

## Лабораторная работа №2. Редактирование структуры БД. Манипуляция данными. Нормализация.

*Базы данных.*

*2курс. 5 группа.*

*Кушнеров А.В. 2020-2021 г.*

*Уровень сложности: Средний*

*Формат работы: Индивидуальная по вариантам.*

*Срок выполнения: 2 недели.*

### Цель работы

Получение углубленных знаний по DDL. Приобретение навыков в манипулировании данными БД, а также в нормализации БД.

### Минимальные теоретические сведения

**DDL** (Data Definition Language) – язык описания данных. Часть SQL операторов, отвечающая за определение данных и описание их структуры. Основные операторы: CREATE, ALTER, DROP.  
**DML** (Data Manipulation Language) – операторы манипулирования данными. Основные представители: INSERT, UPDATE, DELETE, SELECT.

Разработчик БД всегда должен иметь возможность редактировать уже созданные таблицы отношений. В СУБД MySQL предусмотрены соответствующие операторы.

ALTER TABLE – ключевое слово для вызова различных модификаторов редактирования структуры таблицы.

#### **Пример 1:**

Начнём с простейшей операции добавления и удаления столбца. Предположим, нам необходимо снабдить таблицу products из БД bar столбцом с указанием страны-производителя.

```
ALTER TABLE products  
ADD COLUMN product_country VARCHAR(30) NOT NULL;
```

С такой же лёгкостью можем удалить добавленный столбец.

```
ALTER TABLE products  
DROP COLUMN product_country;
```

Обратите внимание, что при таком вызове столбец всегда будет добавлен в конец таблицы. Для того, чтобы поставить его в определённое место, необходимо использовать ключевые слова AFTER или FIRST. Более того, разработчик имеет возможность добавить сразу несколько столбцов.

```
ALTER TABLE products  
ADD COLUMN product_country VARCHAR(30) NOT NULL AFTER product_id,  
ADD COLUMN product_code VARCHAR(10) NOT NULL FIRST;
```

Данная последовательность команд добавит в таблицу 2 столбца, один из них будет следовать после столбца product\_id, а второй будет первым.

### Пример 2:

Для изменения спецификации уже определённого столбца можно использовать оператор MODIFY. Например, поменяем тип столбца sell\_amount.

```
ALTER TABLE sells  
MODIFY sell_amount INT4;
```

Если же необходимо также поменять и имя столбца, то следует использовать оператор CHANGE. Он позволяет полностью поменять спецификацию поля таблицы.

```
ALTER TABLE sells  
CHANGE sell_amount sell_money INT4;
```

Оператор ALTER также позволяет изменять и добавлять/удалять индексы, ограничения, ключи и многое другое.

Всё вышесказанное относится к изменению структуры таблицы. А что же о данных? Конечно, существует возможность редактировать и сами данные. Для этого существует оператор UPDATE. Оператор DELETE в свою очередь позволяет удалять кортежи.

### Пример 3:

Для начала работы с операторами UPDATE и DELETE отключим безопасный режим работы.

```
SET SQL_SAFE_UPDATES=0;
```

Пусть поступила задача повысить цены на пиво на 10 процентов. Обновим данные в таблице products. Ниже приведём код оператора UPDATE.

```
UPDATE products  
SET product_price = product_price*1.2  
WHERE product_type='Пиво';
```

Как видим, данные обновились. Обратите внимание на оператор условия WHERE. Он позволяет ограничить круг кортежей, к которым будет применён оператор UPDATE.

| product_id | product_name    | product_type | product_price |
|------------|-----------------|--------------|---------------|
| 1          | Крыніца. Моцнае | Пиво         | 1.2           |
| 2          | Аліварыя        | Пиво         | 1.8           |
| 3          | Сваяк           | Водка        | 8             |

| product_id | product_name    | product_type | product_price |
|------------|-----------------|--------------|---------------|
| 1          | Крыніца. Моцнае | Пиво         | 1.44          |
| 2          | Аліварыя        | Пиво         | 2.16          |
| 3          | Сваяк           | Водка        | 8             |

Условие может быть более жёстким. К примеру, повысить на 10 процентов цену самого дешёвого пива.

```
UPDATE products  
SET product_price = product_price*1.2  
WHERE product_type='Пиво'  
ORDER BY product_price  
LIMIT 1;
```

Тут следует обратить внимание на работу операторов ORDER BY и LIMIT. Они позволяют наложить дополнительное ограничение на число обновляемых записей.

