

Лабораторная работа № 2. Создание БД для приложения

Одним из основных применений РСУБД является хранение и обработка данных небольших пользовательских приложений. Специфика такого использования заключается в том, что данные постоянно добавляются/изменяются/удаляются. При этом данных сравнительно немного. Схема данных достаточно устойчива и редко изменяется. В этом случае непосредственно с СУБД взаимодействует не человек, а программа, что уменьшает требования к способу взаимодействия с базой данных. Однако разработчику необходимо уметь проектировать гибкую и эффективную схему данных, использовать ограничения целостности, манипулировать данными и понимать механизмы поддержки согласованности базы данных, такие как транзакции и триггеры.

Данная лабораторная работа призвана сформировать у студента понимание особенностей хранения данных приложения в РСУБД и умение это хранение настраивать и поддерживать.

1. Проектирование схемы базы данных

Постановка задачи

Второе практическое задание связано с проектированием схемы базы данных для работы приложения (WEB/Mobile/Desktop). Каждый индивидуальный вариант содержит предметную область, из которой должна быть проектируемая база данных. К данной предметной области необходимо добавить не менее 2х дополнительных таблиц, необходимых для детального решения поставленной задачи. Задачей студента является решить, для чего будет использоваться создаваемая база данных, и, исходя из этого, построить её концептуальную схему. Результатом данной части лабораторной работы является схема базы данных (в виде ER-диаграммы, содержащей таблицы и связи между ними, с уточнением типов столбцов, с описанием внешних и первичных ключей). При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

<https://www.lucidchart.com/pages/examples/er-diagram-tool>

<https://dbeaver.io/>

Темы для проработки

- Модель "сущность-связь" (ER-модель).
- Первичные и внешние ключи.
- Типы связей и их моделирование.
- Нормальные формы и нормализация.

Требования к схеме

- Схема должна соответствовать поставленной задаче.
- Связи между сущностями должны быть правильно смоделированы.
- Таблицы должны удовлетворять, по крайней мере, третьей нормальной форме.
- Желательно придерживаться какой-либо системы в именовании таблиц и столбцов.

2. Создание и заполнение таблиц

Постановка задачи

Необходимо подготовить SQL-скрипт для создания таблиц согласно разработанной схеме, полученной в предыдущем задании (с уточнением типов столбцов). Необходимо определить первичные и внешние ключи, а также декларативные ограничения целостности (возможность принимать неопределенное значение, уникальные ключи, проверочные ограничения и т. д.). Таблицы следует создавать в отдельной базе данных. Кроме того, нужно подготовить данные для заполнения созданных таблиц. Объем подготовленных данных должен составлять не менее 10 экземпляров для каждой из стержневых сущностей и 1000 экземпляров для целевой сущности. На основе этих данных необходимо создать SQL-скрипт для вставки соответствующих строк в таблицы БД.

Темы для проработки

- Язык DDL, операторы CREATE TABLE и ALTER TABLE. <https://postgrespro.ru/docs/postgrespro/11/ddl-basics> <https://postgrespro.ru/docs/postgrespro/11/ddl-default> <https://postgrespro.ru/docs/postgrespro/11/ddl-alter>
- Типы данных. <https://postgrespro.ru/docs/postgrespro/11/datatype>
- Декларативные ограничения целостности. <https://postgrespro.ru/docs/postgrespro/11/ddl-constraints>
- Оператор INSERT. <https://postgrespro.ru/docs/postgrespro/11/dml-insert> <https://postgrespro.ru/docs/postgrespro/11/dml-returning>
- Полное описание синтаксиса встретившихся команд <https://postgrespro.ru/docs/postgrespro/11/sql-commands>

Вопросы по текущей лабораторной работы

- Объяснить, что делают написанные запросы.
- В чем различие типов CHAR и VARCHAR? VARCHAR и TEXT?
- Что такое внешний ключ?
- Какие существуют способы поддержания ссылочной целостности?
- Что такое уникальный ключ?
- Что такое SERIAL?
- Рассказать о значениях по умолчанию и неопределенных значениях.
- Как можно хранить даты и время?
- Рассказать о числовых типах данных.
- Каким образом можно вставить несколько строк с помощью одного оператора INSERT?
- Как ведет себя оператор INSERT, если в списке столбцов перечислены не все столбцы?
- Добавить какие-либо ограничения целостности.
- Добавить SERIAL.
- Исправить выявленные при проверке недочеты.

