Tuntitehtävät vko 3
Tietoturvamäärítykset ja käyttäjien hallinta. Tietokannan eheyden palautus. Tutustuminen Oracle Database:en:

a) Tutustu Oracle Database:een. Se löytyy Oracle VM VirtualBox:sta, johon on asennettu Debian Linux. Debian Linux:iin kirjaudutaan student tunnuksella. Kyseistä ympäristöä on esitelty Moodle:sta edellisen viikon materiaalista (koulun VirtualBox ohje käyttäjälle). Käynnistä ensimmäiseksi Oracle Database ja kirjaudu Oracle Database Home Page:lle. Kirjautumiseen tarvitset käyttäjätunnuksen ja salasana, joka löytyy siis VirtualBox ohjeesta. Voit käyttää esimerkiksi ylläpitäjän system tunnusta. Tutustu kyseiseen työkaluun. Mitä sillä pystyy tekemään?

   Vastaus:

   Oracle Database:n saa käynnitettyä Debianissa, valitsemalla:

   - Applications -> Oracle Database -> Start Database

   Samasta paikasta löytyy myös Oracle Database Home Page:

   - Applications -> Oracle Database -> Go To Database Home Page
   - Kirjaudu system tunnuksella tietokantapalvelimeen

   Oracle Database Home Page:stä löytyy Administration, ObjectBrowser, SQL ja Utilities. Administrator:lla pystyy nimensä mukaisesti tekemään joitain tietokannan ylläpitotöitä, kuten esimerkiksi lisäämään käyttäjiä, tutkimaan levyjen ja muistin käyttöä sekä monitoroimaan istuntoja ja statistiikkaa tietokannan tilasta.

   ObjectBrowser:lla voi luoda tauluja, sequence:ja (vrt MySQL:n auto_increment), trigger:eitä, indeksejä, proseduureja jne. Niitä voi myös selata.

   SQL –ikonin alta puolestaan löytyy editori ja SQL kääntäjä, jonka avulla voi suorittaa haluamiaan SQL lauseita. Oracle:a ja muitakin tietokantoja pidetään yllä pitkälti erilaisten scriptien avulla.

   Utilities –ikoni pitää sisällään työkalut datan vientiin ja tuontiin, ddl lauseiden generointiin (esim generoidaan lauseet, joilla jokin tietokanta on luotu). Se sisältää myös erilaisia raportteja tauluista, proseduureista, tietoturvasta yms. Myös roskakori löytyy täältä.

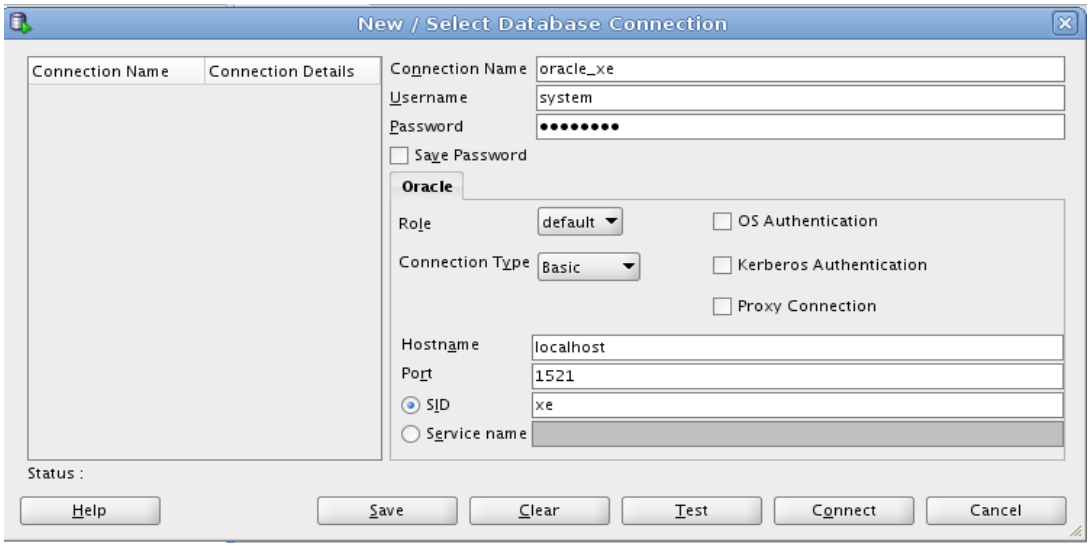
b) Tutustu seuraavaksi Debianista löytyvään Oracle SQL Developer:iin. Mitä sillä pystyy tekemään?

