POZNAN UNIVERSITY OF TECHNOLOGY

Modeling Data Warehouse

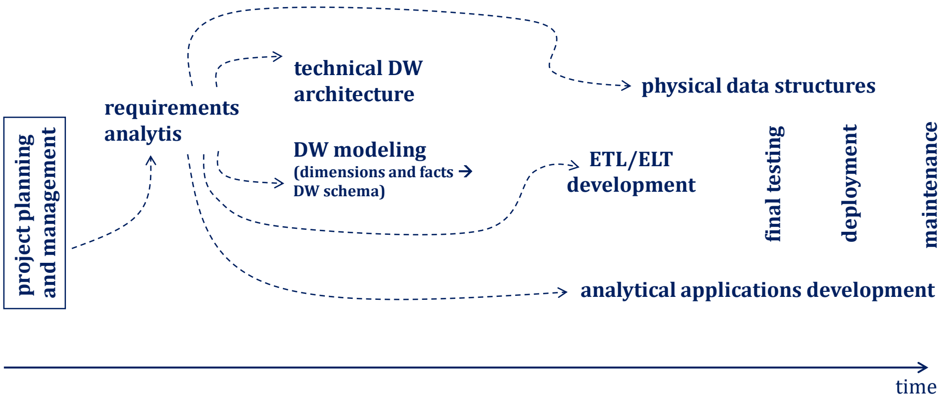
Part 1

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Outline

- **Data warehouse modeling pipeline**
- **Conceptual mulitdimensional data model**
- **Logical multidimensional data models**
- **Relational DW schemas**

DW development pipeline

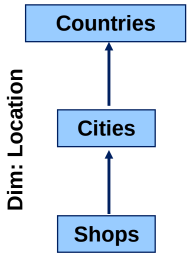


DW Data models

- **Conceptual**
  - **mutlidimensional data model (MD)**
- **Logical (implemenation)**
  - **relational →ROLAP**
  - **multidimensional → MOLAP**
  - **hybrid → HOLAP**

Conceptual DW data model

- **Only two categories of data**
- **Facts**
  - **data to be analyzed in a given context**
    - **sales, phone calls**
    - **characterized quantitatively by measures**
    - **number of intems sold, phone call duration time**
- **Dimensions**
  - **define the context of an analysis**
    - **chocolate sales (product) in Auchan (shop) in months (time)**
  - **typically composed of levels that form hierachies**

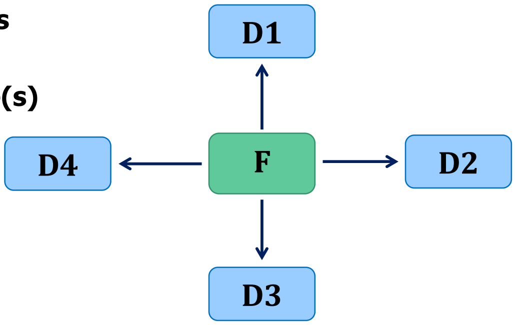


Logical DW data model

- **ROLAP**
  - **star schema**
  - **snowflake schema**
  - **star-flake schema**
  - **fact constellation schema**

DW modeling: some remarks

- Identify facts → key types of transactions
  - commerce: sales transactions
  - banks: financial operations on accounts
  - stock exchange: sell/buy quotations
  - insurance: buying a policy, damage payment
  - telecom: phone calls
- Identify key dimensions
- Design a fact table
- Design dimension table(s)

