

~~Tisdag 22 november~~ Onsdag 23 november – Hantera säkerhet, Optimera databaser

Även denna dag undervisning på distans.

Före lunch mestadels teoretisk genomgång. Efter lunch mestadels egna övningar.

Vid kl 14.30 samling för summering av dagen och möjlighet att ställa frågor.

Mer om användare, behörigheter, mm

Lite repetition från gårdagens installation, “Windows authentication mode” vs “Mixed mode”.

Begreppen “login” vs “user”.

På servernivå “logins” och “server roles”.

På databasnivå “users” och “database roles”.

Vem är jag? Testa med “select current_user”. Login “sa” är alltså mappat mot user “dbo”.

Skapa ett nytt login: create login my_login with password='Testing!';

Kolla vilka login som finns: select * from sys.sql_logins;

Skapa en ny användare: create user my_user for login my_login;

Kolla vilka användare som finns: select * from sys.database_principals;

Men kan min användare göra något i databasen? Nedanstående bör resultera i “select permission was denied”.

execute as user='my_user'; select * from Students; revert;

Först måste vi dela ut några rättigheter: grant select on database::Test to my_user;

Förkortat skrivsätt: grant control ... ger rätt att göra insert+update+delete.

Vad har jag för rättigheter? Testa med “select * from fn_my_permissions('Students', 'OBJECT');”.

En snyggare lösning hade naturligtvis varit att skapa en roll, ge rollen rättigheter, och lägga till en eller flera användare som medlemmar av rollen.

Utforska gärna på egen hand. Sök efter “create role” och “alter role” + “add member”.

Skapa index, begränsningar (constraints), mm

Introduktion till index.

Kommer någon ihåg bibliotekens gamla kortregister med alfabetisk ordning och systematisk ordning?

Index går att skapa för enstaka kolumner eller för kombinationer av två eller flera kolumner.

Poängen är att förenkla och snabba upp sökning i tabeller.

Nyttan av index är i högsta grad beroende av vad för slags frågor vi ställer till databasen.

Ska index vara i stigande eller fallande ordning?

Ett smart drag är att indexera kolumner som används vid join-operationer.

Primära nycklar blir alltid också index, men inte samma sak för främmande nycklar.

Om “select” går fortare kommer dock “insert”, “update” och “delete” att gå långsammare.

När vi ändrar i en tabell måste ju även index uppdateras.

Via till exempel “Estimated plan” i Azure Data Studio går att skapa sig bilder av hur olika SQL-frågor utförs.

Det går även att använda sig av SQL-kommandot “set showplan_xml on”.

En specialvariant av index är dem vi skapar med “unique”.

Andra slags begränsningar, eller “constraints”, vi har stött på tidigare:

- Default-värden (“default”).
- Giltiga värden (“check”).
- Obligatoriska fält (“not null”).

Använda transaktioner

Nyttan av att kunna använda transaktioner, sammanfattad med ett enkelt exempel:

```
begin transaction;  
update Students set name='Nisse' where name='Johan';  
select * from Students;  
rollback transaction;  
select * from Students;
```

Hade vi velat spara ändringarna skulle vi istället för “rollback” ha gjort “commit”.

“Allt eller inget.”

I princip ett måste att ha transaktionshantering när flera olika användare nyttjar samma databas.

Inne i en transaktion ser jag min uppdaterade eller ändrade “bubbla”. Övriga användare ser ännu inte ändringarna.

Men vad händer om två transaktioner låser varandra?

Lite överkurs: Med “save transaction” går också att sätta en “savepoint” i en transaktion.

Övningar

Om man inte är klar med gårdagens övningar går att spara dem till senare.

Testa att skapa egna login och users till din databas, och dela ut olika slags rättigheter till dem.

Pröva att skapa roller för att gruppera databas användare, och ge rollerna rättigheter, istället för att ge enskilda användare rättigheter.

Utforska olika sätt att granska “query execution plans”, och testa vilka slags index som verkar förbättra (eller försämma?) prestandan.

Ta reda på hur man kan använda funktioner i en check-constraint (till exempel för att jämföra med innevarande års årtal).

Bekanta dig med transaktionshantering.

Använd två olika klientprogram och undersök hur poster blir låsta för en användare om annan användare är inne i en transaktion.

Presentation av inlämningsuppgift

Inlämning senast fredag 2 december kl 16. I praktiken måndag 5 december kl 09.

