Telerik Academy

```
SELECT e.EmployeeID, e.LastName, e.DepartmentID,
       d.DepartmentID, d.Name AS DepartmentName
FROM Employees e
  INNER JOIN Departments d
    ON e.DepartmentID = d.DepartmentID
WHERE d.Name = 'Sales'
```

Structured Query Language (SQL)

SQL Intro (MS SQL Server and MySQL)

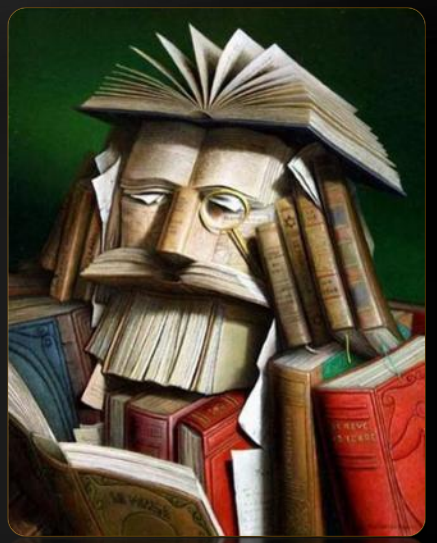
Databases

Telerik Software Academy

http://academy.telerik.com

Telerik Academy

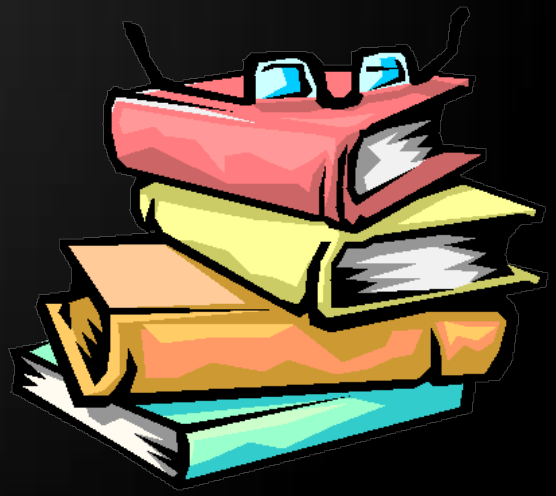
Table of Contents



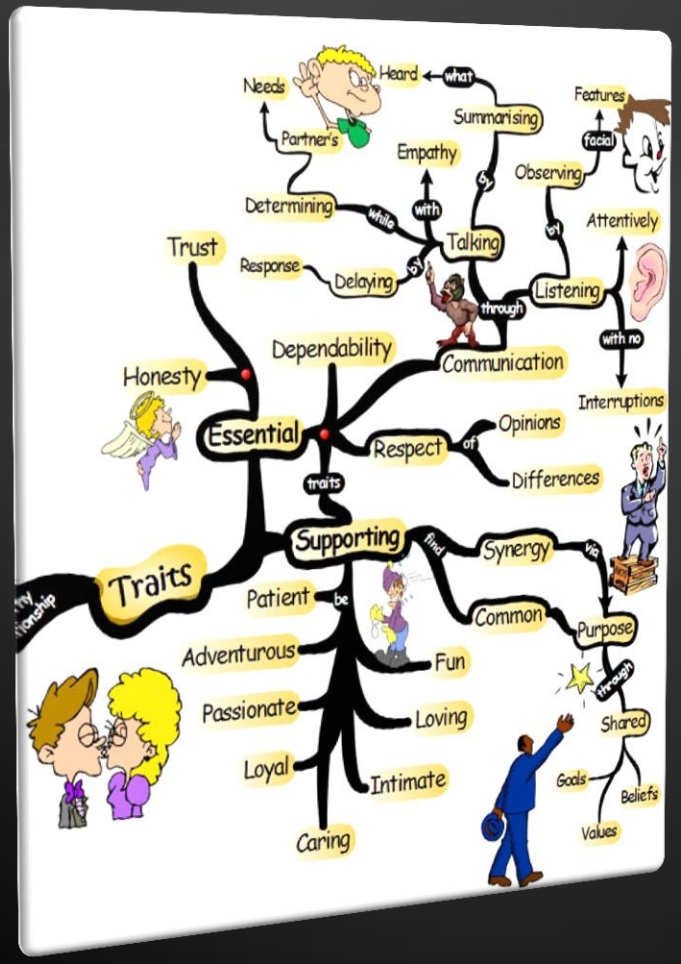
Telerik Academy

Table of Contents (2)

4. Selecting Data From Multiple Tables
   - Natural Joins
   - Join with `USING` Clause
   - Inner Joins with `ON` Clause
   - Left, Right and Full Outer Joins
   - Cross Joins
5. Inserting Data
6. Updating Data
7. Deleting Data



Telerik Academy



Relational Databases and SQL

The SQL Execution Model

SQL

Telerik Academy

Relational Databases and SQL

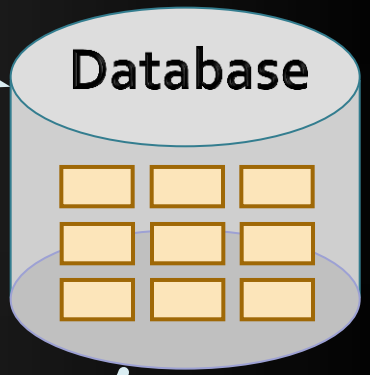
- A relational database can be accessed and modified by executing SQL statements
  - SQL allows
    - Defining / modifying the database schema
    - Searching / modifying table data
  - A set of SQL commands are available for extracting subset of the table data
  - Most SQL commands return a single value or record set

