# Teoretiska frågor Andreas Wendel

**1. Beskriv kort hur en relationsdatabas fungerar.**

**En relationsdatabas organiserar data i tabeller, där varje tabell består av rader och kolumner. Varje rad innehåller data och varje kolumn representerar en typ av identitet. Relationen mellan tabeller upprätthålls genom att använda nycklar. En kolumn tilldelas en primärnyckel som sedan kan användas för att sätta upp en relation till en annan tabell. Med andra ord en databas med sammankopplade relationer**

**2. Vad menas med ”CRUD” flödet?**

**CRUD står för Create, Read, Update och Delete, vilket är de grundläggande operationerna som kan utföras på data i en databas.**

**Create: Skapa nya data. Read: Läsa och hämta befintliga data.**

**Update: Uppdatera befintliga data. Delete: Ta bort data.**

**3. Beskriv kort vad en ”left join” och ”inner join” är. Varför använder man det?**

**En "left join" och "inner join" är typer av SQL operationer:**

**Inner Join: Returnerar rader när det finns matchande värden i båda tabellerna baserat på det gemensamma kolumnvärdet.**

**Left Join: Returnerar alla rader från den vänstra tabellen och matchande rader från den högra tabellen. Om det inte finns någon matchning i högra tabellen, kommer resultatet att innehålla null-värden för högra tabellens kolumner.**

**Man använder dessa operationer för att kombinera data från olika tabeller baserat på relationer och gemensamma kolumner.**

**4. Beskriv kort vad indexering i SQL innebär.**

**Indexering i SQL innebär att skapa en struktur för att snabbare söka och åtkomma data. Index används för att förbättra prestanda genom att skapa en separat struktur som länkar till de datavärden i tabellen. En typisk struktur är ett träddiagram som kan halvera sökområdet flera gånger det gör det möjligt att hitta och hämta data snabbare genom att minska mängden data som behöver sökas igenom.**

**5. Beskriv kort vad en vy i SQL är.**

**En vy i SQL är en ”virtuell” tabell som baseras på resultatet av en SELECT-fråga. Vyer används ofta för att skriva komplexa frågor och göra det lättare att hämta specifik data flera gånger om. Ditt select statement sparas då i en view som senare kan redovisas igen och tar datan från tabellerna istället för att skapa en helt ny tabell som kommer ta mer lagringsutrymme. De tillåter också att begränsa åtkomst till vissa kolumner eller rader för användare, vilket ger ett extra lager av säkerhet**

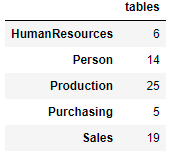
**6. Beskriv kort vad en lagrad procedur i SQL är.**

**Kort och enkelt är en lagrad procedure i SQL en sparad funktion som i andra programmerings språk som till låter dig att göra operationer med parametrar utan att behöva skriva om koden flera gånger**

# 2. Programmeringsuppgift och Rapport

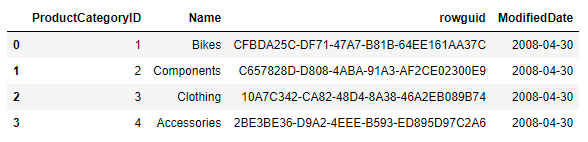
1. Deskriptiv sammanfattning

**Databasen** Adventureworks2022 Innehåller 5 Scheman med x antal tabeller.

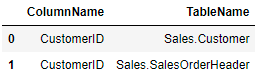


Utifrån tabellen kan vi göra ett snabbt antagande att företaget har hand om personal (HR och person) samt även producering, handel och försäljning utav produkter.

Genom att kolla sig in på produktions delen av databasen kan vi hitta en liten tabell som innehåller production.productcatagory



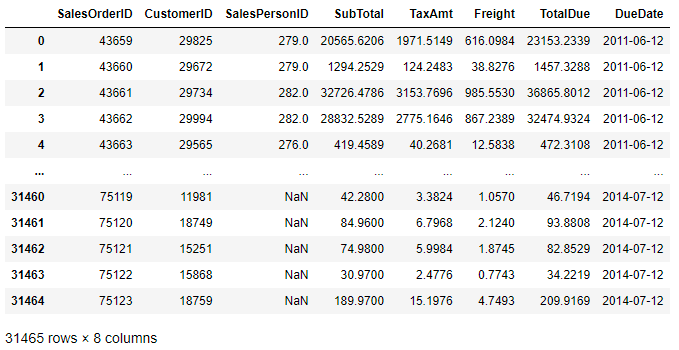
**Här** kan vi se att vi framför allt producerar och säljer cyklar, cykel komponenter samt även kläder och andra tillbehör. VI söker oss vidare i databasen men väljer att förhålla oss till personal och försäljning för simplifieringens skull eftersom det sammanlagt finns 69 tabeller att gå igenom.



