

## Лабораторная работа №2. SQL-запросы на выборку. Нормализация БД.

ППСУБД  
2курс. 5 группа.  
Кушнеров А.В. 2017-2018 г.

### *Цель работы:*

Изучение SQL-запросов на выборку, изменение и удаление данных. Получение навыков самостоятельного создания элементарных запросов, запросов с подзапросами, запросов с соединением. Получение базовых сведений о нормализации и нормальных формах БД.

### Минимальные теоретические сведения

### *Запросы на выборку*

Все примеры, представленные ниже, иллюстрируют работу с БД “world” предустановленную с MySQL.

Команда SELECT – позволяет получить некоторую выборку данных из таблиц.

Общий вид:

```
SELECT [DISTINCT | ALL] { * | [columnExpression [AS newName]] [, ...] }  
FROM TableName [alias] [, ...]  
[WHERE condition]  
[GROUP BY columnList] [HAVING condition]  
[ORDER BY columnList]
```

Пример 1:

Вывести все города Аргентины с населением более 1000000 человек.

```
SELECT * FROM city  
WHERE Population>1000000 AND CountryCode = 'ARG'.
```

Пример 2:

Вывести страны, в которых среднее население в городах превышает 500000.

```
SELECT AVG(population), CountryCode  
FROM city  
GROUP BY CountryCode  
HAVING AVG(population)>500000
```

Оператор INNER JOIN позволяет соединить две таблицы по какому-либо условию. Очень полезная вещь при соединении таблиц по первичному и внешнему ключам.

Пример 3:

Получить список стран и их официальных языков.

```
SELECT c.name, cl.language  
FROM country as c  
INNER JOIN  
countrylanguage as cl
```

```
ON c.Code=cl.CountryCode
WHERE cl.IsOfficial ='T'
```

Пример 4:

Вывести среднее население в городах на каждом континенте.

```
SELECT c.Continent, avg(city.Population)
FROM country as c
INNER JOIN
city
ON c.Code=city.CountryCode
GROUP BY c.Continent
```

Пример 5:

Вывести список всех городов Европы и их население

1) *Через вложенный запрос*

```
SELECT distinct name, population
FROM city
WHERE CountryCode IN
(
SELECT Code FROM country
WHERE Continent ='Europe'
)
```

2) *Используя INNER JOIN*

```
SELECT city.name, city.population
FROM city
INNER JOIN
country AS c
ON city.CountryCode=c.Code
WHERE c.Continent='Europe'
```

## *Процесс нормализации*

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

Пример: Таблица departments

dep_id	dep_name	dep_phone
1	DU	205-73,205-74
2	WWW	201-35,206-37

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.

```
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	DU	205-74	205-73
2	WWW	206-37	201-35

- 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	205-74
WWW	206-37
DU	205-73
WWW	201-35

- 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 одностолбцового 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.

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

1. Изучите тестовую БД world, которая устанавливается вместе с MySQL.  
Также её можно скачать по ссылке и импортировать на ваш локальный сервер.  
<http://downloads.mysql.com/docs/world.sql.zip>  
Используйте команды SHOW TABLES и DESC.
2. Выполните с помощью запросов языка SQL следующие задания.  
(Каждое задание выполняется в один запрос)
  1. Выведите список всех стран и столиц.
  2. Выведите список всех стран с населением более 10 млн. человек.
  3. Выведите список всех стран Европы, ВВП которых превышает 3000.
  4. Сколько стран мира используют русский язык?
  5. Найдите среднее население в городах каждой страны.
  6. Найдите количество городов на каждом континенте.
  7. Найдите страны, в которых население столицы превышает среднее население по городам страны. (используйте вложенный запрос)
  8. Сколько стран представлены в таблице, только своими столицами?
  9. Найдите все страны, в которых используют 1 язык.
  10. Найдите страны по площади превышающие Республику Беларусь.
  11. Найдите страны, названия которых начинаются с буквы «А».
  12. Найдите страны, в которых код страны и название страны начинаются и заканчиваются одинаково.
  13. Какое количество населения используют русский язык в Республике Беларусь?
  14. Выведите список всех стран, которые получили независимость в 20-ом веке.
  15. Найдите страны с максимальным ВВП на каждом континенте.
3. Самостоятельно попрактикуйте различные запросы к БД world. Будьте готовы сконструировать любой запрос по требованию преподавателя во время сдачи ЛР.
4. На рисунке представлена таблица некоторой ненормализованной БД.  
С помощью команд SQL приведите её (БД) к 3 нормальной форме. При проектировании новой БД учитывайте ограничения на первичные и внешние ключи.

Author 1 (Автор 1)	Author 2 (Автор 2)	Title (Название)	ISBN	Price (Цена)	Cust. name (Имя покупателя)	Cust. address (Адрес покупателя)	Purch. date (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	<b>Darren Ryder</b>	<b>4758 Emily Drive, Richmond, VA 23219</b>	<b>Dec 19 2008</b>
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	<b>Darren Ryder</b>	<b>4758 Emily Drive, Richmond, VA 23219</b>	<b>Dec 19 2008</b>
Rasmus Lerdorf	Kevin Tatroe & Peter MacIntyre	Programming PHP	0596006815	39,99	David Miller	3647 Cedar Lane, Waltham, MA 02154	Jan 16 2009