# Harjoitus 3

## SQL

1. **Mitä SQL-standardeja on olemassa ja tukevatko kaikki tietokannat näitä samalla tavalla? Mitä yleisiä tietotyyppejä SQL-tietokannoissa on?**

<https://en.wikipedia.org/wiki/SQL>

<http://www.cs.helsinki.fi/u/laine/tkp/sql/sql_historia.html>

Ensimmäiset SQL-kieleen pohjautuvat tietokannan hallintajärjestelmät tulivat kaupallisesti saataville 1970-luvun lopulla, myös vaihtoehtoisia. Standardoinnin kohteeksi valittiin SQL. Standardointia hoitivat sekä amerikkalainen ANSI että kansainvälinen ISO. 1986 hyväksyttiin ensimmäinen SQL-standardi (ydin noin 40s + moduulikieli, upotus ohjelmointikieliin, yhteensä noin 100s). 1989 valmistui laajennus (noin 20 sivua). 1992 hyväksyttiin uusi versio SQL-92 (SQL2), joka sisälsi laajennuksia kieleen (laajuus noin 600 sivua + kuvaustietostandardi, yhteensä lähes 1000 sivua). 1995 ja 1996 lisäyksiä- 1999 SQL-99 (SQL3, laajuus noin 2100 sivua).

SQL was adopted as a standard by the American National Standards Institute (ANSI) in 1986 as SQL-86 and the International Organization for Standardization (ISO) in 1987. Nowadays the standard is subject to continuous improvement by the Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 32, Data management and interchange, which affiliate to ISO as well as IEC. It is commonly denoted by the pattern: ISO/IEC 9075-n:yyyy Part n: title, or, as a shortcut, ISO/IEC 9075.

The SQL standard is divided into nine parts:

Framework, Foundation, Call-Level Interface, Persistent stored modules, management of external data, object language bindings, information and definition of schemas, SQL routines and types using the java programming language and XML-related specifications.

Another criticism is that SQL implementations are incompatible between vendors and do not necessarily completely follow standards. Popular implementations of SQL commonly omit support for basic features of Standard SQL, such as the DATE or TIME data types. The most obvious such examples, and incidentally the most popular commercial and proprietary SQL DBMSs, are Oracle (whose DATE behaves as DATETIME,[33][34] and lacks a TIME type)[35] and MS SQL Server (before the 2008 version). As a result, SQL code can rarely be ported between database systems without modifications.

There are several reasons for this lack of portability between database systems:

The complexity and size of the SQL standard means that most implementors do not support the entire standard.

The standard does not specify database behavior in several important areas (e.g. indexes, file storage...), leaving implementations to decide how to behave.

The SQL standard precisely specifies the syntax that a conforming database system must implement. However, the standard's specification of the semantics of language constructs is less well-defined, leading to ambiguity.

Many database vendors have large existing customer bases; where the newer version of the SQL standard conflicts with the prior behavior of the vendor's database, the vendor may be unwilling to break backward compatibility.

There is little commercial incentive for vendors to make it easier for users to change database suppliers (see vendor lock-in).

Users evaluating database software tend to place other factors such as performance higher in their priorities than standards conformance.

Yleisiä tietotyyppejä ovat mm. vaihtelevan pituinen merkkijono VARCHAR(n), totuusarvo BOOLEAN, kokonaisluku INTEGER, liukuluku FLOAT ja päivämäärä DATE.

1. **Käyttötapauksia** [viime viikon](https://github.com/kasper/tikape-kesa-2014/blob/master/viikko-2/harjoitus.md) **tehtävän 8 chat-palveluun liittyen. Kerro minkälaisella SQL-kyselyllä seuraavat käyttötapaukset luonnistuisivat:**

**Palvelun ylläpitäjä haluaa lisätä uuden chat-huoneen. Chat-huoneen nimi on viherpeukalot ja kuvaus on Vihreä kesä.**

Insert into huone (id, nimi, kuvaus)

values (1, "viherpeukalot", "Vihreä kesä");

**Palvelun ylläpitäjä haluaa lisätä uuden käyttäjän. Uuden käyttäjän käyttäjätunnus on aimo ja salasana porkkana.**

Insert into henkilo (id, kayttajatunnus, salasana)

values (1, "aimo", "porkkana");

