

Tietokantojen perusteet, suunnitteluprojekti

11.2.2019

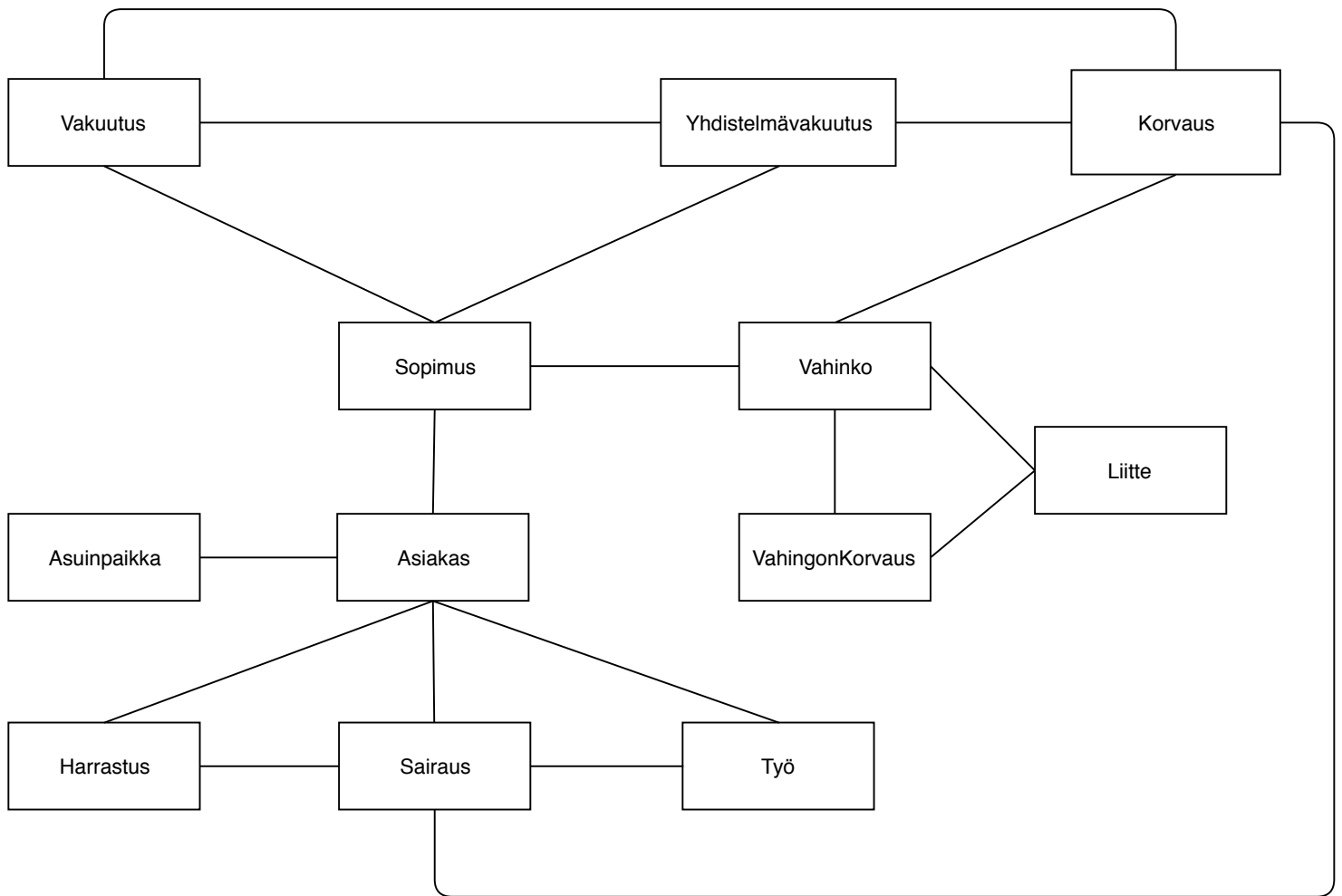
Käsiteanalyysin vaiheet

Tunnista käsite-ehdokkaat

- Vakuutus - erilaisia vakuutuksia
- Yhdistelmävakuutus - vakutuuksien yhdistelmiä
- Asiakas - vakuutusyhtiön asiakas
- Asuinpaikka - asuinpaikka, missä asiakas asuu
- Työ - asiakkaan työpaikka
- Harrastus - asiakkaan harrastukset
- Sopimus - sopimus asiakkaan ja vakuutusyhtiön välillä
- Korvaus - korvauksia, mitkä vakuutusyhtiö korvaa
- Sairaus - asiakkaidenn sairaudet, harrastuksiin ja työhön liittyvät sairaudet
- Vahinko - asiakkaan vahingot, joista hän haluaa saada korvauksen
- VahingonKorvaus - vahingon korvauksia
- Liite - liitteet, jotka liittyvät vahinkoon tai korvauksen loppupäätökseen

