

Валентин Арьков

Введение в базы данных



Валентин Юльевич Арьков

Введение в базы данных

Учебное пособие

Шрифты предоставлены компанией «ПараТайп»

© Валентин Юльевич Арьков, 2024

Данное учебное пособие предназначено для изучения основ реляционных баз данных и языка запросов SQL. Кроме общего знакомства с материалом, студентам предлагается выполнить спроектировать и реализовать собственную базу данных. Подробно описано выполнение итогового проекта и процедура получения сертификата.

12+

Создано в интеллектуальной издательской системе Ridero

ОГЛАВЛЕНИЕ

[Введение в базы данных](#)

[Благодарности](#)

[Основные термины](#)

[Начало обучения](#)

[Зачет](#)

[Ориентация](#)

[Машинный перевод](#)

[Реляционные базы данных](#)

[Сертификат](#)

[Произношение: Youglish](#)

[Интеллектуальные чат-боты](#)

[Интеллектуальный помощник CS50AI](#)

[Облачная среда Visual Studio Code](#)

[СУБД SQLite](#)

[Скачивание файлов базы данных](#)

[Начало работы с СУБД](#)

[Notepad++](#)

[Облачные инструменты разработки](#)

[Google Colab](#)

[Онлайн-курс CS50 SQL](#)

[Итоговый проект](#)

[Курсовой проект](#)

[Начало проектирования](#)

[Прототип таблиц](#)

[Модели базы данных](#)

[Диаграмма как код](#)

[SQL запросы](#)

[Подготовка текста](#)

[Отработка произношения](#)

[Markdown](#)

[Описание проекта](#)

[Шаблоны документов](#)

[Scope — Рамки проекта](#)

[Functional Requirements — Функциональные требования](#)

[Representation — Представление данных и реализация](#)

[Entities — Информационные сущности](#)

[Relationships — Связи между таблицами](#)

[Optimizations — Средства оптимизации](#)

[Limitations — Ограничения](#)

[Пояснительная записка](#)

[Ссылки](#)

БЛАГОДАРНОСТИ

В данном учебном пособии использованы материалы курса Гарвардского университета **CS50 SQL**:

<https://cs50.harvard.edu/sql/>

Эти материалы используются как помощь студентам, которые знакомятся с базами данных.

Указанные материалы используются в их исходном, оригинальном виде, без изменений. Студентам предлагаются только ссылки на эти открытые источники.

В соответствии с лицензионным соглашением **CC BY-NC-SA 4.0**, данные материалы используются для некоммерческих целей. Данное учебное пособие также распространяется бесплатно на сайте издательства «Ридеро».

Условия лицензии расположены по адресу:

<https://cs50.harvard.edu/sql/2024/license/>

Автор данного пособия выражает признательность создателям онлайн-курса **CS50 SQL** за их многолетнюю кропотливую работу по разработке учебных материалов и предоставлению свободного доступа к этой информации и к образовательной платформе.

Изображение на обложке сгенерировано с помощью **GigaChat** от Сбера. Использована следующая команда (промпт): "нарисуй компьютерную базу данных с подключением к интернету и бумажные документы".

ОСНОВНЫЕ ТЕРМИНЫ

Для полного взаимного понимания определимся с названиями.

Онлайн-курс CS50 SQL: Интернет-курс Гарвардского университета CS50's Introduction to Databases with SQL.

Курс «Базы данных»: Курс по базам данных для студентов заочного отделения.

Итоговый проект (Final Project): Разработка базы данных в рамках онлайн-курса CS50 SQL. Выполняется по желанию.

Курсовой проект: Проект в рамках курса «Базы данных» для студентов заочного отделения. Выполняется и защищается обязательно. Результаты защиты выставляются в ведомость и зачетную книжку студента.

Описание итогового проекта: Файл README.MD, который загружается в систему Гарвардского университета для получения сертификата.

Пояснительная записка (ПЗ): Описание курсового проекта, представляемое на кафедру вуза на защите курсового проекта. ПЗ оформляется в соответствии с требованиями вуза. ПЗ хранится в бумажном виде на кафедре в течение нескольких лет. Электронный вариант ПЗ студент загружает по адресу, указанному преподавателем, для проверки — до защиты.

Зачет: Оценка, подтверждающая завершение курса студентом. Для получения зачета студент выполняет все практические задания онлайн-курса CS50 SQL и демонстрирует понимание изученного материала.

НАЧАЛО ОБУЧЕНИЯ

Мы начинаем работу с учебными материалами очно — в рамках нескольких «постановочных» занятий. Смысл в том, чтобы набрать разгон, набрать скорость, а затем самостоятельно пройти оставшуюся часть курса.

Почему нужно «разогнаться» и «набрать скорость»? Потому что если вы начнете с нуля и будете проходить онлайн-курс дома... Вы скажете: «Что-то непонятно... Давай-ка я через неделю вернусь». Потом через месяц вспомните и скажете, что уже поздно. И на этом всё изучение закончится.

Самое тяжелое — начать, разогнаться и сделать первые шаги. Полностью освоить автоматическую проверку заданий. И первые шаги гораздо легче сделать под присмотром опытного преподавателя и в окружении других таких же студентов.

ЗАЧЕТ

Изучение курса завершается защитой курсового проекта и зачетом.

Нас ожидает курсовой проект — тоже в соответствии с рекомендациями составителей курса. Студент самостоятельно выбирает то, что интересно. То, что действительно нравится. Вокруг выбранного объекта или деятельности мы строим небольшую базу данных: 4–5 табличек. Связь между таблицами по ключу. В этой базе данных мы реализуем индексы, триггеры и хранимые процедуры.

К моменту проведения зачета мы ожидаем, что студенты в состоянии написать SQL-запросы на основе материала, который обсуждается в рамках лекций и практических занятий в этом курсе. Для этого нужно самостоятельно прорешать все практические задания. Автоматическая система сразу проверяет, насколько правильно или неправильно вы это делаете. Добиваемся, чтобы автоматическая проверка подтвердила правильность решения.

ОРИЕНТАЦИЯ

Материалы и ссылки по нашему предмету размещены на страничке курса на гитхабе:

<https://github.com/Valentin-Arkov/Database>

В нашем курсе будет много практики. Это будет очень необычный курс. Таких курсов у вас никогда не было и больше не будет. В соответствии с учебным планом нам предстоит три лекции, одна практика и парочка лабораторных работ. Остальное — это самостоятельная работа по изучению материалов и выполнению курсового проекта. Так что это очень вводный, ознакомительный курс. Вряд ли кто-нибудь надеется, что за три лекции из вас планируется сделать крутых специалистов по базам данных. Кроме лекций, заочникам предстоит проделать большое количество самостоятельной работы — около 50 часов, см. рис.

Дисциплина	Лек	Лаб	Прак	СРС	КСР	Кон- ль	ФКР	З.Е.	Зач	КП
Базы данных	6	8	2	47	0	9	0	2	✓	✓

Рис. Учебный план заочной формы обучения

Следующая наша встреча планируется через пару месяцев –для защиты курсового проекта и проведения зачета.

Для самостоятельной работы мы будем использовать бесплатный онлайн-курс от Гарвардского университета. Это введение в базы данных на примере SQLite: Harvard CS50 SQL.

<https://cs50.harvard.edu/sql/>

Для дальнейшей работы понадобится зарегистрироваться на платформах GitHub и EdX.

GitHub

<https://github.com>

EdX

<https://www.edx.org/>

Материалы онлайн-курса CS50 SQL размещены также и на платформе EdX, где их тоже можно изучать. Но регистрация на курс будет происходить через сайт Гарвардского университета.

Если вы уже проходили курсы по базам данных, будет полегче. Но всё равно здесь есть, над чем поработать и чему научиться. Мы разбираем самые основы, самый фундамент. Если вы с этим материалом знакомы, вам предстоит эти знания закрепить и продемонстрировать свои навыки.

МАШИННЫЙ ПЕРЕВОД

Материал гарвардского онлайн-курса представлен, естественно, на английском. Если кому-то нужно на русском, есть варианты. Напомним, что браузеры Google Chrome и Яндекс Браузер позволяют нам автоматически включать машинный перевод веб-страничек, см. рис. Имеет смысл оба варианта смотреть – и оригинал, и машинный перевод. Если есть какие-то неточности в переводе, можно всегда в оригинале уточнить.

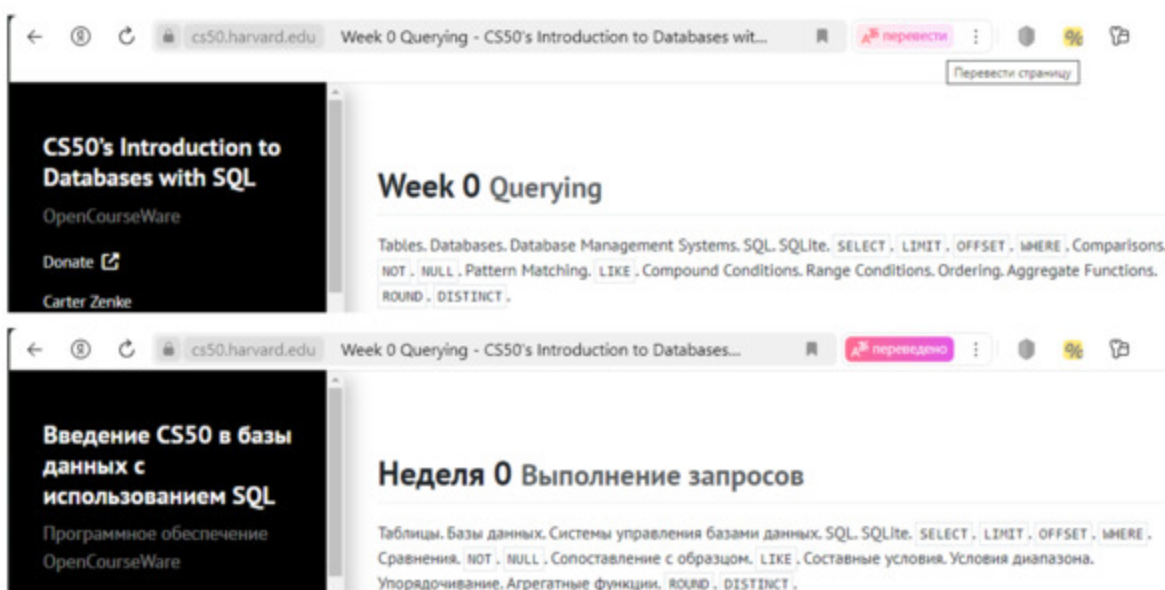


Рис. Машинный перевод сайта

Насчет машинного перевода. Эта функция есть во многих браузерах. Конечно, есть специализированные сайты и сервисы типа переводчика Google и Яндекса.

Плюс к этому, в рамках онлайн-курса представлен ряд видеороликов на английском языке. Их тоже можно смотреть

в оригинале и с переводом на русский. Эта технология реализована в браузере Яндекс, см. рис.

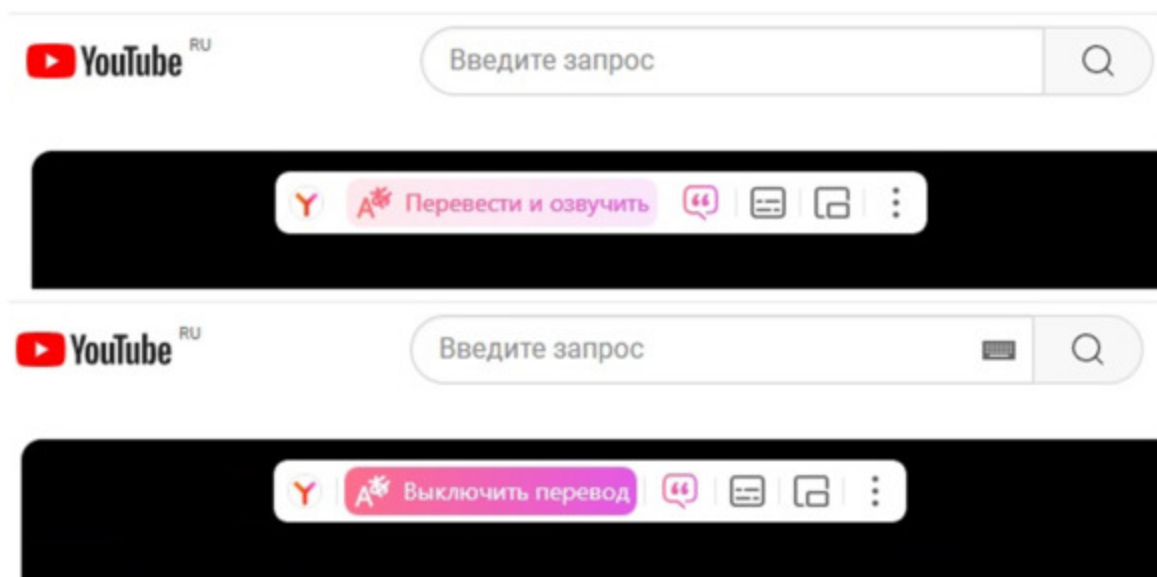


Рис. Машинный перевод видео

Заходим на YouTube. Вокруг ролика в Яндекс Браузере появляется желтая рамочка. Нам предлагают перевести видео на русский язык.

Плюс к этому Яндекс Браузер включает суммаризацию видео, см. рис. Достаточно любопытная функция. Если мы хотим получить краткое содержание, нам его автоматически сгенерируют.

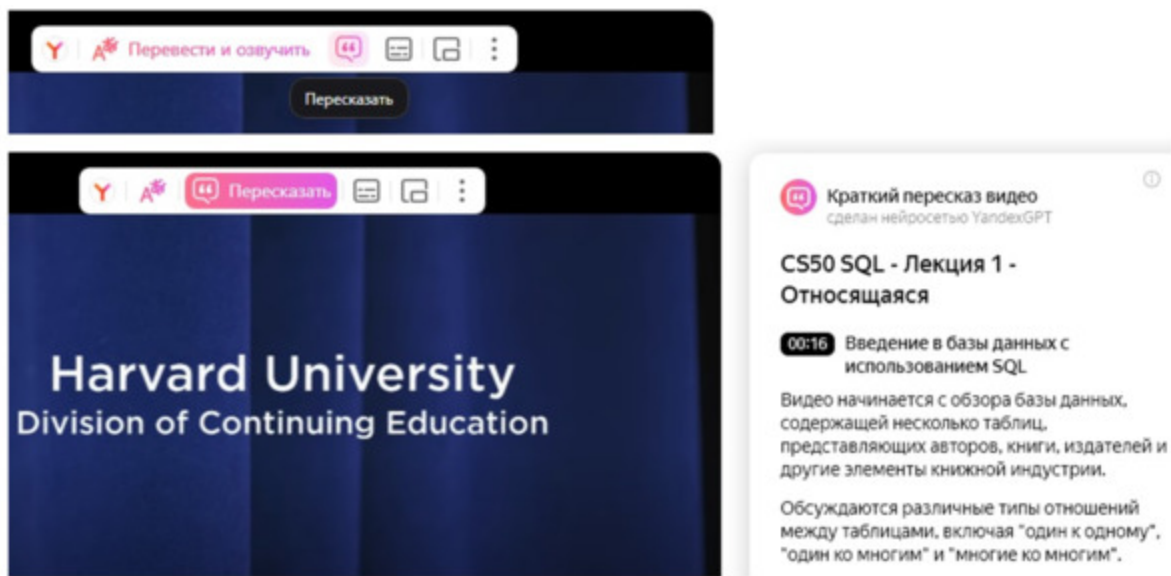


Рис. Краткое содержание видео

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

В рамках курса «Базы данных» мы разбираем только табличные базы данных (хотя есть и другие базы данных), см. рис. Официальное название: реляционные базы данных. Это красивое название, оно придумано для научности и для запутывания людей. Звучит очень красиво, чем-то напоминает «реляционную теорию».

