

Databases: Exercises 9 (25p/25p)

Assignment 1 [15p/15p]

Task: Design and implement the database.

Subject: register of certified competence (TOR)

Client: Ministry of Education and Culture or e.g. project office of a company/educational institution.

More information: Ari Rantala

Mandatory functions/recordable data:

P1. Maintenance of personal data (at least the name of the person).

P2. Maintenance of skills information (at least the name of the skill and a more detailed description).

P3. The person's self-assessment and/or the expert's assessment of the person's competence (on a scale of 0-10).

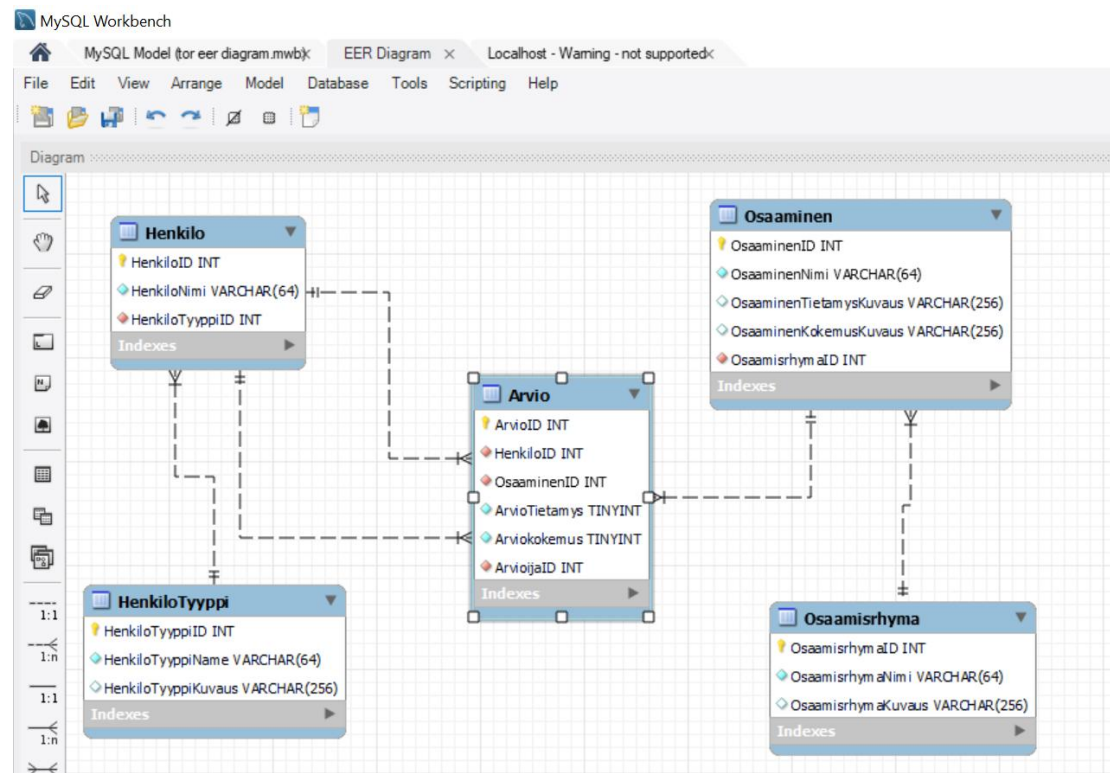
More information and requirements

Examples of competences, competence descriptions and their groupings: certification self-assessment form.xls

The database does not store competence information for individuals without an assessment (self-assessment or expert assessment)

As personal data, it must be possible to store information about the type of person with descriptions (e.g. Tavis -> Ordinary evaluator; Expert -> OKM expert evaluator; ...)

One evaluator can give a certain person ONLY ONE evaluation per competence!



Explanation:

- 1) An henkilotyyppi table has different types of henkilo, where henkilotyyppiid as primary key, henkilotyyppiname with its descriptions and each henkilotyyppiid has many henkiloid in henkilo table.
- 2) An henkilo table has henkiloid as primary key, henkiloname and henkilotyyppiid as reference key.
- 3) An osaaminenrhyma table has osaaminenrhymaid as primary key, osaaminenrhymaname with its description and each osaaminenrhymaid has many osaaminenid in osaaminen table.
- 4) An Osaaminenrhyma table has osaaminenid as primary key, osaaminenrhymaname, osaaminenrhyma description and osaaminenrhymaid.
- 5) An henkiloid in henkilo table has many to many connection with the osaaminenid in osaaminen table with this connection or relationship we have table called arvio.
- 6) An arvio table has arvioid as primary key, henkiloid, osaaminenid as reference which are primary keys in the henkilo table and osaaminen table respectively, arviotietamys, arviokokemus, along with we have arviojiaid which is the reference key and connection with henkiloid in henkilo table due to one(henkilo) to many(arvio) connection.

