**Completion Progress** 









O T2\_Q4

O T2\_Q5 O T2 Q6

O T2 Q7

O T2 O8 O T2\_Q9

O T2\_Q10 O T2 Q11

O T2 Q12

Dashboard / My courses / DATA.DB.100-2024-2025-1-TAU / 2. Useita tauluja - Multiple tables

## DATA.DB.100-2024-2025-1 Tietokantojen perusteet (Syksyn toteutus

Participants



## 2. Useita tauluja - Multiple tables

## Oppimateriaalit - Study materials

Theme 2. Study materials in English



## Teeman 2 harjoitustehtävät T2 - Theme 2 exercise tasks T2

- Suositeltu takaraja: 9.9.2024 klo 23:59 / Lopullinen takaraja: 16.9.2024 klo 23:59
- Recommended deadline: 9.9.2024 at 23:59 / Final deadline: 16.9.2024 at 23:59
- T2-harjoitustehtävien suositeltu tehtäväpisteiden minimimäärä: 8 pistettä
- . The recommended minimum number of exercise points for T2 exercise tasks: 8 points

Tehtävissä T2\_Q1-T2\_Q11 tarvitaan kuvitteellisen ja yksinkertaistetun perunatietokannan tauluja

- Taulujen luontilauseet ja tietojen lisäyslauseet on annettu tiedostossa potato.sql. Kannattaa luoda potato.db-tietokanta ja taulut SQLitella ja testata kyselyjä
- Tietokannan tila on esitetty tiedostossa potato database state.pdf.
- Tehtävät T2\_Q1 ja T2\_Q11 ovat monivalintakysymyksiä.
- Tehtävässä T2 Q2 piirretään perunatietokannan kaavion graafinen esitys, SQL-tietokannan kaavion graafinen esitys on selitetty Viiteavain, viite-eheyskalvopaketissa. Katso myös piirto-ohje sivulta Kurssin työkalujen ohjeita
- Tehtävät T2\_Q3-T2\_Q10 ovat kyselytehtäviä. Kiinnitä kyselyissä huomiota seuraaviin asioihin:
  - liitosehto
  - o liitosoperaatio ja liitosehto
  - o taulun nimen käyttö sarakenimen tarkenteena niin SELECT-, WHERE- kuin ORDER BY -osassa, jos kahdessa tai useammassa kyselyn FROM-osasssa esiintyvässä taulussa on samanniminen sarake
  - Huom. CodeGrade käyttää tällä hetkellä SQLiten vanhempaa versiota, jossa on toteutettu vain LEFT OUTER JOIN, ei RIGHT and FULL OUTER JOIN. Käytä siis LEFT OUTER JOINia niissä tehtävissä, joissa tarvitaan ulkoliitosta. Selvitämme, voimmeko saada myöhemmin kurssilla CodeGrad käyttöön SQLiten myöhemmän version, jossa on toteutettu myös muut ulkoliitokset. (4.9.)
- Tehtävässä T2\_Q12 kirjoitetaan taulujen luonti- ja tietojen lisäyslauseita

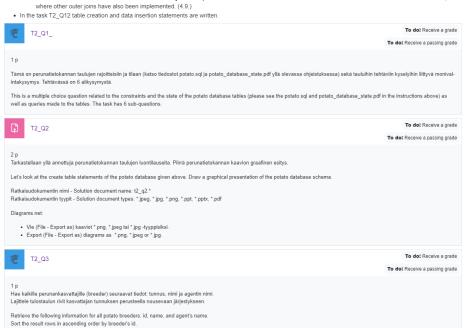
In the tasks T2 Q1-T2 Q11 you need tables of the fictitious and simplified potato database

- Create table statements and data insertion statements are given in the file potato.sql. It is good to create the potato.db database and the tables by SQLite and
- . The state of the database is presented in the file potato\_database\_state.pdf.
- . Tasks T2 Q1 ia T2 Q11 are multiple choice questions.
- In the task T2\_Q2 a graphical representation of the potato database schema is drawn. The graphical representation of an SQL database schema is explained in the slide set Foreign key, referential integrity slide set. See also drawing instructions on the page Instructions for course tools
- Tasks T2\_Q3-T2\_Q10 are query tasks. In the queries, pay attention to the following:
  - ioin condition

breeder id | breeder name | agent name Ratkaisudokumentti - Solution document: t2\_q3.sql

joissa pitää kirjoittaa SQL-lauseita.

