

Högskolan i Halmstad  
Sektionen för ekonomi och teknik  
Affärssystemprogrammet  
Databashantering och beslutsstöd, 7,5 hp  
Examinator Jesper Hakeröd  
2011-02-25

# Databashantering och Beslutsstöd

Namn

## Innehållsförteckning

1. Inledning.....	2
1.1 Syfte .....	2
1.2 Mål .....	2
2. Projekt .....	3
2.1 Data Dictionary .....	3
2.2 Star schema.....	4
2.3 Flödesschema .....	4
2.3.1 Stagingtabell.....	5
2.4 Mapping och härledda attribut.....	6
2.5 SQL- frågor .....	7
2.6 Stored procedure.....	9
2.7 Vyer.....	9
2.8 Rapporter.....	11
3. Reflektion .....	13

# **1. Inledning**

## **1.1 Syfte**

Syftet är att få en förståelse för processerna och hur ett data warehouse byggs upp från grunden. Vi ska även formulera frågor och hitta tabeller i Jeeves som våra frågor ska kopplas till.

## **1.2 Mål**

Målet med detta projekt är att lyckas med alla fyra labbar och att hela tiden ha en förståelse för vad som görs. Ett delmål som vi har är att vara i fas vid varje labbtillfälle så att vi inte hamnar efter. Ett annat delmål är att inför delredovisningen vara klara med laboration ett och två för att kunna presentera projektet på ett bra och överskådligt sätt. Slutmålet är att allt ska vara klart inför slutredovisningen och inget ska vara halvfärdigt.

## 2. Projekt

### 2.1 Data Dictionary

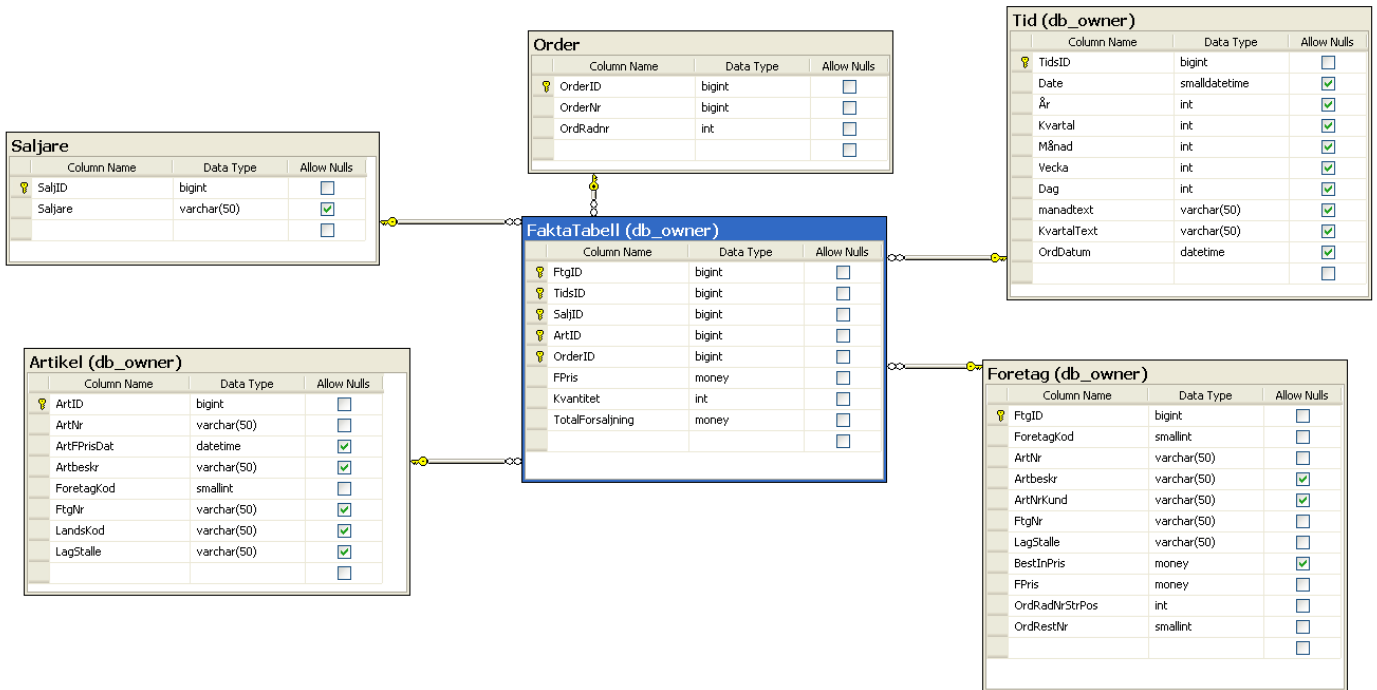
Data dictionary är ett dokument som beskriver databasen och dess relationer. Data dictionary innehåller databasens olika objekt och vilka attribut som ingår. Den innehåller även en kort beskrivning av attributen, datatyper och om null får existera.