Telerik Academy

Communicating with the DB

```
SELECT Name
FROM Departments
```

SQL statement is sent to the DB server



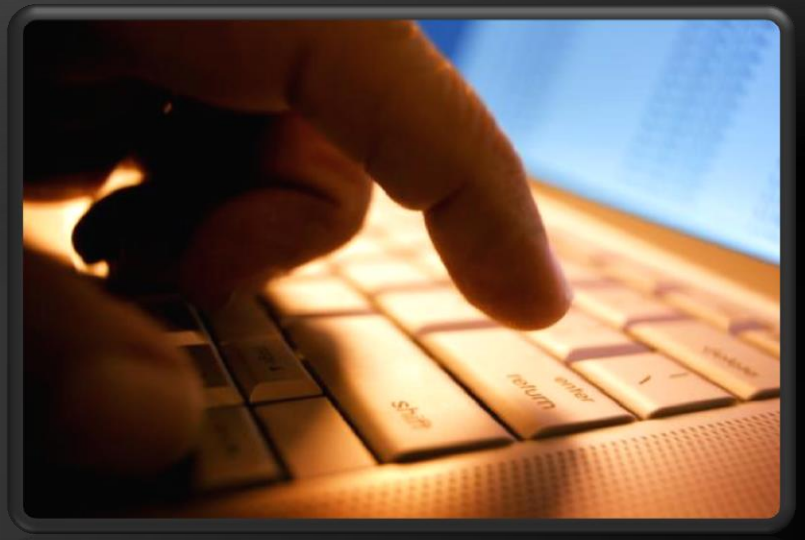
Name
Engineering
Sales
Marketing
…

The result is returned (usually as a record set)

SQL Execution

- SQL commands are executed through a database connection
  - DB connection is a channel between the client and the SQL server
  - DB connections take resources and should be closed when no longer used
  - Multiple clients can be connected to the SQL server at the same time
  - SQL commands can be executed in parallel
    - Transactions and isolation deal with concurrency

Telerik Academy



SQL and T-SQL

Introduction

What is SQL?

- Structured Query Language (SQL)
  - Declarative language for query and manipulation of relational data
  - en.wikipedia.org/wiki/SQL
- SQL consists of:
  - Data Manipulation Language (DML)
    - `SELECT`, `INSERT`, `UPDATE`, `DELETE`
  - Data Definition Language (DDL)
    - `CREATE`, `DROP`, `ALTER`
    - `GRANT`, `REVOKE`

SQL – Few Examples

```
SELECT FirstName, LastName, JobTitle FROM Employees
```

```
SELECT * FROM Projects WHERE StartDate = '1/1/2006'
```

```
INSERT INTO Projects(Name, StartDate)
VALUES('Introduction to SQL Course', '1/1/2006')
```

```
UPDATE Projects
SET EndDate = '8/31/2006'
WHERE StartDate = '1/1/2006'
```

```
DELETE FROM Projects
WHERE StartDate = '1/1/2006'
```

What is T-SQL?

- T-SQL (Transact SQL) is an extension to the standard SQL language
  - T-SQL is the standard language used in MS SQL Server
  - Supports `if` statements, loops, exceptions
    - Constructions used in the high-level procedural programming languages
  - T-SQL is used for writing stored procedures, functions, triggers, etc.

T-SQL – Example

```
CREATE PROCEDURE EmpDump AS
  DECLARE @EmpId INT, @EmpFName NVARCHAR(100),
    @EmpLName NVARCHAR(100)
  DECLARE emps CURSOR FOR
    SELECT EmployeeID, FirstName, LastName FROM Employees
  OPEN emps
  FETCH NEXT FROM emps INTO @EmpId, @EmpFName, @EmpLName
  WHILE (@@FETCH_STATUS = 0) BEGIN
    PRINT CAST(@EmpId AS VARCHAR(10)) + ' '
      + @EmpFName + ' ' + @EmpLName
    FETCH NEXT FROM emps INTO @EmpId, @EmpFName,
@EmpLName
  END
  CLOSE emps
  DEALLOCATE emps
GO
```

