Tietokantojen perusteet

Harjoitustyö – Tietokannan suunnittelu

Lappeenrannan teknillinen yliopisto
Tietotekniikan koulutusohjelma

Tietokantojen perusteet
Kesä 2019

AK-uni-git

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO 1
1 MÄÄRITYS 2
2 MALLINNUS 3
2.1 Käsitemalli 3
2.2 Relaatiomalli 3
3 TIETOKANTATOTEUTUS 5

1 MÄÄRITYS

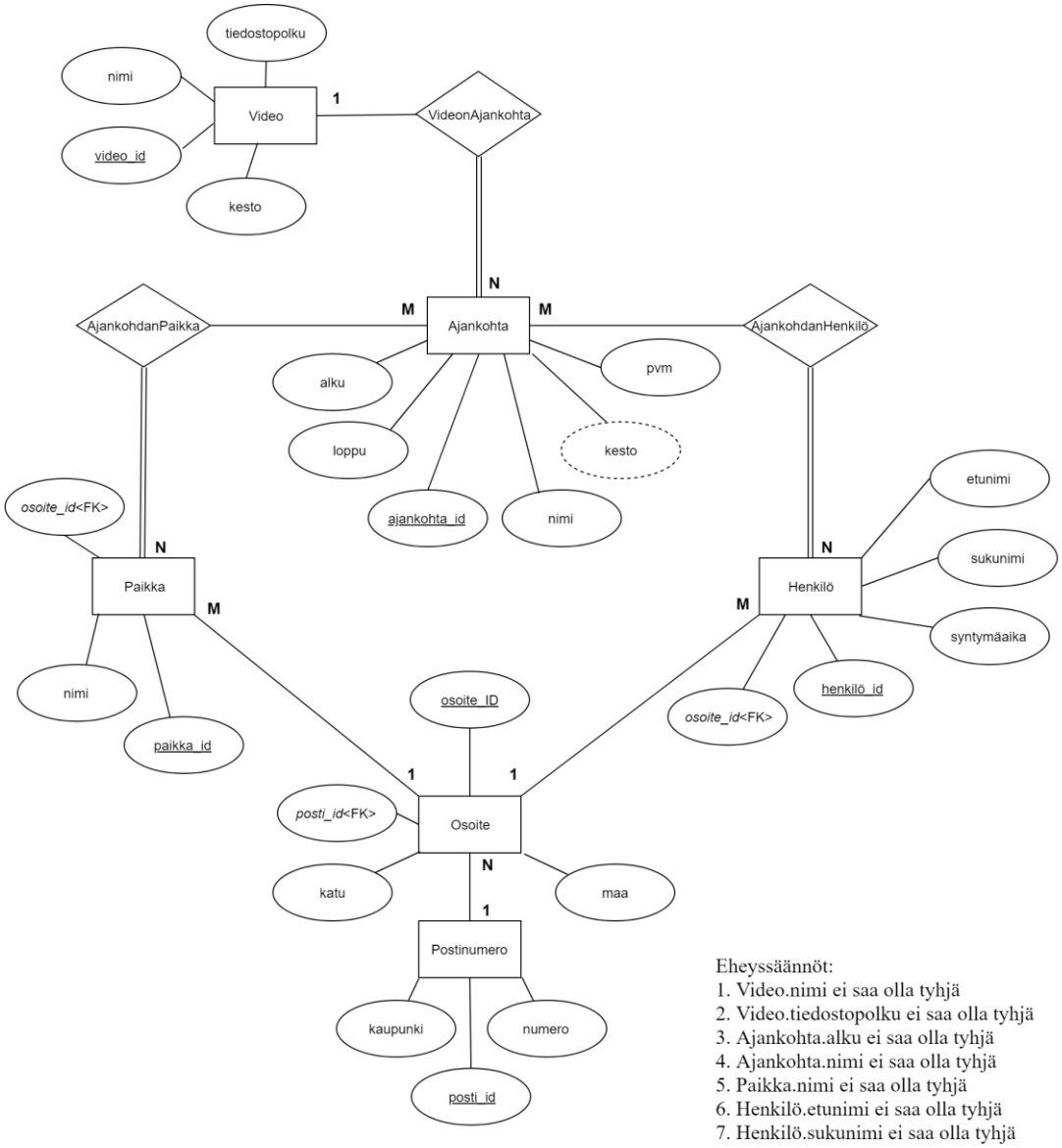
VHS-digitointia suunniteltaessa huomattiin, että videoille olisi hyvä luoda tietokanta, sillä videot on tarkoitus digitoida koko kasetti kerrallaan ja videot ovat täynnä erilaisia tapahtumia eri ajankohdilla. Tietokannassa olisi videon nimi, jolla voi olla ajankohtia, joilla voisi taas olla paikkoja ja henkilöitä, joilla voi olla osoitteet. Videolla on nimi, tiedostopolku ja tunnus. Ajankohdalla on nimi, ajankohtaan liittyvä päivämäärä, tunnus, kesto ja aikaleima aloitus- ja päättymiskohdasta. Paikalla on nimi ja osoite. Henkilöllä on nimi, syntymäaika ja osoite. Osoite on lisäksi jaettu osoite ja postinumero -tauluun.

