

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.

## 2. 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.

## 3. 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.

```
mysql> DELETE FROM catalogs WHERE cat_ID>2;
Query OK, 3 rows affected (0.05 sec)

mysql> SELECT * FROM catalogs;
+-----+-----+
| cat_ID | cat_name      |
+-----+-----+
|      1 | Программирование |
|      2 | Интернет      |
+-----+-----+
2 rows in set (0.00 sec)
```

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

```
mysql> DELETE FROM catalogs;
Query OK, 2 rows affected (0.03 sec)

mysql> SELECT * FROM catalogs;
Empty set (0.00 sec)
```

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

```
mysql> DELETE FROM orders LIMIT 3;
Query OK, 3 rows affected (0.01 sec)

mysql> SELECT * FROM orders;
+-----+-----+-----+-----+-----+
| order_ID | o_user_ID | o_book_ID | o_time                | o_number |
+-----+-----+-----+-----+-----+
|      4   |      4   |      20   | 2009-03-10 18:20:00   |      1   |
|      5   |      3   |      20   | 2009-03-17 19:15:36   |      1   |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

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.

```
mysql> TRUNCATE TABLE orders;
Query OK, 5 rows affected (0.03 sec)

mysql> SELECT * FROM orders;
Empty set (0.00 sec)
```

**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».

```
mysql> UPDATE catalogs SET cat_name='Компьютерные сети'  
-> WHERE cat_name='Сети';  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM catalogs;  
+-----+-----+  
| cat_ID | cat_name |  
+-----+-----+  
| 1 | Программирование |  
| 2 | Интернет |  
| 3 | Базы данных |  
| 4 | Компьютерные сети |  
| 5 | Мультимедиа |  
+-----+-----+  
5 rows in set (0.00 sec)
```

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

```
mysql> UPDATE books SET b_price=b_price*0.95;
Query OK, 30 rows affected (0.03 sec)
Rows matched: 30  Changed: 30  Warnings: 0
```

```
mysql> SELECT book_ID, b_name, b_price FROM books;
```

book_ID	b_name	b_price
1	JavaScript в кармане	39.90
2	Visual FoxPro 9.0	627.00
3	C++ Как он есть	207.10
4	Создание приложений с помощью C#	160.55
5	Delphi. Народные советы	230.85
6	Delphi. Полное руководство	475.00
7	Профессиональное программирование на PHP	293.55
8	Совершенный код	732.45
9	Практика программирования	203.30
10	Принципы маршрутизации в Internet	406.60
11	Поиск в Internet	101.65
12	Web-конструирование	168.15
13	Самоучитель Интернет	114.95
14	Популярные интернет-браузеры	77.90
15	Общение в Интернете	80.75
16	Базы данных	309.70
17	Базы данных. Разработка приложений	179.55
18	Раскрытие тайн SQL	190.00
19	Практикум по Access	82.65
20	Компьютерные сети	598.50
21	Сети. Поиск неисправностей	412.30
22	Безопасность сетей	438.90
23	Анализ и диагностика компьютерных сетей	326.80
24	Локальные вычислительные сети	77.90
25	Цифровая фотография	141.55
26	Музыкальный компьютер для гитариста	206.15
27	Видео на ПК	219.45
28	Мультипликация во Flash	200.45
29	Запись CD и DVD	158.65
30	Запись и обработка звука на компьютере	48.45

```
30 rows in set (0.00 sec)
```

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 %.

