МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

**«Московский технический университет связи и информатики»**

**(МТУСИ)**

Кафедра

«Сетевые информационные технологии и сервисы»

Лабораторная работа 3

По дисциплине «Интеллектуальные базы данных»

«Вставка, удаление и обновление данных»

Выполнил студент:

М092301(75) Леонов Н.Н.

Вариант 17

Проверил:

Ст.пр. Фатхулин Т.Д.

Москва 2024

1. **Цель работы**

В данной лабораторной работе необходимо заполнить согласованными данными таблицы БД, а затем исправить часть введенной информации.

1. **Задание**

При выполнении лабораторной работы необходимо для заданной предметной области средствами MySQL:

* заполнить согласованными данными таблицы БД;
* при необходимости исправить введенную информацию;
* составить отчет по лабораторной работе.

1. **Теоретические сведения**

Рассмотрим следующие вопросы:

* вставка данных с помощью оператора *INSERT*;
* удаление данных операторами *DELETE* и *TRUNCATE*;
* обновление данных с помощью оператора *UPDATE*.

После создания БД и таблиц перед разработчиком встает задача заполнения таблиц данными. В реляционных БД традиционно применяют три подхода:

* однострочный оператор *insert* – добавляет в таблицу новую запись;
* многострочный оператор *insert* – добавляет в таблицу несколько записей;
* пакетная загрузка *LOAD DATA INFILE* – добавление данных из файла.

**Вставка данных с помощью оператора *INSERT***. Однострочный оператор *insert* может использоваться в нескольких формах. Упрощенный синтаксис первой формы:

*insert [IGNORE] [INTO] имя\_таблицы [(имя\_столбца, ... )]*

*VALUES (выражение, ... );*

Оператор вставляет новую запись в таблицу *имя\_таблицы*. Значения полей записи перечисляются в списке *(выражение, ... )*. Порядок следования столбцов задается списком *(имя\_столбца, ... )*. Список столбцов *(имя\_столбца, ... )* позволяет менять порядок следования столбцов при добавлении.

Первичный ключ таблицы является уникальным, и попытка добавить уже существующее значение приведет к ошибке. Чтобы новые записи с дублирующим ключом отбрасывались без генерации ошибки, следует добавить после оператора *insert* ключевое слово *IGNORE*.

Другая форма оператора *insert* предполагает использование слова *set*:

*insert [IGNORE] [INTO] имя\_таблицы*

*SET имя\_столбца1 = выражение1, имя\_столбца2 = выражение2, ... ;*

Оператор заносит в таблицу *имя\_таблицы* новую запись, столбец *имя\_столбца* в которой получает значение *выражение*.

Многострочный оператор *INSERT* совпадает по форме с однострочным оператором, но после ключевого слова *values* добавляется через запятую несколько списков *(выражение, ... )*.

Практические примеры использования оператора *insert* для заполнения учебной БД *book* см. ниже, в пункте «Пример выполнения работы».

**Удаление данных.** Для удаления записей из таблиц предусмотрены:

* оператор *DELETE;*
* оператор *TRUNCATE TABLE.*

Оператор *DELETE* имеет следующий синтаксис:

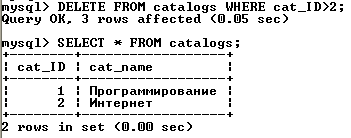
*DELETE FROM имя\_таблицы*

*[Where* *условие]*

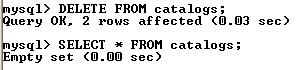
*[ORDER BY имя\_поля]*

*[LIMIT число\_строк];*

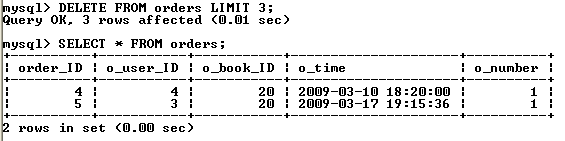
Оператор удаляет из таблицы *имя\_таблицы* записи, удовлетворяющие условию. В следующем примере из таблицы *catalogs* удаляются записи, имеющие значение первичного ключа *catalog\_id* больше двух.



Если в операторе отсутствует условие *where*, удаляются все записи таблицы.

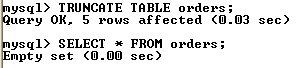


Ограничение *limit* позволяет задать максимальное число записей, которые могут быть удалены. Следующий запрос удаляет все записи таблицы *orders*, но не более 3 записей.



Конструкция *order by* обычно применяется вместе с ключевым словом *limit*. Например, если необходимо удалить 20 первых записей таблицы, то производится сортировка по полю типа *datetime* – тогда в первую очередь будут удалены самые старые записи.

Оператор *truncate table* полностью очищает таблицу и не допускает условного удаления. Он аналогичен оператору *delete* без условия *where* и ограничения *limit*. Удаление происходит гораздо быстрее, т. к. осуществляется не перебор записей, а полное очищение таблицы.