Tietokannan käyttöliittymäksi toteutetaan python pohjainen komentorivikäyttöliittymä, jolla voidaan ohjelman prototyyppitasolla lisätä videoita, päivittää henkilön tiedot, visualisoida eri videoiden ajankohtien lukumäärää ja toteuttaa seuraavia kyselyitä: (1) Listaa kaikkien videoiden tiedot. (2) Listaa videoiden kaikki ajankohdat. (3) Listaa henkilöiden osoitteet. (4) Etsi kaikki ajankohdat tietyn paikan mukaan (5) Etsi kaikki ajankohdat tietyn henkilön mukaan.

2 MALLINNUS

2.1 Käsitemalli

Toteuttamalla määritys vaiheen ominaisuudet saadaan seuraavanlainen ER-käsitekaavio:



Eheyssäännöt:
1. Video.nimi ei saa olla tyhjä
2. Video.tiedostopolku ei saa olla tyhjä
3. Ajankohta.alku ei saa olla tyhjä
4. Ajankohta.nimi ei saa olla tyhjä
5. Paikka.nimi ei saa olla tyhjä
6. Henkilö.etunimi ei saa olla tyhjä
7. Henkilö.sukunimi ei saa olla tyhjä

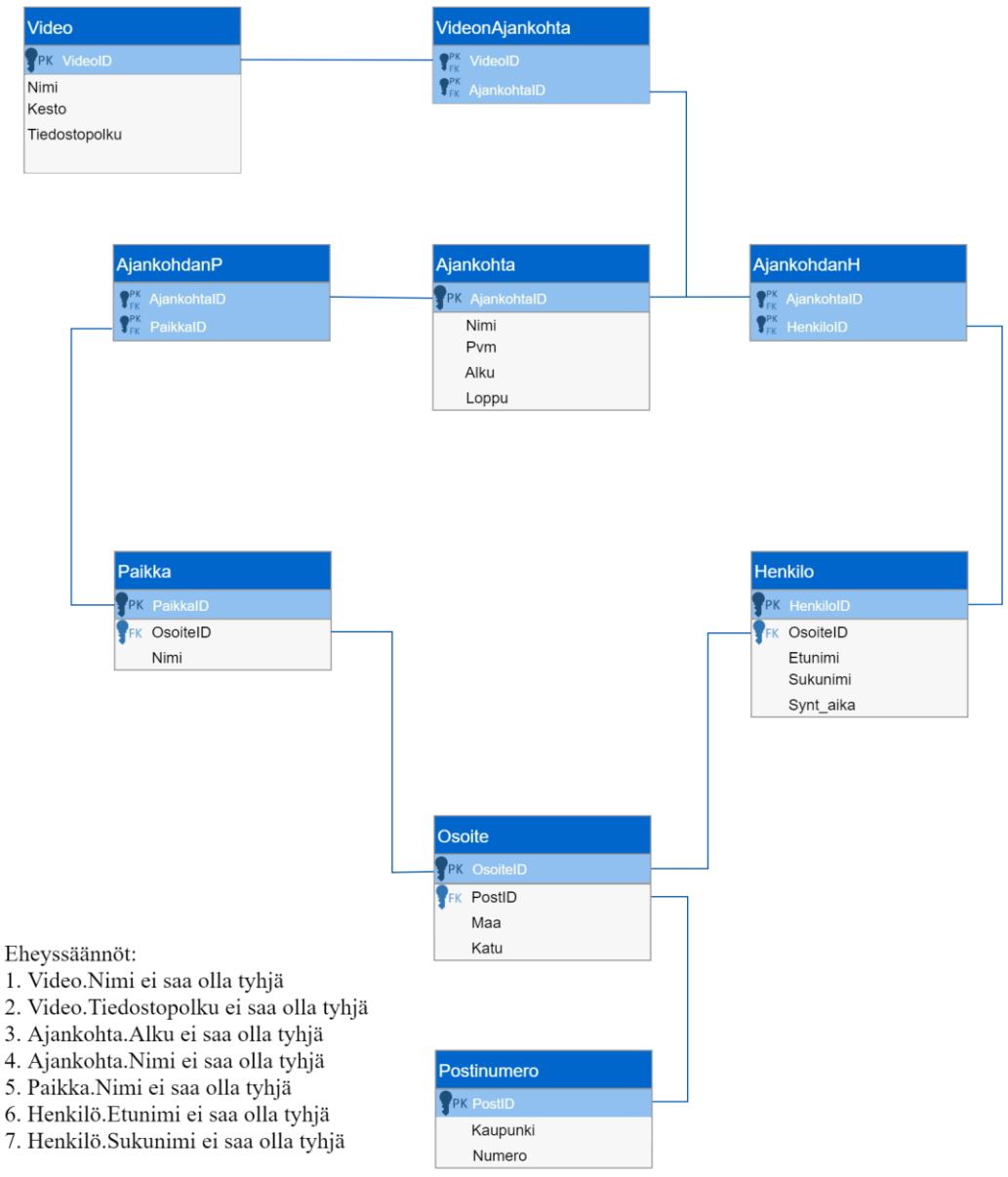
2.2 Relaatiomalli

ER-mallin transformointiin relaatiomalliksi tarvittiin seuraavat säännöt luentodioista:

