

# Viserrin - Tietokantasuunnitelma

Tietokantojen perusteet  
Kesä 2014

Jarno Louhelainen  
jarno.louhelainen@helsinki.fi  
013448510

# Viserrin:

"Tietojenkäsittelytieteen laitos on suunnitellut uuden modernin sosiaalisen median alustan. Käyttäjät luovat *Visertimeen* uuden käyttäjätunnuksen ja voivat kirjautuneena julkaista viserryksiä. Käyttäjätunnukset ovat uniikkeja. Viserrykset sisältävät rajoitetun pituisen viestin (141 merkkiä) ja ajankohdan jolloin viserrys julkaistiin. Mobiililaitteita käytettäessä viserrykseen liittyy myös lokaatitieto. Järjestelmä liittää lokaatitietoon paikannimen koordinaattien perusteella.

Viserryksiin voi liittyä myös vapaasti muotoiltavia avainsanoja eli risuaitoja, joilla käyttäjät voivat tarkentaa viserrystensä aihealueita. Käyttäjä voi myös liittää viserrykseen useampia kuvia. Käyttäjä voi asettaa yhden viserryksen itselleen suosikiksi, joka näkyy myös käyttäjän profiilisivulla.

Kaikki käyttäjät, myös ne, jotka eivät ole rekisteröityneet voivat lukea käyttäjien viserryksiä. Rekisteröityneet käyttäjät voivat seurata muita visertelijöitä. Rekisteröityneistä käyttäjistä halutaan tallentaa kaikki oleelliset tiedot: nimi, sähköpostiosoite, ikä ja niin edelleen."

Tehdään kuvauksen perusteella Visertimestä tietosisältökartoitus ja UML-käsitekaavio, jonka jälkeen lähemme muodostamaan tietokantakaaviota ja SQL-lauseita joilla tämän tietokannan taulut voidaan muodostaa. Seuraavaksi tehdään Boyce-Codd-riippuvuusanalyysi, ja esitetään muutama esimerkkitaulu ja esimerkkejä käyttötapauksista.