Vastaus:

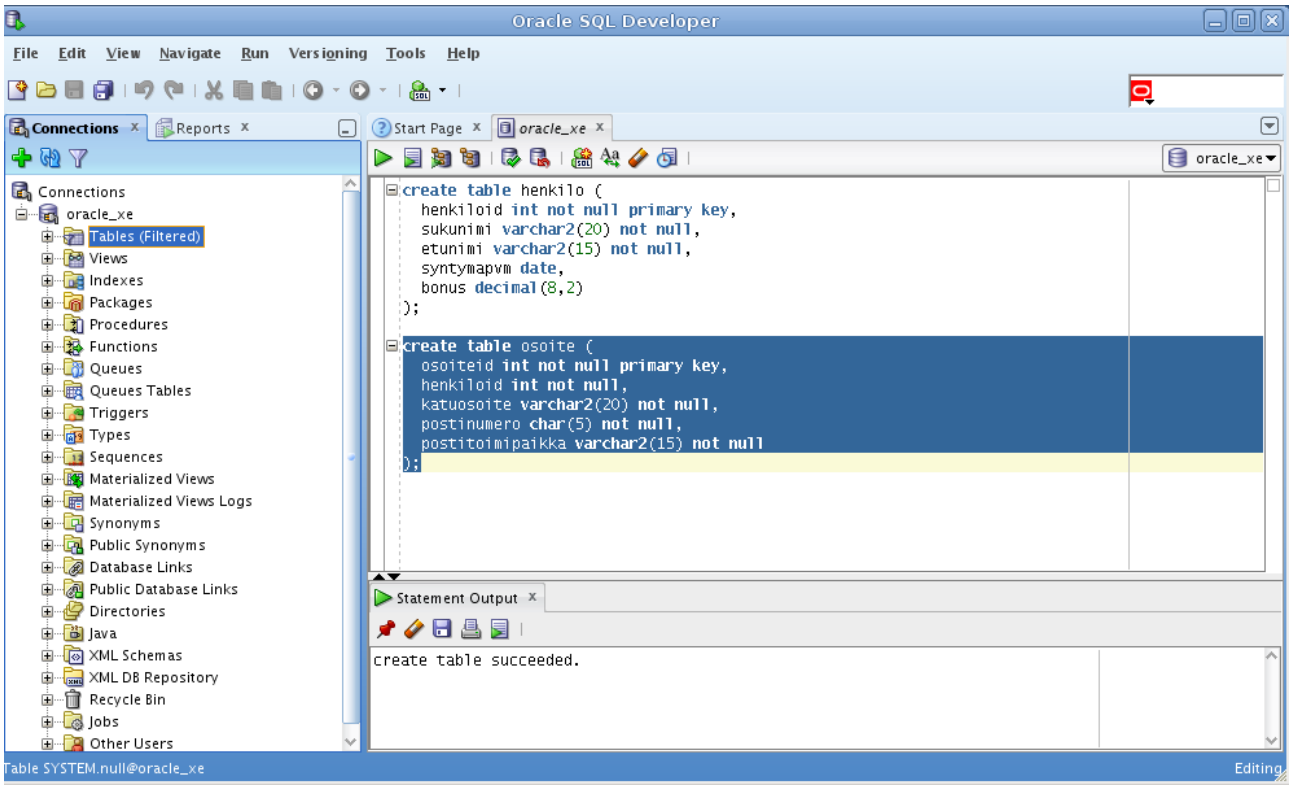
Oracle SQL Developer on ilmainen työkalu tietokannan kehittäjälle, joka löytyy osoitteesta: http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html. Siellä on esitelty työkalua laajemminkin. Debianista se löytyy valitsemalla:

- Applications -> Programming -> Oracle SQL Developer

- Alla on miten SQL Developer:lla voi kirjautua Oracle Database:een:



- Kirjautumisen jälkeen avautuu alla olevan kuvan mukainen käyttöliittymä, joka kertoo jo paljon mitä tällä työkalulla pystyy tekemään:



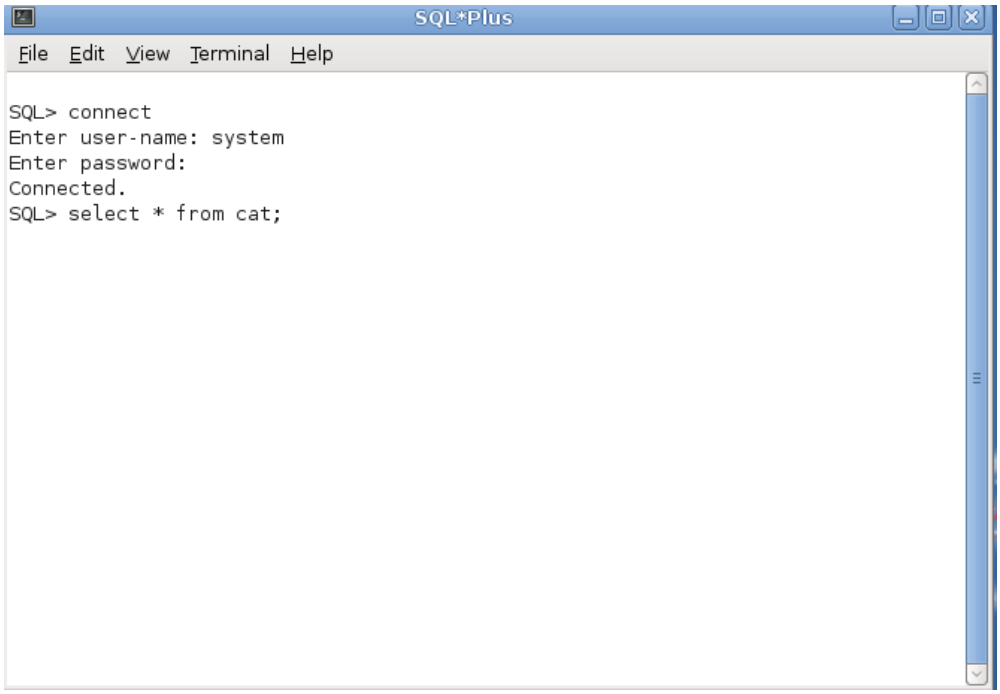
Voit siis kokeilla Oracle Database:ä vaikkapa alla olevalla script:llä:

```
create table henkilo (
          henkiloid int not null primary key,
          sukunimi varchar2(20) not null,
          etunimi varchar2(15) not null,
          syntymapvm date,
          bonus decimal(8,2)
);

create table osoite (
          osoiteid int not null primary key,
          henkiloid int not null,
          katuosoite varchar2(20) not null,
          postinumero char(5) not null,
          postitoimipaikka varchar2(15) not null,
          foreign key (henkiloid) references henkilo(henkiloid)
);
```

