

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

Base de données relationnelle

Qu'est-ce donc ?

Comment la représenter ?

SQL

Créer une base de données, une table

Insérer, Rechercher, Modifier

Postgresql

Installation

Fonctions et Trigger

Transaction

API : SQLOBJECT

pyPgSQL ou psycopg2

Objets API

Qu'est-ce donc ?

En 1970, Edgar Frank Codd publia un article où il proposait de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles.



Comment l'utiliser ?

Créer une base de données

Definition

```
CREATE DATABASE name  
[ WITH [ OWNER [=] dbowner ]  
TEMPLATE [=template ]  
ENCODING [=encoding ]  
... ]
```

Definition

```
createdb [ option... ] [ dbname ] [ description ]  
-E –encoding Encoding  
-O –owner owner
```

Créer une table

Definition

```
CREATE TABLE table_name ( [  
  {column_name data_type [ DEFAULT default_expr ] [  
  column_constraint [ ... ] ]}
```

...

```
]
```

where column_constraint is :

```
{ NOT NULL ||
```

```
NULL ||
```

```
UNIQUE index_parameters ||
```

```
PRIMARY KEY index_parameters ||
```

```
CHECK ( expression ) ||
```

```
REFERENCES reftable }
```

Créer une table

Example

```
CREATE TABLE machines (  
  id SERIAL PRIMARY KEY,  
  nom varchar(255) NOT NULL UNIQUE,  
  ip inet NOT NULL UNIQUE,  
  mac macaddr NOT NULL UNIQUE,  
  proprietaire integer NOT NULL REFERENCES adherents  
);
```


Insérer

Example

```
INSERT INTO machines (ip,mac,nom,proprietaire)  
VALUES ('138.231.136.34','00 :00 :ca :4e :5f :ac','toto',243) ;
```

Rechercher

Example

```
SELECT * FROM machines WHERE nom * '$a';
```

Example

```
SELECT a.nom, a.prenom FROM adherents AS a, machines  
AS m  
WHERE m.proprietaire = a.id AND m.ip='138.231.136.1';
```

Example

```
SELECT * FROM machines WHERE mac << inet  
'138.231.136/24';
```

Modifier

Example

```
UPDATE machines SET nom = 'tata' WHERE nom = 'toto' ;
```

Installation

- Utiliser le paquet
- Créer un autre utilisateur que postgres, createuser
- Créer base de donnée,...
- Utiliser psql

Trigger

Execution d'une fonction lors de certaine opération sur une table :

Definition

```
CREATE TRIGGER name BEFORE — AFTER event [ OR ... ]  
ON table [ FOR [ EACH ] ROW — STATEMENT ]  
EXECUTE PROCEDURE funcname ( arguments )
```

Trigger

Execution d'une fonction lors de certaine opération sur une table :

Definition

```
CREATE TRIGGER name BEFORE — AFTER event [ OR ... ]  
ON table [ FOR [ EACH ] ROW — STATEMENT ]  
EXECUTE PROCEDURE funcname ( arguments )
```

- BEFORE : avant les checks, et l'action désignée

Trigger

Execution d'une fonction lors de certaine opération sur une table :

Definition

```
CREATE TRIGGER name BEFORE — AFTER event [ OR ... ]  
ON table [ FOR [ EACH ] ROW — STATEMENT ]  
EXECUTE PROCEDURE funcname ( arguments )
```

- BEFORE : avant les checks, et l'action désignée
- AFTER : apres les modifications qui sont alors visible

Trigger

Execution d'une fonction lors de certaine opération sur une table :

Definition

```
CREATE TRIGGER name BEFORE — AFTER event [ OR ... ]  
ON table [ FOR [ EACH ] ROW — STATEMENT ]  
EXECUTE PROCEDURE funcname ( arguments )
```

- BEFORE : avant les checks, et l'action désignée
- AFTER : apres les modifications qui sont alors visible
- FOR EACH ROW : une requete avec 4 lignes impliquées execute 3 fois la fonctions

Trigger

Execution d'une fonction lors de certaine opération sur une table :

Definition

```
CREATE TRIGGER name BEFORE — AFTER event [ OR ... ]  
ON table [ FOR [ EACH ] ROW — STATEMENT ]  
EXECUTE PROCEDURE funcname ( arguments )
```

- BEFORE : avant les checks, et l'action désignée
- AFTER : apres les modifications qui sont alors visible
- FOR EACH ROW : une requete avec 4 lignes impliquées execute 3 fois la fonctions
- FOR EACH STATEMENT : une requete par requête, même si la table n'a pas été touché.

PL/SQL

Example

```

CREATE OR REPLACE FUNCTION aftertransaction()
RETURNS trigger AS '
DECLARE
BEGIN
IF NEW.valide THEN
PERFORM updatesoldes(NEW.emetteur, NEW.destinataire,
NEW.conso, NEW.montant, NEW.quantite);
ELSE
PERFORM updatesoldes(NEW.emetteur, NEW.destinataire,
NEW.conso, NEW.montant, -NEW.quantite);
END IF;
RETURN NEW;
END ' LANGUAGE 'plpgsql';
CREATE TRIGGER
aftertransactionAFTERUPDATEONtransactions

```

transaction

pyPgSQL ou psycopg2

- Gèrent l'accès à la base de données.
- Lèvent les exceptions adéquates.

pyPgSQL ou psycopg2

- Gèrent l'accès à la base de données.
- Lèvent les exceptions adéquates.
- Protègent de l'injection de code sql (sanitize).

pyPgSQL ou psycopg2

- Gèrent l'accès à la base de données.
- Lèvent les exceptions adéquates.
- Protègent de l'injection de code sql (sanitize).
- Convertissent les types python en sql et inversement.

Objets API

Donnent une interface objet à la base de données :

Objets API

Donnent une interface objet à la base de données :

- Les tables sont des classes.
- Les lignes sont des instances.

Objets API

Donnent une interface objet à la base de données :

- Les tables sont des classes.
- Les lignes sont des instances.
- Forment les requêtes sql de recherche.

SQLObject

- Construit la base de donnée à partir de la définition des classes.

SQLObject

- Construit la base de donnée à partir de la définition des classes.
- Gère les relations 1-1, 1-n, n-n.

SQLObject

- Construit la base de donnée à partir de la définition des classes.
- Gère les relations 1-1, 1-n, n-n.
- La recherche est assez puissante pour faire des jointures, limit, offset, orderby, distinct,...

SQLObject

- Construit la base de donnée à partir de la définition des classes.
- Gère les relations 1-1, 1-n, n-n.
- La recherche est assez puissante pour faire des jointures, limit, offset, orderby, distinct,...
- Cache ou pas

SQLObject

- Construit la base de donnée à partir de la définition des classes.
- Gère les relations 1-1, 1-n, n-n.
- La recherche est assez puissante pour faire des jointures, limit, offset, orderby, distinct,...
- Cache ou pas
- Modifie la BDD à la volée ou pas

SQLObject

- Construit la base de donnée à partir de la définition des classes.
- Gère les relations 1-1, 1-n, n-n.
- La recherche est assez puissante pour faire des jointures, limit, offset, orderby, distinct,...
- Cache ou pas
- Modifie la BDD à la volée ou pas
- Semble aisément modulable.