

1. Teoretiska frågor

- Beskriv kort hur en relationsdatabas fungerar.

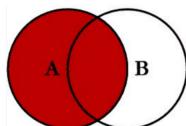
Relationsdatabas fungerar igenom att bygga relationer mellan olika informationsbitar / identifiera samma informationsbitarna utan att upprepa dem, t.ex. igenom att ha samma primary och/eller foreign keys i olika tabeller. På detta sätt sparar man plats både fysiskt igenom att använda mindre lagringsutrymme men också logiskt igenom undvika upprepningar.

EXEMPEL: T.ex. man kan lagra tusentals beställningar från tiotal kunder igenom att spara beställning/kundID information i en tabell med tusen rader, och kundID/kundaddress i annan tabell med bara tiotal rader.

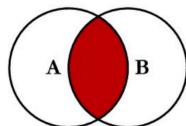
- Vad menas med "CRUD" flödet?

CRUD är en akronym som står för fyra grundläggande funktioner som används när man arbetar med databaser: **c**reate, **r**ead, **u**pdate, **d**elete data. I SQL motsvarar dessa funktioner och operationer operatorerna Insert (skapa databasposter), Select (läsa databasposter), Update (redigera databasposter), Delete (ta bort databasposter). CRUD kan ses som standard metod av att arbeta med lagrade data i vilken form som helst: inte bara rader i databasens tabeller, utan även filer, minnesinnehåll, etc.

- Beskriv kort vad en "left join" och "inner join" är. Varför använder man det?



en "left join" returnerar alla databasposter från den vänstra tabellen (tabell A), och de matchande databasposterna från den högra tabellen (tabell B).



"inner join" returnerar alla databasposter som matchar båda tabeller.

Man använder "left join" inte tappa bort databasposter från den första tabellen om de har inga motsvarigheter i den andra. T.ex. i EXEMPEL (fråga 1) kan man ha alla beställningar även för kunder utan adress. Med "inner join" kan man hitta bara beställningar med viss adress (beställningar utan adress kommer inte upp i förfrågan).

- Beskriv kort vad indexering i SQL innebär.

Indexering används för att underlätta sökning och läsning av data. Ett kolumn kan fungera som index om den har unika värden, t.ex. beställningsID. Indexering används också för att bevara datas integritet, t.ex. man kan undvika beställningar utan adress om man kräver att kundID för beställning är foreign key och som index finns i tabell med adresser till kinderna.

- Beskriv kort vad en vy i SQL är.

Vy är tabell som fylls med data från andra tabeller när den användes. T.ex. man behöver inte lagra beställningar utan adress, men man kan skapa vy av sådana för att inspektera ofullständiga beställningar. Om alla beställningar är fullständiga kommer denna vy vara tom i momang.

- Beskriv kort vad en lagrad procedur i SQL är.

En lagrad procedur är en subrutin tillgänglig att komma åt en relationsdatabas och är faktiskt lagrade i databasen. Den kan spara på nätverks trafik vid åtkomst, kan ha åtkomsträttigheter inställda, kan underlätta åtkomst till data för användare.

===== Svar till teorifrågor ska ta max 1 sida (men detta är flexibel)=====

=====Teorifrågorna ingår inte i den 2-5 sidor och räknas separat=====

=====Börja rapporten på en ny sida=====

2. Programmeringsuppgift och Rapport

1. Deskriptiv sammanfattning

AdventureWorks-databasen är ett Microsoft-produktexempel som ger ett exempel på en databas för onlinetransaktionsbehandling (OLTP). *Adventure Works Cycles* är ett fiktivt multinationellt tillverkningsföretag som stöds av AdventureWorks databas.

Databas har följande schemas: HumanResources, Person, Production, Purchasing och Sales. Till höger visas HumanResources och Person schemor som dem syns på OS X, där jag installerat 2017 version av databas, men jag också installerade 2022 versionen på Win OS och verifierade att den innehåller samma data.

▼ SERVERS	
▼	AdventureWorks2017
▼	Tables
>	dbo.AWBuildVersion
>	dbo.DatabaseLog
>	dbo.ErrorLog
>	HumanResources.Department
>	HumanResources.Employee
>	HumanResources.EmployeeDepartmentHist...
>	HumanResources.EmployeePayHistory
>	HumanResources.JobCandidate
>	HumanResources.Shift
>	Person.Address
>	Person.AddressType
>	Person.BusinessEntity
>	Person.BusinessEntityAddress
>	Person.BusinessEntityContact
>	Person.ContactType
>	Person.CountryRegion
>	Person.EmailAddress
>	Person.Password
>	Person.Person
>	Person.PersonPhone
>	Person.PhoneNumberType
>	Person.StateProvince

2. Statistisk analys (konfidensintervall)

Hypotes #1: AdventureWorks-databasen är artificiell, den andra genererade databasen som vi studerade, WideWorldImporters, hade normal fördelade försäljningsdata: fanns städer med få och vissa med mycket beställningar, men vanligtvis har man mellan 90 – 140 beställningar per staden (kommun). Jag vill utforska om AdventureWorks är genererade på liknande sätt.