**Обновление данных.** Обновление данных (изменение значений полей в существующих записях) обеспечивают:

* оператор *Update*;
* оператор *Replace*.

Оператор *UPDATE* позволяет обновлять отдельные поля в существующих записях. Имеет следующий синтаксис

*Update [IGNORE] имя\_таблицы*

*SET имя\_столбца1= выражение1 [, имя\_столбца2 = выражение2 … ]*

*[WHERE условие]*

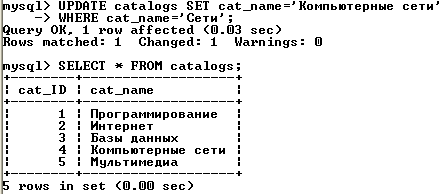
*[ORDER BY имя\_поля ]*

*[LIMIT число\_строк] ;*

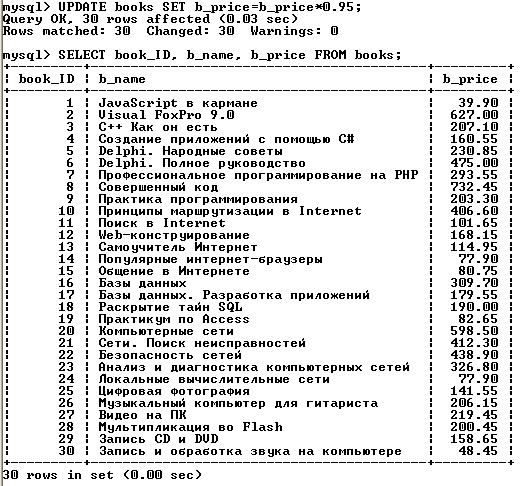
После ключевого слова *update* указывается таблица, которая изменяется. В предложении *set* указывается, какие столбцы обновляются и устанавливаются их новые значения. Необязательное условие *WHERE* позволяет задать критерий отбора строк (обновляться будут только строки, удовлетворяющие условию).

Если указывается необязательное ключевое слово *ignore*, то команда обновления не будет прервана, даже если при обновлении возникнет ошибка дублирования ключей. Строки, породившие конфликтные ситуации, обновлены не будут.

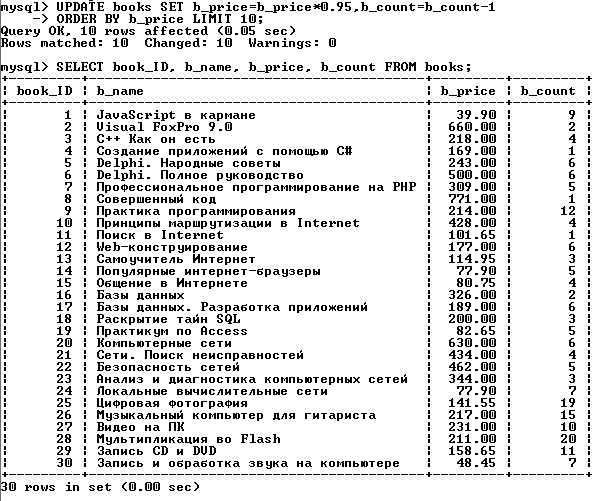
Запрос, изменяющий в таблице *catalogs* «Сети» на «Компьютерные сети».



Обновлять можно всю таблицу. Пусть требуется уменьшить на 5 % цену на все книги. Для этого следует старую цену в руб­лях умножить на 0,95.



Инструкции *limit* и *order by* позволяют ограничить число изменяемых записей. При этом за один запрос можно обновить несколько столбцов таблицы. Например, необходимо в таблице *books* для десяти самых дешевых товарных позиций уменьшить количество книг на складе на единицу, а цену – на 5 %.



Оператор *REPLACE* работает как оператор *insert,* за исключением того, что старая запись с тем же значением индекса *unique* или *primary key* перед внесением новой будет удалена. Если не используются индексы *unique* или *primary key,* то применение оператора *replace* не имеет смысла.

Синтаксис оператора *REPLACE* аналогичен синтаксису оператора *insert*:

*REPLACE* *[INTO] имя\_таблицы [(имя\_столбца, ... )]*

*VALUES (выражение, ... )*

В таблицу вставляются значения, определяемые в списке после ключевого слова *VALUES*. Задать порядок столбцов можно при помощи необязательного списка, следующего за именем таблицы. Как и оператор *Insert*, оператор *replace* допускает многострочный формат.

1. **Выполнение лабораторной работы**

Заполним таблицы базы данных Cost Accounting данными и внесем некоторые изменения. Заполнение таблицы данными осуществляется с помощью команды INSERT INTO … VALUES (…), где после INTO указывается название таблицы, а в скобках непосредственно данные.

