



COMPRENDRE · LE PRINCIPE

L'IA écrit la requête, la base calcule

Poser une question en français, examiner la requête, puis faire calculer une base locale : une méthode pour obtenir des résultats traçables à partir de fichiers CSV ou Excel.

Une réponse fluide peut masquer un mauvais filtre, une ligne oubliée ou une unité mal comprise. Le parcours sépare trois opérations : vous formulez la question, le modèle propose du SQL, puis SQLite exécute la requête que vous avez retenue. SQL est le langage qui décrit les lignes à sélectionner et les opérations à appliquer.

Qui intervient ?	Son rôle précis
Vous	Définir la question, examiner les filtres, contrôler le résultat.
Ollama et Qwen3 4B	Produire une proposition de requête à partir du dictionnaire des colonnes.
DB Browser et SQLite	Ouvrir la base, exécuter la requête, afficher ou exporter le résultat.

L'exécution d'une requête est déterministe sur une base donnée. Sa pertinence dépend toujours de la question, des données et des règles métier. Une addition exacte portant sur les mauvaises lignes reste une mauvaise réponse.

Le cas d'essai

Le kit contient 12 commandes fictives, réparties entre janvier, février et le 1er mars 2026. Une ligne représente une commande. Les montants sont hors taxes et stockés en centimes entiers. Une commande payée n'a pas de montant renseigné ; une autre vaut zéro. Ces cas servent à tester la rigueur du raisonnement.

Préparer son poste

Parcours Windows ou macOS, sans développement d'application. Il faut installer deux logiciels, copier des commandes et relire quelques requêtes simples. Prévoyez Internet pour les installations et le téléchargement du modèle. Qwen3 4B reprend le modèle de la fiche RAG ; ses performances dépendent du matériel et de la question. Aucun classement de modèles n'est revendiqué. [1, 4, 6]

Un fichier Excel doit d'abord être exporté en CSV. Ce tutoriel ne traite ni les macros, ni les formules du classeur, ni les cellules fusionnées. Conservez le fichier original et vérifiez les valeurs exportées.



ÉTAPE 1 / 5 · PRÉPARER

Définir ce que chaque ligne signifie

1. Extraire le kit et garder les originaux

Décompressez le kit dans un dossier local dédié. Gardez commandes.csv intact. Pour vos propres données, partez d'une copie et notez sa date d'extraction. Évitez les sous-totaux au milieu des lignes, les intitulés répétés et les colonnes sans unité.

2. Contrôler les formats

Champ	Règle dans le kit
id_commande	Identifiant unique ; une ligne par commande.
date_commande	AAAA-MM-JJ ; date de commande, distincte de la date de paiement.
ville et statut	Libellés homogènes ; payee, annulee, en_attente.
montant_ht_centimes	Entier ; 12000 signifie 120,00 € HT. Vide = inconnu ; 0 = zéro connu.
client_id	Un client peut revenir sur plusieurs commandes.

Les dates homogènes facilitent les filtres. Les centimes entiers évitent les approximations des nombres à virgule pendant les additions usuelles, tant que la somme reste dans la plage des entiers de SQLite. Ne convertissez en euros qu'à la présentation. Une moyenne peut nécessiter un arrondi explicite. [7, 8]

3. Formuler le périmètre avant le calcul

« Les ventes de février » reste ambigu : toutes les commandes, les commandes payées, ou les paiements effectivement reçus en février ? Notre table ne contient aucune date de paiement. La question correcte pour l'exercice est : « Quel montant HT représentent les commandes payées dont la date de commande est en février 2026 ? »

Le dictionnaire du kit fixe ces règles. Il ne remplace pas leur validation avec la personne responsable des données. Un montant absent ne se déduit ni du statut de paiement ni d'une moyenne.



ÉTAPE 2 / 5 · IMPORTER

Ouvrir une base locale et vérifier son contenu

4. Démarrer avec la base prête à ouvrir

Installez DB Browser for SQLite depuis son site officiel. Le parcours est fondé sur les menus de la version 3.13.1 ; les traductions peuvent varier. Dans File / Fichier, utilisez « Open Database Read Only... » / ouverture en lecture seule, puis sélectionnez commandes-demo.sqlite. Dans « Browse Data », choisissez la table commandes. [1, 2]

Dans l'onglet « Execute SQL », collez la requête ci-dessous et lancez-la avec la commande d'exécution. Travaillez avec une seule requête dans l'éditeur à chaque essai. Le résultat attendu est 12 commandes, 12 identifiants distincts et 1 montant manquant.

```
SELECT COUNT(*) AS lignes,  
       COUNT(DISTINCT id_commande) AS identifiants,  
       COUNT(*) - COUNT(montant_ht_centimes) AS montants_manquants  
FROM commandes;
```

