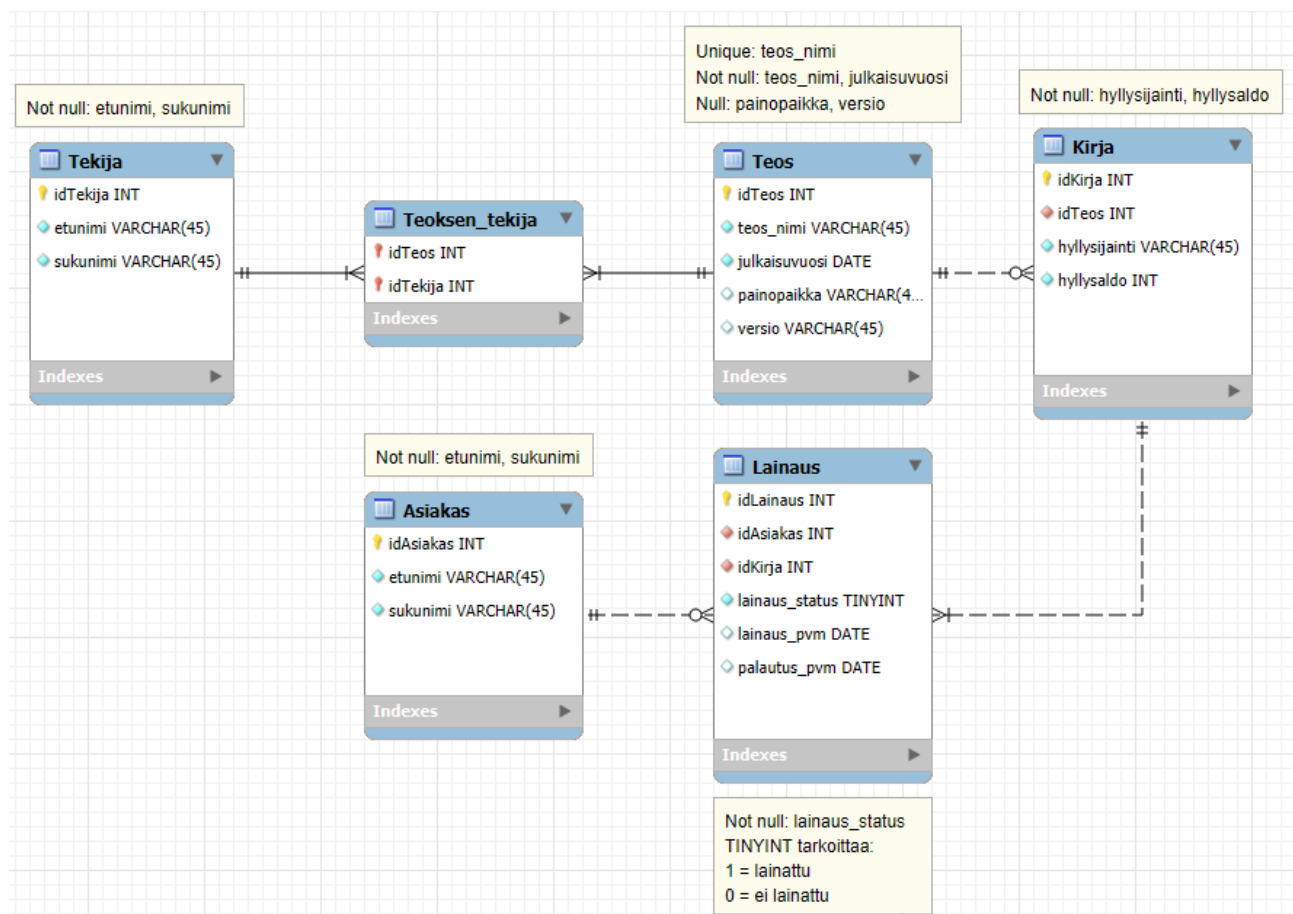


Viikon 4 tehtävänä oli mallintaa kuvitteellisen kirjaston lainausjärjestelmän tietokantaa MySQL Workbench-sovelluksella. Tehtävänannossa laaditut vaatimukset tietokannalle alla:

- Tekijällä voi olla kirjastossa yksi tai useampia teoksia.
- Teoksella voi olla yksi tai useampi tekijä.
- Yksittäinen kirja voi olla joko lainassa jollain asiakkaalla tai hyllyssä.
- Teos ei voi olla olemassa ilman että sillä on joku tekijä.
- Toisaalta teos voi olla olemassa, vaikka kirjastossa ei olisi yhtään kyseistä kirjaa.
- Asiakkaan tiedot voivat olla tietokannassa, vaikka hän ei ole lainannut yhtään kirjaa.
- Kirja voi olla kirjastossa hyllyssä, vaikka sitä ei olisi lainattu yhtään kertaa.

