Data Intensive Systems - Miniproject - Part 1

Erik Sidelmann Jensen
ejens11@student.aau.dk

Lasse Vang Gravesen
lgrave11@student.aau.dk

Dennis Jakobsen
djakob11@student.aau.dk

1 DIS Miniproject

1.1 Task B

- Sales
- Payments
- Members
- Location

We pick the 'Sales' business process, because it is the most interesting. For that we need to model sales, products, members along with the time and date.

As for granularity, when it comes to sales the time goes down to seconds. We decided to cut it off at minutes, because seconds are not that important. When it comes to members, the granularity goes down to years and there is no more information so that is the only thing that is retained.

For product it is somewhat possible to add extra levels to the hierarchy, such as category of product like dairy or soda.

The data is reasonable for paying customers and detailed enough to ask valuable questions.

Examples of questions:

- How much is bought at some point during some day?
- How does the amount sold change over time?
- Which days are the most busy?
- When is it best to restock, given low activity?
- How much revenue is gained each day, week, month, year?
- Which products have changed the most in price from year to year?
- Which department or member have spent the most?

1.2 Task C

Slowly changing dimensions are not that important.

The business process proposed to be modelled in the data warehouse is the 'Sales' business process. With regards to dimensions we pick out product, time, date, and member because they allow for the most interesting queries. Our granularity with regards to time goes down to minutes across two different dimensions to reduce the amount of rows, with regard to product it only has a name attribute, with regard to member it has balance and year. It might be a good idea to split products into categories, but the data does not directly allow for that and it would have to be done manually based on the name. It might also be a good idea to show if its a special day, like if its a holiday or if a day falls in a vacation.

The schema for the dimensions can be seen below.

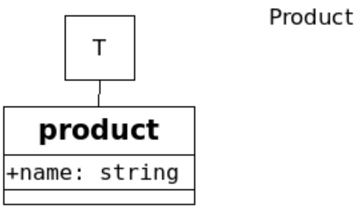


Figure 1.1: The product dimension.

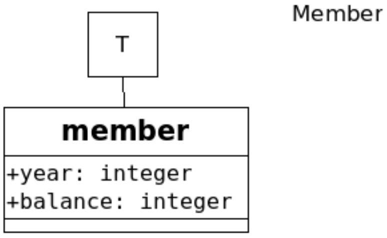


Figure 1.2: The member dimension.

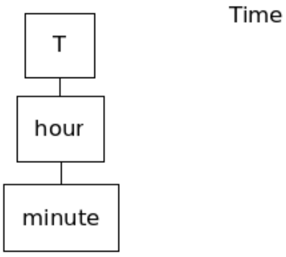


Figure 1.3: The time dimension.

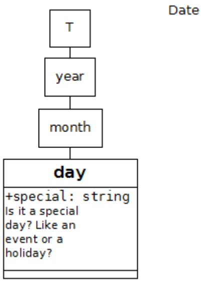


Figure 1.4: The date dimension.

The star schema for it can be seen below. With the schema it is important to note that the dimensions have a surrogate key with a serial integer(it auto increments).

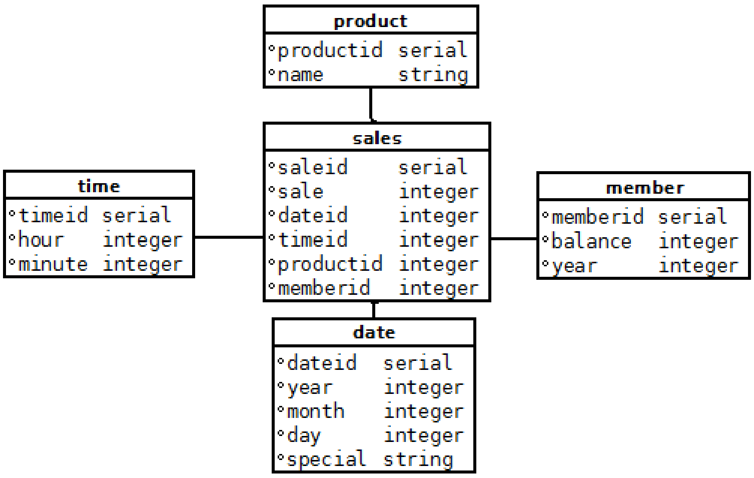


Figure 1.5: The star schema for the data warehouse.

