Exercise 1 Data Warehouse Architecture

Task 1.1: Case Study: Mountain States Health Alliance

In order to make its health care system more efficient, 29 counties in Tennessee, Virginia, Kentucky and North Carolina decided to establish the Mountain States Health Alliance (MSHA), which includes among others the following medical institutions of the region:

- The Franklin Woods Community Hospital (FWCH) in Johnson City: An 80-bed hospital offering a full array of primary care and some specialty services.
- The James H. and Cecile C. Quillen Rehabilitation Hospital (QRH) in Johnson City: A 47-bed comprehensive inpatient rehab hospital.
- The Norton Community Hospital (NCH), located in Norton, Virginia: A 129-bed, acute-care facility that has been serving Southwest Virginia and Southeastern Kentucky since 1949.
- SYNERGY Laboratories (SYN) is a full-service laboratory (e.g. for blood count), comprised of the laboratories within Mountain States Health Alliance.
- The Mountain States Imaging Center (MSIC) provides a wide array of high-quality diagnostic imaging services such as MRI, CT scan, and X-Ray.

MSHA also has a close cooperation with the Emory University School of Medicine (EUSM) in Atlanta. The university runs various projects on Rehabilitation Medicine and wants to use data from MSHA for its studies.

Making all relevant information available to doctors and nurses in all hospitals as well as to the management of all institutions in MSHA is important to achieve an effective and efficient health care system. In particular, the management wants to establish an early warning system for the following events: (1) A certain type of drugs is running out of stock in the hospitals of MSHA, (2) a critical level of MRSA (Methicillin-resistant Staphylococcus aureus) contamination is reached.

Here is a short excerpt of the description for various data collections locally available in each of the institutions:

Institution	**Data Set**	**Notes**
FWCH	department and beds	Information on beds per department and when they were occupied
	patients	Patients with names, health status, insurance information, …
	employees	Staff of the hospital
	doctors	Doctors
	stock	Used drugs and their stock level
	procedures	Treatments of patients with date, treatment, doctor, etc.
QRH	patients	Patients with names, health history, insurance information, …
	departments	Subject of department, available equipment
	bed	Beds and when they were occupied

Institution	Data Set	Notes
	treatments	Treatments of patients with date, treatment, staff, etc.
	payments	Billing information, open accounts, …
	staff	Nurses and other staff of the hospital
	doctors	Doctors and their specialization
	drugs	Stock level of drugs
MSIC	treatments	Protocol and results of MRI, CT scan, and X-Ray, etc.
	schedule	Planned treatments
SYN	treatments	Blood counts etc. with time, patient, result, etc.
Pharmacy	Inventory	Drugs and their stock level
	Orders	Orders and deliveries by/to hospitals and other institutions

In a first step, the MSHA decided to leave the IT systems of each of these institutions as they are. Nevertheless, an integrated view on all the information related to patients, diagnoses and treatments is needed. Thus, a data warehouse project is initiated. Discuss the following questions related to the motivation for and the architecture of a data warehouse system providing this integrated view.

Task 1.2: Information Needs

Try to find examples for questions that might be relevant for decisions of (a) a doctor at the FWCH, (b) a doctor at the MSIC, (c) the manager of the QRH, (d) the general manager of MSHA, and (e) the pharmacy supplying drugs to all members of MSHA. Which data of which of the institutions mentioned above is needed to answer these questions?

Task 1.3: Issues of Information Integration

Think of potential problems that could arise when integrating data of all members of MSHA. Think of various dimensions of data sources and possible issues with data quality.

Task 1.4: Data Warehouse Architecture

Sketch the architecture of a Data Warehouse for MSHA. Focus on data flow and internal users (nurses, doctors, managers, …) first. (focus on data flow). and Operational Data Stores

Task 1.5: Data Marts

What are the advantages and disadvantages of using data marts? Compare them to those of a central data warehouse architecture. How do dependent data marts differ from independent data marts? What are appropriate usage scenarios for each of these types of data marts? Discuss possible applications of Data Marts in the MSHA data warehouse.

Task 1.6: Operational Data Store

Discuss possible applications of Operation Data Stores (ODS) in the MSHA data warehouse.

