

Fonctionnement des tables

-- La table "members" :

```
CREATE TABLE members ( // regroupe les informations sur chaque membre
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, // id d'un membre
pseudo VARCHAR(25) NOT NULL, // pseudo d'un membre
mail VARCHAR(25) NOT NULL, // mail d'un membre
mdp CHAR(32) NOT NULL); // mot de passe d'un membre
```

-- La table "surveys" :

```
CREATE TABLE surveys ( // regroupe les informations sur chaque sondage
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, // id d'un sondage
owner INT NOT NULL, // id du membre propriétaire
title VARCHAR(100) NOT NULL, // titre d'un sondage
status INT NOT NULL, // si le sondage est en cours (0) ou terminé (1)
FOREIGN KEY(owner) REFERENCES members(id)); // reference owner
```

-- La table "questions" :

```
CREATE TABLE questions ( // regroupe les informations sur chaque question
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, // id d'une question
poll INT NOT NULL, // id du sondage associé à la question
question VARCHAR(255) NOT NULL, // la question posé
mandatory BOOLEAN, // si la question est obligatoire (TRUE) ou non (FALSE)
type INT NOT NULL, // si sa réponse est unique (0), multiple (1) ou textuel (2)
FOREIGN KEY(poll) REFERENCES surveys(id)); // reference poll
```

-- La table "answers" :

```
CREATE TABLE answers ( // regroupe les informations sur chacune des réponses
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, // id d'une réponse
choice INT NOT NULL, // id de la question associée à la réponse
answer VARCHAR(255) NOT NULL, // une réponse possible
FOREIGN KEY(choice) REFERENCES questions(id)); // reference choice
```

-- La table "analyses" :

```
CREATE TABLE analyses ( // regroupe les informations sur les réponses aux questionnaires soumis
respondent INT NOT NULL, // détermine l'id de celui qui répond
creator INT NOT NULL, // détermine l'id du créateur du sondage
survey INT NOT NULL, // détermine l'id du questionnaire
question INT NOT NULL, // détermine l'id du questionnaire
answer VARCHAR(255) NOT NULL, // indique le nom de la réponse
FOREIGN KEY(respondent) REFERENCES members(id), // reference respondent
FOREIGN KEY(creator) REFERENCES surveys(owner), // reference creator
FOREIGN KEY(survey) REFERENCES surveys(id), // reference survey
FOREIGN KEY(question) REFERENCES questions(id)); // reference question
```

COMMANDES D'INFORMATIONS :

--list of all tables and views

```
show tables;
```

--list of all tables and views with tables types

```
show full tables;
```

--description d'une table

```
describe membre;
```

--voir tous les utilisateurs

```
select * from members;
```

--voir tous les sondages

```
select * from surveys;
```

--voir toutes les questions

```
select * from questions;
```

COMMANDE DE MANAGEMENT :

--ajouter une colonne :

```
ALTER TABLE nom_table
```

```
ADD nom_colonne type_donnees
```

--supprimer une colonne :

```
ALTER TABLE nom_table
```

```
DROP nom_colonne
```

--modifier une colonne :

```
ALTER TABLE nom_table
```

```
MODIFY nom_colonne type_donnees
```

--renommer une colonne :

```
ALTER TABLE nom_table
```

```
CHANGE colonne_ancien_nom colonne_nouveau_nom type_donnees
```

--ajouter des valeurs dans une ligne:

```
INSERT INTO nom_table (nom_colonne_1, nom_colonne_2, ...)
```

```
VALUES ('valeur 1', 'valeur 2', ...)
```

--mettre à jour les valeurs d'une ligne :

```
UPDATE table_name
```

```
SET column1 = value1, column2 = value2, ...
```

```
WHERE condition;
```

--supprimer une ligne :

```
DELETE FROM nom_table WHERE condition;
```

--supprimer une table :

```
DROP TABLE nom_table;
```

--supprimer une base de données :

```
DROP DATABASE nom_base ;
```

--reset l'auto increment :

```
SELECT MAX( `id` ) FROM `members` ; -- sélectionne l'id_max
```

```
ALTER TABLE `members` AUTO_INCREMENT = id_max; -- // réinitialisation = 0
```