МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ”

Специальность 2-40 01 01

Учебный предмет Базы данных и системы

управления базами данных

УТВЕРЖДАЮ

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_М.М. Федоськова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЛАБОРАТОРНАЯ РАБОТА № 8**

**Исследование работы команд для обновления и удаления данных в таблице**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

Разработал преподаватель

Базанова А.Д.

2022

Обсуждено и одобрено

на заседании цикловой комиссии

специальностей в области

программного обеспечения

информационных систем

Протокол № \_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 1 Цель работы

1.1 Формирование умений использовать команды UPDATE и DELETE для обновления и удаления данных в таблицах

**2 Методическое и материальное обеспечение**

2.1 Методические рекомендации по выполнению лабораторной работы

2.2 Персональный компьютер IBM PC

2.3 СУБД MySQL

**3 Последовательность выполнения работы**

3.1 Изучить теоретическое обоснование лабораторной работы

3.2 Выполнить индивидуальное задание

3.3 Составить ответы на контрольные вопросы

3.4 Составить отчет

**4 Теоретическое обоснование**

UPDATE — это команда, которая обновляет данные в таблице. Ее общий синтаксис такой:

UPDATE [table] table\_name

SET column1 = value1, column2 = value2, ...

[WHERE condition]

[ORDER BY expression [ ASC | DESC ]]

[LIMIT number\_rows];

Сначала мы указываем обязательные параметры: название таблицы, названия колонок и нужные значения для обновления. Обратите внимание, что в MySQL можно использовать ключевое слово table (update table), а можно его опустить и сразу указать название таблицы.

Затем идут необязательные блоки WHERE (условие обновления), ORDER BY (сортировка) и LIMIT (ограничение количества обновляемых записей).

Мы будем рассматривать работу с командой UPDATE на примере простой схемы БД. Представим, что есть сеть магазинов бытовой техники. База данных состоит из 3 таблиц:

– categories: таблица категорий товаров. В ней хранятся только идентификаторы и названия категорий (рисунок 1).

– stores: таблица магазинов. В ней хранятся идентификаторы, названия магазинов, город и адрес(рисунок 2).

– products: таблица товаров. В ней хранятся идентификаторы, названия товаров, ссылки на категорию товара и магазин, цена товара и его количество(рисунок 3).

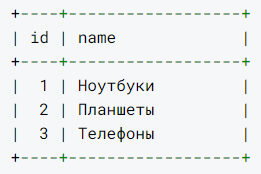


Рисунок 1 – Таблица «categories»

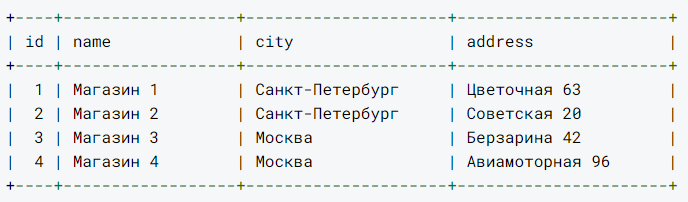


Рисунок 2 – Таблица «stores»

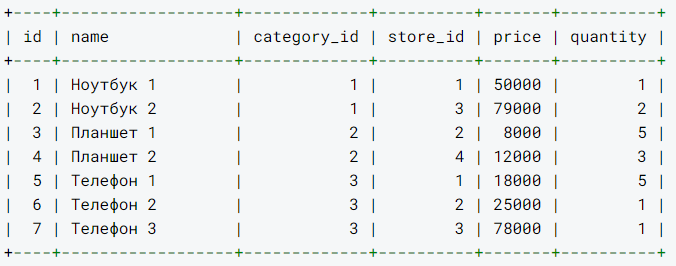


Рисунок 3 – Таблица «products»

Допустим, необходимо изменить стоимость одного конкретного товара. Для этого указываем имя MySQL таблицы (update products), название обновляемого поля и значение (set price = 50500), а также условие, какую именно стоку нужно обновить (where id = 1):

UPDATE products

SET price = 50500

WHERE id = 1;

Также есть возможность обновить несколько полей в одной строке. Например, у товара одновременно изменилась стоимость и количество. Вместо того, чтобы писать два отдельных запроса, мы укажем сразу оба изменяемых поля и их значения:

UPDATE products

SET price = 78500, quantity = 3

WHERE id = 2;

Также можно обновить сразу несколько строк в одной таблице, или даже все. Например, мы хотим выровнять количество всех товаров во всех магазинах, и установить его равным трем. Для этого нам нужно просто опустить блок WHERE, и тогда оператор UPDATE применит все, что указано в блоке SET сразу ко всем строкам в таблице:

UPDATE products

SET quantity = 3;

