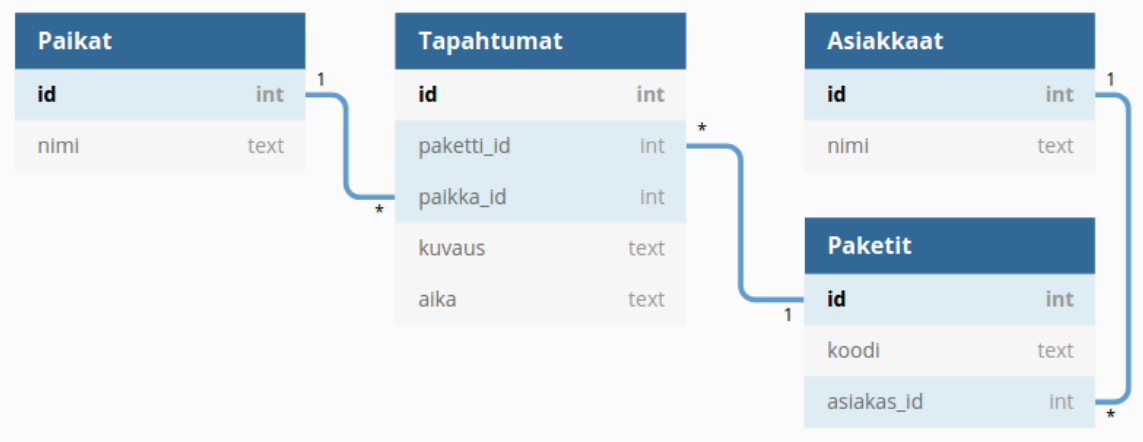
Harjoitustyön tekijä

Antti Laaksonen / 012345678 / ahslaaks@cs.helsinki.fi

Mitä toimintoja on toteutettu?

Harjoitustyössä on toteutettu kaikki vaaditut toiminnot. Lisäksi sovelluksessa on ylimääräinen toiminto “99”, joka sulkee sovelluksen.

Tietokantakaavio



SQL-skeema

```
CREATE TABLE Paikat (id INTEGER PRIMARY KEY, nimi TEXT UNIQUE);
CREATE TABLE Asiakkaat (id INTEGER PRIMARY KEY, nimi TEXT UNIQUE);
CREATE TABLE Paketit (id INTEGER PRIMARY KEY, koodi TEXT UNIQUE, asiakas_id
INTEGER REFERENCES Asiakkaat);
CREATE TABLE Tapahtumat (id INTEGER PRIMARY KEY, paketti_id INTEGER REFERENCES
Paketit, paikka_id INTEGER REFERENCES Paikat, kuvaus TEXT, aika TEXT);
```

Tehokkuustestin tulokset

Ilman indeksejä:

Vaihe 1: 0.02752375602722168 s
Vaihe 2: 0.025425434112548828 s
Vaihe 3: 0.032929420471191406 s
Vaihe 4: 4.179397821426392 s
Vaihe 5: 0.07545161247253418 s
Vaihe 6: 73.66464495658875 s

Indeksien kanssa:

Vaihe 1: 0.023173809051513672 s
Vaihe 2: 0.022482872009277344 s
Vaihe 3: 0.028714656829833984 s
Vaihe 4: 6.7462852001190186 s
Vaihe 5: 0.023403406143188477 s
Vaihe 6: 0.07644820213317871 s

Varmistukset

Taulujen määrittelyssä on käytetty UNIQUE-määrettä, minkä ansiosta tietokanta varmistaa, että näissä sarakkeissa joka rivillä on eri arvo. Jos sovelluksessa yritetään laittaa sama arvo uudestaan, tapahtuu SQL-virhe, jonka seurauksena sovellus näyttää virheviestin.

Koska jokainen INSERT-komento on oma transaktionsa, varmistukset toimivat myös silloin, kun sovelluksella on useampi samanaikainen käyttäjä.

