



OPPIMISTEHTÄVÄ

Marika Verger (3003)

Tomi Salminen (3003)

Hanna Rantanen (3004)

Jenna Räty (3004)

DATAN VARASTOINTITEKNOLOGIAT
Huhtikuu 2022

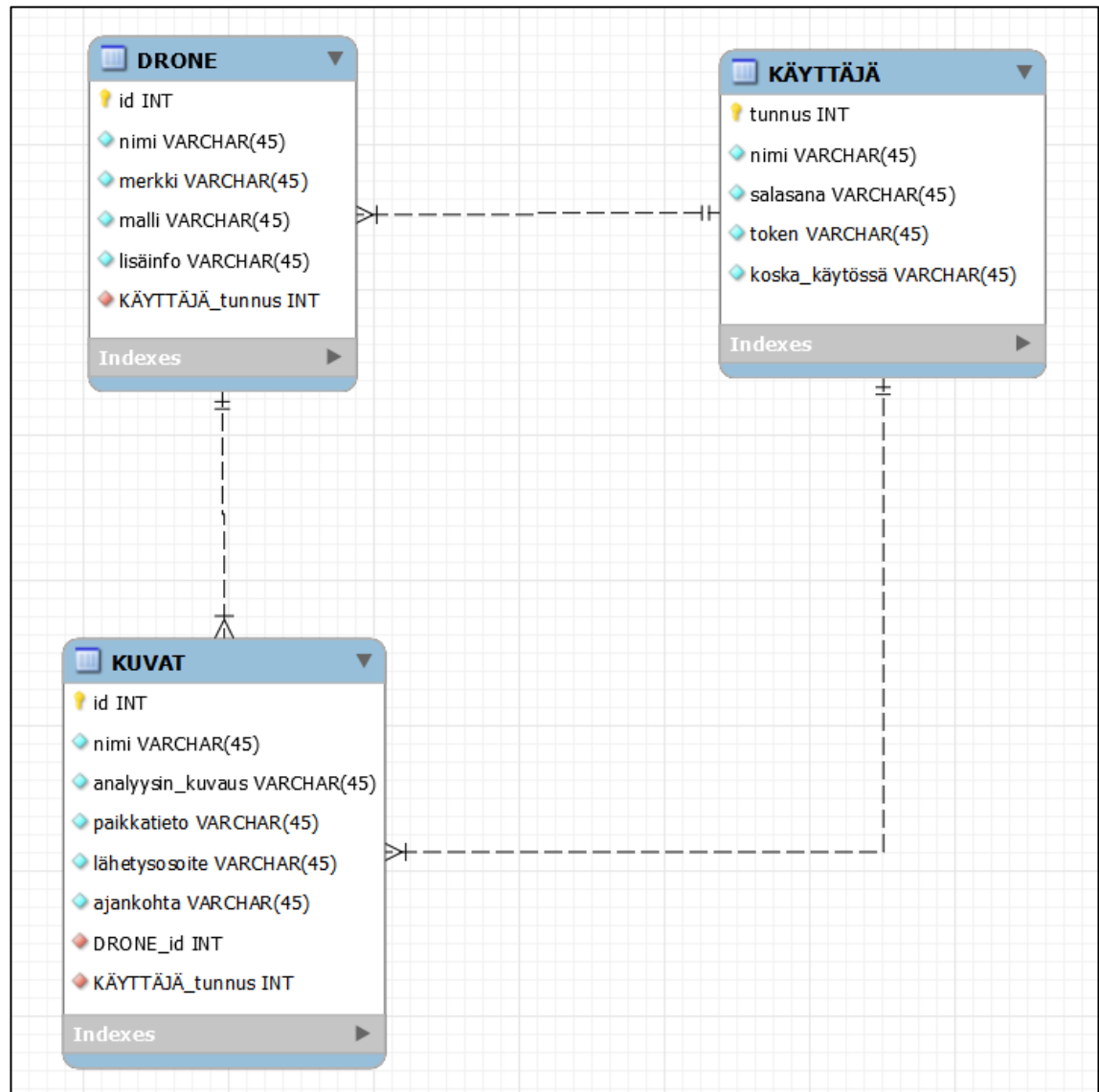
Tietotekniikka

SISÄLLYS

1	ER-KAAVIO JA TIETOKANTASUUNNITTELU	3
2	TIETOKANNAN TOTEUTUS	5
3	RAPORTTIEN TOTEUTUS.....	12

1 ER-KAAVIO JA TIETOKANTASUUNNITTELU

Aloitimme oppimistehtävän luomalla ER-kaavion MySQL Workbenchillä. Tässä pohjana hyödynsimme aiemmin luomiamme Chen-tyylisiä ER-kaavioita. Luomamme ER-kaavio on esitelty seuraavassa kuvassa.



