

TIETOKANTASUUNNITELMA

1. Johdanto

Suunnitelma sisältää kuvauksen tietokannan suunnittelussa käytetyistä periaatteista, kuvan relaatiotietokannasta, sekä sen luomiseen liittyvät SQL-lauseet kommentteineen.

2. Suunnitteluperiaatteita

Tietokannan suunnittelun alkuvaiheessa havaittiin opintokokonaisuuksien ja tutkintovaatimusten koostuvan samanlaisesta rekursiivisesta ryhmittelystä. Näin nähtiin järkeväksi luoda tietokanta mahdollisimman dynaamiseksi useamman sisäisen ryhmittelyn mahdollistamiseksi. Tietokannan suunnittelun lähtökohtana on siis ollut luoda taulurakenteesta mahdollisimman laajennettava pyrkien mahdollisimman vähän rajoituksia asettavaan kokonaisuuteen.

Kokonaisuutena tietorakenne perustuu ajatukseen, että tutkintorakenne esitetään rekursiivisista ryhmistä koostuvana ryhmänä. Tämän rakenteen avulla voidaan esittää myös opiskelijan suunnitelmat, joista tallennetaan perustiedot ja siihen liittyvien kurssien ja kokonaisuuksien suunnitellut ajankohdat, mutta ei rakennetta.

Opiskelijan suunnitelmien lisäksi ja niiden teon tueksi, Korppi-järjestelmästä löytyy kurssien suoritustiedot. Ne voidaan esittää käyttöliittymässä rinnalla, mutta niitä ei muuten huomioida opiskelusuunnitelman tietorakenteessa.

2. Teknisiä ratkaisuja

Taulujen ja taulujen kenttien nimissä käytetään Korppi-järjestelmän kehityksessä käytettyä nimeämistapaa soveltuvin osin. Pääsääntöisesti kaikki HOPS-sovellukseen liittyvät taulut alkavat study-alulla. Id-kentät nimetään taulujen mukaan Korppi-perinteitä kunnioittaen.

Tietojen poistamisessa menetellään edelleen Korppi-perinteen mukaisesti, jottei tietoa vahingossa tuhota. Jokaiseen tauluun liittyy siis kenttä deleted, joka kuvaa tietueen poistamista. Viite-eheyksien säilyminen taataan käyttämällä ON DELETE RESTRICT -määrittystä viite-avaimiin liittyen. Viitteisiin liittyvät päivitykset peritään, eli yleensä viite-avaimiin liittyen asetetaan ON UPDATE CASCADE.

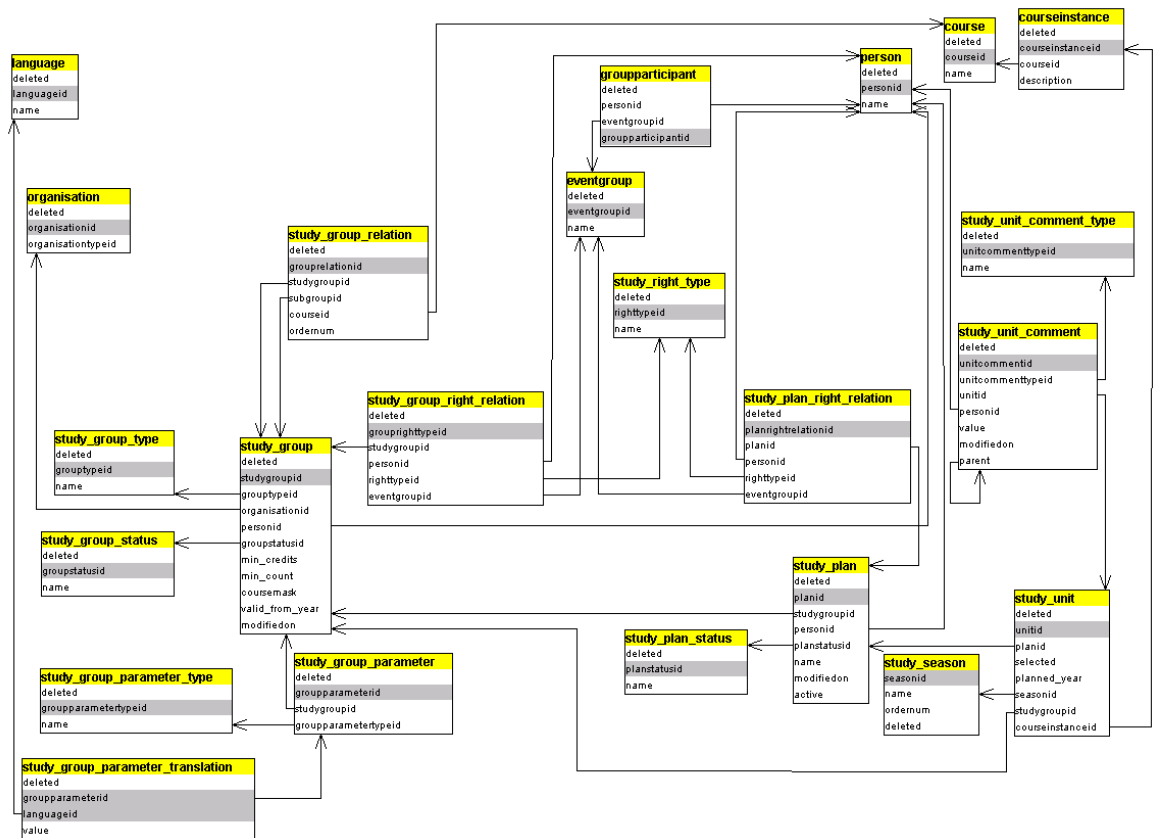
3. Olemassa olevien Korppi-taulujen hyödyntäminen