Telerik Academy

SQL Language

Introducing SELECT Statement

Telerik Academy

Capabilities of SQL SELECT

Projection

Take some of the columns

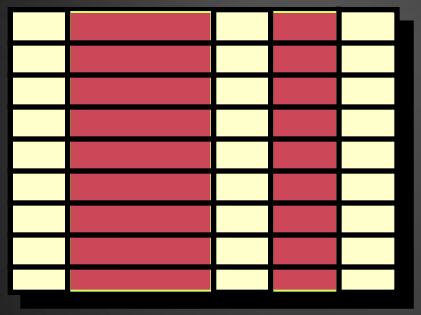


Table 1

Selection

Take some of the rows

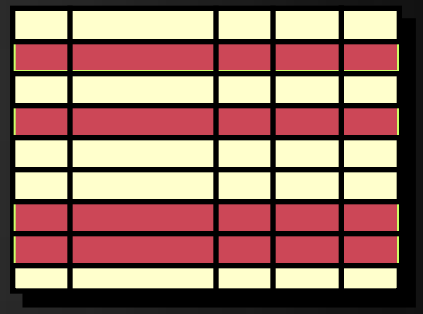


Table 1

Join

Combine tables by some column

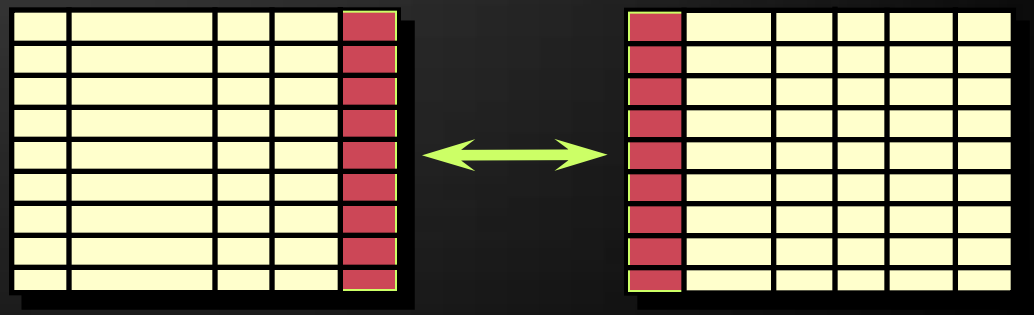
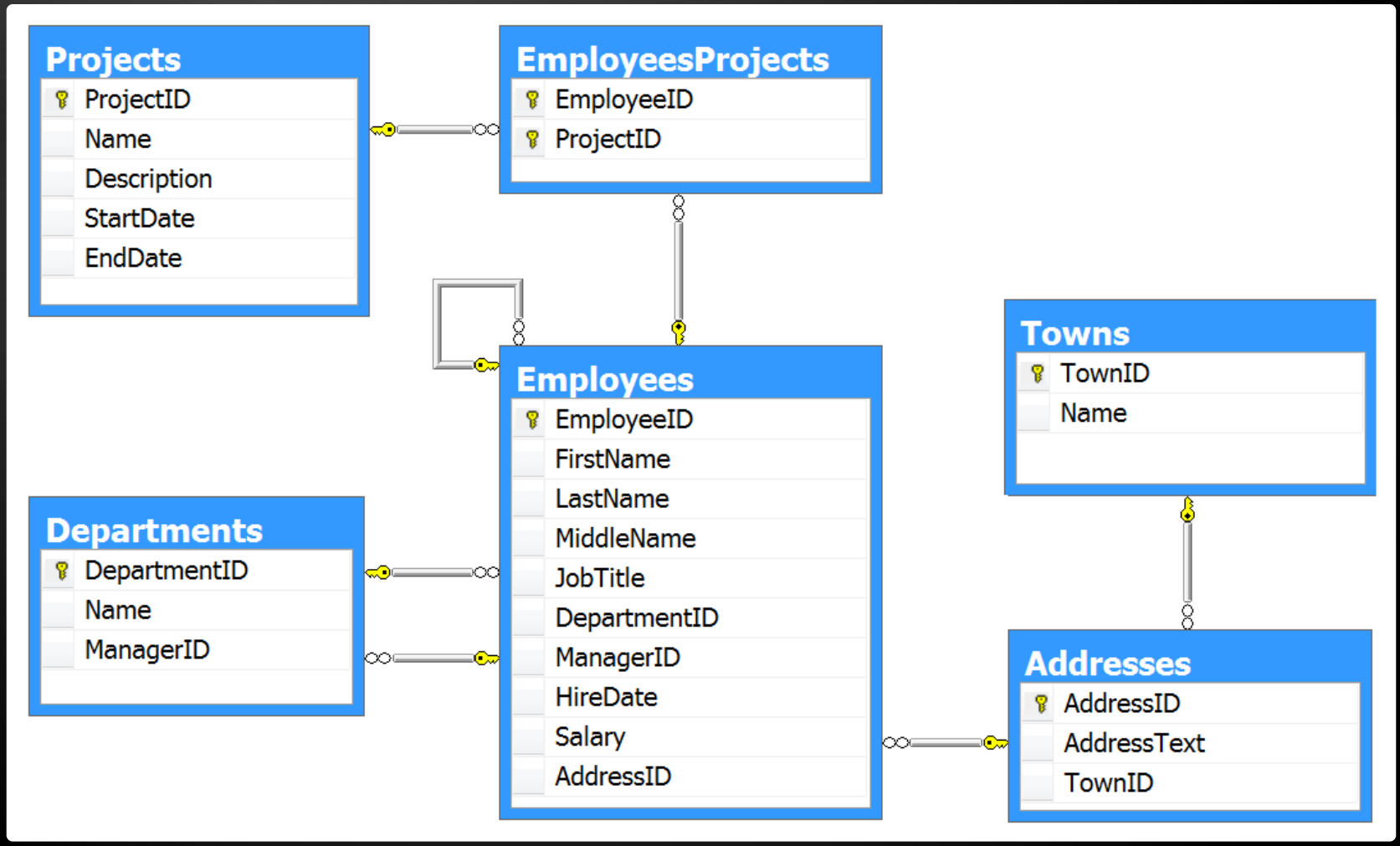


Table 1

Table 2

Telerik Academy

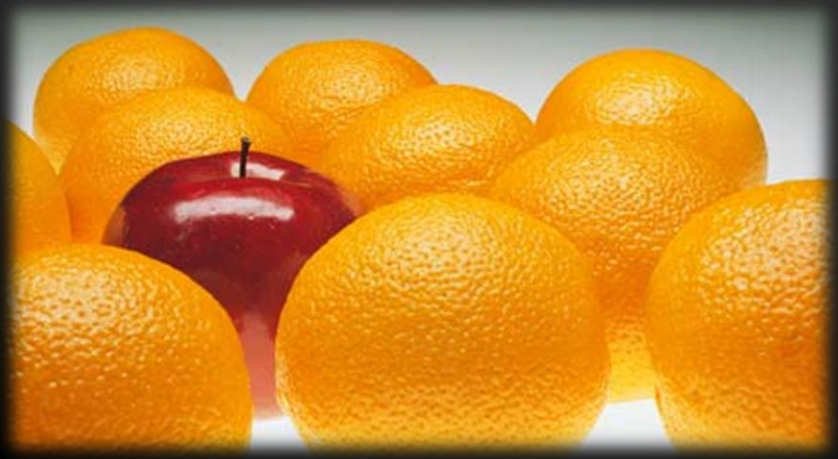
The Telerik Academy Database Schema in SQL Server



