Database basics

Logical operations

Tables used in queries

- The query examples in the following slides utilize the employee and department tables.
  - The employee table has 8 columns and 5 rows of employee data
    - The department_id is a foreign key that refers to the department table, and supervisor_id is a foreign key that refers to the employee table
  - The department table has 4 columns and 3 rows of department data

```
employee table
id  first_name  last_name  locality  salary  phone     department_id  supervisor_id
--  ----------  ---------  --------  ------  --------  -------------  -------------
88  Jukka       Susi       Tampere   5500    444 1234  1
33  Jenni       Joki       Nokia     4300    444 4343  5              88
12  Pekka       Puro       Tampere   3000              5              33
98  Ville       Viima      Lempäälä  4000.5  444 4488  4              88
99  Alli        Kivi       Nokia     2500    444 5555  4              98
```

```
department table
id  name            manager_id  manager_start_date
--  --------------  ----------  ------------------
1   Headquarters    88          2012-06-19
4   Administration  98          2015-01-01
5   Research        33          2018-05-22
```

Logical operations (1/2)

- In SELECT, UPDATE and DELETE statements, conditions in the WHERE clause are used to select rows,
  - which are included in the result table of the query
  - which are updated (data is changed)
  - which are removed

```
SELECT column {, column}
FROM table {,table}
[WHERE condition]

UPDATE table
SET column = expression {, column = expression}
[WHERE condition]

DELETE FROM table
[WHERE condition]
```

- Data manipulation operations are applied to the rows, in which the condition of the WHERE clause yields the truth value true.

Logical operations (2/2)

- In the condition of the WHERE clause, operations AND, OR and NOT can be used.
  - Meanings correspond to logical AND, OR and NOT operations.
- The three truth values of SQL behave in logical expressions according to truth tables presented on slides 5-7.

AND

- A and B are elementary conditions.
- Condition A AND B is true, when both A and B are true.
- The query

```
SELECT last_name, locality, salary
FROM employee
WHERE locality = 'Tampere' AND
      salary > 3000;
```

retrieves last names, localities and salaries for employees whose locality is Tampere and salary exceeds 3000.

```
Result table
last_name  locality  salary
---------  --------  ------
Susi       Tampere   5500
```

A	B	A AND B
true	true	true
true	false	false
true	unknown	unknown
false	true	false
false	false	false
false	unknown	false
unknown	true	unknown
unknown	false	false
unknown	unknown	unknown

OR

- Condition A OR B is true, when
  - either A or B is true
  - both A and B are true
- The query

```
SELECT last_name, locality, salary
FROM employee
WHERE locality = 'Tampere' OR
      salary > 3000;
```

retrieves last names, localities and salaries for employees whose locality is Tampere or whose salary exceeds 3000.

```
Result table
last_name  locality  salary
---------  --------  ------
Susi       Tampere   5500
Joki       Nokia     4300
Puro       Tampere   3000
Viima      Lempäälä  4000.5
```

A	B	A OR B
true	true	true
true	false	true
true	unknown	true
false	true	true
false	false	false
false	unknown	unknown
unknown	true	true
unknown	false	unknown
unknown	unknown	unknown

NOT

- Condition NOT A is true,
  - when A is false
- The query

```
SELECT last_name, locality
FROM employee
WHERE NOT locality = 'Tampere';
```

retrieves last names and localities for employees whose locality is not Tampere.

A	NOT A
true	false
false	true
unknown	unknown

```
Result table
last_name  locality
---------  --------
Joki       Nokia
Viima      Lempäälä
Kivi       Nokia
```

Order of evaluation

- The operations are prioritized as follows:
  1. NOT
  2. AND
  3. OR
  - The value of the NOT operation is therefore evaluated first when evaluating the truth value of the entire expression.
  - If an expression consists of several of the same operations, proceed from left to right when evaluating the truth value of the whole expression.
- The order of evaluation can be changed by parentheses.
  - Parentheses have the highest priority: parentheses are evaluated first, then NOT, AND and OR.
  - If there are nested parentheses in the expression, proceed from the innermost parentheses to the outermost.
- You can use parentheses for clarity as well.

Order of evaluation - example 1 (1/3)

- The query

```
SELECT last_name, locality, salary
FROM employee
WHERE locality = 'Nokia' OR
      locality = 'Tampere' AND
      salary > 3000;
```

retrieves last name, locality and salary for employees
  - whose locality is Nokia
  - whose locality is Tampere and whose salary exceeds 3000

```
Result table of the 1st query
last_name  locality  salary
---------  --------  ------
Susi       Tampere   5500
Joki       Nokia     4300
Kivi       Nokia     2500
```

- Parentheses change the order of evaluation and the semantics of the query.
- The query

```
SELECT last_name, locality, salary
FROM employee
WHERE (locality = 'Nokia' OR
      locality = 'Tampere') AND
      salary > 3000;
```

retrieves last name, locality and salary for employees
  - whose locality is Nokia or Tampere and whose salary exceeds 3000

```
Result table of the 2nd query
last_name  locality  salary
---------  --------  ------
Susi       Tampere   5500
Joki       Nokia     4300
```

Order of evaluation - example 1 (2/3)