Vi söker igenom databasen med en query för att hitta intressanta tabeller som tillhör CustomerID (kunder)

till svars får vi 2 tabeller som vi söker igenom

I tabellen SalesOrderHeader har vi all försäljnings data, den är ordnad per orderID och anknyter till ett customerID.

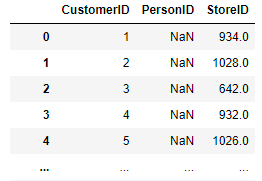


Datan innehåller 31465 rows × 26 columns, där Rows också kan ses som antal sålda beställningar och 26 kolumner med information om varje beställning.

**Tabellen** är för stor för att visa så vi väljer ut dem ”viktigaste” kolumner

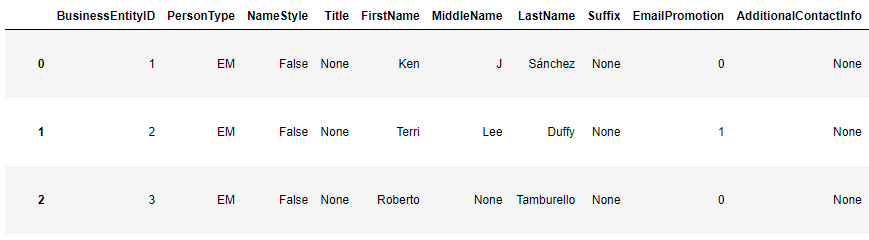
Här kan vi få all typ av data ifrån försäljning, SalesOrderID kan kopplas till SalesOrderDetail där vi ser

Specifikt vad vi har sålt, Customer ID som visar vem det är som har handlat, salespersonID som naturligtvis är försäljaren till kunden samt deras belopp kunden handlat för. Vi backar ett steg och förhåller oss till kunden och kollar vem den personen är. Vi söker härnäst igenom Sales.Customer.

**Fokusera** på kolumnen PersonID och StoreID båda dessa enligt deras beskrivning kopplar sig till BusinessEntityID men skillnaden är att PersonID kopplas till Person.Person tabellen och StoreID till Sales.Store.

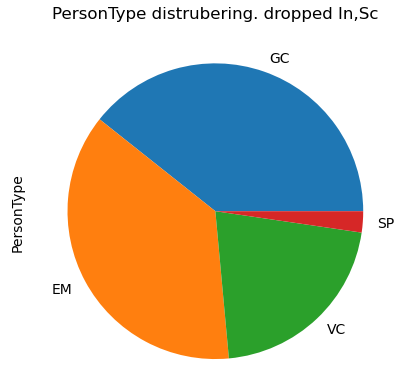
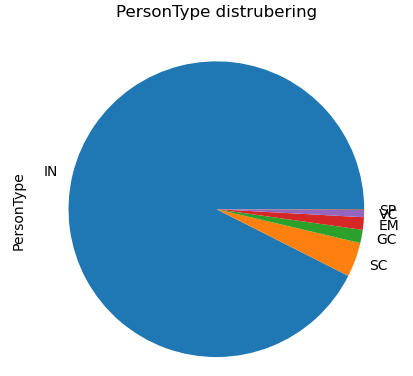
Sales.Store ger oss information om dess butiksnamn och den ansvariga försäljaren till dem.

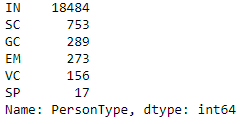
Person.Person är drastiskt mycket mer informativ för den innehåller alla människor med någon koppling till företaget vare sig det är säljare, kunder eller andra anställda.



Här får vi en del viktig information, den viktigaste är BusinessEntityID som kopplar näst in till alla personer till någon typ av ID. Förutom namn så har vi även PersonType som står för vad denna personen har för anknytning till företaget. Vid kolumnens beskrivning så får vi denna förklaringen:

**Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Sales person,**

**EM = Employee (non-sales), VC = Vendor contact, GC = General contact**.Vi kan med hjälp av matplotlib visa hur den distruberingen ser ut bland PersonType