Seuraavaksi loimme komentorivillä kansion oppimistehtävän sekä sen alle tiedostot 'db' ja 'docker-compose.yml'.

```
hannarantanen@LAPTOP-TD6RDIB6:~/DV22/db/harkkatyo$ ls
db  docker-compose.yml
```

db-tiedoston sisältö on esitelty tässä. Muokkasimme tietokannan nimeksi, käyttäjänimeksi ja salasanaksi 'drone'.

```
FROM library/mysql:8
ENV MYSQL_ROOT_PASSWORD=secret
ENV MYSQL_DATABASE=drone
ENV MYSQL_USER=drone
ENV MYSQL_PASSWORD=drone
```

docker-compose.yml -tiedosto on esitelty tässä.

```
version: '2.4'
services:
  db:
    build:
      context: ./
      dockerfile: ./db
    ports:
      - '3306:3306'
    volumes:
      - /var/lib/mysql
    restart: always

  adminer:
    image: adminer
    restart: always
    ports:
      - 8080:8080
```

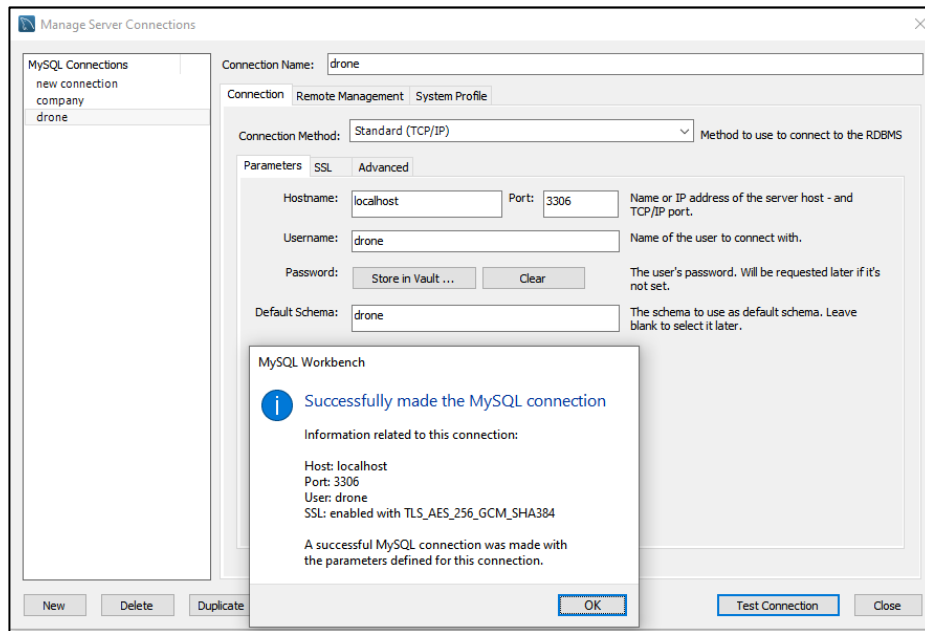
Seuraavaksi ajoimme kontit ylös 'docker-compose up -d' -komennolla:

```
hannarantanen@LAPTOP-TD6RDI86:~/DV22/db/harkkatyo$ docker-compose up -d
[+] Building 10.6s (5/5) FINISHED
=> [internal] load build definition from db                                0.1s
=> => transferring dockerfile: 151B                                       0.0s
=> [internal] load .dockerignore                                           0.0s
=> => transferring context: 2B                                             0.0s
=> [internal] load metadata for docker.io/library/mysql:8                 10.5s
=> CACHED [1/1] FROM docker.io/library/mysql:8@sha256:fc77d54cacef90ad3d75964837fad0f2a9a368b69e7d799665a3f4e90e 0.0s
=> exporting to image                                                     0.0s
=> => exporting layers                                                    0.0s
=> => writing image sha256:253e6c20884857991f8f5b4ae20be47a65a24f65b0a2b0435208d0f2de1d1ead 0.0s
=> => naming to docker.io/library/harkkatyo_db                           0.0s

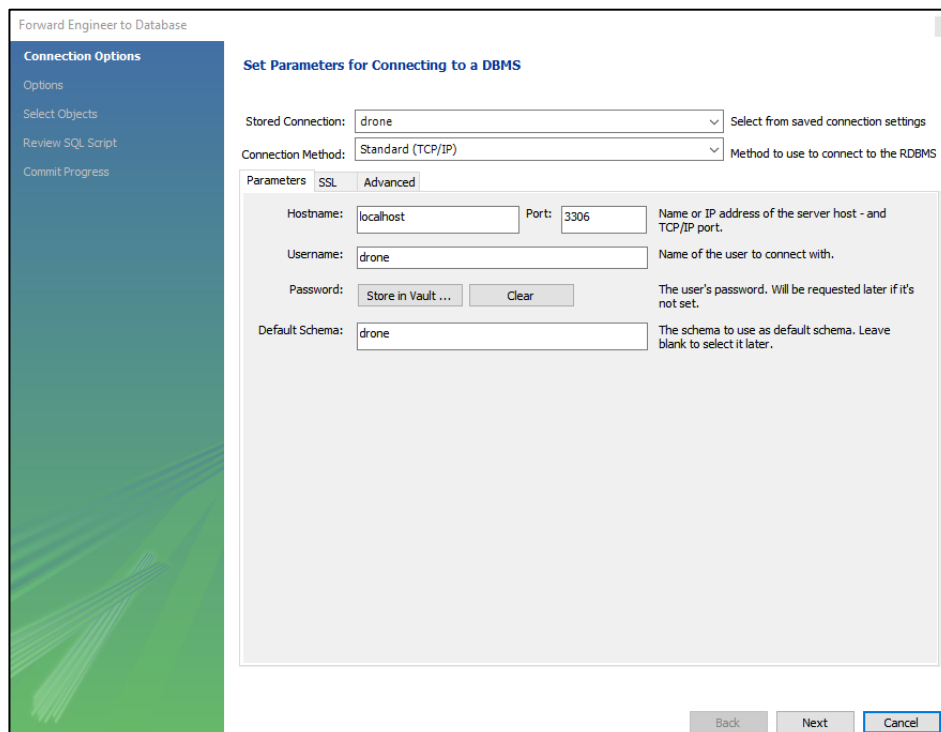
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 3/3
# Network harkkatyo_default        Created                                0.0s
# Container harkkatyo-db-1         Started                               1.1s
# Container harkkatyo-adminer-1    Started                               1.1s
```