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----
-- Schema tor_db
-----
```

```
-----
-- Schema tor_db
-----
```

```
CREATE SCHEMA IF NOT EXISTS `tor_db` ;
USE `tor_db` ;
```

```
-----
-- Table `tor_db`.`HenkiloTyyppi`
-----
```

```
CREATE TABLE IF NOT EXISTS `tor_db`.`HenkiloTyyppi` (
  `HenkiloTyyppiID` INT NOT NULL AUTO_INCREMENT,
  `HenkiloTyyppiName` VARCHAR(64) NOT NULL,
  `HenkiloTyyppiKuvaus` VARCHAR(256) NULL,
  PRIMARY KEY (`HenkiloTyyppiID`))
ENGINE = InnoDB;
```

```
-----
-- Table `tor_db`.`Henkilo`
-----
```

```
CREATE TABLE IF NOT EXISTS `tor_db`.`Henkilo` (
  `HenkiloID` INT NOT NULL AUTO_INCREMENT,
  `HenkiloNimi` VARCHAR(64) NOT NULL,
  `HenkiloTyyppiID` INT NOT NULL,
  PRIMARY KEY (`HenkiloID`),
  INDEX `fk_Henkilo_HenkiloTyyppi1_idx` (`HenkiloTyyppiID` ASC),
```

```

CONSTRAINT `fk_Henkilo_HenkiloTyyppi1`
  FOREIGN KEY (`HenkiloTyyppiID`)
  REFERENCES `tor_db`.`HenkiloTyyppi` (`HenkiloTyyppiID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `tor_db`.`Osaamisrhyma`
-----

```

```

CREATE TABLE IF NOT EXISTS `tor_db`.`Osaamisrhyma` (
  `OsaamisrhymaID` INT NOT NULL AUTO_INCREMENT,
  `OsaamisrhymaNimi` VARCHAR(64) NOT NULL,
  `OsaamisrhymaKuvaus` VARCHAR(256) NULL,
  PRIMARY KEY (`OsaamisrhymaID`))
ENGINE = InnoDB;

```

```

-----
-- Table `tor_db`.`Osaaminen`
-----

```

```

CREATE TABLE IF NOT EXISTS `tor_db`.`Osaaminen` (
  `OsaaminenID` INT NOT NULL AUTO_INCREMENT,
  `OsaaminenNimi` VARCHAR(64) NOT NULL,
  `OsaaminenTietamysKuvaus` VARCHAR(256) NULL,
  `OsaaminenKokemusKuvaus` VARCHAR(256) NULL,
  `OsaamisrhymaID` INT NOT NULL,
  PRIMARY KEY (`OsaaminenID`),
  INDEX `fk_Osaaminen_Osaamisrhyma1_idx` (`OsaamisrhymaID` ASC),
  CONSTRAINT `fk_Osaaminen_Osaamisrhyma1`
    FOREIGN KEY (`OsaamisrhymaID`)
    REFERENCES `tor_db`.`Osaamisrhyma` (`OsaamisrhymaID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `tor_db`.`Arvio`
-----

```

```

CREATE TABLE IF NOT EXISTS `tor_db`.`Arvio` (
  `ArvioID` INT NOT NULL AUTO_INCREMENT,
  `HenkiloID` INT NOT NULL,
  `OsaaminenID` INT NOT NULL,
  `ArvioTietamys` TINYINT NOT NULL,
  `Arviokokemus` TINYINT NOT NULL,
  `ArvioijaID` INT NOT NULL,
  PRIMARY KEY (`ArvioID`),
  INDEX `fk_Henkilo_has_Osaaminen_Osaaminen1_idx` (`OsaaminenID` ASC),
  INDEX `fk_Henkilo_has_Osaaminen_Henkilo_idx` (`HenkiloID` ASC),
  INDEX `fk_Arvio_Henkilo1_idx` (`ArvioijaID` ASC),
  UNIQUE INDEX `uq_HenkiloID_OsaaminenID_ArvioijaID` (`HenkiloID` ASC, `OsaaminenID` ASC, `ArvioijaID` ASC),
  CONSTRAINT `fk_Henkilo_has_Osaaminen_Henkilo`
    FOREIGN KEY (`HenkiloID`)
    REFERENCES `tor_db`.`Henkilo` (`HenkiloID`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Henkilo_has_Osaaminen_Osaaminen1`
FOREIGN KEY (`OsaaminenID`)
REFERENCES `tor_db`.`Osaaminen` (`OsaaminenID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Arvio_Henkilo1`
FOREIGN KEY (`ArvioijaID`)
REFERENCES `tor_db`.`Henkilo` (`HenkiloID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Justification:

Created a table by forward engineer in mysql workbench, everything went smooth and the unique indexing is mentioned for the arvio table with the fields henkiloid, osaaminenid and the arvioijaid. Check constraint is also applied for the table arvio with alter table both arviokokemus and arviokokemus are from 0 to 10.

Task 2 [5p/5p]

Add information to your TOR database so that each board has at least two lines and is stored there
The SQL queries of the answer to the assignment must be in text format.

A. fictitious information of at least two laymen and two experts

```

ALTER TABLE henkilotyyppi AUTO_INCREMENT=201;

insert into henkilotyyppi (HenkiloTyyppiName, HenkiloTyyppiKuvaus)
values
('Erityisasiantuntija', 'Työskentelemään erityisosaamista vaativissa
asiantuntijatehtävissä'),
('Sihteeri', 'Joka työskentelee toimistopalvelu');

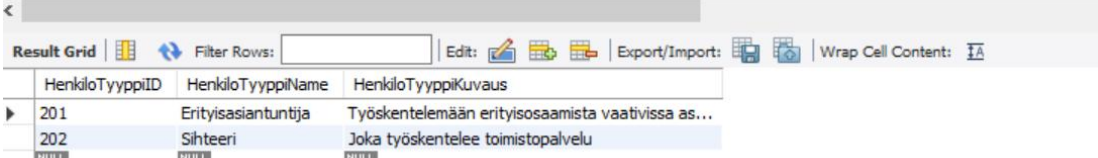
select * from henkilotyyppi;