Here is the SQL Table creation text, with some things like serial sequences stripped:

```
CREATE TABLE date (
    dateid integer NOT NULL,
    year integer,
    month integer,
    day integer,
    special text
);

CREATE TABLE member (
    balance integer,
    year integer,
    memberid integer NOT NULL
);

CREATE TABLE product (
    productid integer NOT NULL,
    name text
);

CREATE TABLE sales (
    memberid integer,
    dateid integer,
    timeid integer,
    productid integer,
    sale integer,
    saleid integer NOT NULL
);

CREATE TABLE ”time” (
    timeid integer NOT NULL,
    hour integer,
    minute integer
);

ALTER TABLE ONLY date
    ADD CONSTRAINT date_pkey PRIMARY KEY (dateid);

ALTER TABLE ONLY member
    ADD CONSTRAINT member_pkey PRIMARY KEY (memberid);

ALTER TABLE ONLY product
    ADD CONSTRAINT productid PRIMARY KEY (productid);

ALTER TABLE ONLY sales
    ADD CONSTRAINT sales_pkey PRIMARY KEY (saleid);

ALTER TABLE ONLY ”time”
    ADD CONSTRAINT timeid PRIMARY KEY (timeid);

ALTER TABLE ONLY sales
```

```
    ADD CONSTRAINT dateid FOREIGN KEY (dateid) REFERENCES date(dateid);

ALTER TABLE ONLY sales
    ADD CONSTRAINT memberid FOREIGN KEY (memberid) REFERENCES member(memberid);

ALTER TABLE ONLY sales
    ADD CONSTRAINT producid FOREIGN KEY (productid) REFERENCES product(productid);

ALTER TABLE ONLY sales
    ADD CONSTRAINT timeid FOREIGN KEY (timeid) REFERENCES "time"(timeid);
```

Data Intensive Systems - Miniproject - Part 2

Erik Sidelmann Jensen
ejens11@student.aau.dk

Lasse Vang Gravesen
lgrave11@student.aau.dk

Dennis Jakobsen
djakob11@student.aau.dk

1 DIS Miniproject

1.1 Task D

We choose to load the data from the provided CSV files, according to the following high-level ETL Flow:

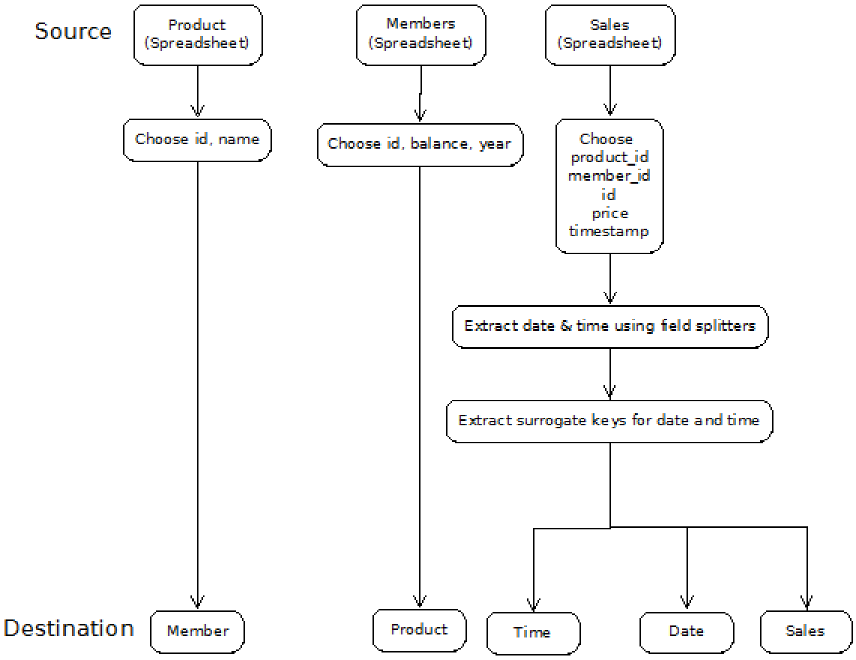


Figure 1.1: High Level ETL Flow

This is the flow it follows in Pentaho (aka Spoon):