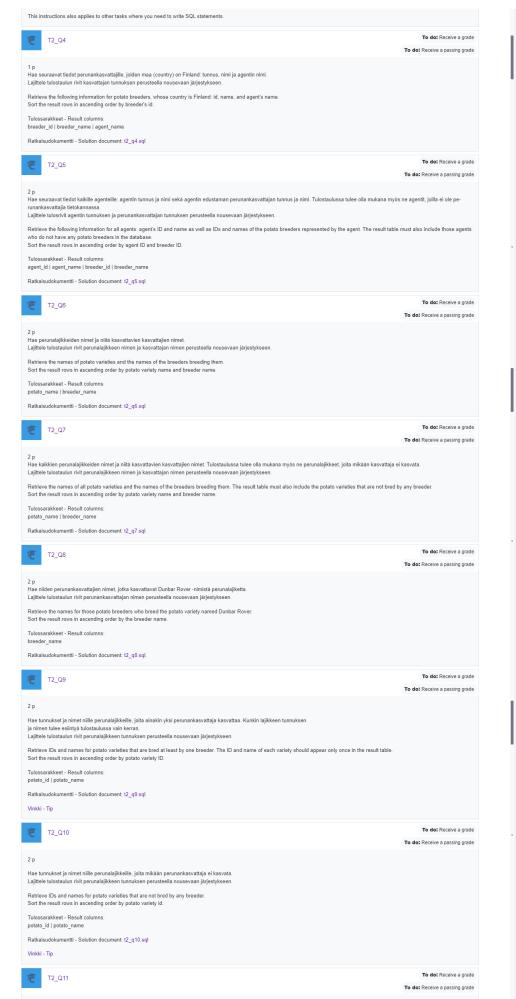
- · join operation and join condition
- o using table names as qualifier for column name in SELECT, WHERE and ORDER BY clauses, if more than one table given in the FROM clause have a column with the same name.
- N.B. CodeGrade currently uses an older version of SQLite with only LEFT OUTER JOIN implemented, not RIGHT and FULL OUTER JOIN. Therefore, use LEFT OUTER JOIN in those tasks where an outer join is required. We will find out if we can use a later version of SQL in CodeGrade later in the course,



isudokumentissasi vain tekemäsi SQL-lauseen eli SELECT -lauseen, ei SQLiten .open -komentoa, joka avaa tietokannan. Tämä ohje pätee myös muihin tehtäviin,

N.B. In your solution document, please submit only the SQL statement you made, i.e. the SELECT statement, not the SQLite .open command that opens the database.





Tämä on INSERT INTO-, UPDATE- ja DELETE-lauseisiin ja perunatietokannan taulujen rajoitteisiin ja tilaan liittyvä monivalintakysymys, jossa on 9 alikysymystä.

This is a multiple choice question related to the INSERT INTO, UPDATE and DELETE statements and the constraints and state of the tables of the potato database, with 9 sub-questions.



To do: Receive a grade

To do: Receive a passing grade

2 p Edellisen viikon harjoituksissa käytettiin bike-nimistä taulua.

Avaa ensin SQLitessa tietokanta, joka sisältää bike-taulun, tai luo bike-taulu ja lisää siihen tietoja

Laita seuraavaksi SQLitessa viite-eheyden valvonta päälle antamalla komento PRAGMA foreign\_keys = ON;

Tee luontilause bike\_shop-nimiselle taululle, johon voidaan tallentaa tietoja pyöräkaupoista. Taulussa on seuraavat sarakkeet

- Taulun pääavain on id-sarake, jossa ei sallita puuttuvia arvoja (NULL-arvoja)

name-sarake on yksilövi siihtuvamittainen merkkijono ja siinä ei sallita puuttuvia arvoja.
 address on vaihtuvamittainen merkkijono Da siinä ei sallita puuttuvia arvoja.
 address on vaihtuvamittainen merkkijono
Luo laulu SQLitella. Lisää tauluun muutaman pyöräkaupan tiedot; keksi pyöräkauppojen tunnus-, nimi- ja osoiteliedot itse.

Tee luontilause sells-taululle, joka yhdistää pyörät ja pyöräkaupat toisiinsa: yhdessä pyöräkaupassa voidaan myydä useaa eri pyörää ja samaa pyörää voidaan myydä useassa eri pyöräkaupassa. Mäarittele sells-tauluun viiteavainsarakkeiden nimiin niiden taulujen sarakkeiden nimiin niiden taulujen nimet, joihin viiteavainsarakkeet viittaavat.

Luo taulu SQLitella ja lisää siihen muutama rivi tietoja

Huom. Määrittele pääavaimet, avaimet (yksilöivä, unique) ja viiteavaimet luontilauseen lopussa oppimateriaalien esimerkkien tapaan automaattisen tarkastuksen vuoksi.

Palauta ratkaisudokumentissa t2\_q12.sql bike\_shop- ja sells-taulujen luontilauseet ja tietojen lisäyslauseet.

In the previous week's exercises we used a table called bike

In SQLite, first open the database containing the bike table or create the bike table and add data to it.

Next, in SQLite, turn the referencial integrity control on by entering the command PRAGMA foreign\_keys = ON;

Make a creation statement for a table called bike\_shop, where information about bike shops can be stored. The table has the following columns:

- The primary key of the table is the id column, which does not allow missing values (NULL values).
  The name column is a unique variable-length string and does not allow missing values.
  address is a variable-length character string

Create the table with SQLite. Insert data of a few bike shops to the table; you can make up the id, name and address data of the bike shops.

Make a creation statement for a table sells that connects bikes and bike shops: Several different bikes can be sold in one bike shop and the same bike can be sold in several different bike shops. Define foreign key columns for the sells table that refer to the bike table and bike\_shop table, as well as the primary key. In the foreign key column names, include the names of the tables that the foreign key columns refer to.

Create the table with SQLite and add a few rows of data to it.

N.B. Specify the primary key, keys (unique) and foreign keys at the end of the creation statement, as in the examples in study materials, for automatic evaluations are considered as a constant of the creation statement.

In the solution document t2\_q12.sql, submit the creation statements of the bike\_shop and sells tables and the data insertion statements

You are logged in as Shamsur Raza Chowdhury (Log out) Policies Dashboard