom,
count(V.fnum) AS nbr_vues
from vues V, films F
where V.Fnum=F.Fnum
group by F.fNom;
```

Vues

```
CREATE VIEW filmsvus AS
SELECT F.fnom,
count (V.Fnum) AS nbr_vues
from vues V, films F
where V.fnum=F.fnum
```

```
group by F.fnom;
```

```
SELECT * FROM filmsvus where 2=nbr_vues;
```

Redirection

```
\o resultat.txt  
SELECT * FROM annu;  
\o
```

Clés primaires et externes

Redéfinir le schéma de la base de données en incluant cette fois des clés primaires et externes.