Bases de données relationnelles

Les bases du SQL avec SQLite

par Nicolas Hurtubise

Bases de données

Objectifs

- Stocker des données organisées
- Relations entre les données

Éléments de notre base de données

- Locaux
- Utilisateurs
- Réservations

Schéma de notre base de données

- Local
  - Numéro (ex.: 2165)
  - Catégorie (soit Cours, soit Réunion)
- Utilisateur
  - Infos de login/password
  - Nom
  - Permissions
  - ...
- Réservation
  - Raison (ex.: Meeting du *MILA*)
  - Date/heure/durée
  - Local réservé **(relation)**
  - Utilisateur responsable **(relation)**
  - Cours qui fait la réservation si applicable (ex.: IFT1015)

Schéma SQL (SQLite)

Locaux

```
CREATE TABLE `locaux` (
  `_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  `numero` TEXT NOT NULL,
  `categorie` TEXT NOT NULL,
  CHECK(`categorie` in ('reunion', 'cours'))
);
```

Schéma SQL (SQLite)

Utilisateurs

```
CREATE TABLE `users` (
  `_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  `login` TEXT NOT NULL,
  `password` TEXT NOT NULL,
  `name` TEXT NOT NULL,
  `access_level` TEXT NOT NULL,
  -- date du dernier login
  `last_login` INTEGER DEFAULT NULL,
  CHECK(`access_level` in ('admin', 'user'))
);
CREATE UNIQUE INDEX `unique_login`
    ON `users`(`login`);
```

Schéma SQL (SQLite)

Réservations

```
CREATE TABLE `reservations` (
  `_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  -- id du local réservé
  `local_id` INTEGER NOT NULL,
  -- id du user qui réserve
  `user_id` INTEGER NOT NULL,
  -- Penser à l'autocomplete pour éviter de
  -- retaper la raison à chaque fois !
  `raison` TEXT NOT NULL,
  `debut` INTEGER NOT NULL,
  -- Durée en secondes (simplifie les calculs)
  `duration` INTEGER NOT NULL,
  `cours` TEXT NOT NULL -- Sigle du cours
);
```

Créer la base de données


```
$ adb shell
$ cd /sdcard/data
$ sqlite3
sqlite> .read db-sqlite.sql
```

Opérations - SELECT

Requête de base

```
SELECT login, name FROM users;

-- pour toutes les colonnes :

SELECT * FROM users;
```

Opérations - SELECT

Filtrer : WHERE condition [AND/OR conditions ...]

```
SELECT * FROM users WHERE _id=1;

SELECT * FROM users
WHERE login="admin" AND password="$up3rP4$$";

SELECT * FROM reservations WHERE local_id=1;

SELECT * FROM reservations
WHERE local_id=1 OR user_id=1;
```

Opérations - SELECT

ORDER BY col sens

```
SELECT * FROM reservations ORDER BY debut;

SELECT * FROM reservations ORDER BY debut ASC;

SELECT local_id FROM reservations
ORDER BY duration DESC;
```

Opérations - SELECT

LIMIT nbr & OFFSET nbr

Utile pour de la pagination :

```sql
-- Trouver les 5 premiers numéros de locaux
SELECT numero FROM locaux ORDER BY numero LIMIT 5;

-- Trouver les 5 suivants
SELECT numero FROM locaux
ORDER BY numero LIMIT 5 OFFSET 5;
```

Opérations - SELECT

JOIN table ON condition

```
SELECT * FROM reservations
JOIN locaux ON locaux._id=reservations.local_id;
```

Opérations - INSERT

Format

```
INSERT INTO table(col1, col2, ...)
VALUES (row1_val1, row1_val2, ...),
       (row2_val1, row2_val2, ...);
```

Ajouter des utilisateurs à notre BD

```
INSERT INTO
    `users`(`login`, `password`,
            `name`, `access_level`)
VALUES
    ("test", "1234", "Nicolas", "user");
```

Opérations - UPDATE


```sql
UPDATE users SET password="newpass" WHERE _id=1;

UPDATE users
SET password="newpass", name="John Johnson"
WHERE _id=1;
```

Opérations - DELETE


```sql
DELETE FROM users WHERE _id=2;
```

Fonctions et opérations

Utiliser une fonction

```sql
SELECT name, length(name) AS longueur_nom
FROM users;

SELECT name, 1 + 1 AS deux
FROM users;

SELECT * FROM users
WHERE length(name) <= 10;
```

Dates

- Pour SQLite, le type de données `Date` n'existe pas
- SQLite ne propose que `INTEGER`, `REAL`, `TEXT` et `BLOB`
- On peut noter le `Timestamp`[1] utiliser les fonctions de date pour faire la conversion en format textuel

[1] Nombre de secondes écoulées depuis 1er janvier 1970

Dates


```sql
SELECT datetime(debut, 'unixepoch') AS start
FROM reservations;

SELECT _id,
    datetime(debut, 'unixepoch') AS debut,
    datetime(debut + duration, 'unixepoch') AS fin
FROM reservations
WHERE local_id=2;
```