



**WEB** разработка

**CRUD**

**заявки за напреднали**

# Съдържание

- CRUD – **C**reate **R**ead **U**ppdate **D**eleate
- **JOINS**
- DB queries – **Aggregate** Functions

CRUD

# CRUD

**C**reate – създава запис в БД

**R**ead – прочита/вади запис от БД

**U**ppdate - променя запис в БД

**D**elete – изтрива запис в БД

**C reate**

**R**

**U**

**D**

# Create

Чрез форма получаваме данни от потребителя и правим запис в БД – с **INSERT** добавяме получената информация в БД.

**!** При въвеждане на запис само в едно от много полета на таблицата за другите трябва да има стойност по подразбиране! /например date\_deleted/  
В противен случай няма да направите запис в БД.

**!** Когато записвате стрингове в БД – променливата, в която е записана стойността им трябва да бъдат в кавички – например **'\$city\_name'**!

```
$insert_query = "INSERT INTO cities (city_name)  
VALUES ('$city_name')";
```

C

R ead

U

D

# Read

Избираме записи от БД и ги отпечатваме в брауъра –  
Използваме **SELECT** query

Селектираме само тези записи, които отговарят на условието  
**date\_deleted IS NULL**

! Добавяме бутон/линк

Промяна / Edit

Изтриване / Delete

С тях ще реализираме

**U**ppdate и **D**elete

```
$read_query = "SELECT * FROM cities  
WHERE date_deleted IS NULL";
```

```
//update or to delete  
echo ' ' . '<a href="update.php?id=' . $row['id_city'] . '>Edit</a>';  
echo ' ' . '<a href="delete.php?id=' . $row['id_city'] . '>Delete</a>';  
echo '</li>';
```



**C**

**R**

**U pdate**

**D**

# U pdate

Избираме запис от БД и го променяме – с **UPDATE** query

Избираме го с id – в случая - **`$row['id_city']`**

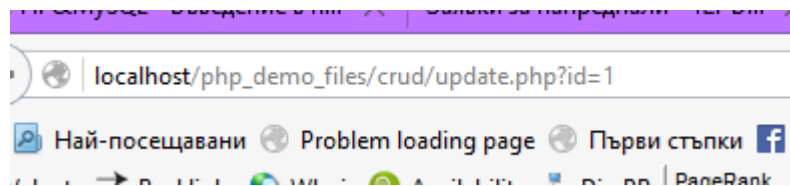
```
'<a href="update.php?id=' . $row['id_city'] . '">Edit</a>'
```

добавяме параметър за GET заявка към линка на Edit бутона

**`?id=$row['id_city']`**

така всеки бутон-линк

ще ‘знае’ кой запис да промени



Мога да взема id на записа за промяна от **`$_GET['id']`**

# Update - 2

Промяната извършваме като извикваме със **SELECT** данните във форма за промяна /форма с попълнени данни, които идват от БД./

Формата е идентична с тази в **create.php**

```
$q = "SELECT * FROM cities WHERE id_city = $id_city";
```

...

```
echo "<input type='hidden' name = 'id_city' value="."$row['id_city'].>";
```

```
echo "<input type='text' name='city_name' value="."$row['city_name'].>";
```

...

!Проверяваме в браузъра формата и данните – дали са отпечатани коректно name и value на всяко поле!

# Update - 3

Следва **UPDATE** query на избрания запис

```
$update_query = "UPDATE cities  
                SET city_name ='$city_name'  
                WHERE id_city=$id_city";
```

C

R

U

D elete

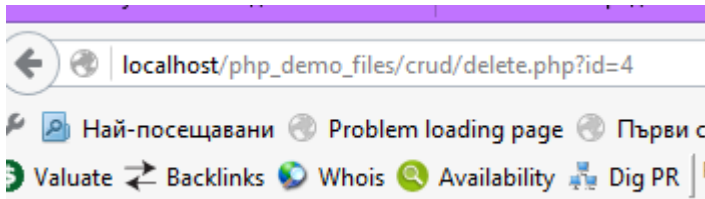
# Delete

Променяме **date\_deleted** от NULL на **текущата дата**.