2 TIETOKANNAN TOTEUTUS

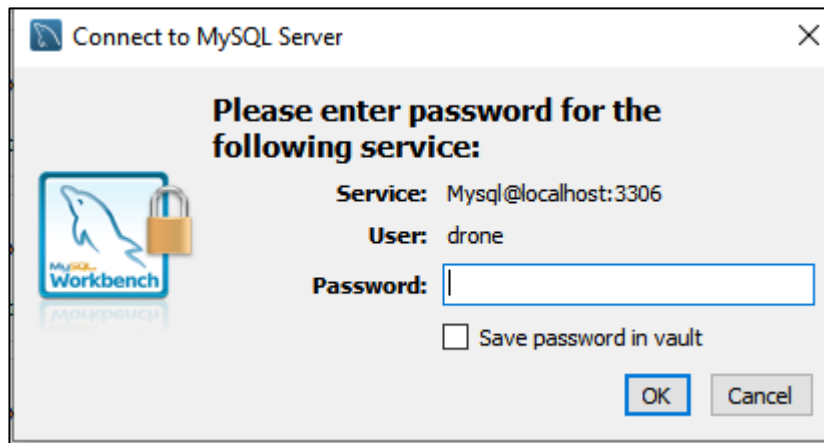
Aluksi loimme yhteyden tietokantaan. Määrittelimme yhteyden nimeksi, käyttäjänimeksi ja oletus-schemaksi asetettiin 'drone'. Hostname on 'localhost' ja portti 3306. 'Test connection' -napilla voitiin todeta, että yhteys on toimiva.



