



Tässä olisi ER diagrammi tuohon 11 tehtävään.

Jonkin aikaa meni suunnitella, mutta nyt tuntuu että olisi ihan hyvä.

Tämän pdf:n lopussa on vielä liitettyä tulostus liittyen taulujen luontiin josta näkyy että mitä rajoituksia milläkin kentällä on.

Tässä jotain selitystä että miksi yhteydet taulujen välillä ovat mitä ovat ja miksi tietyt kentät on jollain rajoituksilla.

Tekija_Teos välitaulu mahdollistaa monesta moneen yhteyden ja tämä onkin ainoa taulu tässä tietokannassa missä on pakotettu yhteys eli molemmat avaimet ovat primary keytä. Tämä oli siitä syystä että jokaisen entryn välitauluun on kuvattava oikeata (tietokantaan) lisättyä riviä. Eli tämän Tekija_Teos taulun lisäyksen koko identiteetti on se että minkä teoksen ja tekijän se yhdistää. Ei siitä syystä että oikeassa maailmassa tekijällä on oltava joku teos vaan juuri tämän taulun luonteen vuoksi. Siinä on siis kaksi primary keytä jotka ovat foreign keytä.

Tekijä taulussa Non nullina on nimet vaikkapa kait sitä voisi olla nimetön tekijä, mutta silloin sen nimi taulussa olisi vaikka tuntematon. Primary key on se idTekija joka on UNSIGNED INT AUTO_INCREMENT. Tämä siitä syystä että unsigned antaa lisää vaihtoehtoja kun nolasta lähdetään ylöspäin.

Teos taulussa ID Primary key, auto increment ja non null. Nimen ja teoksen laitoin not nulliksi, jotta on pakko jotain sinne laittaa. ISBN tässä taulussa on ainoa uniikki avain koko tietokannassa. Tämä siitä syystä jotta ISBN on aina omansa vaikka kirja olisi saman kirjan eri painos.

Asiakas on voi olla vaikka kirjoja ei olisi lainannut ja asiakkaalla voi olla monta kirjaa lainassa joten taulussa vain perustietoja kuten etunimi, sukunimi ja osoite IDn lisäksi. ID on normi tapaan Primary key ja non null auto increment. Ajattelin laittaa ihan huvikseen sukunimen ja etunimen pakollisiksi vaikka voi kait sitä olla asiakas jolla ei ole nimiä. Kait ne olisi voinut jättää että ne voi olla non nulleja ja laittaa vaikka sosiaaliturva tunnus joka sitten ei voisi olla null. Tässä taulussa tämä kuitenkin varmaan riittää. Osoite voi tietysti olla null koska eihän sitä kaikilla ole. Pohdin että olisiko lisännyt asiakkaalle esim lainauskielto booleanin (tinyint) koska siitä olisi helppo katkoa että voiko asiakas lainailla kirjoja, mutta en jaksanut taulua hankaloittaa.

Kirja taulu kuvastaa lainattavissa olevaa teosta joten sillä on oma idKirja normaaliin tapaan. Lisäksi löytyy idTeos joka on non null foreign key. Tämä non null foreign key pakottaa sen että krijaan kirjasta löytyy lisäys siitä mikä teos ja painos jne se on. Luokitus nyt kuvaa sitä hyllypaikkaa joten se voi olla ei mitään jos se nyt sattuu varastossa olemaan. Toki luokitus kuvaa enemmän varmaankin kategoriaa, mutta...

Sitten Lainaus.

Lainauksia voi olla samalla kirjalla monta. Toki nämä ovat oltava eri aikaan, mutta sitä ei voi pakottaa tätä taulua suunnitellessa. Tästä syystä joka lainauksella on oma primary key ID normi tapaan. Se ettei tuplalainausta samalla kirjalla ole, pitää tehdä joko triggerillä tai jollain aliohjelmalla tai jotain. idLainauksen lisäksi on tässä taulussa oltava kirja jota lainataan ja asiakas jolle lainataan. Nämä ovat siis molemmat foreign keytä ja evät voi olla ei mitään. Lisäksi vielä löytyy lainaus aika, eräpäivämäärä ja palautus päivämäärä. Ensimmäiset kaksi syötetään current timellä ja eräpvm lisätään se laina aika lisäksi joten nämä eivät voi olla nulleja. Palautus pvm on tietysti voitava olla nolla kunnes palautus tapahtuu. Palautettu löytyy tinyinttinä koska booleania ei löytynyt workbenchistä lisättävänä vaihtoehtona. Tähän tulee defaulttina 0 kunnes se laina on sitten palautettu.