Kyseessä on siis nimensä mukaisesti SQL kehittäjälle ja ylläpitäjälle tarkoitettu työkalu.SQL Developer on Oraclen mukaan SQLPlus:san graafinen versio. SQL Developer:illa pystyy esimerkiksi luomaan taulua, indeksejä, sequence:jä, trigger:eitä, proseduudeja jne. Luomansa scriptit voi myös laittaa talteen versionhallintaan, joka onkin kehittäjän kannalta ok. Tietokannan ylläpitäjälle on siis hyötyä tästäkin työkalusta, koska siinä voi ajaa scriptejä. Tosin eniten niitä ylläpito tehtävissä ajetaan suoraan komentotasolta SQLPlus:saa käyttäen (ks. alla olevaa kuvaa tästä työkalusta). Kyseistä työkalua on esitelty Oracle:n sivuilla osoitteessa: http://docs.oracle.com/cd/B25329_01/doc/appdev.102/b25108/xedev_sqlplus.htm.

- SQL Plus:san saa käynnistettyä Debianissa, valitsemalla:
  Applications -> Oracle Database -> Run SQL Command Line



Oracle Database:een voi siis kirjautua sisään käyttäjäksi connect lauseella. Tämän jälkeen voi käyttää mitä tahansa PL/SQL:n kielen lausetta jos oikeudet riittävät (PL/SQL:stä on lisää tietoa osoitteessa http://www.oracle.com/technetwork/database/application-development/index-101230.html). Yllä on kirjauduttu system tunnuksella, joka on ylläpitäjälle eli dba:lle tarkoitettu tunnus Oracle Database:ssa. Oikeudet siis riittävät ylläpitoon.

Yllä olevassa kuvassa on seuraavaksi tutkittu select * from cat; lauseella system catalog:ia.

c) Olet saanut ylläpidettäväksi verkkokaupan tietokannan, joka sisältää tiedot myytävistä tuotteista, sekä varastotilanteen ja tilaustiedot, joita asiakkaat ovat tehneet. Alla on esitetty kyseisen tietokannan luontilauseet. Luo SQL Server:iin ja Oracle Database:een tarvittavat roolit ja käyttäjät tarvittavine oikeuksineen tietokantaan. Anna tarvittavat T-SQL lauseet:

```
DROP TABLE Maksuehto;
DROP TABLE Toimitusrivi;
DROP TABLE Toimitukset;
DROP TABLE Varasto;
DROP TABLE Tilausrivi;
DROP TABLE Tilaus;
DROP TABLE Tuote;
DROP TABLE Asiakas;

drop schema ostil;

CREATE TABLE Asiakas (
AsiakasID nvarchar(256) not null primary key,
Sukunimi nvarchar(25) not null,
Etunimi nvarchar(10) not null,
Osoite nvarchar(50) null,
sposti nvarchar(50) null
);
CREATE TABLE Tuote (
TuoteID int not null,
Nimi nvarchar(50) not null,
Hinta Decimal(8,2) not null,
Veroprosentti Decimal(8,2) not null,
PRIMARY KEY (TuoteID)
);
CREATE TABLE Tilaus (
TilausID int not null primary key IDENTITY(1,1),
AsiakasID nvarchar(256) not null,
pvm DATE not null,
maksuehto nvarchar(50) not null,
FOREIGN KEY (AsiakasID) REFERENCES Asiakas(AsiakasID));

CREATE TABLE Tilausrivi (
RiviID int not null primary key IDENTITY(1,1),
TilausID int not null,
TuoteID int not null,
lukumaara int not null,
FOREIGN KEY (TilausID) REFERENCES Tilaus(TilausID),
FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
);
CREATE TABLE Varasto (
TuoteID int not null primary key,
vapaana_lkm int not null,
varattujen_lkm int not null,
FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
);
CREATE TABLE Toimitukset (
ToimitusID int not null primary key IDENTITY(1,1),
AsiakasID nvarchar(256) not null,
```

```
tilauspvm DATE not null,
toimituspvm DATE not null,
maksuehto nvarchar(50) not null,
FOREIGN KEY (AsiakasID) REFERENCES Asiakas(AsiakasID)
);
CREATE TABLE Toimitusrivi (
RiviID int not null primary key IDENTITY(1,1),
ToimitusID int not null,
TuoteID int not null,
lukumaara int not null,
FOREIGN KEY (ToimitusID) REFERENCES Toimitukset(ToimitusID),
FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
);
CREATE TABLE Maksuehto (
MaksuehtoID int not null primary key IDENTITY(1,1),
Maksuehto nvarchar(100) not null,
Kuvaus nvarchar(100));
```