```
mysql> UPDATE books SET b_price=b_price*0.95,b_count=b_count-1
-> ORDER BY b_price LIMIT 10;
Query OK, 10 rows affected (0.05 sec)
Rows matched: 10  Changed: 10  Warnings: 0
```

```
mysql> SELECT book_ID, b_name, b_price, b_count FROM books;
```

book_ID	b_name	b_price	b_count
1	JavaScript в кармане	39.90	9
2	Visual FoxPro 9.0	660.00	2
3	C++ Как он есть	218.00	4
4	Создание приложений с помощью C#	169.00	1
5	Delphi. Народные советы	243.00	6
6	Delphi. Полное руководство	500.00	6
7	Профессиональное программирование на PHP	309.00	5
8	Совершенный код	771.00	1
9	Практика программирования	214.00	12
10	Принципы маршрутизации в Internet	428.00	4
11	Поиск в Internet	101.65	1
12	Web-конструирование	177.00	6
13	Самоучитель Интернет	114.95	3
14	Популярные интернет-браузеры	77.90	5
15	Общение в Интернете	80.75	4
16	Базы данных	326.00	2
17	Базы данных. Разработка приложений	189.00	6
18	Раскрытие тайн SQL	200.00	3
19	Практикум по Access	82.65	5
20	Компьютерные сети	630.00	6
21	Сети. Поиск неисправностей	434.00	4
22	Безопасность сетей	462.00	5
23	Анализ и диагностика компьютерных сетей	344.00	3
24	Локальные вычислительные сети	77.90	7
25	Цифровая фотография	141.55	19
26	Музыкальный компьютер для гитариста	217.00	15
27	Видео на ПК	231.00	10
28	Мультипликация во Flash	211.00	20
29	Запись CD и DVD	158.65	11
30	Запись и обработка звука на компьютере	48.45	7

