

Управление данными

Лекция 9

# Объединение запросов и визуализация данных

Свечников С.В.

# Содержание занятия

01. Объединение запросов

02. РНР

# 01. Объединение запросов

# UNION

Объединить несколько запросов в один можно с помощью предложения **UNION**. Оно представляет выходные данные двух и более запросов в виде одного набора строк и столбцов.

## Пример

```
SELECT Iname AS 'ФИО' FROM person
```

```
UNION
```

```
SELECT lector FROM course;
```

# UNION-совместимость

Объединить несколько запросов с помощью UNION можно в том случае, если их выходные столбцы **UNION-совместимы**.

Требования UNION-совместимости:

1. Количество выбираемых столбцов должно быть одинаковым во всех запросах.
2. Типы данных в выбираемых столбцах должны быть сравнимыми.

## Примеры

```
SELECT lname, fname FROM person  
UNION  
SELECT lector FROM course;
```

Не выполнится

```
SELECT concat(lname, ' ', fname)  
AS 'ФИО' FROM person  
UNION  
SELECT lector FROM course;
```

Выполнится

# UNION и дублирование

Оператор UNION автоматически исключает повторяющиеся строки из выходных данных.

## Пример

```
SELECT lector FROM course;
```

10 записей

```
SELECT title FROM course;
```

10 записей

```
SELECT lector FROM course  
UNION  
SELECT title FROM course;
```

$10 + 10 \neq 20$

# Включение дубликатов

Отменить автоматическое удаление дубликатов строк можно, указав предложение **UNION ALL** вместо UNION.

## Пример

```
SELECT lector FROM course  
UNION ALL  
SELECT title FROM course;
```

20 записей

# UNION и выражения

Вместе с UNION можно использовать константы и выражения, но они должны удовлетворять UNION-совместимости.

## Пример

```
SELECT 'Максимальный возраст ', MAX(year), 'лет' FROM person  
UNION  
SELECT 'Минимальный возраст ', MIN(year), 'лет' FROM person;
```

# UNION и сортировка

Для размещения выходных данных в определенном порядке можно использовать предложение ORDER BY.

Поскольку столбцы в UNION не имеют имен, на них нужно ссылаться по номеру.

## Пример

```
SELECT concat(lname, ' ', fname)
AS 'ФИО' FROM person
UNION
SELECT lector FROM course
ORDER BY 1;
```

# INTERSECT

Оператор **INTERSECT** находит пересечение строк, выданных несколькими запросами. Другими словами, выбираются только те строки, которые присутствуют в результатах каждого запроса.

## Пример

```
SELECT pid FROM person
```

```
INTERSECT
```

```
SELECT gid FROM группа;
```

В MySQL INTERSECT не поддерживается.

# EXCEPT

**EXCEPT** принимает в качестве аргументов два запроса A и B и выводит только те строки из запроса A, которые отсутствуют в результатах запросов B. Кроме того, он исключает строки, выданные запросом B и отсутствующие в запросе A.

## Пример

```
SELECT pid FROM person
```

**EXCEPT**

```
SELECT gid FROM группа;
```

В MySQL EXCEPT не поддерживается.

# 02. PHP

# Обеспечение

Для использования языка PHP нам понадобится следующее ПО:

- веб-сервер Apache – должен быть запущен
- сам язык PHP
- СУБД MySQL – должен быть запущен

Кроме того, потребуется знание основ HTML, CSS

Для базы данных нужно создать в папке www директорию, в которой будут храниться файлы php.

В браузере обращаться к файлам будем следующим образом:

**<http://localhost/person/stud.php>**

person – название папки в директории www

stud.php – название файла с выполняемым кодом

# Соединение с БД

```
<?php
```

```
    // данные доступа к базе данных
```

```
    $db_host="localhost"; // имя сервера, обычно localhost
```

```
    $db_user="root"; // имя пользователя БД
```

```
    $db_password="pass"; // пароль БД
```

```
    $db_name = "stud"; // имя БД
```

```
    // соединение с БД
```

```
    mysql_connect($db_host, $db_user, $db_password) or die (mysql_error());
```

```
    mysql_select_db($db_name) or die (mysql_error());
```

```
    mysql_query("SET NAMES 'utf8'");
```

```
    // Выполняем запрос
```

```
        $res_all = mysql_query('select * from person');
```

```
        $res_count = mysql_query('select count(*) from person');
```

```
?>
```

# Результаты запроса

**mysql\_result** - возвращает данные результата запроса

Возвращает содержимое одного поля из набора результата MySQL.

<p>Всего студентов:

<strong><? echo mysql\_result(\$res\_count,0) ?></strong>

</p>

# Массив

**mysql\_fetch\_array** - обрабатывает ряд результата запроса, возвращая ассоциативный массив, численный массив или оба.

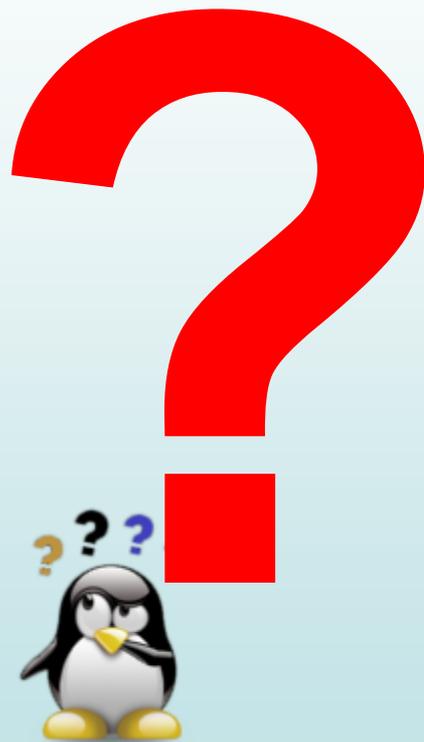
```
<?      // выводим записи в цикле
$sql_data = mysql_fetch_array($res_all);
        echo '<tr><td>'.$sql_data['pid'].'</td><td>'.$sql_data['lname'].'</td></tr>';
while ($sql_data = mysql_fetch_array($res_all)) {
        echo '<tr><td>'.$sql_data['pid'].'</td><td>'.$sql_data['lname'].'</td></tr>';
    }
?>
```

# Для курсового проекта

В курсовом проекте необходимо:

1. Создать несколько страниц, которые будут выводить информацию из таблиц БД в соответствии с запросами.
2. Создать не менее 10 запросов, которые будут выводить информацию из таблиц БД и использоваться в интерфейсе.
3. Создать на страницах меню, которое будет помогать ориентироваться в информации, представленной в БД.
4. Можно использовать формы html для задания ограничения выборки данных, фильтрации данных, поиска подстрок и т.п.
5. Можно визуально красиво представить выводимые данные с помощью таблиц стилей CSS.
6. Можно реализовать работу с данными не только для выборки, но и для добавления, редактирования и удаления данных из таблиц БД.
7. ...

# Вопросы



# Домашнее задание



- Изучить главу 13 книги М.Грабера Введение в SQL
- Почитать информацию о языке PHP, например, тут [php.ru](http://php.ru)
- Подготовиться к контрольной работе по лекции
- Выполнять главы 3 и 4 курсового проекта

# Контроль



Для выполнения контрольного теста используем ссылку:

[app.startexam.com/Center/Web/student](https://app.startexam.com/Center/Web/student)

Короткая ссылка:

[clck.ru/2Undm](https://clck.ru/2Undm)

Пароль на тест:

**4752**