

Fiche à étudier — Définir et modifier ses données

Tout ce qu'il faut savoir par cœur sur CREATE, ALTER, DROP, INSERT, UPDATE et DELETE : vocabulaire, syntaxe, contraintes, bons réflexes et pièges, avec des questions pour te tester.

4TTR

5TTR

6TTR

 Synthèse

Cette fiche rassemble ce qu'il faut retenir pour créer une table et écrire dans ses lignes, avec SQLite et DB Browser for SQLite.

Le vocabulaire

Mot	Définition	En an
DDL	Les instructions qui définissent la structure de la base : <code>CREATE TABLE</code> , <code>ALTER TABLE</code> , <code>DROP TABLE</code> .	Data I
DML	Les instructions qui travaillent sur les lignes : <code>SELECT</code> , <code>INSERT</code> , <code>UPDATE</code> , <code>DELETE</code> .	Data I
Schéma	La structure d'une base : ses tables, leurs colonnes, leurs types et leurs contraintes, sans aucune donnée.	scher
Clé primaire	La colonne qui désigne une ligne et une seule : jamais deux fois la même valeur, jamais vide, une seule par table.	prima
NULL	L'absence de valeur : ni un texte vide, ni zéro, mais une case que personne n'a remplie.	null
Contrainte	Une règle écrite dans la structure d'une table, que la base vérifie à chaque ajout ou modification.	const
NOT NULL	La contrainte qui rend une colonne obligatoire.	not nu
UNIQUE	La contrainte qui interdit les doublons dans une colonne.	uniqu
Valeur par défaut	La valeur que la base écrit elle-même quand on ajoute une ligne sans remplir la colonne.	defau
ALTER TABLE	L'instruction qui modifie la structure d'une table existante sans effacer ses lignes.	alter
Script SQL	Un fichier texte <code>.sql</code> qui contient une suite d'instructions SQL qu'on peut rejouer.	SQL s
INSERT	L'instruction qui ajoute une ligne dans une table.	insert
INSERT ... SELECT	L'instruction qui ajoute dans une table les lignes renvoyées par un <code>SELECT</code> .	insert
Fichier CSV	Un fichier texte qui contient un tableau, avec les valeurs séparées par un caractère fixe (, ou ;).	Comn
UPDATE	L'instruction qui modifie des lignes existantes.	updat
Nombre de lignes affectées	Le nombre de lignes réellement touchées par un <code>INSERT</code> , un <code>UPDATE</code> ou un <code>DELETE</code> .	rows :
Écrire / Annuler les modifications	Dans DB Browser : enregistrer dans le fichier les changements en attente, ou les abandonner.	Write
DELETE	L'instruction qui supprime des lignes entières d'une table.	delete
Sauvegarde	Une copie de la base, rangée à part, faite avant une opération risquée.	backu
Export SQL	La transformation d'une base en script SQL (<code>CREATE TABLE</code> + <code>INSERT</code>) capable de la reconstruire.	expor

La syntaxe

Créer une table

```
1 CREATE TABLE professeur (  
2   id      INTEGER PRIMARY KEY,  
3   nom     TEXT NOT NULL,  
4   prenom  TEXT NOT NULL,  
5   email   TEXT NOT NULL UNIQUE,  
6   actif   INTEGER NOT NULL DEFAULT 1,  
7   telephone TEXT  
8 );
```

Chaque colonne : `nom TYPE contraintes` . Une virgule entre les colonnes, jamais après la dernière. Variante sans erreur si la table existe : `CREATE TABLE IF NOT EXISTS` .

Type	Pour quoi
TEXT	des mots, des codes, des téléphones, des dates : AAAA-MM-JJ
INTEGER	des nombres entiers avec lesquels on calcule ou qu'on compare
REAL	des nombres à décimales, écrits avec un point : 13.5

Modifier une table

```

1 ALTER TABLE professeur ADD COLUMN telephone TEXT;
2 ALTER TABLE professeur RENAME COLUMN telephone TO gsm;
3 ALTER TABLE professeur RENAME TO enseignant;
```

Supprimer une table

```

1 DROP TABLE IF EXISTS import_prof;
```

Ajouter des lignes

```

1 INSERT INTO professeur (nom, prenom, email)
2 VALUES ('Leroy', 'Anne', 'a.leroy@ecole.be');
3
4 INSERT INTO professeur (nom, prenom, email)
5 VALUES
6     ('Martin', 'Julie', 'j.martin@ecole.be'),
7     ('Renard', 'Paul', 'p.renard@ecole.be');
```

Copier les lignes d'une table vers une autre

```

1 INSERT INTO professeur (nom, prenom, email)
2 SELECT nom, prenom, email
3 FROM import_prof;
```

Modifier des lignes

```

1 UPDATE eleve
2 SET absences = absences + 1
3 WHERE id = 9;
4
5 UPDATE professeur
6 SET actif = 0,
7     telephone = NULL
8 WHERE id = 6;
```

Supprimer des lignes

```

1 DELETE FROM professeur
2 WHERE id = 7;
```

Les contraintes

Contrainte	Effet	Valeur vide (NULL) acceptée ?	Message quand elle refuse
PRIMARY KEY	désigne la ligne ; avec INTEGER , se remplit toute seule	non	UNIQUE constraint failed: table.id
NOT NULL	la colonne est obligatoire	non	NOT NULL constraint failed: table.colonne
UNIQUE	pas deux fois la même valeur	oui, et même plusieurs fois	UNIQUE constraint failed: table.colonne
DEFAULT valeur	valeur écrite quand on ne remplit pas la colonne	ne refuse rien	—

Une colonne qu'on ne cite pas dans un INSERT reçoit sa valeur DEFAULT si elle en a une, sinon NULL . Une ligne refusée n'est pas ajoutée, même en partie.