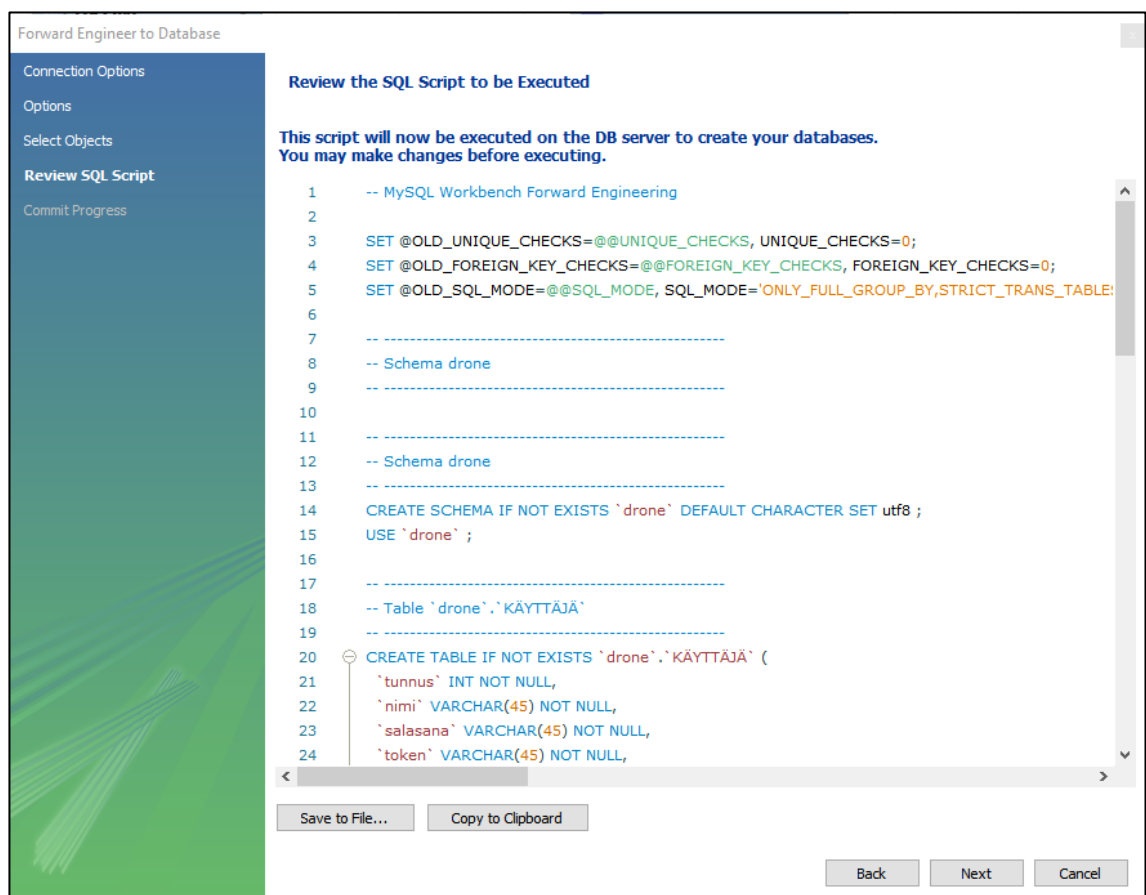
Aloitimme forward engineering -prosessin määrittelemällä parametrit yhteydelle. Yhteyden nimeksi, käyttäjänimeksi sekä oletus-schemaksi asetettiin 'drone'. Hostname on 'localhost' ja portti 3306.



Sitten syötettiin itse määrittelemämme salasana 'drone'.



Seuraavaksi voitiin esikatsella skriptiä, jolla tietokanta luodaan. Tallensimme skriptin 'save to file' -napilla myöhempää käyttöä varten.



Seuraavassa vaiheessa forward engineering suoritettiin onnistuneesti, ja selaimen osoitteessa *localhost:8080* voitiin kirjautua Admineriin aiemmassa vaiheessa määrittelemillämme sisäänkirjautumistiedoilla.

Language: English ▼

Adminer 4.8.1

Login

System	MySQL ▼
Server	db ⓘ
Username	drone
Password	••••• ⓘ
Database	drone

☐ Permanent login

Adminerissa oli nähtävillä luomamme tietokanta *drone*.