- SQL
 - Табличные/реляционные
- NoSQL (Not only SQL)
 - Иерархические (JSON)
 - Сетевые
 - MongoDB
 - Redis
 - ...

Рис. Классификация БД

На самом деле изначально слово relation — «отношение» — применялось к табличке с данными. Фактически можно иметь одну табличку и уже называть ее базой данных.

То, с чем мы имеем дело сейчас, — это несколько табличек и связи между ними. Схема такой базы называется Entity-Relationship, или ER-диаграмма. Здесь Relation уже означает «связь между таблицами». Но когда говорят «реляционные базы данных», имеются в виду именно таблицу (в единственном числе) или таблицы (несколько штук).

Обычно к таким базам обращаются с помощью SQL-запросов. Поэтому их условно называют SQL базы данных.

В процессе изучения данного курса вам постепенно будут демонстрировать, что на самом деле хранение в табличках не очень удобно. Особенно если у нас появляется отношение «один ко многим» или «многие ко многим». В этом случае приходится держать отдельную, вспомогательную таблицу, в которой организуется эта взаимосвязь.

Соответственно, есть другие типы баз данных, где это ограничение совершенно ни на что не влияет. Это могут быть структуры типа JSON. Все остальные базы данных условно называются NoSQL. Расшифровывается это не как «противоположность SQL». На самом деле это означает «не только SQL». Фактически, современные СУБД, такие как Postgres, уже включают в себя возможность поддержки JSON с индексацией. А это уже не таблица, это более сложная, часто иерархическая структура.

Есть много красивых названий, есть много популярных технологий, но для того, чтобы к ним перейти, сначала должен быть фундамент, база, основа. И вот этими вещами мы и будем с вами заниматься. Разобравшись с табличными базами данных, можно быстро освоить остальные типы — если понадобится.

СЕРТИФИКАТ

Конечный результат изучения онлайн-курса — если заинтересует — это сертификат Гарвардского университета, см. рис. Бесплатный сертификат, который можно добавить в свое резюме. Если честно выполнить все задания и отправить их все на проверку, то можно получить такой подарок. На этом сертификате есть ссылка и QR-код. Любой работодатель может убедиться, что это не нарисованная картинка, а подлинный документ.



Рис. Сертификат о завершении онлайн-курса

Итак, онлайн курс называется CS50. CS означает Computer Science, то есть информатика. Число 50 намекает, что это половина от сотни. Фокус в том, что на Западе курсы начального уровня часто называют 101. И, соответственно, 50 — это половинка от 101. Так что это «введение во введение».

Но если посмотреть на объем, этот онлайн-курс потребует примерно один-два месяца работы — вдумчивой, неторопливой — и над заданиями, и над итоговым проектом. В процессе выполнения заданий на отдельной страничке сайта студент получает «зеленые галочки». Это так называемый GradeBook, то есть своего рода «зачётка», см. рис.

CS50 SQL



Gradebook for Valentin-Arkov

edX Account: valentinarkov

View submissions on <https://submit.cs50.io/courses/1997>

Congratulations! You've completed CS50 SQL! [Click here](#) for your CS50 Certificate. If you signed up for a [verified certificate](#) from edX, it should appear at <https://edx.org/dashboard> within 30 days. Email certificates@cs50.harvard.edu with any questions!

Current Progress

8 of 8 weeks complete.

0 Week 0

Complete Cyberchase, 36 Views or Normals, and Players.

- ✓ Cyberchase
- ✓ 36 Views or Normals
- ✓ Players

■ ■ ■

FP Final Project

- ✓ Final Project

Рис. «Зачетная книжка» курса

Здесь отмечается прохождение каждого этапа. Например, неделя первая — здесь у нас три зеленых галочки. Если загрузить итоговый проект, получаем галочку Final Project. После этого система генерирует сертификат о завершении.

ПРОИЗНОШЕНИЕ: YUGLISH

Среди прочего мы расширяем кругозор, повышаем грамотность и культуру.

И начнем мы с самого простого, с произношения. Потому что традиционная база данных условно называется SQL. Желательно знать, как это название правильно и грамотно читается. Здесь нам пригодится инструмент под названием YouGlish. Это поисковая машина поверх YouTube. Она позволяет находить выбранное слово или фразу в видеороликах, см. рис.

YouGlish.com = Youtube + English

The screenshot shows the YouGlish.com interface. At the top, the logo 'YouGlish' is followed by 'for English'. Below it is a search bar containing the text 'SQL' and a red button labeled 'Say It!'. Under the search bar are tabs for 'All', 'US', 'UK', and 'AUS'. Below the tabs, the text 'How to pronounce SQL in English (1 out of 4883):' is displayed. The main content area shows a video player with a man speaking. The video title is 'Roadmap: Data Infrastructure'. The video content includes a line graph titled 'dbt Github Stars' showing an upward trend. The video player has a progress bar at 7:50 / 10:05. Below the video player, the text 'DBT leverage the preexisting very large community of SQL users and' is displayed, with 'SQL' highlighted in yellow.

Рис. Произношение носителей

Попробуйте проделать такое упражнение. Заходим на **YouGlish.com** и вводим в строке поиска SQL. Теперь можно посмотреть фрагменты роликов, где это слово произносят носители языка. Тогда мы будем готовы к тому, как это будут другие люди называть.

При воспроизведении будет подсвечиваться слово, которое сейчас произносится. Это тоже удобно. Можно перемотать на следующий ролик, затем на следующий, на следующий.

Под роликом выводятся комментарии и дополнительные сведения. Это тоже может быть полезно.

ИНТЕЛЛЕКТУАЛЬНЫЕ ЧАТ-БОТЫ

Далее можно подробнее разобраться с произношением с помощью интеллектуального помощника. Сейчас доступно очень много чат-ботов и сервисов — как оригинальный ChatGPT, так и его разнообразные аналоги.

Можно порекомендовать сервис **Perplexity**:

<https://www.perplexity.ai/>

Этот сервис бесплатный. Самое интересное, что он дает ссылки на источники — в отличие от большинства чат-ботов. Нам сообщают, откуда были взяты данные. Результат представляют их в обобщенном красивом виде. Работает и на русском языке, и на английском.

Попробуем спросить, как правильно произносить SQL, см. рис. Можете даже по-русски спросить об этом. Если на этом сайте зарегистрироваться, у нас еще будет история всех наших запросов. Но это уже не обязательно.

Your role is an experienced IT specialist. Please could you explain to a beginner how to pronounce "SQL" and why it is done this way.

✿ Answer

How to Pronounce "SQL" and Why

Pronunciation:

- "SQL" is pronounced as either "ess-cue-el" or "sequel."
- The most common pronunciation is "sequel," which is widely accepted in the tech industry.

Reason for Pronunciation:

- The term "SQL" stands for "Structured Query Language," which is a standard

Рис. Запрос и ответ чат-бота

Итак, с помощью этого чат-бота попробуйте выяснить, как правильно произносится SQL и почему. Запрос может быть длинным, мы можем многое уточнять. Мы можем сказать: «Действуй как ИТ-специалист, расскажи простыми словами, объясни шаг за шагом». Есть несколько таких ключевых фраз, которые позволяют направить этот сервис в нужную сторону, в нужную точку.

При прохождении курса «Базы данных» можно регулярно консультироваться у этого бота. Он будет подсказывать — вплоть до генерации текста программы.

В отличие от поисковых машин, он может давать разные ответы — в разное время и на разных компьютерах. Кому-то он может сказать: «S-Q-L», а кому-то может сказать: «Sequel». А кому-то и то, и другое. Мне, например, этот чат-бот дал два варианта. Причем он объясняет, что предпочтительное, наиболее распространенное — это все-таки «Sequel».

И далее — небольшая история вопроса. Первоначально этот язык запросов так и назывался: Sequel. Но потом возникли проблемы с лицензированием. Кто-то уже запатентовал это название, и пришлось придумать другой, обходной вариант. Сократили название до трех букв, но первоначальное произношение сохранили. И на Западе специалисты произносят SQL по-прежнему: «Сиквел». Ну а в России, соответственно, более привычно читать по буквам S-Q-L, причем L с мягким знаком. Мы русифицируем всё, к чему прикасаемся. Это нормально. Это жизнь.

ИНТЕЛЛЕКТУАЛЬНЫЙ ПОМОЩНИК CS50AI

Теперь что касается искусственного интеллекта (ИИ) на онлайн-курсе. На сайте курса Гарварда имеется встроенный интеллектуальный помощник. Это ссылка CS50AI с изображением утенка. Выясняется, что этот резиновый утенок — любимая игрушка программистов в Америке. Не знаю, почему. И с этим утенком у них связана одна технология. Если вы хотите в чем-то разобраться — объясните проблему утенку. Представьте, что вы объясняете этот предмет маленькому глупому утенку. Его надо держать в руках и ему рассказывать: «Понимаешь, я делаю базу данных. В ней будет две таблицы. Ты понял?»

Фактически перед нами урезанный ChatGPT, который только помогает с этим курсом. А на другие вопросы он, скорее всего, откажется отвечать. И генерировать программу он тоже вряд ли будет. Но он скажет: «Вот эта команда работает так-то. Например...» И дает пример. Решение задачи он не обязан генерировать, но зато он объяснит, как работает вот эта команда, см. рис. Вот такой любопытный инструмент.

CS50.ai      

ddb

Quack. I am CS50's duck debugger (ddb), an experimental AI for [rubberducking](#). Quack quack. My replies might not always be accurate, so always think critically and let me know if you think that I've erred. Conversations are logged for debugging's sake. Quack quack quack.