# Tietosisältökartoitus

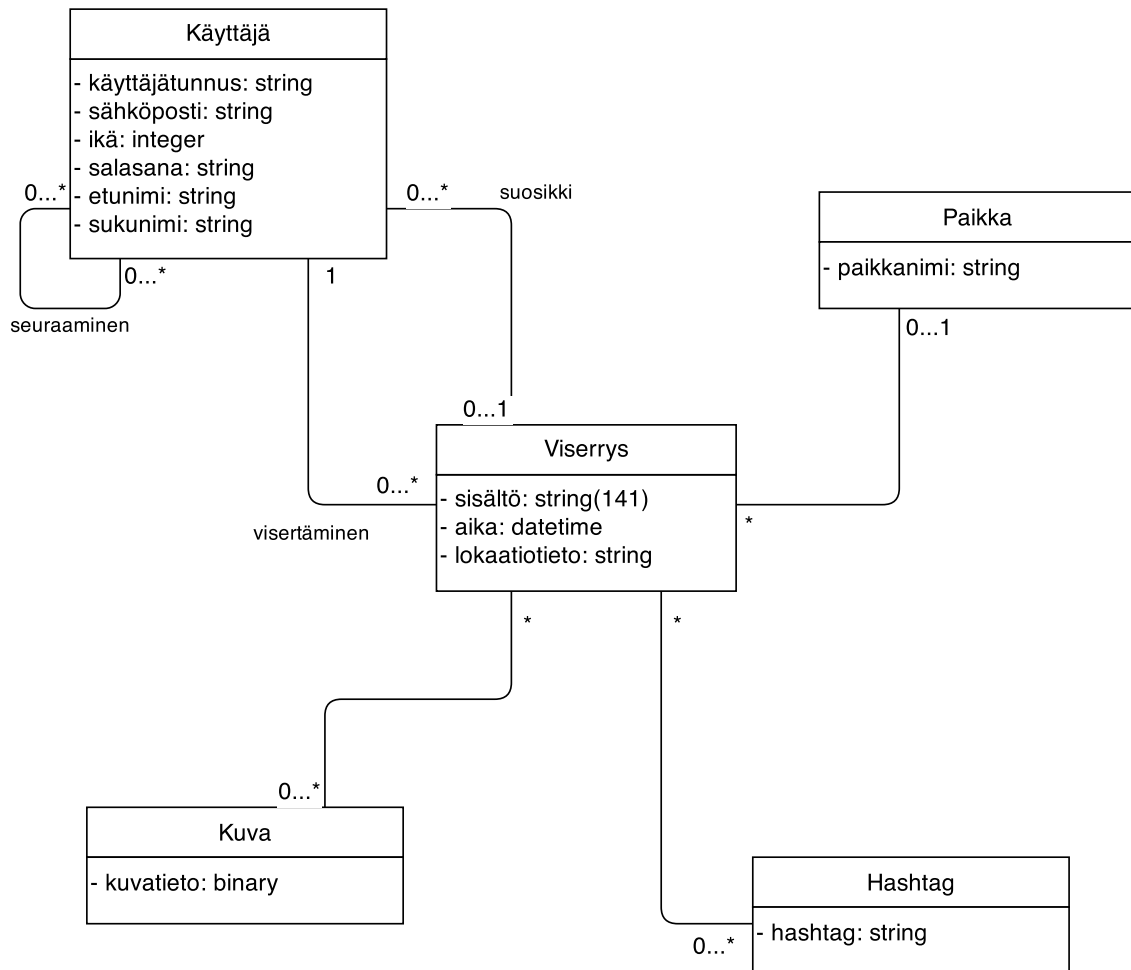
## Rekisteröitynyt käyttäjä

- Käyttäjätunnus - Oltava uniikki
- Sähköposti
- Ikä
- Salasana
- Etunimi
- Sukunimi
- Suosikki - Suosikkiviserrys
- Seuraajat - Muut käyttäjät voivat seurata

## Viserrys

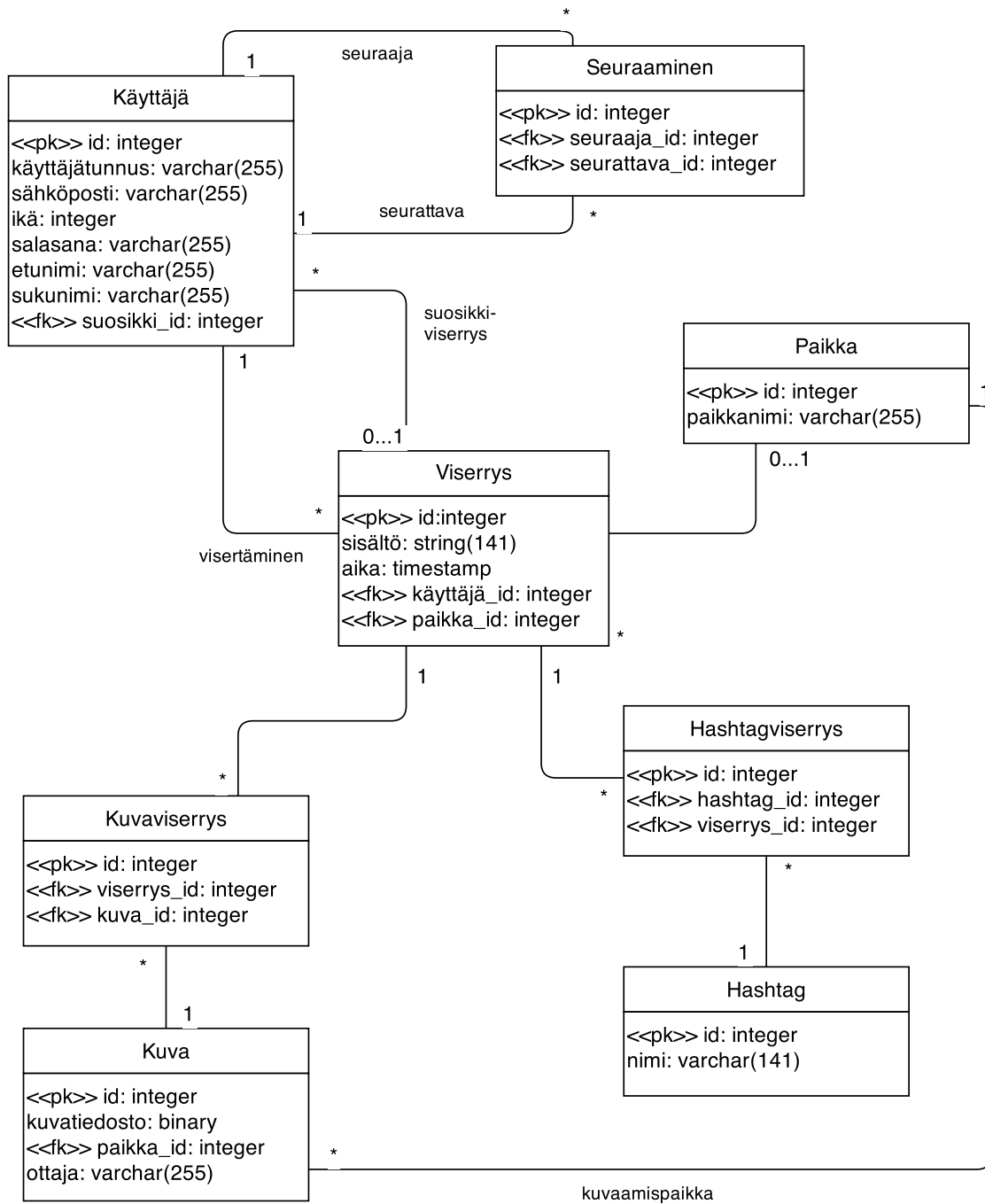
- Lukea voi rekisteröitymättä, lähettää ei
- Sisältö - Enintään 141 merkkiä
- Aika
- Lokaatiotieto - Jos tehty mobiililaitteella
- Kuva - Voi liittää useampia kuvia
- Hashtag - Liittää viserrykset yhteen

