

Basic CRUD in MySQL Server

Create, Retrieve, Update, Delete
using SQL queries



SoftUni Team
Technical Trainers
Software University
<http://softuni.bg>



Java DB

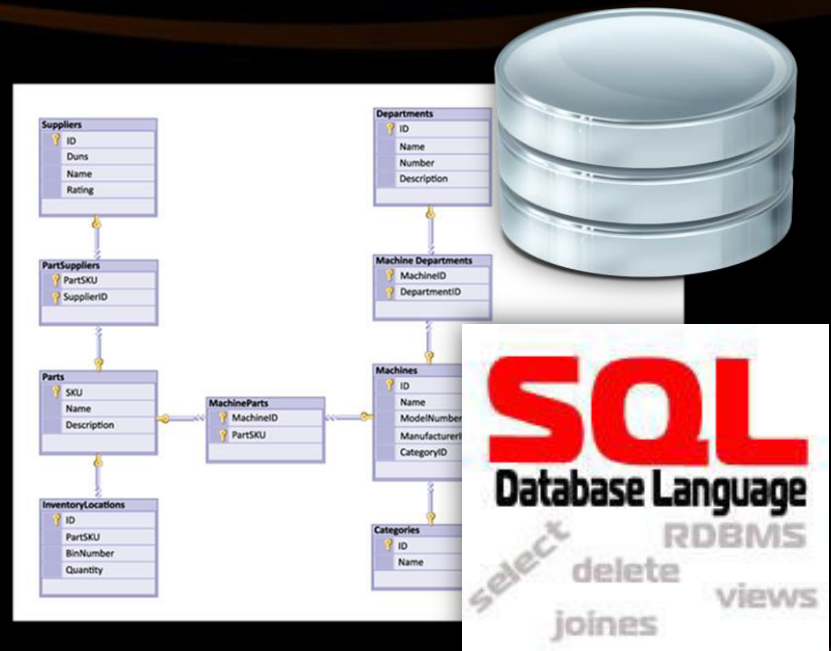


Table of Contents



Query Basics

SQL Introduction



Retrieving Data

Using SQL SELECT



Writing Data in Tables

Using SQL INSERT



Modifying Existing Records

Using SQL UPDATE and DELETE

Have a Question?

sli.do

#JavaDB



Query Basics

SQL Introduction

SQL Queries – Few Examples

- Select first, last name and job title about employees:

```
SELECT first_name, last_name, job_title FROM employees;
```

- Select projects which start on 01-06-2003:

```
SELECT * FROM projects WHERE start_date='2003-06-01';
```

- Inserting data into table:

```
INSERT INTO projects(name, start_date)  
VALUES('Introduction to SQL Course', '2006-01-01');
```

SQL Queries – Few Examples

- Update end date of specific projects:

```
UPDATE projects
  SET end_date = '2006-08-31'
 WHERE start_date = '2006-01-01';
```

- Delete specific projects:

```
DELETE FROM projects
  WHERE start_date = '2006-01-01';
```



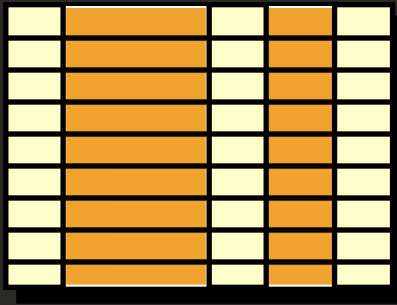
Retrieving Data

Using SQL SELECT

Capabilities of SQL SELECT

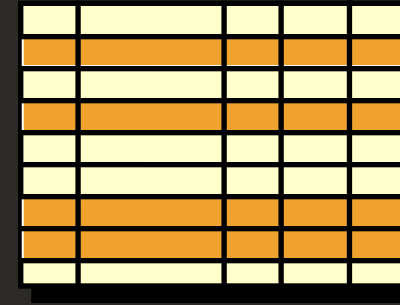
Projection

Take a subset of the columns



Selection

Take a subset of the rows



Join

Combine tables by
some column

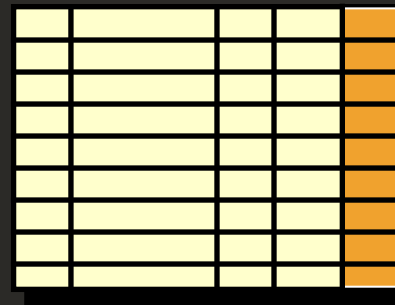


Table 1

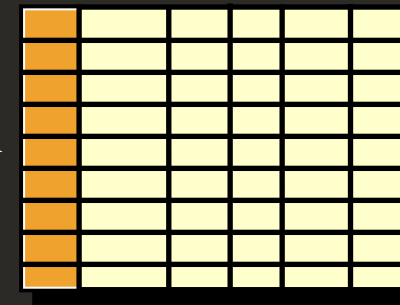


Table 2

SELECT – Examples

- Selecting all columns from the "departments" table

```
SELECT * FROM departments;
```

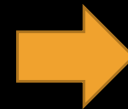
List of columns
(* for all)

department_id	name	manager_id
1	Engineering	12
2	Tool design	4
3	Sales	273
...