you

Could you tell about the order of keywords inside the SELECT clause

ddb

Sure, I'd be happy to explain! In a SQL SELECT statement, the order of keywords is important. Here's a basic structure:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition  
GROUP BY column1, column2, ...  
HAVING condition  
ORDER BY column1, column2, ... ASC|DESC;
```

См. Интеллектуальный помощник онлайн-курса

ОБЛАЧНАЯ СРЕДА VISUAL STUDIO CODE

Следующий инструмент онлайн-курса – Visual Studio Code. Фактически это Visual Studio, который работает через GitHub в облаке. Эта виртуальная среда подготовлена для работы именно в данном онлайн-курсе. Через эту среду мы отправляем наши практические задания на автоматическую проверку. В процессе выполнения заданий нам придется перейти по этой ссылке. При первом входе будет сгенерирована эта среда (собирается соответствующий контейнер), потом она будет открываться в браузере, см. рис.

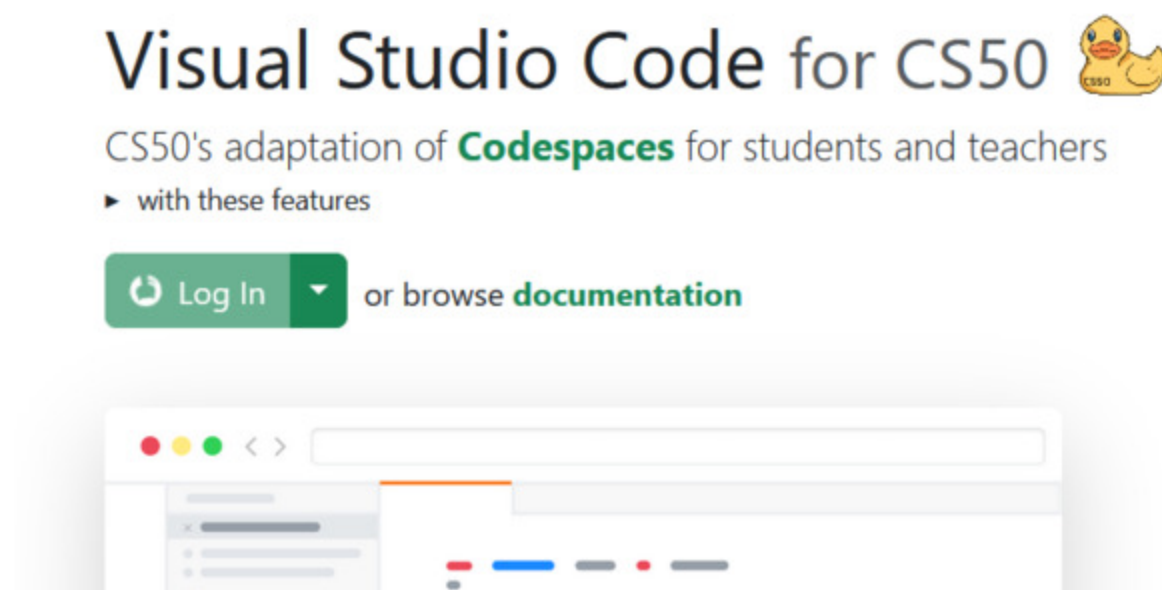


Рис. Запуск VS Code

Мы получаем облачную виртуальную машину. Работа выполняется в ОС Linux, см. рис.

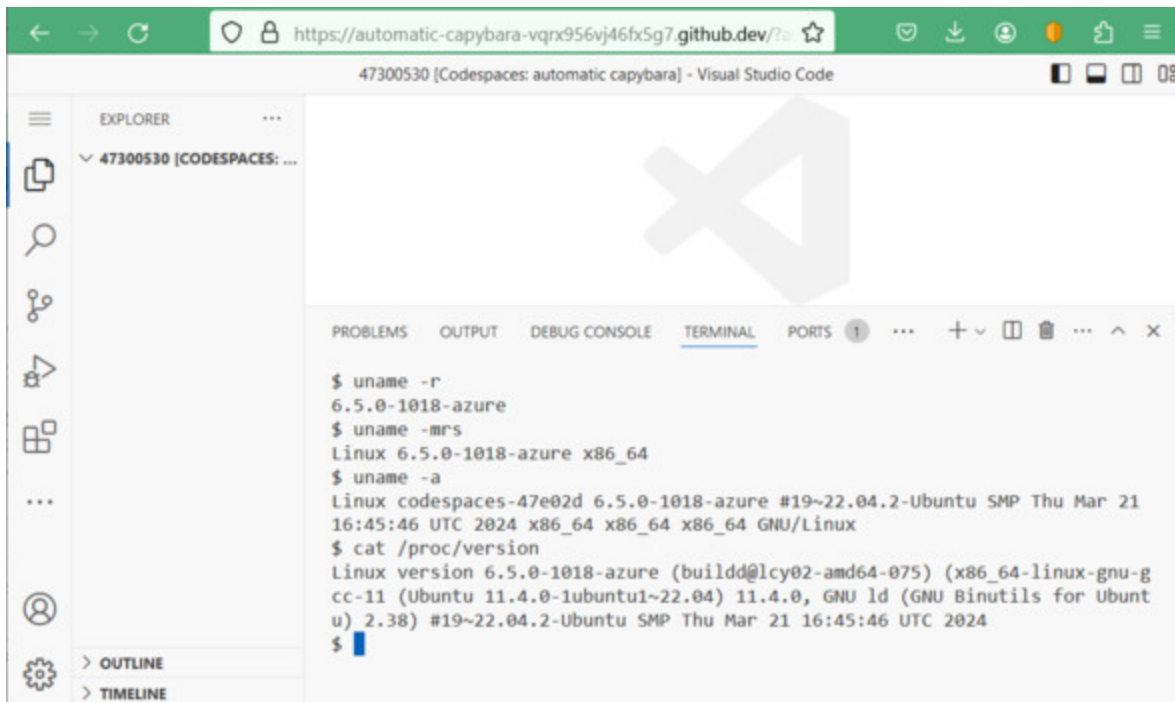


Рис. Облачная среда разработки

Конечный этап онлайн-курса — представление итогового проекта (если вас интересует сертификат Гарварда). И для этого нужно не просто разработать игрушечную базу данных, но и сделать небольшую презентацию. Это будет видеоролик минуты на 2–3 с рассказом на английском языке про свой проект. Его нужно записать, выложить на YouTube, а ссылочку передать на проверку в составе описания проекта.

СУБД SQLITE

Ну и теперь, собственно, наш главный инструмент — система управления базами данных (СУБД) SQLite. Здесь можно отработать фундаментальные, базовые принципы, самые основы работы с базами данных.

В простейшем случае мы можем скачать и запускать эту программу — даже без установки на компьютер. Такие варианты программ называются Portable Applications — «переносимые приложения».

Чем это интересно? Не нужно засорять реестр, не нужно тратить время на инсталляцию. Мы выбираем исполняемые файлы для нужной операционной системы, скачиваем, разворачиваем в каталоге и запускаем. На этом всё. Программа работает из локального каталога.

Для загрузки SQLite переходим на сайт:

<https://www.sqlite.org/>

Выбираем исполняемые файлы Binaries для своей операционной системы, см. рис.



[Home](#) [About](#) [Documentation](#) [Download](#) [License](#) [Support](#) [Purchase](#)

SQLite Download Page

Precompiled Binaries for Windows

sqlite-dll-win-x86-3450200.zip (1.00 MiB)	32-bit DLL (x86) for SQLite version 3.45.2. (SHA3-256: 7385e36417a2b660f4630c39045202b77eae79588e)
sqlite-dll-win-x64-3450200.zip (1.26 MiB)	64-bit DLL (x64) for SQLite version 3.45.2. (SHA3-256: 9a8c9e0580f722ce51ef89766c35e55713d81ed1c44)
sqlite-tools-win-x64-3450200.zip (4.77 MiB)	A bundle of command-line tools for managing SQLite (SHA3-256: 3b35d5c27cbd4ee3f5cc91669cad273ea46dec89ec4)

Рис. Загрузка SQLite

Создаем новый рабочий каталог для опытов с базой данных, скачиваем в него выбранные заархивированные файлы и распаковываем архивы. Открываем командное окно в текущем каталоге.

СКАЧИВАНИЕ ФАЙЛОВ БАЗЫ ДАННЫХ

Для выполнения практических заданий требуется скачать соответствующий файл с базой данных. Например, для заданий Cyberchase переходим в нужный раздел онлайн-курса:

CS50 SQL – Week 0 Querying – Problem Set 0 – Problems – Submit Cyberchase – Distribution Code – Download the distribution code – wget...

Здесь нас интересует ссылка на файл, см. рис.

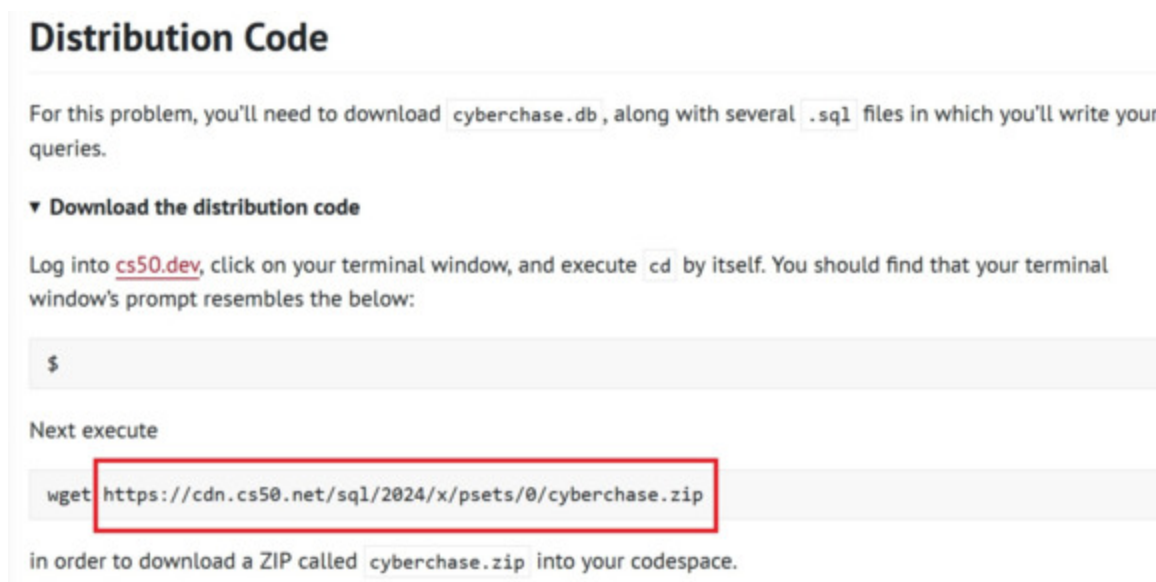
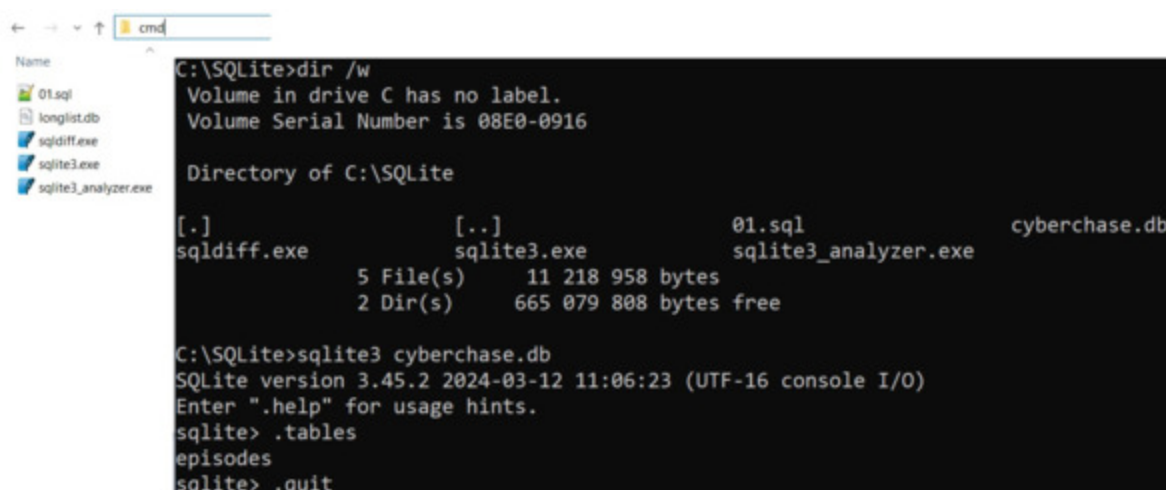


Рис. Файл с базой данных

Скачиваем файл и распаковываем его в том же рабочем каталоге.

НАЧАЛО РАБОТЫ С СУБД

Далее в Проводнике Windows щелкаем мышкой в строчку адреса, пишем cmd и нажимаем Enter. Открывается командное окно именно в этом каталоге, см. рис.



```
C:\SQLite>dir /w
Volume in drive C has no label.
Volume Serial Number is 08E0-0916

Directory of C:\SQLite

[.]                [..]                01.sql                cyberchase.db
sqldiff.exe        sqlite3.exe        sqlite3_analyzer.exe
5 File(s)          11 218 958 bytes
2 Dir(s)           665 079 808 bytes free

C:\SQLite>sqlite3 cyberchase.db
SQLite version 3.45.2 2024-03-12 11:06:23 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> .tables
episodes
sqlite> .quit
```

Рис. Работа в командной строке

Вводим sqlite3 и через пробел название файла cyberchase.db. Выводим список таблиц в этой базе данных. Выходим. Встроенные команды начинаются с точки.

Задаем табличный режим. mode table для вывода результатов SQL-запросов, см. рис. Просматриваем командой. schema схему базы данных. Фактически, это SQL запрос CREATE TABLE со всеми параметрами.

```

C:\SQLite>sqlite3 cyberchase.db
SQLite version 3.45.2 2024-03-12 11:06:23 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> .mode table
sqlite> .schema
CREATE TABLE IF NOT EXISTS "episodes" (
  "id" INTEGER,
  "season" INTEGER,
  "episode_in_season" INTEGER,
  "title" TEXT,
  "topic" TEXT,
  "air_date" NUMERIC,
  "production_code" TEXT,
  PRIMARY KEY("id")
);
sqlite>

```

Рис. Знакомство с базой данных

Формат вывода в виде таблицы выглядит таким образом, см. рис. Для каждого столбца выводится заголовок. Имеются разделительные «линии» между столбцами. Выход: .exit.