Entity name	Attributes	Description	Data type & Length
Företag orp	FtgID	Identifieringsnummer för företag	bigint
	ForetagsKod	Numerisk beteckning på det egna företaget.	Smallint
	ArtNr	Artikelnummer	Varchar (50)
	Artbeskr	Beskrivning av artikel/produkt/vara	Varchar (50)
	ArtNrKund	Kunds artikelnummer	Varchar (50)
	FtgNr	Beteckning för företaget	Varchar (50)
	LagStalle	Kod för lagerställe	Varchar (50)
	BestnPris	Inköpspris på aktuell beställning	Money
	FPris	Försäljningspris	Money
Säljare	OrdRadNrStrPos	Löpnummer för orderrad till artiklar	Int
	OrdRestNr	Ett restordernummer	Smallint
Säljare	SaljID	Identifieringsnummer för säljare	Smallint
	Saljare	Beteckning för säljare	Varchar (50)
Artikel ar	ArtID	Identifieringsnummer för artikel	bigint
	ArtFPrisDat	Datum då försäljningspris senast sattes	datetime
	Artbeskr	Beskrivning av artikel/produkt/vara	Varchar (50)
	ArtNr	Artikelnummer	Varchar (50)
	ForetagsKod	Numerisk beteckning på det egna företaget	Smallint
	FtgNr	Beteckning för företaget	Varchar (50)
	LandsKod	Förkortning på landet	Varchar (50)
	LagStalle	Kod för lagerställe	Varchar (50)
Tid	TidsID	Datum för order	bigint
	Date	Visar datum	smalldatetime
	År	Året	Int
	Kvartal	Kvartalet	Int
	Månad	Månaden	Int
	manadtext	Månad i text	Varchar (50)
	kvartaltext	Kvartal i text	Varchar (50)
	OrdDatum	Orderdatum	datetime
	Vecka	Veckan	Int
Dag	Dagen	Int	
Order	OrderID	Identifieringsnummer för order	bigint
	OrderNr	Ordernummer i numeriskt format	bigint
	OrdRadNr	Radnummer inom order eller offert/avtal	Int
Stag	orderantal	Antal order	Int
	fpris	Försäljningspris	Money
	radsumma	Antal order gånger försäljningspris	Money
	ftgnr	Beteckning för företaget	Varchar (50)
	artnr	Artikelnummer	Varchar (50)
	saljare	Beteckning för säljare	Varchar (50)
	orddatum	Orderdatum	smalldatetime
	ordernr	Ordernummer i numeriskt format	bigint
	ordradnr	Radnummer inom order eller offert/avtal	Int

Syftet med vårt data dictionary var att få en överblick över våra valda objekt. Det var ett hjälpmedel för oss när vi senare skulle skapa tabeller eftersom det var enkelt att gå tillbaka och se vilken datatyp ett attribut skulle ha. Ett problem som uppstod under arbetets gång var att vi hade skrivit in fel datatyper i vårt data dictionary. Det visade sig att vi inte hade tagit datatyperna från Jeeves utan från hjälppilen. Vi lärde oss att det är viktigt att vara noggrann redan från början och att inte slarva utan hela tiden kontrollera det vi gjort.

## 2.2 Star schema

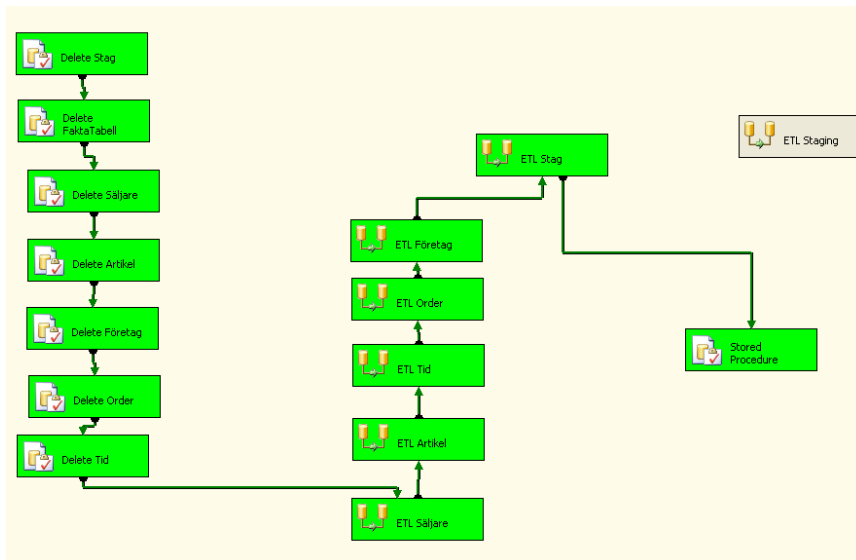
Vi använde oss av en datamodell som heter star schema som är vanligast att använda. Den består av en huvudtabell som kallas faktatabell och runt den finns olika dimensionstabeller. Faktatabellen består av de mest centrala attributen, som behövs för att fatta rätt beslut och innehåller främmande nycklar som är kopplade till de olika dimensionerna. Från star schemat kan rapporter tas ut som kan visa försäljningsvolymen och produktionskostnaden. I dimensionstabellerna finns endast de attribut som är relevanta och förklarar hur data ska sammanfattas i faktatabellen.



I vårt star schema har vi valt dimensionerna säljare, företag, artikel, order och tid för att kunna få svar på de frågor vi hade. För att kunna koppla de olika dimensionerna till faktatabellen var vi tvungna att skapa surrogat nycklar, som är påhittade nycklar med automatisk räknare. Genom att göra detta underlättade det när vi skulle göra vår ETL- process. De problem som uppstod var att vi inte kunde få flera primärnycklar i faktatabellen men lösningen var lätt. Det var bara att gå in i faktatabellen och markera alla genom att trycka på Ctrl- tangenten. Ett annat problem vi hade var att vi satte attributen i faktatabellen till att tillåta null, vilket gjorde att vi inte fick ut värden.