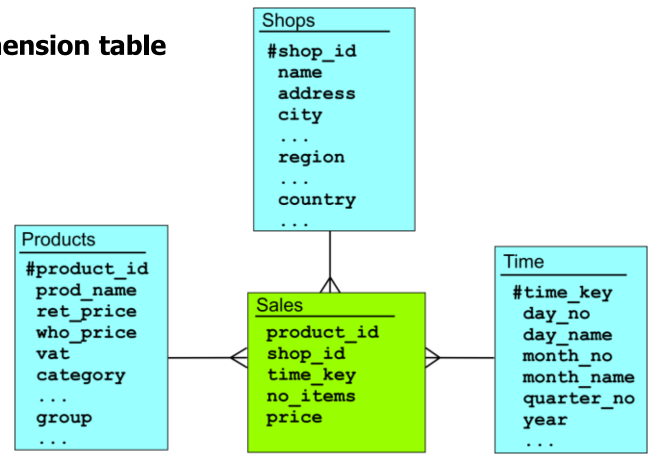


DW modeling: some remarks

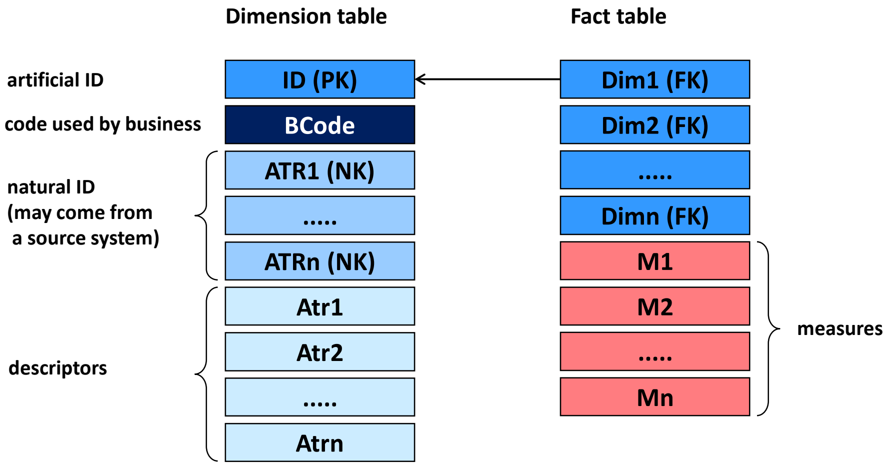
- Fact data: how detailed?
  - storing every single product purchase record
  - storing the value of a whole basket
  - derived attributes
    - net, vat, gross → net, vat, and dynamically computed gross
  - storing only necessary attributes
    - dim table Customer with $8*10^6$ records
    - each customer makes daily 2 phone calls
    - one year time span of a fact table
    - decreasing the length of each fact row by 10B → size decrease by 54GB

ROLAP - star schema

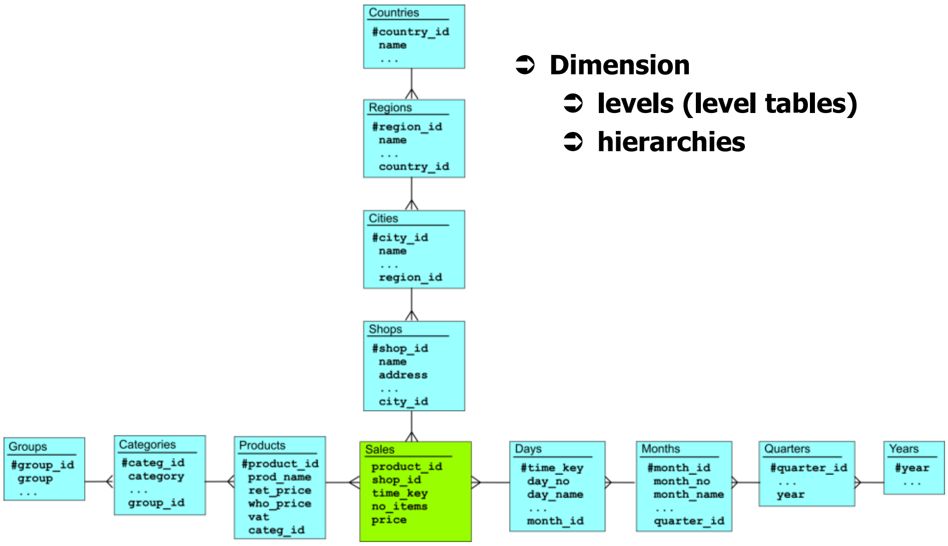
- One fact table
  - measures
- At least one dimension table



