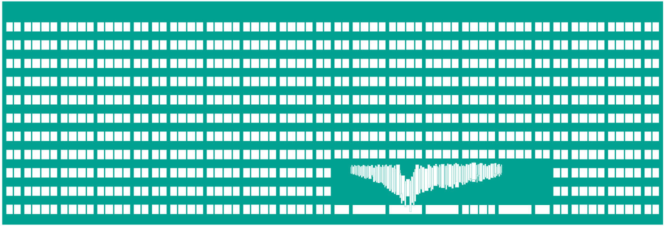
VŠB TECHNICKÁ
UNIVERZITA
OSTRAVA

VSB TECHNICAL
UNIVERSITY
OF OSTRAVA



www.vsb.cz

Database and Information Systems

db.cs@vsb.cz

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VSB - Technical University of Ostrava

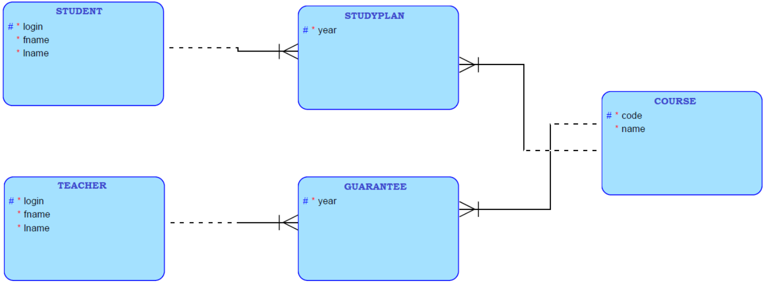
2019/2020

Task 1: Database Connection

1. Run Oracle SQL Developer[1].
2. Create connection to a database:
   host: dbsys.cs.vsb.cz; port: 1521; SID: oracle
3. Connect to a server and use credentials received by email
4. Change your password using the SQL command
   `ALTER USER <login> IDENTIFIED BY <password>`

[1] https://www.oracle.com/database/technologies/appdev/sql-developer.html

Task 2: Data Model



Analyze data model[2]:

1. Which attributes should be part of entity types `StudyPlan` and `Guarantee` to realize connection N:1 with other entity types?
2. Define keys of entity types `StudyPlan` and `Guarantee`.

[2] Dotted line means non-required members, dotted lines in `StudyPlan` and `Guarantee` mean that entity type key is a part of primary key.

Task 2.1: Create Table 1/2

1. Create SQL script `create.sql` to create tables according data model
2. Data types are as below:

   `Student.login`: string with constant length 6 characters
   `Teacher.login`: string with constant length 5 characters
   `Student.fname`, `Teacher.fname` : string with variable length up to 30 characters
   `Student.lname`, `Teacher.lname` : string with variable length up to 50 characters
   `StudyPlan.year`, `Guarantee.year`: integer
   `Course.code`: string with constant length 11 characters
   `Course.name`: string with variable length up to 50 characters

Task 2.1: Create Table 2/2

Define for all tables:

1. primary keys,
2. foreign keys,
3. required attributes.

Task 2.2: Scripts delete.sql and drop.sql

Create scripts:

- `delete.sql`, which will delete records from tables
- `drop.sql`, which will drop created tables

Task 2.3: Inserts, script init.sql

Create SQL script `init.sql`, which will insert data into the tables:

- 3 records into table `Student`:
  ('pla457', 'John', 'Mnemonik')
  ('sob458', 'Jake', 'Casper')
- 2 records into table `Course`:
  ('456-dais-01', 'Database and information systems')
  ('456-tzd-01', 'Theory of data')
- 2 records into table `Teacher`:
  ('bay01', 'Joseph', 'Bayer')
  ('cod02', 'Peter', 'Codd')
- Insert 3 records in the year 2009 for each student into table `StudyPlan`
- Insert 2 records in the year 2009 for each teacher into table `Guarantee`

Task 2.4: Select

Create SQL commands:

- List of students (all attributes), who have in the year 2009 courses with teacher `Codd`.
- List of courses, which student `Mnemonik` has in the year 2009.
- List of courses, which student `Mnemonik` has.
- List of courses assigned by at least one student in the year 2009 (each course only once).
- List of teachers, who in year 2009 teach courses, assigned by at least one student in the year 2009 (each teacher only once).