## 2.3 Flödesschema

I ETL- processen ingår tre steg; extract, transform och load. I extract- fasen konsolideras data från en eller flera databaser. I transform- fasen konverteras datan från den tidigare formen till en ny form som den behöver vara i för att kunna placeras in i data warehouse. I load- fasen läggs datan in i data warehouse. Syftet med ETL- processen är att ladda data warehouse med integrerad och rensad data för att senare kunna plocka ut sina önskade data.



Innan ETL- processen kan starta måste vi rensa alla tabeller för att inte gammal data ska finnas kvar för att sedan kunna få ut vår önskade data. När alla tabeller är rensade är det dags för ETL- processen för respektive dimension. I vardera ETL- process i vårt flödesschema konsolideras, konverteras och laddas data in i tabeller i data warehouse.

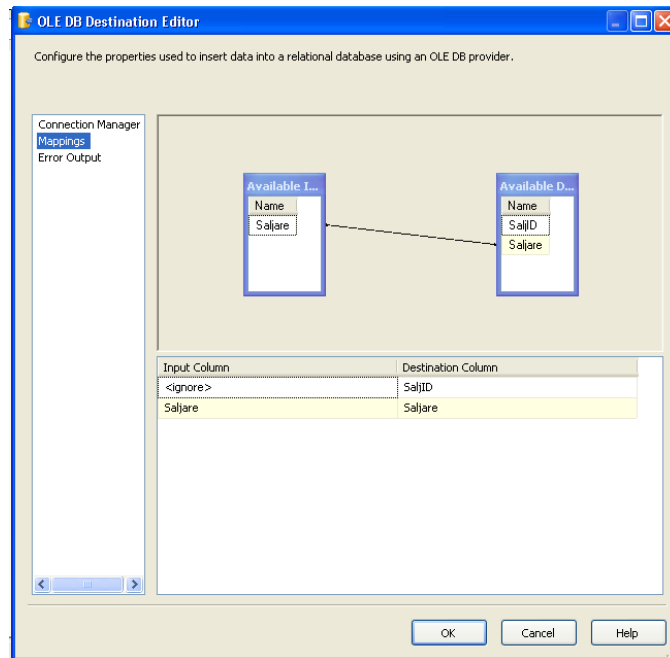
### 2.3.1 Stagingtabell

En stagingtabell är en övergångstabell innan datan förs in i data warehouse.

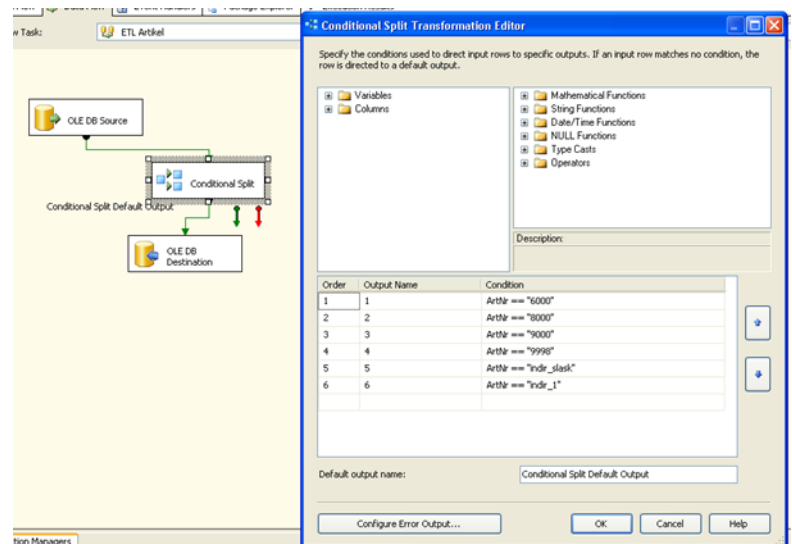
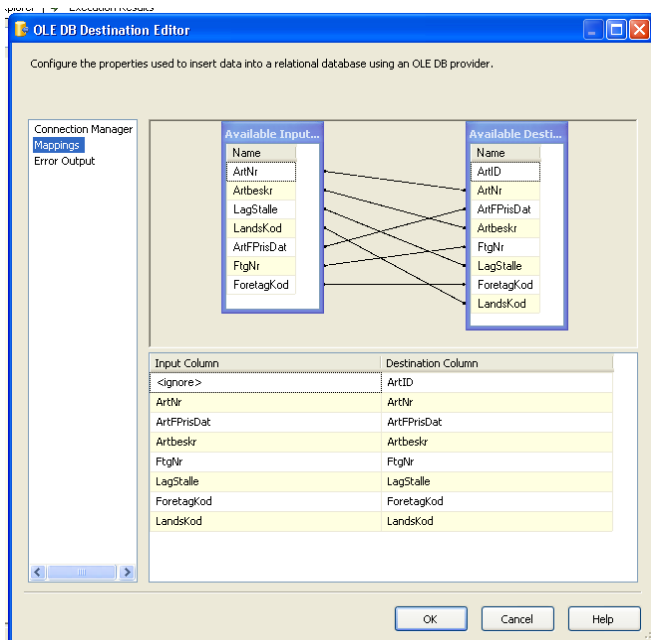
Column Name	Data Type	Allow Nulls
ordantal	int	<input checked="" type="checkbox"/>
fpris	money	<input checked="" type="checkbox"/>
radsumma	money	<input checked="" type="checkbox"/>
ftgnr	varchar(50)	<input checked="" type="checkbox"/>
artnr	varchar(50)	<input checked="" type="checkbox"/>
saljare	varchar(50)	<input checked="" type="checkbox"/>
orddatum	smalldatetime	<input checked="" type="checkbox"/>
ordernr	bigint	<input checked="" type="checkbox"/>
ordradnr	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Det största problemet med flödesschemat var att stagingtabellen inte hämtade några värden. Efter flera försök med att ändra den ursprungliga tabellen löste vi problemet genom att göra en ny stagingtabell. Den nya stagingtabellen liknade den första med undantag för ID- attributen som inte användes och därmed togs bort. Vi förstår fortfarande inte vad det var som var fel med den första stagingtabellen eftersom den andra fungerade direkt och ID- attributen inte hade någon påverkan.