Basic SELECT Statement

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM   table
```

- SELECT identifies what columns
- FROM identifies which table



SELECT Example

- Selecting all columns from departments

```
SELECT * FROM Departments
```

DepartmentID	Name	ManagerID
1	Engineering	12
2	Tool design	4
3	Sales	273
…	…	…

- Selecting specific columns

```
SELECT
   DepartmentID,
   Name
FROM Departments
```

DepartmentID	Name
1	Engineering
2	Tool design
3	Sales

Arithmetic Operations

- Arithmetic operators are available:
  - +, -, *, /
- Examples:

```
SELECT (2 + 3) * 4   --> returns 20
```

```
SELECT LastName, Salary, Salary + 300
FROM Employees
```

LastName	Salary	(No column name)
Gilbert	12500,00	12800,00
Brown	13500,00	13800,00
Tamburello	43300,00	43600,00

The NULL Value

- A NULL is a value that is unavailable, unassigned, unknown, or inapplicable
  - Not the same as zero or a blank space
- Arithmetic expressions containing a NULL value are evaluated to NULL

```
SELECT LastName, ManagerID FROM Employees
```

LastName	ManagerID
Sánchez	NULL
Duffy	300
Wang	1

NULL is displayed as empty space or as NULL

Column Aliases

- Aliases rename a column heading
- Useful with calculations
- Immediately follows the column name
  - There is an optional AS keyword
- Double quotation marks if contains spaces

```
SELECT FirstName, LastName, Salary,
Salary*0.2 AS Bonus FROM Employees
```

FirstName	LastName	Salary	Bonus
Guy	Gilbert	12500,00	2500.00000
Kevin	Brown	13500,00	2700.00000

Concatenation Operator

- Concatenates columns or character strings to other columns
- Is represented by plus sign "+"
- Creates a resultant column that is a character expression

```
SELECT FirstName + ' ' + LastName AS [Full Name],
EmployeeID as [No.] FROM Employees
```

Full Name	No.
Guy Gilbert	1
Kevin Brown	2
Roberto Tamburello	3

Literal Character Strings

- A literal is a character, a number, or a date included in the SELECT list
- Date and character literal values must be enclosed within single quotation marks
- Each character string is output once for each row returned

```
SELECT FirstName + '''s last name is ' +
LastName AS [Our Employees] FROM Employees
```

Our Employees
Guy's last name is Gilbert
Kevin's last name is Brown
Roberto's last name is Tamburello

Telerik Academy

Removing Duplicate Rows

- The default display of queries is all rows, including duplicate rows

```
SELECT DepartmentID
FROM Employees
```

DepartmentID
7
7
2
…

- Eliminate duplicate rows by using the DISTINCT keyword in the SELECT clause

```
SELECT
  DISTINCT DepartmentID
FROM Employees
```

DepartmentID
7
2
…

Telerik Academy

Set Operations: UNION, INTERSECT and EXCEPT

- UNION combines the results from several SELECT statements
  - The columns count and types should match

```
SELECT FirstName AS Name
FROM Employees
UNION
SELECT LastName AS Name
FROM Employees
```

Name
A. Scott
Abbas
Abercrombie
…

- INTERSECT / EXCEPT perform logical intersection / difference between given two sets of records

Telerik Academy

Limiting the Rows Selected

- Restrict the rows returned by using the `WHERE` clause:

```
SELECT LastName,
DepartmentID FROM
Employees WHERE
DepartmentID = 1
```

LastName	DepartmentID
Tamburello	1
Erickson	1
Goldberg	1
…	…

- More examples:

```
SELECT FirstName, LastName, DepartmentID FROM
Employees WHERE LastName = 'Sullivan'
```

```
SELECT LastName, Salary FROM Employees
WHERE Salary <= 20000
```

Telerik Academy

Other Comparison Conditions

- Using BETWEEN operator to specify a range:

```
SELECT LastName, Salary FROM Employees
WHERE Salary BETWEEN 20000 AND 22000
```

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

```
SELECT FirstName, LastName, ManagerID FROM
Employees WHERE ManagerID IN (109, 3, 16)
```

- Using LIKE operator to specify a pattern:

```
SELECT FirstName FROM Employees
WHERE FirstName LIKE 'S%'
```

  - % means 0 or more chars; _ means one char

Comparing with NULL

- Checking for NULL value:

```
SELECT LastName, ManagerId FROM Employees
WHERE ManagerId IS NULL
```

```
SELECT LastName, ManagerId FROM Employees
WHERE ManagerId IS NOT NULL
```

- Attention: COLUMN=NULL is always false!

```
SELECT LAST_NAME, MANAGER_ID FROM EMPLOYEES
WHERE MANAGER_ID = NULL
```

This is always false!