Language: English ▼ MySQL » db » Database: drone

Adminer 4.8.1

Database: drone

DB: drone ▼

[SQL command](#) [Import](#)
[Export](#) [Create table](#)

select DRONE
select KUVAT
select KÄYTTÄJÄ

[Alter database](#) [Database schema](#) [Privileges](#)

Tables and views

Search data in tables (3)

<input type="checkbox"/>	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
<input type="checkbox"/>	DRONE	InnoDB	utf8mb4_0900_ai_ci	16,384	16,384	0		0	
<input type="checkbox"/>	KUVAT	InnoDB	utf8mb4_0900_ai_ci	16,384	32,768	0		0	
<input type="checkbox"/>	KÄYTTÄJÄ	InnoDB	utf8mb4_0900_ai_ci	16,384	0	0		0	
	3 in total	InnoDB	utf8mb4_0900_ai_ci	49,152	49,152	0			

Selected (0)

Move to other database: drone ▼ ☐ overwrite

[Create table](#) [Create view](#)

Routines

[Create procedure](#) [Create function](#)

Events

[Create event](#)

Seuraavana on esitelty kommentoitu SQL-skripti, jonka tallensimme forward engineering -vaiheessa.

-- MySQL Workbench Forward Engineering

-- Alustetaan tietokanta

```
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 drone
-- -----
```

```
-- -----
-- Schema drone
-- -----
```

-- Mikäli schemaa ei ole, se luodaan ja nimeksi asetetaan 'drone'

```
CREATE SCHEMA IF NOT EXISTS `drone` DEFAULT CHARACTER SET utf8 ;
USE `drone` ;
```

-- Luodaan taulu 'KÄYTTÄJÄ'

```
-- -----
-- Table `drone`.`KÄYTTÄJÄ`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `drone`.`KÄYTTÄJÄ` (
```

-- Seuraavana on luomamme avaimet ja kohdetypit

```
`tunnus` INT NOT NULL,
`nimi` VARCHAR(45) NOT NULL,
`salasana` VARCHAR(45) NOT NULL,
`token` VARCHAR(45) NOT NULL,
`koska_käytössä` VARCHAR(45) NOT NULL,
PRIMARY KEY (`tunnus`))
ENGINE = InnoDB;
```


-- Luodaan taulu 'DRONE'

-- Table `drone`.`DRONE`

CREATE TABLE IF NOT EXISTS `drone`.`DRONE` (

-- Seuraavana on luomamme avaimet ja kohdetypit

`id` INT NOT NULL,
 `nimi` VARCHAR(45) NOT NULL,
 `merkki` VARCHAR(45) NOT NULL,
 `malli` VARCHAR(45) NOT NULL,
 `lisäinfo` VARCHAR(45) NOT NULL,
 `KÄYTTÄJÄ_tunnus` INT NOT NULL,
 PRIMARY KEY (`id`),

-- Seuraavana määritellään ulkoinen avain one-to-many -relaation pohjalta sekä asetukset poistamisen tai päivittämisen yhteydessä

INDEX `fk_DRONE_KÄYTTÄJÄ_idx` (`KÄYTTÄJÄ_tunnus` ASC) VISIBLE,
 CONSTRAINT `fk_DRONE_KÄYTTÄJÄ`
 FOREIGN KEY (`KÄYTTÄJÄ_tunnus`)
 REFERENCES `drone`.`KÄYTTÄJÄ` (`tunnus`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Luodaan taulu 'KUVAT'

-- Table `drone`.`KUVAT`

CREATE TABLE IF NOT EXISTS `drone`.`KUVAT` (

-- Seuraavana on luomamme avaimet ja kohdetypit

`id` INT NOT NULL,
 `nimi` VARCHAR(45) NOT NULL,
 `analyysin_kuvaus` VARCHAR(45) NOT NULL,
 `paikkatieto` VARCHAR(45) NOT NULL,
 `lähetysosoite` VARCHAR(45) NOT NULL,
 `ajankohta` VARCHAR(45) NOT NULL,
 `DRONE_id` INT NOT NULL,
 `KÄYTTÄJÄ_tunnus` INT NOT NULL,
 PRIMARY KEY (`id`),

-- Seuraavana määritellään ulkoinen avain one-to-many -relaation pohjalta sekä asetukset poistamisen tai päivittämisen yhteydessä