Заполнение таблицы Department и результат представлен на рисунке 1.

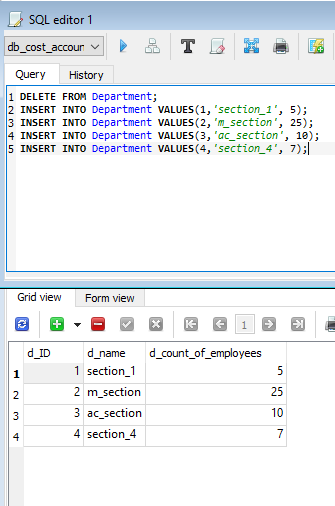


Рисунок 1 – Ввод данных в таблицу Department

Заполнение таблицы Employee и результат представлен на рисунке 2.

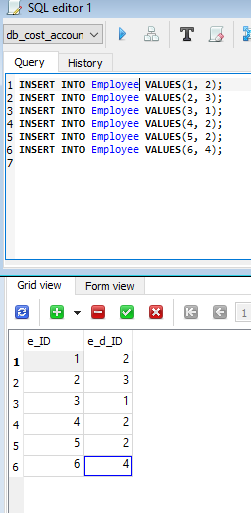


Рисунок 2 – Ввод данных в таблицу Employee

Заполнение таблицы Expenses и результат представлен на рисунке 3.

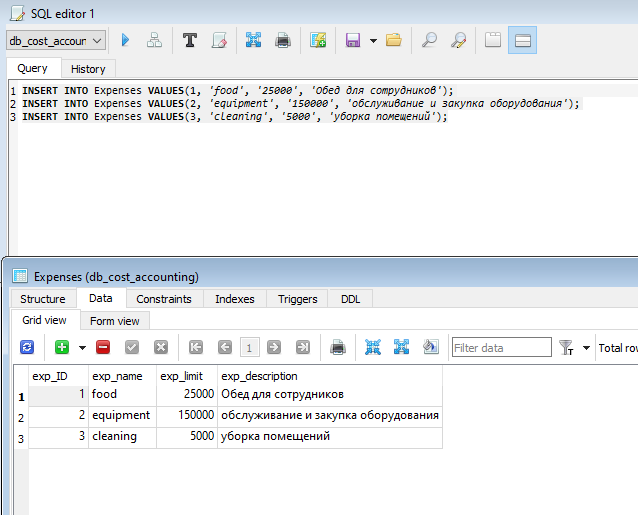


Рисунок 3 – Ввод данных в таблицу Expenses

Заполнение таблицы Purchase и результат представлен на рисунке 4.

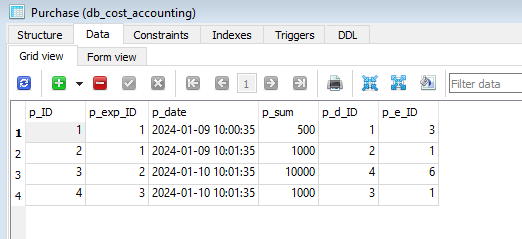


Рисунок 4 – Ввод данных в таблицу Purchase

Удаление сотрудника с ID = 4 представлено на рисунке 5.

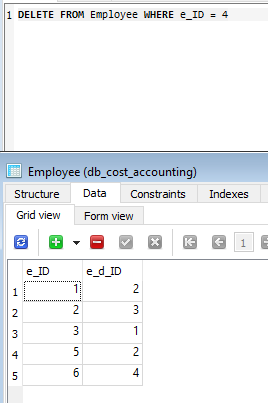
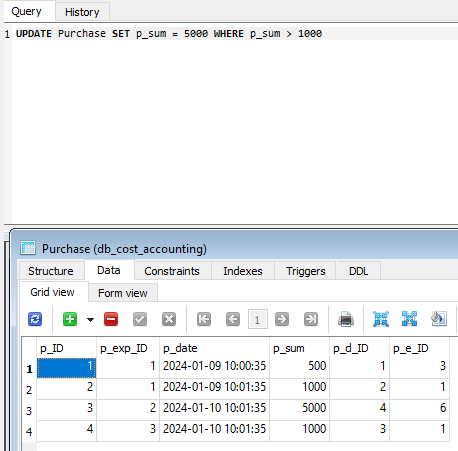


Рисунок 5 – Результат удаления из таблицы Employee

Изменим цену покупки, цена которой больше 1000 на 5000 с помощью команды UPDATE … SET … (рис. 6).



# **Выводы**

В ходе лабораторной работы были заполнены согласованными данными таблицы созданной ранее базы данных. Были рассмотрены и применены основные команды для обновления, удалениями вставки информации в соответствующих таблицах. В результате работы сданными в таблицах были получены и продемонстрированы результаты выполнения использованных команд.