Korppi-järjestelmästä käytetään ainakin seuraavia olemassaolevia tauluja:

- Henkilö-taulu (person)
- Organisaatio-taulu (organisatio & organisationtranslation)
- Kurssi-nimike -taulu (course & coursetranslation)
- Kielitaulu (language)
- Ryhmätaulu (eventgroup & groupparticipant)

Ko. taulut on kuvattu tässä suunnitelmassa vain tarvittavin osin ja pyritty esittämään erillään muista. Näiden taulujen lisäksi 'opiskelija'-liittymässä hyödynnetään kurssiesiintymä taulua suoritusten esittämiseen.

4. Kokonaiskuva tietorakenteesta



Kuva 1. Kuva relaatiotietokannasta

5. Tietokantataulujen rakenne

Seuraavassa on kuvattu tietokantataulujen rakenne, sekä pyritty lyhyesti ajantuksenomaisesti kuvaamaan niiden sisältöä. Tarkempi käyttötarkoitus ilmenee Vaatimusmäärittelyn ja muiden suunnitteludokumenttien avulla.

5.1 Yleistaulut

```
CREATE TABLE study_right_type (  
-- 'sihteerin' ja 'opiskelijan' -toimintoihin liittyvät oikeudet  
--  
  deleted boolean,  
  righttypeid integer NOT NULL,  
  name varchar NOT NULL,  
  PRIMARY KEY (righttypeid)  
);
```

5.2 'Tutkintorakenteet'

```
CREATE TABLE study_group_status (  
-- ryhmittelyihin liittyvät tilat  
--  
  deleted boolean,  
  groupstatusid integer NOT NULL,  
  name varchar NOT NULL,  
  PRIMARY KEY (groupstatusid)  
);
```

```
CREATE TABLE study_group_type (  
-- ryhmittelyn tyyppi, lähinnä ohjelmoijan käyttöön  
--  
  deleted boolean,  
  grouptypeid integer NOT NULL,  
  name VARCHAR NOT NULL,  
  PRIMARY KEY (grouptypeid)  
);
```

```
CREATE TABLE study_group (  
-- ryhmittelyn perustiedot  
-- yhteistä kaikille ryhmille, kielestä riippumaton data  
--  
  deleted boolean,  
  studygroupid integer NOT NULL,  
  grouptypeid integer,  
  organisationid integer NOT NULL,  
  personid integer, -- henkilökohtaiset tutkintovaatimukset  
  (korvaavuudet)  
  groupstatusid integer,  
  min_credits integer, -- pakollisuus, valinnaisuus, vaihtoehtoisuus  
  min_count integer, -- pakollisuus, valinnaisuus, vaihtoehtoisuus  
  coursemask varchar, -- ryhmään kuuluvien kurssien määrittely hakumaskilla  
  valid_from_year integer, -- minkä vuoden  
  modifiedon TIMESTAMP,  
  PRIMARY KEY (studygroupid),  
  FOREIGN KEY (grouptypeid) REFERENCES study_group_type (grouptypeid)
```

```

    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (organisationid) REFERENCES organisation (organisationid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (groupstatusid) REFERENCES study_group_status (groupstatusid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (personid) REFERENCES person (personid)
    ON UPDATE cascade
    ON DELETE restrict
);