Exercise 2 Data Integration and Monitoring

Task 2.1: Intro ODPS

In later exercises, you will create and execute SQL queries. We use the ODPS system for this purpose. ODPS basically is a web interface to an IBM DB2 database. The necessary password for registration in ODPS (course DWDM2015) is announced during the exercise lesson. The document ODPS_Intro_v04.pdf gives a short introduction to ODPS, which is available here: https://odps.informatik.uni-stuttgart.de

ODPS also includes a tutorial called "Simple Queries in SQL" covering the main types of SQL queries you should be familiar with for this and the following exercises. If you are not sure about your SQL skills, it is recommended that you go through this tutorial as soon as possible.

Task 2.2: Sample Scenario trendy4all

This task introduces the available data set for a sample scenario used in several other exercises. The clothing company trendy4all operates branch offices that are each assigned to one of the regions "South" and "North East". For each region, data is managed in a separate database covering each store and the sales that take place there. In addition, in the headquarter and also in each region the complete data on all items offered by trendy4all is available.

The following figure outlines the schema of the available databases. Their content is briefly described below.

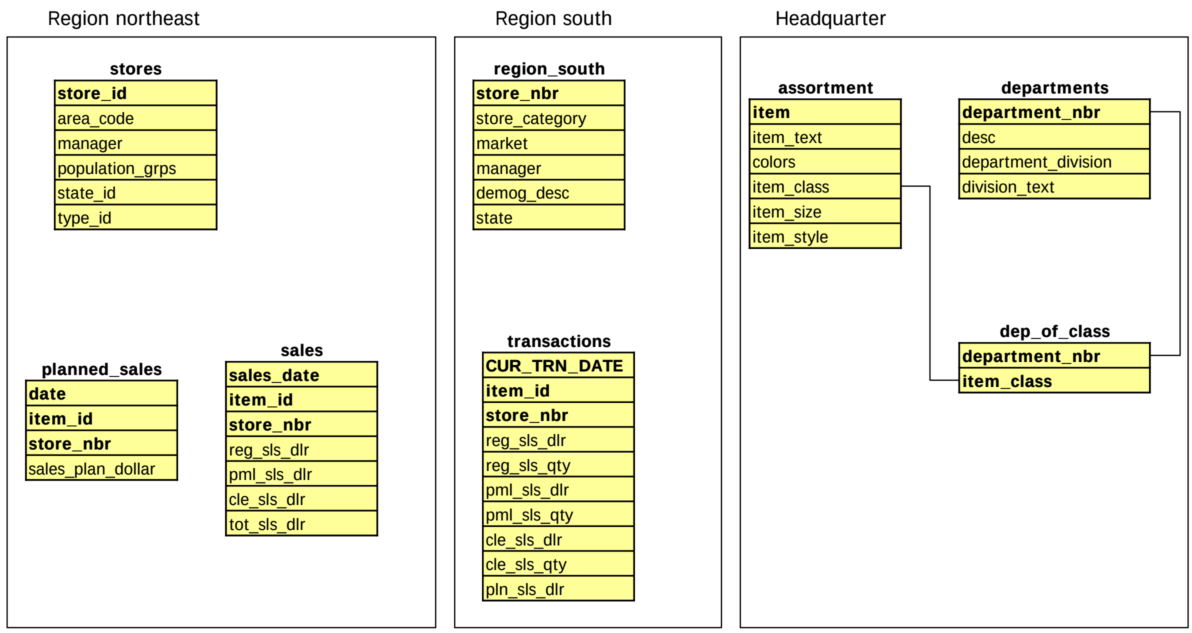


Table	Column	Description
stores	store_id	For each store that is a unique number. A textual desription of the stores is not available in the database.
	area_code	It indicates, which region the store belongs to, e.g. 1 for "Mid-Atlantic", 2 for "New England"…
	manager	Name of the responsible manager
	population_grps	Population groups that typically go shopping in this store. An entry consists of a list of numbers separated by a comma. They indicate a group, e.g. 3 for "Latino", 7 for "Hispanic"…
	state_id	Number of the state where the store is located
	type_id	Type of store, e.g. 1 for "Super Malls", 3 for „Retail Only".
planned_sales	date	date
	item_id	Product number
	store_nbr	Store number
	sales_plan_dollar	Planned turnover for the indicated day, product and store. Turnover is in dollars.
sales	sales_date	date
	item_id	Product number
	store_nbr	Store number
	reg_sls_dlr	Turnover in dollars of regular goods
	pml_sls_dlr	Turnover in dollars of regular goods that are advertised
	cle_sls_dlr	Turnover in dollars of goods in closing sale
	tot_sls_dlr	Total turnover