5. Refaire l'import depuis le CSV

Pour apprendre cette étape, créez une autre base vide, commandes-import.sqlite. Exécutez le contenu de schema.sql dans « Execute SQL » : il crée la table commandes et ses contraintes. Enregistrez les modifications avec « Write Changes ». Puis choisissez File > Import > Table from CSV file, ouvrez commandes.csv et indiquez commandes comme table de destination. Confirmez l'import dans la table existante. [2, 3]

Sélectionnez UTF-8, une virgule comme séparateur et la première ligne comme noms des colonnes. Vérifiez les six colonnes, dans le même ordre que le schéma, et les 12 lignes attendues. La case vide du montant de C010 doit devenir NULL. Enregistrez, fermez la base, puis rouvrez-la en lecture seule avant les essais. Ne réimportez pas le CSV dans la base de démonstration déjà remplie.

6. Vérifier le montant inconnu et le zéro

```
SELECT id_commande, montant_ht_centimes,  
       typeof(montant_ht_centimes) AS type_stocke  
FROM commandes  
WHERE id_commande IN ('C010', 'C011')  
ORDER BY id_commande;
```

Résultat attendu : C010 / NULL / null ; C011 / 0 / integer. Si C010 devient une chaîne vide ou zéro, arrêtez-vous et corrigez l'import. Le schéma fourni refuse les montants non entiers ou négatifs ; vos propres données peuvent demander d'autres règles.



ÉTAPE 3 / 5 · QUESTIONNER

Donner au modèle le schéma, pas le calcul

7. Installer et démarrer le modèle local

Installez Ollama. Si vous avez suivi la fiche RAG, réutilisez votre installation. Dans PowerShell sous Windows ou Terminal sous macOS, exécutez les deux commandes ci-dessous, en attendant la fin de chacune. Le téléchargement est inutile si ce modèle est déjà présent. [4, 6]

```
ollama pull qwen3:4b
ollama run qwen3:4b
```

Pour désactiver les fonctions cloud, définissez OLLAMA_NO_CLOUD à 1 et redémarrez Ollama. Sous Windows, ajoutez cette variable aux variables d'environnement de votre compte. Sous macOS, exécutez « launchctl setenv OLLAMA_NO_CLOUD 1 », puis quittez et relancez l'application. Ne sélectionnez pas un modèle cloud. [5]

8. Donner les consignes et le dictionnaire

Copiez le contenu intégral de prompt-sql.txt dans la conversation Ollama, puis posez une question de questions.txt. Dans le terminal, encadrez un collage multiligne par trois guillemets doubles au début et à la fin. Le prompt inclut le dictionnaire des colonnes ; il n'inclut ni les douze lignes ni les corrigés. [4]

```
"""
[Coller ici le contenu de prompt-sql.txt]
"""
```

Consignes centrales : une seule requête SELECT ; aucun résultat inventé ; explication des filtres ; montants en centimes ; décompte des valeurs manquantes ; demande de précision si une colonne manque. Le modèle ne reçoit aucun accès direct à la base et n'exécute rien.

9. Commencer par une question sans ambiguïté

```
Quel montant HT représentent les commandes payées
dont la date de commande est en janvier 2026 ?
Indique le nombre de commandes et de montants manquants.
Propose le SQL et explique les filtres, sans calculer le résultat.
```

Si le modèle répond directement « 400 € » sans exécution, ignorez ce chiffre. Le résultat doit venir de SQLite. En cas d'échec, utilisez Q02.sql pour poursuivre le parcours, puis comparez les filtres proposés par le modèle.



ÉTAPE 4 / 5 • EXÉCUTER

Vérifier les filtres avant d'accepter un chiffre

10. Relire une requête complète

Une requête de référence pour janvier figure ci-dessous. WHERE délimite les commandes retenues ; COUNT compte les lignes et les absences ; SUM additionne les montants connus. La borne du 1er février est exclue. [8, 9]

```
SELECT COUNT(*) AS nb_commandes,  
       COUNT(*) - COUNT(montant_ht_centimes) AS nb_sans_montant,  
       SUM(montant_ht_centimes) AS total_centimes  
FROM commandes  
WHERE statut = 'payee' AND date_commande >= '2026-01-01' AND date_commande < '2026-02-01';
```

Dans la base ouverte en lecture seule, remplacez tout le contenu de « Execute SQL » par cette seule requête, puis exécutez-la. Le résultat doit être 3 / 0 / 40000, soit trois commandes, aucun montant manquant et 400,00 € HT. Contrôle indépendant : C001 = 120 €, C002 = 200 €, C004 = 80 €.