```

sqlite> SELECT title FROM episodes LIMIT 5;
+-----+
|      title      |
+-----+
| Lost My Marbles |
| Castleblanca    |
| R-Fair City     |
| Snow Day to be  |
| Exact           |
| Sensible Flats  |
+-----+
sqlite> .exit

```

Рис. Табличный формат вывода

NOTEPAD++

Редактировать собственные SQL запросы можно в обычном блокноте. Удобнее использовать любую среду с подсветкой синтаксиса. Например, можно использовать инструмент Notepad++:

<https://notepad-plus-plus.org/>

Запрос сохраняем в виде файла *.sql в том же рабочем каталоге. Для выполнения запроса используем команду. read, см. рис. Запрос заканчивается точкой с запятой. Это позволяет вводить запросы в несколько строк.

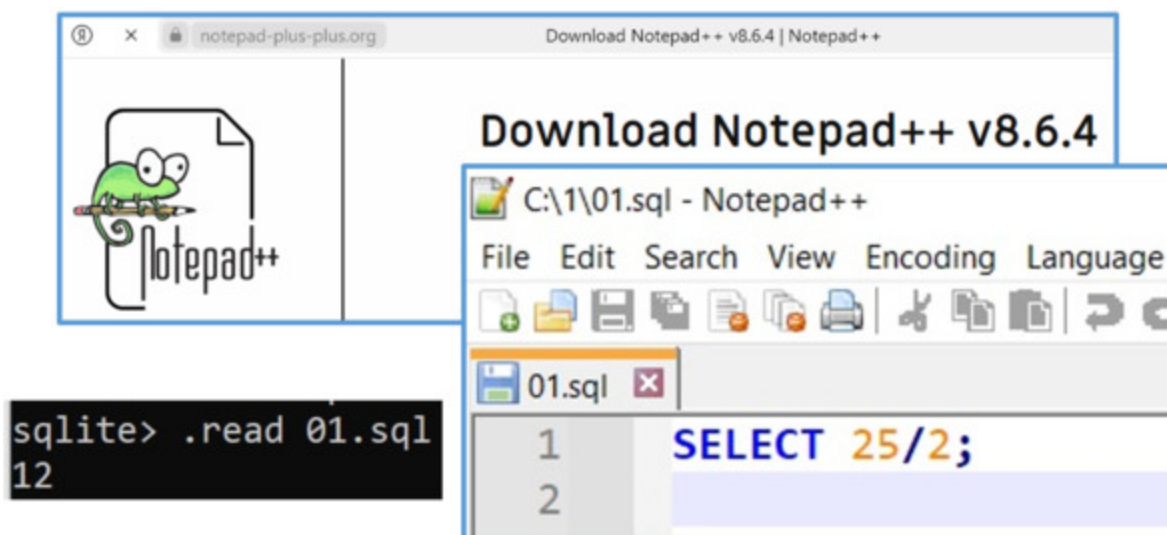


Рис. Редактирование и выполнение SQL запроса

В Notepad++ подсвечиваются ключевые слова. Такая подсветка синтаксиса будет хорошо смотреться в отчете, в описании курсового проекта. Если в курсовом проекте текст программы выводится одним цветом, это читается очень тяжело. Если есть подсветка синтаксиса, материал воспринимается гораздо веселее.

При копировании и печати из современных редакторов и сред разработки мы получаем не только текст, но и форматирование, включая шрифт и цвет, см. рис.

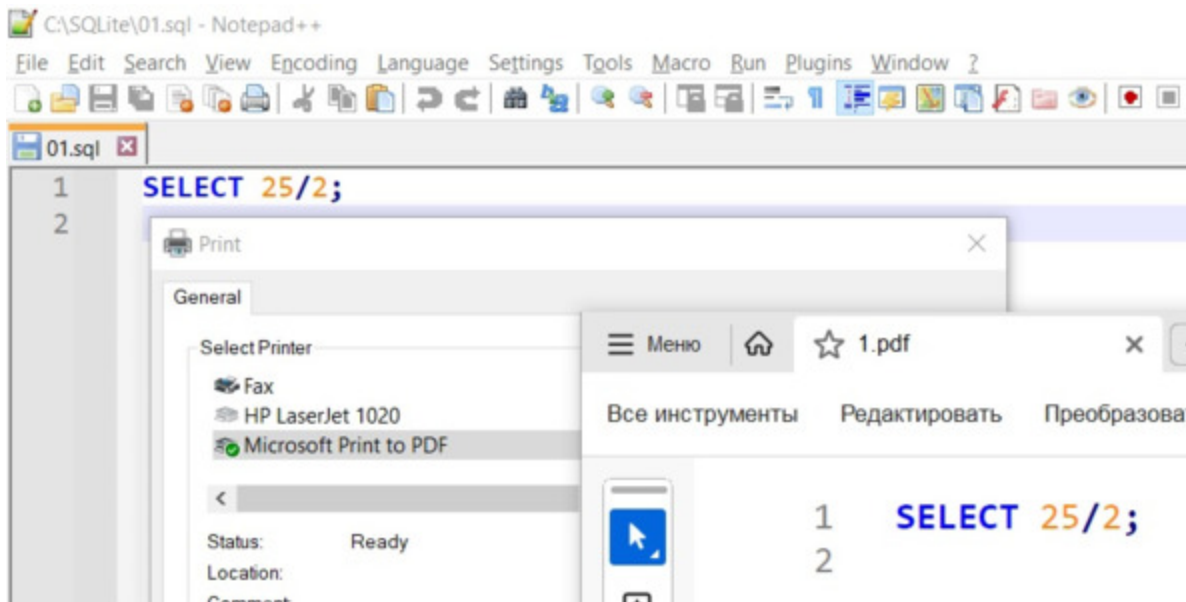


Рис. PDF с подсветкой синтаксиса и нумерацией строк

ОБЛАЧНЫЕ ИНСТРУМЕНТЫ РАЗРАБОТКИ

Есть и бесплатные онлайн инструменты, например, SQLite Online:

<https://sqliteonline.com/>

Здесь вообще ничего не потребуется делать на локальном компьютере. Можно загружать файл базы данных и с ним производить любые действия.

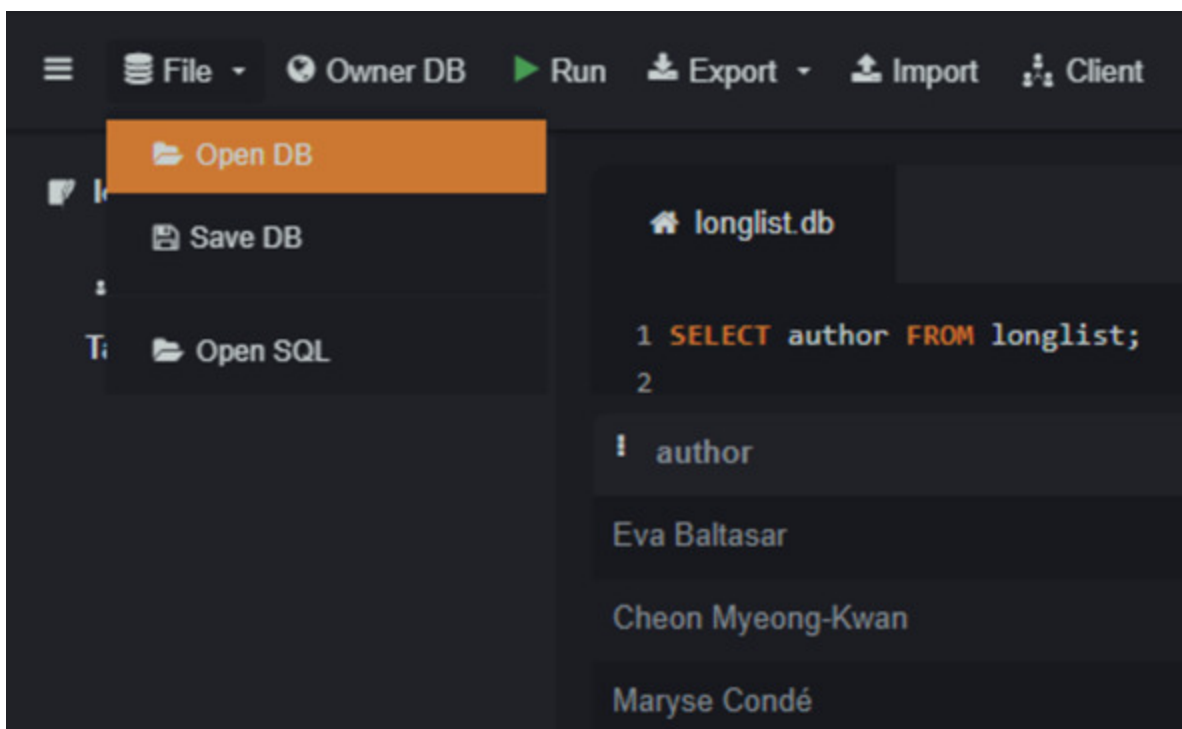


Рис. Работа в SQLite Online

Либо можно сразу работать в среде гарвардского курса Visual Studio Code.

GOOGLE COLAB

Наконец, SQLite доступен имеется в облачном инструменте под названием Google Colab (Co-Laboratory):

<https://colab.research.google.com/>

Colab изначально рассчитан на работу в Python. Но через восклицательный знак мы можем выйти в командную строку и работать в Linux, вводить любые команды. Кроме того, мы можем из Python выполнять SQL-запросы, см. рис.

- Mount Google Drive
 - `drive.mount()`
- Connect to SQLite Database
 - `sqlite3.connect()`
- Query the Database
 - `pd.read_sql_query()`
- Execute SQL Statements
 - `%%sql` magic
- Load CSV & Save Table
 - `pd.read_csv()`
 - `df.to_sql()`

Рис. Работа с SQLite в Google Colab

ОНЛАЙН-КУРС CS50 SQL

Рассмотрим материалы курса CS50 Introduction to Databases with SQL, см. рис. В рамках этого онлайн-курса есть ряд лекций и практические задания, а также итоговый проект.

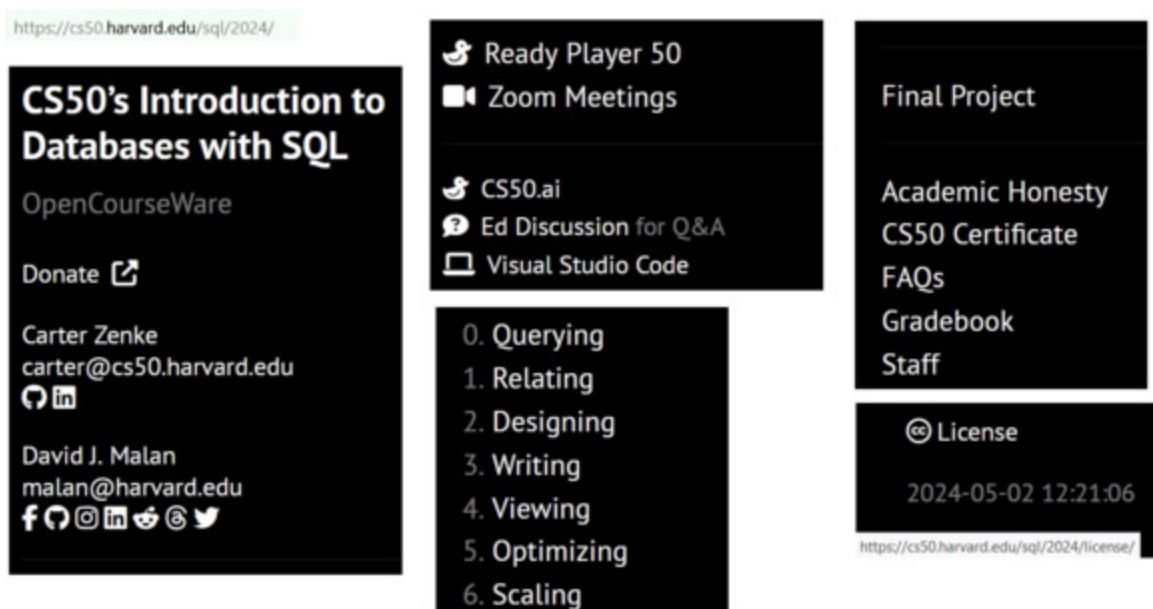


Рис. Разделы сайта CS50 SQL

В левой панели имеется список разделов от 0 до 6.

Дальше идет итоговый проект — **Final Project**.

Кроме этого, есть раздел **Academic Honesty**, что означает «Честное изучение предмета». Здесь раскрываются правила по поводу списывания, по поводу подсказывания и по поводу генерирования — рекомендуем ознакомиться. Например, не рекомендуется демонстрировать тексты своих программ, чтобы

другие студенты сами работали с задачами и приходили к своим собственным решениям.

После того, как вы ознакомитесь с вводным материалом, можно переходить к нулевой теме: **Querying** — Создание и использование запросов. Здесь есть ссылка на видеозапись лекции и ссылка на YouTube — если кому-то интереснее работать в отдельном окне.

Здесь же имеется **Notes** — полный текст лекции с примерами кода и с описанием команд. И здесь, обратите внимание, тоже идет подсветка синтаксиса. Если включить машинный перевод, то, скорее всего, SQL-запросы останутся на английском.

При изучении лекции рекомендуем выполнять все демонстрации и примеры. Для этого скачиваем материалы по ссылке `How to download lecture source code into Visual Studio Code`. Можно работать и на локальном компьютере, и в облачной среде курса.

Для работы в облаке переходим по ссылке **Visual Studio Code** в левой части страницы курса:

<https://cs50.dev/>

При первом входе нам потребуется войти на сервис **GitHub** со своей учетной записью и разрешить доступ к нашему аккаунту группе разработки Гарварда. Облачный сервис **Visual Studio Code** развернут на GitHub и адаптирован для потребностей курса CS50 SQL. Здесь в окне терминала можно загрузить файлы. Для этого копируем команды из указанного выше раздела **How to download...** Теперь можно поработать с базой данных и позапустить готовые примеры SQL запросов, см. рис.

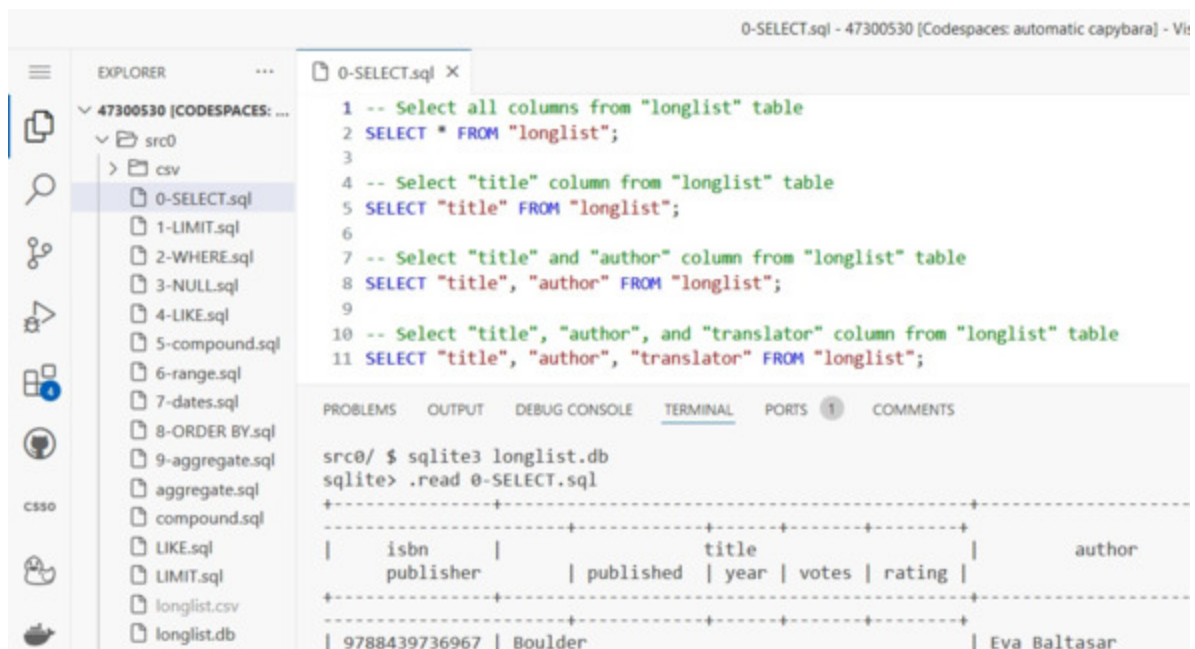


Рис. Облачная среда разработки VS Code

В левой панели есть набор разнообразных инструментов, с которыми будет полезно ознакомиться. Укажем только один — интеллектуальный помощник **CS50's duck debugger (ddb)**, см. рис. Он отвечает на вопросы, заданные на английском. Хорошо работает вместо справочной системы.

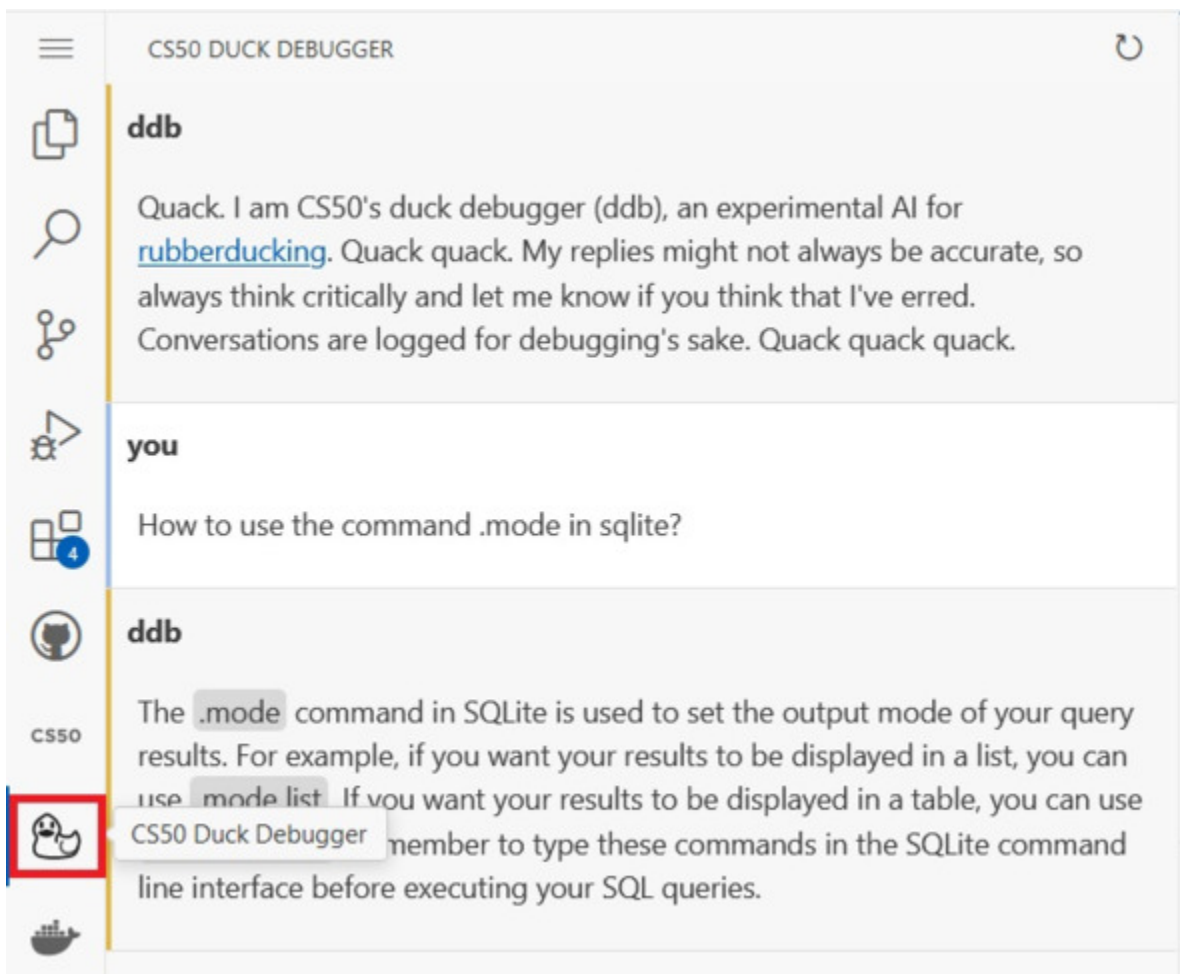


Рис. Интеллектуальный помощник курса

После того, как вы ознакомились с материалом лекции, можно переходить к практическим заданиям. Внизу страницы находим **Problem Set 0** — практические задания для нулевой недели. Здесь разделы курса условно называют «неделями». И этот набор заданий начинается с того, что мы подключаемся к системе со своим логином и паролем. Действия по подключению подробно описаны в разделе **What to Do — Preparation**. В качестве первой команды нам предлагают ввести **update50** для обновления программ.

Теперь можно выполнять сами задания — раздел **Problems**. Начинаем с первого задания: **Submit Cyberchase**. Читаем описание, смотрим демонстрацию и обращаем внимание на раздел

Distribution code. Здесь описано, как скачать и развернуть файлы для работы. Это Linux, и для нас это возможность познакомиться с новой операционной системой — если раньше не приходилось.

Если вы делаете это на локальной машине, мы просто выделяем адрес этого zip-файла, скачиваем и разворачиваем в рабочем каталоге.

Дальше у нас идет раздел **Schema** — это описание схемы базы данных. Описана каждая таблица, поля и их содержание.

Ну и самое главное — раздел **Specification** — это описание заданий. В рамках этого практического задания у нас есть ряд пунктов. Каждый выполняется в отдельном SQL-запросе.