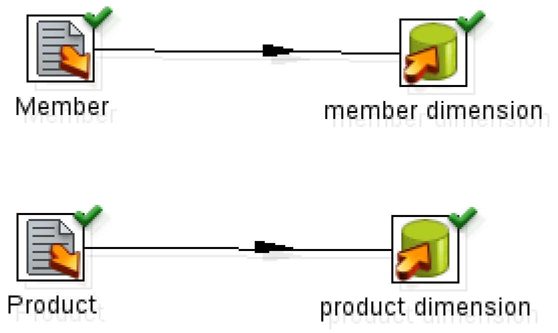


Figure 1.2: Pentaho ETL Flow for Member and Product

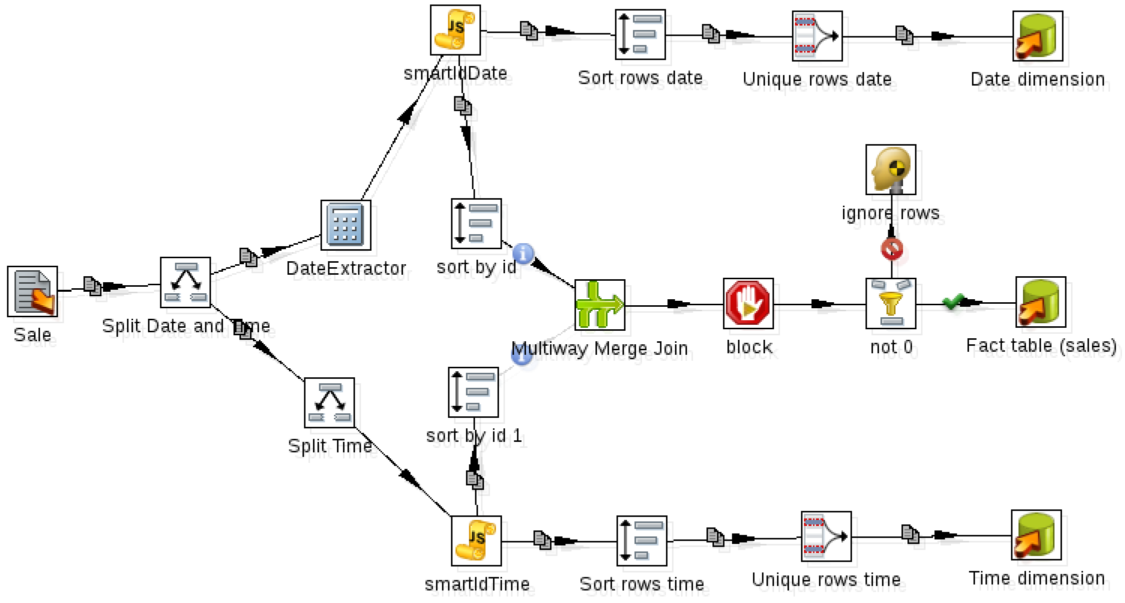


Figure 1.3: Pentaho ETL Flow for Sales

The ETL Flow for ‘sales’, ‘date’ and ‘time’ is most interesting, so we go into more detail there. The first step we take is split the timestamp into date and time, by splitting on space. For the ‘date dimension’ we extract the date from that, and create an easily determinable id such that we do not have to do any lookups. Then we sort the date result on the smart id, and make it unique and add it to the ‘date’ dimension. Then for the ‘time’ dimension we also create an easily determinable id, sort it on the smart id and make it unique and put it into the ‘time’ dimension.

For the sales fact table we reuse the smart ids made for the ‘date’ dimension and ‘time’ dimension, sort on the key ’id’ and multiway merge join. Then we block execution of this until the rest of the dimensions have been filled because it needs to refer to foreign keys that not necessarily have been inserted yet. Then we filter out some rows from the ‘sales’ fact table because it has foreign keys that do not match the ‘member’ dimension, send the bad ones to a dummy block and the good ones to the ‘sales’ fact table.

1.2 Cut Corners

We TRUNCATE CASCADE all tables before running the ETL flow and load all the data fully.

We cut a corner by filtering out rows in the ‘sales’ fact table if it contained a member_id that equalled 0 by using a simple equation.