```
SELECT LAST_NAME, MANAGER_ID FROM EMPLOYEES
WHERE NULL = NULL
```

This is always false!

Telerik Academy

Logical Operators and Brackets

- Using NOT, OR and AND operators and brackets:

```
SELECT FirstName, LastName FROM Employees
WHERE Salary >= 20000 AND LastName LIKE 'C%'
```

```
SELECT LastName FROM Employees
WHERE ManagerID IS NOT NULL OR LastName LIKE '%so_'
```

```
SELECT LastName FROM Employees
WHERE NOT (ManagerID = 3 OR ManagerID = 4)
```

```
SELECT FirstName, LastName FROM Employees
WHERE
    (ManagerID = 3 OR ManagerID = 4) AND
    (Salary >= 20000 OR ManagerID IS NULL)
```

Sorting with ORDER BY

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

```
SELECT LastName, HireDate
FROM Employees ORDER BY
HireDate
```

LastName	HireDate
Gilbert	1998-07-31
Brown	1999-02-26
Tamburello	1999-12-12

```
SELECT LastName, HireDate
FROM Employees ORDER BY
HireDate DESC
```

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

Telerik Academy

SQL Language

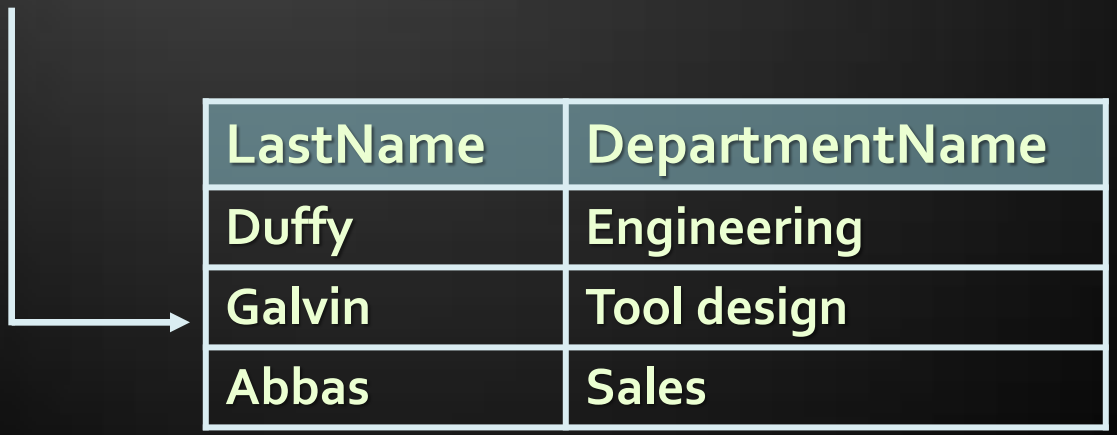
Joins: Selecting Data From Multiple Tables

Data from Multiple Tables

- Sometimes you need data from more than one table:

LastName	DepartmentID
Duffy	1
Abbas	3
Galvin	2

DepartmentID	Name
1	Engineering
2	Tool design
3	Sales



LastName	DepartmentName
Duffy	Engineering
Galvin	Tool design
Abbas	Sales

Cartesian Product

- This will produce Cartesian product:

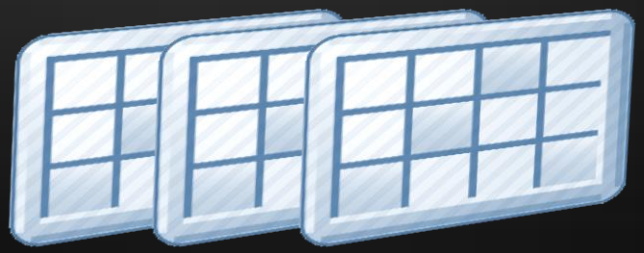
```
SELECT LastName, Name AS DepartmentName
FROM Employees, Departments
```

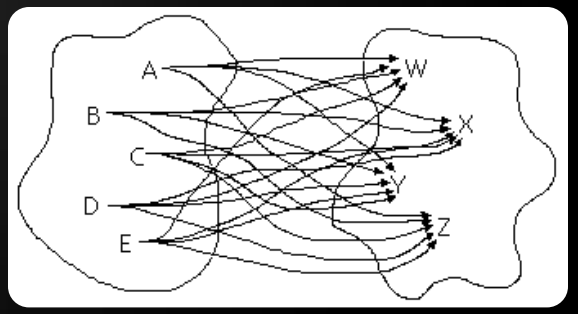
- The result:

LastName	DepartmentName
Duffy	Document Control
Wang	Document Control
Sullivan	Document Control
Duffy	Engineering
Wang	Engineering
..	..

Cartesian Product (2)

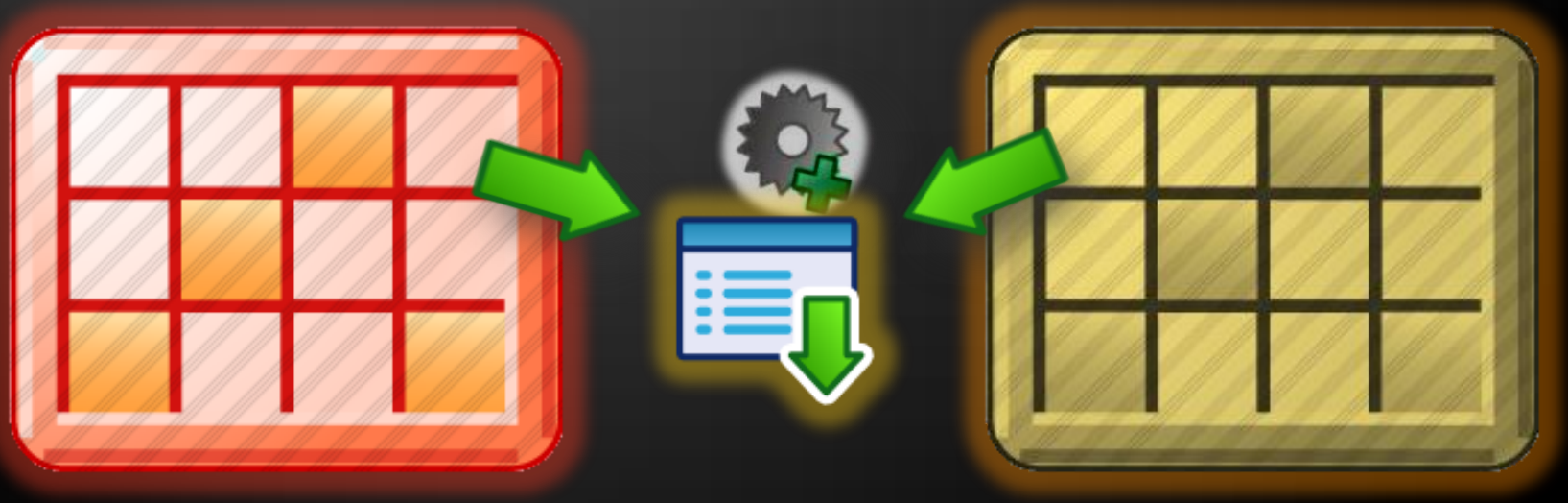
- A Cartesian product is formed when:
  - A join condition is omitted
  - A join condition is invalid
  - All rows in the first table are joined to all rows in the second table
- To avoid a Cartesian product, always include a valid join condition





Types of Joins

- Inner joins
- Left, right and full outer joins
- Cross joins



Inner Join with ON Clause

- To specify arbitrary conditions or specify columns to join, the ON clause is used
  - Such JOIN is called also INNER JOIN