region_south	store_nbr	Unique number of a store within region „south"
	store_category	Category of a store, e.g. „Wholesale Only", „Retail Only" or „Super Malls".
	market	Part of the region where the store is located, e.g. „Deep South"
	manager	Name of responsible manager
	demo_desc	Population groups that typically go shopping in this store. An entry consists of a list of numbers separated by a comma. They indicate a group, e.g. 3 for "Latino", 7 for "Hispanic"…
	state	Two letters for the state, where the store is located
transactions	cur_trn_date	Date of transaction
	item_id	Product number
	store_nbr	Store number
	reg_sls_dlr	Turnover and quantity of regular goods
	reg_sls_qty	
	pml_sls_dlr	Turnover and quantity of regular goods that are advertised
	pml_sls_qty	
	cle_sls_dlr	Turnover and quantity of goods in closing sale
	cle_sls_qty	
	pln_sls_dlr	Planned total turnover

assortment	item	Product number
	item_text	Product description
	colors	List of colors available for the product
	item_class	Assigned product group
	item_size	Indication of size of the product (from 1= small to 5=X-Large)
	item_style	Describes textually the season in which the product is used (Winter, Spring, Summer).
departments	department_nbr	Department number
	desc	Description of the department
	department_division	Number of the department devision
	division_text	Description of the department devision, e.g. „Casual Clothing"

The eight tables are available in ODPS as exercise data in a schema called dw ("Data Warehouse"), i.e., with the query "Select * from assortment" you receive 17 rows of the table assortment in the schema of this exercise data (dw). You are only able to read the data, but cannot change it.

If you want to create a trigger or add/delete/change data entries, you should copy the tables into your own schema. You have to execute following steps to get your own copy of the data:

1. Create the tables in your own schema. You find the necessary SQL statements in the file *create_warehouse_tables.sql*.
2. You may also copy the data into your newly created tables. You find the necessary SQL-Statements in the file *fill_ warehouse_tables.sql*.

Task 2.3: Accessing and exploring trendy4all data

In order to get familiar with the data of trendy4all try to formulate SQL queries that provide appropriate information regarding the following aspects:

a) Explore how stores are identified in each region.
b) Compare how states and various kinds of stores are represented in the databases.
c) Which regions are relevant for trendy4all?
d) Analyze available population groups.
e) Compare the sales information.

Explore the data with respect to the mentioned aspects. Think about the possible issues that may arise when you are asked to integrate the data of trendy4all in one database.

Task 2.4: Data Integration and Aggregation by SQL

Given the database schema of trendy4all as described in Task 2.2. We want to derive the following aggregated information for trendy4all:

1. Provide an overview of stores from all regions.
2. Aggregate turnover and quantity (regular, promotional, clearance, total) on the department and store level. We also need this aggregation per year and month.

Discuss the difficulties that could arise when we try to provide this aggregated information. Provide the requested data using a single SQL query (if possible).

Task 2.5: Monitoring

Given the database schema of data sources of trendy4all as described in Task 2.2. To find the data the ETL process should move into the data warehouse, the monitoring component has to identify data changes in the source systems.

1. Discuss under the following assumptions how monitoring could be achieved for trendy4all. Consider each set of assumptions separately.
    a. All source systems use a centralized RDBMS for data management.
    b. The source systems use a distributed RDBMS for data management.
    c. All source systems store the data in a set of files in the file system.
    d. Some of the source systems use files for data management, others use a centralized RDBMS.

2. We assume that the monitoring of these source systems is realized by triggers. Create the triggers that are necessary in the region "northeast" to log the changes of the source data in the following tables:
   - Table new_planned_sales:
     This table stores all new entries for the table planned_sales. An additional column should store when data was added to table new_planned_sales.
   - Table new_sales:
     This table stores all new entries of the table sales. Here, the time of the entry should also be stored.
   - Table upd_stores:
     This table stores all changes in table stores. Both the type of change (insert/update/delete) and the time of change should be stored in additional columns.

Exercise 3 Conceptual Data Warehouse Design

Task 3.1: Case Study Maintenance Work on Airplanes

In this exercise, we cover the conceptual design of a data warehouse using the approaches introduced in the lecture (Dimensional Fact Model, UML Profile).