#### Пример 4:

Для удаления записей таблицы использую оператор DELETE. Подобно оператору UPDATE он использует оператор условия WHERE. К примеру, удалим все записи типа «пиво» с ценой меньше 2 из таблицы products.

```
DELETE FROM products
WHERE product_type='Пиво' AND product_price<2;
```

*Нормализация* – процесс приведения БД к приемлемому виду, согласно потребностям организации. Как правило, нормализация позволяет избежать дублирования данных и снизить вероятность возникновения аномалий модификации данных.

Для удобства и унификации был разработан механизм *нормальных форм*.

1НФ (первая нормальная форма): в любой ячейке таблицы содержится атомарное значение, есть первичный ключ.

#### Пример 5:

Таблица departments содержит информацию о кафедрах.

| dep_id | dep_name | dep_phone            |
|--------|----------|----------------------|
| 1      | DU       | 209-73-16, 209-23-23 |
| 2      | WWW      | 210-32-03, 229-09-11 |

Как можем заметить, в столбце dep\_phone нарушается условие 1НФ. В столбце указано не атомарное значение – указаны два номера телефона. Решение – создать отдельную таблицу для номера телефона кафедры. Действуем с помощью команд SQL.

1) Создаём временный столбец

```
ALTER TABLE departments
ADD COLUMN phone1 VARCHAR(30);
```

2) Разделим значения в проблемном столбце и выделим их в отдельный (временный) столбец.

```
UPDATE departments
SET phone1 = substring_index(dep_phone, ',', 1)
WHERE dep_id>0;
```

3) Удаляем из первого столбца то, что добавили во второй. А затем удаляем пробел.

```
UPDATE departments
SET dep_phone = right(dep_phone, length(dep_phone)-length(phone1)-1)
WHERE dep_id>0;
```

|   | dep_id | dep_name | dep_phone | phone1    |
|---|--------|----------|-----------|-----------|
| ▶ | 1      | DUIA     | 209-11-22 | 209-73-15 |
|   | 2      | Web      | 209-00-17 | 209-12-35 |
| * | NULL   | NULL     | NULL      | NULL      |

```
UPDATE departments
SET dep_phone = trim(dep_phone)
WHERE dep_id>0;
```

|   | dep_id | dep_name | dep_phone | phone1    |
|---|--------|----------|-----------|-----------|
| ▶ | 1      | DUIA     | 209-11-22 | 209-73-15 |
|   | 2      | Web      | 209-00-17 | 209-12-35 |
| * | NULL   | NULL     | NULL      | NULL      |

- 4) Создаём отдельную таблицу для номеров телефонов

```
CREATE TABLE dep_phones
(
  dep_name VARCHAR(20),
  dep_phone VARCHAR(30)
)
```

- 5) Заполняем её с помощью вложенных запросов

```
INSERT INTO dep_phones
(dep_name, dep_phone)
SELECT dep_name, phone1
FROM departments;
```

```
INSERT INTO dep_phones
(dep_name, dep_phone)
SELECT dep_name, dep_phone
FROM departments;
```

| dep_name | dep_phone |
|----------|-----------|
| DU       | 209-73-16 |
| WWW      | 210-32-03 |
| DU       | 209-23-23 |
| WWW      | 229-09-11 |

- 6) Удаляем лишние столбцы в исходной таблице.

```
ALTER TABLE departments
DROP COLUMN dep_phone;

ALTER TABLE departments
DROP COLUMN phone1;
```

**2НФ (вторая нормальная форма):** 1НФ + отсутствуют частичные зависимости составного первичного ключа и не ключевых полей.