3. Операторы манипулирования

Постановка задачи

Данный раздел посвящен манипулированию данными с помощью операторов SQL. В ходе выполнения задания необходимо:

- Подготовить 3-4 выборки, которые имеют осмысленное значение для предметной области, и также составить для них SQL-скрипты.
- Сформулировать 3-4 запроса на изменение и удаление из базы данных. Запросы должны быть сформулированы в терминах предметной области. Среди запросов обязательно должны быть такие, которые будут вызывать срабатывание ограничений целостности. Составить SQL-скрипты для выполнения этих запросов.

Темы для проработки

Оператор SELECT.

<https://postgrespro.ru/docs/postgrespro/11/queries>

Оператор UPDATE.

<https://postgrespro.ru/docs/postgrespro/11/dml-update>

Оператор DELETE.

<https://postgrespro.ru/docs/postgrespro/11/dml-delete>

Декларативные ограничения целостности.

<https://postgrespro.ru/docs/postgrespro/11/ddl-constraints>

Полное описание синтаксиса встретившихся команд <https://postgrespro.ru/docs/postgrespro/11/sql-commands>

Примеры вопросов

- Объяснить, как работают написанные запросы.
- Исправить неверно работающий запрос (запросы).
- Упростить один или несколько запросов.
- Написать или модифицировать запрос по сформулированному заданию.

4. Контроль целостности данных

Постановка задачи

Данный раздел посвящен контролю целостности данных, который производится с помощью механизма транзакций и триггеров. Транзакции позволяют рассматривать группу операций как единое целое, либо отрабатывают все операции, либо ни одной. Это позволяет избежать несогласованности данных. Триггеры позволяют проверять целостность данных в момент выполнения транзакций, поддерживать целостность, внося изменения, и откатывать транзакции, приводящие к потере целостности.

Необходимо подготовить SQL-скрипты для проверки наличия аномалий (потерянных изменений, грязных чтений, неповторяющихся чтений, фантомов) при параллельном исполнении транзакций на различных уровнях изолированности SQL/92 (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE). Подготовленные скрипты должны работать с одной из таблиц, созданных в практическом задании №2.1. Для проверки наличия аномалий потребуются два параллельных сеанса, операторы в которых выполняются пошагово:

1. Установить в обоих сеансах уровень изоляции READ UNCOMMITTED. Выполнить сценарии проверки наличия аномалий потерянных изменений и грязных чтений.
2. Установить в обоих сеансах уровень изоляции READ COMMITTED. Выполнить сценарии проверки наличия аномалий грязных чтений и неповторяющихся чтений.
3. Установить в обоих сеансах уровень изоляции REPEATABLE READ. Выполнить сценарии проверки наличия аномалий неповторяющихся чтений и фантомов.
4. Установить в обоих сеансах уровень изоляции SERIALIZABLE. Выполнить сценарий проверки наличия фантомов.

Необходимо составить скрипт для создания триггера, а также подготовить несколько запросов для проверки и демонстрации его полезных свойств:

- Изменение данных для сохранения целостности.
- Проверка транзакций и их откат в случае нарушения целостности.

Темы для проработки

- Понятие транзакции, свойства транзакций.
<https://postgrespro.ru/docs/postgrespro/11/tutorial-transactions>
- Уровни изолированности и аномалии
<https://postgrespro.ru/docs/postgrespro/11/transaction-iso>
- Триггеры и триггерные функции
<https://postgrespro.ru/docs/postgrespro/11/trigger-definition>
<https://postgrespro.ru/docs/postgrespro/11/plpgsql-trigger>
- Сообщения и ошибки
<https://postgrespro.ru/docs/postgrespro/11/plpgsql-errors-and-messages>
- Полное описание синтаксиса встретившихся команд
<https://postgrespro.ru/docs/postgrespro/11/sql-commands>

Примеры вопросов

- Рассказать об аномалиях доступа к БД.
- Перечислить аномалии, возникающие на каждом из уровней изолированности.
- Рассказать о свойствах транзакций.
- Рассказать об управлении транзакциями.
- Что такое тупики? Как бороться с тупиками?
- На каком уровне изолированности возможны тупики?
- Как обеспечивается изолированность транзакций в СУБД?
- Как бороться с проблемой фантомов?
- Что такое журнал транзакций?
- Как обеспечивается постоянство хранения (durability) в СУБД?
- Объяснить принцип работы написанного триггера.
- Какие бывают типы триггеров?
- Когда может срабатывать триггер?
- В каком порядке срабатывают триггеры?
- Можно ли менять порядок срабатывания триггеров?
- Сработает ли триггер, если оператор, выполненный пользователем, не затрагивает ни одну строку таблицы?
- Продемонстрировать откат транзакции при возникновении ошибок.
- Продемонстрировать возникновение тупика.
- Исправить неверные сценарии проверки аномалий
- Исправить ошибки в работе триггера.
- Модифицировать триггер каким-либо образом.