```
SELECT e.EmployeeID, e.LastName, e.DepartmentID,
       d.DepartmentID, d.Name AS DepartmentName
FROM Employees e
  INNER JOIN Departments d
    ON e.DepartmentID = d.DepartmentID
```

EmployeeID	LastName	DepartmentID	DepartmentID	DepartmentName
1	Gilbert	7	7	Production
2	Brown	4	4	Marketing
3	Tamburello	1	1	Engineering

Equijoins

- Inner joins with join conditions pushed down to the WHERE clause

```
SELECT e.EmployeeID, e.LastName, e.DepartmentID,
       d.DepartmentID, d.Name AS DepartmentName
FROM Employees e, Departments d
WHERE e.DepartmentID = d.DepartmentID
```

EmployeeID	LastName	Depart-mentID	Depart-mentID	Department-Name
1	Gilbert	7	7	Production
2	Brown	4	4	Marketing
3	Tamburello	1	1	Engineering

INNER vs. OUTER Joins

- Inner join
  - A join of two tables returning only rows matching the join condition
- Left (or right) outer join
  - Returns the results of the inner join as well as unmatched rows from the left (or right) table
- Full outer join
  - Returns the results of an inner join as well as the results of a left and right join

INNER JOIN

```
SELECT e.LastName EmpLastName,
       m.EmployeeID MgrID, m.LastName MgrLastName
FROM Employees e INNER JOIN Employees m
  ON e.ManagerID = m.EmployeeID
```

EmpLastName	MgrID	MgrLastName
Erickson	3	Tamburello
Goldberg	3	Tamburello
Duffy	109	Sánchez
Johnson	185	Hill
Higa	185	Hill
Ford	185	Hill
Maxwell	21	Krebs
…	…	…

LEFT OUTER JOIN

```
SELECT e.LastName EmpLastName,
       m.EmployeeID MgrID, m.LastName MgrLastName
FROM Employees e LEFT OUTER JOIN Employees m
  ON e.ManagerID = m.EmployeeID
```

EmpLastName	MgrID	MgrLastName
Sánchez	NULL	NULL
Benshoof	6	Bradley
Miller	14	Maxwell
Okelberry	16	Brown
Hill	25	Mu
Frum	184	Richins
Culbertson	30	Barreto de Mattos
…	…	…

RIGHT OUTER JOIN

```
SELECT e.LastName EmpLastName,
       m.EmployeeID MgrID, m.LastName MgrLastName
FROM Employees e RIGHT OUTER JOIN Employees m
  ON e.ManagerID = m.EmployeeID
```

EmpLastName	MgrID	MgrLastName
Lertpiriyasuwat	38	Liu
NULL	39	Hines
NULL	40	McKay
Berglund	41	Wu
Koenigsbauer	123	Hay
NULL	124	Zabokritski
NULL	125	Decker
…	…	…

FULL OUTER JOIN

```
SELECT e.LastName EmpLastName,
       m.EmployeeID MgrID, m.LastName MgrLastName
FROM employee e FULL OUTER JOIN employee m
  ON e.ManagerID = m.EmployeeID
```

EmpLastName	MgrID	MgrLastName
Sanchez	NULL	NULL
…	…	…
Cracium	3	Tamburello
Gilbert	16	Brown
…	…	…
NULL	17	Hartwig
NULL	1	Gilbert
…	…	…

Three-Way Joins

- A three-way join is a join of three tables

```
SELECT e.FirstName, e.LastName,
  t.Name as Towns, a.AddressText
FROM Employees e
  JOIN Addresses a
    ON e.AddressID = a.AddressID
  JOIN Towns t
    ON a.TownID = t.TownID
```

FirstName	LastName	Towns	AddressText
Guy	Gilbert	Monroe	7726 Driftwood Drive
Kevin	Brown	Everett	2294 West 39th St.
Roberto	Tamburello	Redmond	8000 Crane Court
…	…	…	…

Self-Join

- Self-join means to join a table to itself
  - Always used with table aliases

```
SELECT e.FirstName + ' ' + e.LastName +
  ' is managed by ' + m.LastName as Message
FROM Employees e JOIN Employees m
ON (e.ManagerId = m.EmployeeId)
```

Message
Ovidiu Cracium is managed by Tamburello
Michael Sullivan is managed by Tamburello
Sharon Salavaria is managed by Tamburello
Dylan Miller is managed by Tamburello
…

Cross Join

- The CROSS JOIN clause produces the cross-product of two tables
  - Same as a Cartesian product
  - Not often used