Vastaus:

Ks. SQL Server http://msdn.microsoft.com/en-us/library/ff929055.aspx, http://msdn.microsoft.com/en-us/library/ms173463.aspx.
Esimerkiksi:
Tarvittaessa haluttu schema voidaan luoda tietokantaan seuraavalla tavalla (ks. http://msdn2.microsoft.com/en-us/library/ms189462.aspx):

```
USE AdventureWorks;
CREATE SCHEMA Sprockets AUTHORIZATION Annik
    CREATE TABLE NineProngs (source int, cost int, partnumber int)
    GRANT SELECT TO Mandar
    DENY SELECT TO Prasanna;
GO
```

Ja tarvittavat kirjautumisessa käytettävät käyttäjätunnukset (login:t) ja tietokantaoikeudet (user) tietokannalle ja schemalle voidaan luoda seuraavalla tavalla:

```
CREATE LOGIN matti
WITH PASSWORD = 'salasana1',
DEFAULT_DATABASE = AdventureWorks

CREATE USER matti FOR LOGIN matti
WITH DEFAULT_SCHEMA = HumanResources
```

`CREATE LOGIN` ks. http://msdn2.microsoft.com/en-us/library/ms189751.aspx
`CREATE USER` ks. http://msdn2.microsoft.com/en-us/library/ms173463.aspx

```
(GRANT CONNECT TO guest)
```

Tietokannan käyttäjän omistaman tietokanta roolin luonti (ks. http://msdn2.microsoft.com/en-us/library/ms187936.aspx):

```
USE AdventureWorks;
CREATE ROLE buyers AUTHORIZATION BenMiller;
GO
```

Tietokanta roolin luonti, jonka omistaa kiinteä tietokanta rooli:

```
USE AdventureWorks;
CREATE ROLE auditors AUTHORIZATION db_securityadmin;
GO
```

Rooleja ovat SQLServer:ssä:

1. Palvelinroolit:
   `bulkadmin`
   `dbcreator`
   `diskadmin`
   `processadmin`
   `securityadmin`
   `serveradmin`
   `setupadmin`
   `sysadmin`
   Lisäksi jokaisella login id:llä on:
   `public` –rooli
2. Tietokantaroolit:
   `db_accessadmin`
   `db_backupoperator`
   `db_datareader` (asiakas käyttäjälle)
   `db_datawriter` (asiakas käyttäjälle)
   `db_ddladmin`
   `db_denydatareader`
   `db_denydatawriter`
   `db_owner`
   `db_securityadmin`

Oikeuksien määrittely:
`GRANT` –komennosta on kolme eri tyyppiä:

I) `SELECT` tietokantaoikeuden myöntäminen `user` matille scheman taululle, jonka hän voi antaa muillekin `GRANT` komennolla:
```
GRANT SELECT ON SCHEMA.TAULU1 TO matti WITH GRANT OPTION
```
II) `INSERT` ja `DELETE` tietokantaoikeuksien myöntäminen roolille scheman tauluun:
```
GRANT INSERT, DELETE ON SCHEMA.TAULU1 TO testirooli
```
III) Roolin myöntäminen tunnukselle (`user`) tehdään proseduurin avulla:
```
sp_addrolemember 'testirooli', 'matti'
```

