

Une entité devient une table

Le directeur d'une école veut suivre ses élèves. Tu pars de sa demande, tu choisis comment ranger chaque information, puis tu crées une vraie table dans une vraie base de données avec DB Browser for SQLite.

4TTR

5TTR

6TTR



Découverte

Un schéma sur papier ne suffit pas pour retrouver des informations : il faut les ranger quelque part. Dans cet article, tu passes de la demande d'un directeur d'école à une table remplie, enregistrée dans un fichier que tu pourras interroger.



Installe **DB BROWSER FOR SQLITE** (lien dans la sidebar). C'est un logiciel libre et gratuit, disponible pour Windows, macOS et Linux.



Objectifs

À la fin de cet article, tu seras capable de :

1. Traduire une entité en table : chaque attribut devient une colonne, chaque occurrence une ligne.
2. Choisir un type de données adapté à chaque colonne.
3. Écrire la structure d'une table en notation MLD.
4. Créer une base de données et une table avec DB Browser for SQLite.
5. Saisir des lignes et enregistrer les modifications.
6. Reconnaître le SQL que DB Browser écrit à ta place.

Partir de la demande

Le directeur d'une école t'explique ce qu'il attend :

« J'aimerais un outil pour suivre mes élèves. Pour chacun, je veux connaître son nom, son prénom, sa date de naissance et la commune où il habite. S'il a une adresse e-mail, je veux la noter aussi. Et j'aimerais savoir combien de demi-jours d'absence il a cette année. »

Avant d'ouvrir un logiciel, on analyse ce texte. Le directeur veut garder **une liste d'élèves** : c'est une **entité**, c'est-à-dire une catégorie de choses dont on garde une liste. Pour chaque élève, il veut connaître six informations : ce sont ses **attributs**.



Lucas Adam, né le 12 mars 2010, qui habite à Jodoigne, est un élève bien réel : c'est une **occurrence** de l'entité **ELEVE**.

Du modèle à la table

Le rectangle ci-dessus décrit ce qu'on veut retenir. Pour le ranger dans un ordinateur, on le transforme en tableau.

nom	prenom	date_naissance	commune	email	absences
Adam	Lucas	2010-03-12	Jodoigne	lucas.adam@ecole.be	2
Bastin	Emma	2010-07-04	Wavre	emma.bastin@ecole.be	0
Charlier	Noah	2009-11-23	Perwez		5

NOUVELLE NOTION : LA TABLE

Une **table** range toutes les occurrences d'une entité sous forme de tableau. Elle porte un nom, ses colonnes portent un nom, et chaque ligne décrit un élément.

En anglais : GB **table**.

Le passage du modèle à la table suit toujours la même règle :

 **UNE ENTITÉ DEVIENT UNE TABLE. UN ATTRIBUT DEVIENT UNE COLONNE. UNE OCCURRENCE DEVIENT UNE LIGNE.**

Dans le modèle (MCD)	Dans la base de données	Exemple
entité	table	eleve
attribut	colonne	prenom
occurrence	ligne	Lucas Adam

Le logiciel qui garde la table : le SGBD

Une table ne vit pas dans un tableur. Elle est confiée à un logiciel spécialisé, qui la range, la protège et répond aux questions qu'on lui pose.

NOUVELLE NOTION : LE SGBD

Un **SGBD**, ou **système de gestion de base de données**, est le logiciel qui stocke les données et les met à disposition. Quand il range les données dans des tables, on parle de **SGBDR** : système de gestion de base de données **relationnelle**.

En anglais : GB **Database Management System** (DBMS).

Dans ce parcours, le SGBDR s'appelle **SQLite**. Sa particularité : toute la base de données tient dans **un seul fichier**, qui porte l'extension **.db**. On peut le copier, l'envoyer par mail ou le sauvegarder sur une clé USB comme n'importe quel fichier.

DB Browser for SQLite n'est pas le SGBD lui-même. C'est une fenêtre qui permet de voir et de manipuler un fichier SQLite sans écrire de programme.

 D'autres SGBDR, comme MySQL ou PostgreSQL, fonctionnent autrement : la base vit dans un programme qui tourne en permanence sur un serveur, et on l'interroge à distance. Les tables, elles, s'utilisent de la même façon.

Choisir le type de chaque colonne

Avant de créer la table, SQLite veut savoir ce que chaque colonne va contenir : du texte ? un nombre ?

NOUVELLE NOTION : LE TYPE DE DONNÉES

Le **type de données** d'une colonne indique la nature des valeurs qu'elle contient : du texte, un nombre entier ou un nombre à virgule. Il aide le SGBD à ranger, trier et comparer correctement les valeurs.