Table name

- Selecting specific columns

```
SELECT department_id, name  
FROM departments
```



department_id	name
1	Engineering
2	Tool design
3	Sales
...	...

Column Aliases

- Aliases rename a table or a column heading

```
SELECT employee_id AS id, first_name, last_name  
FROM employees;
```

id	first_name	last_name
1	Guy	Gilbert
2	Kevin	Brown
...

Display name

- You can shorten fields or clarify abbreviations

```
SELECT c.duration,  
       c.acg AS 'Access Control Gateway'  
FROM calls AS c;
```

Concatenation

- You can concatenate column names or strings using the **concat()** function
 - String literals are enclosed in [''](single quotes)
 - Table and column names containing special symbols use [`](backtick)

```
SELECT concat(`first_name`, ' ', `last_name`) AS 'full_name',  
       `job_title` as 'Job Title',  
       `id` AS 'No.'  
FROM `employees`;
```

Problem: Employee Summary

- Find information about all employees, listing their:
 - Full Name
 - Job title
 - Salary
- Use **concatenation** to display first and last names as **one field**
- Note: Query **Hospital** database

Employee Summary - Solution

Concatenation

```
SELECT concat(`first_name`, ' ', `last_name`) AS  
       'full_name',  
       `job_title` as 'job_title',  
       `salary` AS `salary`  
FROM `employees` WHERE salary >= 1000;
```

Column alias

Filtering the Selected Rows

- Use **DISTINCT** to eliminate duplicate results

```
SELECT DISTINCT `department_id`  
FROM `employees`;
```

- You can filter rows by specific conditions using the **WHERE** clause

```
SELECT `last_name`, `department_id`  
FROM `employees`  
WHERE `department_id` = 1;
```

- Other **logical operators** can be used for greater control

```
SELECT `last_name`, `salary`  
FROM `employees`  
WHERE `salary` <= 20000;
```

Other Comparison Conditions

- Conditions can be combined using **NOT**, **OR**, **AND** and brackets

```
SELECT `last_name` FROM `employees`  
WHERE NOT (`manager_id` = 3 OR `manager_id` = 4);
```

- Using **BETWEEN** operator to specify a range:

```
SELECT `last_name`, `salary` FROM `employees`  
WHERE `salary` BETWEEN 20000 AND 22000;
```

- Using **IN / NOT IN** to specify a set of values:

```
SELECT `first_name`, `last_name`, `manager_id`  
FROM `employees`  
WHERE `manager_id` IN (109, 3, 16);
```

Comparing with NULL

- **NULL** is a special value that means missing value
 - Not the same as 0 or a blank space
- Checking for **NULL** values

```
SELECT `last_name`, `manager_id`  
FROM `employees`  
WHERE `manager_id` = NULL;
```

This is always false!

```
SELECT `last_name`, `manager_id`  
FROM `employees`  
WHERE `manager_id` IS NULL;
```

```
SELECT `last_name`, `manager_id`  
FROM `employees`  
WHERE `manager_id` IS NOT NULL;
```


Sorting with ORDER BY

- Sort rows with the **ORDER BY** clause
 - ASC**: ascending order, default
 - DESC**: descending order

```
SELECT `last_name`, `hire_date`  
FROM `employees`  
ORDER BY `hire_date`;
```

```
SELECT `last_name`, `hire_date`  
FROM `employees`  
ORDER BY `hire_date` DESC;
```

ASC is the default
sorting order

LastName	HireDate
	1998-07-31
	1999-02-26
Tamburello	1999-12-12
...	...

LastName	HireDate
Valdez	2005-07-01
Tsoflias	2005-07-01
Abbas	2005-04-15
...	...