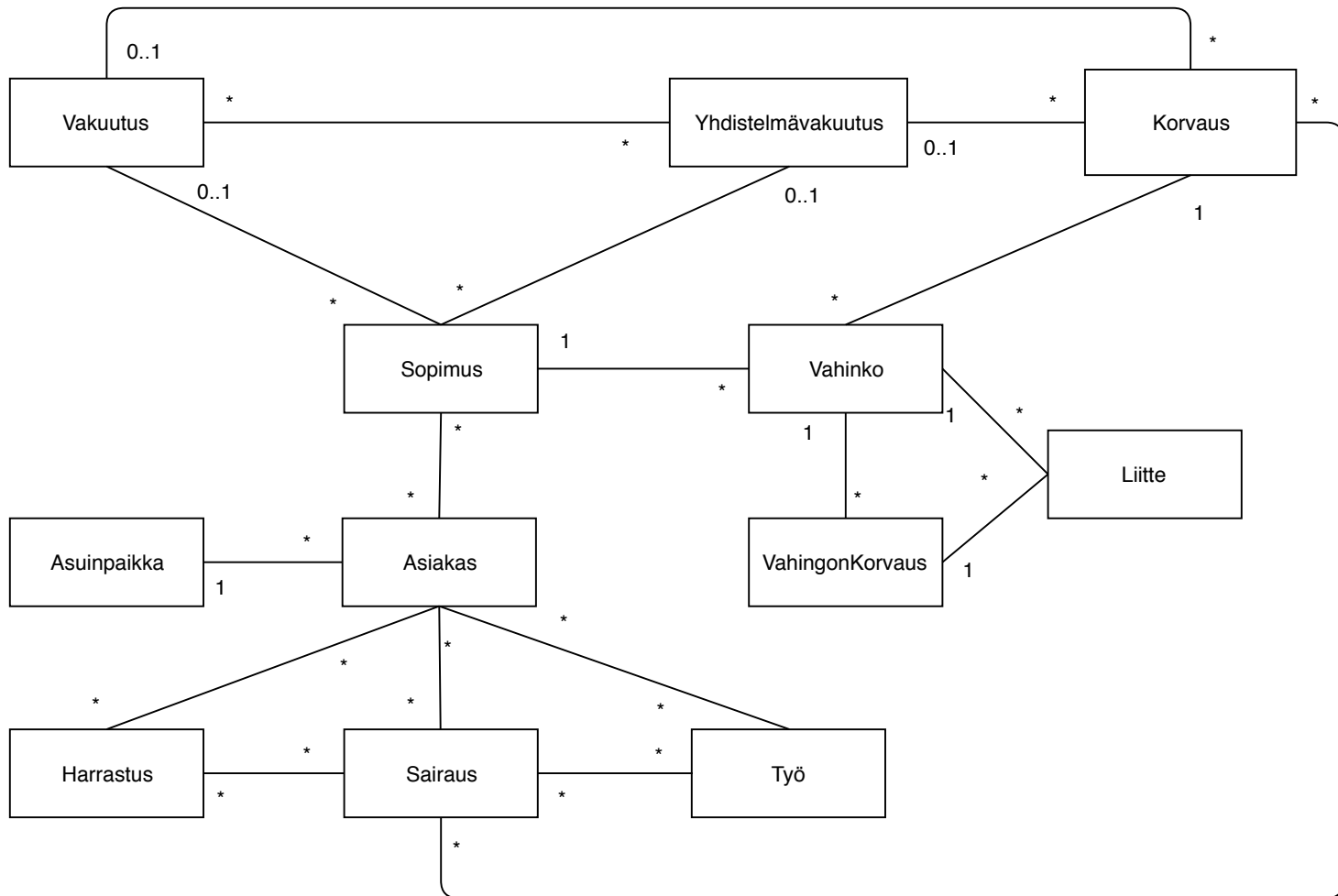
Tunnista käsitteiden väliset yhteydet

- Vakuutuksen liittyy Yhdistelmävakuutus - Yhdistelmävakuutus voi koostua monista vakuutuksista
- Vakuutuksen liittyy Korvaus - vakuutukseen liittyy tiettyjä korvauksia
- Yhdistelmävakuutuksen liittyy Korvaus - yhdistelmävakuutukseen liittyy tiettyjä korvauksia
- Korvaukseen liittyy Vahinko - asiakas pitää raportoida, mihin korvaustapaukseen hänen vahinko liittyy
- Korvaukseen liittyy Sairus - korvauksen kuvauksessa tulisi lukea, mitkä sairaudet vakuutusyhtiö korvaa
- Asiakkaaseen liittyy Asuinpaikka - asiakas antaa tiedot omasta asuinpaikasta
- Asiakkaaseen liittyy Työ - asiakas antaa tiedot omasta työstä
- Asiakkaaseen liittyy Harrastus - asiakas antaa tiedot omista harrastuksista
- Asiakkaaseen liittyy Sairaus - asiakas antaa tiedot omista sairauksista
- Sopimukseen liittyy Asiakas - asiakkalla on sopimuksia vakutusyhtiön kanssa
- Sopimukseen liittyy Vakuutus - sopimus kuuluu tiettyyn vakuutukseen
- Sopimukseen liittyy Yhdistelmävakuutus - sopimus kuuluu tiettyyn yhdistelmävakuutukseen
- Sopimukseen liittyy Vahinko - sopimukseen voi liittyä asiakkaan vahingot
- Harrastukseen liittyy Sairaus- harrastuksien mahdolliset liittyvät sairaudet
- Työhöön liittyy Sairaus - työhön mahdolliset liittyvät sairaudet