INDEX `fk_KUVAT_DRONE1_idx` (`DRONE_id` ASC) VISIBLE,
 INDEX `fk_KUVAT_KÄYTTÄJÄ1_idx` (`KÄYTTÄJÄ_tunnus` ASC) VISIBLE,

```

CONSTRAINT `fk_KUVAT_DRONE1`
  FOREIGN KEY (`DRONE_id`)
  REFERENCES `drone`.`DRONE` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_KUVAT_KÄYTTÄJÄ1`
  FOREIGN KEY (`KÄYTTÄJÄ_tunnus`)
  REFERENCES `drone`.`KÄYTTÄJÄ` (`tunnus`)
  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;

```

Seuraavaksi lisäsimme dataa tietokannan tauluihin. Aloitimme lisäämällä KÄYTTÄJÄ-tauluun kaksi käyttäjää.

```

INSERT into KÄYTTÄJÄ VALUES (1, "Matti Maijala", "qwerty", "token1", "käytössä")

Query executed OK, 1 row affected. (0.010 s) Edit

INSERT into KÄYTTÄJÄ VALUES (2, "Maija Meikäläinen", "uiopå", "token2", "ei_käytössä")

Query executed OK, 1 row affected. (0.006 s) Edit

-- Lisätään insert-komennolla KÄYTTÄJÄ-tauluun dataa: tunnus, nimi, salasana, token, koska käytössä
INSERT into KÄYTTÄJÄ VALUES (1, "Matti Maijala", "qwerty", "token1", "käytössä");
INSERT into KÄYTTÄJÄ VALUES (2, "Maija Meikäläinen", "uiopå", "token2", "ei_käytössä");

```

KÄYTTÄJÄ-taulussa tiedot näkyivät seuraavasti:

Select: KÄYTTÄJÄ

Select data

Show structure

Alter table

New item

Select

Search

Sort

Limit 50

Text length 100

Action Select

SELECT * FROM `KÄYTTÄJÄ` LIMIT 50 (0.003 s) Edit

<input type="checkbox"/> Modify	tunnus	nimi	salasana	token	koska_käytössä
<input type="checkbox"/> edit	1	Matti Maijala	qwerty	token1	käytössä
<input type="checkbox"/> edit	2	Maija Meikäläinen	uiopå	token2	ei_käytössä

Lisäsimme DRONE-tauluun kaksi dronea.

```
-- Lisätään insert-komennolla DRONE-tauluun dataa: id, nimi, merkki, malli, lisäinfo, käyttäjätunnus
INSERT into DRONE VALUES (1, "Lennätin", "Apple", "XR", "ei_lisäinfoa", 1);

INSERT into DRONE VALUES (2, "Ilmadrone", "DJI", "Mini-SE", "ei_lisäinfoa", NULL);
```

DRONE-taulussa tiedot näkyivät seuraavasti:

Select: DRONE

Select data

Show structure

Alter table

New item

Select

Search

Sort

Limit50

Text length100

ActionSelect

SELECT * FROM `DRONE` LIMIT 50 (0.001 s) Edit

<input type="checkbox"/> Modify	id	nimi	merkki	malli	lisäinfo	KÄYTTÄJÄ_tunnus
<input type="checkbox"/> edit	1	Lennätin	Apple	XR	ei_lisäinfoa	1
<input type="checkbox"/> edit	2	Ilmadrone	DJI	Mini-SE	ei_lisäinfoa	NULL

Lisäsimme KUVAT-tauluun kolme kuvaa.

```
-- Lisätään insert-komennolla KUVAT-tauluun dataa: id, nimi, analyysin kuvaus, paikkatieto, lähetysosoite,
-- ajankohta, drone-id, käyttäjätunnus
INSERT into KUVAT VALUES (1, "Maisema", "laaduntarkkailu", "Tampere", "kuvat@gmail.com", "01012022", 1, 1);
INSERT into KUVAT VALUES (2, "Muumitalo", "kuvan_tunnelma", "Naantali", "tuutikki@gmail.com", "01042022", 2, 2);
INSERT into KUVAT VALUES (3, "Auto", "kuvan_tarkkuus", "Ysitie", "topgear@gmail.com", "01032022", 2, 1);
```

KUVAT-taulussa tiedot näkyivät seuraavasti:

Select: KUVAT

Select data

Show structure

Alter table

New item

Select

Search

Sort

Limit

50

Text length

100

Action

Select

SELECT * FROM `KUVAT` LIMIT 50 (0.001 s)

