

Работа с данными - часть №1

СУБД, SQL, mysql



LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

Цель занятия

1. Рассмотреть общие сведения о базах данных
2. Рассмотреть основы языка SQL
3. Рассмотреть интерфейс PHP mysqli для взаимодействия с СУБД MySQL



СУБД?

СУБД - это совокупность программных средств, обеспечивающих управление созданием и использованием баз данных.

Основные функции СУБД:

- | управление данными во внешней памяти (на дисках);
- | управление данными в оперативной памяти с использованием дискового кэша;
- | журнализация изменений;
- | резервное копирование и восстановление базы данных после сбоев;
- | поддержка языков БД (язык определения данных, язык манипулирования данными).



Структура БД

Проектирование БД - создание эффективной структуры данных, обеспечивающее хранение требуемой информации

Хорошая структура

Максимально упрощает взаимодействие с БД

Гарантирует непротиворечивость данных

Выжимает максимум производительности из системы

Позволяет быстро добавлять новые сущности к БД



Принцип организации данных

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц

Таблицы состоят из столбцов и строк

Каждый элемент таблицы – один элемент данных

Все ячейки в столбце таблицы имеют одинаковый тип

Каждый столбец имеет уникальное имя

Одинаковые строки в таблице отсутствуют

Порядок следования строк и столбцов может быть произвольным

Запросы к базе данных возвращают результат в виде таблиц, которые тоже могут выступать как объект запросов



Нормализация БД

Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. Конечной целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.



Первая нормальная форма

Первая нормальная форма (1 NF): говорят, что таблица находится в первой нормальной форме, если она описывает одну сущность и не содержит повторяющихся атрибутов.

Код предмета	Предмет	Учитель	Код ученика	Фамилия ученика	Имя ученика
A01	Проектирование	Линд	S01	Роовяли	Марек
			S02	Петерсон	Мария
			S03	Мартсон	Элла
A02	Построение	Сепп	S02	Петерсон	Мария
			S03	Мартсон	Элла



Первая нормальная форма

Первая нормальная форма (1 NF): говорят, что таблица находится в первой нормальной форме, если она описывает одну сущность и не содержит повторяющихся атрибутов.

Занятия	
Предмет	Учитель
Проектирование	Линд
Построение	Сепп

Зарегистрированные учащиеся			
Код предмета	Код ученика	Фамилия ученика	Имя ученика
A01	S01	Роовяли	Марек
A01	S02	Петерсон	Мария
A01	S03	Мартсон	Элла
A02	S02	Петерсон	Мария
A02	S03	Мартсон	Элла



Вторая нормальная форма

Вторая нормальная форма (2 NF): таблица находится во второй нормальной форме, если она в первой нормальной форме и все атрибуты зависят только от первичного ключа.

Зарегистрированные учащиеся	
Код предмета	Код ученика
A01	S01
A01	S02
A01	S03
A02	S02
A02	S03

Ученики		
Код ученика	Фамилия ученика	Имя ученика
S01	Роовяли	Марек
S02	Петерсон	Мария
S03	Мартсон	Элла



Третья нормальная форма

Третья нормальная форма (3 NF): таблица находится в третьей нормальной форме, если она во второй нормальной форме и все атрибуты, не связанные с первичным ключом, зависят друг от друга.

Ученики					
Код	Фамилия	Имя	Дата рождения	Возраст	Последнее обновление
S01	Роовяли	Марек	11.09.1955	50	12.11.2001
S02	Петерсон	Мария	01.02.1990	14	15.12.2004
S03	Мартсон	Элла	09.06.1978	26	26.01.2005



Третья нормальная форма

Третья нормальная форма (3 NF): таблица находится в третьей нормальной форме, если она во второй нормальной форме и все атрибуты, не связанные с первичным ключом, зависят друг от друга.

Ученики					
Код	Фамилия	Имя	Дата рождения	Возраст	Последнее обновление
S01	Роовяли	Марек	11.09.1955	50	12.11.2001
S02	Петерсон	Мария	01.02.1990	14	15.12.2004
S03	Мартсон	Элла	09.06.1978	26	26.01.2005



Третья нормальная форма

Третья нормальная форма (3 NF): таблица находится в третьей нормальной форме, если она во второй нормальной форме и все атрибуты, не связанные с первичным ключом, зависят друг от друга.