```

```

6 • ALTER TABLE henkilotyyppi AUTO_INCREMENT=201;
7
8 • insert into henkilotyyppi (HenkiloTyyppiName, HenkiloTyyppiKuvaus)
9 values
10 ('Erityisasiantuntija', 'Työskentelemään erityisosaamista vaativissa asiantuntijatehtävissä'),
11 ('Sihteeri', 'Joka työskentelee toimistopalvelu');
12
13 • select * from henkilotyyppi;
14

```



HenkiloTyyppiID	HenkiloTyyppiName	HenkiloTyyppiKuvaus
201	Erityisasiantuntija	Työskentelemään erityisosaamista vaativissa as...
202	Sihteeri	Joka työskentelee toimistopalvelu
NULL	NULL	NULL

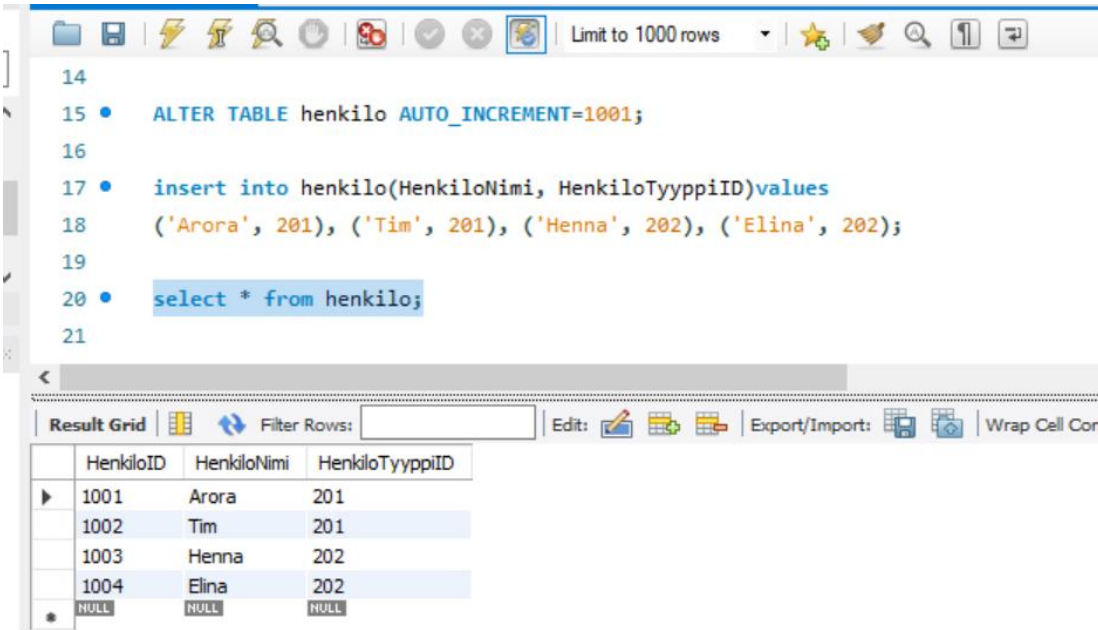
```

ALTER TABLE henkilo AUTO_INCREMENT=1001;

insert into henkilo(HenkiloNimi, HenkiloTyyppiID)values
('Arora', 201), ('Tim', 201), ('Henna', 202), ('Elina', 202);

select * from henkilo;

```



```

14
15 • ALTER TABLE henkilo AUTO_INCREMENT=1001;
16
17 • insert into henkilo(HenkiloNimi, HenkiloTyyppiID)values
18 ('Arora', 201), ('Tim', 201), ('Henna', 202), ('Elina', 202);
19
20 • select * from henkilo;
21

```

HenkiloID	HenkiloNimi	HenkiloTyyppiID
1001	Arora	201
1002	Tim	201
1003	Henna	202
1004	Elina	202
NULL	NULL	NULL

B. at least two different competences per competence group (Technical qualifications, etc.)

```

ALTER TABLE osaamisrhyma AUTO_INCREMENT=51;

insert into Osaamisrhyma (OsaamisrhymaNimi, OsaamisrhymaKuvaus)
values
('Tekniset pätevyyydet', 'se on tarvitaan projektin alulle panossa tai
aloituksessa, johtamisessa ja lopettamisessa'),

```

```

('Käytösptevyydet', 'Ne perustuvat muutamien käytöstä kuvaaviin
lähdedokumentteihin ja niihin sisältyy henkilokehtaisia asenteita'),
('Toteutusympäristö pätevyudet', 'se tarkoittaa nimenomaan
kulloiseakin projektin alaan liittyvää osaamista');

```

```

select *from osaamishyma;

```

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane shows a tree view with 'henkilotyyppi', 'osaaminen', and 'osaamishyma'. The 'Information' pane shows details for the 'osaamishyma' table, including columns: 'OsaamishymaID' (int(11) AI PK), 'OsaamishymaNimi' (varchar(64)), and 'OsaamishymaKuvaus' (varchar(25)). The main pane displays a SQL query and its results.

```

27 ALTER TABLE osaamishyma AUTO_INCREMENT=51;
28
29
30 insert into Osaamishyma (OsaamishymaNimi, OsaamishymaKuvaus) values
31 ('Tekniset pätevyudet', 'se on tarvitaan projektin alulle panossa tai aloituksessa, johtamisessa ja lopettamisessa'),
32 ('Käytösptevyydet', 'Ne perustuvat muutamien käytöstä kuvaaviin lähdedokumentteihin ja niihin sisältyy henkilokehtaisia asenteita'),
33 ('Toteutusympäristö pätevyudet', 'se tarkoittaa nimenomaan kulloiseakin projektin alaan liittyvää osaamista');
34
35 select *from osaamishyma;