Siinäpä pohdiskeluja siitä miksi yhteydet ja arvot ovat niikuin ovat. Toki jotain saattoi unohtua.

```
SHOW CREATE TABLE Asiakas \G SHOW CREATE TABLE Kirja \G SHOW CREATE TABLE  
Lainaus \G SHOW CREATE TABLE Tekija \G SHOW CREATE TABLE Tekija_Teos \G SHOW  
CREATE TABLE Teos \G
```

Table: Asiakas

```
Create Table: CREATE TABLE `Asiakas` (  
  `idAsiakas` int unsigned NOT NULL AUTO_INCREMENT,  
  `Etunimi` varchar(45) NOT NULL,  
  `Sukunimi` varchar(45) NOT NULL,  
  `Osoite` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`idAsiakas`)  
)
```

Table: Kirja

```
Create Table: CREATE TABLE `Kirja` (  
  `idKirja` int NOT NULL AUTO_INCREMENT,  
  `idTeos` int unsigned NOT NULL,  
  `Luokitus` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`idKirja`),  
  KEY `fk_Kirja_Teos1_idx` (`idTeos`),  
  CONSTRAINT `fk_Kirja_Teos1` FOREIGN KEY (`idTeos`) REFERENCES `Teos` (`idTeos`)  
)
```

Table: Lainaus

```
Create Table: CREATE TABLE `Lainaus` (  
  `idLainaus` int unsigned NOT NULL AUTO_INCREMENT,  
  `Kirja_idKirja` int NOT NULL,  
  `Asiakas_idAsiakas` int unsigned NOT NULL,  
  `LainaPVM` datetime NOT NULL,  
  `EraPVM` datetime NOT NULL,  
  `PalautusPVM` datetime DEFAULT NULL,  
  `Palautettu` tinyint NOT NULL DEFAULT '0',  
  PRIMARY KEY (`idLainaus`),  
  KEY `fk_Lainaus_Kirja1_idx` (`Kirja_idKirja`),  
  KEY `fk_Lainaus_Asiakas1_idx` (`Asiakas_idAsiakas`),  
  CONSTRAINT `fk_Lainaus_Asiakas1` FOREIGN KEY (`Asiakas_idAsiakas`) REFERENCES  
  `Asiakas` (`idAsiakas`),  
  CONSTRAINT `fk_Lainaus_Kirja1` FOREIGN KEY (`Kirja_idKirja`) REFERENCES `Kirja`  
  (`idKirja`)  
)
```

Table: Tekija

```
Create Table: CREATE TABLE `Tekija` (  
  `idTekija` int unsigned NOT NULL AUTO_INCREMENT,  
  `Etunimi` varchar(45) NOT NULL,  
  `Sukunimi` varchar(45) NOT NULL,  
  PRIMARY KEY (`idTekija`)  
)
```

Table: Teos

```
Create Table: CREATE TABLE `Teos` (  
  `idTeos` int unsigned NOT NULL AUTO_INCREMENT,  
  `Nimi` varchar(45) NOT NULL,  
  `ISBN` varchar(45) NOT NULL,  
  PRIMARY KEY (`idTeos`),  
  UNIQUE KEY `ISBN_UNIQUE` (`ISBN`)  
)
```

Table: Tekija_Teos

```
Create Table: CREATE TABLE `Tekija_Teos` (  
  `idTekija` int unsigned NOT NULL,  
  `idTeos` int unsigned NOT NULL,  
  PRIMARY KEY (`idTekija`, `idTeos`),  
  KEY `fk_Tekija_has_Teos_Teos1_idx` (`idTeos`),  
  KEY `fk_Tekija_has_Teos_Tekija_idx` (`idTekija`),  
  CONSTRAINT `fk_Tekija_has_Teos_Tekija` FOREIGN KEY (`idTekija`) REFERENCES `Tekija`  
  (`idTekija`),  
  CONSTRAINT `fk_Tekija_has_Teos_Teos1` FOREIGN KEY (`idTeos`) REFERENCES `Teos`  
  (`idTeos`)  
)
```