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

a) 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:

```
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'
```

Ks. Oracle Database http://docs.oracle.com/cd/E11882_01/network.112/e16543.pdf kappale 2.
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;
```

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

Vastaus:

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

c) 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):

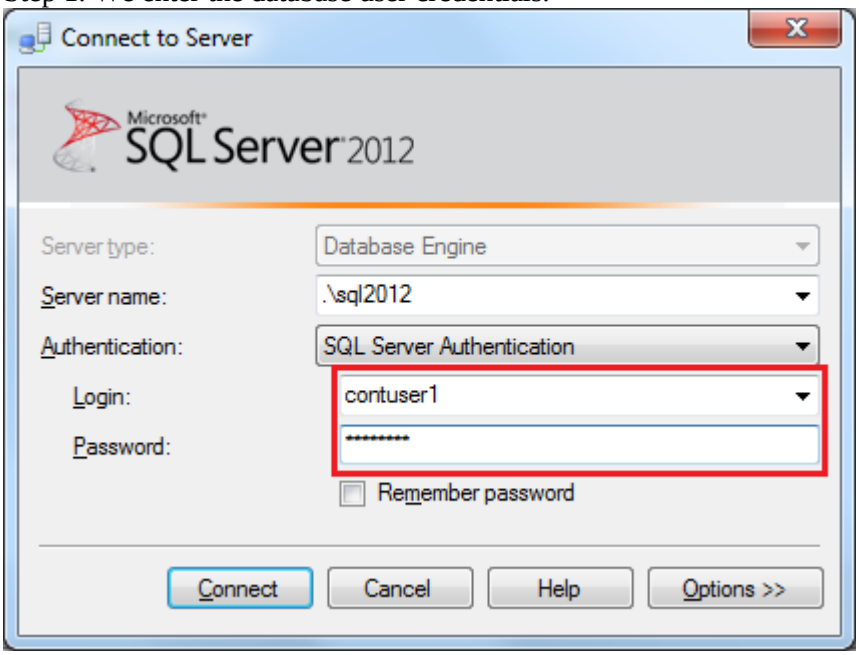
```
--Create partially contained database
USEmaster
GO
CREATE DATABASE[ContDB]
CONTAINMENT=PARTIAL
GO
--Create sample table with sample records
USE [ContDB]
GO
CREATE TABLEtblSample(
id int ,
descrvarchar (250)
)
GO
INSERT INTO tblSample
VALUES
(10,'Sample value 1'),
(20,'Sample value 2'),
(30,'Sample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

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

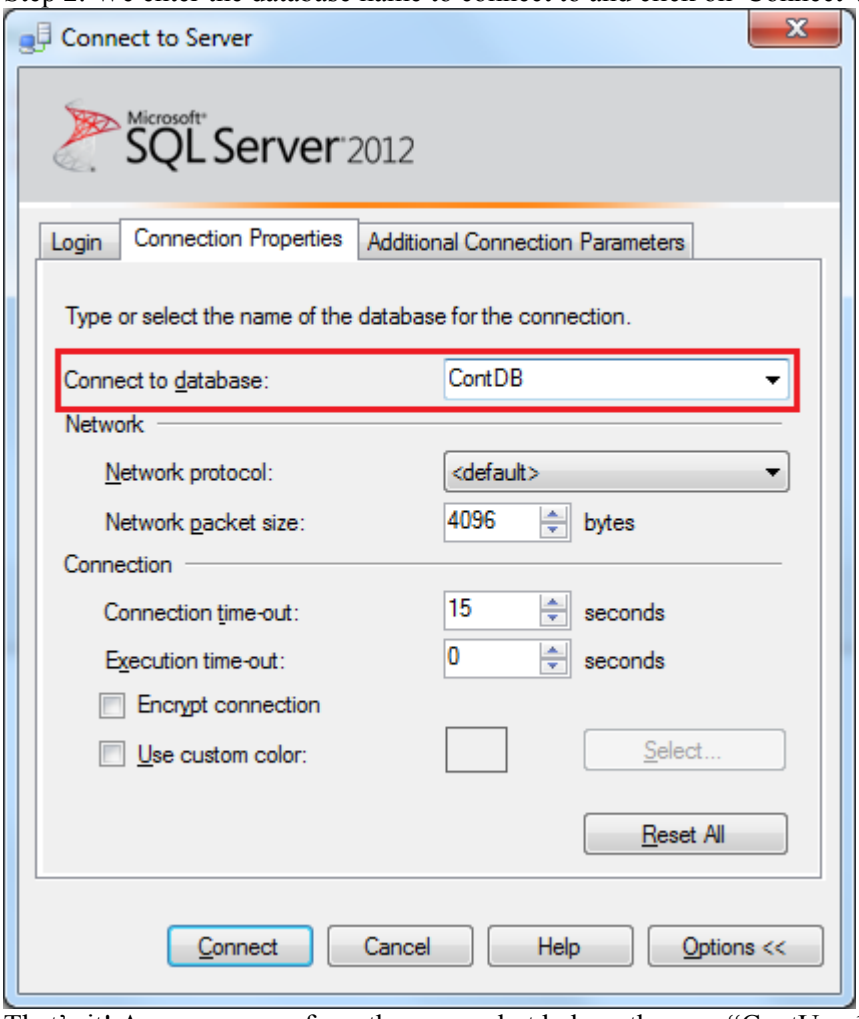
```
USE [ContDB]
GO
CREATE USER ContUser1WITHPASSWORD=N'secure1$',DEFAULT_SCHEMA=[dbo]
GO
EXEC sp_addrolemember'db_owner', 'ContUser1'
GO
```