```

The 'Result Grid' shows the following data:

OsaamishymaID	OsaamishymaNimi	OsaamishymaKuvaus
51	Tekniset pätevyudet	se on tarvitaan projektin alulle panossa tai aloituksessa, johtamisessa ja lopettamisessa
52	Käytösptevyydet	Ne perustuvat muutamien käytöstä kuvaaviin lähdedokumentteihin ja niihin sisältyy henkilokehtaisia asenteita
53	Toteutusympäristö pätevyudet	se tarkoittaa nimenomaan kulloiseakin projektin alaan liittyvää osaamista

```

ALTER TABLE osaaminen AUTO_INCREMENT=21;

```

```

insert into Osaaminen (OsaaminenNimi, OsaaminenTietamysKuvaus,
OsaaminenKokemusKuvaus, OsaamishymaID) values
('Projektinjohtollinen onnistuminen', NULL, 'on yhtä kuin
projektinjohtamisen tulosten saama arvostus olennaisilta
sidosryhmiltä',
51),
('Viestintä', 'Viestintä kattaa tehokkaan tiedonvaihdon ja
ymmärryksen osapuolten kesken', NULL, 51),
('Itsehillintä', NULL, 'Itsehillintä tai itsensä hallinta on
systemaattinen ja kurinalainen
lähestymistapa selviytyä päivittäisestä työstä vaatimusten muuttuessa
ja stressaavista tilanteista', 52),
('Luovuus', 'Luovuus on kyky ajatella ja toimia omalaatuisilla ja
mielikuvituksellisilla tavoilla',
'Projektipäällikkö käyttää yksilöiden sekä projektiryhmän ja
organisaation, jossa he työskentelevät, kollektiivista
luovuutta projektinsa eduksi', 52),
('Liiketoiminta', 'se on teollista, kaupallista tai asiantuntija
toimintaa sisältäen korvauksen tavaroista tai palveluista',
'Heillä on kokemus ja tiedän hyvin liiketoiminta tietoa', 53),
('Rahoitus', NULL, NULL, 53);

```

```

select * from osaaminen;

```

Limit to 1000 rows

```

42 'Projektipäällikkö käyttää yksilöiden sekä projektiryhmän ja organisaation, jossa he työskentelevät, kollektiivista
43 luovuutta projektinsa eduksi', 52),
44 ('Liiketoiminta', 'se on teollista, kaupallista tai asiantuntija toimintaa sisältäen korvauksen tavaroista tai palvelu...
45 'Heillä on kokemus ja tiedän hyvin liiketoiminta tietoa', 53),
46 ('Rahoitus', NULL, NULL, 53);
47
48 • select * from osaaminen;
49

```

OsaaminenID	OsaaminenNimi	OsaaminenTietamysKuvaus	OsaaminenKokemusKuvaus	OsaamisryhmaID
21	Projektinjohdollinen onnistuminen	NULL	on yhtä kuin projektinjohtamisen tulosten saam...	51
22	Viestintä	Viestintä kattaa tehokkaan tiedonvaihdon ja ym...	NULL	51
23	Itsehillintä	NULL	Itsehillintä tai itsensä hallinta on systemaattinen...	52
24	Luovuus	Luovuus on kyky ajatella ja toimia omalaatuisilla...	Projektipäällikkö käyttää yksilöiden sekä projekti...	52
25	Liiketoiminta	se on teollista, kaupallista tai asiantuntija toimin...	Heillä on kokemus ja tiedän hyvin liiketoiminta ti...	53
26	Rahoitus	NULL	NULL	53
•	NULL	NULL	NULL	NULL