The data warehouse should cover information related to the maintenance work on airplanes. Here is some information on the available data:

- Maintenance is usually summarized in so-called check types (A, B, C, D). They are composed of many work steps. However, there are also independent maintenance operations that cannot be assigned to a check type.
- For each work step, the mechanics that executed it are registered. In each work step, several mechanics from different teams may participate. They register timestamps for the start and the end of the work step. In the end, every work step is marked with a state, which describes, e.g., whether the operation had to be cancelled before completion. Mechanics are in particular characterized by their name and their education.
- Maintenance work on air planes of different years of construction, types and configurations is done in a hangar next to the airport. In such a hangar, there are typically a number of maintenance halls. Owner of these hangars are airlines, but possibly also independent companies. Sometimes the hangars are also operated by multiple owners. Airplanes of a certain airline may also be checked in hangars of other airlines.
- The data warehouse should support the analysis of maintenance work for days, weeks, months, quarters, and years. Additionally, a business year (October - September) should be considered.

Based on this data warehouse, users should be able to derive the following information:

- What is the average time for check type A in a maintenance hangar?
- How many mechanics have participated in check type D for airbus A320 of a specific airline? Provide the plane's configuration as additional descriptive information.
- In which months is most maintenance work executed? Report this for a specific maintenance hangar during the last year.
- How often had work step 078/015 of the check type D to be interrupted?
- Mechanic M was involved in the maintenance of how many planes last month? Provide also the education level of the mechanic.

Task 3.2: Facts and Dimensions

Which facts and which dimensions have to be covered? Discuss the granularity of the facts with respect to the different dimensions.

Task 3.3: DFM

Create the conceptual design using the Dimensional Fact Model (graphical representation). Give examples of aggregation statements.

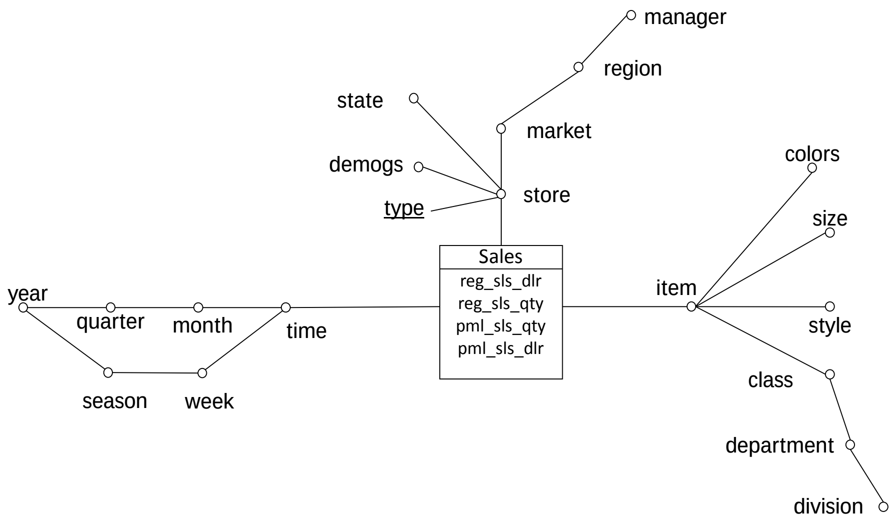
Task 3.4: UML Profile

Use the UML Profile for creating the conceptual schema. How do the DFM version and the UML version differ?

Exercise 4 Logical Data Warehouse Design

Task 4.1: Star Schema and Snowflake Schema

The data trendy4all has to manage was already described in a previous exercise. The conceptual schema for trendy4all is shown in the following figure.



1. Derive the logical schema from this conceptual schema using the star schema approach.
2. Derive the logical schema from this conceptual schema using the snowflake schema approach.
3. Name at least three aspects according to which the star schema and the snowflake schema differ. Compare both schema types with respect to these aspects.

Task 4.2: Extended Dimension Table Design

Based on the star schema from task 4.1, consider the following aspects of extended dimension table design.

1. The state of a store changes from time to time. We want to keep the complete history of state information over time. It should be possible to assign to each sales fact the store with the correct state information. There is no need to store when and for what reason the state information has changed. What is the appropriate design and usage of the store dimension table?
2. All the detailed information stored for each item changes regularly. We want to keep the complete history. In particular, we want to track when and for what reason, the item data changes. What is the appropriate design and usage of the store dimension table?
3. Comment on the following statements in one to three sentences:

a. There are three options to model slowly changing dimensions. All of them keep the complete history of changes.
b. The technique for modelling frequently changing large dimensions introduced in the lecture allows to keep all the details for each dimension attribute.
c. The technique for modelling frequently changing large dimensions introduced in the lecture results in frequent changes of the data volume of the dimension table.