*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":

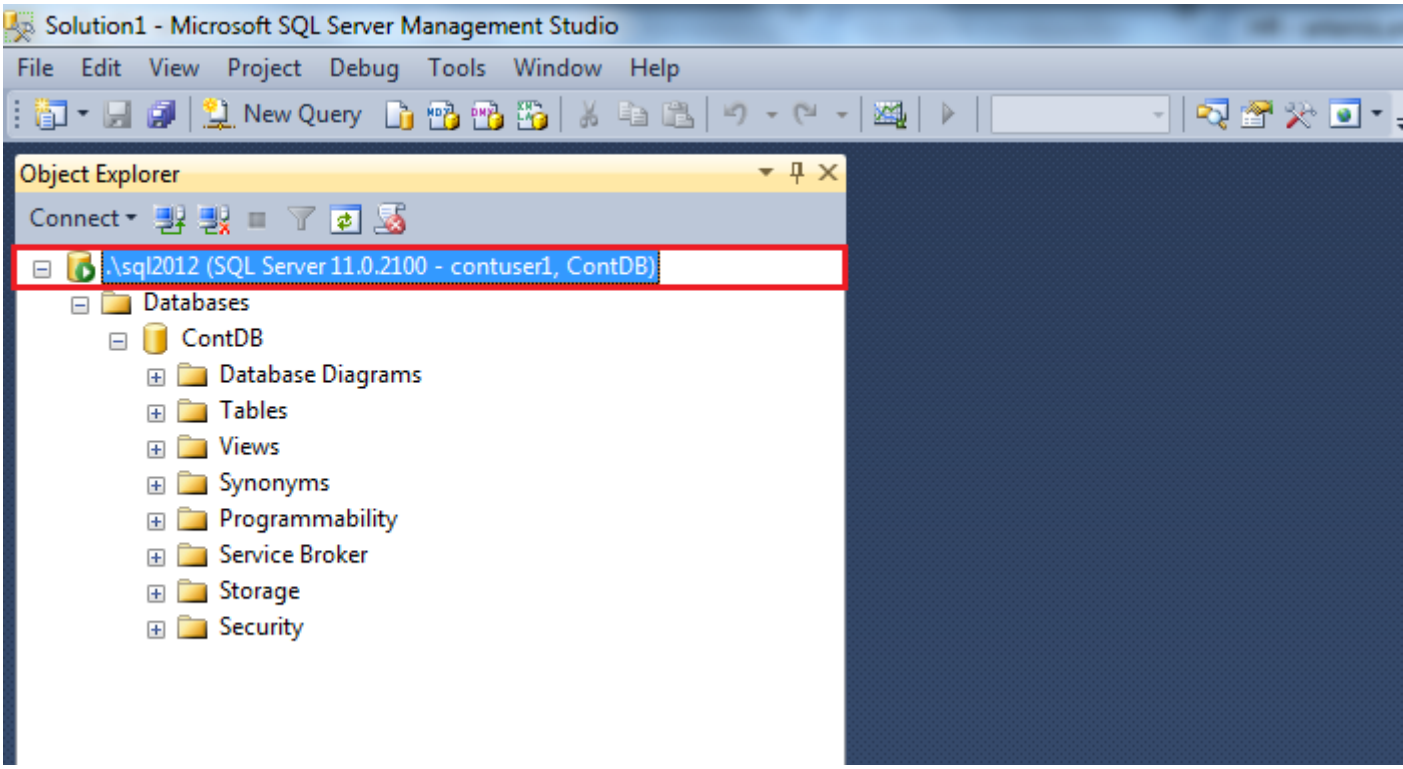
Step 1: We enter the database user credentials.