# UML-Käsitekaavio



Tästä käsitekaaviosta tulee ilmi Visertimen rakenne. Rekisteröityneet käyttäjän voivat visertää, eli lähettää viserryksiä palveluun. Jokainen käyttäjä voi halutessaan seurata muita käyttäjiä, ja valita profiilisivulleen yhden suosikki-viserryksen. Viserryksiin voi liittää halutessaan yhden tai useamman kuvan, jotka tallennetaan omaan tietokantaansa, ja/tai yhden tai useamman hashtagin, eli avainsanan, joille on oma tietokantansa. Jos viserrys tehdään mobiililaitella, tallennetaan myös lokaatio-tiedot, joiden avulla haetaan erillisestä tietokannasta paikannimi.

# Tietokantakaavio



Tietokanta sisältää käyttäjän tiedot sisältävän taulun, joka yhdistyy viserryksiin ja käyttäjien muiden käyttäjien seuraamisen ilmaisevaan tauluun. Mobiililaitteilla tehtyihin viserryksiin liitetään paikka\_id:n avulla paikannimi, ja käyttäjä voi liittää viserryksiinsä kuvia ja/tai hashtagia, jotka haetaan tietokannasta aputaulujen avustuksella. Tietokannassa sijaitseviin kuviin voidaan liittää metatietona kuvan ottaja ja paikka\_id:n avulla haettu kuvauspaikan nimi. Viserryksien maksimipituus on 141 merkkiä, joten myöskään hashtagit eivät voi olla tätä pidempiä.

Seuraavaksi tietokanta on luotava SQL-komentojen avulla. Jokaisella luotavalla attribuutilla on tyyppi (integer, varchar eli string, binary, datetime), ja näihin voi liittää ehtoja (not null, unique), ja nämä voivat olla joko pää- tai viiteavaimia (primary key, foreign key). Luodaan tietokantakaavion mukaiset taulut:

```
CREATE TABLE käyttäjä (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  käyttäjänimi VARCHAR(255) NOT NULL UNIQUE,  
  sähköposti VARCHAR(255) NOT NULL,  
  ikä INTEGER NOT NULL,  
  salasana VARCHAR(255) NOT NULL,  
  etunimi VARCHAR(255) NOT NULL,  
  sukunimi VARCHAR(255) NOT NULL,  
  suosikki_id INTEGER,  
  FOREIGN KEY (suosikki_id) REFERENCES viserrys(id),  
);
```

```
CREATE TABLE seuraaminen (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  seuraaja_id INTEGER,  
  seurattava_id INTEGER,  
  FOREIGN KEY (seuraaja_id) REFERENCES käyttäjä(id),  
  FOREIGN KEY (seurattava_id) REFERENCES käyttäjä(id),  
);
```

```
CREATE TABLE viserrys (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  sisältö VARCHAR(141) NOT NULL,  
  aika DATETIME NOT NULL,  
  käyttäjä_id INTEGER,  
  paikka_id INTEGER,  
  FOREIGN KEY (käyttäjä_id) REFERENCES käyttäjä(id),  
  FOREIGN KEY (paikka_id) REFERENCES paikka(id),  
);
```

```
CREATE TABLE paikka (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  paikkanimi VARCHAR(255) NOT NULL UNIQUE,  
);
```

```
CREATE TABLE kuva (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  kuvatiedosto BINARY NOT NULL UNIQUE,  
  ottaja VARCHAR(255),  
  paikka_id INTEGER,  
  FOREIGN KEY (paikka_id) REFERENCES paikka(id),  
);
```

```
CREATE TABLE kuvaviserrys (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  kayttaja_id INTEGER,  
  kuva_id INTEGER,  
  FOREIGN KEY (kayttaja_id) REFERENCES kayttaja(id),  
  FOREIGN KEY (kuva_id) REFERENCES kuva(id),  
);
```

```
CREATE TABLE hashtag (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  nimi VARCHAR(141) NOT NULL UNIQUE,  
);
```

```
CREATE TABLE hashtagviserrys (  
  id INTEGER PRIMARY KEY NOT NULL UNIQUE,  
  hashtag_id INTEGER,  
  viserrys_id INTEGER,  
  FOREIGN KEY (hashtag_id) REFERENCES hashtag(id),  
  FOREIGN KEY (viserrys_id) REFERENCES viserrys(id),  
);
```