Занятия				
Код предмета	Предмет	Учитель	Сидячие места в помещении	Свободные места
A01	Проектирование	Линд	5	2
A02	Построение	Сепп	6	4



Третья нормальная форма

Третья нормальная форма (3 NF): таблица находится в третьей нормальной форме, если она во второй нормальной форме и все атрибуты, не связанные с первичным ключом, зависят друг от друга.

Занятия				
Код предмета	Предмет	Учитель	Сидячие места в помещении	Свободные места
A01	Проектирование	Линд	5	2
A02	Построение	Сепп	6	4



SQL

SQL (structured query language — «язык структурированных запросов») - формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД)



SELECT - выборка данных

- | `SELECT * FROM students;`
- | `SELECT name, lastname, birthday FROM students;`
- | `SELECT name, lastname FROM student WHERE birthday > '12-06-2001';`
- | `SELECT id, title FROM courses WHERE start_data = '03-08-2015' AND teacher_id = 2;`

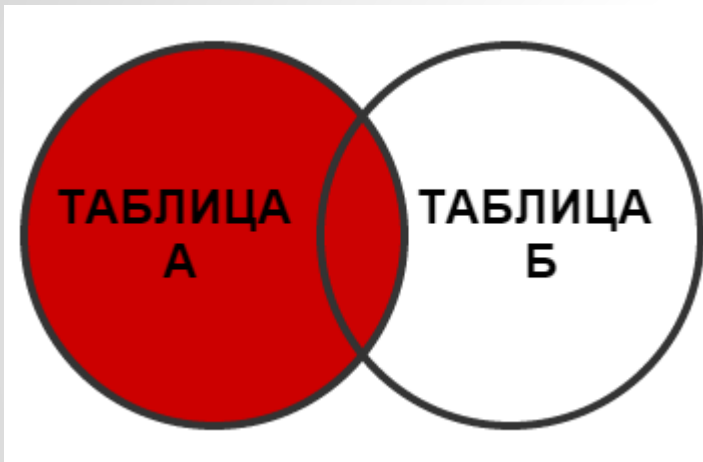


SELECT - выборка данных

- | `SELECT * FROM students;`
- | `SELECT name, lastname, birthday FROM students;`
- | `SELECT name, lastname FROM student WHERE birthday > '12-06-2001';`
- | `SELECT id, title FROM courses WHERE start_data = '03-08-2015' AND teacher_id = 2;`



Объединение таблиц



SELECT

course.id, course.title, teacher.id,
teacher.name, teacher.lastname,
teacher.level

FROM course

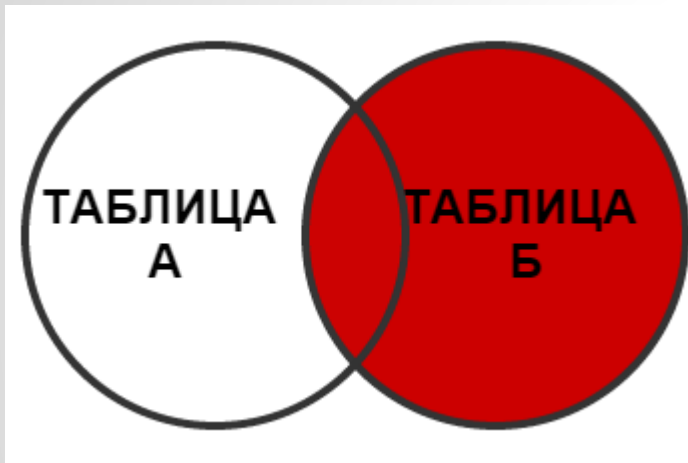
LEFT JOIN teacher **ON** course.id =
teacher.id



LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ



Объединение таблиц



SELECT

course.id, course.title, teacher.id,
teacher.name, teacher.lastname,
teacher.level

