

# CS-A1150 Tietokannat kevät 2020

## Harjoitustyö

Antti Huttunen, [antti.t.huttunen@aalto.fi](mailto:antti.t.huttunen@aalto.fi)

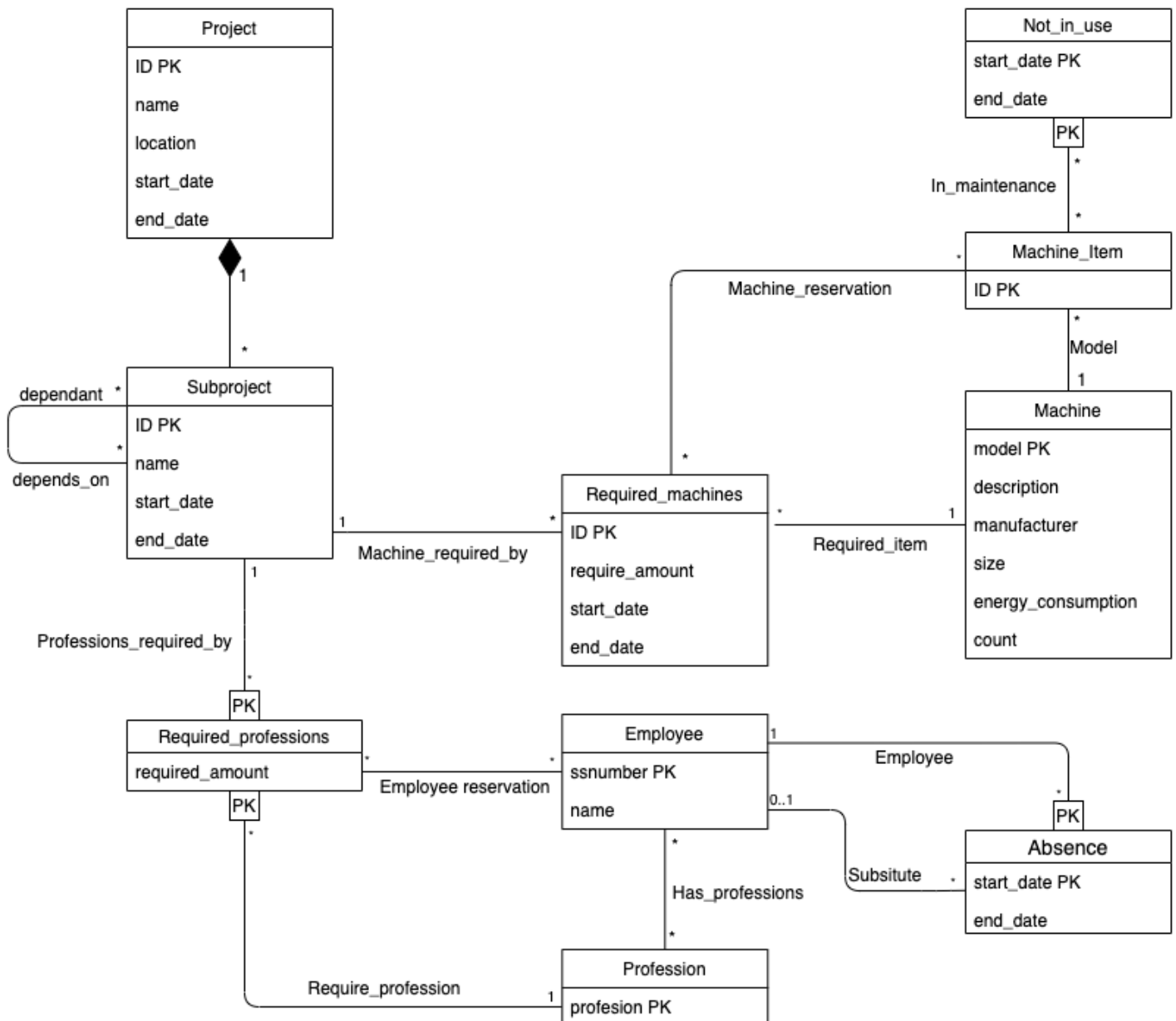
Oula Siljamo, [oula.siljamo@aalto.fi](mailto:oula.siljamo@aalto.fi)

# Sisällysluettelo

<b>OSA 1</b>	<b>2</b>
UML-kaavio	2
Relaatiot	3
Selostus	3
Projektin luonti osaprojekteineen	4
Työkoneen poisto käytöstä väliaikaisesti	5
Uuden työkoneen hankinta	5
Uuden työntekijän lisääminen	6
Työntekijän poissaolo	6
Työntekijän pätevyyksien päivittäminen	6
Projektien historian katsominen	6
Osaprojektin myöhästyminen	7
Tietokannan funktionaaliset riippuvuudet	8
Tietokannan anomalia	9
Ensimmäisen osan palautuksen jälkeen tehdyt muutokset	10
UML-kaavio	10
Relaatiot	10
<b>OSA 2</b>	<b>12</b>
Taulukoiden luonti	12
Hakemistot	16
Näkymät	16
Datan lisäys	17
Esimerkki käyttötapaukset	21
Varataan uusi kone tiettyyn osaprojektiin	21
Työkoneen poisto käytöstä	22
Uuden työkoneen hankinta	24
Uuden työntekijän lisäys osaprojektiin	25
Projektin historian katsominen	26
Tarkastetaan osaprojektien riippuvuuksia	27
Työntekijän poissaolo ja sijaisen etsiminen	27
Yleisten tilastojen hakemista	29

# OSA 1

## UML-kaavio



## Relaatiot

Projects(ID, name, location, start\_date, end\_date)

Subprojects(ID, name, start\_date, end\_date, projectID)

Depends\_on(dependantID, depends\_onID)

Required\_professions(subprojectID, profession, required\_amount)

Employee\_reservations(subProjectID, profession, employeeSSnumber)

Employees(ssnumber, name)

Absences(employeeID, substituteID, start\_date, end\_date)

Has\_professions(employeeID, profession)

Professions(profession)

Machines(model, description, manufacturer, size, energy\_consumption, count)

Machine\_items(ID, model)

Required\_machines(ID, subprojectID, model, required\_amount, start\_date, end\_date)

Machine\_reservations(machine\_itemID, requirementID)

Not\_in\_use(machine\_itemID, start\_date, end\_date)

## Selostus

- Projects relaatio kuvaa kokonaista projektia, se koostuu jostakin määrästä osaprojekteja.
- Subprojects relaatio kuvaa yksittäistä osaprojektia.
- Depends on relaatio kuvaa osaprojektin riippuvuutta toisista osaprojekteista.
- Required\_professions relaatio kuvaa mitä pätevyyskysymyksiä ja kuinka paljon jokin tietty osaprojekti vaatii.
- Employee\_reservations relaatio kuvaa työntekijää joka on Required\_profession relaation kautta liitetty tiettyyn osaprojektiin.
- Employees relaatio kuvaa tietoa kaikista työntekijöistä.
- Absences relaatio kuvaa tietoa poissaoloista ja sijaisista työntekijöille.
- Has\_professions relaatio kuvaa pätevyyskysymyksiä mitä työntekijöillä on.
- Professions relaatio kuvaa pätevyyskysymyksiä kaikkia pätevyyskysymyksiä mitä työntekijöillä on tai on ollut kaiken kaikkiaan.
- Machines relaatio kuvaa tietoa kaikista yksittäisistä työkonemalleista joita voi olla useita kappaleita.
- Machine\_items relaatio kuvaa yksittäisiä koneita mitä yrityksellä on käytössä, ei vain niiden malleja.
- Required\_machines relaatio kuvaa minkä tyyppisiä koneita tarvitaan mihinkin osaprojektiin ja mille aikavälille.
- Machine\_reservations relaatio kuvaa varausta yksittäiseen työkoneseen. Samalle aikavälille kuin siihen liitettyssä required\_machines relaatiossa on määriteltä.
- Not\_in\_use relaatiossa on tallennettuna kaikki varauksien ulkopuolisten asioiden aikavälit kun laite ei ole käytössä.

