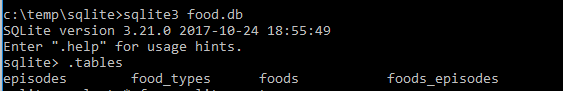
Лабораторна робота №3.

**SQLite. Виконання запитів на вибірку даних. Представлення**

1. Завантажимо тестову БД із дампу foods.sql.

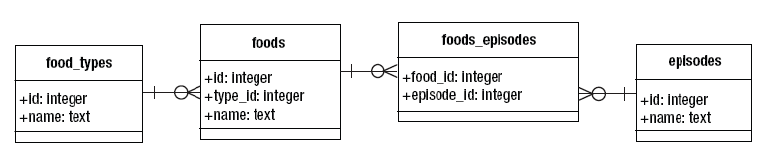


Далі запустимо sqlite3 із вказанням поточної БД food.db та переглянемо перелік таблиць БД



Як бачимо, БД складається з 4-х таблиць. Наведемо даталогічну модель БД та її загальний огляд.

База даних, що використовується містить імена страв з кожного епізоду серіалц Seinfeld. (If you’ve ever watched Seinfeld, you can’t help but notice a slight preoccupation with food). Глядач цього серіалу не може не відчути занепокоєність їжею ☺. Більш ніж ста вісімдесяти епізодів згадується більш ніж чотириста різних страв. На рисунку показано схему бази даних.



foods є головною таблицею. Кожен запис відповідає окремому блюду, чиє ім'я реєструється в полі name. Поле Type\_id посилається на таблицю food\_types, яка містить класифікацію продуктів (тобто, фаст-фуд, напої і так далі). Нарешті, таблиця food\_episodes поєднує продукти з великою кількістю епізодів.

1. Перед розглядом конкретних прикладів щодо вибірки даних із БД, наведемо загальний огляд команди select.

Оператор select включає реляційні операції за допомогою серії фраз (речення - clause). Кожна фраза відповідає своїй реляційної операції. У SQLite майже всі фрази не обов'язкові. Користувач SQLite може використовувати тільки ті, які операції він потребує. Найбільша загальна форма select в SQLite може бути представлена як



Одна із інтерпретацій оператора select - це уявити його як конвеєр, який обробляє відношення. На конвеєрі є необов'язкові процеси, виконання яких можна пропускати. Незалежно від того, використовуються чи не використовуються конкретні операції (процеси), конвеєр завжди працює однаково. На малюнку нижче можна подивитися порядок виконання. Виконання оператора select починається з фрази from, яка приймає одне або більше відношень і з'єднує їх в одне складне відношення і, потім, передає ланцюжку послідовних операцій.



Фраза from містить список декількох таблиць, представлень і підзапитів (представлених у вигляді змінної tables), розділених комами. Більш ніж одна таблиця (представлення або підзапит) буде з'єднуватися в одне відношення, яке на тому ж рисунку представляються іменем R1. Різні елементи поєднуються в одне відношення операцією join.

Фраза where відбирає необхідні записи з R1. За ключовим словом where слідує предикат (логічне вираз), який визначає критерій відбору записів з R1, які повинні бути включені в слідуючого відношення. Обрані записи утворюють нове відношення R2.

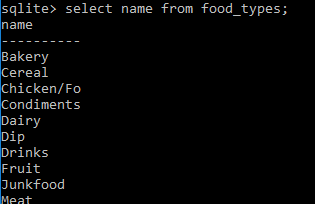
Як показує наступний малюнок, фраза select в SQLite з'єднує всі дані, розглянуті у фразі from, відбирає записи (обмежує) їх у фразі where і відбирає поля (проектує) у фразі select.



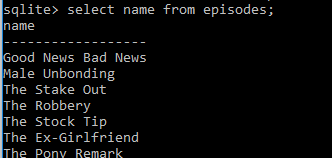
1. Далі переймемо до розгляду простих прикладів.
   1. Вибрати список типів блюд



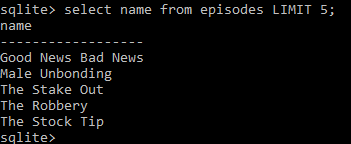
* 1. Вибрати назви типів блюд



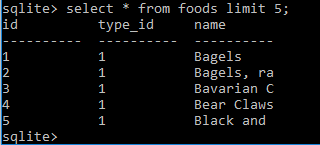
* 1. Вибрати назви серій серіалу



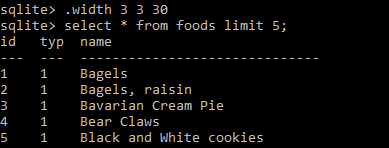
* 1. Кількість рядків у результаті попереднього запиту досить велика, виберемо лише 10 серій із загального списку