Views

- Views are **virtual tables** made from others tables, views or joins between them
- Usage:
 - To simplify writing complex queries
 - To limit access to data for certain users

Views (2)

Table 1		
Column 1	Column 2	Column 3

Table 2		
Column 1	Column 2	Column 3



v_table1_table2		
Column 1	Column 2	Column 3

Views - Example

- Get employee names and salaries, by department

```
CREATE VIEW `v_hr_result_set` AS  
SELECT  
    CONCAT(`first_name`, ' ', `last_name`) AS 'Full Name', `salary`  
FROM `employees` ORDER BY `department_id`;
```

```
SELECT * FROM `v_hr_result_set`;
```


Problem: Top Paid Employee

- Create a **view** that selects all information about the **top paid employee**
 - Name the view **v_top_paid_employee**

```
SELECT * FROM `v_top_paid_employee`;
```



id	first_name	last_name	job_title	department_id	salary
8	Pedro	Petrov	Medical Director	3	2,100

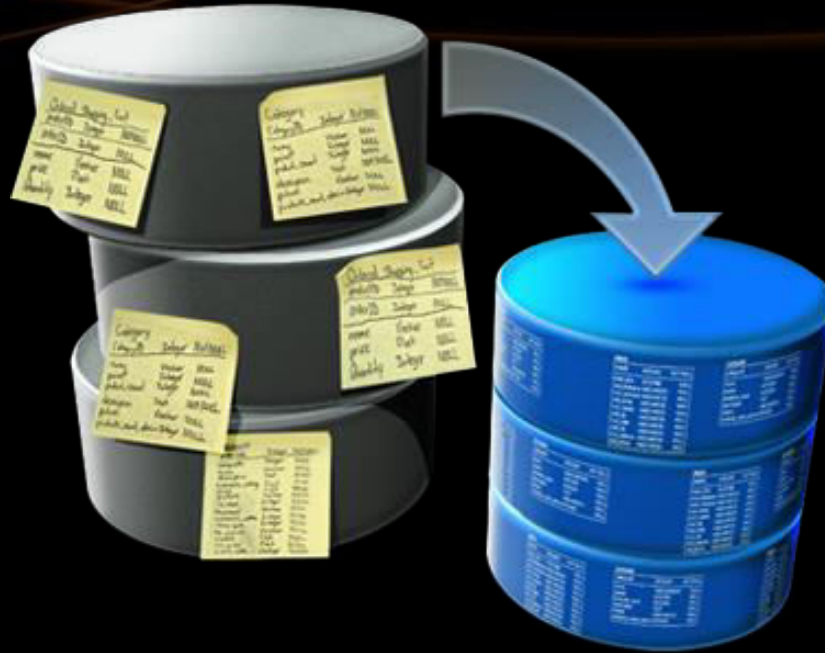
- Note: Query **Geography** database

Solution: Top Paid Employee

```
CREATE VIEW `v_top_paid_employee`  
AS  
  SELECT * FROM `employees`  
  ORDER BY `salary` DESC LIMIT 1;
```

Sorting column

Greatest value first



Writing Data in Tables

Using SQL INSERT

Inserting Data

- The SQL **INSERT** command

```
INSERT INTO `towns` VALUES (33, 'Paris');
```

Values for
all columns

```
INSERT INTO projects(`name`, `start_date`)  
VALUES ('Reflective Jacket', NOW())
```

Specify
columns

- Bulk data can be recorded in a single query, separated by comma

```
INSERT INTO `employees_projects`  
VALUES (229, 1),  
      (229, 2),  
      (229, 3), ...
```


Inserting Data (2)

- You can use existing records to create a **new table**

```
CREATE TABLE `customer_contacts`  
AS SELECT `customer_id`, `first_name`, `email`, `phone`  
FROM `customers`;
```

New table name

Existing source

- Or into an existing table

```
INSERT INTO projects(name, start_date)  
SELECT CONCAT(name, ' ', ' Restructuring'), NOW()  
FROM departments;
```

List of columns



Modifying Existing Records

Using SQL UPDATE and DELETE

Deleting Data

- Deleting specific rows from a table

Condition

```
DELETE FROM `employees`  
WHERE `employee_id` = 1;
```

- Note: Don't forget the **WHERE** clause!
- Delete all rows from a table (**TRUNCATE** works faster than **DELETE**)

```
TRUNCATE TABLE users;
```

Updating Data

- The SQL **UPDATE** command

```
UPDATE `employees`  
  SET `last_name` = 'Brown'  
WHERE `employee_id` = 1;
```

New values

```
UPDATE `employees`  
  SET `salary` = `salary` * 1.10,  
      `job_title` = CONCAT('Senior', ' ', `job_title`)  
WHERE `department_id` = 3;
```

- Note: Don't forget the **WHERE** clause!

Summary

- We can easily manipulate our database with SQL queries

```
SELECT *  
FROM `projects`  
WHERE `start_date` = '2006-01-01';
```

- Queries provide a flexible and powerful method to manipulate records



Database Basics MySQL – Basic CRUD



Questions?



License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



**Software
University**