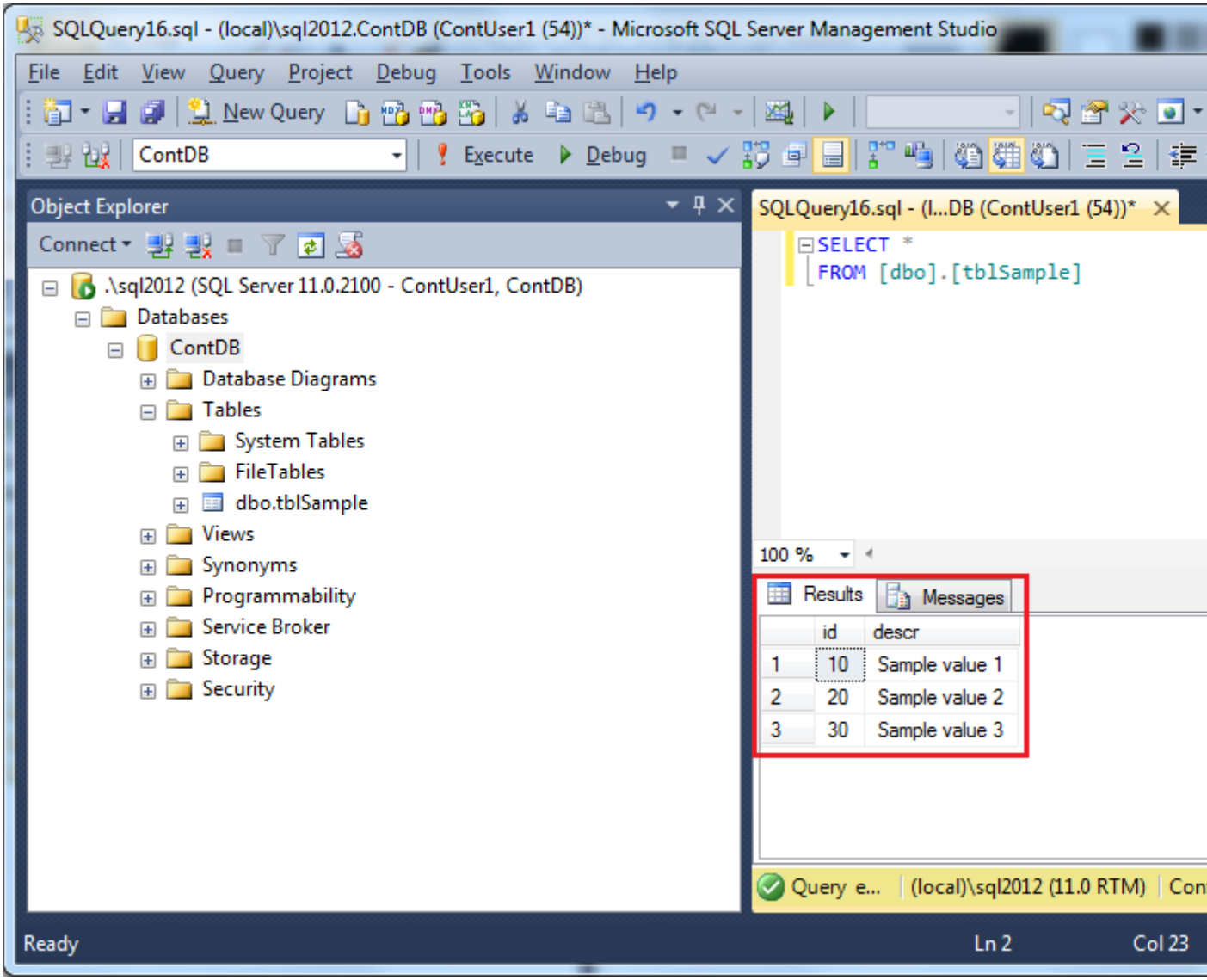
Step 2: We enter the database name to connect to and click on"Connect".



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.