Sovelluksen lähdekoodi

```
import sqlite3
from time import time
from random import randint

db = sqlite3.connect("testi.db")
db.isolation_level = None

c = db.cursor()

while True:
    t = input("Valitse toiminto (1-9): ")
    if t == "1":
        c.execute("CREATE TABLE Paikat (id INTEGER PRIMARY KEY, nimi TEXT
UNIQUE)")
        c.execute("CREATE TABLE Asiakkaat (id INTEGER PRIMARY KEY, nimi TEXT
UNIQUE)")
        c.execute("CREATE TABLE Paketit (id INTEGER PRIMARY KEY, koodi TEXT
UNIQUE, asiakas_id INTEGER REFERENCES Asiakkaat)")
        c.execute("CREATE TABLE Tapahtumat (id INTEGER PRIMARY KEY, paketti_id
INTEGER REFERENCES Paketit, paikka_id INTEGER REFERENCES Paikat, kuvaus TEXT,
aika TEXT)")
        print("Tietokanta luotu")
    if t == "2":
        paikka = input("Anna paikan nimi: ")
        try:
            c.execute("INSERT INTO Paikat (nimi) VALUES (?)",[paikka])
            print("Paikka lisätty")
        except:
            print("VIRHE: Paikka on jo olemassa")
    if t == "3":
        asiakas = input("Anna asiakkaan nimi: ")
        try:
            c.execute("INSERT INTO Asiakkaat (nimi) VALUES (?)",[asiakas])
            print("Asiakas lisätty")
        except:
            print("VIRHE: Asiakas on jo olemassa")
    if t == "4":
        koodi = input("Anna seurantakoodi: ")
        asiakas = input("Anna asiakkaan nimi: ")
        asiakas_id = c.execute("SELECT id FROM Asiakkaat WHERE nimi=?",
[asiakas]).fetchone()
        if asiakas_id != None:
            try:
                c.execute("INSERT INTO Paketit (koodi,asiakas_id) VALUES (?,?)",
[koodi,asiakas_id[0]])
                print("Paketti lisätty")
            except:
                print("VIRHE: Seurantakoodi on jo käytetty")
```

```
        else:
            print("VIRHE: Asiakasta ei löytynyt")
    if t == "5":
        koodi = input("Anna seurantakoodi: ")
        paikka = input("Anna paikka: ")
        kuvaus = input("Anna kuvaus: ")
        paketti_id = c.execute("SELECT id FROM Paketit WHERE koodi=?",
[koodi]).fetchone()
        if paketti_id == None:
            print("VIRHE: Pakettia ei löytynyt")
            continue
        paikka_id = c.execute("SELECT id FROM Paikat WHERE nimi=?",
[paikka]).fetchone()
        if paikka_id == None:
            print("VIRHE: Paikkaa ei löytynyt")
            continue
        c.execute("INSERT INTO Tapahtumat (paketti_id, paikka_id, kuvaus, aika)
VALUES (?,?,?,datetime('now'))",[paketti_id[0],paikka_id[0],kuvaus])
        print("Tapahtuma lisätty")
    if t == "6":
        koodi = input("Anna seurantakoodi: ")
        r = c.execute("SELECT T.aika, B.nimi, T.kuvaus FROM Paketit A, Paikat B,
Tapahtumat T WHERE A.koodi=? AND A.id=T.paketti_id AND B.id=T.paikka_id",
[koodi])
        for t in r.fetchall():
            print(t[0],"-",t[1],"-",t[2])
    if t == "7":
        asiakas = input("Anna asiakkaan nimi: ")
        r = c.execute("SELECT P.koodi, COUNT(T.id) FROM Asiakkaat A, Paketit P
LEFT JOIN Tapahtumat T ON P.id=T.paketti_id WHERE P.asiakas_id=A.id AND A.nimi=?
GROUP BY P.id",[asiakas])
        for t in r.fetchall():
            print(t[0],"-",t[1])
    if t == "8":
        paikka = input("Anna paikan nimi: ")
        paiva = input("Anna päivämäärä: ")
        r = c.execute("SELECT COUNT(*) FROM Tapahtumat T, Paikat P WHERE
T.paikka_id=P.id AND P.nimi=? AND T.aika LIKE ?",[paikka,paiva+"%"])
        print("Tapahtumien määrä:",r.fetchone()[0])
    if t == "9":
        if input("Haluatko indeksit (K/E)? ") == "K":
            c.execute("CREATE INDEX idx1 ON Paketit (asiakas_id)")
            c.execute("CREATE INDEX idx2 ON Tapahtumat (paketti_id)")
        aika1 = time()
        c.execute("BEGIN TRANSACTION")
        for i in range(1,10**3+1):
            c.execute("INSERT INTO Paikat (nimi) VALUES (?)",["P"+str(i)])
        c.execute("COMMIT")
        aika2 = time()
        print("Vaihe 1:",(aika2-aika1),"s")
        c.execute("BEGIN TRANSACTION")
        for i in range(1,10**3+1):
            c.execute("INSERT INTO Asiakkaat (nimi) VALUES (?)",["A"+str(i)])
        c.execute("COMMIT")
        aika3 = time()
        print("Vaihe 2:",(aika3-aika2),"s")
        c.execute("BEGIN TRANSACTION")
        for i in range(1,10**3+1):
            c.execute("INSERT INTO Paketit (koodi,asiakas_id) VALUES (?,?)",
["K"+str(i),randint(1,10**3)])
```

```
        c.execute("COMMIT")
        aika4 = time()
        print("Vaihe 3:",(aika4-aika3),"s")
        c.execute("BEGIN TRANSACTION")
        for i in range(1,10**6+1):
            c.execute("INSERT INTO Tapahtumat (paikka_id,paketti_id,kuvaus)
VALUES (?,?,?)",[randint(1,10**3),randint(1,10**3),"abc"])
        c.execute("COMMIT")
        aika5 = time()
        print("Vaihe 4:",(aika5-aika4),"s")
        for i in range(1,10**3+1):
            c.execute("SELECT COUNT(*) FROM Paketit WHERE asiakas_id=?",
[randint(1,10**3)])
        aika6 = time()
        print("Vaihe 5:",(aika6-aika5),"s")
        for i in range(1,10**3+1):
            c.execute("SELECT COUNT(*) FROM Tapahtumat WHERE paketti_id=?",
[randint(1,10**3)])
        aika7 = time()
        print("Vaihe 6:",(aika7-aika6),"s")
    if t == "99":
        break
```