Käydään UML kaavion ja relaatioiden rakenne läpi muutamalla tapausesimerkillä: projektin luonti ja siihen liittyvien osaprojektien luonti ja muut prosessit, työkonene väliaikainen poistuminen käytöstä huollon takia, uuden työkonene hankinta, uuden työntekijän lisääminen, työntekijän poissaolo, työntekijän pätevyysien päivittäminen, projekti historian katsominen ja osaprojektin myöhästyminen.

## Projektin luonti osaprojekteineen

Luodaan uusi projekti talon rakentamiseen. Se tarvitsee osaprojektit: perusteiden rakennus, seinien rakennus ja katon rakennus. Osaprojektit ovat riippuvaisia toisistaan siten että perusteet täytyy tehdä ennen seinä ja seinät ennen kattoa.

Lisään ensiksi uusi Projekti "talon rakentaminen" *Projects* relaatioon. Sen jälkeen käydään luodaan uudet osaprojektit perusteiden rakennus, seinien rakennus ja katon rakennus. Käydään yksityiskohtaisesti läpi vain yksi osaprojekti, koska niihin liittyvät prosessit ovat pitkälti samat.

Tehdään uusi osaprojekti "seinien rakennus" tälle osaprojektille pitää lisäksi määritellä alkamispäivä (*start\_date*), arvioitu loppumispäivä (*end\_date*), tunniste (*ID*), joka on uniikki kaikille osaprojekteille, sekä tunniste projektiin mihin tämä osaprojekti liittyy (*projectID*). Tässä tapauksessa projektiin "talon rakentaminen". Seinien rakennuksen aloittaminen kuitenkin riippuu siitä, onko perustukset saatu rakennettua. Projektiä ei voida aloittaa ennen kuin tämä toinen osaprojekti on valmis. Lisään siis relaatioon *Depends\_on* molemmat osaprojektit niiden tunnisteilla kuvaamaan riippuvuutta. Osaprojektin aloituspäivä tulee voidaan tarkistaa ja ovatko kaikki osaprojektit, joista tämä osaprojekti on riippuvainen, valmiita eli ovatko niiden loppupäivämäärät (*end\_date*) jo ohitettu.

Osaprojekti tarvitsee työntekijöitä. Tämä tarve merkitään *Required\_professions* relaatioon jokaista eri pätevyyttä kohden erikseen ja kuinka paljon kyseisen pätevyyden omaavia henkilöitä tarvitaan. Nyt voidaan selvittää vapaana olevat työntekijät ja merkitä ne tähän osaprojektiin. Työntekijät varataan koko projektin ajaksi, joten ensin selvitetään työntekijät, jotka ovat vapaana kyseisellä aikavälillä. Kaikki työntekijät, jotka on joskus liitetty johonkin osaprojektiin löytyvät relaatiosta *Employees\_reservations*. Tästä relaatiosta voidaan selvittää mihin osaprojektiin kukin työntekijä liittyy ja näin voidaan selvittää, millä aikaväleillä työntekijä on varattuna yhdistämällä *Employees\_reservations* relaatioon ja *subprojects* relaatiosta osaprojektia vastaavat alku- ja loppupäivämäärät. Tämän jälkeen voidaan valita sieltä vain ne jotka ovat varattuina kyseisellä aikavälillä. Lisäksi pitää tarkistaa, onko joku työntekijöistä muusta syystä poissa käytöstä tarkistamalla *Absences* -relaatio. Relaatiossa kerrotaan poissaolot, jos työntekijä on kyseisellä aikavälillä muusta syystä poissa esim. loman takia. Liitto-operaatiolla nämä kaksi taulua voidaan yhdistää, jolloin meillä on tuloksena kaikki työntekijät, jotka eivät ole vapaana koko osaprojektin aikana.

Vapaana olevat työntekijät voidaan näin ollen selvittää kaikki työntekijät sisältävän *Employees* relaation ja edellisen tuloksen varatuista työntekijöistä erotuksen perusteella. Näistä vapaista työntekijöistä voidaan sitten varata tarvittavat työntekijät *Require\_professions* -relaation perusteella tarvittavista pätevyyksistä vertailemalla niitä

vapaiden työntekijöiden pätevyyksiin, jotka on määritelty *Has\_professions* relaatiossa. Vapaana oleva työntekijä varataa lisäämällä hänet *Employee\_reservations* -relaatioon siihen liittyvän työntekijätarve *Required\_professions* -relaation tunnistavien osaprojektin tunnisteeseen (subprojectID) ja tietyn pätevyyden (profession) kanssa. Jos kaikkiin pätevyyksiin ei löydy heti tarpeeksi työntekijöitä, voidaan niitä etsiä myöhemmin. Puutteelliset varaukset voidaan selvittää vertaamalla *Required\_professions* -relaatioon liitettyjä työntekijä varausten määrää *Required\_professions* -relaatiossa määriteltyyn tarvittavien varausten (*required\_amount*) määrään. Tämän jälkeen vapaita työntekijöitä voidaan etsiä edellä mainitulla tavalla.

Osaprojekti tarvitsee myös työkoneita. Työkoneita voidaan tarvita vain osaksi aikaa tiettyä osaprojektia, mutta myös useamman kerran. Jokainen tarvevaatimus tiettyä konemallia ja tietyllä aikavälillä kohden tehdään *Required\_machines* -relaatioon. Tähän relaatioon myös merkitään konemallin (model) ja osaprojektin tunnuksen (subprojectID) lisäksi konetarpeen alkupäivä (start\_date), loppupäivä (end\_date) ja tarvittava määrä kyseistä konemallia (*required\_amount*). Jokaisella tarvevaatimuksella on myös yksilöllinen tunniste (ID). Tämän jälkeen selvitetään, mitkä koneet ovat vapaana käytettäväksi tarvittaville aikaväleillä. Tämä saadaan selvitettyä helposti. Kaikki tehdyt varaukset sisältävästä *Machine\_reservations* -relaatiosta valitaan vain ne, jotka ovat varattuina kyseisellä aikavälillä. Tämä selvitetään yhdistämällä päivämäärät varaukseen liittyvästä tarvevaatimuksesta *Required\_machines* -relaatiossa. Tämän lisäksi valitaan *Not\_in\_use* -relaatiosta kaikki koneet, jotka eivät ole käytössä tällä aikavälillä. Molempien tulokset yhdistetään ja kaikki koneet sisältävästä *Machine\_items* -relaatiosta poistetaan edellisen tulokset ja näin saadaan sillä aikavälillä vapaana olevat koneet. Näistä voidaan varata tarvittavat koneet lisäämällä ne *Machine\_reservations* -relaatioon, joka ottaa tunnisteeksi siihen liittyvän tarvevaatimuksen tunnisteeseen (requirementID) ja kyseisen koneen yksilöllisen tunnisteeseen (machine\_itemID). Sama operaatio joudutaan tekemään kaikille erillisille aikaväleille. Jos koneita ei ole oikealla aikavälillä tarpeeksi vapaana, haku hetkellä voidaan tarvittavat koneet pyrkiä varaamaan myöhemmin vertailemalla *Machine\_reservations* -relaation kenttien määriä tietyn tarvevaatimuksen tunnisteeseen kanssa (requirementID). Jos niitä on vähemmän kuin *Required\_machines* -relaatiossa määritelty *required\_amount*, voidaan vapaita koneita etsiä edellä kuvatun mukaisesti.

Nyt osaprojektille on määritelty muut osaprojektit josta se on riippuvainen, tarvittavat työntekijät ja tarvittavat työkoneet. Kaikkia näitä voidaan päivittää tarvittaessa.