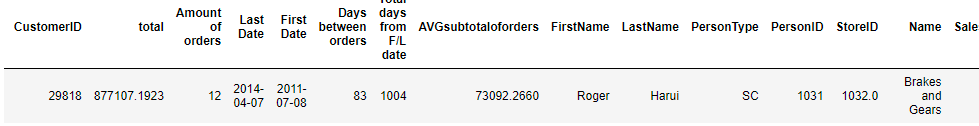


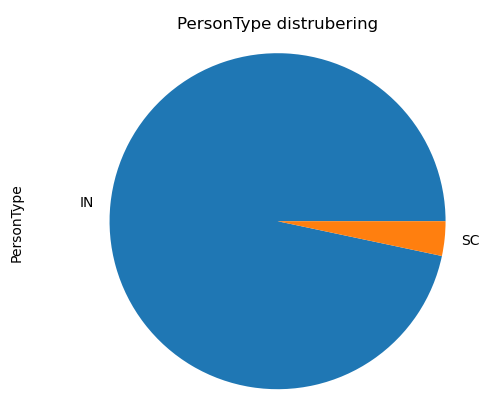
I person.person tabellen består mesta dels utav kunder IN och SC kunder. Utav självklara anledningar så är det inte så konstigt, majoriteten kommer självklart vara Data om kunder, men vi kan dra slutsatsen att det jobbar 290 människor (EM+SP) och om man räknar bort våra kunder så består den andelen av anställda av 40% (290/735) utav de resterande kontaktpersonerna.

2.Statistisk analys

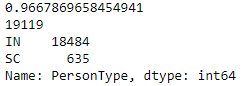
**Tesen** för denna analysen kommer handla om försäljning och analysering av kunder samt i syfte om att ”flagga” kunder som spenderar mer än andra och som kan hjälpa oss optimera försäljning.

Vi går tillbaka till SalesOrderHeader och bygger ihop en lite mer komplex query så vi kan se vilka det är som handlar hos oss. Den nya datan är nu grupperad per CustomerID och deras information istället för per order. Nu har vi tillgång till hur mycket en kund har beställt, total spenderat, fullständinga namn, PersonType, om kunder handlar för en butik så får vi även deras namn och ansvariga försäljaren samt även en kalkylering på vad kundens spenderar i genomsnitt per order. Dessutom så har vi fått ut 19119 rader vilket betyder att vi har 19119 kunder som handlat hos oss

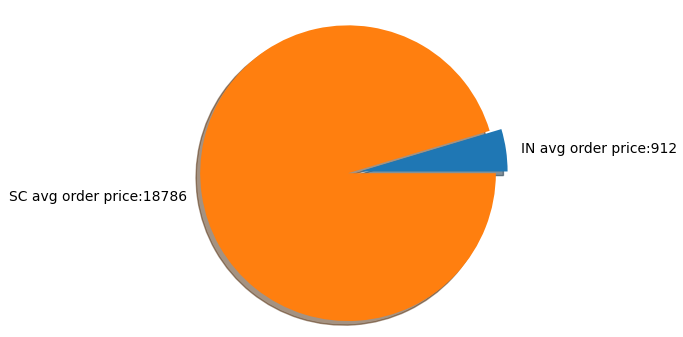
 OBS (alla kolumner syns ej, de som saknas är SP information.

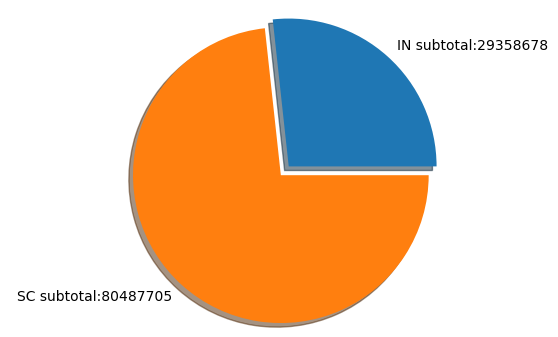


**Om vi kollar på** denna nya Sales datans distribution bland Persontype så ser vi att majoriteten av kunder hos oss är individuella kunder, Man kan anta att detta är kunder som är hobby entusiaster eller liknande som inte köper för att återsälja i affärer. Och dessa kunder består av hela 96,6% av alla kunder.

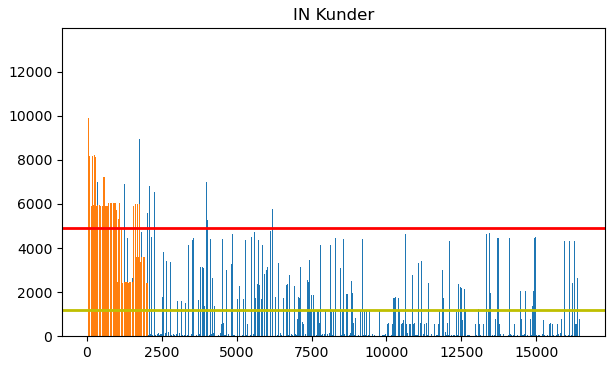
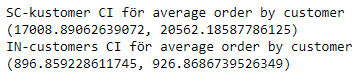


Något o anmärka är att trots att det finns 735 SC personer i Person.person tabellen så är det bara 635 som gjort affärer hos oss men varenda IN kund har handlat. Kan det vara så att IN kunder har hittat oss och SC kunder är dem vi söker efter och försöker sälja till?