SQLite propose surtout trois types :

Type	Traduction	Pour stocker	Exemples
TEXT	texte	des mots, des codes, des dates	Adam , B14 , 2010-03-12
INTEGER	entier	des nombres sans virgule	0 , 17 , 250
REAL	réel	des nombres à virgule	13.5 , 29.99

Pour notre table, le choix donne :

Colonne	Type	Pourquoi
nom	TEXT	du texte
prenom	TEXT	du texte
date_naissance	TEXT	une date écrite AAAA-MM-JJ (voir plus bas)
commune	TEXT	du texte
email	TEXT	du texte
absences	INTEGER	un nombre entier, sur lequel on fera des calculs

Les dates : du texte au format AAAA-MM-JJ

SQLite n'a pas de type réservé aux dates. On les range donc en **TEXT**, en respectant toujours le même format : l'année, puis le mois, puis le jour.

1 | 2010-03-12 et non 12/03/2010

Ce choix a un avantage énorme. Trie ces dates comme on trie des mots, caractère par caractère :

1 | 2009-11-23
2 | 2010-03-12
3 | 2010-07-04

Elles se retrouvent automatiquement dans l'ordre chronologique. Avec le format **JJ/MM/AAAA**, le tri commencerait par le jour, et **04/07/2010** passerait avant **23/11/2009**.

 **UNE DATE SE RANGE EN TEXT, TOUJOURS AU FORMAT AAAA-MM-JJ.**

Des chiffres qui ne sont pas des nombres

Un numéro de téléphone comme `0475123456` ne contient que des chiffres. Pourtant, on ne l'additionne jamais, et le `0` du début fait partie de l'information : en `INTEGER`, il disparaîtrait. Un téléphone, un code postal ou un numéro de local se rangent en `TEXT`.

💡 Pour choisir entre `TEXT` et `INTEGER`, demande-toi : **vais-je faire des calculs avec cette valeur ?** Si la réponse est non, c'est probablement du texte.

Une colonne pour désigner chaque ligne

Deux élèves peuvent porter le même nom et le même prénom. Pour les distinguer à coup sûr, on ajoute à la table une colonne `id` : un numéro unique pour chaque ligne, que SQLite attribue lui-même.

i Pourquoi un numéro plutôt que le nom ou l'adresse e-mail ? Et comment la base empêche-t-elle deux lignes d'avoir le même ? C'est tout le sujet de l'article [Identifier ses données : la clé primaire](#). Pour l'instant, retiens que chaque table de ce parcours commence par une colonne `id`.

Écrire la structure en une ligne : le MLD

Dessiner un tableau pour décrire une table prend de la place. Les informaticiens utilisent une écriture plus courte, sur une seule ligne.

📖 NOUVELLE NOTION : LE MLD

Le **MLD**, ou **modèle logique de données**, décrit les tables d'une base : pour chaque table, son nom suivi de la liste de ses colonnes entre parenthèses. La colonne qui identifie chaque ligne est **soulignée**.

En anglais : GB *logical data model*.

Notre table s'écrit :

eleve (id, nom, prenom, date_naissance, commune, email, absences)

Tu remarques que l'entité `ELEVE` du MCD devient la table `eleve` du MLD. Les noms suivent une convention précise, qu'on garde toute l'année :

Élément	Convention	Exemple
Entité (MCD)	singulier, MAJUSCULES	ELEVE
Table (MLD, base)	singulier, minuscules, sans accent	eleve
Colonne	minuscules, sans accent, mots séparés par _	date_naissance

⚠ Pas d'espace ni d'accent dans les noms : `date_naissance` et jamais `Date de naissance`. Les espaces et les accents compliquent l'écriture des requêtes et provoquent des erreurs difficiles à repérer.

Créer la base dans DB Browser

Place-toi devant ton ordinateur. Tu vas créer le fichier `mon-ecole.db` et sa première table.

1. Lance **DB Browser for SQLite**.
2. Clique sur **Nouvelle base de données**.
3. Choisis ton dossier de travail et enregistre le fichier sous le nom `mon-ecole.db`.
4. La fenêtre **Éditer la définition de la table** s'ouvre. Dans le champ du nom, écris `eleve`.
5. Clique sur **Ajouter** pour créer chaque colonne, puis règle son nom et son type :

Nom	Type	Cases à cocher
<code>id</code>	<code>INTEGER</code>	PK uniquement
<code>nom</code>	<code>TEXT</code>	aucune
<code>prenom</code>	<code>TEXT</code>	aucune
<code>date_naissance</code>	<code>TEXT</code>	aucune
<code>commune</code>	<code>TEXT</code>	aucune
<code>email</code>	<code>TEXT</code>	aucune
<code>absences</code>	<code>INTEGER</code>	aucune

⚠ Pour `id`, coche **PK** mais laisse **AI** décoché. En SQLite, une colonne `INTEGER` cochée PK se numérote déjà toute seule.