Exercise 5 Data Cleansing and OLAP

Task 5.1: Data transformation and data cleansing

The Mountain States Health Alliance (MSHA) (see Exercise 2) identified the following issues for data integration and data cleansing:

a) Diagnosis information is available on a daily basis for inpatient and a weekly basis for outpatient care.
b) All hospitals use the same diagnosis code, but store them in different formats.
c) Hospitals use local patient Ids. Laboratories and the imaging center use patient Ids that also encode additional information related to patients (inpatient, outpatient, hospital).
d) In the MSIC data, patient Ids are sometimes missing for certain imaging services.
e) Only some hospitals deliver information on the expected duration of patients' stay at the hospital.

1. What are the data quality issues (if any) related to each of these aspects?
2. Discuss advantages and disadvantages of possible steps for data transformation and data cleansing. Which of them can automatically be applied?
3. What actions to improve data quality could be taken?

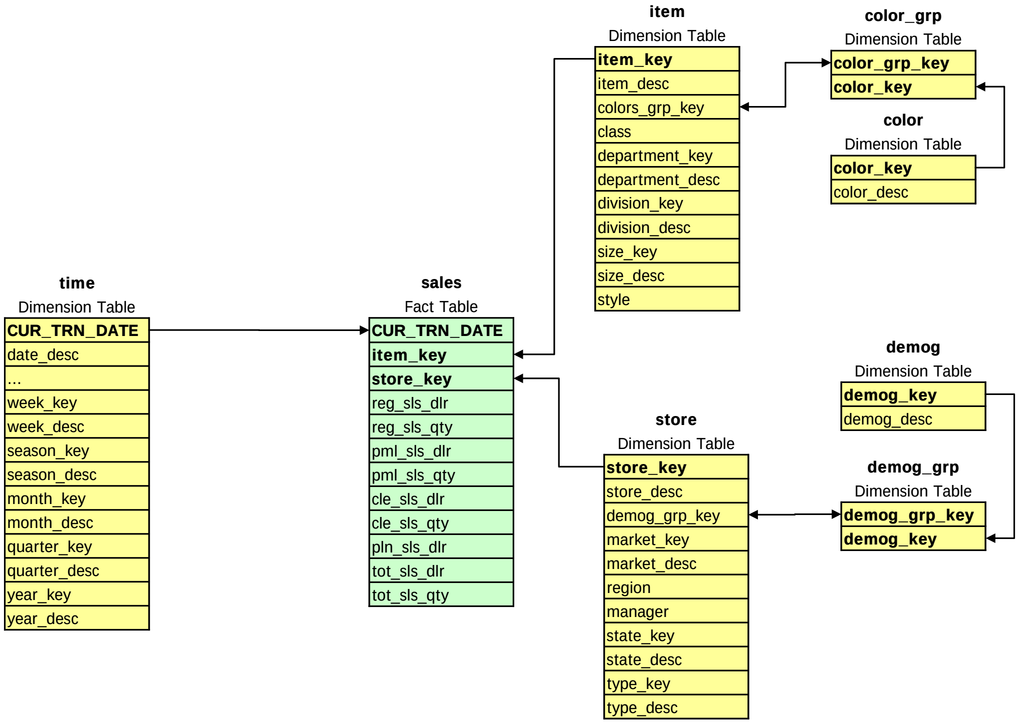
Task 5.2: OLAP

There are three options for the OLAP architecture: MOLAP, ROLAP and HOLAP. Which of the following statements are true? Argue why the others are wrong.

1. In MOLAP, ROLAP and HOLAP, the OLAP Servers generate SQL queries.
2. In MOLAP and HOLAP, the OLAP Server host multidimensional data cubes.
3. Drill through is an important operation and hence possible in ROLAP and MOLAP.
4. The same data warehouse schema might be used to build ROLAP or MOLAP functionality on top of it.
5. MDX is a query language for ROLAP and MOLAP.

Task 5.3: OLAP Operations

For this exercise we use the star schema given below. You may test the SQL queries from this exercise directly in a database. In ODPS you will find the tables sales, time, item and store in schema DW2.