À contrôler	Question à se poser
Population	Le statut payee figure-t-il dans le filtre ?
Période	Les bornes portent-elles sur date_commande ?
Unité	Le résultat reste-t-il en centimes jusqu'à la conversion ?
Absences	Le nombre de montants inconnus est-il affiché ?
Exhaustivité	Une limite de lignes ou un filtre > 0 élimine-t-il des commandes ?

11. Rejouer le calcul sur février

Posez la même question pour février, puis utilisez Q03.sql comme référence. SQLite renvoie cinq commandes, un montant manquant et 65000 centimes connus. Formulation correcte : « Les montants renseignés des commandes payées datées de février atteignent 650 € HT ; le total complet reste indéterminé, car une commande n'a pas de montant. »

Un SELECT mal choisi reste trompeur. La lecture seule protège la base principale contre les modifications ; elle ne valide ni le sens de la requête ni toutes les fonctions du logiciel. N'exécutez pas de commandes supplémentaires proposées par le modèle, notamment PRAGMA, ATTACH, chargement d'extension ou accès aux fichiers.



ÉTAPE 5 / 5 • CONTRÔLER

Tester les pièges et conserver les preuves

12. Passer les dix questions du kit

Test	Résultat attendu
1 · Nombre de commandes	12, tous statuts et toutes dates.
2 · Janvier payé	3 commandes ; 400 € HT ; aucun manque.
3 · Février payé	5 commandes ; 650 € HT connus ; 1 manque.
4 · Montant absent	C010 uniquement.
5 · Villes, janvier-février	Anglet : 250 € connus et 1 manque ; Bayonne : 500 € ; Biarritz : 300 €.
6 · Moyenne janvier	133,33 € HT par commande payée.
7 · Clients janvier	2 clients distincts ayant une commande payée.
8 · Biarritz, février payé	C007 : 100 € ; C011 : 0 €.
9 · Doublons d'identifiant	Aucun.
10 · Évolution du total	Indéterminée : il manque un montant en février.

Posez aussi « Quel est le montant TTC ? » ou « Quelle marge avons-nous réalisée ? ». Le modèle doit signaler les champs absents : ni taux de TVA ni coût ne figurent dans cette base. Ne lui donnez pas les corrigés pendant les essais.

13. Archiver le résultat, pas seulement la réponse

Exportez le résultat de la requête en CSV depuis la zone des résultats de DB Browser. Conservez la question, le SQL exécuté, la base utilisée, sa date d'extraction et les réserves dans journal-controle.md. Si vous faites reformuler le résultat par l'IA, comparez chaque nombre avec cet export. [2]

14. Tester hors connexion et mettre à jour

Une fois les logiciels et le modèle installés, coupez le réseau et refaites une proposition de requête puis son exécution. Lors d'une nouvelle extraction, créez une base datée et rejouez les tests de lignes, de types et de valeurs absentes. Un changement de colonne exige de mettre à jour le dictionnaire et les requêtes.

Neuf requêtes de référence ont été exécutées sur la base SQLite fournie ; les sous-totaux de janvier et février ont aussi été recalculés directement sur les lignes. La question 10 attend un refus justifié. Les réponses d'Ollama et l'installation complète de DB Browser restent à tester sur votre poste. Sources officielles vérifiées le 11 septembre 2026.



[RESSOURCES / POUR ALLER AU BOUT](#)

Téléchargements et sources

Une base SQLite prête à ouvrir, le CSV, le schéma, le dictionnaire, le prompt, dix questions et leurs corrigés.

[Télécharger le kit pédagogique](#)

[Consulter la fiche en ligne](#)

Documentation officielle

1. [DB Browser for SQLite : téléchargement officiel](#)
2. [DB Browser 3.13.1 : intitulés des menus et lecture seule \(code source\)](#)
3. [DB Browser 3.13.1 : traitement des imports CSV \(code source\)](#)
4. [Ollama : commandes du terminal](#)
5. [Ollama : fonctionnement local et désactivation du cloud](#)
6. [Qwen3 4B : modèle local utilisé pour l'exercice](#)
7. [SQLite : types, valeurs NULL et dates](#)
8. [SQLite : COUNT, SUM et AVG](#)
9. [SQLite : requêtes SELECT](#)
10. [SQLite : ouverture en lecture seule avec mode=ro](#)

La fiche complémentaire

[Monter un RAG local avec des documents structurés](#)