Task 2.5: Alter Table

- Add into table `Teacher` attributes `workBeg` and `workEnd` with information about dates of work contracts begin and end.
- Insert several records into table `Teacher`, for insert use function `TO_DATE`[3][4].
- Get a list of all teachers, who teach in year 2009 at least one course and their contract is longer than 3 years[5].

[3] https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/TO_DATE.html

[4] For example: `TO_DATE('15.08.2009', 'DD.MM.YYYY')`

[5] https://www.oracletutorial.com/oracle-date-functions/

Task 3: Grant/Revoke Privileges

1. Study GRANT and REVOKE SQL commands[6].
2. Grant a privilege for a processing a SELECT operation on Student table to your collegue using the GRANT command.
3. Check the privilege for a SELECT was granted using SQL command:

   ```
   SELECT * FROM <login>.Student;
   ```
4. Revoke the privilege using the REVOKE command.
5. Check the privilege was removed.

[6] http://www.techonthenet.com/oracle/grant_revoke.php

Task 4.1: System Catalog 1/3

1. Study system catalog[7].
2. Get the list of tables where the owner is the actual user

   ```
   SELECT * FROM USER_TABLES;
   ```

3. Using SQL command

   ```
   SELECT * FROM USER_TAB_COLUMNS WHERE
   TABLE_NAME='STUDENT';
   ```

   get from system catalog information about `Student` table attributes.

[7] http://docs.oracle.com/html/B10100_01/wncat.htm

Task 4.2: System Catalog 2/3

1. Select the names of the actual user tables.
2. Select names and owners of tables, where actual user has at least one access priviledge.
3. Select names of collumns and their data types of the chosen table [8].

[8] Information about particular columns is possible to get by command `SELECT * FROM ALL_TAB_COLUMNS WHERE TABLE_NAME='<table_name>';`, e.g. `SELECT * FROM ALL_TAB_COLUMNS WHERE TABLE_NAME='USER_TABLES';`

Task 4.3: System Catalog 3/3

1. Study commands necessary for adding comments for tables and their attributes.
2. Add a comment for table `Student` and attribute `lname`.
3. Get the comments from a system catalog.
4. Study tables of system catalog containing the user privileges.
5. Print all privileges granted by a selected user.

Task 5: Build-in SQL functions

1. Study build-in SQL function [9].
2. Some attributes of the system catalog contain the table names in upper case. Therefore, we can not use:

   ```
   SELECT COLUMN_NAME, DATA_TYPE FROM
   USER_TAB_COLUMNS WHERE TABLE_NAME='Student';
   ```
3. Use the build-in function `UPPER` to solve the problem.

[9] http://www.sqlinfo.net/oracle/oracle_function_upper_lower.php

Task 6.1: CHECK

1. Define new attribute `Age` in the table `Student`:

   ```
   ALTER TABLE Student ADD Age DECIMAL(5,2)
   CHECK(VALUE BETWEEN 0 AND 150);
   ```

2. Study the restrictions for attributes and tables: Check, NOT NULL, Primary Key, References[10].
3. Add new attribute `department` into the table `Teacher` with allowed values 100, 200, 300.

[10] https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/constraint.html

Task 6.2: Another SQL queries

1. Select course with at least student in age between 20 and 26 years. The course can occur only once in the result. Use construction `BETWEEN` in the query.
2. Select all students of the courses with codes 456098 and 456102. Use construction `IN` in the query.
3. Select all students together with the number of their courses in the year 2009.
4. Select teachers according to their departments.

Task 7: LIKE, NOT LIKE

Insert students with the surnames: `Novak, Novotny, Novakova, Holy, Hladky, Klo_kan` into the table `Student`. Select all students with the surname[11]:

1. begins by `"Nov"`
2. ends by `"y"`
3. contains `"a"`
4. does not contain `"n"`
5. has second letter `"o"`
6. contains `"_"`

[11] https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/Pattern-matching-Conditions.html

Task 8: NULLIF, NVL, NVL2, COALESCE

1. Select all teachers with the empty department. Print out "Not defined" instead of empty value `(NVL)`[12].
2. Select all teachers with their departments. Print out "Unknown" in the case of empty value, otherwise print out "Specified" `(NVL2)`.
3. Explain when function `NULLIF` will return `NULL` in following query:

   ```
   SELECT NULLIF(fname, lname) result FROM Teacher
   ```
4. What following query returns[13]?

   ```
   SELECT COALESCE(department, fname) FROM Teacher
   ```

[12] https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/NVL.html

[13] https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/COALESCE.html