В разделе **How to Test** даются пояснения по самопроверке — какие результаты ожидаются при выполнении каждого запроса. Напомним, что для подсчета количества строк можно использовать конструкцию **SELECT COUNT (*)**.

После того, как мы убедились в правильности работы запросов, мы вызываем общее тестирование — раздел **Correctness** — Проверка правильности. Копируем команду **check50** и запускаем в **VS Code**. Далее можно просмотреть подробный протокол тестирования и исправить обнаруженные ошибки, см. рис.

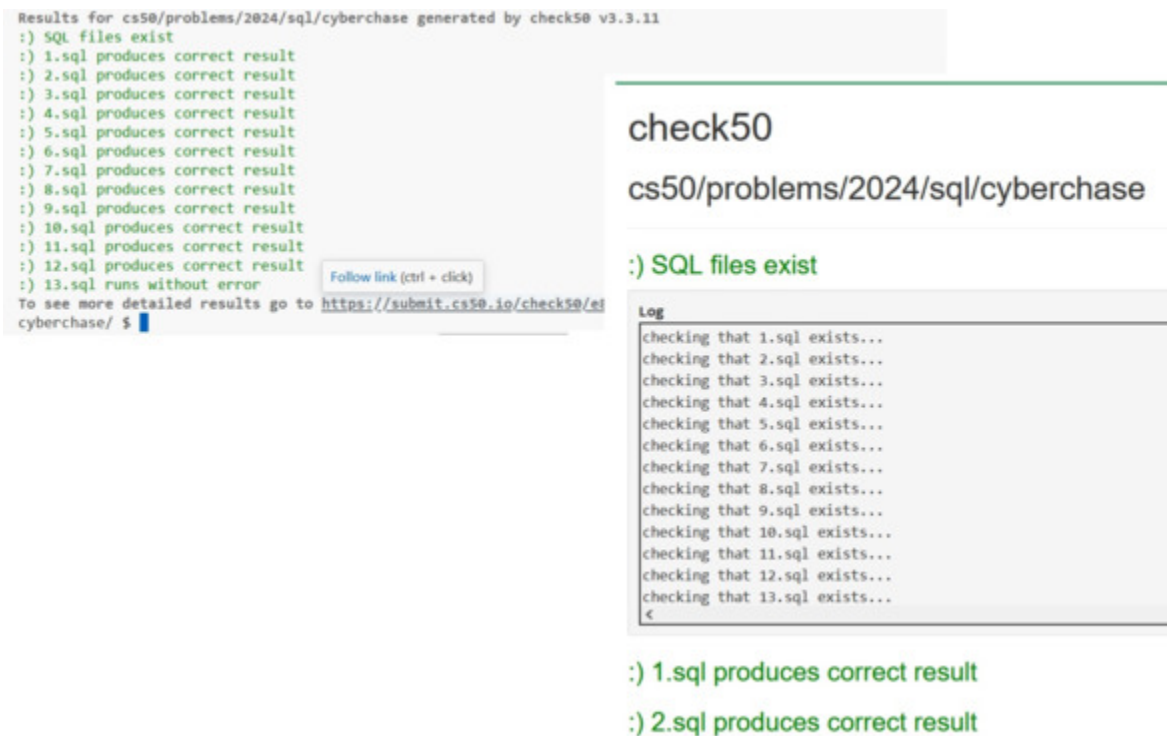


Рис. Общее тестирование решения

Окончательно отлаженное решение мы загружаем в систему в разделе How to Submit. Копируем и выполняем команду `submit50`. Результаты просматриваем в отдельной вкладке. Здесь можно убедиться, что все тесты прошли успешно. В нашем примере прошли все 14 тестов из 14 возможных: 14/14.

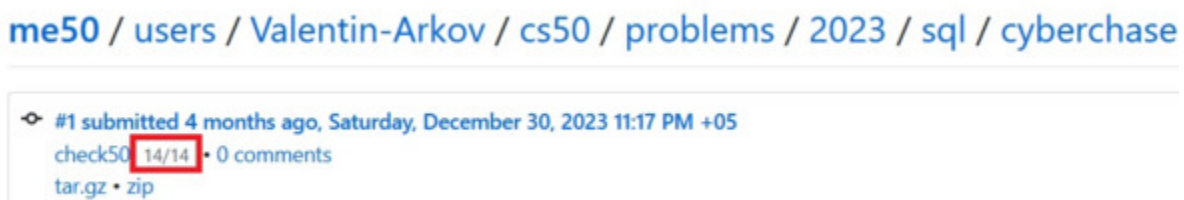


Рис. Отправка решения

После загрузки и проверки решения убеждаемся, что результаты отмечены в зачетной книжке — **Gradebook**, см. рис. Нас интересует «зеленая галочка» рядом с названием задания.

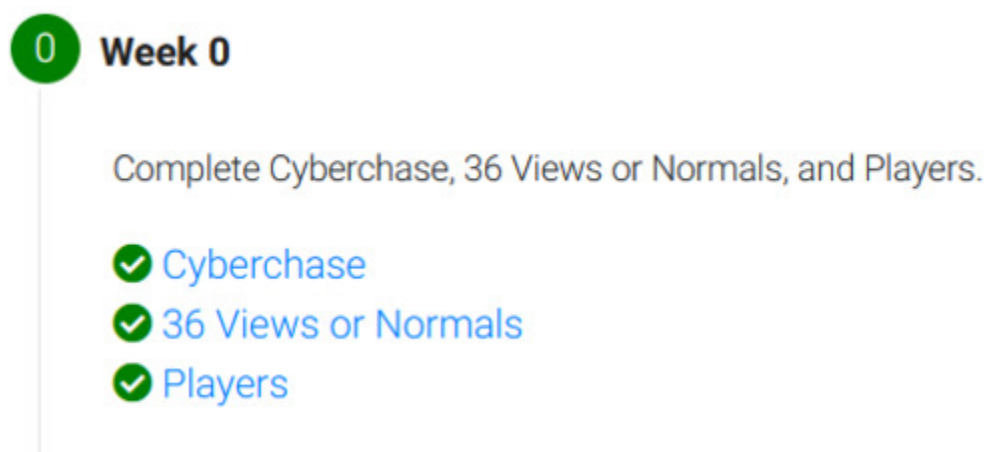


Рис. Отметки о выполнении заданий

ИТОГОВЫЙ ПРОЕКТ

После изучения материалов и выполнения практических заданий нас ожидает итоговый проект – Final Project. В этом разделе все наши действия описаны достаточно подробно. Рассмотрим только основные шаги.

Тема проекта. Нам предлагают выбрать тему, которая нам интересна. Есть примеры некоторых идей. Выберите свое любимое занятие, хобби, любую область деятельности, которая интересна именно вам. Вокруг любого занятия или объекта можно построить базу данных. Кого-то интересует посещение фитнес-зала, а кому-то интересно послушать музыку или сходить на концерт. Любое занятие окружено большим количеством данных.

Пример. Книги по базам данных. Меня интересует, какие есть книги, кто авторы, есть ли связь авторов с университетами или ИТ-компаниями, есть ли онлайн-курсы на данную тему. Окончательная формулировка темы: «Учебные материалы по базам данных».

Шаблоны файлов. В разделе Getting Started – Template Files нам понадобится скачать три шаблона: DESIGN.md, schema.sql и queries.sql.

Задание на проектирование. В разделе Specification нам предлагают заполнить эти три шаблона.

Описание проекта: DESIGN.md. Здесь мы описываем предназначение базы данных и ее структуру. Сюда включается диаграмма «сущность-связь» и видеоролик не дольше 3 минут. Ролик размещаем на Youtube или аналогичной платформе.

Файл schema.sql для создания нашей базы данных. Этот SQL запрос должен создать таблицы, индексы и представления. Запросы должны сопровождаться комментариями.

Файл queries. sql с запросами к базе данных. Здесь заполняем нашу базу и проводим по ней поиск, вносим изменения и удаляем записи.

По каждому файлу даются подробные пояснения.

Все материалы размещаем в каталоге project в облачной среде VS Code. Затем отправляем проект по инструкциям раздела How to Submit.

Когда работа принята, в зачетной книжке Gradebook появится отметка о выполнении курсового проекта Final Project. В верхней части страницы будет выведено поздравительное сообщение и ссылка на скачивание сертификата. Переходим по ссылке и входим под своей учетной записью на GitHub. На странице CS50 Certificates выбираем формат A4 / Letter, см. рис.

CS50 Certificates

Awarded from Cambridge, Massachusetts

You are logged in via GitHub as **Valentin-Arkov**. [Log out](#).

Letter ▾

CS50 SQL, generated on January 13, 2024

[50a6af4d-63f8-485c-af06-f936540a57fa](#)

- [PDF](#), for printing
- [PNG](#), for [Facebook](#), Instagram, LinkedIn, Snapchat, TikTok, [Twitter](#), and other social media

Рис. Скачивание сертификата

КУРСОВОЙ ПРОЕКТ

В рамках курса «Базы данных» мы предлагаем студентам заочного отделения выполнить итоговый проект онлайн-курса CS50 SQL и сделать его подробное описание — пояснительную записку. Более подробное, чем требуется в рамках онлайн-курса. ПЗ выполняется на русском языке и оформляется по требованиям вуза и действующим отечественным стандартам.

Защита курсового проекта — это представление пояснительной записки плюс публичное выступление перед своей группой с презентацией, ответы на вопросы и демонстрация работоспособного решения. База данных должна быть заполнена данными, она должна работать и отзываться на SQL-запросы.

НАЧАЛО ПРОЕКТИРОВАНИЯ

После выбора темы нам предстоит сориентироваться в материале. Подступиться к работе можно с разных сторон. Можно начать работу по принципу «снизу-вверх», отталкиваясь от примеров реальных данных — что мы и рекомендуем сделать в нашем курсовом проекте. Если действовать «сверху-вниз», то нужно заранее очень хорошо разобраться в материале и оставить за кадром большой объем подготовительной работы.

Итак, начинаем с примеров сведений, которые можно хранить в нашей базе данных. Затем попробуем расположить эти сведения в форме нескольких таблиц.

Поищем книги по базам данных. Если искать книги на английском языке, можно — просто в качестве примера — зайти на сайт компании Amazon, см. рис. Выбираем книги с высоким рейтингом читателей и большим числом отзывов. Обратим внимание, что Amazon в основном занимается продажей книг, а не их изданием.



Рис. Поиск популярных книг

Далее для каждой книги можно получить подробные сведения на её странице, см. рис. Нас также интересует адрес веб-страницы. Это достаточно длинная строка символов. Адрес можно сократить с помощью бесплатных сервисов. В нашем примере мы используем сервис **TinyURL**. Компания Яндекс предлагает аналогичный сервис **Кликер**:

<https://clck.ru/>.

<https://www.amazon.com/SQL-QuickStart-Guide-Simplified-Manipulating/dp/1945051752/>

SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL (QuickStart Guides™ - Technology) Paperback – Illustrated,
 November 18, 2019
 by [Walter Shields](#) (Author)
 4.6 ★★★★★ 1,570 ratings

Product details
 Publisher : ClydeBank Media LLC; Illustrated edition (November 18, 2019)
 Language : English
 Paperback : 242 pages
 ISBN-10 : 1945051752
 ISBN-13 : 978-1945051753
 Item Weight : 15.5 ounces
 Dimensions : 7.5 x 0.55 x 9.25 inches
 Best Sellers Rank: #10,560 in Books (See Top 100 in Books)
 #1 in [Microsoft SQL Server](#)
 #1 in [SQL](#)
