DATABASE METHODOLOGY

Stockholm University

Structured Query Language SQL

Simple SELECT

Nested SELECT

SELECT with joins

SQL-DML

SQL–DML: The SELECT Command

- The SELECT command is what we use for **reading data** from the database
- A SELECT command, or **query** as it is called, can be:
  - very simple (a few rows)
  - extremely complicated (several A4 pages)
  - anything in between
- In this module you will learn how to use:
  - SELECT with single table (“simple” SELECT)
  - Nested SELECT with multiple tables
  - SELECT with joins of multiple tables

Basic Form of the SELECT Command

In the SELECT *clause*:
Column list
Specifies which column(s)
to be in the result.

In the FROM *clause*:
Table list
Specifies which table(s)
data is to be retrieved from.

```
SELECT name, salary
FROM   Employee
WHERE department = ’Shoes'
```

In the WHERE *clause*:
Condition
Specifies condition(s)
to be fulfilled by the
rows in the result.

The result:
The name and salary of all the employees working at the shoe department.

More About The WHERE Clause

- The WHERE clause is optional:
  - But… used for filtering data…
    - …therefore almost always necessary

SELECT
FROM
WHERE

- The WHERE clause can contain:
  - Comparison operators, =, <>, >, >=, <, <=
  - Logical operators, e.g. AND, OR and NOT
  - Parentheses to control the evaluation
  - BETWEEN … AND for testing intervals
  - LIKE for pattern matching % _
  - IN and EXISTS for handling sets (of tuples)

A Simple SQL Query

Employee

name	salary	manager	department
Berg	20000	Flod	Perfume
Flod	16000	Kvist	Perfume
Bundy	19000	Kvist	Shoes
Kvist	17000	Kvist	Toys
Rot	18000	Flod	Groceries
Sten	18000	Kvist	Perfume

What is the name and department of the employees earning more than 17000?

```
SELECT name, department
FROM Employee
WHERE salary > 17000
```

name	department
Berg	Perfume
Bundy	Shoes
Rot	Groceries
Sten	Perfume

A Simple SQL Query With DISTINCT

Stockholm University

Employee

name	salary	manager	department
Berg	20000	Flod	Perfume
Flod	16000	Kvist	Perfume
Bundy	19000	Kvist	Shoes
Kvist	17000	Kvist	Toys
Rot	18000	Flod	Groceries
Sten	18000	Kvist	Perfume

What is the name of all departments?

SELECT department
FROM Employee

Result ***without*** DISTINCT

department
Perfume
Perfume
Shoes
Toys
Groceries
Perfume

SELECT DISTINCT department
FROM Employee

Result ***with*** DISTINCT

department
Perfume
Shoes
Toys
Groceries

An SQL Query With IN

Employee

name	salary	manager	department
Berg	20000	Flod	Perfume
Flod	16000	Kvist	Perfume
Bundy	19000	Kvist	Shoes
Kvist	17000	Kvist	Toys
Rot	18000	Flod	Groceries
Sten	18000	Kvist	Perfume

Find all employees working in the shoe department or the perfume department! Show name and department.

```
SELECT name, department
FROM Employee
WHERE department IN (’Shoes', ’Perfume')
```

name	department
Berg	Perfume
Flod	Perfume
Bundy	Shoes
Sten	Perfume

A Nested SQL Query With IN

Stockholm University

Employee

<u>name</u>	salary	department
Berg	20000	Perfume
Flod	16000	Perfume
Kvist	17000	Toys
Bundy	19000	Shoes

Department

<u>dname</u>	manager
Perfume	Berg
Toys	Berg
Shoes	Bundy

dname
Perfume
Toys

What is the name and department of the employees that have Berg as manager?

```
SELECT name, department
FROM Employee
WHERE department IN
        (
        SELECT dname
        FROM Department
        WHERE manager = 'Berg'
        )
```

name	department
Berg	Perfume
Flod	Perfume
Kvist	Toys

A Nested SQL Query With EXISTS

Stockholm University

EXISTS returns **true** if the result of a nested query is *not* empty, **false** if it *is* empty.

```
SELECT manager
FROM Department D
WHERE EXISTS (
   SELECT *
   FROM Employee E
   WHERE E.department = D.dname
   AND Salary > 17000)
```

Nested query to test with EXISTS

Department

dname	manager
Perfume	Berg
Toys	Kvist

Employee

name	salary	department
Berg	20000	Perfume
Flod	16000	Perfume
Kvist	17000	Toys

The nested query is here run twice, once for each of the two existing department managers

1

```
SELECT * FROM Employee E
WHERE E.department = ’Perfume’
AND Salary > 17000
```

2

```
SELECT * FROM Employee E
WHERE E.department = ’Toys’
AND Salary > 17000
```

Who manages a department where at least one person earns more than 17000?

manager
Berg

Combining Tables Using Joins

- Joins are used for combining tables
  - Fundamentally crucial property of SQL
  - Most often between PK/FK pairs
  - Join conditions are found in the WHERE-clause.

Department

<u>dname</u>	floor
Perfume	3
Shoes	4

Employee

<u>name</u>	salary	dept
Berg	20000	Perfume
Flod	16000	Perfume
Bundy	19000	Shoes

Employee.dept is FK to Department.dname

Find out at which floor each employee is working! Show name and floor.

```
SELECT name, floor
FROM Employee, Department
WHERE Employee.dept = Department.dname
```

name	floor
Berg	3
Flod	3
Bundy	4

Joins Between Tables - Cont.

Stockholm University

City

city	country
Oslo	23
Madrid	50
Rome	15

Country

coid	name	continent
15	Italy	NULL
23	Norway	5
50	Spain	5
5	Angola	1
29	Peru	7

Continent

cnid	name
1	Africa
2	Antarctica
3	Asia
4	Australia
5	Europe
6	N. America
7	S. America

Find out to which continent each city belongs! Show city name and continent name.

```
SELECT C.city AS City,
       CN.name AS Continent
FROM City C, Country CO,
       Continent CN
WHERE C.country = CO.coid
AND CO.continent = CN.cnid
```

City	Continent
Oslo	Europe
Madrid	Europe

Note that Rome is not in the result. *Why not?*

SQL – DML Summary

- In this presentation you have learnt some basics about the SELECT command, i.e. how to **query** a database for desired data
  - “Simple” SELECT (in one table only)
  - Nested SELECT with IN and EXISTS
  - SELECT with table joins in the WHERE clause

Medverkande

Anders Thelemyr – Lärare

Lars In De Betou – Mediepedagog

Inspelat 2015-09-02
Institutionen för data- och systemvetenskap, DSV

Stockholms universitet