**Лабораторное занятие №11.**

**Создание таблиц просмотров.**

**Лабораторное (практическое) занятие** рассчитано на 2 часа, относится к теме «»

**Формируемые компетенции:** ОК.1-9 ПК 2.1 - 2.4

**Цель:** Изучить используемые в реляционных СУБД операторы создания и удаления представлений. Получить навыки работы с представлениями с помощью команд SQL и с помощью программы "SQLiteStudio".

**Методическое и техническое обеспечение:** IBM PC-совместимый компьютер, SQLiteStudio.

## Исходные данные

Студент получает индивидуальный вариант исходных данных с кратким описанием предметной области, который используется при выполнении всех лабораторных работ. При этом каждая очередная лабораторная работа является продолжением выполненной ранее и поэтому они должны обязательно выполняться последовательно. Вариант задания приведен в лабораторной работе №1.

**Теоретические сведения**

**Представления (Views)**

Представление представляет собой запрос на выборку, которому присвоили имя и сохранили в базе данных. Другими словами это виртуальная таблица, которая реально не содержит данных, а всяких раз, когда требуется его содержимое, происходит выполнение запроса.

Когда СУБД встречает в инструкции SQL ссылку на представление, она находит его определение, сохраненное в базе данных, преобразует пользовательский запрос, ссылающийся на представление, в эквивалентный запрос к исходным таблицам представления и выполняет этот запрос. Таким образом, СУБД создает иллюзию существования представления в виде отдельной таблицы.

Причины использования представлений

1. Они изолируют реализацию хранения данных от пользователя. Если клиенты будут работать с данными через представление, то у разработчика базы данных появляется возможность менять запрос, лежащий в основе представления, а клиент ничего не будет замечать.

2. С их помощью можно ограничить доступ к данным, разрешая определенным пользователям иметь права на чтение/изменение данных в представлении, но даже не иметь понятия о таблицах, лежащих в основе этого представления.

3. Они упрощают доступ к базе данных, показывая пользователям структуру хранимых данных в наиболее подходящем для них виде.

Представления могут основываться на данных из нескольких таблиц и других представлений. Представления можно использовать как обычные таблицы. Оператор **SELECT**, образующий представление, может делать то же, что и обычный оператор **SELECT**, кроме включения предложения **ORDER BY** и использования в качестве источника данных хранимых процедур.

Синтаксис оператора создания представления:

**CREATE VIEW** Имя\_Представления [**(**Колонка [, Колонка ...]**)**]

**AS** <select> [**WITH CHECK OPTION**];

<select> - оператор SELECT, который выбирает данные включаемые в представление

**WITH CHECK OPTION** - если указать этот необязательный параметр, то при вставке или изменении строки представления будет проверяться условие "попадания" этой строки в представление. Если новая или измененная запись не удовлетворяет условиям запроса, на котором основано представление, то операция будет отменена и возникнет ошибка.

Пример создания представления:

**CREATE VIEW** ViewStudent (St\_ID, LastName, FirstName)

**AS SELECT** Pr\_ID, Pr\_LastName, Pr\_FirstName

**FROM** Person, Student

**WHERE** Pr\_ID = St\_ID;

Чтобы изменить какое-либо представление, его надо удалить и создать заново. При удалении представления необходимо также удалить все зависимые от его объекты - триггеры, хранимые процедуры и другие представления. Программа "IB Expert" пересоздает зависимые объекты автоматически.

Пример удаления представления:

**DROP VIEW** ViewStudent;

**Модифицируемые представления**

Для некоторых представлений вставку, удаление или обновление строки можно преобразовать в эквивалентные операции по отношению к исходным таблицам этого представления. Могут существовать представления, для которых добавление, удаление или обновление строк не имеет смысла. Такие представления можно только читать.

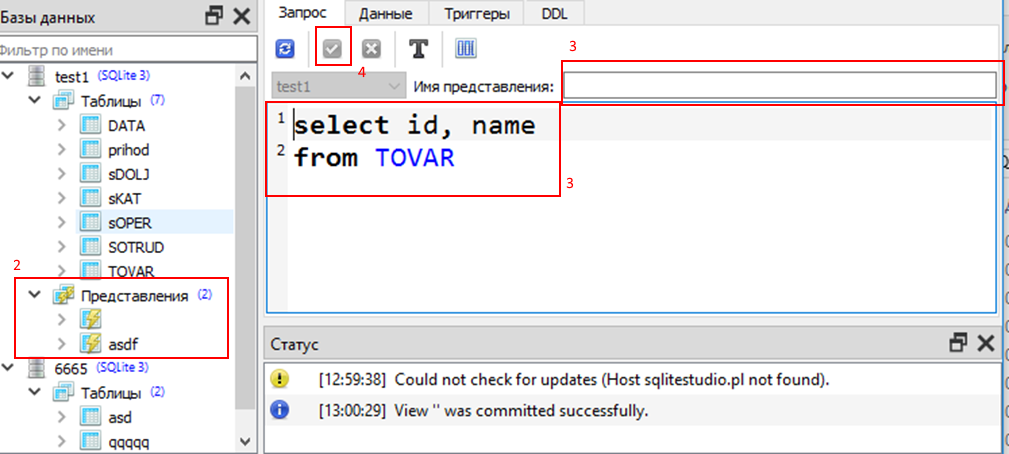
Содержимое представления может изменяться командами модификации, которые будут автоматически перенаправлены к базовой таблице. Чтобы представление было модифицируемым, оно должно быть создано на основе одной таблицы (или другого модифицируемого представления), а запрос, на котором основано представление, не должен содержать подзапросов, агрегатных функций, UDF, хранимых процедур, предложений **DISTINCT**, **GROUP BY** и **HAVING**. Если выполняются все эти условия, то представление автоматически становится модифицируемым, т. е. для него можно выполнять запросы DELETE, INSERT и UPDATE, которые будут изменять данные в таблице-источнике.

Чтобы сделать модифицируемым представление, которое нарушает любое из вышеперечисленных условий, применяется механизм триггеров. Триггеры для модификации представления должны срабатывать при событиях BEFORE DELETE, BEFORE UPDATE и BEFORE INSERT. В теле триггеров необходимо описать, что должно происходить с данными при удалении, изменении и вставке.

***Ход работы***

Для создания представлений в диалоговом режиме программы "SQLiteStudio" необходимо выполнить следующие действия:

1. Подключиться к базе данных.
2. В левой колонке выбрать «View» («Представления») -> Create a view.
3. В отрывшемся окне вписать имя представления и сам программный код.
4. После этого нажать на зеленую кнопку.

Рис. 1: Окно ввода оператора создания представления

В результате в диалоговое окно «Статус» выведется сообщение об успешном создание представлении или об ошибке в коде.

***Задание***

Работу следует выполнять в следующем порядке:

1. Открыть в приложении " SQLiteStudio ".

2. Зарегистрировать созданную базу данных в программе " SQLiteStudio" и подключиться к ней.

3. Создать в своей базе данных несколько представлений в диалоговом режиме, которое не должно быть модифицируемым. Оно должно называться "TestView" и основываться, например, на нескольких таблицах или других представлениях.

***Отчет о выполнении работы***

Отчет о выполнении необходимо оформить на листах формата A4. Отчет должен содержать описание и результаты работы, представляемые в следующей последовательности:

1. Словесное описание и сценарий создания представлений для своей базы данных.

2. Краткое описание запросов с использованием всех созданных представлений. Распечатка этих запросов с указанием количества возвращенных ими строк.