 #3 in [Data Mining \(Books\)](#)
 Customer Reviews: 4.6 ★★★★★ 1,570 ratings


<https://tinyurl.com/app>
TINYURL
 Your Long URL
<https://www.amazon.com/SQL-QuickStart-Guide-Simplified-Manipulating/dp/1945051752/>
 TinyURL
<http://tinyurl.com/499uwez>

Рис. Сведения о выбранной книге

Нас также интересует адрес странички книги на сайте издательства. Это исходная информация о книге — в отличие от многочисленных интернет-магазинов. Адрес веб-страницы тоже сокращаем для хранения в базе данных.

Book Webpage

<https://www.quickstartguides.com/collections/best>

 TinyURL

<http://tinyurl.com/2732pd7f>




Рис. Домашняя страница книги

У любой книги есть автор — один или несколько. Находим сведения об авторе. У него может быть своя страничка в сети, см. рис. В конечном счете, можно выяснить, связан ли автор с университетом или крупной ИТ-компанией. Есть и независимые авторы, которые пишут популярные книги.

Author Details

About the author

Follow authors to get new release updates, plus improved recommendations.



Follow

Walter Shields

Walter Shields has worked with SQL and databases for over eighteen years, helping organizations such as Target successfully leverage and understand their data using SQL. While Walter's self-described path through the emergence of data science was not linear, he firmly believed that SQL did not have to be so daunting for everyone else.

Walter's desire to simplify the learning process eventually led him to start teaching students in a coffee shop in Tampa, FL.


Connect with Walter Shields on his website www.waltershields.com, on LinkedIn, or on Instagram @datashields.

BOOK RELATED CONTACT INFO:

For all inquiries related to Walters's titles or ClydeBank Media LLC (publisher of the QuickStart Guides series) please contact info@claybankmedia.com.

AUTHOR CONTACT INFO:

For all inquiries related to the author please visit www.waltershields.com.

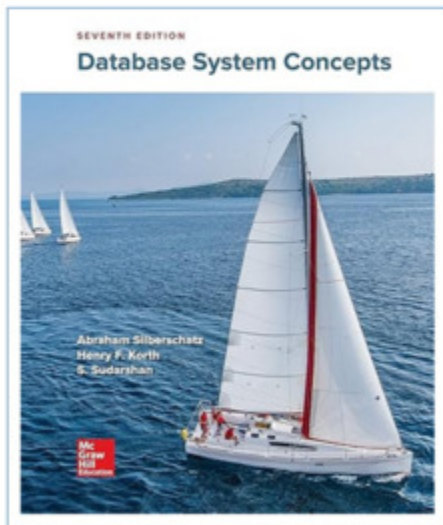


WALTER SHIELDS
DATA
ACADEMY

Рис. Страница автора книги

Есть книги, написанные преподавателями вузов. Такие книги можно назвать их университетскими учебниками, см. рис. Они переиздаются много раз. В нашем примере представлено седьмое издание одного учебника.

University Textbook



McGraw Hill PreK-12 Higher Ed Professional International About ▾

Database System Concepts, 7th Edition
ISBN10: 0078022150 | ISBN13: 9780078022159
By Abraham Silberschatz, Henry Korth and S. Sudarshan
© 2020

Product details
Publisher : McGraw Hill; 7th edition (February 19, 2019)
Language : English
Hardcover : 1376 pages
ISBN-10 : 0078022150
ISBN-13 : 978-0078022159
Item Weight : 4.37 pounds
Dimensions : 9 x 2 x 11 inches
Best Sellers Rank: #147,565 in Books ([See Top 100 in Books](#))
#11 in [Database Storage & Design](#)
#231 in [Databases & Big Data](#)
#7,481 in [Unknown](#)
Customer Reviews: 4.1 ★★★★★ ~ 54 ratings

Рис. Страница учебника

Сведения о таком авторе мы находим не только на его личной странице. Можно обнаружить страницу на сайте вуза. Для известных авторов есть сведения и в интернет-энциклопедиях типа Википедии, см. рис.

Author Webpage

The image shows a screenshot of the author's webpage for Avi Silberschatz, Professor of Computer Science at Yale University. The page includes contact information, a welcome message for the 'Database System Concepts, Seventh Edition' home page, and a list of links. To the right, there is a smaller version of the author's photo and a snippet of his Wikipedia page.

Avi Silberschatz
Professor of Computer Science at Yale University

Contact Information

Office: Watson, Room 308
Phone: (203) 432-4713
Fax: (203) 436-4918
Email: avi@yale.edu
Address: Department of Computer Science
Yale University

Welcome to the home page of **Database System Concepts, Seventh Edition.**

- [What is New in The Seventh Edition](#)
- Online book content
 - [Table of Contents](#)
 - [Preface](#)
 - [Online Chapters](#)
 - [Bibliography](#)
- [Slides](#)
- [Solutions to Practice Exercises](#)
- [Laboratory material](#)

WIKIPEDIA
The Free Encyclopedia

Scientific career

Fields	Computer Science
Institutions	Yale University
Doctoral advisor	Arthur Bernstein Richard Kieburtz
Doctoral students	C. Mohan Raghu Ramakrishnan
Website	http://www.cs.yale.edu/~avi/

Рис. Сведения об авторе

Еще один вид книг — это руководства по подготовке к сертификационным экзаменам — Exam Guide. Такие экзамены проводят крупные ИТ-компании. Чаще всего это разработчики СУБД, см. рис. Можно видеть, что автор такого руководства имеет непосредственное отношение к ИТ-компаниям.



Рис. Руководство

Следующий вид учебных материалов по базам данных – это разнообразные онлайн-курсы, см. рис. Здесь появляются самые разные сведения, включая авторов, университет или компанию, продолжительность обучения и веб-адрес.

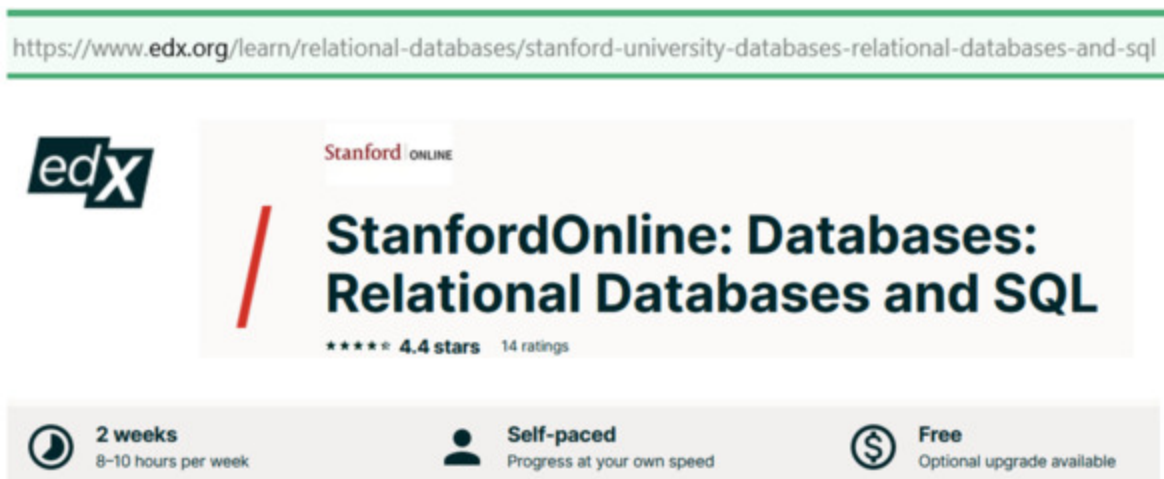


Рис. Онлайн-курс

Итак, мы провели предварительное знакомство с материалами и их источниками. Такую работу иногда называют «предпроектное обследование». То есть это обследование объекта автоматизации/информатизации.

ПРОТОТИП ТАБЛИЦ

Оставим только самые важные сведения и сформируем из них макеты (прототипы) будущих таблиц. Например, для книги оставим название и автора. Есть и другие сведения, например, год выпуска и число страниц. Добавим сюда также издательство и адреса веб-страниц, см. рис. Составим макет такой таблички с помощью Excel. Попутно мы сможем прояснить такие моменты, как название столбцов таблицы (полей базы данных) и типы данных по столбцам.

Book, Author, Organization, Course

book_title	author	year	pages	publisher	reviews	book_web	bookstore_web
SQL QuickStart Guide	W. Shields	2019	242	ClydeBank	1570	tinyurl.com/2732pd7f	tinyurl.com/499uwezf
Database System Concepts	A. Silberschatz	2019	1376	McGraw Hill	54	tinyurl.com/2ee86e6x	tinyurl.com/3n6vtsae
Oracle Database Exam Guide	J. Watson	2014	961	McGraw Hill	67	tinyurl.com/8kv3rsph	tinyurl.com/2jav5wp7
SQL All-in-One For Dummies	A. Taylor	2019	768	Wiley	917	tinyurl.com/yuf83846	tinyurl.com/4yxesx8k
The Manga Guide to Databases	M. Takahashi	2009	224	No Starch	206	tinyurl.com/yz4x4wy6	tinyurl.com/mw7cnahh

author_name	author_web	affiliation
Walter Shields	www.waltershields.com	