It is important to note here that we do not use surrogate keys for the ‘member’, ‘product’ and ‘sales’, as such the result does not support versioning. We did this because we felt that it was not all that important to support versioning for this data warehouse because it’s a prototype and because we did not figure out until pretty late that for real data warehouses using those surrogate keys is very important and at that point it would have required large changes to what we had already made.

Data Intensive Systems - Miniproject - Part 3

Erik Sidelmann Jensen
ejens11@student.aau.dk

Lasse Vang Gravesen
lgrave11@student.aau.dk

Dennis Jakobsen
djakob11@student.aau.dk

1 DIS Miniproject

1.1 Task E & F

We defined our multidimensional cube using the following xml:

```xml
<?xml version="1.0"?>

<Schema name="FKlubDW">
    <Cube name="Sales">
        <Table name="sales"/>
        <Dimension name="Product" foreignKey="productid">
            <Hierarchy hasAll="true" allMemberName="All Products" primaryKey="productid">
                <Table name="product"/>
                <Level name="Name" column="name" uniqueMembers="false"/>
            </Hierarchy>
    </Dimension>
  <Dimension name="time" type="TimeDimension" foreignKey="timeid">
    <Hierarchy hasAll="true" allMemberName="All Times" primaryKey="timeid">
      <Table name="time"/>
      <Level name="hour" column="hour" uniqueMembers="true" levelType="TimeHours" type="Numeric"/>
      <Level name="minute" column="minute" uniqueMembers="false" levelType="TimeMinutes" type="Numeric"/>
    </Hierarchy>
  </Dimension>
  <Dimension name="hour" type="TimeDimension" foreignKey="timeid">
    <Hierarchy hasAll="true" allMemberName="All Hours" primaryKey="timeid">
      <Table name="time"/>
      <Level name="hour" column="hour" uniqueMembers="true" levelType="TimeHours" type="Numeric"/>
    </Hierarchy>
  </Dimension>
   <Dimension name="date" type="TimeDimension" foreignKey="dateid">
    <Hierarchy hasAll="true" allMemberName="All Dates" primaryKey="dateid">
      <Table name="date"/>
      <Level name="Year" column="year" uniqueMembers="true" levelType="TimeYears" type="Numeric"/>
      <Level name="Month" column="month" uniqueMembers="false" levelType="TimeMonths" type="Numeric"/>
      <Level name="Day" column="day" uniqueMembers="false" levelType="TimeDays" type="Numeric"/>
    </Hierarchy>
  </Dimension>
        <Dimension name="Member" foreignKey="memberid">
            <Hierarchy hasAll="true" allMemberName="All Members" primaryKey="memberid">
                <Table name="member"/>
    <Level name="memberid" column="memberid" uniqueMembers="true" type="Numeric" />
                <Level name="Balance" column="balance" uniqueMembers="false" type="Numeric" />
                <Level name="Year" column="year" uniqueMembers="false" type="Numeric" />
            </Hierarchy>
    </Dimension>
        <Measure name="Sale Amount" column="sale" aggregator="sum" formatString="#,###"/>
    </Cube>
</Schema>
```

We then could answer the following questions:

How much is bought at some point during some day?

We can answer this by looking at summarized hours, we look at the hours with the most spent.

28,618,675 Ører has been spent at the 12th hour and 20,400,875 Ører has been spent at the 13th hour.

How does the amount sold change over time? We can answer this by looking at individual years and how much is total spent on each of the years.

Year	Sale Amount
1996	187,650
1997	2,359,375
1998	13,696,025
1999	15,180,275
2000	15,763,800
2001	16,402,075
2002	19,679,050
2003	18,017,550
2004	19,876,325
2005	17,882,675
2006	20,225,450
2007	19,932,050
2008	61,650

As it can be seen the first years had little sales compared to the later years such as 2002 through 2007. The last year presented 2008 may be an invalid year since the total sales may be misleading if sales for a whole year is not considered.

When is it best to restock, given low activity?

We can answer this by looking at which hours have the least activity during 8-16.

Those hours are 8 with 9,632,450 and 16 with 9,981,050.

A better way of answering this would be to look at the weekdays too, but we do not have that field.

