

Kotitehtävät vko 2

Eheysmalli. Tietokannan luonti. Tietojen siirto tietokantojen välillä. Varmistukset. MySQL:ään tutustuminen:

- a) Mitä työkaluja MySQL:stä löytyy tietokantojen siirtoon toiseen tietokantaan (esim. SQL Server)? Entä mitä työkaluja löytyy vastaavasti SQL Server:stä?

Vastaus:

MySQL:stä löytyy esimerkiksi MySQL Workbench:stä Server Administration:in puolelta Data Export and Restore. SQL Server:stä löytyy vastaavanlaisen sql kielisten scriptien avulla tehtävän siirron lisäksi SQL Server Management Studiosta Tasks -> Import Data / Export Data.

- b) Vertaa miten SQL Server:in ja MySQL:n tietotyypit poikkeavat toisistaan.

Vastaus:

Tavallisimmat tietotyypit kuten char, varchar, date, decimal löytyvät molemmista. Kannattaa kuitenkin aina tarkistaa niiden lukualue. Esimerkiksi aikoinaan SQL Server 2005 oli tietotyyppi DATETIME, jonka lukualue oli melko suppea (alkoin 1700 kieppeiltä!). Tähän tietotyyppiin ei siis voinut tallentaa esimerkiksi historiaa liittyviä päivämääriä. MySQL:ssa tilanne oli tietysti toinen. Eli pelkkä tietotyypin löytyminen ei riitä. On vielä katsottava, että onko sen lukualue riittävä. Streamattavien datojen osalta SQL Server:in ja MySQL:n välillä on ehkä suurien ero. SQL Server:ssä voi luoda streamatun tietokannan omaan tiedostoryhmään. MySQL:ssä vastaava ei ole mahdollista vaan yleensä näkee tehtävän niin, että tietokannassa on vain videotiedoston tai muun suuren binääritiedoston nimi ja polku mistä se löytyy. MySQL:stä löytyy varbinary, mutta silloin binääritiedosto talletetaan samaan taulun tiedostoon muun tietokannan kanssa (ks. <http://dev.mysql.com/doc/refman/5.6/en/binary-varbinary.html>). Seurauksena on hitautta haussa ja myös tallennuksessa, päivityksessä jne. On nopeampaa tallentaa suuret binääritiedostot omaan tiedostoon SQL Server:in tapaan.

Muitakin eroja on, joista on kerrottu alla olevissa osoitteissa:

Ks. SQL Server: <http://msdn.microsoft.com/en-us/library/ms187752.aspx> ja

MySQL: MySQL_refman-5.6-en.a4.pdf kpl 11 (<http://dev.mysql.com/doc/refman/5.6/en/data-types.html>)

- c) Mikä on suositeltava eheysmalli eli recovery model tuotannolliselle tietokannalle esimerkiksi verkkokaupalle tai pankin tietokannalle? Millaisissa tilanteissa voisi muita eheysmalleja tarvita? Tutki asiaa SQL Server:ssä ja MySQL:ssä.

Vastaus:

SQL Server:ssä suositeltavin eheysmalli on full (ks. SQL Server: [http://technet.microsoft.com/en-us/library/ms366344\(v=sql.105\)](http://technet.microsoft.com/en-us/library/ms366344(v=sql.105))).

MySQL: MySQL:ssä ei ole SQL Server:iä vastaavia eheysmalleja, mutta MySQL:stä löytyy erilaisia database engine:jä. MySQL_refman-5.6-en.pdf kpl 7.1 (<http://dev.mysql.com/doc/refman/5.0/en/innodb-storage-engine.html>). Niistä kannattaa valita InnoDB, jota onkin käytetty mm. kotitehtävissä.

- d) Perehdy SQL Server:in dbcc lauseeseen.

Vastaus:

Transact-SQL ohjelmointikielessä on DBCC lauseet tietokannan ylläpitoon, informaation saamiseen tietokannasta, tietokannan validointioperaatioihin ja muihin sekalasiin tehtäviin ks. <http://technet.microsoft.com/en-us/library/ms188796>. DBCC lausetta voi käyttää esimerkiksi tietokannan taulujen fragmentoitumisasteen selvittämiseen:

```
dbcc showcontig('Person.Contact')
```

(ks. <http://msdn2.microsoft.com/en-us/library/ms175008.aspx>).

DBCC lauseen avulla voi tehdä myös tietokannan eheyden tarkistamisen:

```
dbcc checkdb ('tietokannan_nimi')
```

(ks. <http://msdn2.microsoft.com/en-us/library/ms176064.aspx>).

Nämä olivat vain joitain esimerkkejä DBCC lauseesta.

e) Perehdy SQL Server:in Agent:iin. Mihin sitä voi käyttää?

Vastaus:

SQL Server Agentin avulla voi automatisoida ylläpidon erilaisia tehtäviä kuten esimerkiksi varmistusten oton.

SQL Server:in Agent on Microsoft Windows palvelu, joka pystyy käynnistämään ajatettuja ylläpidon tehtäviä (tasks). Niitä kutsutaan job:ksi. Asiasta on kerrottu lisää osoitteessa:

[http://msdn.microsoft.com/en-us/library/ms189237\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms189237(v=sql.105).aspx).

f) Mitä kaikkea on ensin huolehdittava, että on kunnossa ennen kuin suorittaa varmistuksen esimerkiksi SQL Server:ssä?

Vastaus:

Tietokantojen taulut ja niiden indeksit on oltava eheitä. Niiden eheyden voi tarkistaa edellä esitellyn dbcc lauseen avulla SQL Server:ssä. Jos taulut ja indeksit eivät ole eheitä, voi varmistus epäonnistua. Tämän vuoksi on tärkeää, että niiden eheys tarkistetaan ja näin varmistutaan, että tietokantojen varmistukselle menee kunnossa oleva tietokanta.