## 2.4 Mapping och härledda attribut

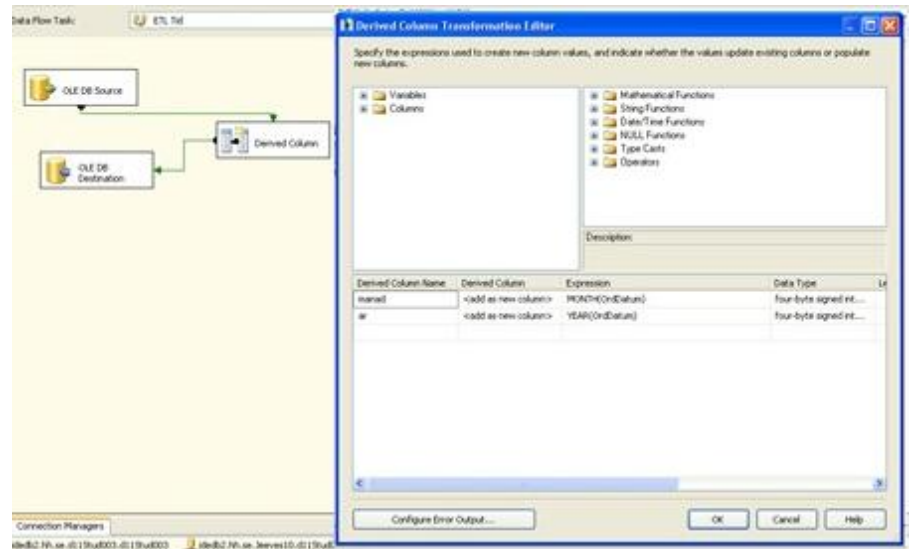
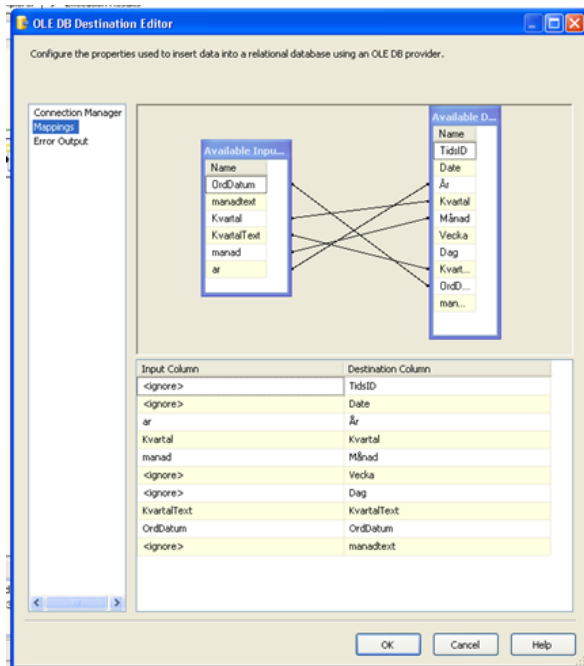


Mapping görs för att kunna få ut värden i tabellerna och behöver matchas från Jeeves databas till vår databas. Säljare i vår databas måste ha samma innebörd som säljare i Jeeves databas för att värdena ska hittas. På samma sätt har vi gjort med objekten order, staging och företag, det vill säga att vi kopplat ihop attribut med samma namn.



I mappningen för objektet artikel hade vi problem eftersom vi hade artiklar som innehöll null. För att kunna se vilka artiklar som inte hade något värde, var vi tvungna att göra en preview. Den skapar en vy på hur tabellen ser ut och kan därmed se vilka artiklar som kunde tas bort.

Genom att använda oss av conditional split kunde vi lösa problemet och rensa bort de artiklar som innehöll null.