## **Työkoneen poisto käytöstä väliaikaisesti**

Kun työkone halutaan poistaa väliaikaisesti käytöstä esimerkiksi huollon takia, se voidaan tehdä yksinkertaisesti lisäämällä se halutulle aikavälille *Not\_in\_use* relaatioon. Mitään muuta ei tarvita

## **Uuden työkoneen hankinta**

Kun hankitaan uusi työkone ja halutaan lisätä se käytettäväksi täytyy muokata kahta paikkaan *Machine\_item* relaatiota johon lisätään työkoneen yksilöllinen tunniste kaikkien koneiden välillä se koneen mallin tunniste joka viittaa *Machine* relaatioon, jos kyseistä mallia ei vielä ole tietokannassa se täytyy lisätä ennen kuin työkone voidaan lisätä *Machine\_item*

relaatioon. Tällöin uusi työkonemalli lisätään *Machines* relaatioon tarvittaen tietojen kanssa. Tämän jälkeen voidaan uusi kone lisätä yksilöllisellä tunnisteellaan *Machine\_item* relaatioon. Kun *Machine\_item* relaatioon lisätään uusi työkone täytyy *Machine* relaation uuteen koneeseen liittyvän mallin tiedoissa kasvattaa count kentän arvoa yhdellä.

## Uuden työntekijän lisääminen

Kun halutaan lisätä uusi työntekijä se, lisätään tarvittavat tiedot *Employee* -relaatioon. Tämän lisäksi työntekijälle määritetään pätevyudet lisäämällä kaikki työntekijän pätevyudet yksitellen työntekijän tunnisteeseen kanssa *Has\_professions* -relaatioon. Jos kyseistä pätevyyttä ei ole vielä määritetty *Professions* -relaatiossa, täytyy se lisätä sinne ennen kuin voidaan lisätä kyseinen pätevyys työntekijälle.

## Työntekijän poissaolo

Kun työntekijä sairastuu, menee lomalle tai ei ole käytettävissä jostain muusta syystä, hänet merkitään *Absences* -relaatioon. Tähän lisätään työntekijän tunniste, mahdollisen sijaisen tunniste, aloituspäivä ja mahdollinen loppupäivä, jos se on tiedossa.

Sijainen etsitään ensin selvittämällä, mihin osaprojektiin työntekijä kuuluu tällä aikavälillä *Employee\_reservations* -relaatiosta tarkistamalla liitetty osaprojekti kyseisellä aikavälillä kuin myös pätevyys, joka tarvitaan. Tämän jälkeen voidaan hakea aikaisemmin mainitulla tavalla vapaita työntekijöitä tarvittavalla pätevyydellä sopivalle aikavälille. Tämän jälkeen voidaan lisätä sopivan työntekijän tunniste, jos sellainen löytyy, kyseisen työntekijän tämän aikavälin poissaolon kohdalle substitutelD kenttää *Absences* relaatiosta.

## Työntekijän pätevyyksien päivittäminen

Jos työntekijälle halutaan lisätä pätevyksiä esim. koulutuksen myötä tai halutaan poistaa pätevyksiä vanhentuneen lisenssin myötä. Ne voidaan yksinkertaisesti tehdä muokkaamalla *Has\_professions* -relaatiota lisäämällä uusi pätevyys työntekijälle. Jos kyseistä pätevyyttä ei vielä ole tietokannassa, täytyy se ensin lisätä *Professions* relaatioon. Jos työntekijältä halutaan poistaa pätevyys, tähän pätevyYTEEN viittaava rivi poistetaan työntekijän tunnisteella *Has\_professions* -relaatiosta. Oletetaan kuitenkin, että hän saa jatkaa osaprojektin loppuun, jos hänet on merkitty siihen tämän poistetun pätevyyden kanssa.

## Projektien historian katsominen

Joskus voidaan haluta selvittää aikaisempien projektien tietoja kaikista osaprojekteista ja kaikki työntekijät ja työkoneet, joita osaprojektissa on käytetty. Kaikki nämä pysyvät tallessa tietokannassa.

Jos halutaan selvittää osaprojektit, jotka liittyvät tiettyyn projektiin voidaan suoraan valita *Subprojects* -relaatiosta ne, joille ovat merkittynä projektin tunnisteeseen *projectID* kohdalle. Osaprojektin työntekijät voidaan selvittää suoraan katsomalla *Employee\_reservations* relaatiota, josta voidaan valita vain osaprojektin tunnisteeseen liitetty työntekijät. Tämän lisäksi pitää tarkistaa sijaisuudet, jokaiselta työntekijältä, jotka äskeisen haun perusteella saatiin osaprojektin aikaväliltä että saadaan selvitettyä myös sijaisena toimineet työntekijät.

Työkoneet selvitetään niin ikää yhdistämällä *Machine\_reservations* ja *Required\_machines*. Tästä voidaan valita ne, jotka ovat liitettyinä haluttuun osaprojektiin sen tunnisteiden avulla. Näin saadaan kaikki työkoneet, jotka ovat olleet käytössä kyseisen osaprojektin aikana. Muitakin tietoja pystytään selvittämään tai tarkemmin tiettyjä aikavälejä ja ne menevät pitkälti samoilla logiikoilla kuin edelle kuvatut.

## Osaprojektin myöhästyminen

Kun osaprojekti myöhästyy esim. riippuvaisen osaprojektin myöhästymisen takia, täytyy osaprojektin varaustenkin tietoja muuttaa ja tarkistaa ovatko tarvittavat työntekijät ja koneet edelleen vapaana uusilla aikaväleillä.

Työntekijöiden osalta ei tarvitse tehdä mitään suurempia muutoksia. Osaprojektin omia alkua ja loppupäivämääriä täytyy muuttaa. Sen lisäksi täytyy kuitenkin tarkistaa että osaprojektin työntekijät ovat edelleen vapaina tarvittavalla työajalla. Tämä voidaan selvittää helposti vertailemalla osaprojektin työntekijöitä kaikkiin työntekijä varauksiin ja katsoa, onko samalla työntekijä tunnisteella päällekkäisyyksiä uudella aikavälillä.

Työkoneiden suhteen täytyy muuttaa tarvevaatimusten ja varausten aikoja yksitellen, koska näitä ei tarvita koko osaprojektin ajaksi. Nämä tarkistetaan samalla tavalla jokaiselle koneelle ovatko ne varattuina uudella ajalla vertailemalla *Machine\_reservations* -relaatiossa yksittäisten työkoneyksilöiden varauksia ja tarkistamalla niiden päällekkäisyydet.



## Tietokannan funktionaaliset riippuvuudet

Relaatioissa ei ole muita funktionaalisia riippuvaisuuksia kuin ne jotka ylivaimet määrittävät, koska relaation muut kentät eivät määritä toisten kenttien arvoja.

- Projects
  - $ID \rightarrow name, location, start\_date, end\_date$   
 $\{ID\}^+ = \{ID, name, location, start\_date, end\_date\}$
- Subprojects
  - $ID \rightarrow name, start\_date, end\_date, projectID$   
 $\{ID\}^+ = \{ID, name, start\_date, end\_date, projectID\}$
- Required\_professions
  - $subprojectID, profession \rightarrow required\_amount$   
 $\{subprojectID, profession\}^+ = \{subprojectID, profession, required\_amount\}$
- Employees
  - $ssnumber \rightarrow name$   
 $\{ssnumber\}^+ = \{ssnumber, name\}$
- Absences
  - $employeeID, start\_date \rightarrow substituteID, end\_date$   
 $\{employeeID, start\_date\}^+ = \{employeeID, start\_date, substituteID, end\_date\}$
- Machines
  - $model \rightarrow description, manufacturer, size, energy\_consumption, count$   
 $\{model\}^+ = \{model, description, manufacturer, size, energy\_consumption, count\}$
- Machine\_items
  - $ID \rightarrow model$   
 $\{ID\}^+ = \{ID, model\}$
- Required\_machines
  - $ID \rightarrow subprojectID, model, required\_amount, start\_date, end\_date$   
 $\{ID\}^+ = \{ID, subprojectID, model, required\_amount, start\_date, end\_date\}$
- Not\_in\_use
  - $machine\_itemID, start\_date \rightarrow end\_date$   
 $\{machine\_itemID, start\_date\}^+ = \{machine\_itemID, start\_date, end\_date\}$

Tietokanta on Boyce-Codd-normaali muodossa, koska tietokannan funktionaaliset riippuvuudet eivät aiheuta toisteista tietoa eli relaatioiden vasemmanpuoleiset sulkeumat

sisältävät kaikki riippuvuuksien attribuutit. Kaikki funktionaaliset riippuvuudet muodostavat relaatioiden yliavaimet.