Проблема решается посредством введения искусственного одно столбцового ключа или вынесения данных в отдельную таблицу.

**3НФ (третья нормальная форма):** 2НФ + отсутствуют зависимости между не ключевыми полями.

Проблема решается посредством вынесения зависимых данных в отдельную таблицу.

Синтаксис вынесения в отдельную таблицу см. выше с пункта 4.

**НФБК (нормальная форма Бойса-Кодда):** Усиленный аналог 3НФ. Не допустимы любые функциональные зависимости, кроме как от потенциального ключа.

Проблема решается путём разделения зависимых атрибутов на 2 таблицы.

*4НФ (четвёртая нормальная форма):* НФБК + отсутствуют многозначные зависимости.

Проблема решается путём разделения не зависящих атрибутов на 2 таблицы.

Подробнее о НФБК и 4НФ см. лекционные материалы.

### ***Задания для самостоятельной работы***

1. Создайте таблицу в новой БД. Она должна отражать информацию о студентах вашей группы. В таблице должно быть не менее 10 полей максимально различных типов. *Для упрощения работы, разрешается создать одну такую таблицу на всю группу. При создании таблицы, сначала прочитайте все задания пункта 1.*
  - 1.1. Измените тип, название, порядок следования 3 –ёх произвольных столбцов.  
Обратите внимание на приведение типов в MySQL.
  - 1.2. Добавьте столбец со средним баллом студента и размером стипендии и размером оплаты за учёбу (если таковых не было). Пусть необходимо увеличить все стипендии на 10% и поднять плату за обучение на 15%. Реализуйте необходимый скрипт для обновления данных.
  - 1.3. Реализуйте скрипт для увеличения стипендии на 20% тем студентам, у которых в фамилии больше гласных букв, чем согласных.
    - 1.3.1. Увеличьте также стипендию на столько процентов каждому студенту, сколько дней осталось до конца текущего года.
  - 1.4. Реализуйте последовательность команд для разделения мальчиков и девочек на 2 разных таблицы.
  - 1.5. Используя ваш проект из ЛР1 продемонстрируйте свои умения во владении операторами обновления структуры таблиц и данных. Реализуйте не менее 5 различных скриптов.
    - 1.5.1. Реализуйте оператор обновления, который изменяет некоторую дату на ближайший четверг чётного числа.
    - 1.5.2. Продемонстрируйте работу с обновлением столбца типа SET.
    - 1.5.3. *Реализуйте оператор обновления, заменяющий строковые значения (например, имена) на их шифровку шифра Вижинера. Использовать хранимые функции и процедуры нельзя. Всё сделать в один запрос! \**
  - 1.6. Добавьте к имени студента приставку в зависимости от места рождения.  
Например, «Саша» -> «Саша.Бобруйск». Важно! Названия городов хранятся в другой таблице. Используйте оператор обновления с соединением таблиц.
2. На рисунке представлены таблицы некоторой ненормализованной БД. С помощью команд SQL приведите их (БД) к 3 нормальной форме. При проектировании новой БД учитывайте ограничения на первичные и внешние ключи.



| Author 1<br>(Автор 1) | Author 2<br>(Автор 2)          | Title<br>(Название) | ISBN       | Price<br>(Цена) | Cust. name<br>(Имя покупателя) | Cust. address<br>(Адрес покупателя)      | Purch. date<br>(Дата покупки) |
|-----------------------|--------------------------------|---------------------|------------|-----------------|--------------------------------|--|-------------------------------|
| David Sklar           | Adam Trachtenberg              | PHP Cookbook        | 0596101015 | 44,99           | Emma Brown                     | 1565 Rainbow Road, Los Angeles, CA 90014 | Mar 03 2009                   |
| Danny Goodman         |                                | Dynamic HTML        | 0596527403 | 59,99           | Darren Ryder                   | 4758 Emily Drive, Richmond, VA 23219     | Dec 19 2008                   |
| Hugh E. Williams      | David Lane                     | PHP and MySQL       | 0596005436 | 44,95           | Earl B. Thurston               | 862 Gregory Lane, Frankfort, KY 40601    | Jun 22 2009                   |
| David Sklar           | Adam Trachtenberg              | PHP Cookbook        | 0596101015 | 44,99           | Darren Ryder                   | 4758 Emily Drive, Richmond, VA 23219     | Dec 19 2008                   |
| Rasmus Lerdorf        | Kevin Tatroe & Peter MacIntyre | Programming PHP     | 0596006815 | 39,99           | David Miller                   | 3647 Cedar Lane, Waltham, MA 02154       | Jan 16 2009                   |