```
SELECT LastName [Last Name], Name [Dept Name]
FROM Employees CROSS JOIN Departments
```

Last Name	Dept Name
Duffy	Document Control
Wang	Document Control
Duffy	Engineering
Wang	Engineering
…	…

Additional Conditions

- You can apply additional conditions in the WHERE clause:

```
SELECT e.EmployeeID, e.LastName, e.DepartmentID,
       d.DepartmentID, d.Name AS DepartmentName
FROM Employees e
  INNER JOIN Departments d
    ON e.DepartmentID = d.DepartmentID
WHERE d.Name = 'Sales'
```

EmployeeID	LastName	Depart-mentID	Depart-mentID	Department-Name
268	Jiang	3	3	Sales
273	Welcker	3	3	Sales
275	Blythe	3	3	Sales

Complex Join Conditions

- Joins can use any Boolean expression in the ON clause:

```
SELECT e.FirstName, e.LastName, d.Name as DeptName
FROM Employees e
  INNER JOIN Departments d
  ON (e.DepartmentId = d.DepartmentId
  AND e.HireDate > '1/1/1999'
  AND d.Name IN ('Sales', 'Finance'))
```

FirstName	LastName	DeptName
Deborah	Poe	Finance
Wendy	Kahn	Finance
…	…	…

Telerik Academy

SQL Language

Inserting Data in Tables

Inserting Data

- INSERT command
  - INSERT INTO <table> VALUES (<values>)
  - INSERT INTO <table>(<columns>) VALUES (<values>)
  - INSERT INTO <table> SELECT <values>

```
INSERT INTO EmployeesProjects
VALUES (229, 25)

INSERT INTO Projects(Name, StartDate)
VALUES ('New project', GETDATE())

INSERT INTO Projects(Name, StartDate)
  SELECT Name + ' Restructuring', GETDATE()
  FROM Departments
```

Bulk Insert

- Bulk INSERT can insert multiple values through a single SQL command

```
INSERT INTO EmployeesProjects VALUES
 (229, 1),
 (229, 2),
 (229, 3),
 (229, 4),
 (229, 5),
 (229, 6),
 (229, 8),
 (229, 9),
 (229, 10),
 (229, 11),
 (229, 12),
 (229, 26)
```

Telerik Academy

SQL Language

Updating Data in Tables

Updating Joined Tables

- We can update tables based on condition from joined tables

```
UPDATE Employees
SET JobTitle = 'Senior ' + JobTitle
FROM Employees e
  JOIN Departments d
    ON e.DepartmentID = d.DepartmentID
WHERE d.Name = 'Sales'
```

Updating Data

- UPDATE command
  - UPDATE <table> SET <column=expression> WHERE <condition>
  - Note: Don't forget the WHERE clause!

```
UPDATE Employees
SET LastName = 'Brown'
WHERE EmployeeID = 1

UPDATE Employees
SET Salary = Salary * 1.10,
  JobTitle = 'Senior ' + JobTitle
WHERE DepartmentID = 3
```

Telerik Academy

SQL Language

Deleting Data From Tables

Deleting Data

- Deleting rows from a table
  - `DELETE FROM <table> WHERE <condition>`

```
DELETE FROM Employees WHERE EmployeeID = 1
DELETE FROM Employees WHERE LastName LIKE 'S%'
```

  - Note: Don't forget the `WHERE` clause!
- Delete all rows from a table at once
  - `TRUNCATE TABLE <table>`

```
TRUNCATE TABLE Users
```

Deleting from Joined Tables

- We can delete records from tables based on condition from joined tables

```
DELETE FROM Employees
FROM Employees e
  JOIN Departments d
    ON e.DepartmentID = d.DepartmentID
WHERE d.Name = 'Sales'
```

Telerik Academy

SQL Syntax in MySQL

MySQL Extensions to Standard SQL

http://dev.mysql.com/doc/refman/5.7/en/extensions-to-ansi.html

SQL Syntax in MySQL

- Specific SQL syntax for paging:

```
SELECT * FROM city LIMIT 100, 10
```

- Commands for retrieving database metadata:

```
SHOW DATABASES
```

```
USE <database>
```

```
SHOW TABLES
```

- Database table maintenance commands:

```
CHECK TABLE <table> / REPAIR TABLE <table>
```

```
OPTIMIZE TABLE <table>
```

Telerik Academy

SQL Syntax in MySQL (2)

- Show table schema

```
DESCRIBE <table_name>
```

- Replace data (delete + insert)

```
REPLACE INTO City(ID, Name, CountryCode, District, Population)
VALUES(100000, 'Kaspichan', 'BG', 'Shoumen', 3300);
```

- Regular expression matching