Yllä olevan mukaan siis eKauppa tietokantaan saadaan luotua ylläpitäjän käyttäjätunnus ja tarvittavat roolit sekä ostil schema ja sen alle asiakas, tuote, ... , maksuehto taulut alla olevalla T-SQL lauseella. Lopuksi on luotu asiakkaan käyttäjätunnus ja myyjän käyttäjätunnus, sekä lisätty ne asianmukaisiin rooleihin:

```
USE eKauppa;

CREATE LOGIN ostil_admin
WITH PASSWORD = 'aTvLxg7#',
DEFAULT_DATABASE = eKauppa;

CREATE USER ostil_admin FOR LOGIN ostil_admin;

CREATE ROLE Asiakas AUTHORIZATION ostil_admin;
GO

CREATE ROLE Myyja AUTHORIZATION ostil_admin;
GO

CREATE ROLE Yllapito AUTHORIZATION ostil_admin;
GO

CREATE SCHEMA ostil AUTHORIZATION ostil_admin

  CREATE TABLE Asiakas (
```

```
   AsiakasID nvarchar(256) not null primary key,
   Sukunimi nvarchar(25) not null,
   Etunimi nvarchar(10) not null,
   Osoite nvarchar(50) null,
   sposti nvarchar(50) null
)

CREATE TABLE Tuote (
   TuoteID int not null,
   Nimi nvarchar(50) not null,
   Hinta Decimal(8,2) not null,
   Veroprosentti Decimal(8,2) not null,
   PRIMARY KEY (TuoteID)
)

CREATE TABLE Tilaus (
   TilausID int not null primary key IDENTITY(1,1),
   AsiakasID nvarchar(256) not null,
   pvm DATE not null,
   maksuehto nvarchar(50) not null,
   FOREIGN KEY (AsiakasID) REFERENCES Asiakas(AsiakasID)
)

CREATE TABLE Tilausrivi (
   RiviID int not null primary key IDENTITY(1,1),
   TilausID int not null,
   TuoteID int not null,
   lukumaara int not null,
   FOREIGN KEY (TilausID) REFERENCES Tilaus(TilausID),
   FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
)

CREATE TABLE Varasto (
   TuoteID int not null primary key,
   vapaana_lkm int not null,
   varattujen_lkm int not null,
   FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
)

CREATE TABLE Toimitukset (
   ToimitusID int not null primary key IDENTITY(1,1),
   AsiakasID nvarchar(256) not null,
   tilauspvm DATE not null,
   toimituspvm DATE not null,
   maksuehto nvarchar(50) not null,
   FOREIGN KEY (AsiakasID) REFERENCES Asiakas(AsiakasID)
)

CREATE TABLE Toimitusrivi (
   RiviID int not null primary key IDENTITY(1,1),
   ToimitusID int not null,
   TuoteID int not null,
   lukumaara int not null,
   FOREIGN KEY (ToimitusID) REFERENCES Toimitukset(ToimitusID),
   FOREIGN KEY (TuoteID) REFERENCES Tuote(TuoteID)
)
CREATE TABLE Maksuehto (
   MaksuehtoID int not null primary key IDENTITY(1,1),
   Maksuehto nvarchar(100) not null,
   Kuvaus nvarchar(100)
```

```
    )

      GRANT SELECT TO asiakas
      DENY ALTER TO asiakas;
GO

CREATE LOGIN PeltonenMAtti
WITH PASSWORD = 'aTvLxg7#',
DEFAULT_DATABASE = eKauppa;

CREATE USER PeltonenMAtti FOR LOGIN PeltonenMAtti;

sp_addrolemember 'Asiakas', 'PeltonenMAtti';


CREATE LOGIN PoppanenMaija
WITH PASSWORD = 'aTvLxg7#',
DEFAULT_DATABASE = eKauppa;

CREATE USER PoppanenMaija FOR LOGIN PoppanenMaija;

sp_addrolemember 'Myyja', 'PoppanenMaija';

sp_addrolemember 'Yllapito', 'ostil_admin';
```

Vastaavan asiaan tarvittavat Orclen PL/SQL lauseet on esitelty alla:
Ks. http://docs.oracle.com/cd/E11882_01/network.112/e16543.pdf kappale 2.

Oraclessa schemaa käytetään käyttäjän omistamana loogisten tietorakenteiden kokoelmana, jonka nimi on sama kuin omistajan käyttäjätunnus (ks. http://docs.oracle.com/cd/B19306_01/server.102/b14220/schema.htm). Tämän vuoksi T-SQL:ssä yllä toteutettua ostil schema:aa ei voi tehdä samaan tapaan vaan on tehtävä ostil käyttäjä ja sitä vastaava schema. Tähän schemaan luodaan asiakas, tuote, ... , maksuehto taulut. Create table lauseissa oleva T-SQL:n identity luodaan Oracle:ssa create sequence lauseella, jonka avulla siis saadaan luotua yksikäsitteisiä (unique) id:tä generoiva sekvenssi samaan tapaan kuin SQL Server:ssä IDENTITY ja MySQL:ssä auto_increment.

Esimerkiksi:
Käyttäjätunnuksia voi luoda Oracle:n tietokannassa lauseessa.

```
CREATE USER Käyttäjätunnus1
        IDENTIFIED BY Salasana1;

CREATE USER Käyttäjätunnus2
        IDENTIFIED BY Salasana2;
```

Käyttäjäryhmiä voi luoda lauseella:

```
CREATE ROLE Käyttäjäryhmä1;
```

Käyttäjiä voi lisätä haluamaansa käyttäjäryhmään lauseella:

```
GRANT Käyttäjäryhmä1 TO Käyttäjätunnus1, Käyttäjätunnus2;
```

Oikeuksia voi antaa suoraan käyttäjille tai tavallisimmin käyttäjäryhmille lauseella:

```
GRANT insert, select, update, delete TO Käyttäjäryhmä1;
```

Käyttäjä voi muuttaa esimerkiksi salasanansa `alter user` lauseella:

```
ALTER USER Käyttäjätunnus1
        IDENTIFIED BY Salasana3;
```

d) Miten voit käyttää näkymiä (view) tietokannan käytön yksinkertaimiseen ja tietoturvan lisäämiseen?