Hypotes #2: Beställningar i databas förknippas till vissa personer. Man kan lista hur många anställda finns per stad och kolla om fler anställda genererar flera beställningar.

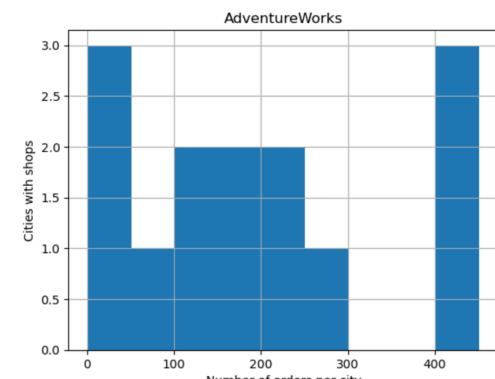
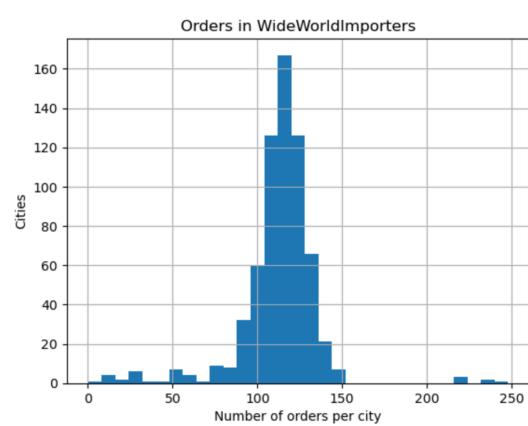
3. Slutsatser och rekommendationer

Beställningar i olika städer (kommuner) är medeltal över stora antal av beställningar över ganska slumpmässiga territorier. De borde vara normalfordelade, som i WideWorldImporters databas (se bild). Trots detta är histogram bara ojämnt. En annat forskning som jag hittade

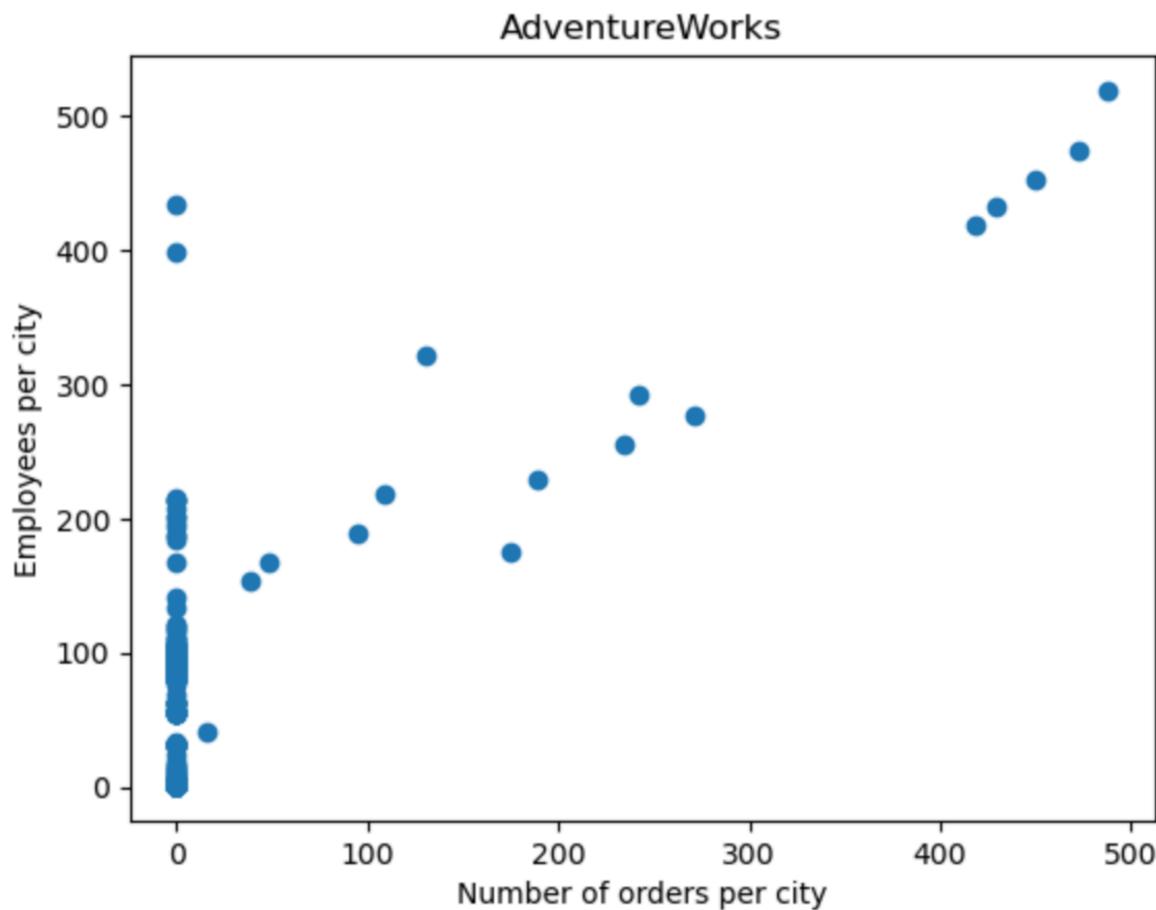
<https://www.red-gate.com/simple-talk/databases/sql-server/t-sql-programming-sql-server/visual-checks-data-distributed-sql-server/>