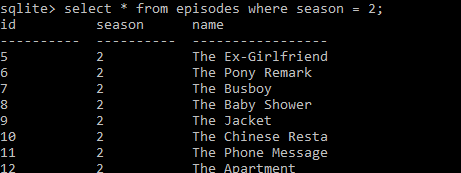
* 1. Виберемо 5 страв



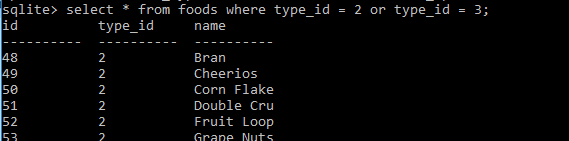
Для зручності виводу можна змінити ширину стовбців



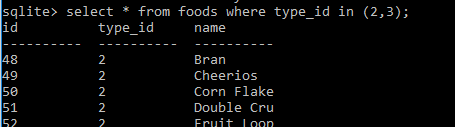
1. Фільтрування даних
   1. Список назв серій другого сезону серіалу



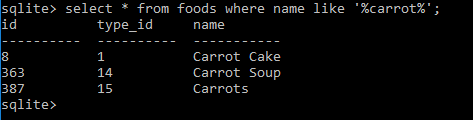
* 1. Список страв, що відноситься до 2-го та 3-го типу



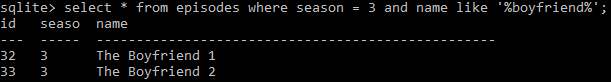
Або



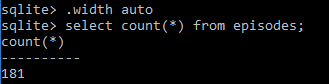
* 1. Страви, назва яких пов’язана із морквою



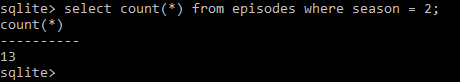
* 1. Серії 3-го сезону про бойфренда



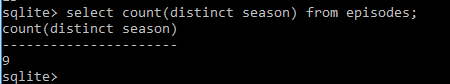
1. Групування та функції агрегації
   1. Загальна кількість серій у серіалі



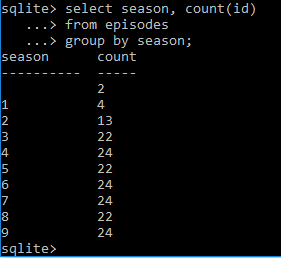
* 1. Кількість серій 2-го сезону



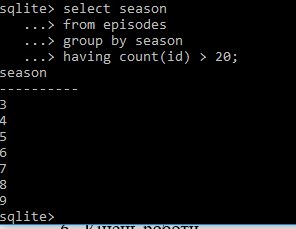
* 1. Кількість сезонів



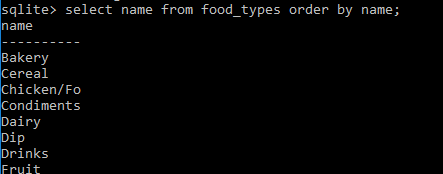
* 1. Кількість серій по сезонах



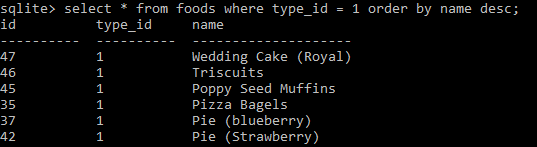
* 1. Сезони, в яких більше 20 серій



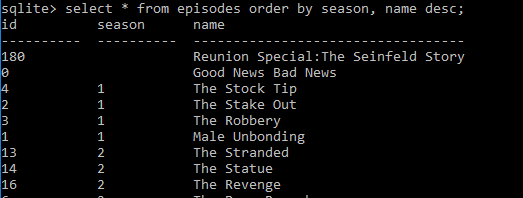
1. Впорядкування виводу
   1. Список типів блюд, впорядкований за назвою



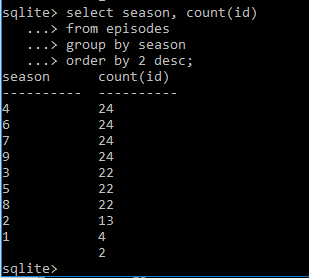
* 1. Список страв 1-го типу, відсортований за назвою в зворотному порядку



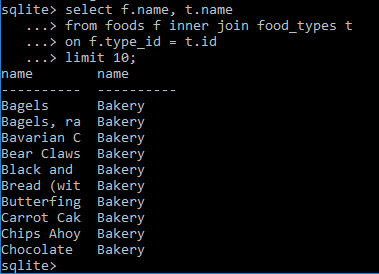
* 1. Список серій, відсортований за сезоном за зростанням, та за назвою серії за спаданням