- Vahinkoon liittyy VahingonKorvaus - vakuutusyhtiö voi korvata asiakkaan vahingot
- Vahinkoon liittyy Liite - vahinkoon voidaan liittää dokumentit
- VahingonKorvaukseen liittyy Liite - korvaukseen voidaan liittää dokumentit



Tunnista ja määrittele osallistumisrajoitteet

- [Vakuutus] * - * [Yhdistelmävakuutus]
- [Vakuutus] 0..1 - * [Korvaus]
- [Yhdistelmävakuutus] 0..1 - * [Korvaus]
- [Korvaus] 1 - * [Vahinko]
- [Korvaus] * - * [Sairaus]
- [Asiakas] * - 1 [Asuinpaikka]
- [Asiakas] * - * [Työ]
- [Asiakas] * - * [Harrastus]
- [Asiakas] * - * [Sairaus]
- [Sopimus] * - * [Asiakas]
- [Sopimus] * - 0..1 [Vakuutus]
- [Sopimus] * - 0..1 [Yhdistelmävakuutus]
- [Sopimus] 1 - * [Vahinko]
- [Harrastus] * - * [Sairaus]
- [Työ] * - * [Sairaus]
- [Vahinko] 1 - * [VahingonKorvaus]
- [Vahinko] 1 - * [Liitte]
- VahingonKorvaus 1 - * [Liitte]



Tunnista attribuutit ja lisää ne käsitteille

Vakuutus
nimi: String
kuvaus: String
hinta: Double

Korvaus
nimi: String
kuvaus: String
omavastu: Double
max_hinta: Double

Asiakas
nimi: String
sukunimi: String
email: String
phone: String
iban: String

Asuinpaikka
osoite: String
postinumero: Integer
kaupunki: String
asuinneliot: Double
rakennuksen_tyyppi: String
rakennusvuosi: Integer
irtaimiston_arvo: Integer