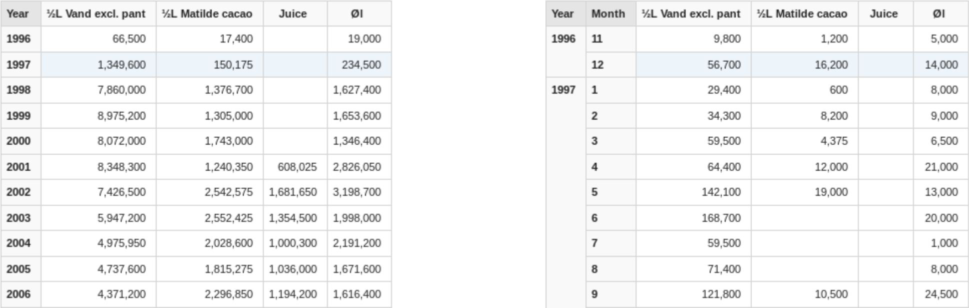
Which product gives the most revenue?

Name	Sale Amount
$\frac{1}{2}$L Vand excl. pant	66,970,550
Øl	20,272,250
$\frac{1}{2}$L Matilde cacao	19,476,750
Kaffe/Choko(1 kop)	11,578,400
Juice	8,738,675

What does the increase in sale for specific products look like over time? See Figure 1.1 for a graphical overview of how the sales for soda without deposit(pant), Matilde kakao, juice, and beer changes over each year. For example it can be seen that beer was a popular drink in 2002. The figure is split in a non-drilled version in Figure 1.1a and a drilled in Figure 1.1b

Is it worth it to sell soda without deposit(pant) See Figure 1.2 for an illustration of how the sales are of soda with and without deposit(pant). It can for example be seen that the sale for soda with deposit(pant) is only a fraction of the sales of soda without. The figure is split in a non-sliced and sliced version in Figure 1.2a and Figure 1.2b

Which member have spent the most? The member that has spent the most is '1704' having spent 1,439,650 ører, see Figure 1.3.



Year	½L Vand excl. pant	½L Matilde cacao	Juice	Øl
1996	66,500	17,400		19,000
1997	1,349,600	150,175		234,500
1998	7,860,000	1,376,700		1,627,400
1999	8,975,200	1,305,000		1,653,600
2000	8,072,000	1,743,000		1,346,400
2001	8,348,300	1,240,350	608,025	2,826,050
2002	7,426,500	2,542,575	1,681,650	3,198,700
2003	5,947,200	2,552,425	1,354,500	1,998,000
2004	4,975,950	2,028,600	1,000,300	2,191,200
2005	4,737,600	1,815,275	1,036,000	1,671,600
2006	4,371,200	2,296,850	1,194,200	1,616,400

(a) Normal version

Year	Month	½L Vand excl. pant	½L Matilde cacao	Juice	Øl
1996	11	9,800	1,200		5,000
	12	56,700	16,200		14,000
1997	1	29,400	600		8,000
	2	34,300	8,200		9,000
	3	59,500	4,375		6,500
	4	64,400	12,000		21,000
	5	142,100	19,000		13,000
	6	168,700			20,000
	7	59,500			1,000
	8	71,400			8,000
	9	121,800	10,500		24,500

(b) Drilled down version

Figure 1.1: Different product sales over time.

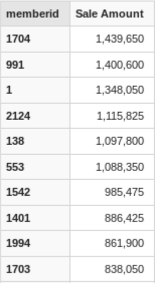
Name	hour	Sale Amount
½L Vand excl. pant	10	6,725,550
	11	8,599,100
	12	12,855,550
	13	8,299,650
	14	6,992,900
½L Vand incl. pant	10	267,200
	11	300,500
	12	450,900
	13	314,800
	14	243,700

(a) Normal version

Name	hour	Sale Amount
½L Vand excl. pant	12	12,855,550
½L Vand incl. pant	12	450,900

(b) Slice and diced

Figure 1.2: Sales of soda with and without deposit(pant)



memberid	Sale Amount
1704	1,439,650
991	1,400,600
1	1,348,050
2124	1,115,825
138	1,097,800
553	1,088,350
1542	985,475
1401	886,425
1994	861,900
1703	838,050

Figure 1.3: Member ids and sales