C. self-assessments and expert assessments for at least a couple of people in such a way that it is possible to calculate averages by competence and competence group.

```
ALTER TABLE arvio AUTO_INCREMENT=111;
```

```

Insert into Arvio (HenkiloID, OsaaminenID, Arviotietamys,
Arviokokemus, ArvioijaID) values
(1003, 21, 6, 8, 1003), (1003, 21, 10, 5, 1001), (1003, 21, 6,
7,1002), (1003, 21, 8, 8, 1004),
(1003, 22, 8, 10, 1003), (1003, 22, 6, 3, 1001), (1003, 22, 9,
7,1002), (1003, 22, 10, 8, 1004),
(1003, 23, 6, 9, 1003), (1003, 23, 8, 7, 1001), (1003, 23, 6,
2,1002), (1003, 23, 4, 4, 1004),
(1003, 24, 3, 8, 1003), (1003, 24, 6, 10, 1001), (1003, 24, 5,
5,1002), (1003, 24, 6, 6, 1004),
(1003, 25, 7, 9, 1003), (1003, 25, 3, 7, 1001), (1003, 25, 6,
7,1002), (1003, 25, 4, 4, 1004),
(1003, 26, 7, 9, 1003), (1003, 26, 5, 7, 1001), (1003, 26, 7,
7,1002), (1003, 26, 7, 7, 1004),

(1004, 21, 6, 10, 1004), (1004, 21, 6, 6, 1001), (1004, 21, 6,
3,1002), (1004, 21, 8, 9, 1003),
(1004, 22, 8, 9, 1004), (1004, 22, 8, 10, 1001), (1004, 22, 6,
2,1002), (1004, 22, 10, 10, 1003),
(1004, 23, 4, 7, 1004), (1004, 23, 6, 10, 1001), (1004, 23, 2,
7,1002), (1004, 23, 5, 5, 1003),
(1004, 24, 7, 3, 1004), (1004, 24, 8, 4, 1001), (1004, 24, 7,
8,1002), (1004, 24, 9, 6, 1003),
(1004, 25, 6, 10, 1004), (1004, 25, 5, 9, 1001), (1004, 25, 6,
6,1002), (1004, 25, 10, 2, 1003),
(1004, 26, 6, 10, 1004), (1004, 26, 3, 4, 1001), (1004, 26, 9,
7,1002), (1004, 26, 10, 7, 1003),

(1002, 21, 8, 2, 1002), (1002, 21, 8, 8, 1003), (1002, 21, 6, 5,
1004), (1002, 21, 2, 8, 1001),
(1002, 22, 10, 2, 1002), (1002, 22, 6, 2, 1003), (1002, 22, 8, 2,
1004), (1002, 22, 10, 8, 1001),
(1002, 23, 9, 4, 1002), (1002, 23, 2, 8, 1003), (1002, 23, 8, 8,
1004), (1002, 23, 3, 4, 1001),
(1002, 24, 4, 6, 1002), (1002, 24, 5, 2, 1003), (1002, 24, 2, 2,
1004), (1002, 24, 5, 6, 1001),
(1002, 25, 4, 4, 1002), (1002, 25, 8, 5, 1003), (1002, 25, 6, 6,
1004), (1002, 25, 4, 2, 1001),

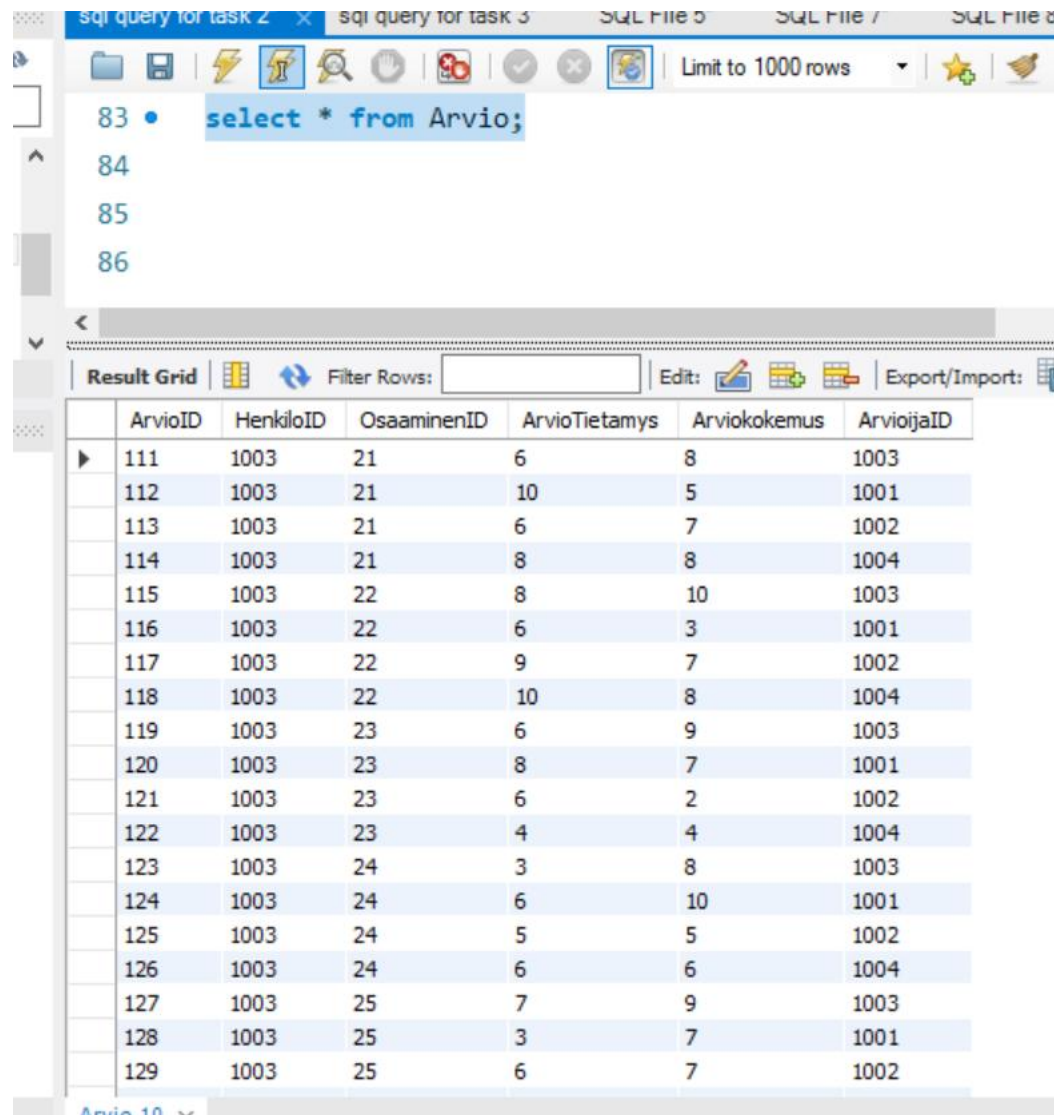
```

```
(1002, 26, 7, 3, 1002), (1002, 26, 9, 9, 1003), (1002, 26, 7, 7, 1004),
(1002, 26, 7, 3, 1001),
```

```
(1001, 21, 4, 7, 1001), (1001, 21, 8, 8, 1002), (1001, 21, 6, 6,
1003), (1001, 21, 8, 8, 1004),
(1001, 22, 5, 5, 1001), (1001, 22, 10, 10, 1002), (1001, 22, 2, 2,
1003), (1001, 22, 7, 7, 1004),
(1001, 23, 9, 8, 1001), (1001, 23, 5, 4, 1002), (1001, 23, 8, 8,
1003), (1001, 23, 6, 2, 1004),
(1001, 24, 10, 5, 1001), (1001, 24, 4, 6, 1002), (1001, 24, 8, 2,
1003), (1001, 24, 8, 2, 1004),
(1001, 25, 6, 8, 1001), (1001, 25, 10, 2, 1002), (1001, 25, 4, 4,
1003), (1001, 25, 9, 9, 1004),
(1001, 26, 8, 5, 1001), (1001, 26, 10, 3, 1002), (1001, 26, 9, 2,
1003), (1001, 26, 5, 2, 1004)
```

```
;
```

```
select * from Arvio;
```