Vastaus:

```
CREATE VIEW ostil.AV_TUOTE
        AS SELECT TuoteID, Nimi, Hinta, Veroprosentti FROM ostil.Tuote;
```

Tai

```
CREATE VIEW ostil.AV_TUOTE
        AS SELECT TuoteID, Nimi, Hinta FROM ostil.Tuote;
```

jolloin näkymän avulla ei voi lisätä enää tuote tauluun dataa. Jos lisäksi poistetaan asiakas roolilta oikeudet ostil.Tuote tauluun, ei salectointi tuote tauluun enää onnistu suoraan. On siis pakko käyttää näkymää.

```
REVOKE INSERT, UPDATE, DELETE, ALTER ON ostil.TUOTE TO asiakas;
```

Tietokannan asiakas taulua voi nyt käyttää ostil.AV_TUOTE näkymän avulla:

```
SELECT * FROM ostil.AV_TUOTE;

tai

SELECT TuoteID, Nimi, Hinta, Veroprosentti FROM ostil.AV_TUOTE;
```

Ks. http://msdn.microsoft.com/en-us/library/ms190174

Grant -lauseella saa annettua tarvittavat oikeudet esimerkiksi asiakas roolille ostil.AV_TUOTE näkymään:

```
GRANT SELECT ON ostil.AV_TUOTE TO asiakas;
```

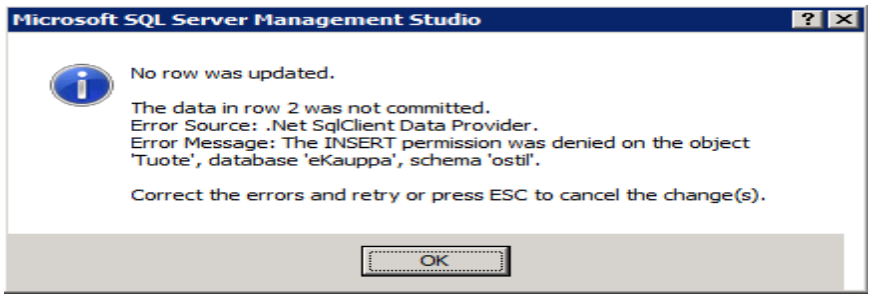
Grant –lauseesta on kerrottu lisää osoitteessa: http://msdn.microsoft.com/en-us/library/ms187965.aspx

Revoke lauseella saat poistettua ostil.Tuote taulusta oikeudet asiakas roolilta.

```
REVOKE INSERT, UPDATE, DELETE, ALTER ON ostil.Tuote TO asiakas;
```

Revoke lauseesta on kerrottu lisää osoitteessa: http://msdn.microsoft.com/en-us/library/ms187728.aspx.

Tarkista myös, että asiakas roolilta poistamasi oikeudet myös toimivat niin kuin pitää. Kirjaudu siis vaikkapa PeltonenMAtti käyttäjätunnuksella SQL Server Management Studioon toista istuntoa käyttäen (jätä siis Administrator tunnuksella kirjatunut ikkuna auki). Kokeile vaikkapa lisätä uutta rivia ostil.Tuote tauluun suoraan. Sen ei pitäisi/saisi onnitua enää, koska tarvittavia oikeuksia ei enää ole:



e) Mitä SQL Server 2012:ssa tarkoitetaan Contained Databases tietokannalla? Mitä vaikutuksia sen käyttämisellä on käyttäjien oikeuksiin? Anna myös T-SQL lauseet tällaisen tietokannan luonnista.

Vastaus:

Ks. http://msdn.microsoft.com/en-us/library/ff929071, http://blogs.msdn.com/b/mvpawardprogram/archive/2012/07/16/contained-databases-in-sql-server-2012.aspx, http://msdn.microsoft.com/en-us/library/ff929055.aspx

First, we need to enable "*contained database authentication*" on the SQL Server instance:

```
sp_configure 'contained database authentication', 1;
GO
RECONFIGURE
GO
```

Then, we create the partially contained database (along with a sample table for demo purposes):

```
--Create partially contained database
USEmaster
GO
CREATE DATABASE[ContDB]
CONTAINMENT=PARTIAL
GO
--Create sample table with sample records
USE [ContDB]
GO
CREATE TABLEtblSample(
id int ,
descrvarchar (250)
)
GO
INSERT INTO tblSample
VALUES
(10,'Sample value 1'),
(20,'Sample value 2'),
(30,'Sample value 3')
GO
```