We focus on the following star query:

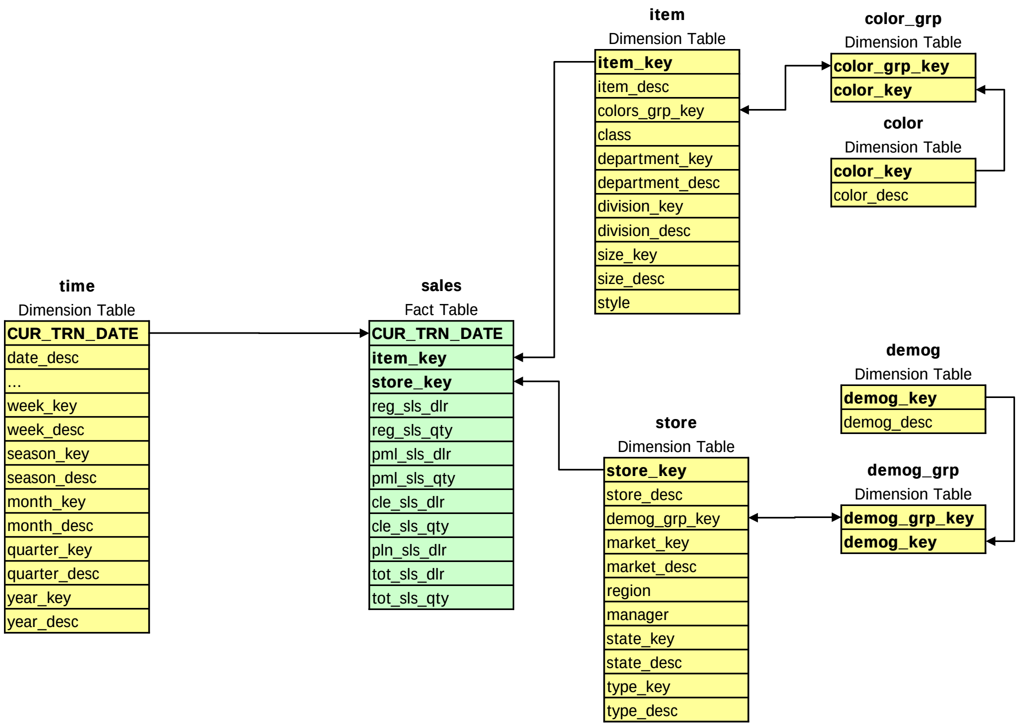
```
SELECT      month_desc, item_desc, state_desc,
            SUM(reg_sls_dlr), SUM(reg_sls_qty)
FROM        sales s, time t, item I, store o
WHERE       s.CUR_TRN_DATE = t.CUR_TRN_DATE
AND         s.item_key = i.item_key
AND         s.store_key = o.store_key
GROUP BY    month_desc, item_desc, state_desc
```

Modify the query to reflect the following OLAP operations:

1. A drill-down in the time dimension to the day level.
2. A roll-up in the item dimension to the division level.
3. Slicing in the time dimension (January 2015).
4. Dicing using all three dimensions (2015, division A, state CA).

Task 5.4: OLAP and SQL

For this exercise we use the star schema given below. You may test the SQL queries from this exercise directly in a database. In ODPS you will find the tables sales, time, item and store in schema DW2.



1. The following facts need to be aggregated: reg_sls_dlr, reg_sls_qty, tot_sls_dlr and pln_sls_dlr. Create an SQL query that covers this grouping options:
   - year, month, Department, Store
   - year, Department, Store
   - year, Department
   - year
   - total aggregate
2. First create a query that returns the total turnover (tot_sls_dlr) per year, week, department and store (one result row per combination). Then extend the query in a way that for each week the aggregated total turnover of the year until this week is additionally returned.

Task 5.5: Multidimensional storage vs. relational storage

Starting point is a multidimensional partitioning of an n-dimensional data cube with m partitions b_1 to b_m:

$$b_1 = \lfloor l_{1,1} : u_{1,1}; \ldots; l_{1,n} : u_{1,n} \rfloor \ldots b_m = \lfloor l_{m,1} : u_{m,1}; \ldots; l_{m,n} : u_{m,n} \rfloor$$

- Provide a generic solution for calculating the needed storage for partition b_i of a data cube using multidimensional or relational storage. Consider the filling degree δ.
- Describe until which filling degree the multidimensional storage is superior compared to the relational storage with respect to the needed storage. Calculate the results for 2, 3 and 4 dimensions. Assume that we need 16 bytes to store one fact value and 8 bytes to store the value of one dimension attribute.

Exercise 6 Data Mining

Task 6.1: Knowledge Discovery Process

Name the phases of the knowledge discovery process and explain the purpose of each of the steps in one or two sentences.

Task 6.2: Data Mining Phases

Mention and explain the three phases of data mining. Which of them are applicable to all four data mining techniques? Why are some phases not relevant for some data mining techniques?