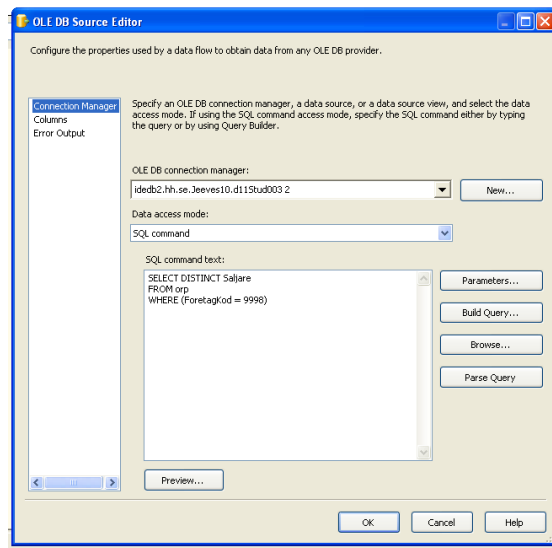
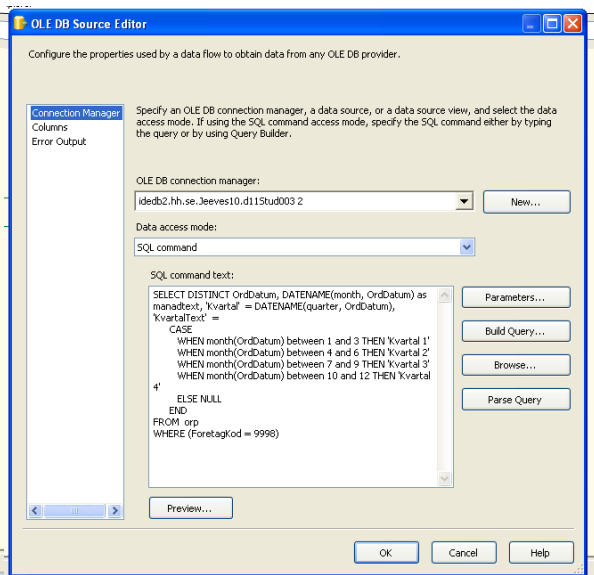
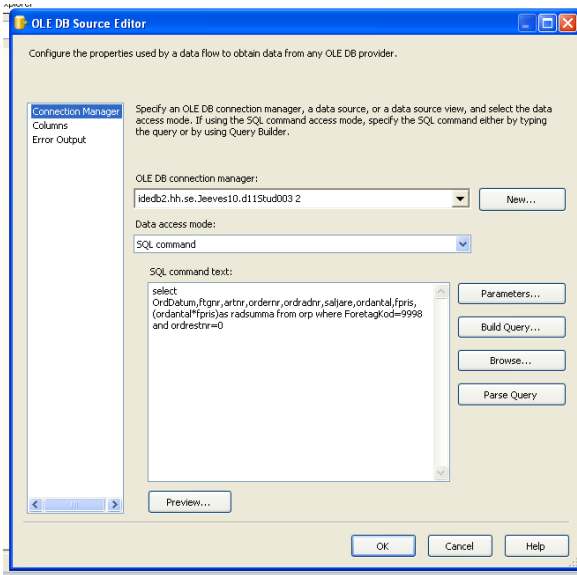


I tidsdimensionen hade vi problem att få fram år och månad som värden. Det fick vi hjälp med att lösa genom att använda derived column som lägger in månad och år i Jeeves, eftersom det inte fanns från början. Vi hade även problem med manadtext som inte ville kopplas, det är ett problem vi inte löst eftersom det inte gick i den gamla versionen.

## 2.5 SQL- frågor

SQL- frågor hämtar värden från Jeeves databas och lägger dessa värden i våra tabeller. I vår stagingtabell härledde vi en formel för radsumma, den går ut på att ta försäljningspris gånger orderantal. I tidstabellen definieras kvartal genom en färdig kod som vi fick av Jesper Hakeröd. Attributet orderdatum innehåller inte kvartal, utan vi måste själva definiera det genom koden. Koden går ut på att de tre första månaderna är kvartal ett, nästa tre månader är kvartal två etcetera.





## 2.6 Stored procedure

```

USE [d11Stud003]
GO
/***** Object:  StoredProcedure [dbo].[spSELLFACT]    Script Date:
02/23/2011 10:29:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[spSELLFACT]
AS
DECLARE @TidsID AS bigint
DECLARE @FtgID AS bigint
DECLARE @OrderID AS bigint
DECLARE @ArtID AS bigint
DECLARE @SaljID AS bigint

DECLARE @FPpris AS money
DECLARE @ordantal AS int
DECLARE @radsumma as MONEY
DECLARE @FtgNr AS varchar(50)
DECLARE @ArtNr AS varchar(50)
DECLARE @Saljare AS varchar(50)
DECLARE @OrdDatum AS smalldatetime
DECLARE @OrderNr AS bigint
DECLARE @OrdRadnr AS int

DECLARE SaljCursor CURSOR FOR
SELECT
FPpris,ordantal,radsumma,ArtNr,FtgNr,OrdDatum,Saljare,OrderNr,OrdRadnr
FROM d11Stud003.dbo.stag
OPEN SaljCursor
FETCH NEXT FROM SaljCursor INTO
@FPpris,@ordantal,@radsumma,@ArtNr,@FtgNr,@OrdDatum,@Saljare,@OrderNr,@OrdRa
dnr
WHILE @@FETCH_STATUS = 0
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;
-- Kontroll av att datum inte finns för de rader vi inserar i
TID
-- IF NOT EXISTS(SELECT Datum,artNr,ftgNr FROM SALJ WHERE
Datum=@datum AND artNr=@artNr AND ftgNr=@ftgNr)
BEGIN

SELECT @TidsID=TidsID FROM db_owner.Tid WHERE OrdDatum=@OrdDatum
SELECT @FtgID=FtgID FROM db_owner.Foretag WHERE @FtgNr=FtgNr
SELECT @ArtID=ArtID FROM db_owner.Artikel WHERE @ArtNr=ArtNr
SELECT @SaljID=SaljID FROM Saljare WHERE @Saljare=Saljare
SELECT @OrderID=OrderID FROM [Order] WHERE OrderNr=@OrderNr AND
OrdRadnr=@OrdRadnr

-- Här inseras datum som inte redan existerar i dimension TID
INSERT INTO db_owner.FaktaTabell
(TidsID,FtgID,OrderID,ArtID,SaljID,FPpris,Kvantitet>TotalForsaljning)
VALUES(@TidsID,@FtgID,@OrderID,@ArtID,@SaljID,@FPpris,@ordantal,@
radsumma);
END
-- Fångar nästa rad