- Sääntö 1: Jokaisesta kohdetyypistä tehdään oma kohderelaation. Tavallisista attribuuteista tehdään relaation attribuutteja. Avainattribuuteista muodostetaan perusavain.

- Sääntö 3: Jokainen johdettu attribuutti hylätään.
- Sääntö 6: Jokaisesta binäärisestä 1:1-suhdetyypistä sijoitetaan toisen kohdetyypin K1 avainattribuutit viiteavaimeksi toisesta kohdetyypistä K2 muodostettuun relaatioon.
- Sääntö 7: Jokaisesta binäärisestä 1:N-suhdetyypistä sijoitetaan 1:n puoleisen kohdetyypin avainattribuutit viiteavaimeksi N:n puoleisesta kohdetyypistä muodostettuun relaatioon.
- Sääntö 8: jokaisesta binäärisestä N:M-suhdetyypistä muodostetaan oma, ns. suhderelaatio. Relaation attribuuteiksi valitaan suhdetyypin mahdolliset attribuutit ja perusavaimeksi N:M-suhteeseen liittyvien kohdetyyppien avainattribuutit.

Lopputuloksena saadaan seuraavanlainen vapaasti UML-mallinnettu kaavio:



Eheyssäännöt:
1. Video.Nimi ei saa olla tyhjä
2. Video.Tiedostopolku ei saa olla tyhjä
3. Ajankohta.Alku ei saa olla tyhjä
4. Ajankohta.Nimi ei saa olla tyhjä
5. Paikka.Nimi ei saa olla tyhjä
6. Henkilö.Etunimi ei saa olla tyhjä
7. Henkilö.Sukunimi ei saa olla tyhjä

3 TIETOKANTATOTEUTUS

Yleistä: Käsitemallin toteutuksessa relaatiomalliksi havaittiin, että osoite ja postinumero yksilötyyppien 1:N suhde oli merkitty väärinpäin, joten käänsin sen oikein päin ja korjasin sen sekä ER-malliin, että relaatiomalliin.

Tietokantaa voitaisiin nopeuttaa käyttämällä indeksejä. Erityisesti Ajankohta-taulun AjankohtaID-avainta käytetään useassa liitoksessa, joten se olisi hyvä indeksoida, jos tietoa on paljon. Lisäksi Henkilo-taulun Etunimiä ja Sukunimiä käytetään henkilön nimen perusteella hakemisessa ja henkilön tietojen päivittämiseen, joten ne olisi myös hyvä indeksoida, mikäli tietokannan koko kasvaa suureksi.

Toteutustapa: Kirjoitin yksinkertaisen valikko pohjaisen Python-ohjelman, joka kertoi toiminnot, josta käyttäjä valitsi haluamansa toiminnon numerolla. Ohjelman toteutus oli aika helppo, sillä koirakoulu harjoituksesta oli opittu kaikki, mitä tietokantojen käsittelyyn Pythonilla tarvitaan. Koska kyseessä on vain niin sanottu ”proof of concept” on tietokannan nimi (”HT_test.sqlite3”) kovakoodattu itse ohjelmaan ja sitä ei tässä versiossa voi vaihtaa. Itse harjoitustyön ohjelman monimutkaisuus oli huomattavasti helpompi kuin Python kurssin tavoitetason, joten Python ohjelmointi ei tuottanut vaikeuksia. Tietenkin minun tarvitse varmistaa muutaman asia nettilähteistä Pythoniin liittyen, sillä Python kurssista oli jo puoli vuotta.

Testatessa huomioitavaa: Käyttäjä voi syöttää virheellisiä tietoja tietokantaan, niin halutessaan. Jos käyttäjä valitsee ” 6) Muokkaa henkilön tietoja” ja syöttää roskaa henkilön tiedoiksi, kaikki menevät sellaisenaan tietokantaan. Tähän voisi kehitellä virheentarkistuksen jatkoversiossa, mutta ”proof of concept” vaiheessa päädyin tarkistamaan vain, että syntymäaika syöte on numero ja kaikki kolme arvoa löytyy. En tarkistanut ovatko ne oikean pituisia muotoon ’DD-MM-YYYY’ eli kaksi, kaksi ja neljä -merkkiä.