Yhdistelmävakuutus
nimi: String
kuvaus: String
hinta: Double

Harrastus
nimi: String

Sairaus
nimi: String

Liite
nimi: String
tiedosto: Blob
pvm: Date

Työ
nimi: String
tyyppi: String
kuvaus: String

VahingonKorvaus
kuvaus: String
pvm: Date
hinta:Double

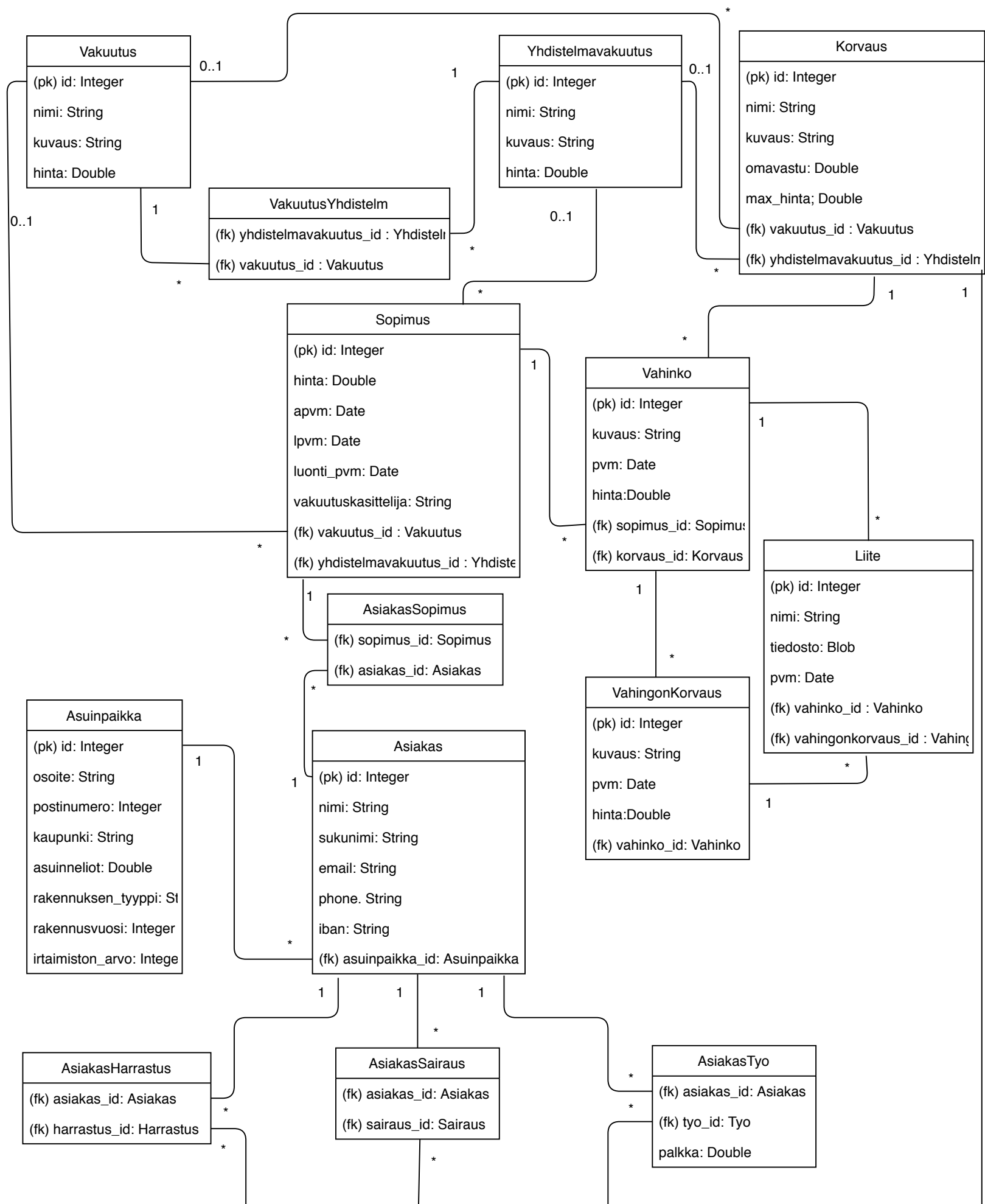
Sopimus
hinta: Double
apvm: Date
lpvm: Date
luonti_pvm: Date
vakuutuskasittelija: String