## **Tietokannan anomaliat**

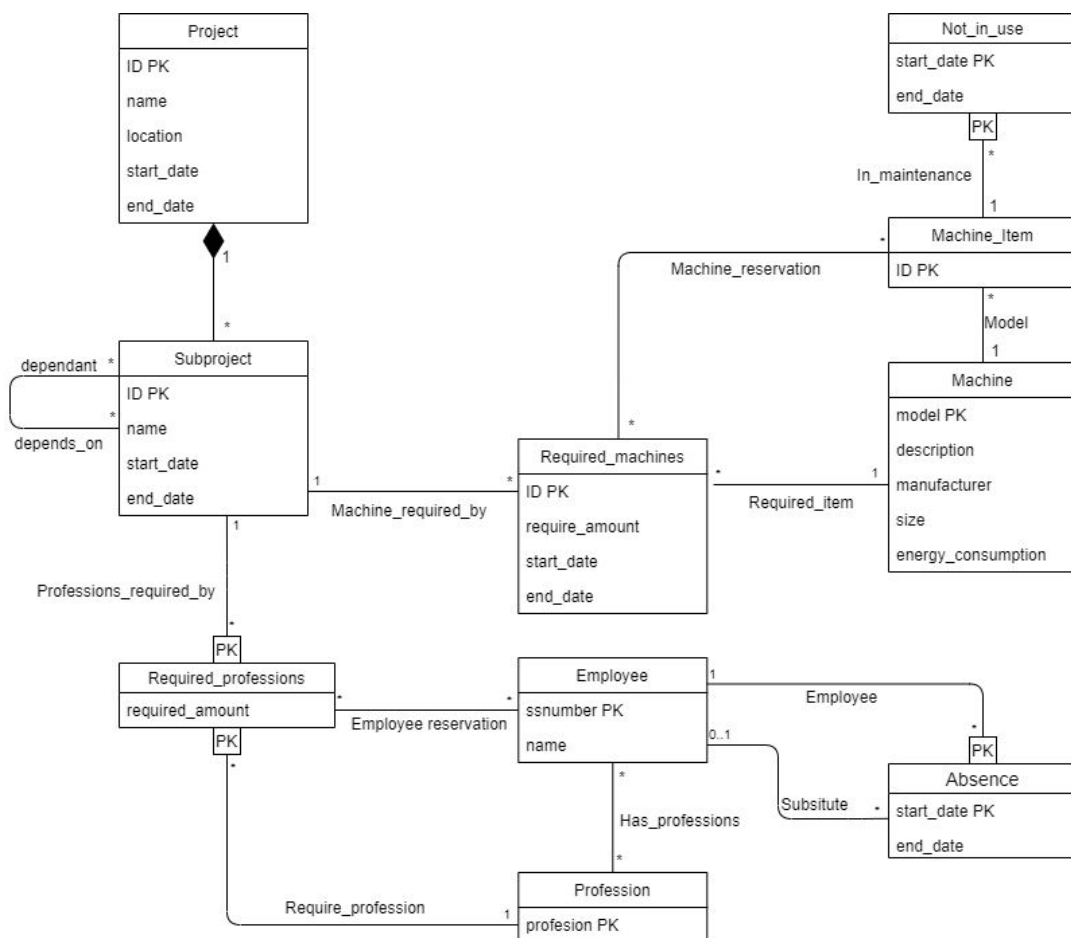
Tietokannassa ei esiinny selkeitä anomalioita. Tietokannan rakenteessa ei ole toisteisuutta, mutta siinä on kuitenkin pätevyysien (Profession) kohdalla anomaliaksi tulkittavaa, koska jos kyseistä pätevyyttä tarvitsee muuttaa, sitä pitää muuttaa kaikkialla. Tämä kuitenkin samalla relaation yliavain, joten sitä ei pitäisi muuttaa. Järkevämpi ratkaisu olisi ollut käyttää eri pätevyyksille omaa tunnistetta, jolloin nimeä voisi muuttaa suoraan ilman sivuvaikutuksia. Tietokannassa ei esiinny poistoanomaliaa koska poistot eivät poista jotakin tietoa, jota ei välttämättä oltaisi haluttu poistaa sivuvaikutuksena.

## Ensimmäisen osan palautuksen jälkeen tehdyt muutokset

Muutettiin UML-kaavion *Not\_in\_use* ja *Machine\_item* välinen relaation monesta yhteen. Poistettiin *count*-attribuutti *Machine*-relaatiosta.

Projektin ensimmäisen osan palaute sanoo että *profession*-attribuutti ei kuulu *Employee\_reservations*-relaatioon tämä kuitenkin tulee *Required\_professions*-relaatiosta ja vaaditaan, jotta voidaan tietää yksiselitteisesti mihin rooliin työntekijä on laitettu, joten tämä päätettiin jättää muuttamatta vaikka periaatteessa puuttuvat työntekijät voidaan selvittää työntekijöiden pätevyyksien avulla, jotka ovat varattuina kyseiseen osaprojektiin.

### UML-kaavio



Relaatiot

### Relaatiot

Projects(ID, name, location, start\_date, end\_date)

Subprojects(ID, name, start\_date, end\_date, projectID)

Depends\_on(dependantID, depends\_onID)

Required\_professions(subprojectID, profession, required\_amount)

Employee\_reservations(subprojectID, profession, employeeSSnumber)

Employees(ssnumber, name)

Absences(employeeID, substituteID, start\_date, end\_date)

Has\_professions(employeeID, profession)

Professions(profession)

Machines(model, description, manufacturer, size, energy\_consumption)

**attribuutti count poistettu Machine-relaatiosta**

Machine\_items(ID, model)

Required\_machines(ID, subprojectID, model, required\_amount, start\_date, end\_date)

Machine\_reservations(machine\_itemID, requirementID)

Not\_in\_use(machine\_itemID, start\_date, end\_date)

# OSA 2

## Taulukoiden luonti

Primary key:t on erikseen määritetty määreellä `NOT NULL` koska `sqlite` ei kiellä niissä `NULL` arvoja muuta kuin jos sarake on `INTEGER PRIMARY KEY` tai taulukko `WITHOUT ROWID` vastoin SQL-standardia.

Luodaan *Projects*-taulukko. ID:tä käytetään. `PRIMARY KEY` on projektin ID ja muilta arvoilta vaaditaan että ne eivät ole `NULL`, koska on oletettavaa että projektin tiedot ovat jo tiedossa sitä luodessa. Lisäksi `end_date` vaatii että se on vasta `start_date` jälkeen, koska projekti ei voi loppu ennen kuin se alkaa.

```
CREATE TABLE Projects(  
  ID INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  location TEXT NOT NULL,  
  start_date TEXT NOT NULL,  
  end_date TEXT CHECK (end_date >= start_date) NOT NULL  
);
```

Luodaan *Subprojects*-taulukko. Tunnistetaan käytetään ID:tä. Kaikille arvoille vaaditaan että ne eivät ole `NULL`, koska on oletettavaa että osaprojektin tiedot tiedetään jo sitä luodessa. Lisäksi vaaditaan että loppupäivämäärä on vasta alkupäivämäärän jälkeen, sekä vaaditaan että projekti johon osaprojekti liittyy on olemassa.

```
CREATE TABLE Subprojects(  
  ID INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  start_date TEXT NOT NULL,  
  end_date TEXT CHECK (end_date >= start_date) NOT NULL,  
  projectID INTEGER NOT NULL,  
  FOREIGN KEY (projectID) REFERENCES Projects(ID)  
);
```

Luodaan *Depends\_on* -taulukko, jossa molempien osaprojektien ID:t määritetään `PRIMARY KEY` arvolla, jotta riippuvuudet voidaan määrittää uniikisti. Lisäksi molempien osaprojektien täytyy löytyä osaprojekteista, jotta riippuvuus voidaan luoda.

```
CREATE TABLE Depends_on(  
  dependantID TEXT NOT NULL,  
  depends_onID TEXT NOT NULL,  
  PRIMARY KEY (dependantID, depends_onID),  
  FOREIGN KEY (dependantID) REFERENCES Subprojects(ID),  
  FOREIGN KEY (depends_onID) REFERENCES Subprojects(ID)  
);
```