**Vi kollar lite** på vad vi har tjänat på dessa på kategorier, vi har som mål att flagga stor spenderare så vi kommer på ett mått. Vi börjar med att dela upp Kunderna i sina PersonTyper eftersom man tenderar o spendera mer om man bulk handlar till en affär jämfört med en Individuell handlare.

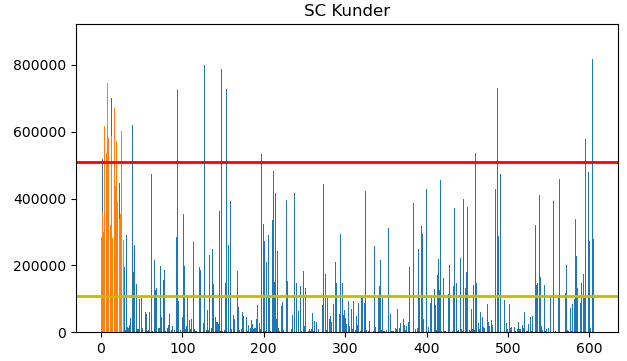


Datan är nu uppdelad och vi kan se att trots den låga 3% andelen SC kunder så står dem nästan för 75% av alla intäkter. Vi kan också se medelvärdet för varje kunds order, alltså vad en kund i genomsnitt spenderar per order. Även här kan vi se den drastiska skillnaden och anledning till detta är att beställningar hos SC kunder tenderar att vara produkter i bulk eller i större sortiment.

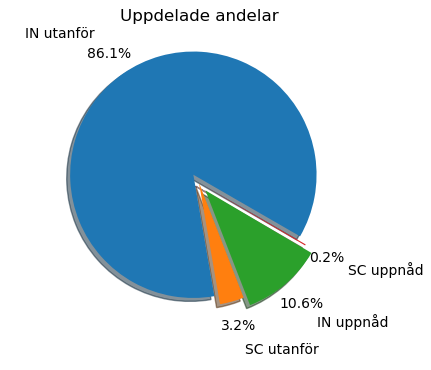
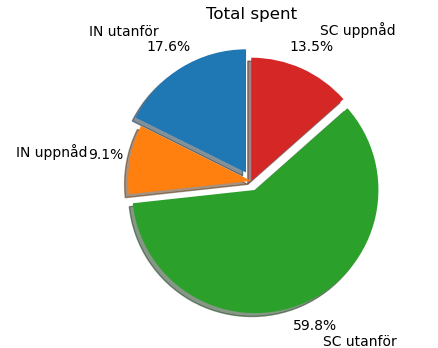
**Något att tänka** på är att datan är ifrån 2011 till 2014 så vi kan göra ett konfidens intervall för att skatta va det ”riktiga” medelvärdet är bland dagens kunder. Konfidens intervallet är beräknat med 95% alpha och ett stickprov på 635 Sc kunder och 18484 IN kunder.

**I tesen** nämner vi att vi vill Flagga potentiella ”Stor Köpare” och kommer på ett kriterium för vilka som skall va de.

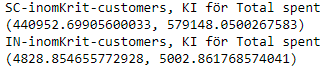
Tesen förlitar också väldigt mycket på att en kund som har spenderat ofta och mycket tidigare kommer att spendera en liknande summa vid ett senare tillfälle igen.

Vi gör så att vi tar ut dem top 20 Kunder som har spenderat mest hos oss, Sedan drar vi ett genomsnitt på vad alla kunder brukar spendera per order, på så får vi ett lite mer komplex kriterium än att ta dem mest spenderande.

En kund kan handla ofta men kanske inte så mycket, genomsnittet för IN respektive SC kunder är 912$, 18786$, så om vi istället letar efter kunder som Spenderar mer per order så har vi ett enklare sett att tidigt flagga dem. Medelvärdet för spendering per order hos de top 20 är 2402,3 resp. 68756,4$.

**Här kan vi** se skillnaden mellan totalt spenderat mellan dem inom(orange) och utanför (blå)kriteriet, röd är medelvärdet för total spenderat hos dem innanför kriteriet och gul för dem utanför kriteriet. Över lag kan vi se en drastisk skillnad.