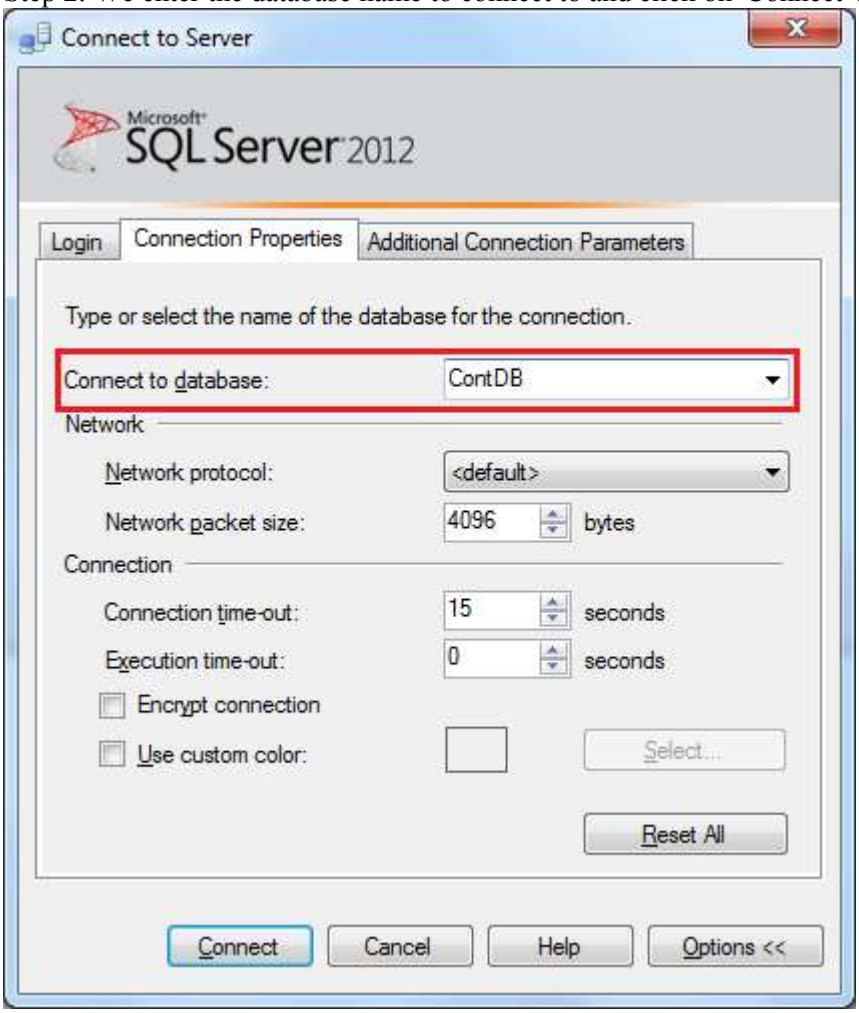
The last step is to create the user(s) that will be accessing the contained database:

```
USE [ContDB]
GO
CREATE USER ContUser1WITHPASSWORD=N'secure1$',DEFAULT_SCHEMA=[dbo]
GO
EXEC sp_addrolemember'db_owner', 'ContUser1'
GO
```

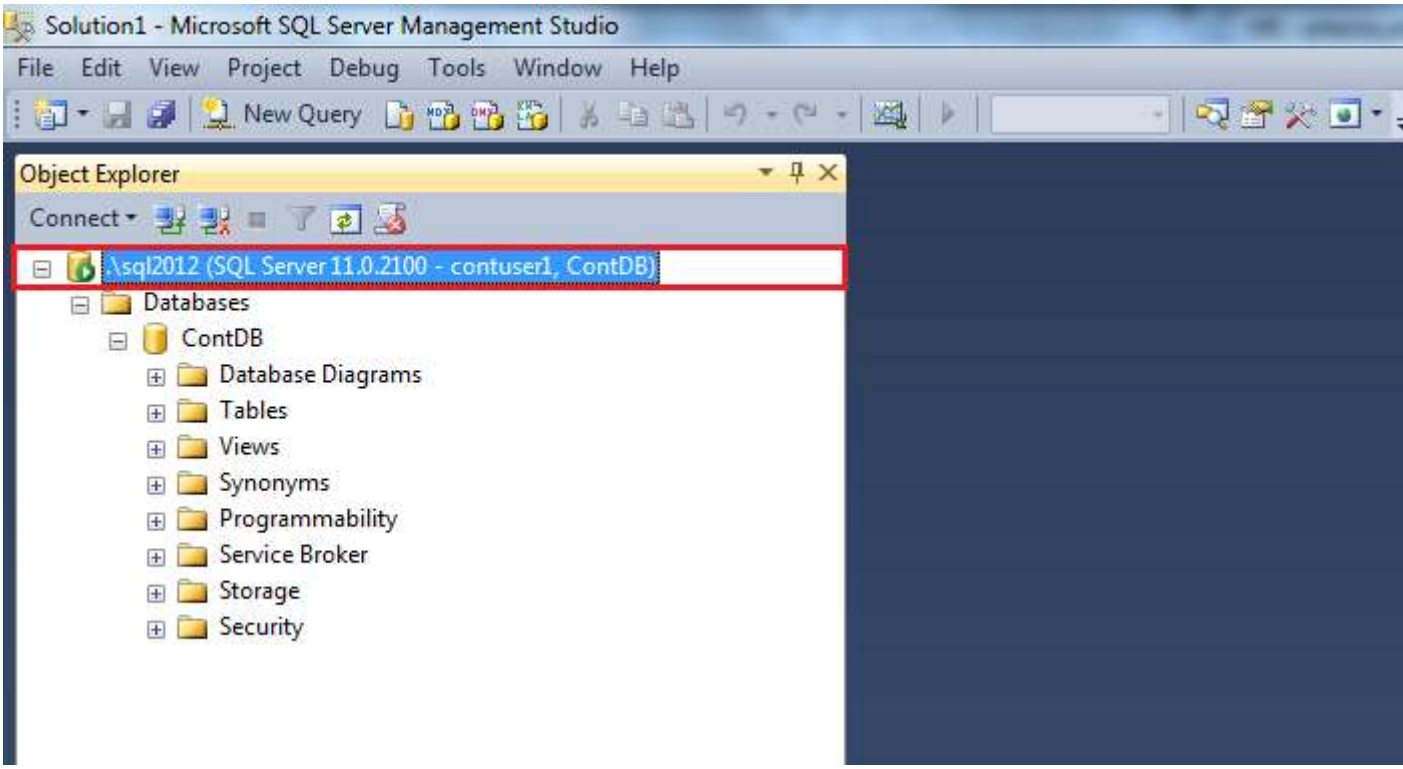
*Note: You are also able to use a Windows login if preferred.
Now let's try to log into the database using the user "ContUser1":