The screenshot shows a SQL IDE interface. At the top, there are tabs for 'sql query for task 2', 'sql query for task 3', 'SQL File 5', 'SQL File 7', and 'SQL File 8'. Below the tabs is a toolbar with various icons, including a 'Limit to 1000 rows' dropdown. The main editor area displays the SQL query 'select * from Arvio;'. Below the editor, there is a 'Result Grid' section. The 'Result Grid' has a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Filter Rows'. The grid contains 19 rows of data with the following columns: ArvioID, HenkiloID, OsaaminenID, ArvioTietamys, Arviokokemus, and ArvioijaID.

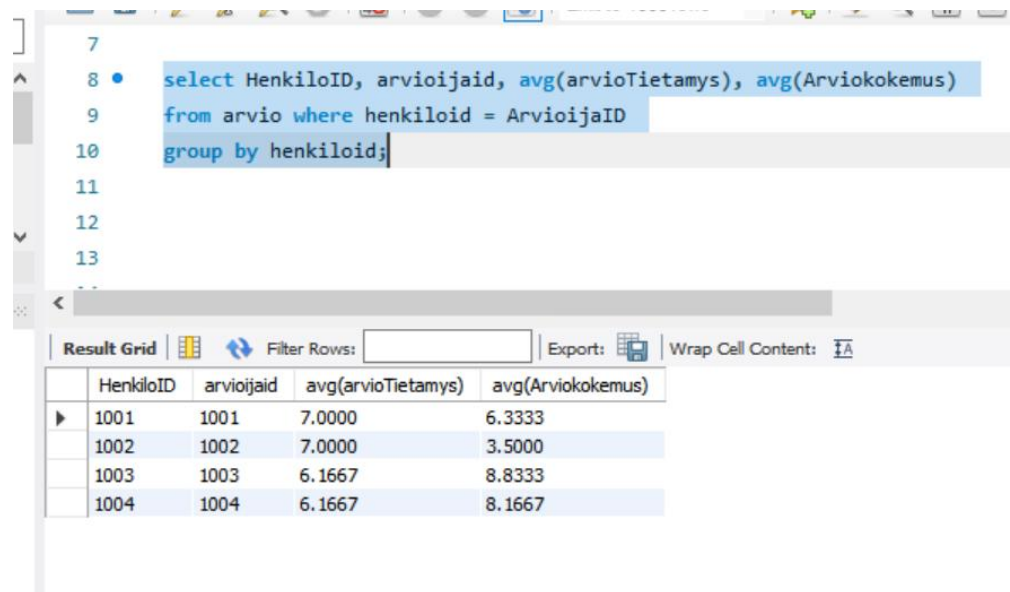
	ArvioID	HenkiloID	OsaaminenID	ArvioTietamys	Arviokokemus	ArvioijaID
▶	111	1003	21	6	8	1003
	112	1003	21	10	5	1001
	113	1003	21	6	7	1002
	114	1003	21	8	8	1004
	115	1003	22	8	10	1003
	116	1003	22	6	3	1001
	117	1003	22	9	7	1002
	118	1003	22	10	8	1004
	119	1003	23	6	9	1003
	120	1003	23	8	7	1001
	121	1003	23	6	2	1002
	122	1003	23	4	4	1004
	123	1003	24	3	8	1003
	124	1003	24	6	10	1001
	125	1003	24	5	5	1002
	126	1003	24	6	6	1004
	127	1003	25	7	9	1003
	128	1003	25	3	7	1001
	129	1003	25	6	7	1002

Task 3 [5p/5p]

Search the information of the tasks with SQL queries from the TOR database with the program of your choice. Save your executed SQL queries in text format.