Вместо конкретного значения можно использовать выражение, на основании которого будет вычисляться значение. Например, в магазинах проходит акция, и нужно снизить цены всех ноутбуков на 10%. Рассчитывать значение вручную для каждого товара неудобно, поэтому мы используем выражение. Укажем, что цену нужно умножить на 0.9, то есть сделать ее равной 90% от начальной:

UPDATE products

SET price = (price \* 0.9);

В условии WHERE можно использовать данные, полученные из других таблиц. Расширим предыдущий пример с выражением. В этот раз акция проходит только в магазине с названием «Магазин 2». Для этого нужно в условии WHERE указать идентификатор магазина, для которого нужно обновить цены. Но вместо того, чтобы искать в таблице идентификатор и потом подставлять его в SQL-запрос, мы можем указать название, а идентификатор подставится сам.

Используем вложенный оператор SELECT, который сначала вернет идентификатор нужного нам магазина, а затем подставит его в блок WHERE:

UPDATE products

SET price = (price \* 0.9)

WHERE store\_id = (

SELECT id

FROM stores

WHERE name = 'Магазин 2'

);

Команда UPDATE может обновить значения сразу в нескольких таблицах за один раз. Допустим, мы хотим обновить адрес одного из магазинов, и тут же обновить количество товара в нем:

UPDATE stores, products

SET stores.address = 'Пятницкая 23', products.quantity = 3

WHERE stores.id = 4

and products.store\_id = 4;

Иногда значение в блоке SET может быть не явным, а зависеть от какого-либо условия. Например, мы хотим уменьшить цены на все ноутбуки на 100р. Мы уже знаем, как это можно сделать с помощью условия WHERE. А теперь покажем, как то же самое можно сделать с помощью условного оператора IF:

UPDATE products

SET price = IF(category\_id=1, price-100, price);

Усложним пример. Теперь нам нужно уменьшить цены на ноутбуки на 100р, на планшеты поднять на 100р, а телефоны — уменьшить на 5%. Для этого лучше подойдет другой условный оператор — CASE. В нем мы можем перечислить сразу несколько условий:

UPDATE products SET price = CASE

WHEN category\_id = 1 THEN price-100

WHEN category\_id = 2 THEN price+100

WHEN category\_id = 3 THEN price\*0.95

END;

Простой синтаксис оператора **DELETE** в MySQL:

DELETE FROM table

[WHERE conditions];

Полный синтаксис оператора DELETE в MySQL:

DELETE [ LOW\_PRIORITY ] [ QUICK ] [ IGNORE ] FROM table

[WHERE conditions]

[ORDER BY expression [ ASC | DESC ]]

[LIMIT number\_rows];

Рассмотрим пример запроса MySQL DELETE, в котором у нас есть только одно условие в операторе DELETE

DELETE FROM products

WHERE name = 'Телефон 1';

Рассмотрим MySQL пример DELETE, где у нас есть только два условия в инструкции DELETE:

DELETE FROM products

WHERE quantity = 3

AND price < 20000;

Также можно выполнять более сложные удаления. Вы можете удалить записи в одной таблице на основе значений в другой таблице. Поскольку вы не можете перечислить более чем одну таблицу в MySQL операторе FROM при выполнении удаления, вы можете использовать MySQL оператор EXISTS.

DELETE FROM products

WHERE EXISTS

( SELECT \*

FROM stores

WHERE stores.id = products.id\_store

AND stores.city = ‘Москва’ );

**5 Индивидуальное задание**

5.1 Создать таблицы с рисунков 1 – 3. Заполнить таблицы данными в соответствии с рисунками 1 – 3.

5.2 Реализовать запросы, приведенные в теоретическом обосновании.

5.3 Для созданных таблиц из лабораторной работы №6, создать по три разнообразных запроса на обновление и удаление

**6 Содержание отчета** **(в электронном виде, в рабочей папке учащегося)**

6.1 Тема работы

6.2 Цель работы

6.3 Выполненное индивидуальное задание

6.4 Ответы на контрольные вопросы

1. **Контрольные вопросы**
   1. Как обновить несколько полей в одной строке?
   2. С помощью какого запроса можно обновить значения сразу в нескольких таблицах за один раз?
   3. Можно ли перечислить несколько таблиц в операторе FROM при выполнении удаления? Как произвести удаление записи в одной таблице на основе значений в другой таблице?

**Список использованных источников**

1 Дюбуа П. MySQL/ П. Дюбуа.– 2-е издание – М.: Вильямс, 2004. – 198-211 с.

2 Хомоненко, А.Д. Базы данных: учеб./ А.Д. Хомоненко [и др.]: под ред. проф. А.Д. Хомоненко. Санкт-Петербург, 2003, с. 277-364.