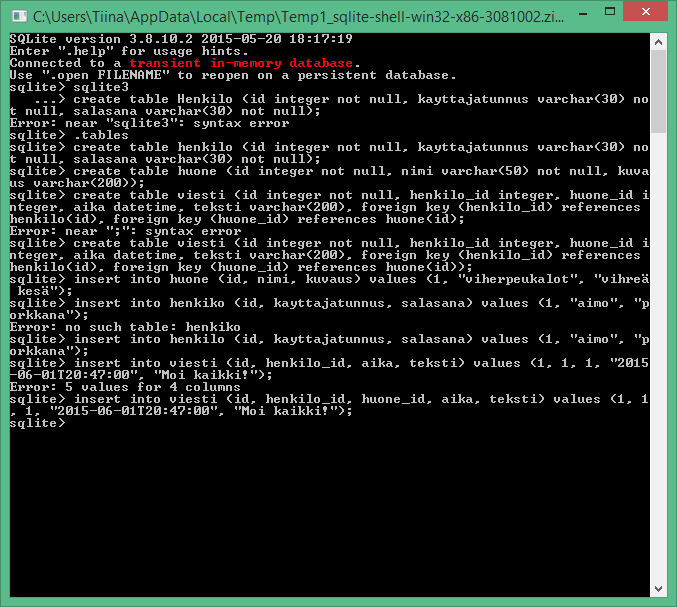
**Käyttäjä aimo kirjoittaa viestin Moi kaikki! huoneeseen virherpeukalot**

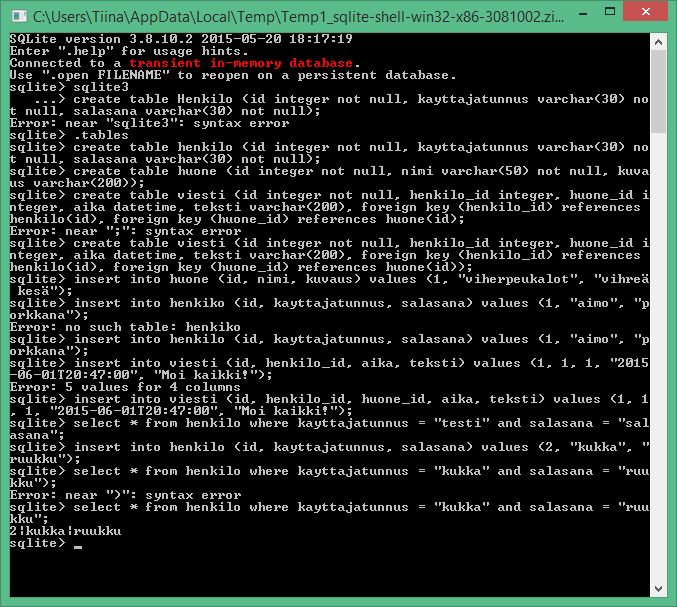
Insert into viesti (id, henkilo\_id, huone\_id, aika, teksti)

values (1, 1, 1, "2015-06-01T20:47:00", "Moi kaikki!");

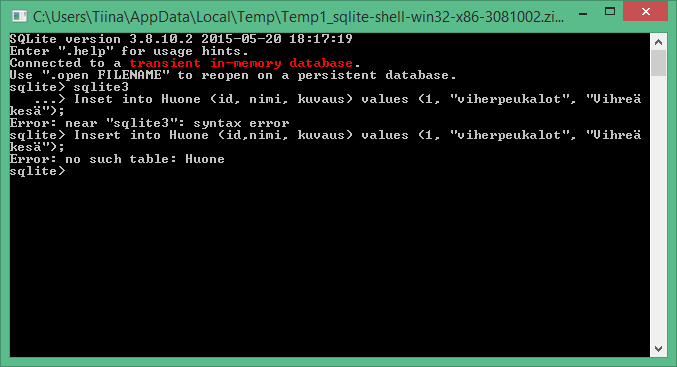
**Tuntematon käyttäjä yrittää kirjautua käyttäen käyttäjätunnusta testi ja salasanaa salasana**

select \* from henkilo where kayttajatunnus = "testi" and salasana = "salasana";





Kantaan kysely, joka palauttaa 0 riviä (jos ei ole lisätty tätä). Eli ei hakutuloksia.