- What truth value does the WHERE clause

```
WHERE locality = 'Nokia' OR
      locality = 'Tampere' AND
      salary > 3000
```

yield when applied to the row of the employee whose id is 99?

```
employee-taulu
id  first_name  last_name  locality  salary  phone     department_id  supervisor_id
--  ----------  ---------  --------  ------  --------  -------------  -------------
99  Alli        Kivi       Nokia     2500    444 5555  4              98
```

	locality = 'Nokia'	OR	locality = 'Tampere'	AND	salary > 3000
1.	true		false		false
2.				false	
3.		true			

Order of evaluation - example 1 (3/3)

- What truth value does the WHERE clause

```
WHERE (locality = 'Nokia' OR
       locality = 'Tampere') AND
       salary > 3000
```

yield when applied to the row of the employee whose id is 99?

```
employee-taulu
id  first_name  last_name  locality  salary  phone     department_id  supervisor_id
--  ----------  ---------  --------  ------  --------  -------------  -------------
99  Alli        Kivi       Nokia     2500    444 5555  4              98
```

	(locality = 'Nokia'	OR	locality = 'Tampere')	AND	salary > 3000
1.	true		false		false
2.		true			
3.				false	

Order of evaluation - example 2 (1/5)

- Suppose you want to make a query that retrieves last names and department names for employees who work in the Administration or Research department.
- Let's try writing the query like this:

```
SELECT last_name, name
FROM employee, department
WHERE employee.department_id = department.id AND
      name = 'Administration' OR
      name = 'Research';
```

- The result table of the query has seven rows. Some employees are incorrectly connected to the Research department.

```
Result table of the erroneous query
last_name  name
---------  --------------
Viima      Administration
Kivi       Administration
Susi       Research
Joki       Research
Puro       Research
Viima      Research
Kivi       Research
```

Order of evaluation - example 2 (2/5)

- The join condition and selection conditions of the WHERE clause

```
employee.department_id = department.id AND
name = 'Administration' OR
name = 'Research'
```

yield the truth value true on the rows of the Cartesian product

- which have the same department ids and the department name is Administration
- where the department name is Research

id	first_name	last_name	department_id	supervisor_id	id	name
88	Jukka	Susi	1		1	Headquarters
88	Jukka	Susi	1		4	Administration
88	**Jukka**	**Susi**	**1**		**5**	**Research**
33	Jenni	Joki	5	88	1	Headquarters
33	Jenni	Joki	5	88	4	Administration
33	**Jenni**	**Joki**	**5**	**88**	**5**	**Research**
12	Pekka	Puro	5	33	1	Headquarters
12	Pekka	Puro	5	33	4	Administration
12	**Pekka**	**Puro**	**5**	**33**	**5**	**Research**
98	Ville	Viima	4	88	1	Headquarters
98	**Ville**	**Viima**	**4**	**88**	**4**	**Administration**
98	**Ville**	**Viima**	**4**	**88**	**5**	**Research**
99	Alli	Kivi	4	98	1	Headquarters
99	**Alli**	**Kivi**	**4**	**98**	**4**	**Administration**
99	**Alli**	**Kivi**	**4**	**98**	**5**	**Research**

Order of evaluation - example 2 (3/5)

- Let's fix the evaluation order of the WHERE clause and the semantics of the query with parentheses. Now the WHERE clause

```
employee.department_id = department.id AND
(name = 'Administration' OR
name = 'Research')
```

yield the truth value true on the rows of the Cartesian product

- which have the same department ids and department name is Administration or Research

id	first_name	last_name	department_id	supervisor_id	id	name
88	Jukka	Susi	1		1	Headquarters
88	Jukka	Susi	1		4	Administration
88	Jukka	Susi	1		5	Research
33	Jenni	Joki	5	88	1	Headquarters
33	Jenni	Joki	5	88	4	Administration
33	**Jenni**	**Joki**	**5**	**88**	**5**	**Research**
12	Pekka	Puro	5	33	1	Headquarters
12	Pekka	Puro	5	33	4	Administration
12	**Pekka**	**Puro**	**5**	**33**	**5**	**Research**
98	Ville	Viima	4	88	1	Headquarters
98	**Ville**	**Viima**	**4**	**88**	**4**	**Administration**
98	Ville	Viima	4	88	5	Research
99	Alli	Kivi	4	98	1	Headquarters
99	**Alli**	**Kivi**	**4**	**98**	**4**	**Administration**
99	Alli	Kivi	4	98	5	Research

Order of evaluation - example 2 (4/5)

- The result table of the whole query

```
SELECT last_name, name
FROM employee, department
WHERE employee.department_id = department.id AND
      (name = 'Administration' OR
       name = 'Research');
```

```
Result table of the semantically correct query
last_name  name
---------  --------------
Viima      Administration
Kivi       Administration
Joki       Research
Puro       Research
```

Order of evaluation - example 2 (5/5)

- The query can also be written using the INNER JOIN operation. In this case, parentheses can be omitted around the combined selection condition.

```
SELECT last_name, name
FROM employee INNER JOIN department
     ON employee.department_id = department.id
WHERE name = 'Administration' OR
      name = 'Research';
```

```
Result table of the semantically correct query
last_name  name
---------  --------------
Viima      Administration
Kivi       Administration
Joki       Research
Puro       Research
```

Material

- The slides are based on the following materials:
  - Elmasri and Navathe. Database systems: models, languages, design, and application programming. Pearson, 2011.