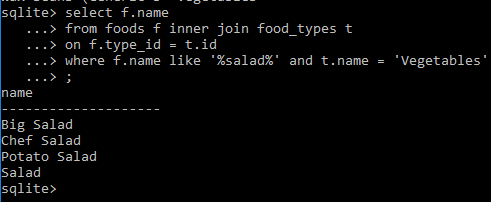
* 1. Список сезонів серіалу, впорядкований за кількістю серій за спаданням



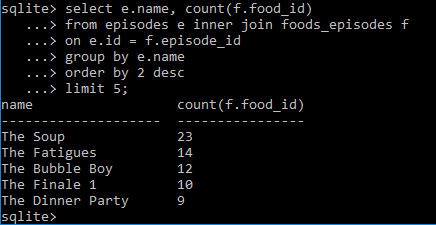
1. Запити до декількох таблиць
   1. Список страв із типом страви



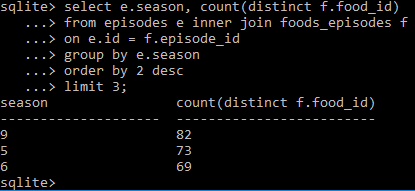
* 1. Салати із категорії овочі



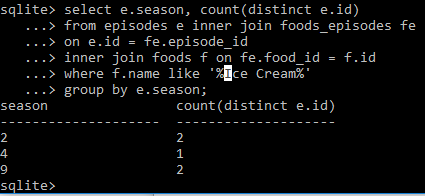
* 1. Перші 5 епізодів за кількістю згадувань про їжу



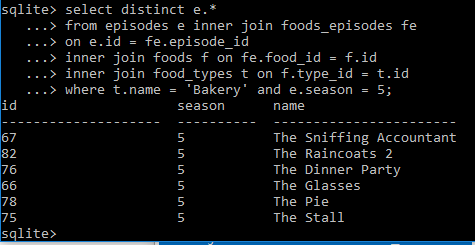
* 1. Перші 3 сезони за кількістю згадувань різних страв



* 1. У скількох серіях кожного із сезонів згадується про морозиво



* 1. У яких серіях 5-го сезону згадується випічка.



1. Крім таблиць та індексів в SQLite є ще 2 типи логічних об’єктів – це представлення та тригери. Почнемо із огляду представлень.

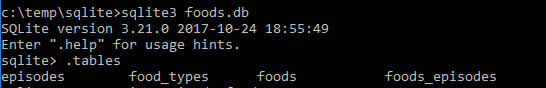
Представлення (view) є віртуальними таблицями. Їх ще називають похідними таблицями, в зв'язку з тим, що їх вміст є результатом виконання запитів до інших таблиць. Насправді, хоча представлення схожі на справжні таблиці, вони ними не є. Вміст справжніх таблиць є справжніми дані, в той час, як вміст представлення динамічно генерується під час звернення до нього.

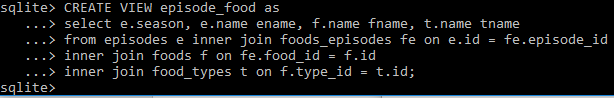
Синтаксис для створення представлення наступний:



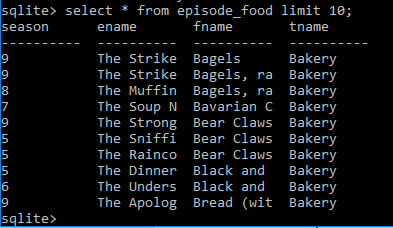
Назва представлення задається текстом name, а визначається оператором select - stmt. В результаті представлення буде виглядати, як таблиця з назвою name. Уявіть, що існує запит, який потрібно часто виконувати. Представлення - це засіб, який дозволяє уникати постійного набрання такого запиту. Інший приклад – якщо запит є частиною деякої кількості складних запитів.

1. Створимо представлення, що буде відображати сезони, назви епізодів, блюд, що в них згадуються та типів блюд, до яких вони відносяться.



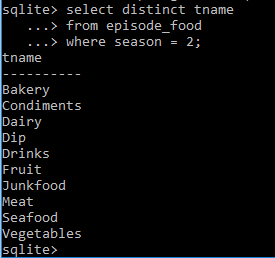


Перевіримо результат, виберимо перші 10 рядків із створеного представлення

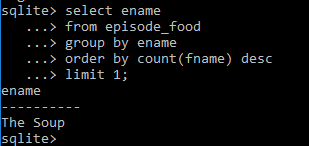


Виконаємо ще декілька запитів із використанням створеного представлення.

Виберимо які типи страв зустрічаються у другому сезоні серіалу



Або назву серії, в якій згадується найбільше страв



1. Кінець роботи