Това ще скрие записите – “ще ги изтрие”, тъй като в Read – условието ни е да се селектират записите, за които **date\_deleted IS NULL**

Благодарение на **?id='.\$row['id\_city']** в

```
echo '<a href="delete.php?id='.$row['id_city'].'">Delete</a>';
```



БД ‘разбира’ на кой ред да промени **date\_deleted** /аналогично с U pdate/

**JOINS**

# JOINS

Предназначение –

свързваме таблици в БД чрез **FK** и **PK** /Internal relation/

Така имаме информация от няколко таблици и можем да изпълним **CRUD** с тази информация



# JOINS - 2

## **INNER JOIN** or

**JOIN:** Returns all rows when there is at least one match in BOTH tables

**LEFT JOIN:** Return all rows from the left table, and the matched rows from the right table

**RIGHT JOIN:** Return all rows from the right table, and the matched rows from the left table

**FULL JOIN:** Return all rows when there is a match in ONE of the tables

**JOIN / INNER JOIN**

# JOIN / INNER JOIN

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name=table2.column_name;
```

**Или**

```
SELECT column_name(s)
FROM table1
JOIN table2
ON table1.column_name=table2.column_name;
```

# JOIN / INNER JOIN - 2

Като резултат виждаме само хотелите, за които има пълна информация от двете таблици

Hotel 1	vratsa	description 1	100
Hotel 2	Burgas	description 2	450

За да виждаме и другите хотели използваме ...

**LEFT JOIN**

# LEFT JOIN

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name=table2.column_name;
```

**ИЛИ**

```
SELECT column_name(s)
FROM table1
LEFT OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

# LEFT JOIN - 2

Резултатът – виждаме и тези,  
хотели, за които няма  
информация за градове.

Hotel 1	vratsa	description 1	100	<a href="#">Edit</a>	<a href="#">Delete</a>
Hotel 2	Burgas	description 2	450	<a href="#">Edit</a>	<a href="#">Delete</a>
hotel 3		description 3	2500	<a href="#">Edit</a>	<a href="#">Delete</a>
hotel 4		description 4	150	<a href="#">Edit</a>	<a href="#">Delete</a>

*Използваме вид JOIN според контекста на проекта ни.*

Допълнителни ресурси JOINS - [http://www.w3schools.com/sql/sql\\_join.asp](http://www.w3schools.com/sql/sql_join.asp)

# Aggregate Functions



# Aggregate functions

Агрегиращите SQL функции връщат един резултат изчислен от много записи

AVG() - средна ст-ст

COUNT() - брой записи

MAX() - макс. ст-ст

MIN() - мин. ст-ст

SUM() - сума от записи

Често се ползва и GROUP BY за групиране на записи по определена колона

# Agregate functions - 1

```
SELECT AVG(column_name)
```

```
FROM table_name /GROUP BY column_name/
```

```
SELECT ProductName, Price FROM Products  
WHERE Price > (SELECT AVG(Price) FROM Products);
```

+ Options

**AVG(`rooms\_quantity`)**

477.8000

+ Options

**COUNT(`hotel\_name`)**

10

```
SELECT COUNT(column_name)
```

```
FROM table_name;
```

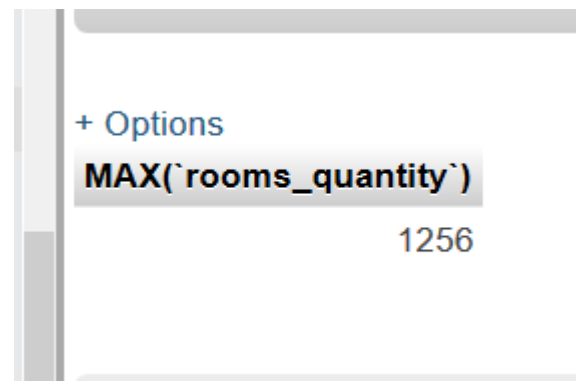
# Aggregate functions - 2

```
SELECT COUNT(*) FROM table_name;
```

**результат** - връща броя на записите в съответната таблица

```
SELECT MAX(column_name) FROM table_name;
```

**результат** – максималната стойност от избраната колона



# Aggregate functions - 2

```
SELECT MIN(column_name) FROM table_name;
```

**резултат – най-малката стойност от избраната колона**

```
SELECT SUM(column_name) FROM table_name;
```

**резултат – връща сумата от записите в съответната колона**

подходящо е да се комбинира с GROUP BY по стойност в друга колона /  
например по id на град, за да сумираме броя на стаите в хотелите в определен град/

**Допълнителни ресурси - [http://www.w3schools.com/sql/sql\\_functions.asp](http://www.w3schools.com/sql/sql_functions.asp)**