Abraham Silberschatz	codex.cs.yale.edu/avi/	Yale University
John Watson	tinyurl.com/yf929n48	Oracle
Christopher J. Date		IBM

org_name	org_web	org_type
Oracle	www.oracle.com	Business
Microsoft	www.microsoft.com	Business
IBM	www.ibm.com	Business
Yale	www.yale.edu	University
Stanford	www.stanford.edu	University
Harvard	www.harvard.edu	University
MIT	www.mit.edu	University

course_name	org_name	course_web	platform
CS50's Introduction to Databases with SQL	Harvard	tinyurl.com/mtsts8c	edX
Databases: Relational Databases and SQL	Stanford	tinyurl.com/bdab32nz	edX
Databases: Advanced Topics in SQL	Stanford	tinyurl.com/39y36234	edX
Introduction to SQL	IBM	tinyurl.com/3z7aashx	edX
Databases and SQL for Data Science with Python	IBM	tinyurl.com/5xhvkzsz	Coursera
Introduction to Relational Databases (RDBMS)	IBM	tinyurl.com/y2zm654d	Coursera
Database Systems	MIT	tinyurl.com/yc7mx8xs	MIT OCW

Рис. Макеты таблиц

Аналогично поступаем и другими таблицами. Это будущие информационные сущности. Попутно выясним, что некоторые ячейки наших таблиц останутся пустыми — данные отсутствуют, сведений просто не существует, или нет связи между таблицами.

Пример — независимый автор не имеет отношения ни к университетам, ни к ИТ-корпорациям.

В пояснительной записке нужно подробно описать и обосновать выбор сведений и формирование прототипов таблиц.

Связи между таблицами тоже начинают проявляться. В нашем примере мы выделяем такие таблицы, как «Книги», «Авторы», «Курсы», «Организации». Теперь мы готовы рисовать схемы и модели.

МОДЕЛИ БАЗЫ ДАННЫХ

По результатам изучения предметной области мы создаем визуальное представление структуры базы данных. Это так называемая ER-диаграмма, то есть **Entity Relationship Diagram**.

В рамках курсового проектирования нам предстоит построить две схемы.

В результате анализа предметной области мы определяем план будущей базы данных — в самых общих чертах. Нашу первую схему можно условно назвать «концептуальной». Иногда встречается наукообразный термин «инфологическая модель».

Если работать с минимальной детализацией, мы изображаем только прямоугольник — «информационную сущность», а в нем — название таблицы. На схеме показаны только основные таблицы и связи между ними, см. рис. Пока что мы не привязаны к конкретной реализации, но уже намечаем связи «один ко многим» и «многие ко многим».

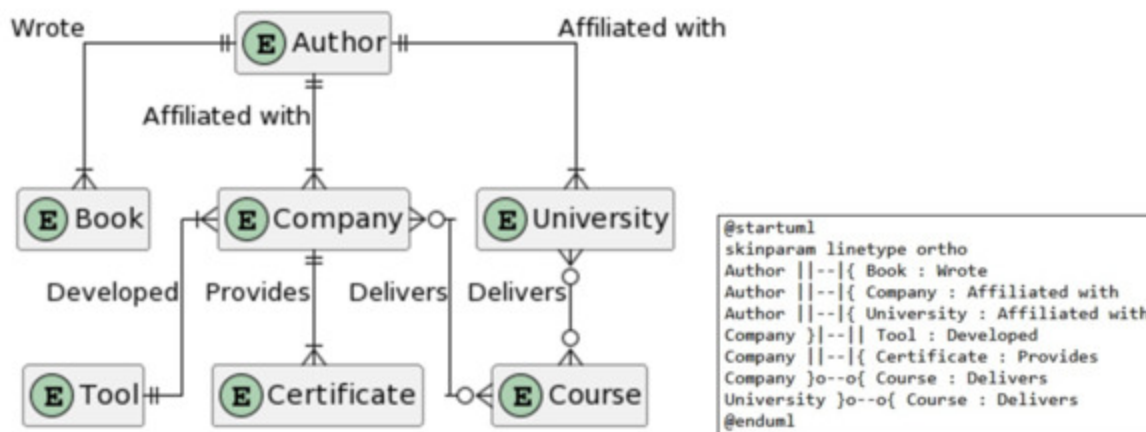


Рис. Концептуальная модель базы данных

Вторая, более подробная схема изображает все таблицы, включая вспомогательные. Мы указываем, какое поле является ключом, где у нас первичный ключ, где у нас внешний ключ. Мы указываем все будущие поля и типы данных, см. рис. Такую схему можно назвать «логической» или «инфологической» моделью.

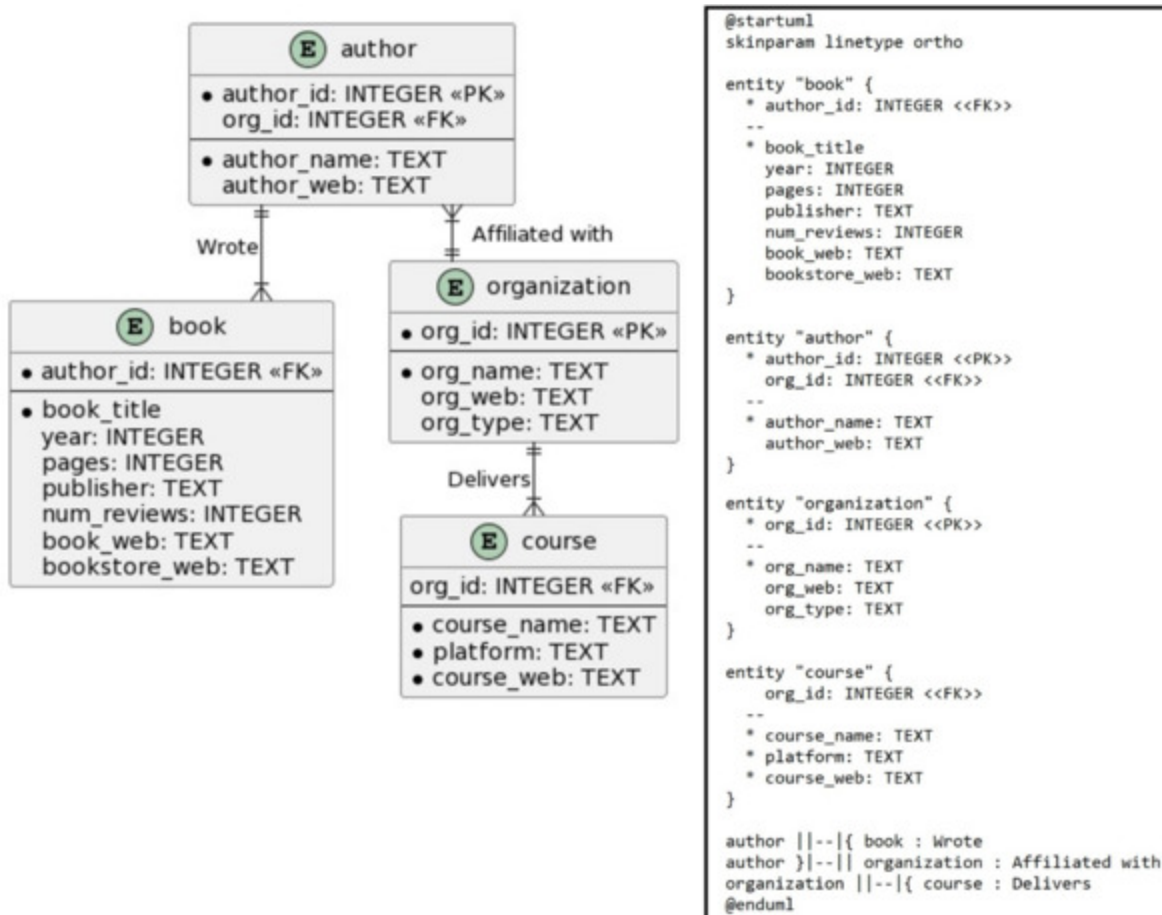


Рис. Логическая модель базы данных

Если на схеме логической модели точно указать типы данных и названия полей для окончательной реализации в конкретной СУБД, получим физическую модель базы данных. По ней уже можно написать SQL скрипт для создания структуры базы данных.

Обратим внимание, что на обоих рисунках рядом с диаграммой приводится код (скрипт) для формирования модели. В следующем

разделе мы обсудим этот инструмент.

ДИАГРАММА КАК КОД

В рамках курсового проектирования мы формируем наши модели не путем рисования мышкой на экране, а путем составления программы. На сегодняшний день развивается особый подход к построению диаграмм: Diagram as code. Диаграмма фактически представляет собой скрипт, то есть текст программы. Это обычный текстовый файл.

А с текстовым файлом можно делать то, чего нельзя сделать с графическим файлом. Можно отслеживать версии, изменения. И это уже управление версиями. Можно использовать такие инструменты, как **Git** и **Subversion**.

Кроме того, открывается возможность организовать групповую работу над проектом. Если совместно редактировать диаграмму, можно будет проследить, кто и какие изменения вносил. В том числе можно будет зафиксировать комментарии автора — что он сделал и с какой целью. Потому что как только мы вносим изменения, мы должны, мы должны сказать, что мы сделали, и кратко это описать. После этого мы можем зафиксировать (закоммитить) изменения в нашей программе — и в диаграмме. Эти технологии управления версиями изучаются в отдельном курсе, но здесь мы должны об этом аспекте хотя бы упомянуть.

Если у нас три блока и пара стрелочек, это не страшно. Но если у нас в нашей модели 20–30 сущностей и в каждой по 10 полей, то это уже большая работа. Отслеживать изменения в больших моделях лучше профессиональными инструментами, см. рис.

- Бесплатные онлайн редакторы
 - PlantUML
 - Mermaid
- Текстовый формат
 - Управление версиями
 - Совместная работа

Рис. Diagram as Code

Здесь можно использовать популярные бесплатные инструменты, такие как **PlantUML** или **Mermaid**. Мы работаем через браузер, экспортируем картинку как графический файл, и потом ее вставляем в отчет — в том числе для Гарварда. Скрипт можно также интегрировать в документ, а картинка будет сгенерирована при каждом обращении к документу.

На рис. показано, как выглядит работа в **PlantUML**. Первая и последняя строчки указывают начало и конец диаграммы: **@startuml** и **@enduml**. Внутри мы используем диаграмму классов из стандарта UML. Примеры можно найти на сайте **PlantUML**. Здесь мы создали два прямоугольника и указали связь между ними.

- <https://plantuml.com/class-diagram>

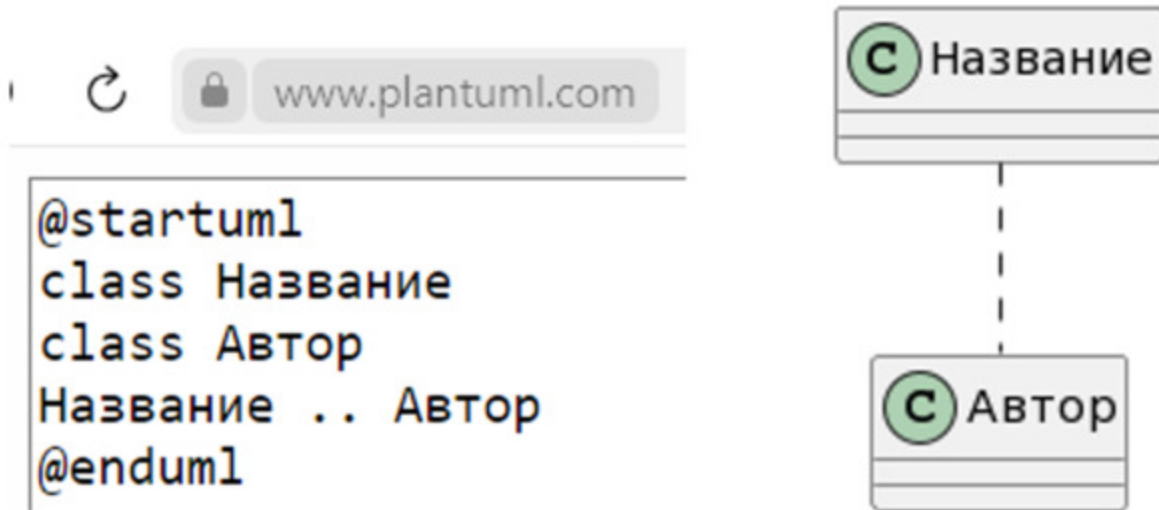


Рис. PlantUML: Class Diagram

Аналогично работает **Mermaid**. Пример кода представлен на рис. В первой строчке объявляем тип диаграммы. Далее описываем два блока (сущности) и связь между ними. Всё это сделано в одной строчке кода. На сайте есть масса примеров – Sample diagrams.

<https://mermaid.js.org/syntax/entityRelationshipDiagram.html>

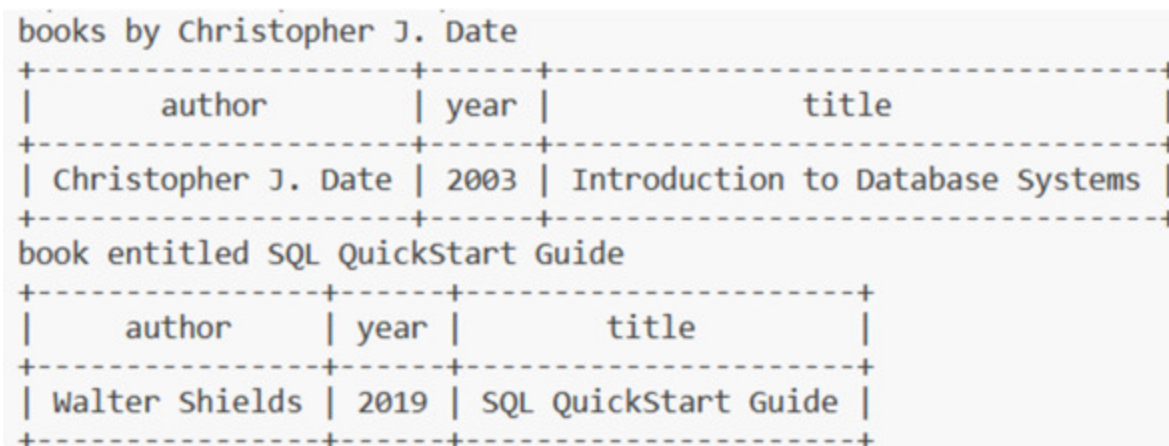


Рис. Mermaid ER Diagram

SQL ЗАПРОСЫ

Итак, мы создали базу данных и заполнили таблицы найденными сведениями. Теперь составляем различные SQL запросы. Представим, какие запросы мы сможем сделать к нашей базе данных.

Например, можно искать книги по фамилии автора, см. рис. Либо мы можем знать название книги, но не знать автора. Тогда это будет поиск по названию.