```

```

CREATE TABLE study_group_right_relation (
-- ryhmittelyjen oikeuksien hallinta
-- oikeuksia voi saada henkilö tai ryhmä
--
deleted boolean,
grouprighttypeid integer NOT NULL,
studygroupid integer NOT NULL,
personid integer,
righttypeid integer,
eventgroupid integer,
PRIMARY KEY (grouprighttypeid),
FOREIGN KEY (studygroupid) REFERENCES study_group (studygroupid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (personid) REFERENCES person (personid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (righttypeid) REFERENCES study_right_type (righttypeid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (eventgroupid) REFERENCES eventgroup (eventgroupid)
    ON UPDATE cascade
    ON DELETE restrict
);

```

```

CREATE TABLE study_group_parameter_type (
-- ryhmittelyihin liittyvät erikoiskentät (siis ei kaikille ryhmittelyille
yhteisiä)
--
deleted boolean,
groupparametertypeid integer NOT NULL,
name varchar UNIQUE NOT NULL,
PRIMARY KEY (groupparametertypeid)
);

```

```

CREATE TABLE study_group_parameter(
-- ryhmittelyihin liittyvät erikoiskentät
-- [yhdistää ryhmittelyn kenttään]
--
deleted boolean,
groupparameterid integer NOT NULL,

```

```

studygroupid integer NOT NULL,
groupparametertypeid integer NOT NULL,
PRIMARY KEY (groupparameterid),
FOREIGN KEY (studygroupid) REFERENCES study_group (studygroupid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (groupparametertypeid) REFERENCES study_group_parameter_type
(groupparametertypeid)
    ON UPDATE cascade
    ON DELETE restrict
);

```

```

CREATE TABLE study_group_parameter_translation (
-- ryhmittelyihin liittyvien erikoiskenttien kielikohtaiset arvot
--
deleted boolean,
groupparameterid integer NOT NULL,
languageid integer NOT NULL,
value varchar,
PRIMARY KEY (groupparameterid,languageid),
FOREIGN KEY (languageid) REFERENCES language (languageid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (groupparameterid) REFERENCES study_group_parameter
(groupparameterid)
    ON UPDATE cascade
    ON DELETE restrict
);

```

```

CREATE TABLE study_group_relation (
-- ryhmittelyihin liittyvien kurssien ja aliryhmien yhdistäminen
--
deleted boolean,
grouprelationid integer NOT NULL,
studygroupid integer NOT NULL,
subgroupid integer,
courseid integer,
ordernum integer, -- ryhmittelyn sisäinenjärjestys
PRIMARY KEY (grouprelationid),
FOREIGN KEY (subgroupid) REFERENCES study_group (studygroupid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (studygroupid) REFERENCES study_group (studygroupid)
    ON UPDATE cascade
    ON DELETE restrict,
FOREIGN KEY (courseid) REFERENCES course (courseid)
    ON UPDATE cascade
    ON DELETE restrict
);

```

5.3 'Opiskelusuunnitelmat'

```

CREATE TABLE study_plan_status (
-- Suunnitelmaan liittyvät tilat
--