1. **Luo** [viime viikon](https://github.com/kasper/tikape-kesa-2014/blob/master/viikko-2/harjoitus.md) **tehtävän 8 tietokantatauluille SQL CREATE-TABLE-lauseet. Mieti sopivat rajoitteet sarakkeille.**

Create table henkilo (

id integer not null,

kayttajatunnus varchar(30) not null,

salasana varchar(30) not null);

Create table huone (

id integer not null,

nimi varchar(50) not null,

kuvaus varchar(200));

create table viesti (

id integer not null,

henkilo\_id integer,

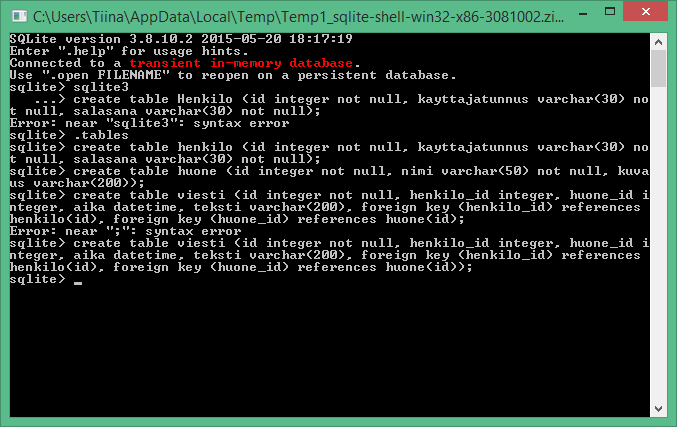
huone\_id integer,

aika datetime, ->timestamp?

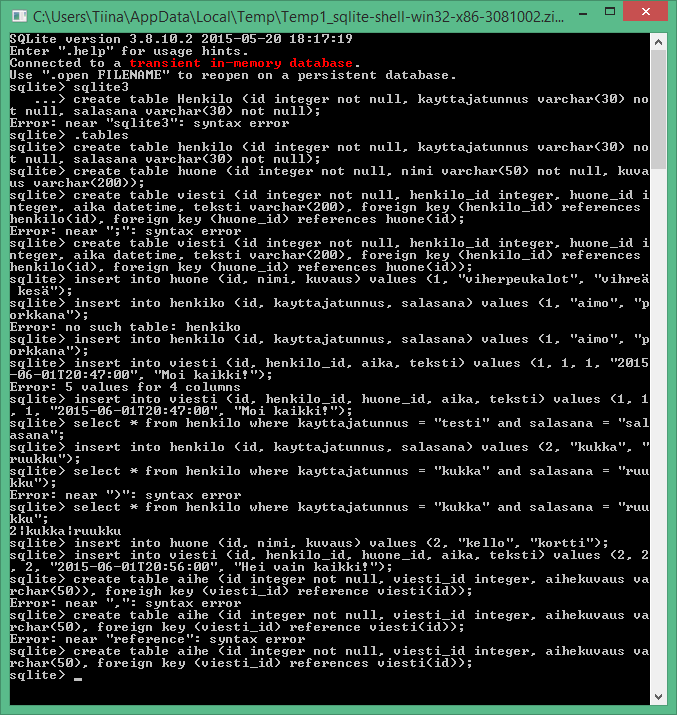
teksti varchar(200),

foreign key (henkilo\_id) references Henkilo(id),

foreign key (huone\_id) references huone(id));

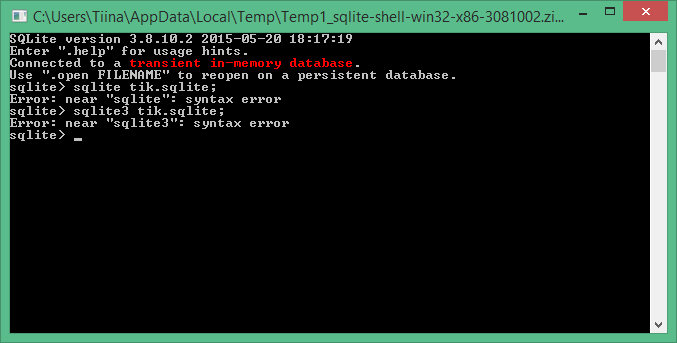


1. Tee SQLite-tietokanta, johon suoritat edelliset tehtävät. Lisää tietokantaan lisää käyttäjiä, kanavia ja viestejä. Lisää tietokantaan myös joku uusi vapaavalintainen taulu ja siihen tietoja. Pääset tehtävässä alkuun katsomalla lyhyen SQLite-oppaan. Ota komennot ja tulokset talteen.