A. Find the averages of all the grades each person gives themselves for both knowledge and experience. Results by person.

```
select HenkiloID, arvioijaid, avg(arvioTietamys), avg(Arviokokemus)
from arvio where henkiloid = ArvioijaID
group by henkiloid;
```



The screenshot shows a SQL query editor with the following query highlighted in blue:

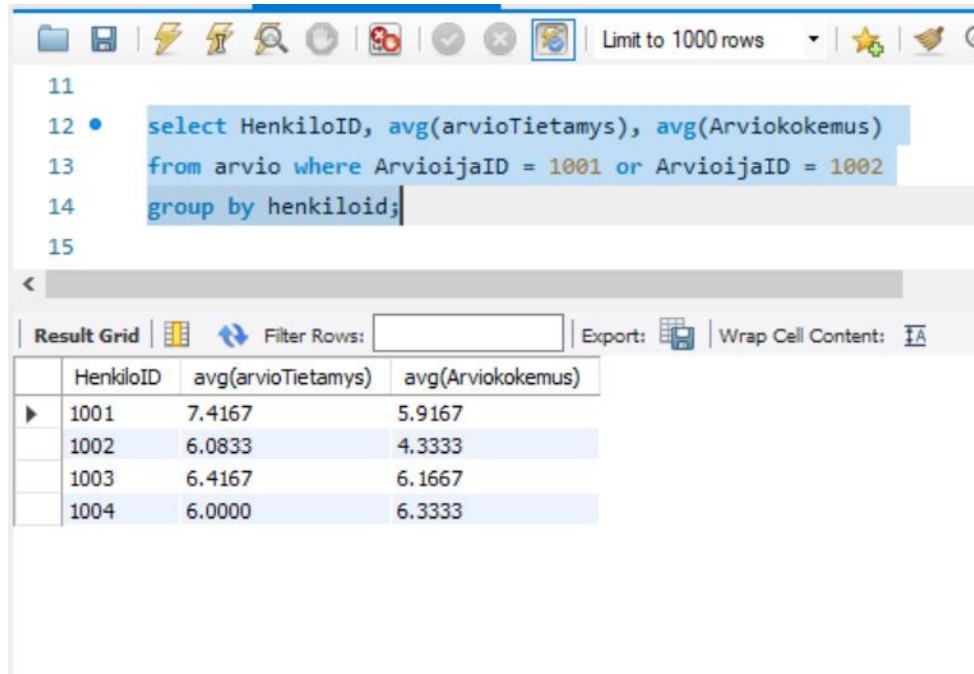
```
select HenkiloID, arvioijaid, avg(arvioTietamys), avg(Arviokokemus)
from arvio where henkiloid = ArvioijaID
group by henkiloid;
```

Below the query editor is a result grid with the following data:

	HenkiloID	arvioijaid	avg(arvioTietamys)	avg(Arviokokemus)
▶	1001	1001	7.0000	6.3333
	1002	1002	7.0000	3.5000
	1003	1003	6.1667	8.8333
	1004	1004	6.1667	8.1667

B. Get the average scores of your chosen experts for both knowledge and experience. Results by person.

```
select HenkiloID, avg(arvioTietamys), avg(Arviokokemus)
from arvio where ArvioijaID = 1001 or ArvioijaID = 1002
group by henkiloid;
```