Luodaan *Required\_professions*-taulukko, jossa määritellään osaprojekti, ammatti ja tarvittava määrä. Vaaditaan että osaprojekti on olemassa *Subprojects*-taulukossa ja että pätevyys on olemassa *Professions*-taulukossa kun uutta tarvetta työntekijöille luodaan.

```
CREATE TABLE Required_professions(  
    subprojectID INTEGER NOT NULL,  
    profession TEXT NOT NULL,  
    required_amount INTEGER CHECK(required_amount > 0) NOT NULL,  
    PRIMARY KEY(subprojectID, profession),  
    FOREIGN KEY(subprojectID) REFERENCES Subprojects(ID),  
    FOREIGN KEY(profession) REFERENCES Professions(profession)  
);
```

Luodaan *Employee\_reservations*-taulukko, johon luodaan uudet varaukset työntekijöistä. Tietoina vaaditaan osaprojekti, johon työntekijä haluaa, pätevyys mihin työntekijä halutaan varata sekä työntekijä joka varataan. Lisäksi vaaditaan että osaprojekti on olemassa *Subprojects*-taulukossa sekä että työntekijä ja haluttu pätevyys yhdistelmä löytyy *Has\_professions*-taulukosta, jossa määritellään työntekijöiden pätevyksiä.

```
CREATE TABLE Employee_reservations(  
    subprojectID INTEGER NOT NULL,  
    profession TEXT NOT NULL,  
    employeeSSnumber TEXT NOT NULL,  
    PRIMARY KEY(subprojectID, profession, employeeSSnumber),  
    FOREIGN KEY(SubprojectID) REFERENCES Subprojects(ID),  
    FOREIGN KEY(employeeSSnumber, profession) REFERENCES Has_professions(employeeID,  
profession)  
);
```

Luodaan *Employees*-taulukko, joka sisältää kaikki työntekijät, jotka ovat yrityksessä. Oletetaan että kaikkien työntekijöiden nimet ovat tiedossa.

```
CREATE TABLE Employees(  
    SSnumber TEXT PRIMARY KEY NOT NULL,  
    name TEXT NOT NULL  
);
```

Luodaan *Absences*-taulukko, jossa on kaikki poissaolot. PRIMARY KEY on työntekijän tunniste sekä poissaolon alkupäivä. Sijaisen tunniste ja loppupäivämäärä voivat olla NULL, koska nämä eivät välttämättä ole tiedossa kun poissaoloa merkitään, mutta on kuitenkin tärkeää merkitä poissaolot kun ne tulevat tietoon. Lisäksi vaaditaan että loppupäivämäärä tulee vasta alkupäivämäärän jälkeen, jos se on tiedossa. Vaaditaan että molemmat työntekijät poissaoleva ja sijainen on olemassa *Employees*-relaatiossa.

```
CREATE TABLE Absences(  
    employeeID TEXT NOT NULL,  
    substituteID TEXT,  
    start_date TEXT NOT NULL,  
    end_date TEXT CHECK (end_date >= start_date),  
    PRIMARY KEY(employeeID, start_date),  
    FOREIGN KEY(employeeID) REFERENCES Employees(SSnumber),  
    FOREIGN KEY(substituteID) REFERENCES Employees(SSnumber)  
);
```

Luodaan *Has\_professions*-taulukko, johon merkitään työntekijöiden pätevyyksiä. Vaaditaan että työntekijä on olemassa *Employees*-taulukossa ja pätevyys on olemassa *Professions*-taulukossa.

```
CREATE TABLE Has_professions(  
    employeeID TEXT NOT NULL,  
    profession TEXT NOT NULL,  
    PRIMARY KEY (employeeID, profession),  
    FOREIGN KEY(employeeID) REFERENCES Employees(SSnumber),  
    FOREIGN KEY(profession) REFERENCES Professions(profession)  
);
```

Luodaan *Professions*-taulukko, jossa on kaikki pätevyydet mitä yrityksessä on.

```
CREATE TABLE Professions(  
    profession TEXT PRIMARY KEY NOT NULL  
);
```

Luodaan *Machines*-taulukko, jossa on kaikki konemallit, jotka yritykselle on merkattu tietoon. malli on PRIMARY KEY, muut arvot voivat olla tyhjiä, koska on mahdollista että kaikkia tietoja ei ole tiedossa koneesta. Vaaditaan kuitenkin että koko ja energiankulutus ovat suurempaa kuin nolla, jos ne ovat tiedossa.

```
CREATE TABLE Machines(  
    model TEXT PRIMARY KEY NOT NULL,  
    description TEXT,  
    manufacturer TEXT,  
    size INTEGER CHECK(size > 0),  
    energy_consumption INTEGER CHECK(energy_consumption > 0)  
);
```

Luodaan *Machine\_items*-taulukko, johon merkitään koneet, joita yrityksellä on. Jokaisen laitteen täytyy viitata johonkin konemalliin, joka on olemassa *Machines*-taulukossa.

```
CREATE TABLE Machine_items(  
  ID INTEGER PRIMARY KEY NOT NULL,  
  model TEXT NOT NULL,  
  FOREIGN KEY(model) REFERENCES Machines(model)  
);
```

Luodaan *Required\_machines*-taulukko, johon merkitään uudet tarpeet koneista. Varaukset tunnistetaan uniikista ID:stä. Jokaisen varauksen tarpeen täytyy viitata johonkin osaprojektiin, joka on olemassa *Subprojects*-taulukossa sekä johonkin konetyyppiin, joka on olemassa *Machines*-taulukossa. Kaikki arvot oletetaan että eivät voi olla NULL, koska kaikki tiedot ovat tärkeitä ja oletettavasti myös tiedossa varausta tehdessä. Erityisesti *end\_date* sen lisäksi että sen täytyy olla vasta *start\_date* jälkeen se ei voi olla NULL koska oletetaan että vähintään alustava loppupäivämäärä on tiedossa.

```
CREATE TABLE Required_machines(  
  ID INTEGER PRIMARY KEY NOT NULL,  
  subprojectID INTEGER NOT NULL,  
  model TEXT NOT NULL,  
  required_amount INTEGER CHECK(required_amount >0) NOT NULL,  
  start_date TEXT NOT NULL,  
  end_date TEXT CHECK(end_date >= start_date) NOT NULL,  
  FOREIGN KEY(model) REFERENCES Machines(model),  
  FOREIGN KEY(subprojectID) REFERENCES Subprojects(ID)  
);
```

Luodaan *Machine\_reservations*-taulukko, johon merkitään varaukset, jotka on kerrottu *Required\_machines*-taulukossa. Jokaisen varauksen täytyy viitata johonkin koneeseen, joka on olemassa *Machine\_items*-taulukossa ja johonkin varauksen tarpeeseen joka löytyy *Required\_machines*-taulukosta.

```
CREATE TABLE Machine_reservations(  
  machine_itemID INTEGER NOT NULL,  
  requirementID INTEGER NOT NULL,  
  PRIMARY KEY(machine_itemID, requirementID),  
  FOREIGN KEY(machine_itemID) REFERENCES Machine_items(ID),  
  FOREIGN KEY(requirementID) REFERENCES Required_machines(ID)  
);
```



Luodaan *Not\_in\_use*-taulukko, johon merkitään kun kone on poissa käytöstä esim. huollon takia. Jokaisen käytöstä poiston täytyy viitata johonkin koneeseen *Machine\_items*-taulukossa. Lisäksi loppupäivämäärän täytyy olla vasta alkupäivämäärän jälkeen, jos se on tiedossa.

```
CREATE TABLE Not_in_use(  
    machine_itemID INTEGER NOT NULL,  
    start_date TEXT NOT NULL,  
    end_date TEXT CHECK(end_date >= start_date),  
    PRIMARY KEY(machine_itemID, start_date),  
    FOREIGN KEY(machine_itemID) REFERENCES Machine_items(ID)  
);
```