```

## 2.7 Vyer

En vy visar resultatet av en SQL- fråga som ställs genom att skriva kod eller grafiskt. I detta projekt valde vi att ställa frågorna grafiskt, det vill säga genom att hämta tabeller som ska användas och kryssa i attribut som är relevanta. Detta är ett enklare sätt att få fram resultatet som önskas.

The screenshot shows the Microsoft Access interface. At the top, there are three table design grids: 'Artikel (db\_ow...)', 'FaktaTabell (db\_...)', and 'Tid (db\_owner)'. Below these is a query design grid with columns for Column, Alias, Table, Output, Sort Type, Sort Order, Group By, and Filter. The query is named 'TOP (10) db\_owner.Tid.Månad, SUM(db\_owner.FaktaTabell.TotalForsaljning) AS Total'. The result grid shows the following data:

Månad	Total
1	2006705,8000
2	1658776,0000
3	1601741,5000
4	1583944,7500

Här har vi fått svar på frågan ”Hur stor är försäljningen per månad under 2006?”. För att få månaderna i ordning använde vi oss av Ascending, vilket innebär att månaderna sorteras från 1 till 12.

Column	Alias	Table	Output	Sort Type	Sort Order	Group By	Filter
FtgID		Foretag (d...	<input checked="" type="checkbox"/>			Group By	
TotalForsaljning	Kunder	FaktaTabell...	<input checked="" type="checkbox"/>	Descending	1	Sum	
År		Tid (db_ow...	<input checked="" type="checkbox"/>			Group By	
År		Tid (db_ow...	<input type="checkbox"/>			Where	= 2007

```

SELECT TOP (5) db_owner.Foretag.FtgID, SUM(db_owner.FaktaTabell.TotalForsaljning) AS Kunden, db_owner.Tid.År
FROM
  db_owner.Foretag INNER JOIN
    db_owner.FaktaTabell ON db_owner.Foretag.FtgID = db_owner.FaktaTabell.FtgID INNER JOIN
      db_owner.Tid ON db_owner.FaktaTabell.TidsID = db_owner.Tid.TidsID
WHERE (db_owner.Tid.År = 2007)
GROUP BY db_owner.Foretag.FtgID, db_owner.Tid.År
ORDER BY Kunden DESC
  
```

Frågan vi ville få svar på här var ”Hur mycket sålde vardera säljare per månad under hela 2006?”. Detta för att få fram vilken säljare som sålt bäst, för att kunna motivera de andra säljarna. Eftersom vi endast har två säljare fungerade frågan bra och behöver därmed inte definiera en specifik säljare.

Column	Alias	Table	Output	Sort Type	Sort Order	Group By	Filter
FtgID		Foretag (d...	<input checked="" type="checkbox"/>			Group By	
TotalForsaljning	Totalfo...	FaktaTabe...	<input checked="" type="checkbox"/>	Descending	1	Sum	
År		Tid (db_ow...	<input checked="" type="checkbox"/>			Group By	
År		Tid (db_ow...	<input type="checkbox"/>			Where	= 2007