Ne clique pas encore sur OK : regarde d'abord le bas de la fenêtre.

Le SQL derrière le clic

Pendant que tu remplis la fenêtre, DB Browser écrit un texte dans la zone du bas. Il ressemble à ceci :

```

1 CREATE TABLE "eleve" (
2     "id" INTEGER,
3     "nom" TEXT,
4     "prenom" TEXT,
5     "date_naissance" TEXT,
6     "commune" TEXT,
7     "email" TEXT,
8     "absences" INTEGER,
9     PRIMARY KEY("id")
10 );

```

Ce texte est une instruction écrite dans le langage que comprennent les SGBDR.

NOUVELLE NOTION : LE SQL

Le **SQL** est le langage qui sert à parler à un SGBDR : créer des tables, y ranger des données, les modifier et poser des questions. Tous les SGBDR le comprennent, avec quelques petites différences.

En anglais : GB **Structured Query Language**, « langage de requêtes structuré ».

Tu n'as pas besoin de savoir l'écrire aujourd'hui, mais tu peux déjà le lire :

- `CREATE TABLE "eleve"` veut dire « crée la table `eleve` » ;
- chaque ligne suivante donne le nom d'une colonne et son type ;
- `PRIMARY KEY("id")` indique que `id` est la colonne qui identifie chaque ligne.

 **DB BROWSER TRADUIT TES CLICS EN SQL.** L'interface et le SQL produisent exactement la même table. L'article [Créer et modifier une table : CREATE et ALTER](#) t'apprend à l'écrire toi-même.

Clique maintenant sur **OK**. La table `eleve` apparaît dans l'onglet **Structure de la base**.

Saisir les premiers élèves

Une table vide ne sert à rien. Remplis-la.

1. Ouvre l'onglet **Parcourir les données** et choisis la table `eleve`.
2. Clique sur **Insérer un nouvel enregistrement**. Une ligne apparaît, et la colonne `id` reçoit automatiquement le numéro `1`.
3. Double-clique sur chaque case pour y écrire une valeur.

Encode ces trois élèves :

nom	prenom	date_naissance	commune	email	absences
Adam	Lucas	2010-03-12	Jodoigne	lucas.adam@ecole.be	2
Bastin	Emma	2010-07-04	Wavre	emma.bastin@ecole.be	0
Charlier	Noah	2009-11-23	Perwez	(ne rien écrire)	5

Noah Charlier n'a pas d'adresse e-mail. Ne touche pas à sa case : DB Browser y affiche `NULL`.

NOUVELLE NOTION : NULL

NULL signifie qu'une case ne contient **aucune valeur** : l'information est absente ou inconnue. Ce n'est ni un zéro, ni un texte vide.

En anglais : GB **null**, « nul, vide ».

Chaque ligne que tu viens d'encoder est une occurrence de l'entité `ELEVE`.

Enregistrer les modifications

Regarde la barre d'outils. Tant que tu n'as pas cliqué sur **Écrire les modifications**, rien n'est enregistré dans le fichier `mon-ecole.db`.

 **SAISIR UNE LIGNE ET L'ENREGISTRER SONT DEUX GESTES DIFFÉRENTS.** Si tu fermes DB Browser sans écrire les modifications, tu perds tout ce que tu as encodé.

Prends dès maintenant le réflexe : **une modification, puis Ctrl+S.**

Le bouton voisin, **Annuler les modifications**, fait l'inverse : il efface tout ce qui a été changé depuis le dernier enregistrement. Il te sauvera plus d'une fois.

Récupérer la base complète

L'école compte douze élèves. Plutôt que de tous les encoder, télécharge `mon-ecole-depart.db` dans la sidebar. Elle contient exactement la même table, avec les douze élèves :

```

1 | id | nom      | prenom | date_naissance | commune | email | absences
2 | ---+-----+-----+-----+-----+-----+-----+-----
```