Step 1: We enter the database user credentials.

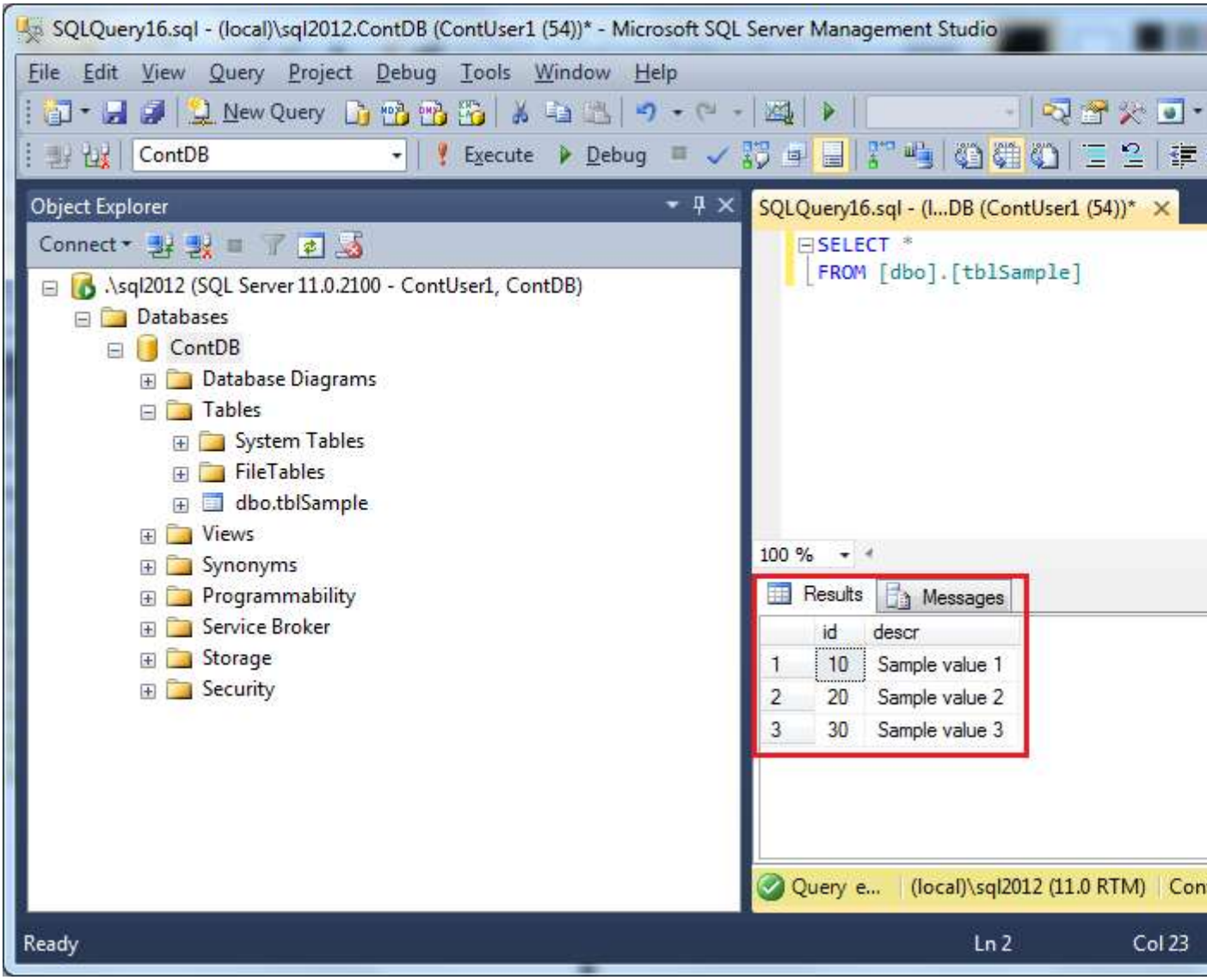
Step 2: We enter the database name to connect to and click on"Connect".



That's it! As you can see from the screenshot below, the user "ContUser1" was able to successfully connect to the SQL Server instance's database engine and has access only to the partially contained database he/she belongs to:



The last step is to run a simple query against the earlier created table just for checking out that our contained database user has access to the database's objects:



As you can see from the above screenshot the table is fully accessible.