```

```

deleted boolean,
planstatusid integer,
name varchar UNIQUE NOT NULL,
PRIMARY KEY (planstatusid)
);

CREATE TABLE study_plan (
-- Opiskelusuunnitelmat
-- Perustiedot
--
deleted boolean,
planid integer,
studygroupid integer NOT NULL, -- suunnitelman pohjatieto
personid integer NOT NULL, -- kenen suunnitelma
planstatusid integer NOT NULL, -- suunnitelman tila (lukitukset...)
name varchar, -- suunnitelmalle annettu nimi
modifiedon TIMESTAMP,
active BOOLEAN, -- onko suunnitelma aktiivinen (vain yksi
saisi olla)
PRIMARY KEY (planid),
FOREIGN KEY (studygroupid) REFERENCES study_group (studygroupid)
ON UPDATE cascade
ON DELETE restrict,
FOREIGN KEY (personid) REFERENCES person (personid)
ON UPDATE cascade
ON DELETE restrict,
FOREIGN KEY (planstatusid) REFERENCES study_plan_status (planstatusid)
ON UPDATE cascade
ON DELETE restrict
);

CREATE TABLE study_season (
-- Opiskeluvuoteen liittyvät jaksotukset
--
seasonid integer,
name varchar,
ordernum integer,
deleted boolean,
PRIMARY KEY (seasonid)
);

CREATE TABLE study_unit (
-- Opiskelusuunnitelmaan liittyvät valinnat
-- (tästä haetun tiedon järjestelyn pohjana käytetään suunnitelman pohjatietoa)
--
deleted boolean,
unitid integer,
planid integer,
selected boolean, -- suorituksen liittyvä valinnaisuus (valittu
suoritettavaksi)
planned_year integer, -- suorituksen suunniteltu vuosi
seasonid integer, -- suorituksen suunniteltu kausi
studygroupid integer, -- suorituksen kohde (ryhmittely tai null)
courseinstanceid integer, -- suorituksen kohde, 'vale ilmoittautuminen'
(kurssi-instanssi tai null)
PRIMARY KEY (unitid),

```

```

FOREIGN KEY (planid) REFERENCES study_plan (planid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (seasonid) REFERENCES study_season (seasonid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (studygroupid) REFERENCES study_group (studygroupid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (courseinstanceid) REFERENCES courseinstance (courseinstanceid)
  ON UPDATE cascade
  ON DELETE restrict
);

```

```

CREATE TABLE study_unit_comment_type (
-- kommentteihin liittyvät tyypit, joihin liittyy myös näkyvyys
--
  deleted boolean,
  unitcommenttypeid integer NOT NULL,
  name varchar UNIQUE NOT NULL,
PRIMARY KEY (unitcommenttypeid)
);

```

```

CREATE TABLE study_unit_comment (
-- Suunnitelmiin liittyvät perustelut ja kommentit
--
  deleted boolean,
  unitcommentid integer,
  unitcommenttypeid integer,
  unitid integer,
  personid integer,           -- kenen lisäämä
  value varchar,
  modifiedon TIMESTAMP,      -- milloin, aikajärjestykseen
  parent integer,
PRIMARY KEY (unitcommentid),
FOREIGN KEY (unitid) REFERENCES study_unit (unitid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (unitcommenttypeid) REFERENCES study_unit_comment_type
(unitcommenttypeid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (personid) REFERENCES person (personid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (parent) REFERENCES study_unit_comment (unitcommentid)
  ON UPDATE cascade
  ON DELETE restrict
);

```

```

CREATE TABLE study_plan_right_relation (
-- Suunnitelmiin liittyvät oikeudet
--
  deleted boolean,
  planrightrelationid integer,
  planid integer NOT NULL,

```

```

personid integer,
rightypeid integer NOT NULL,
eventgroupid integer,
PRIMARY KEY (planrightrelationid),
FOREIGN KEY (planid) REFERENCES study_plan (planid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (personid) REFERENCES person (personid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (rightypeid) REFERENCES study_right_type (rightypeid)
  ON UPDATE cascade
  ON DELETE restrict,
FOREIGN KEY (eventgroupid) REFERENCES eventgroup (eventgroupid)
  ON UPDATE cascade
  ON DELETE restrict
);

```

6. Jatkokehityksessä huomioitavaa

- Kottairainen-sovelluksen kyselyjen liittäminen tutkintorakenteeseen
- Korvaavuuksien esittäminen
- Korvaavuus voi koostua useammasta kurssista
- Yliopiston ulkopuolisten tai suunnitteilla olevien lisääminen
- Suunnitelmien virallisuus, hyväksyminen tarkastettava
- Käyttöoikeuksien hallinta
- Opintokokonaisuuksien suoritusten hallinta
- Pohdittiin perustelujen linkityksen tarkentaminen

7. ToDo - suunnittelun alla

- Voiko tutkintorakenteeseen liittyä useampi organisaatio
 - vai riittääkö liittyminen oikeuksien kautta
- onko tutkintorakenteella henkilöä (luojaa/omistajaa)
 - mihin tarvitaan? (oikeudet, oletukset)
 - viite-eheys, jos käyttäjä häviää?
- onko osa suunnitelmiin liittyvistä kommentteista yksityisiä