```
30 rows in set (0.00 sec)
```

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.

#### 4. 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.

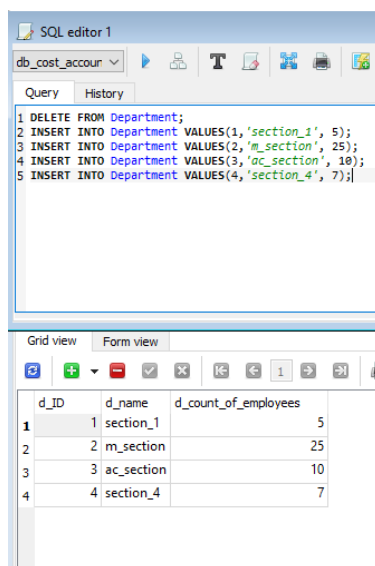


Рисунок 1 – Ввод данных в таблицу Department

Заполнение таблицы Employee и результат представлен на рисунке 2.

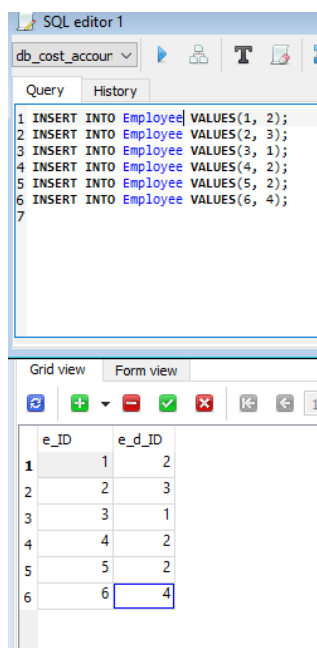


Рисунок 2 – Ввод данных в таблицу Employee



Заполнение таблицы Expenses и результат представлен на рисунке 3.

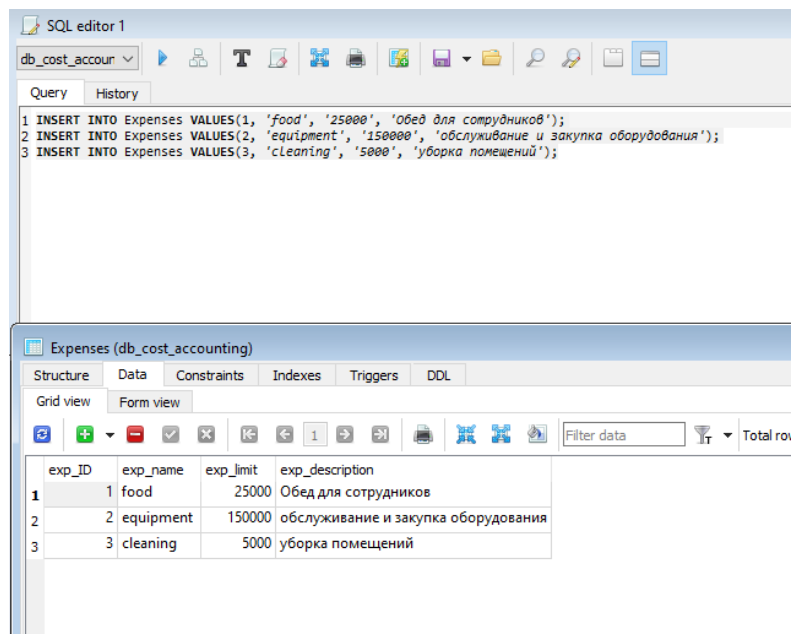


Рисунок 3 – Ввод данных в таблицу Expenses

Заполнение таблицы Purchase и результат представлен на рисунке 4.

The screenshot shows the 'Purchase (db\_cost\_accounting)' table in grid view:

p_ID	p_exp_ID	p_date	p_sum	p_d_ID	p_e_ID
1	1	2024-01-09 10:00:35	500	1	3
2	2	2024-01-09 10:01:35	1000	2	1
3	3	2024-01-10 10:01:35	10000	4	6
4	4	2024-01-10 10:01:35	1000	3	1

Рисунок 4 – Ввод данных в таблицу Purchase

Удаление сотрудника с ID = 4 представлено на рисунке 5.

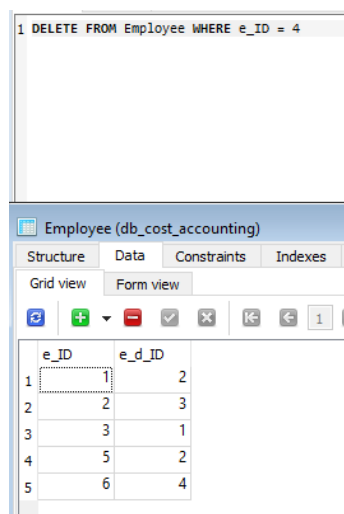
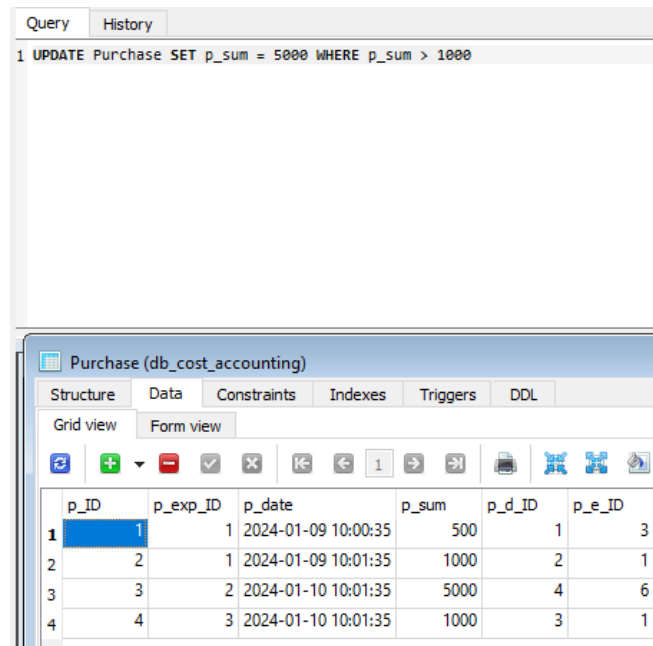


Рисунок 5 – Результат удаления из таблицы Employee

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, удаления и вставки информации в соответствующих таблицах. В результате работы с данными в таблицах были получены и продемонстрированы результаты выполнения использованных команд.