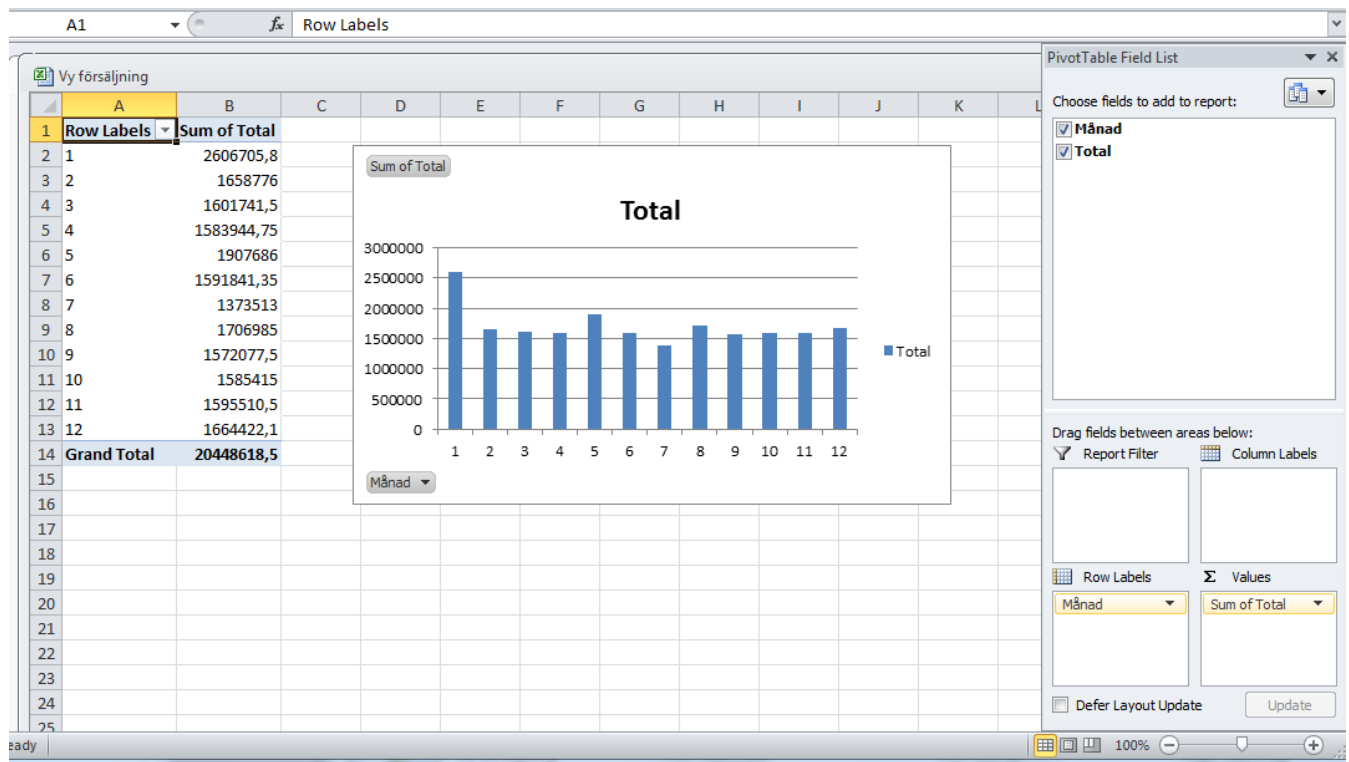
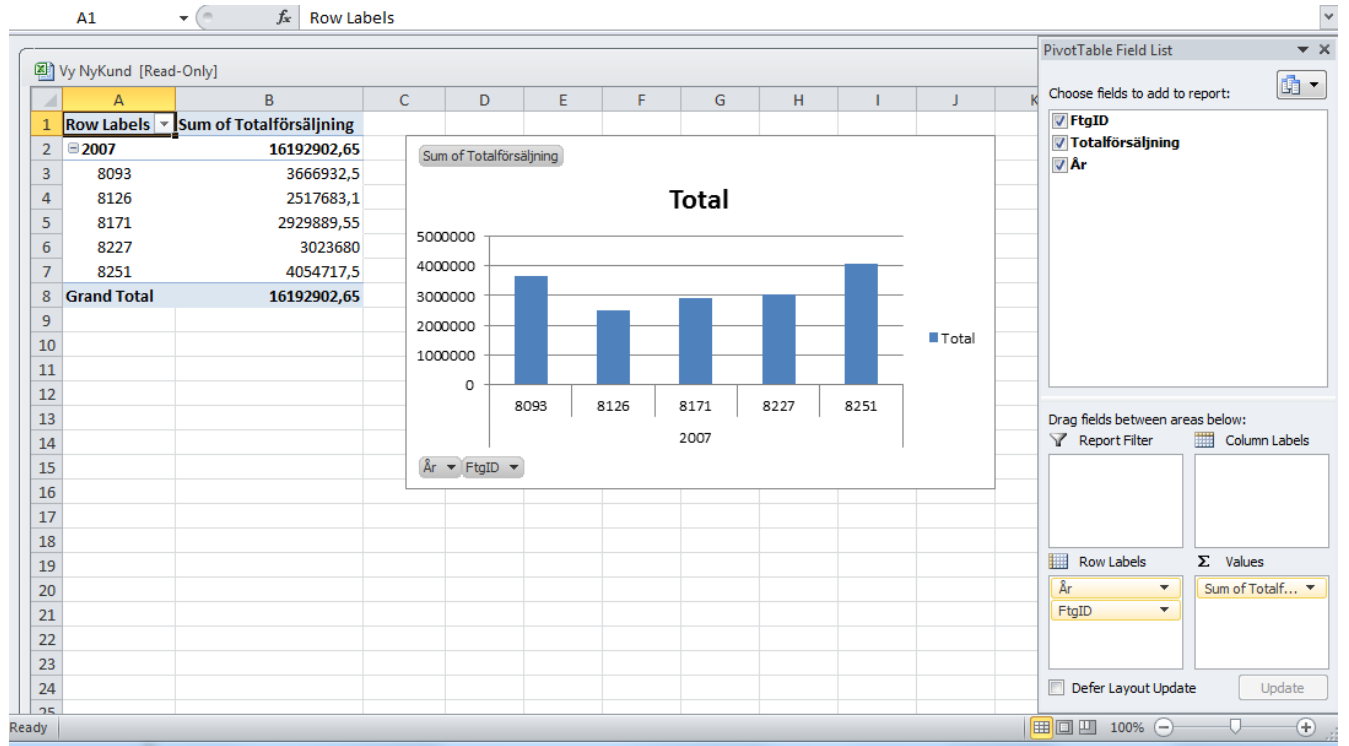
```

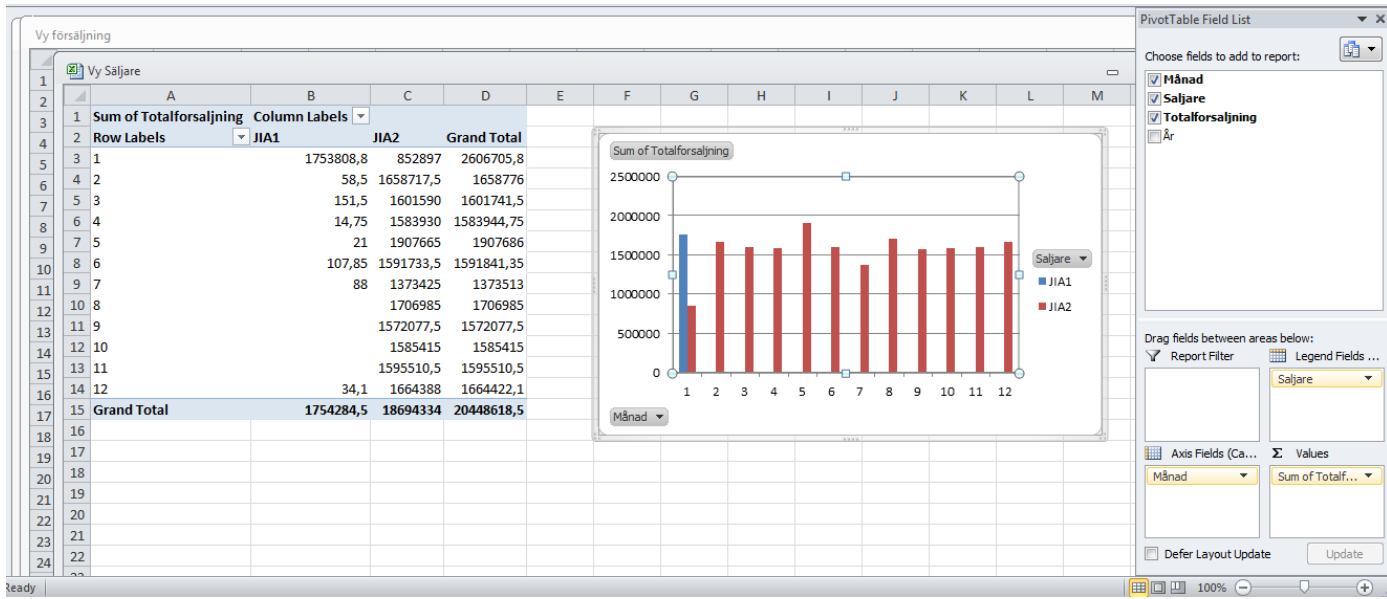
SELECT TOP (5) db_owner.Foretag.FtgID, SUM(db_owner.FaktaTabell.TotalForsaljning) AS Totalforsaljning, db_owner.Tid.År
FROM
  db_owner.Foretag INNER JOIN
    db_owner.FaktaTabell ON db_owner.Foretag.FtgID = db_owner.FaktaTabell.FtgID INNER JOIN
      db_owner.Tid ON db_owner.FaktaTabell.TidsID = db_owner.Tid.TidsID
WHERE (db_owner.Tid.År = 2007)
GROUP BY db_owner.Foretag.FtgID, db_owner.Tid.År
ORDER BY Totalforsaljning DESC
  