Tehtävää 11 varten toteutettu ER-diagrammi:



**Pohdintaa:** Tekijä- ja teostaulujen välillä on monesta moneen yhteys, sillä yhdellä tekijällä voi olla yksi tai useampi teos. Sama pätee myös toisin päin: Teoksella voi olla yksi tai useampi tekijä. Teos ei kuitenkaan voi olla olemassa ilman tekijää, joka tarkoittaa *identifying* -suhdetta. Teoksen ja Kirjan välillä on *non-identifying* -suhde, koska teos voi olla olemassa, vaikka sitä ei ole kirjaston ”saldoilla”. Teoksesta taas voi olla nolla tai useampi fyysinen painos. Kirja voi olla olemassa ilman yhtäkään lainauksetta ja kirjastolla voi olla yksi tai useampi lainaus samanaikaisesti. Asiakas voi olla myös olemassa ilman lainauksia. Kirjaston asiakkaalla voi olla nolla tai useampi lainaus. Erottelin taulukoiden piirteitä kuvan keltapohjaisiin muistilappuihin. Asetin teostaulun sarakke ”teos\_nimi” sai UNIQUE-datatyypin, jotta välttyttäisiin täysin samoilta kopioilta. Tämä ei välttämättä ole tarpeellista tarkentavien

kenttien, kuten ”painopaikka” ja ”versio” jälkeen. Ajattelisin niin, että yksi syy UNIQUE-arvon käyttöä vastaan sarakkeessa ”teos\_nimi” on saman kirjan versioiden lisäämisen hankaloituminen.

**Uniformin serveriltä tietokannan rakenne näyttää seuraavalta:**

```
mysql> use kirjastodb;
Database changed
mysql> show tables;
+-----+
| Tables_in_kirjastodb |
+-----+
| asiakas              |
| kirja                |
| lainaus              |
| tekija               |
| teoksen_tekija       |
| teos                 |
+-----+
6 rows in set (0.00 sec)
```

**Esimerkkinä kirjan ja tekijöiden lisäys sekä yhdistys samaan teokseen. Lisäksi esimerkkitulostus alimpana:**

```
mysql> INSERT INTO teos VALUES(NULL,'Insinöörin Matematiikka 2','2014-04-01','Porvoo','2.Painos');
Query OK, 1 row affected (0.01 sec)

mysql> select * from teos;
+-----+-----+-----+-----+-----+
| idTeos | teos_nimi                | julkaisuvuosi | painopaikka | versio |
+-----+-----+-----+-----+-----+
| 1      | Insinöörin Matematiikka 2 | 2014-04-01    | Porvoo      | 2.Painos |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> INSERT INTO Teoksen_tekija (idTeos, idTekija) VALUES (1,1), (1,2), (1,3), (1,4), (1,5);
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from teoksen_tekija;
+-----+-----+
| idTeos | idTekija |
+-----+-----+
| 1      | 1        |
| 1      | 2        |
| 1      | 3        |
| 1      | 4        |
| 1      | 5        |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT tekija.etunimi AS etunimi, tekija.sukunimi AS sukunimi, teos.teos_nimi AS teos
-> FROM tekija
-> INNER JOIN teoksen_tekija ON tekija.idTekija = teoksen_tekija.idTekija
-> INNER JOIN teos ON teoksen_tekija.idTeos = teos.idTeos;
+-----+-----+-----+
| etunimi | sukunimi | teos |
+-----+-----+-----+
| Jarkko  | Surakka  | Insinöörin Matematiikka 2 |
| Heikki  | Makkonen | Insinöörin Matematiikka 2 |
| Maija   | Huuskonen | Insinöörin Matematiikka 2 |
| Eero    | Holmlund | Insinöörin Matematiikka 2 |
| Ari     | Tuomenlehto | Insinöörin Matematiikka 2 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```