1. [Tik](http://tik.herokuapp.com/) on tehtävälistasovellus. Löydät sovelluksen SQLite-tietokannan [tästä](https://moodle.helsinki.fi/pluginfile.php/972508/mod_assign/intro/tik.sqlite). Suorita seuraavat käyttötapaukset ja ota ne talteen.

* Hae kaikki käyttäjän kasper tehtävät
* select \* from tasks, users where users.id = tasks.creator\_id and users.username = ’kasper’;
* Lisää käyttäjälle jack uusi tehtävä, jonka omistaja on kasper
* insert into tasks (description, owner\_id, created\_at’), creator\_id) values (’pyyhi lattiat’, ’1’. datetime(‘now’), ‘2’);
* Liitä tehtävään avainsanoiksi critical ja enhancement
* insert into tags (name, created\_at) values(‘critical’, datetime(‘now’));
* insert into tags (name, created\_at) values(‘enhancement’, datetime(‘now’));
* Lisää uusi User Experience-tiimi ja liitä sekä kasper että jack sen jäseniksi
* Poista lopuksi Software-tiimi



6. Mitä tarkoittaa tietokantatransaktio? Milloin transaktiota kannattaa käyttää ja mitä hyötyä siitä on?

Tietokantatransaktio on tietokantaa käsittelevä prosessin osa, jonka vaikutusten halutaan muodostavan yhden jakamattoman (atomisen) kokonaisuuden.

Esimerkiksi tilisiirtotapahtuma, kolmiosainen kysely:

* muokkaa siirtäjän tilitietoja
* muokkaa saajan tilitietoja
* lisää tilitapahtumiin tieto.

Esimerkissä jakamattomuus merkitsee sitä, että kaikki 3 tietokantaoperaatiota suoritetaan eikä vain osaa niistä.

Käsittelyssä voi sattua häiriöitä missä vaiheessa tahansa. Nämä voivat johtua ulkoisista syistä tai olla tietokannan itse aiheuttamia. Ilman transaktiota voisi siis käydä siten, että tililtä lähtee rahat, mutta saaja ei ikinä saa rahojaan – tällöin operaatio ei olisi jakamaton.

Transaktiota kannattaa käyttää kun joukko muutoksia pitää tehdä kokonaisuudessaan ollakseen validi tapahtuma. Toisin sanoen: jos on mahdollisuus, että tietokannan tila on väärä kun vain osa kyselystä toteutuu, käytä transaktiota. Transaktio mahdollistaa mm. sen perumisen kokonaisuudessaan sen epäonnistuessa. Kyselyt suoritetaan myös kokonaisuutena, joten muutoksia ei voi tapahtua transaktion "keskellä"

Transaktio – sarja (check) – myös optimoinnin kannalta hyvä

7. Mitä tarkoittaa SQL-injektio ja miten siihen voi varautua? Kokeile suorittaa SQL-injektioita [Väestörekisteriin](http://t-khirviko.users.cs.helsinki.fi/vaestorekisteri/). Arto Vihavaisen lompakko tippui Unicafen kassalle ja tiedämme hänen SOTU:n olevan 010190-111A.

* SQL-injektiossa ideana on saada ohjelma suorittamaan SQL-kyselyitä, joita ohjelmaa ei ole tarkoitettu tekemään.
* <http://en.wikipedia.org/wiki/SQL_injection>
* **SQL injection** is a [code injection](http://en.wikipedia.org/wiki/Code_injection) technique, used to [attack](http://en.wikipedia.org/wiki/Attack_%28computing%29) data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).[[1]](http://en.wikipedia.org/wiki/SQL_injection#cite_note-1) SQL injection must exploit a [security vulnerability](http://en.wikipedia.org/wiki/Security_vulnerability) in an application's software, for example, when user input is either incorrectly filtered for [string literal](http://en.wikipedia.org/wiki/String_literal) [escape characters](http://en.wikipedia.org/wiki/Escape_sequence) embedded in [SQL](http://en.wikipedia.org/wiki/SQL) statements or user input is not [strongly typed](http://en.wikipedia.org/wiki/Strongly-typed_programming_language) and unexpectedly executed. SQL injection is mostly known as an attack [vector](http://en.wikipedia.org/wiki/Vector_%28malware%29) for websites but can be used to attack any type of SQL database.
* select \* from users where sotu = ‘’ or ‘’; -> saa kaikki näkyviin.
* lisätiedolla saa suojattua (voi muuntaa tiedon niin että kyseistä tietoa ei saa hauilla esiin)s