Edit

<input type="checkbox"/> Modify	id	nimi	analyysin_kuvaus	paikkatieto	lähetysosoite	ajankohta	DRONE_id	KÄYTTÄJÄ_tunnus
<input type="checkbox"/> edit	1	Maisema	laaduntarkkailu	Tampere	kuvat@gmail.com	01012022	1	1
<input type="checkbox"/> edit	2	Muumitalo	kuvan_tunnelma	Naantali	tuutikki@gmail.com	01042022	2	2
<input type="checkbox"/> edit	3	Auto	kuvan_tarkkuus	Ysitie	topgear@gmail.com	01032022	2	1

3 RAPORTTIEN TOTEUTUS

Raportti 1: kuka käyttää dronea

```
-- Tulostetaan selectillä dronen ja käyttäjän nimi
-- Tauluista valitaan drone ja käyttäjä
-- Dronen käyttäjä saadaan, kun dronen käyttäjätunnus = käyttäjän tunnus
```

```
SELECT DRONE.nimi, KÄYTTÄJÄ.nimi
FROM DRONE, KÄYTTÄJÄ
WHERE DRONE.KÄYTTÄJÄ_tunnus = KÄYTTÄJÄ.tunnus
```

nimi	nimi
Lennätin	Matti Majjala

Raportti 2. onko drone vapaana

Raportin toteutusta varten loimme DRONE-tiluun uuden sarakkeen 'vapaana':

<input type="checkbox"/> Modify	id	nimi	merkki	malli	lisäinfo	vapaana	KÄYTTÄJÄ_tunnus
<input type="checkbox"/> edit	1	Lennätin	Apple	XR	ei_lisäinfoa	ei	1
<input type="checkbox"/> edit	2	Ilmadrone	DJI	Mini-SE	ei_lisäinfoa	kyllä	NULL

Nyt taulusta saadaan tulostettua tieto dronejen vapaanaolosta.

```
-- Tulostetaan selectillä dronen nimi sekä tieto, onko vapaana
-- Tauluista valittu drone
```

```
SELECT nimi, vapaana
FROM DRONE
```

nimi	vapaana
Lennätin	ei
Ilmadrone	kyllä

Raportti 3. Dronen ottamat kuvat

```
-- Tulostetaan selectillä dronen nimi, kuvan nimi ja käyttäjän nimi
-- Tauluista valitaan KUVAT, DRONE, KÄYTTÄJÄ
-- Halutut tiedot saadaan kun, kuvan drone-id = dronen id JA
-- kuvien käyttäjätunnus = käyttäjän tunnus
```

```
SELECT DRONE.nimi AS Drone, KUVAT.nimi AS Kuva, KÄYTTÄJÄ.nimi AS Kuvan_ottaja
FROM KUVAT, DRONE, KÄYTTÄJÄ
WHERE KUVAT.DRONE_id = DRONE.id AND KUVAT.KÄYTTÄJÄ_tunnus = KÄYTTÄJÄ.tunnus
```

Drone	Kuva	Kuvan_ottaja
Lennätin	Maisema	Matti Majjala
Ilmadrone	Muumitalo	Maija Meikäläinen
Ilmadrone	Auto	Matti Majjala

Raportti 4. Kuvan ottanut drone

```
-- Tulostetaan selectillä kuvan nimi, dronen nimi, käyttäjän nimi, analyysin kuvaus, paikkatieto, ajankohta
-- Tauluiksi valitaan KUVAT, DRONE, KÄYTTÄJÄ
-- Halutut tiedot saadaan, kun kuvan drone-id = dronen id JA
-- kuvan käyttäjätunnus = käyttäjän tunnus

SELECT KUVAT.nimi AS Kuva, DRONE.nimi AS Drone, KÄYTTÄJÄ.nimi AS Kuvan_ottaja, KUVAT.analyysin_kuvaus, KUVAT.paikkatieto, KUVAT.ajankohta
FROM KUVAT, DRONE, KÄYTTÄJÄ
WHERE KUVAT.DRONE_id = DRONE.id AND KUVAT.KÄYTTÄJÄ_tunnus = KÄYTTÄJÄ.tunnus
```

Kuva	Drone	Kuvan_ottaja	analyysin_kuvaus	paikkatieto	ajankohta
Maisema	Lennätin	Matti Maijala	laaduntarkkailu	Tampere	01012022
Muumitalo	Ilmadrone	Maija Meikäläinen	kuvan_tunnelma	Naantali	01042022
Auto	Ilmadrone	Matti Maijala	kuvan_tarkkuus	Ysitie	01032022