## Hakemistot

Luodaan muutama hakemisto taulukoista, joita todennäköisesti tarvittaisiin paljon kun tietokantaa käytettäisiin oikeasti. Tämä nopeuttaa hakuja koska koko taulukkoa ei tarvitse käydä lävitse hakiessa näiden hakemistojen perusteella, jolloin levyjä ei tarvitse lukea yhtä paljon

Tarvittavat pätevyudet osaprojektin mukaan.

```
CREATE INDEX ProfessionReservationIndex ON Required_professions(subprojectID);
```

Varatut työntekijät osaprojektin mukaan.

```
CREATE INDEX EmployeeReservationIndex ON Employee_reservations(subprojectID);
```

Tarvittavat koneet osaprojektin mukaan.

```
CREATE INDEX RequiredMachinesIndex ON Required_machines(subprojectID);
```

Varatut koneet tietyn varaus tarpeen mukaan.

```
CREATE INDEX ReservedMachinesIndex ON Machine_reservations(requirementID);
```

Osaprojektit pääprojektin tunnisteiden mukaan

```
CREATE INDEX ProjectIndex ON Subprojects(ProjectID);
```

## Näkymät

Luodaan joitakin näkymiä mitä voidaan haluta käyttää yleisessä tietokannan jokapäiväisessä käytössä.

Näkymä, josta saadaan kaikki työntekijät, jotka ovat tällä hetkellä poissa

```
CREATE VIEW currentlyOnLeave AS  
    SELECT * FROM Absences WHERE (end_date >= date('now') OR end_date IS NULL) AND  
    start_date <= date('now');
```

Näkymä, josta nähdään kaikki poissaolot joiden loppu ei ole tiedossa.

```
CREATE VIEW missingSubstitute AS  
    SELECT * FROM Absences WHERE substituteID IS NULL;
```

**Näkymä, joka näyttää kaikkien työntekijöiden poissaolojen kestojen keskiarvon yksittäin ja lisäksi näyttää kaikkien työntekijöiden poissaolojen kestojen keskiarvon yhdessä.**

```
CREATE VIEW AverageAbsences AS SELECT employeeID, MAX(A) AS 'Average lenght of
absence'
FROM (SELECT employeeID, AVG(julianday(end_date) - julianday(start_date)) AS A FROM
Absences GROUP BY employeeID
UNION SELECT SSnumber, 0 FROM Employees
UNION SELECT 'All', AVG(julianday(end_date) - julianday(start_date)) FROM Absences)
GROUP BY employeeID;
```

## Datan lisäys

```
INSERT INTO Projects
VALUES (1, 'Inkuterde', 'Päälafka', '2020-6-1', '2020-8-3');

INSERT INTO Projects
VALUES (2, 'Kiltis_laajennus', 'Päälafka', '2021-1-5', '2021-5-1');

INSERT INTO Subprojects
VALUES (11, 'Oven rakennus', '2020-6-1', '2020-6-22', 1);

INSERT INTO Subprojects
VALUES (12, 'Terassin rakennus', '2020-6-15', '2020-8-2', 1);

INSERT INTO Subprojects
VALUES (21, 'seinän hajotus', '2021-1-5', '2021-3-22', 2);

INSERT INTO Subprojects
VALUES (22, 'sähkö ja putkityöt', '2021-3-23', '2021-5-1', 2);

INSERT INTO Depends_on
VALUES (22, 21);

INSERT INTO Employees
VALUES ('12031996-66P', 'Jones');

INSERT INTO Employees
VALUES ('30121995-69X', 'Nalle');

INSERT INTO Employees
VALUES ('25061999-990', 'Tauski');

INSERT INTO Employees
VALUES ('09112015A700', 'Maikki');

INSERT INTO Professions
VALUES ('putkimies');

INSERT INTO Professions
VALUES ('koodari');

INSERT INTO Professions
VALUES ('sähkömies');
```

```

INSERT INTO Professions
VALUES ('kirvesmies');

INSERT INTO Professions
VALUES ('maisema-arkkitehti');

INSERT INTO Has_professions
VALUES ('12031996-66P', 'putkimies');

INSERT INTO Has_professions
VALUES ('12031996-66P', 'koodari');

INSERT INTO Has_professions
VALUES ('30121995-69X', 'koodari');

INSERT INTO Has_professions
VALUES ('30121995-69X', 'sähkömies');

INSERT INTO Has_professions
VALUES ('25061999-99O', 'maisema-arkkitehti');

INSERT INTO Has_professions
VALUES ('09112015A70O', 'kirvesmies');

INSERT INTO Required_professions
VALUES (11, 'kirvesmies', 1);

INSERT INTO Required_professions
VALUES (11, 'sähkömies', 1);

INSERT INTO Required_professions
VALUES (11, 'putkimies', 1);

INSERT INTO Required_professions
VALUES (12, 'maisema-arkkitehti', 1);

INSERT INTO Required_professions
VALUES (12, 'kirvesmies', 1);

INSERT INTO Required_professions
VALUES (21, 'koodari', 2);

INSERT INTO Required_professions
VALUES (21, 'sähkömies', 1);

INSERT INTO Required_professions
VALUES (21, 'putkimies', 1);

INSERT INTO Required_professions
VALUES (21, 'kirvesmies', 1);

INSERT INTO Required_professions
VALUES (22, 'sähkömies', 1);

```

```

INSERT INTO Required_professions
VALUES (22, 'putkimies', 1);

INSERT INTO Required_professions
VALUES (22, 'kirvesmies', 1);

INSERT INTO Employee_reservations
VALUES (21, 'koodari', '12031996-66P');

INSERT INTO Absences
VALUES ('12031996-66P', '30121995-69X', '2020-02-25', NULL);

INSERT INTO Absences
VALUES ('30121995-69X', NULL, '2020-03-12', '2020-11-13');

INSERT INTO Machines
VALUES ('Super500', 'vasara', 'ENG', 2, 1);

INSERT INTO Machines
VALUES ('Mega99', 'pora', 'ENG', 3, 5);

INSERT INTO Machines
VALUES ('DELL-99', 'l  pp  ri', 'SCI', 5, 9);

INSERT INTO Machines
VALUES ('Fiskars33', 'lapio', 'ENG', 6, 1);

INSERT INTO Machines
VALUES ('galaxy5', 'saha', 'ELEC', 3, 5);

INSERT INTO Machine_items
VALUES (1, 'Super500');

INSERT INTO Machine_items
VALUES (2, 'Super500');

INSERT INTO Machine_items
VALUES (3, 'Super500');

INSERT INTO Machine_items
VALUES (4, 'Mega99');

INSERT INTO Machine_items
VALUES (5, 'Mega99');

INSERT INTO Machine_items
VALUES (6, 'DELL-99');

INSERT INTO Machine_items
VALUES (7, 'Fiskars33');

INSERT INTO Machine_items
VALUES (8, 'Fiskars33');

```

```
INSERT INTO Machine_items  
VALUES (9, 'Fiskars33');
```

```
INSERT INTO Machine_items  
VALUES (10, 'galaxy5');
```

```
INSERT INTO Required_machines  
VALUES(1, 11, 'galaxy5', 1, '2020-6-1', '2020-6-22');
```

```
INSERT INTO Required_machines  
VALUES(2, 11, 'Super500', 1, '2020-6-1', '2020-6-22');
```

```
INSERT INTO Required_machines  
VALUES(3, 21, 'DELL-99', 2, '2021-1-5', '2021-3-22');
```

```
INSERT INTO Machine_reservations  
VALUES(10, 1);
```

```
INSERT INTO Machine_reservations  
VALUES(6, 3);
```

```
INSERT INTO Machine_reservations  
VALUES(2, 2);
```

```
INSERT INTO Not_in_use  
VALUES(2, '2020-6-1', '2020-6-22');
```

# Esimerkki käyttötapaukset

## 1. Varataan uusi kone tiettyyn osaprojektiin

Jotta kone voidaan varata osaprojektiin jollekin aikavälille, täytyy tarkistaa, onko kone vapaa halutulla aikavälillä. Tämän jälkeen kun on löydetty vapaat koneet voidaan varata kone aikavälille.