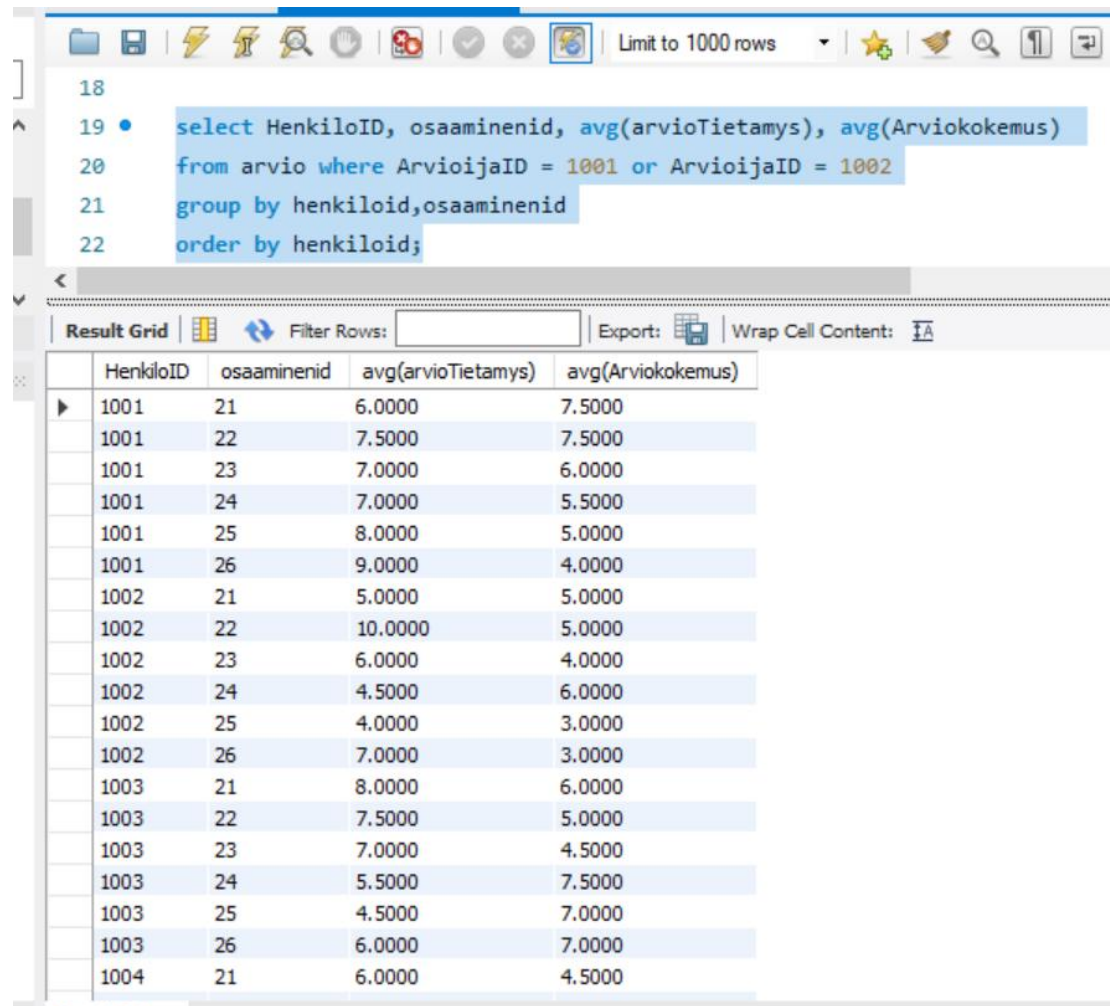


The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, editing, and execution. Below the toolbar, the SQL query is entered in a text area. The query is: `select HenkiloID, avg(arvioTietamys), avg(Arviokokemus) from arvio where ArvioijaID = 1001 or ArvioijaID = 1002 group by henkiloid;`. Below the query editor, there is a section for the results. It includes a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with four columns: 'HenkiloID', 'avg(arvioTietamys)', and 'avg(Arviokokemus)'. The table contains four rows of data.

HenkiloID	avg(arvioTietamys)	avg(Arviokokemus)
1001	7.4167	5.9167
1002	6.0833	4.3333
1003	6.4167	6.1667
1004	6.0000	6.3333

C. Get the average scores of your chosen experts for both knowledge and experience. Results by person and by competence.

```
select HenkiloID, osaaminenid, avg(arvioTietamys), avg(Arviokokemus)
from arvio where ArvioijaID = 1001 or ArvioijaID = 1002
group by henkiloid,osaaminenid
order by henkiloid;
```



The screenshot shows a database query editor with a toolbar at the top. The SQL query is entered in the main text area. Below the query, the 'Result Grid' is displayed, showing the results of the query. The grid has five columns: HenkiloID, osaaminenid, avg(arvioTietamys), and avg(Arviokokemus). The results are grouped by HenkiloID and osaaminenid.

HenkiloID	osaaminenid	avg(arvioTietamys)	avg(Arviokokemus)
1001	21	6.0000	7.5000
1001	22	7.5000	7.5000
1001	23	7.0000	6.0000
1001	24	7.0000	5.5000
1001	25	8.0000	5.0000
1001	26	9.0000	4.0000
1002	21	5.0000	5.0000
1002	22	10.0000	5.0000
1002	23	6.0000	4.0000
1002	24	4.5000	6.0000
1002	25	4.0000	3.0000
1002	26	7.0000	3.0000
1003	21	8.0000	6.0000
1003	22	7.5000	5.0000
1003	23	7.0000	4.5000
1003	24	5.5000	7.5000
1003	25	4.5000	7.0000
1003	26	6.0000	7.0000
1004	21	6.0000	4.5000

D. Find the averages of each self's grades for both knowledge and experience. Results by person and skill group.

```
select a.HenkiloID, a.arvioijaID, o.osaamisryhmaID,
avg(a.arvioTietamys), avg(a.Arviokokemus)
from arvio a inner join osaaminen o on a.osaaminenid = o.osaaminenid
where a.henkiloid = a.ArvioijaID
group by henkiloid, osaamisryhmaID
order by henkiloid, osaamisryhmaID;
```

Limit to 1000 rows

```

30
31
32 • select a.Henkiloid, a.arvioijaid, o.osaamishymaid, avg(a.arvioTietamys), avg(a.Arviokokemus)
33 from arvio a inner join osaaminen o on a.osaaminenid = o.osaaminenid
34 where a.henkiloid = a.ArvioijaID
35 group by henkiloid, osaamishymaid
36 order by henkiloid, osaamishymaid;
37

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Henkiloid	arvioijaid	osaamishymaid	avg(a.arvioTietamys)	avg(a.Arviokokemus)
▶	1001	1001	51	4.5000	6.0000
	1001	1001	52	9.5000	6.5000
	1001	1001	53	7.0000	6.5000
	1002	1002	51	9.0000	2.0000
	1002	1002	52	6.5000	5.0000
	1002	1002	53	5.5000	3.5000
	1003	1003	51	7.0000	9.0000
	1003	1003	52	4.5000	8.5000
	1003	1003	53	7.0000	9.0000
	1004	1004	51	7.0000	9.5000
	1004	1004	52	5.5000	5.0000
	1004	1004	53	6.0000	10.0000

E. Get the averages of ALL experts' ratings for both knowledge and experience. Results by person. (Tip: A subquery could be the easiest)

```

select Henkiloid, avg(arvioTietamys), avg(Arviokokemus)
from arvio where arvioijaid in
(select henkiloid from henkilo where HenkiloTyyppiID =
(select henkilotyyppiid from henkilotyyppi where HenkiloTyyppiName =
'Erityisasiantuntija'))
group by henkiloid;

```

```

32 • select Henkiloid, avg(arvioTietamys), avg(Arviokokemus)
33 from arvio where arvioijaid in
34 (select henkiloid from henkilo where HenkiloTyyppiID =
35 (select henkilotyyppiid from henkilotyyppi where HenkiloTyyppiName = 'Erityisasiantuntija'))
36 group by henkiloid;
37

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Henkiloid	avg(arvioTietamys)	avg(Arviokokemus)
▶	1001	7.4167	5.9167
	1002	6.0833	4.3333
	1003	6.3333	6.5000
	1004	6.0000	7.1667