```

FtgID	Totalforsaljning	År
8251	4054717,5000	2007
8093	3666932,5000	2007
8227	3023680,0000	2007
8171	2929889,5500	2007

Genom att ställa frågan ”Vilka topp 5 kunder hade vi 2007?” ville vi få ut vilka kunder som var mest lönsamma under året. För att få ut detta ser vi vilka kunder vi ska lägga mest fokus på och skapa långvariga relationer med.

## 2.8 Rapporter





### **3. Reflektion**

Det har krävts mycket engagemang och vilja genom hela detta arbete eftersom våra förkunskaper inte varit tillräckliga. En del steg har varit tidskrävande och svåra att ta sig igenom. Vi tyckte att de första två labbarna gick lätt och smidigt men att svårighetsgraden ökade succesivt. Vi hade inställningen från början att det skulle bli relativt lätt att ta sig igenom labbarna och göra bra ifrån oss, men efter ett tag insåg vi att det skulle kräva mycket tid.

Tidigare i arbetet har vi beskrivit våra problem som vi stött på under arbetets gång. Vi tycker vi löst dem på ett bra och hanterbart sätt. Vi har hela tiden suttit tillsammans och arbetat vilket bidragit till ett bra lärande, alla fått suttit vid datorn och fått bra förståelse.

Ett av problemen vi stött på är att vi varit tvungna att ändra våra frågor, på grund av att det inte fanns några värden på vissa attribut. Frågan vi tog bort var: "Vilket land säljer vi mest till?". Det var landskod och lagställe som saknade värden. Vi lade till frågan: " Hur mycket sålde vardera säljare per månad under hela 2006?".

Nu i efterhand tycker vi att vi lärt oss oerhört mycket under projektets gång och att vi förstår varenda steg. Mycket beror nog på att vi fick göra om stagingtabellen och därmed repetera de olika stegen flera gånger.