Task 6.3: Apriori Property

Assume we are trying to identify frequent itemsets in transactions that include some of the items beer, cheese, tomatoes and milk. We already know that

- the following itemsets are frequent: {beer}, {cheese}, {tomatoes}, {beer, milk}
- the following itemset is not frequent: {cheese, tomatoes}

Which of the following facts can we directly derive from this knowledge using the apriori property?

1. Itemsets {beer, cheese, tomatoes}, {cheese, milk, tomatoes}, {beer, cheese, milk, tmoatoes} are not frequent
2. Itemset {milk} is frequent
3. Itemset {beer, cheese} is frequent

Task 6.4: Frequent itemset discovery

The following table contains a set of transactions:

TID	Itemset
100	A, B, E
200	B, D
300	B, C
400	A, B, D
500	A, C
600	B, C
700	A, C
800	A, B, C, E
900	A, B, C
1000	A, B, C, F

Use the Apriori algorithm to calculate the frequent itemsets (minimum support = 0.2) on basis of these transactions. Describe each step in detail and provide the intermediate result for each iteration. What is the stop criteria of the algorithm?

Task 6.5: Association rules

The following algorithm for creating association rules is given:

```
for each frequent k-itemset f∈F, k≥2 do
```

```
    H1={  R | L->R is association rule with |R|=1 and f=L∪R}
    call ap-genrules(f, H1);
end

ap-genrules(Itemset fk, Itemsets rightm) begin
    if (k>m+1) then begin
        rightm+1=generateCandidates(m, rightm)
        for each rm+1∈rightm+1 do begin
            confidence = fk.count/(fk-rm+1).count;
            if confidence ≥ minConfidence then
                rules.add(fk-rm+1, rm+1);
            else
                delete rm+1 from rightm+1;
            end;
        end;
        call ap-genrules(fk, rightm+1);
    end;
end;
```

Remarks:

- The following property is used: If the rule B→A\B should be valid (i.e. confidence>minConfidence), then for each B⊂C also C→A\C has to be valid.
- The algorithm contains the set F of all frequent itemsets as input. For these itemsets the number of items (size) and the support value (count) is available.
- The algorithm uses the function `generateCandidates` of the Apriori algorithm.

Exercises:

1. Use this algorithm on the example of the lecture. (minimum confidence = 0,5)
2. Compare the behavior of the algorithm with the simple algorithm from the lecture. The frequent itemsets in the adjacent table are the starting point. Consider only rules that can be created from itemset {A, B, C, D, E} (minimum confidence = 0.4).
3. Also indicate the lift factor for the created rules. Assume that you have 50 transactions.

Element of F	**count**[1]
{A}, {B}, {C}, {D}, {E}	9
{A, B}, {A, C}, …, {D, E}	8
{A, B, C}, {A, B, D}, {A, B, E}, …, {B, D, E}, {C, D ,E}	7
{B, C, D, E}	6
{A, B, D, E}	6
{A, C, D ,E}	5
{A, B, C, E}	5
{A, B, C, D}	6
{A, B, C, D, E}	2

[1]: The values are ordered schematically for the simplification of the calculation!

Task 6.6: Clustering

The following information on the different stores of a department store chain is available. The attributes area, staff members and turnover have already been normalized to get the same range of values for all dimensions.

Store ID	**area**	**staff**	**turnover**	**Store type**
701	17	18	17	A

Store ID	area	staff	turnover	Store type
702	25	14	1,12	B
703	9	23	7	A
704	20	30	34	A
705	10	11	3,5	B
706	15	15	9	A

1. Divide the stores into three clusters by means of the K-Means algorithm (Euclidian distance) considering the attributes area, staff members and turnover. Initially, the stores with the IDs 701, 702 and 703 need to be chosen as prototypes/centres of the three clusters.
2. Execute the clustering again considering this time only staff members and turnover. Compare the amount of needed iterations and how the result depends on the initial values. Start the algorithm first with the stores 701, 702 and 703 as cluster centres. Try a second run with the initial stores 702, 705 and 706.

Task 6.7: Classification

We assume that the data warehouse of a company contains an employee dimension and that all the information on employees is stored in an employee table. The following table contains aggregated information for the employees. The last column (attribute amount) covers the amount of entries in the employee table for the combinations of the attribute values shown in the other columns.