# Boyce-Codd-riippuvuusanalyysi:

Listataan jokaisen attribuutin (lukuun ottamatta pää- ja viiteavaimia) funktionaaliset riippuvuudet riippuvuusanalyysia varten:

## Käyttäjä

id <- käyttäjätunnus  
id <- sähköposti  
id <- ikä  
id <- salasana  
id <- etunimi  
id <- sukunimi

## Viserrys

id <- sisältö  
id <- aika

## Paikka

id <- paikkanimi

## Kuva

id <- kuvatiedosto  
id <- ottaja

## Hashtag

id <- nimi

Kuten näemme, kaikki attribuutit ovat riippuvuussuhteessa vain pääavaimiinsa.

## Esimerkkitauluja

### Käyttäjä

id	käyttäjätunnus	sähköposti	ikä	salasana	etunimi	sukunimi	suosikki_id
285	mattimies	<a href="mailto:matti@maili.fi">matti@maili.fi</a>	26	asdfniwoG3	Matti	Mattila	763
529	kulkuri69	<a href="mailto:taunoilee@sposti.fi">taunoilee@sposti.fi</a>	19	taunorulesok	Tauno	Taunola	326
652	xhleje	<a href="mailto:lissu.lissula@maili.fi">lissu.lissula@maili.fi</a>	22	zalazana	Liisa	Liisala	NULL
167	minavaan	<a href="mailto:minavaan@sposti.fi">minavaan@sposti.fi</a>	55	maria	Pasi	Pasila	NULL

### Seuraaminen

id	seuraaja_id	seurattava_id
235	285	529
735	285	652
437	529	285
745	167	652

### Viserrys

id	sisältö	aika	käyttäjä_id	paikka_id
326	#lunta #kesällä ?!	2014-06-17 10.58.16	285	646
236	Oli just paras #burgeri!	2014-05-27 18.38.04	167	NULL
854	Onx tääl joQ?	2014-05-03 02.29.26	652	237
474	Nyt #baanalle!	2014-04-27 20.15.52	652	237

### Paikka

id	paikkanimi
273	Helsinki
646	Porvoo
237	Vantaa
347	Oulu

### Kuva

id	kuvatiedosto	paikka_id	ottaja
239	010010111000111...	237	Liisa Liisala
735	010011110111001...	646	Matti Mattila
854	110110100111010...	237	Liisa Liisala
574	101100010000111...	NULL	NULL

### Kuvaviserrys

id	viserrys_id	kuva_id
295	326	735
175	474	239
845	474	854
346	645	855

### Hashtag

id	nimi
627	#lunta
742	#kesällä
234	#burgeri
643	#baanalle!

### Hashtagviserrys

id	hashtag_id	viserrys_id
548	627	326
547	742	326
832	234	236
164	643	474



## Esimerkkejä käyttötapauksista:

Käyttäjä minavaan tekee uuden viserryksen:

```
INSERT INTO Viserrys
  (sisältö, aika, käyttäjä_id)
VALUES
  ("Lähtekö joku kaljalle?", "2014-06-18 18.09.56", 167);
```

Uusi käyttäjä rekisteröityy:

```
INSERT INTO Käyttäjä
  (käyttäjätunnus, sähköposti, ikä, salasana, etunimi, sukunimi)
VALUES
  ("maiija19", "maiija.maijala@sposti.fi", 19, "kissatonkivoi", "Maija", "Maijala");
```

Käyttäjä maiija19 lisää itselleen suosikkiviserryksen:

```
UPDATE Käyttäjä
  SET suosikki_id = 326
  WHERE käyttäjätunnus = "maiija19";
```

Haetaan kaikkia viserryksiä, jotka on tehty Espoossa:

```
SELECT *
  FROM Viserrys
 WHERE paikka_id = (
   SELECT id
     FROM Paikka
    WHERE paikkanimi = "Espoo"
 );
```

## Omia pohdintoja:

Suunnittelimme tietokantamme jokaisen taulun käyttämään erillistä numeerista id:tä, mutta tämä ei ole kaikissa tapauksissa tarpeen. Käyttäjän käyttäjätunnus voisi toimia erittäin hyvin tämän pääavaimena, koska järjestelmän vaatimuksena on käyttäjätunnusten uniikkisuus. Myös seuraamissuhteet muodostavat sekä viserrykset kuviin ja hashtageihin liittävät aputaulut voisivat käyttää pääavaimenaan yhdistelmäavainta, joka muodostuisi näissä olevien viiteavaimien yhdistelmästä, koska tämä riittäisi erottelemaan eri monikot toisistaan.