```
books by Christopher J. Date
+-----+-----+-----+
| author | year | title |
+-----+-----+-----+
| Christopher J. Date | 2003 | Introduction to Database Systems |
+-----+-----+-----+
book entitled SQL QuickStart Guide
+-----+-----+-----+
| author | year | title |
+-----+-----+-----+
| Walter Shields | 2019 | SQL QuickStart Guide |
+-----+-----+-----+
```

Рис. Поиск книг по автору или по названию

Исходя из имеющихся сведений, можно искать книги по «аффилиации» авторов — книги независимых авторов, учебники или руководства ИТ-корпораций, см. рис. Поскольку мы включили в нашу базу данных сведения об отзывах, задаем поиск книг с наибольшим количеством отзывов.

books by university professors			
author	year	title	university
Abraham Silberschatz	2019	Database System Concepts	Yale

top 5 books by reviews			
author	year	title	reviews
Walter Shields	2019	SQL QuickStart Guide	1570
Allen G. Taylor	2019	SQL All-in-One For Dummies	917
Mana Takahashi	2009	The Manga Guide to Databases	206
Christopher J. Date	2003	Introduction to Database Systems	89
Allen G. Taylor	2000	Database Development For Dummies	74

Рис. Поиск учебников и популярных книг

Другие примеры запросов – это поиск книг по году выпуска. Очевидно, современные СУБД отличаются от разработок 1970-х годов. С другой стороны, можно проводить сравнительный анализ и исторические исследования в области баз данных. Тогда книги разных годов выпуска будут представлять интерес для исследователя, см. рис.

author web pages: 3 most recent books		
recent book author	year	author web
Abraham Silberschatz	2019	http://tinyurl.com/5n72awyh
Allen G. Taylor	2019	http://tinyurl.com/4y92b3e3
Walter Shields	2019	http://tinyurl.com/2a4w5afj

Coursera courses by companies		
course on Coursera	company	web
Databases and SQL for Data Science with Python	IBM	http://tinyurl.com/5xhvkks
Introduction to Relational Databases (RDBMS)	IBM	http://tinyurl.com/y2zm654d

Рис. Поиск книг по году издания и онлайн-курсов

Следующий пример запроса — поиск онлайн-курсов. Можно искать по названию ИТ-компании, либо по продолжительности обучения, либо по названию онлайн-платформы, на которой курс был размещен.

Все SQL-запросы размещаем в одном файле.

В пояснительной записке нужно подробно описать каждый запрос и привести результаты его выполнения.

ПОДГОТОВКА ТЕКСТА

Описание проекта (на английском) для загрузки в систему будет достаточно сжатым. На сайте онлайн-курса есть шаблон этого документа.

В рамках онлайн-курса мы также готовим презентацию и видеоролик – на английском. При подготовке презентации мы можем познакомиться с несколькими инструментами. Во-первых, вы можете писать текст доклада на русском, а потом сделать машинный перевод на английский. В этом нам могут помочь те же чат-боты. Мы берем абзац текста и говорим: «Переведи, пожалуйста, на английский язык», см. рис.



Рис. Машинный перевод в чат-боте

Или мы сами пишем доклад на английском – коряво, с ошибками. А потом загружаем этот текст в чат-бот и говорим: «Пожалуйста, подправь грамматику, чтобы звучало грамотно, красиво», см. рис.

Please could you check the following text and correct mistakes: "My prodgekt is fokused on the fizikal exercizes for become fit."

✿ Answer

Please could you check the following text and correct mistakes: "My project is focused on physical exercises to become fit."

Could you paraphrase the previous text to sound like a presentation

✿ Answer

Certainly! Here's a paraphrased version suitable for a presentation:

"Our project is centered around physical exercises aimed at enhancing fitness levels."

Рис. Исправление и улучшение текста

Пояснительная записка для курса «Базы данных» составляется на русском языке, и она должна быть более подробной. Можно взять за основу шаблон оформления описания проекта и расширить его. Нужно будет описать логику построения базы данных — см. примеры рассуждений и исходных материалов в предыдущих разделах.

ОТРАБОТКА ПРОИЗНОШЕНИЯ

После того, как текст готов, нам предстоит его прочитать вслух, произнести, озвучить. И в этом нам тоже помогают интеллектуальные инструменты. Для этого есть разнообразные генераторы речи, см. рис.

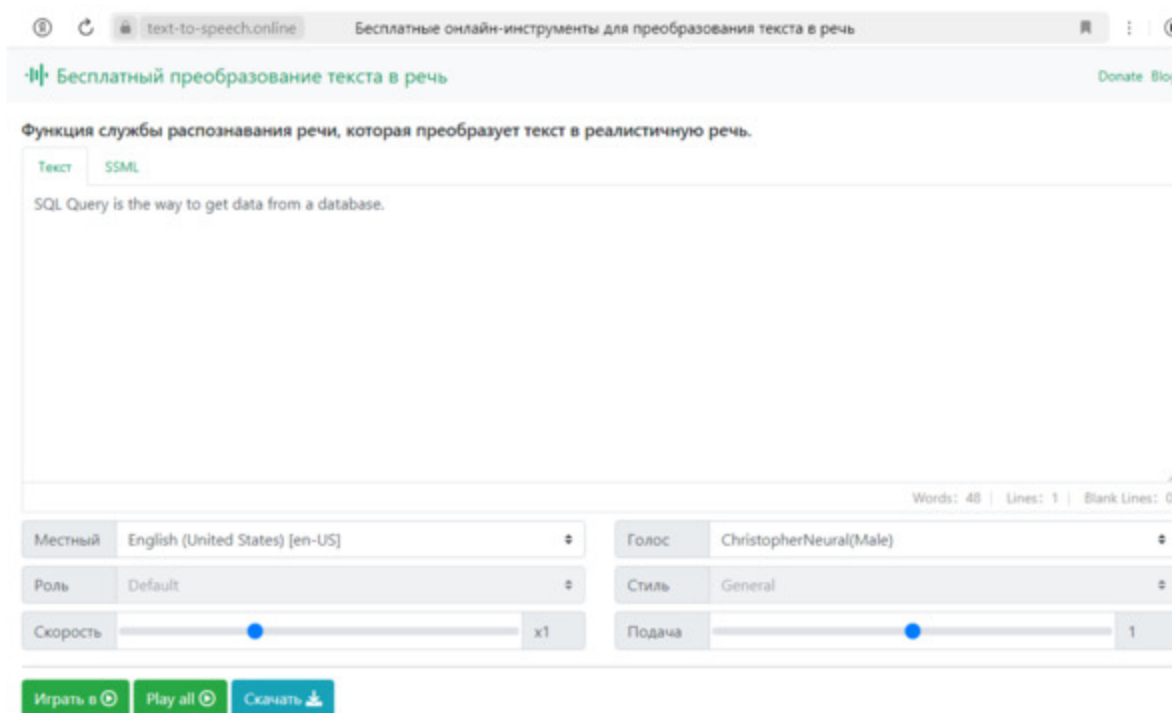


Рис. Free online text to speech converter

Таких инструментов много, можно использовать любой, но смысл здесь простой. Мы загружаем текст и слушаем, как это произносится. После этого можем сами повторить. Несколько раз послушали, повторили. Можно замедлить или ускорить. Можно выбрать акцент и диктора. Так мы фактически без репетитора, без

консультации можем освоить произношение. Этот конвертер довольно прилично работает.

MARKDOWN

Еще один инструмент, с которым мы знакомимся в рамках данного курса, — это язык разметки Markdown. Он набирает популярность в последние годы. На нем можно писать странички на GitHub, его можно вставлять в интерактивные блокноты Jupyter. И на нем мы пишем описание нашей базы данных в рамках курсового проекта, чтобы отчитаться о выполнении проекта.

Markdown — это файл с расширением MD. Внутри используется простейшая разметка. Например, символ «решетка» # — это заголовок верхнего уровня. Две решетки — заголовок следующего уровня и так далее. Чем больше решеток, тем меньше будет уровень заголовка. Можно вставлять гиперссылки, можно вставлять картинки. Это очень простой язык по сравнению с HTML, см. рис.

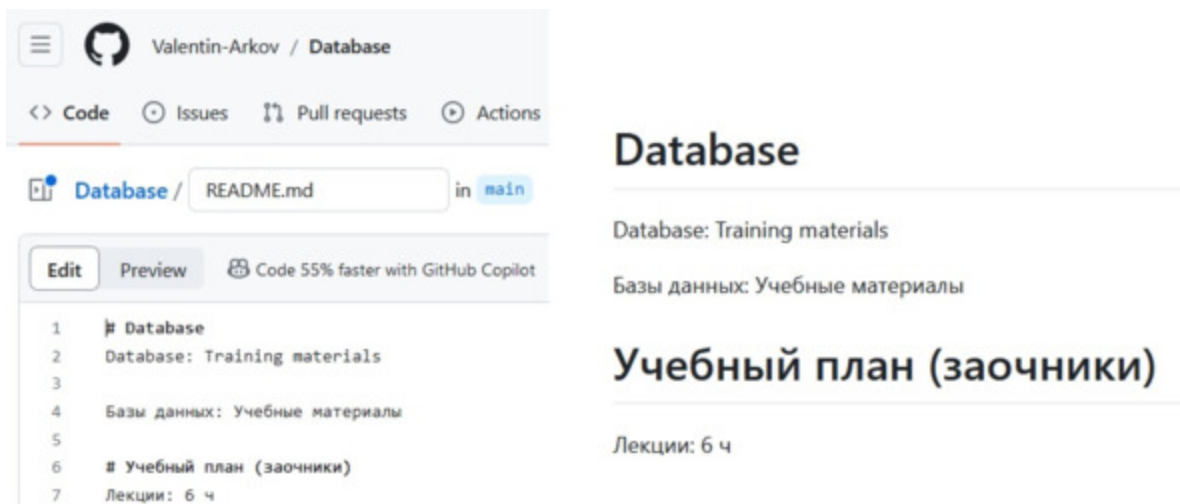


Рис. Пример разметки Markdown

ОПИСАНИЕ ПРОЕКТА

Обсудим основные разделы описания итогового проекта, как это трактуется в онлайн-курсе CS50 SQL. Фактически, это проектная документация по созданию программного продукта, но «в миниатюре». Подробнее можно почитать литературу по управлению проектами и программной инженерии. Для начала можно порекомендовать, например, популярную книгу Вигерса, см ссылки.

Документ, который составляют до начала проектирования, называется «Техническое задание» (ТЗ), или «Задание на разработку». Соответствующее международное название: «Спецификация требований» — Requirements Specification. Такой документ позволяет не только описать пожелания клиента — в форме требований и границ проекта. Описание в виде текста, диаграмм, схем и таблиц позволяет выявить слабые и проблемные места будущего продукта, улучшить взаимодействие между заинтересованными лицами (stakeholders). Кроме того, в хорошем задании нужно определить основные шаги разработки и соответствующие результаты для каждого этапа работ. Тогда можно будет отслеживать развитие событий и управлять таким проектом. При групповой работе над сложным проектом также важно фиксировать в форме документа проектные решения и их обоснование — чтобы обеспечить согласование и координацию работ. Это называется «документировать».

В процессе реализации требования могут постепенно уточняться. И это тоже нужно фиксировать в документе.

В рамках онлайн-курса описание итогового проекта — это не только задание на разработку. Это ещё и описание внутреннего

построения программного решения. А также элементы руководства пользователя. Так что здесь сочетаются несколько проектных документов – по сравнению с практикой разработки.

ШАБЛОНЫ ДОКУМЕНТОВ

Для начала посмотрим, какие файлы нам могут предложить создатели онлайн-курса. Скачиваем шаблоны документов по ссылке в разделе Final Project – Download the template files. Также будет полезно ознакомиться с примером выполнения, который можно скачать по ссылке в разделе Final Project – Sample Project. Мы не будем пересказывать примеры, их лучше посмотреть в оригинале.

SCOPE – РАМКИ ПРОЕКТА

Здесь мы явно очерчиваем границы проекта. Мы указываем, что входит в проект (included in the scope), а также что не включается и не рассматривается (out of the scope). Мы не берем на себя лишние обязательства, которые не актуальны и интересны заказчику на сегодняшний день.

Нам нужно четко определить три главных момента:

- Цель проекта
- Что входит в проект
- Что не входит в проект

Здесь мы говорим о проекте в целом, по-крупному, без подробностей.

FUNCTIONAL REQUIREMENTS – ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Здесь мы описываем функции нашего «программного продукта». Другими словами, это действия — что именно сможет делать пользователь нашей базы данных. И опять уточняем, что выходит за рамки проекта — теперь описываем действия.

Обратим внимание, что мы говорим о пользователе, для которого создается наш продукт. Даже если это сам разработчик — теперь нужно посмотреть на базу данных с точки зрения пользователя. Это особая способность — принять точку зрения, примерить на себя определенную роль.

REPRESENTATION – ПРЕДСТАВЛЕНИЕ ДАННЫХ И РЕАЛИЗАЦИЯ

Наши данные представлены в виде таблиц, а сама база данных реализована на основе СУБД SQLite. Тут будет не лишним пояснить, почему выбрана реляционная СУБД, то есть табличная база данных. Насколько табличное представление соответствует цели проекта. Можно ли будет реализовать заявленные функции и помочь пользователю решать его задачи. Потому что ранее мы заявили цель проекта, а теперь описываем средства для достижения этой цели.

ENTITIES – ИНФОРМАЦИОННЫЕ СУЩНОСТИ

В этом разделе мы описываем наши таблицы (tables). Нужно будет перечислить все поля (attributes) для каждой таблицы и тип данных.

Здесь начинается так называемое «обоснование проектных решений». Привыкаем к тому, что мы что-то решили, и теперь нужно объяснить причину. Это для себя и для других читателей документа, например, для разработчиков. Например, год издания книги записывается как целое число и дробная часть здесь не потребуется. А в других случаях дату и время записывают одним значением

с плавающей точкой — и это другой тип данных, и это тоже решение определенной задачи.

Для каждого поля желательно указать примеры значений — Example values. Это как картинка в дополнение к описанию. Как говорится, лучше один раз увидеть, чем сто раз услышать. Здесь пригодятся результаты нашего предпроектного обследования. Как мы говорили ранее. Не мешает нам сделать заготовки (макеты) будущих таблиц и заполнить в них по несколько строчек реальными данными. Эти сведения нам еще пригодятся при заполнении созданной базы данных.

Мы также указываем ограничения, которые накладываем на наши данные. Это может бы запрет на пропуски данных в определенных полях. Тогда используем ограничение NOT NULL. А для каких-то полей можем и разрешить пропуски. Например, автор книги может работать в вузе или в компании. Или же это «независимый» автор, и тогда в соответствующем поле будет пропуск, то есть NULL или NaN.

Ключи тоже являются ограничениями — для поддержания целостности данных. Когда поле объявлено как первичный ключ PRIMARY KEY, в него можно будет записывать только уникальные значения. Два одинаковых значения будет запрещено вносить.

В другой таблице мы можем указать, что поле является внешним ключом FOREIGN KEY. И это тоже ограничение. В этом случае СУБД будет контролировать соответствие между первичным и внешним ключами. Мы не сможем записать значение внешнего ключа, для которого нет соответствующего значения первичного ключа в другой таблице. И это тоже гарантирует целостность данных.

RELATIONSHIPS — СВЯЗИ МЕЖДУ ТАБЛИЦАМИ

В этом разделе мы описываем связи между таблицами. В реляционных базах данных таблицы связывают с помощью ключей KEY. Обратим внимание, что для организации связей типа

«один ко многим» и «многие ко многим» может потребоваться создать вспомогательные таблицы. И эти виды связей тоже нужно кратко обосновать. Например, один автор может выпустить несколько книг. Поэтому получаем связь «один ко многим». У одной книги может бы несколько авторов. И так далее. Эта логика сразу помогает понять «почему» для каждого проектного решения.

Для иллюстрации мы вставляем ссылку на диаграмму «сущность-связь». Вот пример такой ссылки в формате Markdown:

![Database schema: ER diagram](diagram.jpg)

Как видим, ссылка указывает на графический файл формата *.JPG. Такой файл получается при экспорте диаграммы из онлайн-редактора.

Здесь имеет смысл вставить логическую модель нашей базы данных. В ней уже подробно расписаны все таблицы и все поля вместе с типами данных. Как такие диаграммы рисовать, мы уже обсудили в предыдущих разделах.

OPTIMIZATIONS – СРЕДСТВА ОПТИМИЗАЦИИ

Чтобы ускорить обработку типовых запросов пользователей к базе данных, мы создаем индексы INDEX. Здесь нужно объяснить, какие запросы будут чаще всего выполняться. Именно для них нужно создать индексы — тоже обоснованно. Ведь создание, обновление и хранение индексов — это дополнительные затраты вычислительных ресурсов.

Второй инструмент оптимизации — это представления данных VIEW. Такие вспомогательные таблицы мы создаем с помощью SQL запросов. Затем мы используем эти представления вместо часто выполняемых запросов на поиск информации. Такое проектное решение тоже нужно будет обосновать.

LIMITATIONS – ОГРАНИЧЕНИЯ

Мы описываем ограничения нашего проекта — по данным и по реализации. Возможно, в нашей базе данных не всё можно будет хранить или не всех пользователей можно будет обслуживать. Напомним, что мы уже обозначили рамки нашего проекта — в целом. Теперь мы этот момент более подробно описываем. И не забываем про обоснование — почему мы решили ограничить наши функции и возможности.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Пояснительную записку (ПЗ) по курсовому проекту оформляем в электронном виде, поэтому можно использовать любые цвета в оформлении. Естественно, соблюдаем требования вуза по оформлению текстовых документов вообще и пояснительной записки в частности.

В начале ПЗ даем вводные сведения о проекте и приводим результаты предпроектного обследования с заполненными макетами таблиц.

Основные разделы пояснительной записки соответствуют описанию итогового проекта. В текст включаем обе модели базы данных — концептуальную и логическую.

В конце ПЗ приводим полный текст обоих файлов:

- `schema.sql`: запросы для создания базы данных (CREATE TABLE, CREATE INDEX и CREATE VIEW).

- `queries.sql`: запросы к базе данных для ввода информации, поиска сведений, изменения и удаления записей (SELECT, INSERT, UPDATE и DELETE).

ПЗ завершается заключением и списком литературы.

ПЗ в форме файла формата PDF загружаем в форму на Гитхабе. На защиту приносим распечатанный вариант ПЗ. На защите студент делает доклад по выполненной работе на пять минут и демонстрирует работающую версию базы данных, откликающуюся на SQL-запросы, приведенные в ПЗ.

ССЫЛКИ

Учебные материалы Арькова В. Ю.

<https://github.com/Valentin-Arkov/Database>

Harvard CS50 SQL

<https://cs50.harvard.edu/sql/>

GitHub

<https://github.com>

EdX

<https://www.edx.org/>

SQL

<https://ru.wikipedia.org/wiki/SQL>

NoSQL

<https://ru.wikipedia.org/wiki/NoSQL>

Переводчик Google

<https://translate.google.ru/>

Переводчик Яндекса

<https://translate.yandex.ru/>

Переводчик DeepL

<https://www.deepl.com/>

YouGlish

<https://youglish.com/>

Perplexity

<https://www.perplexity.ai/>

Руководство по Markdown

<https://www.markdownguide.org/>

СУБД SQLite

<https://www.sqlite.org/>

Notepad++

<https://notepad-plus-plus.org/>

SQLite Online

<https://sqliteonline.com/>

Google Co-Laboratory

<https://colab.research.google.com/>

Википедия: Проектирование баз данных

https://ru.wikipedia.org/wiki/Проектирование_баз_данных

Git

<https://ru.wikipedia.org/wiki/Git>

Subversion

<https://ru.wikipedia.org/wiki/Subversion>

PlantUML: Class Diagram

<https://plantuml.com/class-diagram>

Mermaid ER Diagram

<https://mermaid.js.org/syntax/entityRelationshipDiagram.html>

Пример видеоролика по проекту

<https://youtu.be/T6JXscp442A>

CS50 on Youtube

<https://www.youtube.com/cs50>

SQL BASICS – CS50 on Twitch, EP. 6

<https://youtu.be/hAexYxe2YZU>

GigaChat от Сбера

<https://developers.sber.ru/gigachat/>

Вигерс К., Битти Дж. Разработка требований к программному обеспечению. — СПб.: БХВ-Петербург, 2014. — 736 с.

<https://bhv.ru/product/razrabotka-trebovanij-k-programmnomu-obespecheniyu-3-izd/>