Tämä vaatii, että tehdään kysely, joka tarkistaa vapaat koneet ja että päivitetään Required\_machines- ja Machine\_reservations -taulukoihin uudet rivit.

```
-- Vapaana olevan koneen tarkastus
```

```
SELECT ID, model
FROM Machine_items
EXCEPT
SELECT Mi.ID, Mi.model
FROM Machine_items AS Mi, Machine_reservations AS Mres, Required_machines AS
RM,
Not_in_use AS N
WHERE Mi.ID = Mres.machine_itemID AND Mres.requirementID = RM.ID AND
(RM.end_date >= '2020-01-08' AND RM.start_date <= '2020-06-01') OR
(N.end_date >= '2020-01-08' AND N.start_date <= '2020-06-01');
```

```
/* Output
```

ID	model
1	Super500
3	Super500
4	Mega99
5	Mega99
6	DELL-99
7	Fiskars33
8	Fiskars33
9	Fiskars33

```
*/
```

```
--Koneen varaus
```

```
INSERT INTO Required_machines
VALUES(4, 11, 'DELL-99', 1, '2020-01-06', '2020-13-06');
```

```
INSERT INTO Machine_reservations
VALUES(6, 4);
```

## 2. Työkoneen poisto käytöstä

Laitetaan kone huoltoon, mikä voi viivästyttää projektia, jos samaa konetyyppiä ei ole useampaa samanlaista.

Kun työkone poistetaan käytöstä esimerkiksi huollon takia, täytyy päivittää Not\_in\_use-taulukko. Tarkistetaan, onko kone jonkun osaprojektin käytössä. Jos on niin varataan uusi kone osaprojektiin, jos konetyyppiä ei ole vapaana joudutaan viivästyttämään projektia. Kone 6 menee huoltoon 25.4.2021, joka päättyy 1.4.2021. Tarkistetaan, onko samaa konetyyppiä vapaana. Jos ei, viivästetään projektia. Jos projekti viivästyy joudutaan tarkistamaan riippuuko jokin projekti viivästyneestä osaprojektista ja joudutaanko riippuvan osaprojektin koneiden varausta pidentämään.

```
--Poistetaan kone käytöstä
INSERT INTO Not_in_use
VALUES(6, '2021-03-01', '2021-04-01');

--tarkistetaan oliko kone varattu jollekin osaprojektille

SELECT S.ID
FROM Subprojects AS S, machine_reservations AS MR, Required_machines AS R
WHERE S.ID = subprojectID AND MR.requirementID = R.ID AND MR.Machine_itemID
= 6 AND (R.end_date >= '2021-03-01' AND R.end_date <= '2021-04-01');

--kone oli varattu osaprojektiin 21
--tarkistetaan onko vapaana samaa kone tyyppiä olevaa konetta

/* Output:
ID
-----
21

*/

SELECT ID, model
FROM Machine_items
EXCEPT
SELECT Mi.ID, Mi.model
FROM Machine_items AS Mi, Machine_reservations AS Mres, Required_machines AS
RM,
Not_in_use AS N, Machines AS M
WHERE Mi.ID = Mres.machine_itemID AND Mres.requirementID = RM.ID AND
Mi.model = M.model AND
M.description = 'läppäri' AND
(RM.end_date >= '2021-03-22' AND RM.start_date <= '2021-03-01') OR
(N.end_date >= '2021-03-22' AND N.start_date <= '2021-03-01');

/* Output: {NO OUTPUT}*/
/*koska toista samalla kuvauksella olevaa konetta ei löytynyt,
viivästytetään osaprojektia ja projektia ja pidennetään koneen varausta*/
```

```

--tarkistetaan ensin onko jokin muu osaprojekti riippuvainen
osaprojektista, joka viivästyy

SELECT dependantID
FROM Subprojects, Depends_on
WHERE ID = depends_onID AND ID = 21;

/* Output:
dependantID
-----
22
*/

--myöhennetään molempia osaprojekteja, projektia ja pidennetään varausta
UPDATE Subprojects
SET end_date = '2021-04-22'
WHERE ID = 21;

UPDATE Subprojects
SET end_date = '2021-06-01', start_date = '2021-04-23'
WHERE ID = 21;

--tarkistetaan mihin projektiin osaprojekti 21 kuuluu

SELECT P.ID
FROM Projects AS P, Subprojects AS S
WHERE ProjectID = P.ID AND S.ID = 21;

/* Output:
ID
-----
2
*/

--päivitetään projekti 2
UPDATE Projects
SET end_date = '2021-06-01'
WHERE ID = 2;

--tarkistetaan oliko osaprojektilla 22 konevarauksia, joita pitäisi pidentää
SELECT COUNT(RequirementID)
FROM Required_machines, Machine_reservations
WHERE ID = requirementID AND subprojectID = 22;

/* Output:
COUNT(RequirementID)
-----
0
*/

--Koska varauksia ei ollut pidennetään 21 varausta koneesta 6
--Tätä ennen tarkistetaan, onko kone vapaa uudella aikavälillä

SELECT DISTINCT SubprojectID

```



```

FROM Machine_items AS Mi, Machine_reservations AS Mres, Required_machines AS
RM,
Not_in_use AS N
WHERE Mi.ID = Mres.machine_itemID AND Mres.requirementID = RM.ID AND Mi.ID =
6 AND
(RM.end_date >= '2021-04-22' AND RM.start_date <= '2021-04-01');

/* Output: {NO OUTPUT} */

--Koska kone on vapaa pidennetään varausta
UPDATE Required_machines
SET end_date = '2021-04-22'
WHERE subprojectID = 21 AND model = 'DELL-99';

```

### 3. Uuden työkoneen hankinta

Tarkistetaan, onko samaa mallia jo tietokannassa. Jos ei lisätään se tietokantaan.  
 Luodaan uusi Machine\_item.

```

--Hankitaan uusi työkone, jonka malli on Samsung-G55 ja kuvaus puhelin
--Tarkistetaan onko samanmallisia koneita valmiiksi tietokannassa

```

```

SELECT COUNT(model)
FROM Machines
WHERE model = 'Samsung-G55';

```

```

/* Output:
COUNT(model)
-----
0
*/

```

```

--Koska koneita ei ollut lisätään kone tietokantaan

```

```

INSERT INTO Machines
VALUES('Samsung-G55', 'puhelin', 'SCI', 1, 100);

```

```

INSERT INTO Machine_items
VALUES(155, 'Samsung-G55');

```

#### 4. Uuden työntekijän lisäys osaprojektiin

Luodaan uusi pätevyyden tarve osaprojektille ja varataan uusi työntekijä sitä varten. Ensimmäisen luodaan uusi varaus osaprojektille. Sen jälkeen etsitään vapaana olevat työntekijät ja varataan uusi, jolla on sopiva pätevyys. Tämän perusteella voidaan tehdä uusi varaus työntekijästä tarvittavalle ajalle. Tässä tapauksessa halutaan uusi maisema-arkkitehti osaprojektille, jonka ID=22. Vapaana on vain työntekijä tunnuksella 25061999-990 joten hänet varataan kyseiseen rooliin.

```
/* Lisää uusi tarvittava pätevyys osaprojektille jonka ID=22 */
INSERT INTO Required_professions VALUES(22, 'maisema-arkkitehti', 1);

/* Etsi vapaana olevat työntekijät, joilla on tarvittava pätevyys. */
SELECT employeeID FROM Has_professions
  WHERE profession = 'maisema-arkkitehti'
EXCEPT
SELECT employeeSSnumber FROM (SELECT * FROM (
  SELECT DISTINCT employeeSSnumber, start_date, end_date FROM
Employee_reservations AS E
  JOIN Subprojects AS S
  ON S.ID=E.subprojectID
  UNION
  SELECT employeeID, start_date, end_date FROM Absences
  UNION
  SELECT substituteID, start_date, end_date FROM Absences
) CROSS JOIN (SELECT ID AS pID, start_date AS pStart, end_date AS pEnd FROM
Subprojects WHERE pID=22))
WHERE (end_date>=pStart OR end_date IS NULL) AND start_date<=pEnd;

/* Output:
employeeID
-----
25061999-990
*/

/* Varataan osaprojektiin yksi vapaista haluttuun rooliin. */
INSERT INTO employee_reservations VALUES(22, 'maisema-arkkitehti',
'25061999-990');
```