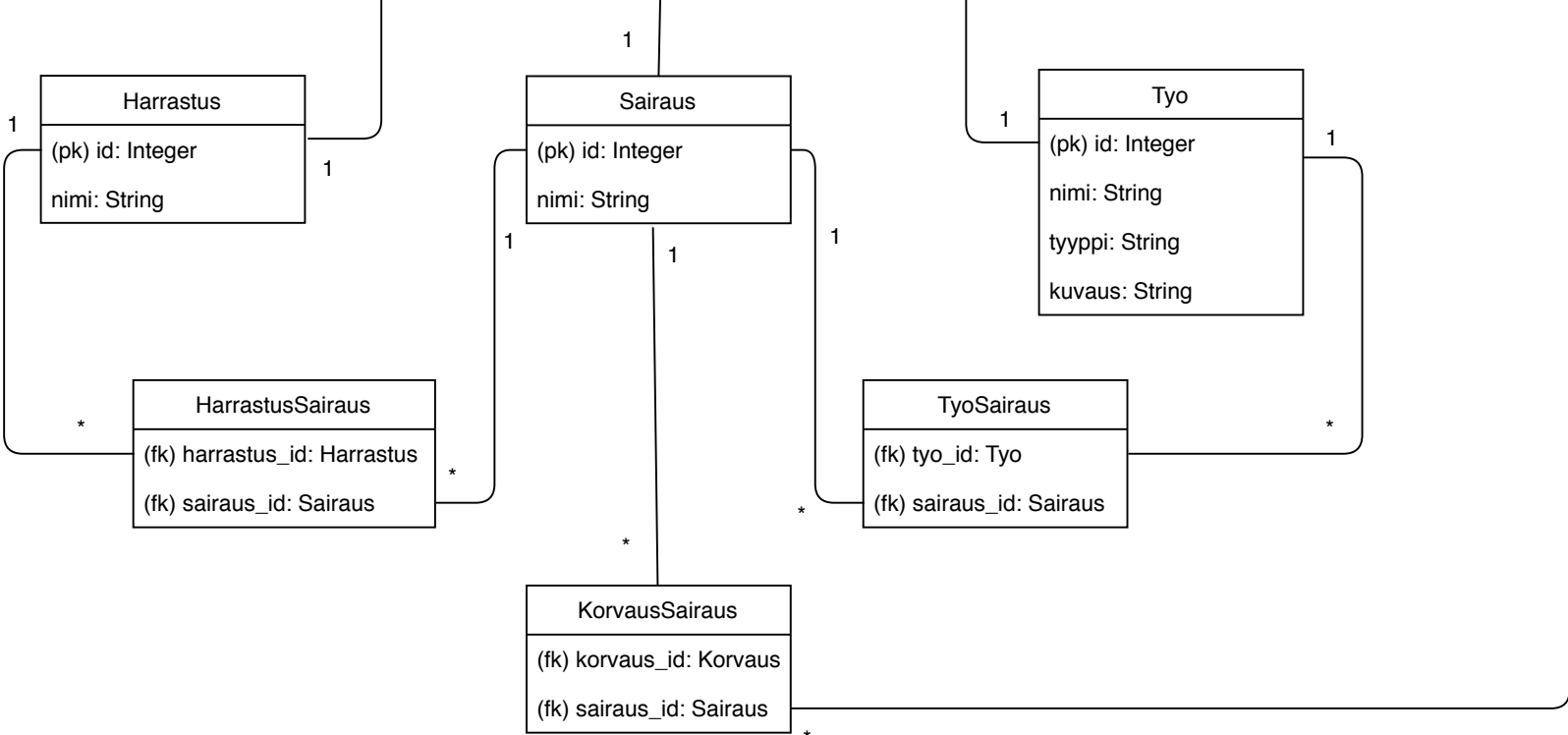
Vahinko
kuvaus: String
pvm: Date
hinta:Double

Yleistä ja eriytä käsitteitä

Yli- ja aliluokkien etsintä kannattaa tehdä matriisina, missä jokaista käsitettä verrataan jokaiseen muuhun käsitteeseen. Käymällä läpi käsitteemme, huomaamme, ettei niissä ole yli- tai aliluokille tarvetta, siksi tässä tapauksessa skippataan matriisi vaihe.

Tietokantakaavio





SQL:n CREATE TABLE

```
1. CREATE TABLE Vakuutus(
    id INTEGER PRIMARY KEY,
    nimi VARCHAR(255) NOT NULL,
    kuvaus VARCHAR NOT NULL,
    hinta FLOAT
);
2. CREATE TABLE Asiakas(
    id INTEGER PRIMARY KEY,
    nimi VARCHAR(255) NOT NULL,
    sukunimi VARCHAR(255) NOT NULL,
    email VARCHAR NOT NULL,
    phone VARCHAR NOT NULL,
    iban VARCHAR NOT NULL,
    asuinpaikka_id INTEGER NOT NULL,
    FOREIGN KEY(asuinpaikka_id) REFERENCES Asuinpaikka(id)
);
3. CREATE TABLE Sopimus(
    id INTEGER PRIMARY KEY,
    hinta FLOAT NOT NULL,
    apvm DATE NOT NULL,
    lpvm DATE NOT NULL,
    luonti_pvm DATE NOT NULL,
    vakuutuskasittelija VARCHAR(255) NOT NULL,
    vakuutus_id INTEGER,
    yhdistelmavakuutus_id INTEGER,
    FOREIGN KEY(vakuutus_id) REFERENCES Vakuutus(id),
    FOREIGN KEY(yhdistelmavakuutus_id) REFERENCES Yhdistelmavakuutus(id),
    CONSTRAINT yksi_foreign_key
        CHECK((vakuutus_id IS NULL OR yhdistelmavakuutus_id IS NULL)
            AND NOT (vakuutus_id IS NULL AND yhdistelmavakuutus_id IS NULL))
);
```