FROM course

RIGHT JOIN teacher **ON** course.id =
teacher.id



Вставка данных

- | **INSERT INTO** teachers (name, lastname, birthday, level) **VALUES** ('Павел', 'Сердюков', '21-02-1978', 'доцент')
- | **INSERT INTO** teachers **VALUES** (NULL, 'Павел', 'Сердюков', '21-02-1978', 'доцент')
- | **INSERT INTO** course **VALUES** (NULL, 'PHP'), (NULL, 'CSS'), (NULL, 'HTML'), (NULL, 'Javascript')



Обновление данных

```
I UPDATE students SET name = 'Антон',  
                        lastname = 'Голомазов',...  
WHERE id = 1
```



Удаление данных

| **DELETE FROM** students **WHERE** id = 2



Работа с MySQL с помощью PHP

Работа с БД состоит из следующих этапов:

- | Подключение к серверу БД
- | Выбор БД
- | Подготовка запроса
- | Выполнение запроса
- | Обработка результатов
- | Отключение от сервера БД



Интерфейс mysqli - подключение и выбор СУБД

```
$connection = @new mysqli('localhost', 'root', '', 'shop', 3306);
```

```
if(mysqli_connect_errno()){  
    die(mysqli_connect_error());  
}
```

```
$connection->query('SET NAMES "UTF-8"');
```



Интерфейс mysqli - выполнение запроса

```
$sql = "SELECT * FROM catalogs";  
$result = $connection->query($sql);
```



Интерфейс mysqli - получение данных

// ПОЛУЧЕНИЕ ВСЕХ ЗАПИСЕЙ

```
$records = $result->fetch_all(MYSQL_ASSOC); // ассоциативный массив
```

```
$records = $result->fetch_all(MYSQLI_NUM); // индексный массив
```

// ПОЛУЧЕНИЕ 1 ЗАПИСИ

```
$record = $result->fetch_assoc();
```



Интерфейс mysqli - подготовленные запросы

```
$sql = "SELECT name FROM catalogs WHERE id =  
?";
```

```
$stmt = $connection->prepare($sql);
```

```
$id = 1;
```

```
// указываем параметр запроса
```

```
$stmt->bind_param('i', $id);
```

```
$stmt->execute();
```

```
// привязка переменных к подготовленному запросу  
для размещения результата
```

```
$stmt->bind_result($name);
```

```
while($stmt->fetch()){ echo $name.'<br />';}
```

i	соответствующая переменная имеет тип integer
d	соответствующая переменная имеет тип double
s	соответствующая переменная имеет тип string
b	соответствующая переменная является большим двоичным объектом (blob) и будет пересылаться пакетами



Интерфейс mysqli - ПОДГОТОВЛЕННЫЕ ЗАПРОСЫ

```
$sql = "UPDATE catalogs SET name = ? WHERE id = ?";
```

```
if ($stmt = $connection->prepare($sql)) {
```

```
..
```

```
    $stmt->bind_param('si', $name, $id);
```

```
    $stmt->execute(); // выполняем запрос
```

```
    printf("Число строк: %d.\n", $stmt->affected_rows);
```

```
    $stmt->close();
```

```
}
```

```
$connection->close();
```

i	соответствующая переменная имеет тип integer
d	соответствующая переменная имеет тип double
s	соответствующая переменная имеет тип string
b	соответствующая переменная является большим двоичным объектом (blob) и будет пересылаться пакетами



Интерфейс mysqli - подготовленные запросы

```
$sql = "DELETE FROM catalogs WHERE id = ?";
```

```
if($stmt = $connection->prepare($sql)){
```

```
    $id = 1;
```

```
    $stmt->bind_param('s', $id);
```

```
    $stmt->execute();
```

```
    echo "Было удалено ".$stmt->affected_rows;
```

```
    $stmt->close();
```

```
}
```

```
$connection->close();
```

i	соответствующая переменная имеет тип integer
d	соответствующая переменная имеет тип double
s	соответствующая переменная имеет тип string
b	соответствующая переменная является большим двоичным объектом (blob) и будет пересылаться пакетами



Демонстрация



LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

Что нужно сделать после вебинара?

1. Пересмотреть запись вебинара
2. Прочитать методичку
3. Выполнить домашнее задание
4. Продолжить чтение книг из списка [“рекомендованной литературы”](#)
5. Задавать свои вопросы в общем чате
6. Ожидать следующий вебинар