Les bons réflexes

- Nommer les colonnes dans chaque `INSERT`.
- Un `SELECT` après chaque écriture, pour voir ce que la table contient vraiment.
- Le `SELECT` d'abord avant un `UPDATE` ou un `DELETE` : même `WHERE`, on compte les lignes, puis on remplace le début de l'instruction.
- Viser la clé primaire (`WHERE id = ...`) pour modifier ou supprimer une ligne précise.
- Annoncer le nombre de lignes avant d'exécuter, et le comparer avec le nombre annoncé par DB Browser.
- Sauvegarder avant une opération risquée : *Fichier > Exporter > Base de données vers fichier SQL...*, et vérifier que la sauvegarde se restaure.
- Écrire les modifications (`Ctrl+S`) seulement quand le résultat est vérifié ; sinon, **Annuler les modifications**.
- Lire le message d'erreur : la contrainte, puis `table.colonne` en cause.

Les pièges à connaître

- Une virgule après la dernière colonne d'un `CREATE TABLE` : `near ")": syntax error`.
- Un téléphone en `INTEGER` perd son zéro du début : un numéro est du `TEXT`.
- `UNIQUE` accepte plusieurs `NULL` : pour « obligatoire et sans doublon », il faut `NOT NULL UNIQUE`.
- `DROP TABLE` et un script qui commence par `DROP TABLE IF EXISTS` détruisent toutes les lignes.
- `ALTER TABLE` de SQLite ne change ni le type ni les contraintes d'une colonne existante, et refuse d'ajouter une colonne `NOT NULL` sans `DEFAULT` à une table qui contient des lignes.
- Un `INSERT` sans liste de colonnes dépend de l'ordre des colonnes de la table.
- Un texte sans apostrophes est pris pour un nom de colonne : `no such column: ...`. Une apostrophe dans un texte se double : `'D''Hondt'`.
- Un `UPDATE` ou un `DELETE` sans `WHERE` touche toutes les lignes, sans confirmation.
- Dans un `SET`, les colonnes se séparent par des virgules : `AND` ne provoque pas d'erreur, mais donne un résultat faux.
- Zéro ligne affectée n'est pas une erreur : c'est la condition qui ne trouve rien.
- `DELETE` vide des lignes, `DROP TABLE` supprime la table.
- Une base n'a pas de corbeille : une fois les modifications écrites, seule une sauvegarde permet de revenir en arrière.
- Les trous dans les `id` sont normaux : on ne renumérote jamais.

Teste-toi

1. Range ces instructions en DDL ou en DML : `SELECT`, `CREATE TABLE`, `UPDATE`, `DROP TABLE`, `INSERT`, `ALTER TABLE`, `DELETE`.
2. Quel type choisis-tu pour un code postal, et pourquoi ?
3. Trouve l'erreur : `CREATE TABLE local (code TEXT, etage INTEGER,);`
4. Quelle différence y a-t-il entre `NULL` et le texte vide `' '` ?
5. Une colonne `email TEXT UNIQUE` peut-elle contenir deux lignes sans adresse ? Que faut-il ajouter pour l'interdire ?
6. Pourquoi n'écrit-on pas `AUTOINCREMENT` avec SQLite ?
7. Tu dois ajouter une colonne `telephone` à une table qui contient déjà 200 lignes. Quelle instruction utilises-tu, et pourquoi pas `DROP TABLE` suivi de `CREATE TABLE` ?
8. Après `INSERT INTO professeur (nom, prenom, email) VALUES ('Leroy', 'Anne', 'a.leroy@ecole.be');`, que valent `actif` et `telephone` ?
9. Que signifie le message `UNIQUE constraint failed: professeur.email` ?
10. Quelles sont les quatre étapes pour importer un fichier CSV dans une table qui n'a pas les mêmes colonnes ?
11. Décris la méthode « le `SELECT` d'abord » avant un `UPDATE`.
12. Pourquoi vise-t-on `WHERE id = 12` plutôt que `WHERE nom = 'Lambert'` ?
13. Tu as exécuté un `UPDATE` sans `WHERE`, sans écrire les modifications. Que fais-tu ?
14. Quelle différence y a-t-il entre `DELETE FROM professeur;` et `DROP TABLE professeur;` ?
15. Après suppression de la ligne 7, un collègue veut renuméroter les `id` pour boucher le trou. Que lui réponds-tu ?

► Réponses