Vakuutus
(pk) id: Integer
nimi: String
kuvaus: String
hinta: Double

Asiakas
(pk) id: Integer
nimi: String
sukunimi: String
email: String
phone: String
IBAN: String
(fk) asuinpaikka_id: Asuinpaikka

Sopimus
(pk) id: Integer
hinta: Double
apvm: Date
lpvm: Date
luonti_pvm: Date
vakuutuskasittelija: String
(fk) vakuutus_id : Vakuutus
(fk) yhdistelmavakuutus_id : Yhdiste

Keskeisiä käyttötapauksia

1. Kysely millä pystyt tarkista paljon voimassolevia sopimuksia vakuutusyhtiöllä on nyt käynnissä. CURDATE() funktio toimii MySQL:ssa. Erilaisissa tietokantajärjestelmissä löytyy vastavia funktioita. SQL:

```
SELECT * FROM Sopimus
WHERE apvm <= CURDATE() AND lpvm >= CURDATE();
```

2. Kysely millä pystyt tarkistaa onko sopimus tuonnut voittoa vakuutusyhtiölle SQL:

```
SELECT Sopimus.id, MAX(Sopimus.hinta) as saatu_rahat,
SUM(VahingonKorvaus.hinta) as maksettu_rahat,
CASE WHEN saatu_rahat > maksettu_rahat THEN 1
ELSE 0 END AS ollaanko_voitossa FROM Sopimus
INNER JOIN Vahinko ON Sopimus.id = Vahinko.sopimus_id
INNER JOIN VahingonKorvaus ON Vahinko.id = VahingonKorvaus.vahinko_id
WHERE Sopimus.id = 1
GROUP BY Sopimus.id;
```

3. Kysely, missä näkyy minkälaisia korvauksia vakuutusyhtiö on maksanut ammatteihin, mikä työ tyyppi on istumatyö SQL:

```
SELECT Tyo.nimi, SUM(VahingonKorvaus.hinta) AS maksettu_rahaa FROM Tyo
INNER JOIN TyoSairaus ON Tyo.id = TyoSairaus.tyo_id
INNER JOIN Sairaus ON TyoSairaus.sairaus_id = Sairaus.id
INNER JOIN KorvausSairaus ON Sairaus.id = KorvausSairaus.sairaus_id
INNER JOIN Korvaus ON KorvausSairaus.korvaus_id = Korvaus.id
INNER JOIN Vahinko ON Korvaus.id = Vahinko.korvaus_id
INNER JOIN VahingonKorvaus ON Vahinko.id = VahingonKorvaus.vahinko_id
WHERE Tyo.tyyppi = 'istumatyö'
GROUP BY Tyo.nimi
ORDER BY Tyo.nimi;
```

Ongelmat

Varmasti ongelmia löytyy minun tietokanta mallista. Tässä on muutaman niistä:

1. Emme pysty lasketa mitä ikinen sairaus maksaa vakuutusyhtiölle. Sitä voisi nopeasti korjata jos yhdistää taulu Vahiko ja Sairaus (voisi olla että se ei ole hyvä ratkaisu, koska sitten taulujen yhteys Vahinko ja Korvaus on merkityksetön)
2. Perjääteässä olisi hyvä pitää myöskin Henkilokunta taulu ja pitää siellä vakuutusyhtiön työntekijöiden data. Se myös otta tyhjän sarakkeen "vakuutuskasittelija" Sopimus taulusta ja vain viitätä Henkilokunta tauluun.

P.S.

Sorry for my Finnish language skills, I have tried my best. Could have done the project better, but since there was not enough time for me I ended up with this solution, but still proud for what I did :)

I felt that I am having a lack of knowledge on how the insurance company really works. So, this is version of how does the insurance company works in my world :)

I did all my work through the draw.io so it is not that easy here to change font weight and looks like font Arial is missing here, so sorry for that as well. I am not a student, I am already working at the software company as a programmer and don't have much time, so that is why I have skipped some stuff because I didn't found them necessary especially in the Käsitemallit section.