## 5. Projektin historian katsominen

Selvitetään erilaisia tietoja tietyistä projekteista, kuten projektista, jonka ID=1. Haetaan ensin tiedot kyseisestä Projektista. Sen jälkeen haetaan projektiin liittyvät osaprojektit. Tämän lisäksi osaprojekteihin varatut työntekijät ja työkoneet.

```
/* Haetaan Projektin tiedot */
SELECT * FROM Projects WHERE ID=1;
```

/\* Output:

ID	name	location	start_date	end_date
1	Inkuterde	Päälafka	2020-06-01	2020-08-03

\*/

```
/* Haetaan Projektiin liittyvien osaprojektien tiedot */
SELECT * FROM Subprojects WHERE projectID=1;
```

/\* Output:

ID	name	start_date	end_date	projectID
11	Oven rakennus	2020-06-01	2020-06-22	1
12	Terassin rake	2020-06-15	2020-08-02	1

\*/

```
/* Haetaan kaikki työntekijät jotka ovat olleet tekemässä projektia kaikista osaprojekteista. */
```

```
SELECT subprojectID, SSnumber, name, profession FROM Employee_reservations
JOIN Employees AS E ON Employee_reservations.employeeSSnumber = E.SSnumber
WHERE subprojectID IN (SELECT ID FROM Subprojects WHERE projectID=1);
```

/\* Output:

subprojectID	SSnumber	name	profession
11	12345678A123	Puuha Pete	putkimies
11	30121995-69X	Nalle	sähkömies
12	30121995-69X	Nalle	sähkömies

\*/

```
/* Haetaan kaikki koneet, jotka ovat olleet käytössä projektin kaikissa osaprojekteissa. */
```

```
SELECT subprojectID, model, machine_itemID FROM Required_machines JOIN
Machine_reservations ON Machine_reservations.requirementID =
Required_machines.ID
WHERE subprojectID IN (SELECT ID FROM Subprojects WHERE projectID=1);
```

/\* Output:

subprojectID	model	machine_itemID
11	galaxy5	10
11	Super500	2
11	DELL-99	6

\*/

## 6. Tarkastetaan osaprojektien riippuvuuksia

Haetaan kaikkien osaprojektien riippuvuudet ja päivitetään niiden alkupäivämäärät joilla, on riippuvuuksia osaprojekteihin jotka loppuvat vasta kun kyseisen osaprojektin pitäisi alkaa

```
/* Haetaan kaikki osaprojektit joilla on riippuvuuksia ja joilla on
aikaisempi alkupäivämäärä kuin riippuvuuden loppumiset */
SELECT dependantID, COUNT(*) FROM (SELECT * FROM Depends_on AS D
JOIN (SELECT ID, end_date FROM Subprojects) AS S1
ON S1.ID = D.depends_onID
JOIN (SELECT ID, start_date FROM Subprojects) AS S2
ON S2.ID = D.dependantID)
WHERE end_date > start_date GROUP BY dependantID;

/* Output:
dependantID  COUNT(*)
-----
12           1
22           1
*/

/* Päivitetään saatujen tuloksien arvoja*/
UPDATE Subprojects SET
end_date = (SELECT DATE((SELECT MAX(julianday(end_date)) FROM Depends_on,
Subprojects WHERE dependantID = 12 AND depends_onID = ID) -
julianday(start_date) + julianday(end_date))),
start_date = (SELECT MAX(end_date) FROM Depends_on, Subprojects WHERE
dependantID = 12 AND depends_onID = ID)
WHERE ID = 12;

UPDATE Subprojects SET
end_date = (SELECT DATE((SELECT MAX(julianday(end_date)) FROM Depends_on,
Subprojects WHERE dependantID = 22 AND depends_onID = ID) -
julianday(start_date) + julianday(end_date))),
start_date = (SELECT MAX(end_date) FROM Depends_on, Subprojects WHERE
dependantID = 22 AND depends_onID = ID)
WHERE ID = 22;
```

## 7. Työntekijän poissaolo ja sijaisen etsiminen

Joskus työntekijöiden täytyy olla poissa esim. sairastumisen tai koulutuksen takia. Etsitään. Lisätään työntekijälle poissaolo, selvitetään mihin osaprojektiin työntekijä kuuluu poissaolon aikana ja etsitään sopiva työntekijä tilalle ja päivitetään poissaoloa sijaisen tiedoilla.

```
/* Lisätään työntekijä poissaolevaksi, sijaista ei vielä tällä hetkellä ole
*/
INSERT INTO Absences
VALUES ('12345678A123', NULL, '2020-06-05', '2020-06-06');
```

```

/* Selvitetään mihin osaprojekteihin työntekijä kuuluu poissaolon aikana */
SELECT subprojectID FROM Employee_reservations AS E
JOIN Subprojects AS S ON S.ID = E.subprojectID
WHERE (end_date>='2020-06-05' AND start_date<='2020-06-06') AND
employeeSSnumber = '12345678A123';

```

```

/* Output:
subprojectID
-----
11
*/

```

```

/* Etsitään vapaa työntekijä sopivalla pätevyydellä sijaiseksi. */
SELECT employeeID FROM Has_professions
WHERE profession = (SELECT profession
                     FROM Employee_reservations WHERE employeeSSnumber = '12345678A123' AND
                     subprojectID = 11)
EXCEPT
SELECT employeeSSnumber FROM (SELECT employeeSSnumber, start_date, end_date
FROM Employee_reservations AS E
JOIN Subprojects AS S ON S.ID = E.subprojectID
UNION
SELECT employeeID, start_date, end_date FROM Absences
UNION
SELECT substituteID, start_date, end_date FROM Absences)
WHERE (end_date>='2020-06-05' AND start_date<='2020-06-06');

```

```

/* Output:
employeeID
-----
12031996-66P
*/

```

```

/* Valitaan tuloksista yksi henkilö ja päivitetään sijaisuus */
UPDATE Absences SET substituteID='12031996-66P' WHERE
employeeID='12345678A123' AND start_date='2020-06-05';

```

## 8. Yleisten tilastojen hakemista

Usein halutaan selvittää yleisiä tilastoja eri asioista tietokannasta. Tässä muutamia esimerkkejä jotka voivat kiinnostaa yritystä.

**Halutaan selvittää kuinka paljon osaprojekteja eri projekteilla on**

```
/* Halutaan selvittää kuinka paljon osaprojekteja eri projekteilla on*/
SELECT Projects.ID, Projects.name, COUNT(subprojects.ID) AS 'osaprojektien
määrä'
FROM Subprojects, Projects
WHERE projectID = Projects.ID
GROUP BY projectID;
```

/\* Output:

ID	name	osaprojektien määrä
1	Inkuterde	2
2	Kiltis_laa	2

\*/

**Ollaan kiinnostuneita siitä millaisia pätevyyskiä ja kuinka paljon yrityksessä on**

```
/* Halutaan selvittää kuinka paljon eri pätevyyskiä yrityksellä on*/
SELECT profession, Count(employeeID) as 'määrä'
FROM Has_professions
GROUP BY profession
ORDER BY profession;
```

/\* Output:

profession	määrä
kirvesmies	1
koodari	3
maisema-ar	1
putkimies	2
sähkömies	1

\*/

**Selvitetään kuinka paljon erilaisia koneita yrityksellä on**

```
/* Millaisia koneita ja kuinka paljon yrityksellä on */
SELECT model, COUNT(*) AS 'määrä' FROM Machine_items GROUP BY model;
```

/\* Output:

model	määrä
DELL-99	1
Fiskars33	3
Mega99	2
Samsung-G5	1
Super500	3
galaxy5	1

\*/