Dimension table

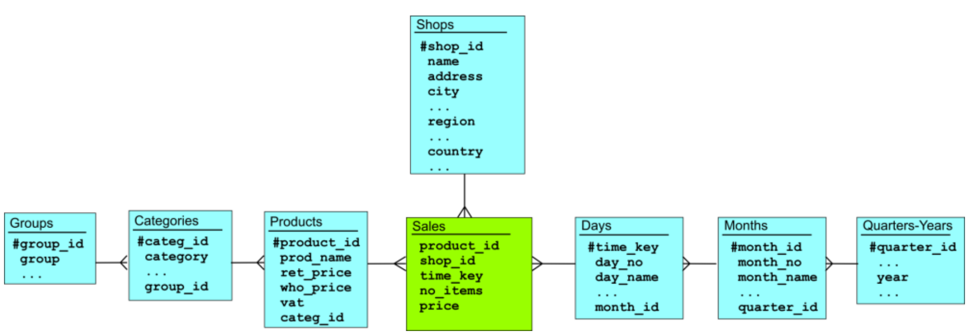


ROLAP - snowflake schema

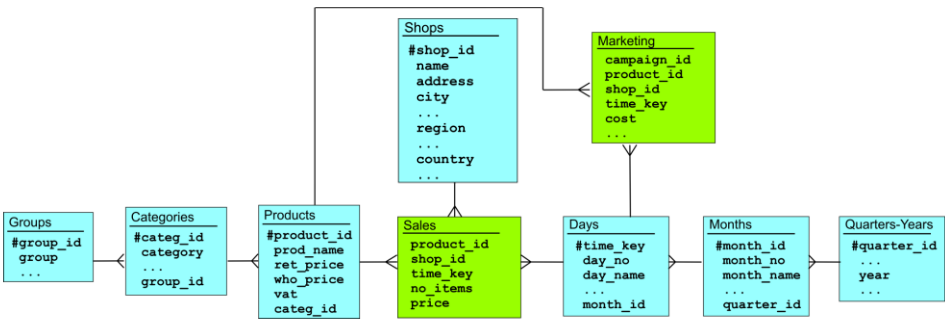


- ➲ **Dimension**
  - ➲ **levels (level tables)**
  - ➲ **hierarchies**

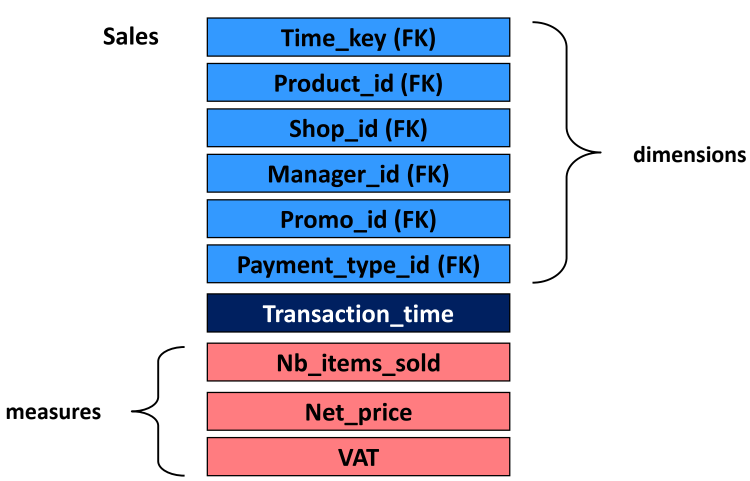
ROLAP - star-flake schema



Fact constellation schema



Fact table



Factless fact

- No explicit measure attribute
- Stores facts that typically represent events

Accident

Accident
Accident_type_id (FK)
Insurance_NO (FK)
Time_key (FK)

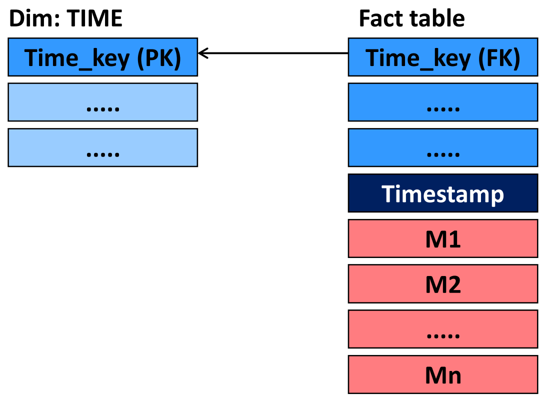
Dimension TIME

- Exists in all DW schemas
- Typical granularity - day
  - Time_key
    - artificial ID: values 1, 2, ..., n
    - date-time type
    - timestamp type
    - numerical type: 11032008 (11-03-2008)

Date_id (PK)
Date
Day_name
Day_no_week
Day_no_month
Day_no_year
Last_day_month
Last_day_year
Week_no_year
Month_name
Month_no
Quarter
Year
Fiscal_day_no_week
Fiscal_day_no_month
Fiscal_day_no_year
FIscal_...
...
Holiday
Holiday_type
Shop_open_holiday
Weekend_day
...

