Национальный исследовательский ядерный университет «МИФИ» (Московский Инженерно–Физический Институт)
Кафедра №42 «Криптология и кибербезопасность»

Лабораторная работа №1-2: «Работа с данными. Простые запросы на выборку»

Цель работы

Приобретение навыков внесения исходных данных в базу данных с поддержкой SQL и извлечения нужных данных из отдельных таблиц.

Ход работы

1. Подготовить данные заполнения таблиц, разработанных в рамках лабораторной работы 1-1. Занести их в базу данных при помощи SQL-инструкции INSERT. Можно использовать клиент sqlite3.exe или любой другой на Ваше усмотрение;
2. Разработать и реализовать не менее 8 SQL-запросов в соответствии с легендой в Вашей предметной области. Какие данные из таблиц могут пригодиться Вашим сотрудникам, и при каких условиях? При реализации запросов необходимо использовать возможности, описанные в следующем разделе;
3. Выполнить эти запросы в реализованной и заполненной базе данных, сохранить результаты выполнения;
4. Оформить отчёт.

Основные возможности простых запросов на выборку

1. Полная выборка (SELECT, FROM). Извлекает значения запрошенных столбцов для всех доступных записей. Например, следующая инструкция служит для получения списка клиентов:

   SELECT name FROM clients;

2. Условная выборка (WHERE). Извлекает значения запрошенных столбцов только для записей, удовлетворяющих условию. Например, следующая инструкция служит для получения списка клиентов, родившихся после 2000 года:

   SELECT name FROM clients WHERE (year_birth > 2000);

3. Упорядоченная выборка (ORDER, DESC). После извлечения сортирует записи по значению в указанном столбце. Например, следующая инструкция служит для получения списка клиентов, отсортированного по возрасту, начиная с самых старых:

   SELECT name FROM clients ORDER BY year_birth;

   …с самых молодых:

   SELECT name FROM clients ORDER BY year_birth DESC;

4. Выборка с использованием простых функций и операторов. RDBMS может применить к запрошенным значениям столбцов набор функций и/или операторов. Например, этот запрос позволяет получить таблицу с величиной чистой прибыли предприятия по кварталам:

   SELECT quarter, (income - expense) AS profit FROM finance;

   …получить зарплату сотрудников с округлением до целых (вторым параметром указывается число разрядов после запятой, до которого надо округлить число. В SQLite3 отрицательное значение эквивалентно нулю):

   SELECT full_name, round(salary, -3) FROM employees;

5. Выборка с использованием агрегатных функций (GROUP BY). RDBMS собирает данные в группы по указанному столбцу, и возвращает по одной записи на каждую группу. При этом, ко всем значениям столбца внутри группы применяется агрегатная функция. Например, этот запрос возвращает максимальную зарплату сотрудника в каждом отделе:

SELECT department, max(salary) FROM employees GROUP BY department;

...этот запрос возвращает число записей, которые содержатся в таблице grades для каждого студента и сортирует их по убыванию:

SELECT student, count(*) FROM grades GROUP BY student ORDER BY count(*) DESC;

Важно: если аргумент функции count() равен нулю, этот ряд игнорируется. При выполнении count(*) все ряды всегда подсчитываются, так как в каждой строке в реляционной базе данных обязательно содержится хотя бы одно ненулевое значение.

6. Выборка уникальных записей (UNIQUE, DISTINCT). Частный случай группировки записей.

Если запрос возвращает несколько одинаковых записей, RDBMS оставляет только одну. Например, этот запрос возвращает список городов, в которых у компании есть клиенты:

SELECT UNIQUE city FROM clients;

7. Псевдо-выборка: если необходимо не получить данные из таблицы, а просто оценить значение выражения, в SQLite3 можно сделать выборку, не указывая наименования таблицы.

SELECT (3+5);

Этот запрос возвращает таблицу, состоящую из одной строки и одного столбца, содержащую значение 8. В других СУБД это может выполняться иначе. Например, в Oracle Database для этого служит системная таблица dual, содержащая незначимую величину в единственной ячейке:

SELECT (round(1782,-2)) FROM dual;

8. VACUUM

Операции записи в SQLite3 оптимизированы для наибольшего быстродействия (без ущерба целостности). Грубо говоря, это достигается за счёт записи изменённых данных в конец файла и объявлением старых данных недействительными. Как следствие, при продолжительном использовании база данных может фрагментироваться и занимать больший объём места, нежели необходимо. В этом случае можно вручную использовать SQL-инструкцию VACUUM, которая производит обслуживание и очистку физических структур хранения данных. Это рекомендуется делать перед архивированием или пересылкой файла базы данных. Заметьте, что VACUUM — это нестандартная инструкция SQL, а не команда клиента SQLite3. Все команды клиента начинаются с точки, и не требуют точки с запятой в конце.

Оформление отчёта

1. Титульный лист: название института, название лабораторной работы, имя, фамилия, номер группы, год,...
2. Список выполненных простых запросов SQL с описаниями их смысла и ожидаемых результатов, а также результатами их выполнения на хранящихся в БД данных;
3. Листинг использованных инструкций SQL;
4. Заключение: краткое описание проделанной работы.

Справочные материалы

1. Т. Кайт. Oracle для профессионалов. Oracle Database нам больше не интересна, но эта книга (особенно в первых главах) содержит очень много полезной информации о принципах устройства и функционирования СУБД, и потому рекомендуется к прочтению.

2. SQL As Understood By SQLite. https://www.sqlite.org/lang.html. Официальная документация по диалекту языка SQL, поддерживаемому SQLite3. Описания инструкций, рельсовые диаграммы.

3. NULL Handling in SQLite Versus Other Database Engines. https://www.sqlite.org/nulls.html Справочные материалы по обработке значений типа NULL в SQLite3 и других распространённых РСУБД. В общем случае, значение NULL можно понимать как «неизвестно» или «не заполнено». При работе с ним, важно понимать особенности его обработки в составе инструкций и выражений.

4. Built-In Scalar SQL Functions. https://www.sqlite.org/lang_corefunc.html. Справочник по всем обычным (скалярным) функциям, которые можно применить в SQLite3. Ассортимент и поведение функ-

ций может существенно различаться в разных моделях РСУБД, поэтому важно работать с документацией той системы, для которой разрабатывается SQL-сценарий. Скалярные функции применяются для преобразования одного значения в другое.

5. Built-in Aggregate Functions. https://www.sqlite.org/lang_aggfunc.html Справочник по всем агрегатным функциям, которые можно применить в SQLite3. Агрегатные функции применяются для преобразования множества значений в одно, и применяются при группировке строк.