# Practical class script Nr 3.

# Selecting the Database

Show and explain student database model.

Show student database creation scripts, explain each of the statements mean, also the meaning of GO statement.

Create new student database.

Show database import scripts, import data into the database.

***Practice – students create their database and import the data.***

# Simple SQL Select queries

Read all the records from Teacher table.

**Read all the columns & records *SELECT \* FROM <table\_name>***

SELECT \* FROM Teacher

**Select specific columns from the table *SELECT column1, column2, ..., columnN FROM <table\_name>***

SELECT Name, Surname

FROM Teacher

**Renaming columns for select only *SELECT column1 new\_name1, column2 new\_name2, ..., columnN new\_nameN FROM <table\_name>***

*The select will return the same columns, but will name them according to the aliases (new\_name) specified*

SELECT Name AS PersonName, Surname AS PersonSurname

FROM Teacher

***Selecting distinct values for a column SELECT DISTINCT***

Selects all the distinct values for Course\_ID column

SELECT DISTINCT Course\_ID

FROM Examination

***Expressions and simple calculated columns***

Concatenate two string columns

SELECT Name+' '+Surname

FROM Teacher

Arithmetic operations

SELECT Grade+5 FROM Grade

Concatenated strings can be defined as a separate column using column alias syntax

SELECT Name+' '+Surname AS FullName

FROM Teacher

***Row conditioning with WHERE***

WHERE command helps to filter out some of the rows in the SELECT statement.

Select teachers, which have Teacher\_ID>2010

SELECT \* FROM Teacher

WHERE Teacher\_ID>2010

Select Teachers without Mentors (where Mentor\_ID is NULL)

SELECT \* FROM Teacher

WHERE Mentor\_ID IS NULL

***Comparison (<,=,>,<=.>=,<>), BETWEEN, IN***

*Column1 BETWEEN value1 AND value2* is the same as *column1 >= value1 AND column1 <= value2*

*Column1 IN (value1, value2, …, valueN)* is the same as *column1=value1 OR column1=value2 OR.... OR column1=valueN*

Select all the grades which are greater than 8

SELECT \* FROM Grade

WHERE Grade>8

Select grades between 5 and 8

SELECT \* FROM Grade

WHERE Grade BETWEEN 5 AND 8

Select grades with 5, 7 or 9

SELECT \* FROM Grade

WHERE Grade IN (5, 7, 9)

***LIKE***

LIKE allows to compare a column value using a simplistic regular expression language. There are two special symbols. % - denotes multiple symbols and \_ - denotes single symbol.

For example

*SELECT \* FROM Customers WHERE City LIKE 'ber%';*

Would match cities like Berlin, Bern, Bergamo

Selecting students with Z as the first name character

SELECT \* FROM Student

WHERE Name LIKE 'Z%'

Finding students which have “el” in their name

SELECT \* FROM Student

WHERE Name LIKE '%el%'

***IS NULL***

*A special comparison IS NULL allows to find rows with empty values*

Find teachers who does not have a mentor

SELECT \* FROM Teacher

WHERE Mentor\_ID IS NULL

Select teachers who has a mentor

SELECT \* FROM Teacher

WHERE Mentor\_ID IS NOT NULL

***Comparison with logical operators AND, OR, NOT***

Condition expressions can be combined with logical operators AND, OR, NOT.

Select examinations, for Course with ID 30018 and Description equals “Exam”

SELECT \* FROM Examination

WHERE Course\_ID=30018 AND Description = 'Exam'

***Sorting the results with ORDER BY***

The resulting data can be ordered by one or multiple fields using command ORDER BY.

Select student names and surnames, order the results by surname.

SELECT Surname, Name

FROM Student

ORDER BY Surname

By default, the records are sorted in ascending (ASC) order. To order records in descending (DESC) order, use DESC keyword.

SELECT Surname, Name

FROM Student

ORDER BY Surname DESC

In case you need to order the results by all the fields, you need to write all the fields in the ORDER BY statement. The records would be sorted by the first field, but when the first field values match (are equal), those will be order by second field, etc. In case one of the fields needs sorting in descending order, add the DESC keyword after the field name.

Select surname and name, sorting by surname and name

SELECT Surname, Name

FROM Student

ORDER BY Surname, Name

Select surname and name, sorting by surname in descending order and name in ascending order

SELECT Surname, Name

FROM Student

ORDER BY Surname DESC, Name

Fetch all examinations, order by Course ID and examination date

SELECT \* FROM Examination

ORDER BY Course\_ID, Date

***Combining results using UNION***

Fetch teachers, who are responsible for any examination.   
Combine those results with Teachers who do not have a mentor.

SELECT Distinct Responsible\_teacher\_ID FROM Examination

UNION

SELECT Teacher\_ID FROM Teacher

WHERE Mentor\_ID IS NULL