Inlämning i pdf-format, ett samlat dokument, inga separata bilagor. Antalet sidor bör inte överstiga 20.

*”Please, do not consider the page limit as a target!
It is in your interest to keep your text as concise as possible,
since experts rarely view unnecessarily long proposals in a positive light.”*

Tänk på att hålla nere filstorleken. Om du infogar bilder, se till att komprimera dem först. Undvik tunga skärmdumpar.

Uppgiften ska redovisas individuellt. Under arbetets gång får du dock gärna ta hjälp av andra för att diskutera designval, problem mm.

Uppgiften och förutsättningarna är dessa:

Skapa en databas åt en hyresvärd i en stad som vill hålla reda på lägenheter uthyrda till personer.

En lägenhet har ett antal rum och en area i kvadratmeter.

En lägenhet är belägen på ett våningsplan och har ett ordningsnummer inom detta våningsplan (plan + nummer är unikt för varje adress).

En lägenhet har en gatuadress. På samma adress kan det finnas en eller flera lägenheter.

En gatuadress hör till ett hus. Ett hus kan ha en eller flera adresser (olika portar).

Ett hus ligger i ett kvarter, och har ett ordningsnummer inom det kvarter där det ligger. Ett kvarter kan ha ett eller flera hus.

Ett kvarter har en fastighetsbeteckning sammansatt av kommunnamn, traktnamn, blocknummer och enhetsnummer (tex “Stockholm Norrmalm 5:3”).

En person har personnummer, för- och efternamn, telefonnummer och epostadress. Telefonnummer och epostadress behöver inte vara unika.

En person kan endast hyra en lägenhet i taget. En lägenhet kan vara uthyrd till flera personer som delar på avtalet.

Ett hyresavtal gäller från ett startdatum och har en årshyra i kronor.

Om hyran ändras skapas ett nytt avtal med nytt startdatum, och det gamla avtalet förses med ett slutdatum som är dagen före det nya avtalets startdatum.

Se även uppgifterna på nästa sida för ledtrådar till vad som ska vara möjligt att få fram ur databasen.

Kom ihåg: Databasen kommer omöjligen kunna motsvara verkligheten fullständigt, men du ska åstadkomma bästa möjliga likhet med verkliga förhållanden.

Fortsättning på nästa sida ...

Inlämningsuppgift, fortsättning

Uppgiften ska lösas genom att ...

1. Rita ER-diagram. Identifiera entiteter och relationer, samt kardinaliteter. Använd helst “crow’s foot notation”.
Beskriv kortfattat hur du har tolkat kraven, och motivera de viktigaste designvalen du har gjort.
Redovisa med bild av diagram och korta förklarande texter.
2. Skapa tabeller, med ER-diagrammet som underlag. Koppla relaterade tabeller med primära och främmande nycklar.
Lägg till de regler för giltiga värden, unika index mm du anser är de viktigaste, och motivera kortfattat varför.
Redovisa med SQL-kommandon för att skapa tabeller mm, i textform, med korta förklarande texter.
3. Lägg till exempeldata, antingen genom att infoga en post i taget eller genom att importera från textfiler,
med som minimum två kvarter med vardera minst två hus, med i sin tur minst två lägenheter per hus, alla uthyrda till olika personer.
Redovisa med SQL-kommandon för att lägga till data.
4. Göra följande urval från databasen.
 1. Lista aktiva hyresgäster i bokstavsordning, med uppgifter om adresser och övriga egenskaper för de lägenheter de hyr.
 2. Lista kvartersnamn i bokstavsordning, med uppgifter per kvarter om antal hus, antal lägenheter, och totala antalet kvadratmeter.
 3. Lista uppgifter om antal kvarter, antal hus, antal lägenheter, och totala antalet kvadratmeter, grupperat per traktnamn (del av fastighetsbeteckning).
 4. Lista vad hyrorna kommer att bli nästa kalendermånad för alla icke avslutade hyresavtal.Redovisa med SQL-kommandon för urval, med vid behov korta förklarande texter.

För betyget G ska uppgifterna 1-3 och 4.1-4.2 vara utförda på ett i stora drag godtagbart sätt.

För betyget VG ska uppgifterna 1-2 dessutom vara utförda med välgrundade resonemang kring designen, samt även uppgifterna 4.3-4.4 vara utförda.