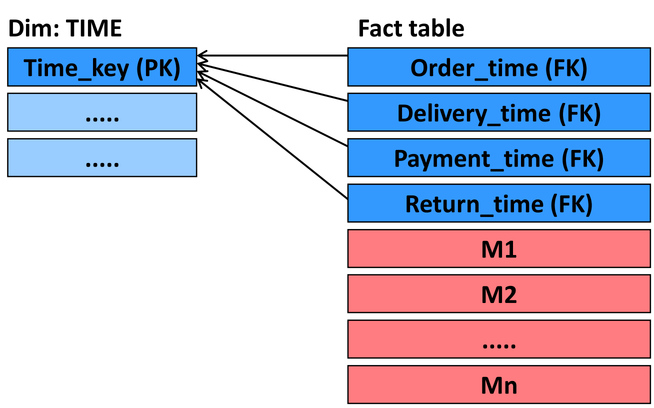
Dimension TIME

- Registering time with granulairty > days
  - timestamp in a fact table



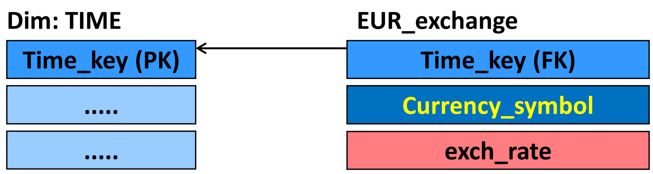
Dimension roles

- The same dimension may play different roles (give contexts) for a fact table
  - e.g., the TIME dimension



Dimension roles

- **Fact dimension: dimension in fact table**



- **Measure being a dimension**
  - **trip length → discretization of values: short, medium, long**

Artificial IDs

- **Created by an ETL process**
- **ID types**
  - **numerical**
    - **efficiency in processing**
    - **chronology represented by values**
    - **typically no semantics**
  - **alphanumerical**
    - **less efficient in processing**
    - **may have semantics**
    - **longer than numerical**
    - **may be constructed as concatenation of natural key and timestamp**

Which schema to use?

- **Advantages of star schema**
  - **less tables → less joins**
  - **simpler DW loading procedure**
- **Disadvantages of star schema**
  - **bigger tables → more I/O to read**
  - **bigger indexes → more I/O to read**
  - **dimension hierarchies may not be visible**

Which schema to use?

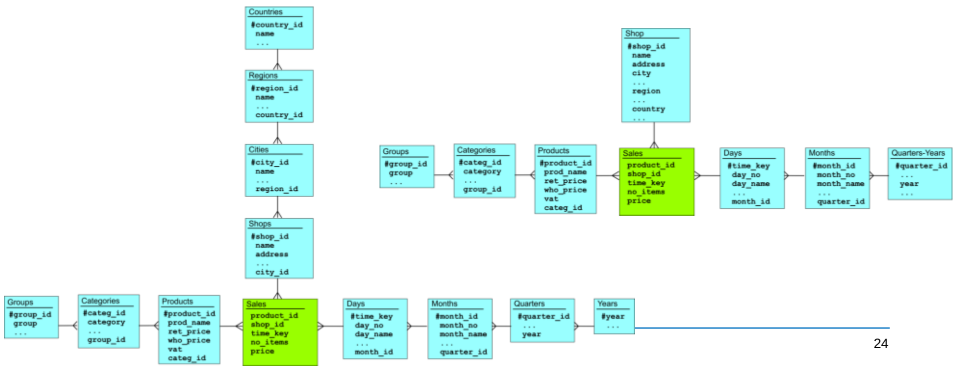
- **Star schema**
  - **denormalized dimension Time → data redundancy**
  - **1 sec granularity: 24*60*60 times the same date is stored**
  - **1 sec granularity and time span of 10 years ⇨ 300 000 000 rows**
  - **DATE datatype occupies 7B**
    - **lost space: 7B * $300*10^6$**

Which schema to use?

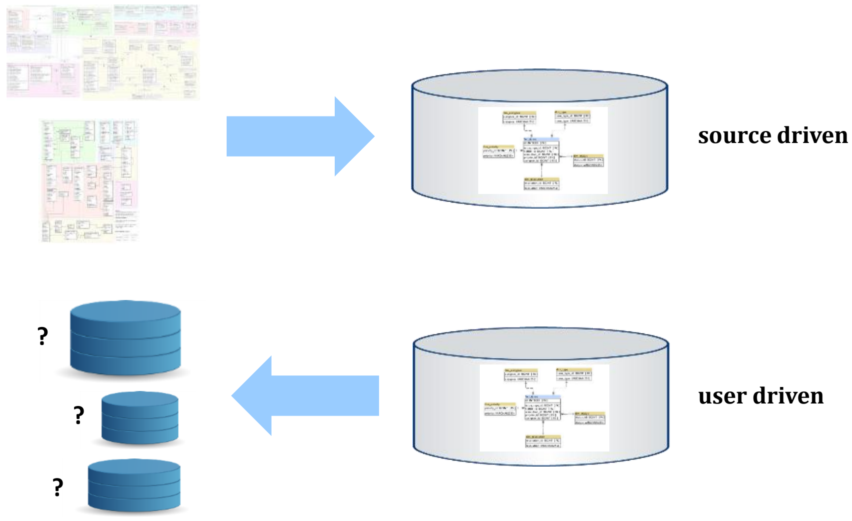
- **Advantages of snowflake schema**
  - **normalized tables → less storage**
  - **clearly visible dimension hierarchies**
- **Disadvantages of snowflake schema**
  - **more tables → more joins**
  - **more complex DW loading procedure**

Which schema to use?