3	1	Adam	Lucas	2010-03-12	Jodoigne	lucas.adam@ecole.be	2
4	2	Bastin	Emma	2010-07-04	Wavre	emma.bastin@ecole.be	0
5	3	Charlier	Noah	2009-11-23	Perwez	NULL	5
6	4	Delvaux	Léa	2010-04-15	Jodoigne	lea.delvaux@ecole.be	1
7	5	Englebert	Hugo	2010-01-30	Hannut	hugo.englebert@ecole.be	8
8	6	Fontaine	Chloé	2009-09-08	Jodoigne	NULL	0
9	7	Gérard	Nathan	2010-05-19	Wavre	nathan.gerard@ecole.be	3
10	8	Hubert	Manon	2010-02-27	Perwez	manon.hubert@ecole.be	0
11	9	Istas	Louis	2009-12-11	Jodoigne	louis.istas@ecole.be	12
12	10	Jacques	Camille	2010-06-02	Hannut	NULL	4
13	11	Kevers	Arthur	2009-10-17	Ramillies	arthur.kevers@ecole.be	1
14	12	Lambert	Sarah	2010-09-01	jodoigne	sarah.lambert@ecole.be	0

Renomme le fichier `mon-ecole.db` si tu veux continuer avec ce nom. Une base SQLite est un simple fichier : pense à en garder une copie après chaque séance.



Exercices

Exercice 1 – Choisir le bon type ★☆☆☆☆

Pour chaque information, choisis `TEXT`, `INTEGER` ou `REAL`, et justifie en quelques mots.

Information	Type	Pourquoi ?
prénom d'un client		
quantité en stock		
prix d'un produit		
adresse e-mail		
année de sortie d'un film		
code postal belge		
date d'inscription		
moyenne d'un bulletin		

Exercice 2 – Une date acceptée n'est pas une date correcte ★★☆☆☆

Dans `mon-ecole-depart.db`, onglet **Parcourir les données** :

1. Insère un nouvel élève de test, avec la date de naissance `15/05/2010`. DB Browser l'accepte-t-il ?
2. Clique sur le titre de la colonne `date_naissance` pour trier la table. Où se place ta ligne de test ?
3. Explique pourquoi SQLite accepte une valeur qui ne respecte pas le format décidé.

4. Clique sur **Annuler les modifications** et vérifie que ta ligne de test a disparu.

Exercice 3 – L'intrus ★★☆☆☆

Parcours la table complète des douze élèves.

1. Une valeur est écrite différemment des autres alors qu'elle désigne la même chose. Laquelle ?
2. Quel mot de vocabulaire désigne ce genre de problème ?
3. Pourquoi SQLite ne l'a-t-il pas signalé ?
4. Imagine le secrétariat qui cherche tous les élèves habitant cette commune. Que risque-t-il de se passer ?

Ne corrige pas la valeur : garde-la telle quelle, elle servira plus tard.

Exercice 4 – Le MLD de la vidéothèque ★★☆☆☆

« Une vidéothèque veut garder la liste de ses films. Pour chaque film, elle veut connaître le titre, le réalisateur, l'année de sortie, la durée en minutes et le prix de location. »

1. Dessine l'entité avec ses attributs.
2. Écris le MLD de la table correspondante, avec une colonne `id`.
3. Donne le type de chaque colonne.
4. Écris deux lignes d'exemple.

Exercice 5 – De la demande à la table ★★☆☆☆

Choisis un domaine que tu connais : tes jeux vidéo, les joueurs d'une équipe, tes livres...

1. Écris une demande de deux ou trois phrases, comme celle du directeur.
2. Repère l'entité et ses attributs, puis écris le MLD.
3. Crée une nouvelle base `essai.db` et la table correspondante dans DB Browser.
4. Avant de valider, recopie le SQL affiché en bas de la fenêtre et relie chaque ligne à une colonne de ton MLD.
5. Encode cinq lignes, puis écris les modifications.



À retenir

- Une entité devient une **table**, un attribut une **colonne**, une occurrence une **ligne**.
- Un **SGBD** stocke les données ; **SQLite** est un SGBDR dont la base tient dans un seul fichier `.db`.
- Chaque colonne a un **type** : `TEXT`, `INTEGER` ou `REAL`. Une date se range en `TEXT`, au format `AAAA-MM-JJ`.
- Le **MLD** décrit une table sur une ligne : `eleve (id, nom, prenom, ...)`, avec l'identifiant souligné.
- Table au singulier, en minuscules, sans accent ; colonnes sans espace ni accent.
- `NULL` signifie qu'une case ne contient aucune valeur.
- DB Browser traduit tes clics en **SQL**, et rien n'est enregistré avant **Écrire les modifications** (Ctrl+S).

Suite

Ta table a une colonne `id`, mais pourquoi un numéro plutôt que le nom ? L'article [Identifier ses données : la clé primaire](#) répond à cette question et montre comment la base refuse les doublons.