Slutligen hänvisar jag till att cirka 11% uppnår till våra krav men dem 11% står för 22,6% av våran inkomst.

**Slutligen gör vi ett Konfidens intervall** totalt spenderat på dem som uppnått våra kriterier för att få en uppfattning på var dagens kunder kommer ha spenderat.

OBS. KI är normalt sätt beräknad som en normalfördelning men SC-inomkrit är beräknad från en t-fördelning

3. Slutsatser och rekommendationer

Vi har kommit fram till att majoriteten av våra kunder är individuella köpare men trots detta så står butiks återförsäljare för 75% av våra intäkter, vårat syfte med våran analys är att ”Flagga” potentiella stor spenderare, vi har utgått ifrån att personer som handlar ofta har en tendens att handla igen, men istället för att kolla efter antal beställning så kollar vi på vad en kund i genomsnitt spenderar per order.

**Nackdelen** med detta sättet är att vi kan få enskilda kunder som handlar stort en enstaka gång.

Med de här kriterierna så kan vi beräkna med att folk som uppnår dessa kriterier kommer spendera i genomsnitt mellan 4.8k-5k resp. 440k-579k.

**Något att konstatera** är ju att självklart om dessa kunder vi analyserat nu fortsätter att handla så kommer siffrorna att ändra speciellt hos totalt spenderat men antagandet är att dem som brukar lägga denna mängden per order kommer att fortsätta spendera lika mycket vid nästa order, och i slut resultat kommer dem fortfarande bestå utav 22,6% utav våra totala intäkter om vi fortsätta att växa i samma takt.

**Rekommendationerna** jag skulle säga är kortfattat att lägga mer tid och energi på dem kunder som uppnår dessa krav, åter igen hänvisar jag till att en kund som har handlat innan kan vilja handla igen för en liknande summa. Och på så sätt kan vi optimera våra försäljningar utan att lägga resurser på att leta efter nya kunder eller försöker sälja till kunder som inte bidrar till högre inkomster.

Fördelen med denna typ av mått är att vi kan filtrera bort kunder som skulle kunna klassificeras som ”shopaholics ” eftersom många små köp inte behöver resultera i ett stort belopp. Vi letar hellre efter kunder som spendera mycket per order i ”förhoppning” om att dem skulle vilja spendera en liknande summa igen om dem har varit bekväma med förut. Och i slutliga resultatet så kommer den 22,6% andelen av alla intäkter att öka teorin stämmer.

4. Executive summary

Utifrån vår analysering ser vi att majoriteten av kunder som står för störst del av våra intäkter är Butiks ägare, vi har dessutom lagt ett mått på dem kunder som spenderar mest hos oss och rekommenderar att lägga mer tid och resurser hos dem kunder som kan vara villiga att lägga liknande stora summor igen för att optimera våran försäljning. Vi har data för dessa kunder samt deras namn och annan typ av information och rekommenderar att hänvisa försäljarna den infon så dem kan fokusera på våra potentiella ”Stor spenderare” för att öka tillväxten hos företaget.

5. Datum för muntlig presentation : 2024-01-04 : 15:00

# Reflektion på eget arbete

1. Utmaningar du haft under arbetet samt hur du hanterat dem.

Den största utmaningen var egentligen att försöka begripa vad man egentligen ville göra, det finns säkert 100 sätt att analysera databasen och alla får ut olika resultat och ider. Jag valde egentligen bara det som såg mest intressant ut för mig, men det kom också med sina egna problem. 1. Hur man skall gå tillväga, 2. Vad man vill leta efter och vad som egentligen behövs, 3. Samt att även inte hacka upp sig på små detaljer och att inte snöa in sig för mycket. Orinalet för min analys innehåller nästan dubbla materialet men allt fick ej plats så man får försöka hålla det så konkret som man kan. 5 sidor lät mycket i början men man kom snabbt till 4 sidan och insåg att man har för mycket material.

Skulle nog också försökt strukturera arbete på ett bättre sätt

1. Vilket betyg du anser att du skall ha och varför.

Kursen handlar om SQL och själv hade jag utan problem sagt att jag kan hantera en databas och utföra CRUD operationer utan problem vare sig det är i ssms, python eller någon ny utvecklar miljö så för första gången ett VG,

Men med det sagt min magiska halv konkreta analys så i alla fall ett Godkänt

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

Läs uppgiften noggrant och planera först vad som skall göra, anteckna det viktiga och jobba utifrån en plan, sen programmera och sök den datan man vill ha, jag gjorde det här lite tvärt om nu och fick om organisera x antal gånger

Hoppas allt är läsbart i värsta fall kontakta mig //mvh Andreas Wendel