påvisar att även andra parametrar i databasen undviker normalfordelning. Jag tror att detta gjordes avsiktligt.

När antal anställda per staden (kommunen) studerades, förkommer det att antal beställningar är ungefärligt samma eller mindre än antal anställda. Detta är inte rimligt modell, för att anställda hus återförsäljare (enligt beskrivande AdventureWorks



Data Dictionary pdf "StoreID" parameter pekar på just återförsäljare) borde ha genererat minst visst antal försäljningar. Det ser ut som att respektive affärer säljer något annat än cyklar.



Vissa städer har anställda som jobbar med produktion, och genererar inte fördelningar direkt. Man kan se, att ungefär 150 anställda behövs för att börja generera försäljningar.

4. Executive summary

- AdventureWorks är fiktivt databas. Trots att den har olika länder och valutor, den har inte genereras med geografisk princip i åtanke. De medeltal, som bor vara helt slumpmässiga, t.ex. försäljning av olika varor från olika återförsäljare, är inte normal fördelade. De troligtvis är avsiktligt genererades för att skapa komplexa statistiska data.
- AdventureWorks är produktexempel som försöker vara inte alltför stor. Detta skapar vissa problem i modellen, t.ex. man kan notera, att antal beställningar hos återförsäljare är jämförbar eller mindre än antal anställda, som inte är rimlig.
- Vissa rimliga nyanser fångas ändå av AdventureWorks modell. T.ex. antal beställningar korrelerar med antal anställda, men inte linjärt: man behöver vissa antal anställda för att börja generera beställningar, men sedan börjar antal beställningar växa med antal anställda.

5. Datum för muntlig presentation

Torsdag den 4 januari 2024 kl. 11:00 – 11:2

==Rapporten ska vara 2-5 sidor. Det går bra att skriva mer, speciellt om man använder grafer== =====MEN skriv konkist och fundera på vad du vill lyfta fram=====

=====Börja reflektionen på en ny sida=====
=====Längden av reflektionen är ungefär 1 sida=====

3. Reflektion på eget arbete

Utmaningar du haft under arbetet samt hur du hanterat dem.

Det var inte helt klart hur klient – server relationen fungerar mellan Python och databas på teknisk sett, så att det tog tid att lista att man måste installera odbc drivrutin (Open Data Base Client) på klients (Pythons) dator och hur gör man så att Python (pyodbc module) kan se och använda denna drivrutinen.

Vilket betyg du anser att du skall ha och varför.

Jag har nått kursens mål och har demonstrerat att jag kan hantera data från SQL med Python samt arbeta och förstå SQL queries. Jag anser att jag ska ha VG betyget.

Tips du hade ”gett till dig själv” i början av kursen nu när du slutfört den.

- Detta är av naturliga skäl svårare att studera mitt i helgdagarna/lovdagarna, man borde ha jobbat mera i förväg under kurserna. Detta är någonting att tänka på före sportlov.
- Databaser som SQL är inte något jag har arbetat med i praktiken, bör ha mer praktik med saken.