Department	Status	age	salary	amount
sale	manager	31 … 35	46K … 50K	30
Sale	Employee	26 … 30	26K … 30K	40
Sale	Employee	31 … 35	31K … 35K	40
Infrastructure	Employee	21 … 25	46K … 50K	20
Infrastructure	manager	31 … 35	66K … 70K	5
Infrastructure	Employee	26 … 30	46K … 50K	3
Infrastructure	manager	41 … 45	66K … 70K	3
Marketing	manager	36 … 40	46K … 50K	10
Marketing	Employee	31 … 35	41K … 45K	4
Office	manager	46 … 50	36K … 40K	4
Office	Employee	26 … 30	26K … 30K	6

1. Create a decision tree (2 split levels) with the target attribute salary. Use the gini index as split criteria with the following generalization: If a set T contains elements from n different classes, then following formula is valid:

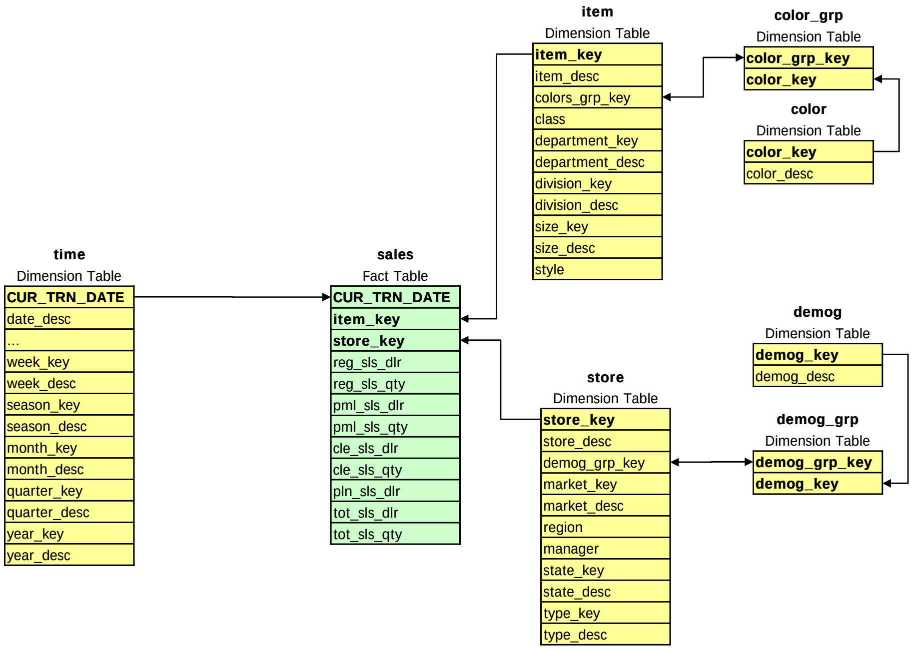
$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

2. How many wrong classifications do you get if the training data is classified by means of the created decision tree?

Exercise 7

Task 7.1: Materialized summary data

Consider the star schema from Task 3.2.:



a) Which options do you have to provide materialized summary data for the fact table sales?

b) Indicate one materialized view that can be used for query 1 of Task 5.1 and has a low runtime for this query?

c) Use the algorithms introduced in the lecture to determine a set of materialized views for the grouping options indicated below. The available storage is sufficient for 5400 rows of materialized views. Use the number of rows in each of these materialized views as a measure for the query costs when querying a view. Start by building the dependency graph.

grouping	Number of rows
C, I, S	5000
Y, I, S	3500
C, I	600
Y, I	600
I, S	300
C, S	185
Y, S	185
S	180
I	120
Y	50
C	50
( )	1

C = CUR_TRN_DATE
I = item_key
S = store_key
Y = year_key

Task 7.2: Bitmap indexes

1. In a bitmap index the TID lists in the leaves of a B-tree are replaced by bit lists. Determine the important factors that define the needed storage. Under which conditions do bit lists need less storage than TID lists?
2. For the processing of some queries several bit lists have to be combined. Indicate the calculation of the predicate A IN (3,4,5,6) by means of the following bit list for attribute A:

A:		[1]				[5]					[10]					[15]	
1:	B_1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
2:	B_2	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
3:	B_3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4:	B_4	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5:	B_5	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0
6:	B_6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7:	B_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
8:	B_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

In a different encoding of bitmap indexes bit lists are deduced from the bitmaps B_j as follows:
$C_j[i] = 1$, if $B_j[i]=1$ or $B_k[i]=1$ for $k<j$

Determine the bitmap index for attribute A according to this encoding and use it to calculate predicate A IN (3,4,5,6).