- **Possible solutions**
  - **attributes from different hierarchy levels frequently used together in roll-up → store them in the same level table**
  - **high level attributes rarely used → store them in separate high level tables**
- **Star-flake schema**



Designing DW schema



Designing DW schema

- **Mappings between DS and DW**
  - **DS1.table1 → DW.Dim_table1**
  - **DS1.table2 → DW.Dim_table1**
  - **...**
  - **DS1.table1.attribute1 → DW.Dim_table1.attributeA**
  - **DS1.table3.attribute5 → DW.F_table1.attributeX**
  - **...**

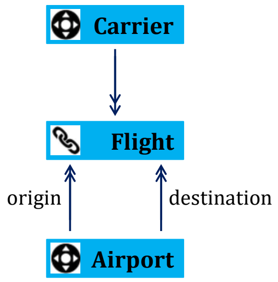
Data Vault Modeling

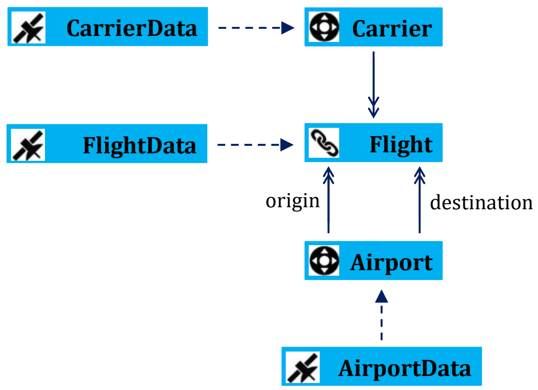
➲ **Model entities**

- **Hub**
  - to store business keys of business objects with some other data
- **Link**
  - connects two or more hubs
  - may be a candidate for a fact
  - may contain its own features (attributes)
- **Satellite**
  - store attributes that belong to a business key (in a hub) or relationship (in a link)
  - attached to only one hub or link

D. Linstedt, M. Olschimke: Building a Scalable Data Warehouse with Data Vault 2.0. Morgan Kaufman, 2016

Data Vault Modeling





Data Vault Modeling

➲ Hub content

- business key (value), may be composed of multiple attributes, like a composite primary key
  - e.g., a natural identifier
- hash key (used to reference the business object in links and satellites; used in joins; plays the role of a primary key)
  - e.g., an artificial identifier
- record source
- business key load date (date + time)
- last seen in a source (optional attribute)

Data Vault Modeling

➲ Hub examples

- Carrier

  - CarrierID (e.g., IATA code)
  - CarrierHashKey
  - LoadDate
  - RecordSource
- Airport

  - AirportCode (e.g., IATA code)
  - AirportHashKey
  - LoadDate
  - RecordSource

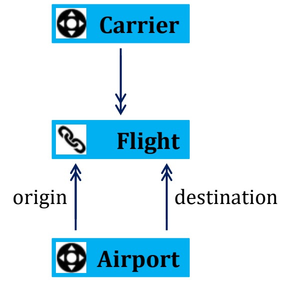
Data Vault Modeling

➲ Link content

- HashKey (counterpart of PK)
- hash keys of connected hubs (counterparts of FKs)
- LoadDate
- RecordSource
- LastSeen

➲ Link example

- Flight
  - HashKey
  - LoadDate
  - RecordSource
  - CarrierHashKey
  - OriginAirportHashKey
  - DestinationAirportHashKey
  - LastSeen



Data Vault Modeling

➲ Satellite

- stores every change to raw data → stores data that evolve in time (like SCDs)
- recommended: to distribute data among various satellites
  - split raw data first by source system and
  - second by rate of change

Data Vault Modeling

➲ Satellite content

- parent hash key (counterpart of FK; part of PK)
- original business features of a satellite
- load date (part of PK)
- load end date (when the record becomes invalid)
- hash diff (hash value of business features; helps in identifying if a record value changed by comparing it with hash of source values)
- extract date (optional; when a record was ingested from a source)

Data Vault Modeling

➲ Satellite example

- FlightData
  - FlightHashKey (FK, PK)
  - DepartureTime
  - ArrivalTime
  - Distance
  - LoadDate (PK)
  - LoadEndDate
  - RecordSource
  - HashDiff