2.1.

| Emp. ID | First | Last  | Children's Names     | Children's Birthdates              |
|---------|-------|-------|----------------------|------------------------------------|
| 1001    | Jane  | Doe   | Mary, Sam            | 1/1/92, 5/15/94                    |
| 1002    | John  | Doe   | Mary, Sam            | 1/1/92, 5/15/94                    |
| 1003    | Jane  | Smith | John, Pat, Lee, Mary | 10/5/94, 10/12/90, 6/6/96, 8/21/94 |
| 1004    | John  | Smith | Michael              | 7/4/96                             |
| 1005    | Jane  | Jones | Edward, Martha       | 10/21/95, 10/15/89                 |

2.2.

3. Приведите указанные таблицы к указанным нормальным формам.

3.1. 1НФ

| Номер      | Автомобиль              | Год  | Стоимость | Характеристики               |
|------------|-------------------------|------|-----------|------------------------------|
| АФ 1233 ФА | Mersedes-Benz G-400     | 2002 | 28000     | Автомат, дизель, 4.0 л.      |
| FG 67 SPV  | Mersedes-Benz G-400 AMG | 2002 | 38500     | Типтроник, дизель, 4.0 л.    |
| АО 1234 ОА | Toyota Sequoia          | 2012 | 32500     | Автомат, бензин, 5.7 л.      |
| АО 4254 АО | Toyota Avalon           | 2015 | 21000     | Автомат, бензин, 3.5 л.      |
| ТТ 777 МН  | Subaru Forester         | 2016 | 18800     | Автомат, бензин, 2.5 л.      |
| SS 908 КЛВ | Suzuki SX4              | 2020 | 19000     | Механическая, бензин, 1.6 л. |

## 3.2. 4НФ.

| Title        | Star             | Producer           |
|--------------|------------------|--------------------|
| Great Film   | Lovely Lady      | Money Bags         |
| Great Film   | Handsome Man     | Money Bags         |
| Great Film   | Lovely Lady      | Helen Pursestrings |
| Great Film   | Handsome Man     | Helen Pursestrings |
| Boring Movie | Lovely Lady      | Helen Pursestrings |
| Boring Movie | Precocious Child | Helen Pursestrings |

## 3.3. 4НФ

| Фамилия | Курс             | Учебник           |
|---------|------------------|-------------------|
| А       | Информатика      | Информатика       |
| А       | Сети ЭВМ         | Информатика       |
| А       | Информатика      | Сети ЭВМ          |
| А       | Сети ЭВМ         | Сети ЭВМ          |
| В       | Программирование | Программирование  |
| В       | Программирование | Теория алгоритмов |

## 3.4. НФБК

| №<br>корта | Время<br>начала | Время<br>окончания | Тариф        |
|------------|-----------------|--------------------|--------------|
| 1          | 09:30           | 10:30              | «Бережливый» |
| 1          | 11:00           | 12:00              | «Бережливый» |
| 1          | 14:00           | 15:30              | «Стандарт»   |
| 2          | 10:00           | 11:30              | «Премиум-В»  |
| 2          | 11:30           | 13:30              | «Премиум-В»  |
| 2          | 15:00           | 16:30              | «Премиум-А»  |

## 3.5. 5НФ\*

| Фамилия | Группа | Мероприятие |
|---------|--------|-------------|
| Пронин  | "А"    | "Прыжок"    |
| Пронин  | "Т"    | "Прыжок"    |
| Исаев   | "Б"    | "Связь"     |
| Бонд    | "S"    | "Контакт"   |