```
SELECT '0888123456' REGEXP '[0-9]+'
```

- Enumerations as column types

```
CREATE TABLE Shirts (Name VARCHAR(20),
Size ENUM('XS', 'S', 'M', 'L', 'XL'))
```

Telerik Academy

Structured Query Language (SQL)

http://academy.telerik.com

Homework

1. What is SQL? What is DML? What is DDL? Recite the most important SQL commands.
2. What is Transact-SQL (T-SQL)?
3. Start SQL Management Studio and connect to the database TelerikAcademy. Examine the major tables in the "TelerikAcademy" database.
4. Write a SQL query to find all information about all departments (use "TelerikAcademy" database).
5. Write a SQL query to find all department names.
6. Write a SQL query to find the salary of each employee.

Homework (2)

7. Write a SQL to find the full name of each employee.
8. Write a SQL query to find the email addresses of each employee (by his first and last name). Consider that the mail domain is telerik.com. Emails should look like "John.Doe@telerik.com". The produced column should be named "Full Email Addresses".
9. Write a SQL query to find all different employee salaries.
10. Write a SQL query to find all information about the employees whose job title is "Sales Representative".
11. Write a SQL query to find the names of all employees whose first name starts with "SA".

Homework (3)

12. Write a SQL query to find the names of all employees whose last name contains "ei".
13. Write a SQL query to find the salary of all employees whose salary is in the range [20000…30000].
14. Write a SQL query to find the names of all employees whose salary is 25000, 14000, 12500 or 23600.
15. Write a SQL query to find all employees that do not have manager.
16. Write a SQL query to find all employees that have salary more than 50000. Order them in decreasing order by salary.

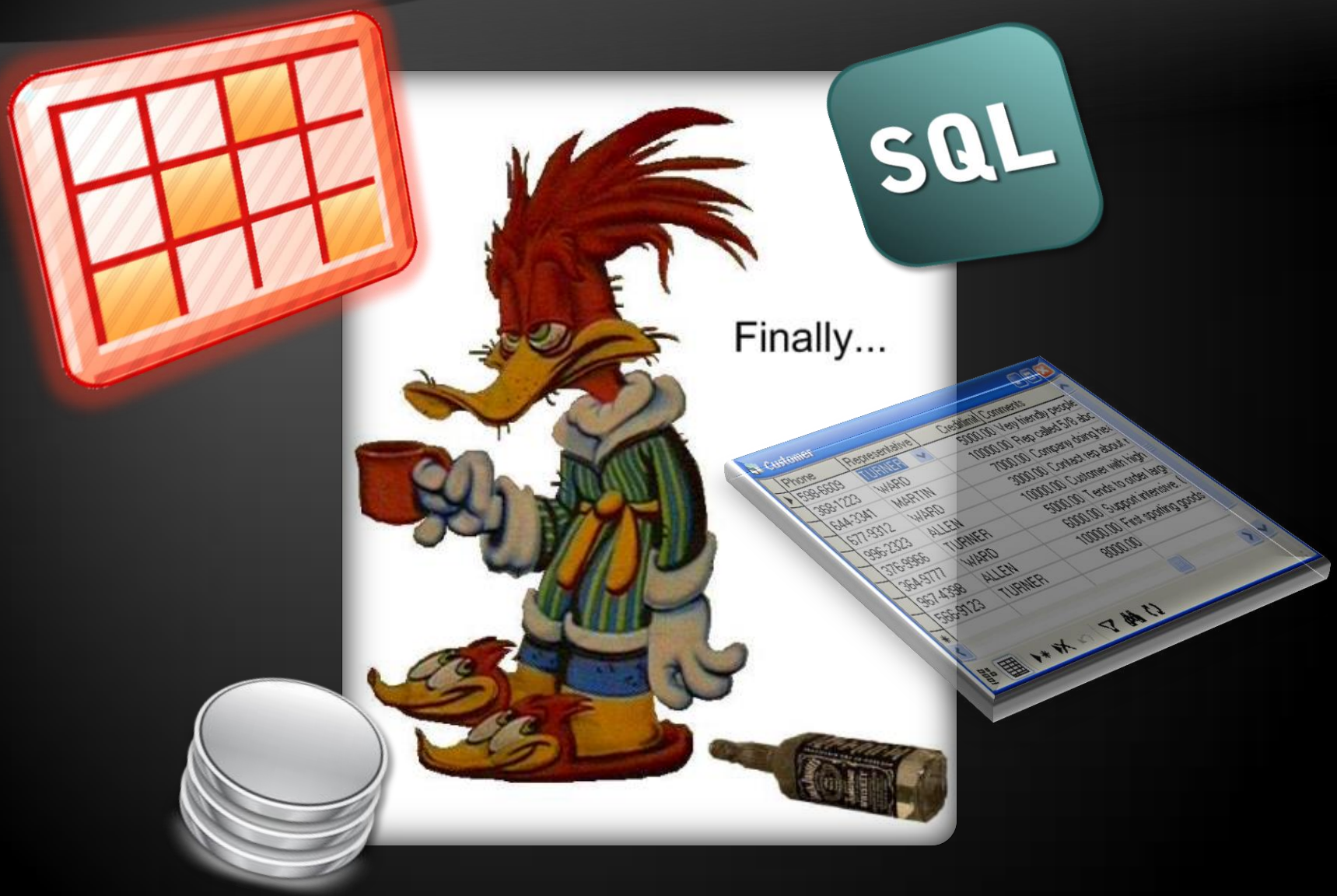
Homework (4)

17. Write a SQL query to find the top 5 best paid employees.
18. Write a SQL query to find all employees along with their address. Use inner join with `ON` clause.
19. Write a SQL query to find all employees and their address. Use equijoins (conditions in the `WHERE` clause).
20. Write a SQL query to find all employees along with their manager.
21. Write a SQL query to find all employees, along with their manager and their address. Join the 3 tables: `Employees e`, `Employees m` and `Addresses a`.

Homework (5)

22. Write a SQL query to find all departments and all town names as a single list. Use `UNION`.
23. Write a SQL query to find all the employees and the manager for each of them along with the employees that do not have manager. Use right outer join. Rewrite the query to use left outer join.
24. Write a SQL query to find the names of all employees from the departments "Sales" and "Finance" whose hire year is between 1995 and 2005.

WTF?


SQL
Finally...
Customer
Phone
Representative
Credit limit
Comments
TURNER
WARD
MARTIN
WARD
ALLEN
TURNER
WARD
ALLEN
TURNER

Free Trainings @ Telerik Academy

- "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy
  - html5course.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com