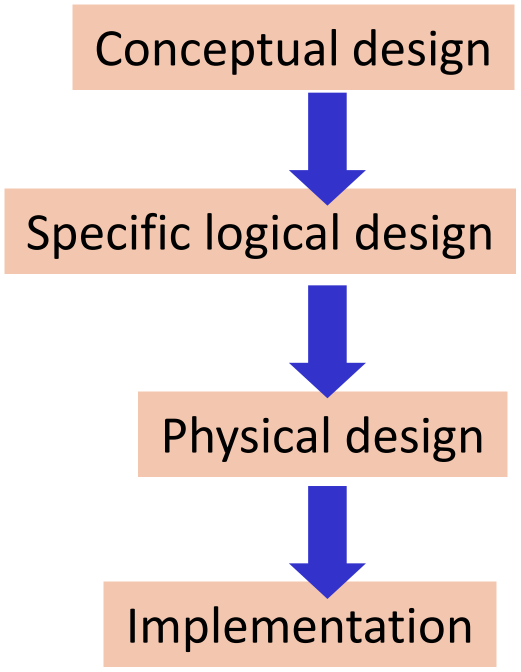
Business intelligence

Unit 2 – Datawarehouse and OLAP
S2-2 – Datawarehouse design

Datawarehouse design

- Does it ring a bell?



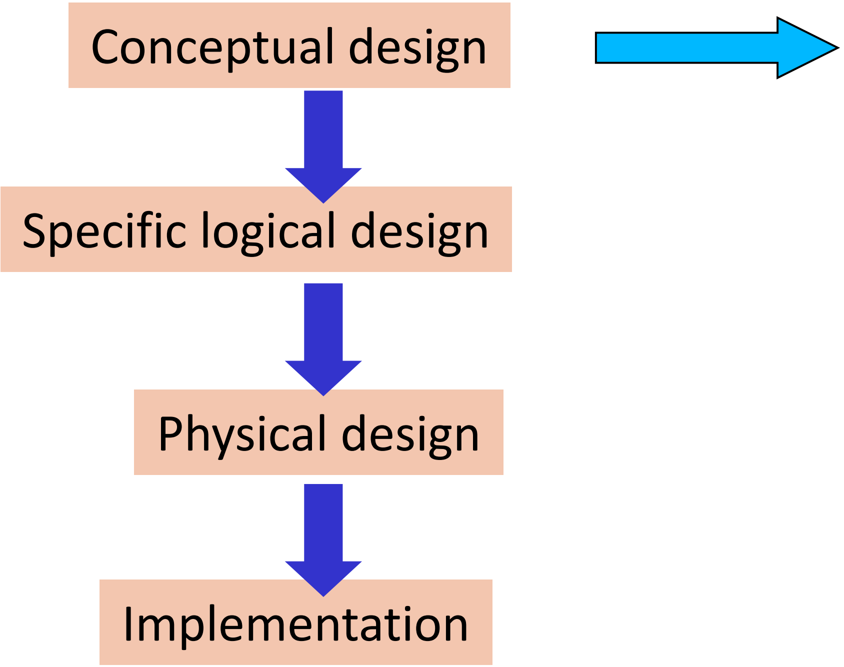
Conceptual design

Specific logical design

Physical design

Implementation

Datawarehouse design


Conceptual design

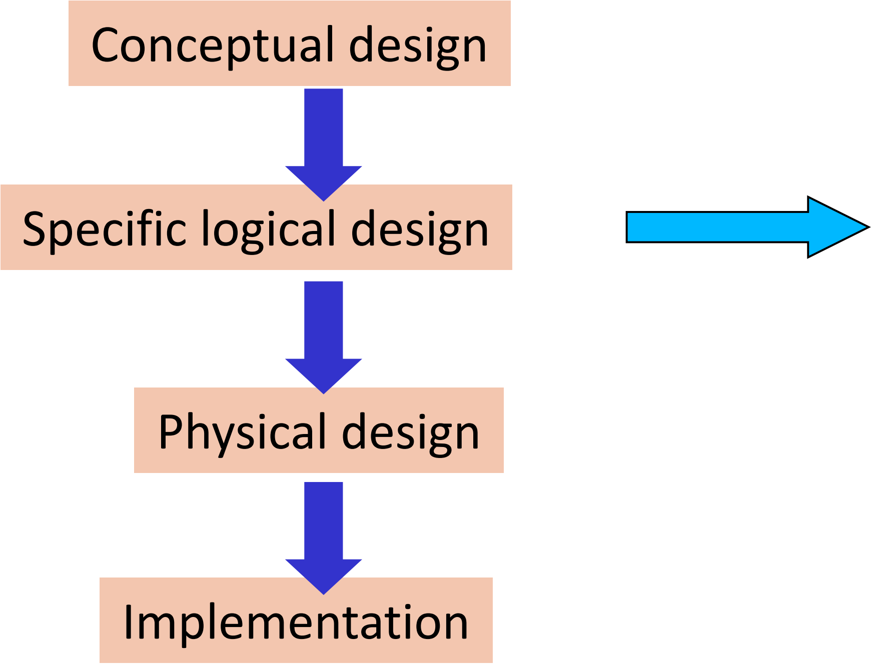
Specific logical design

Physical design

Implementation

- Requirement analysis
  - Identify data sources
  - Identify facts and measures
- Conceptualization
  - Eg: Entity Relationship Model

Datawarehouse design



Conceptual design

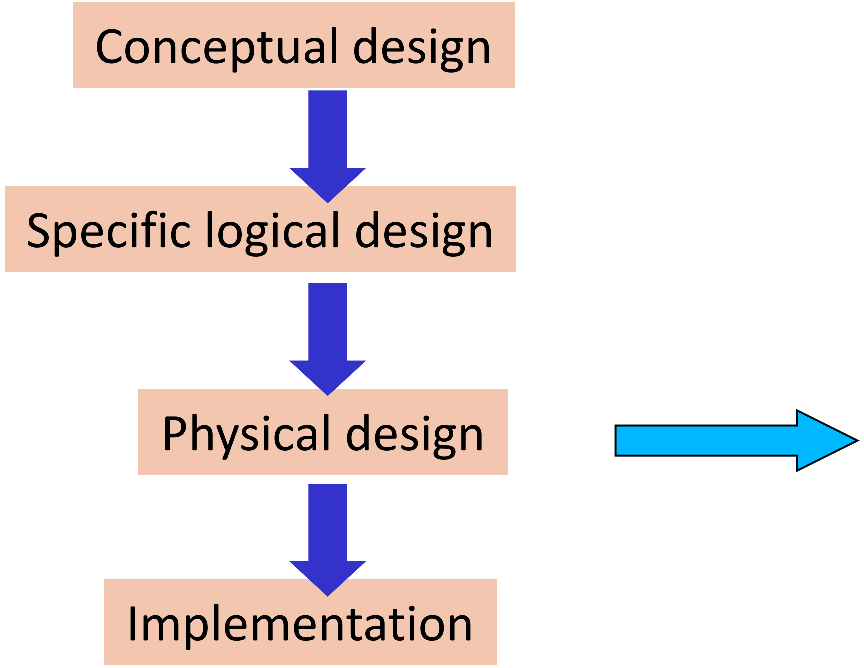
Specific logical design

Physical design

Implementation

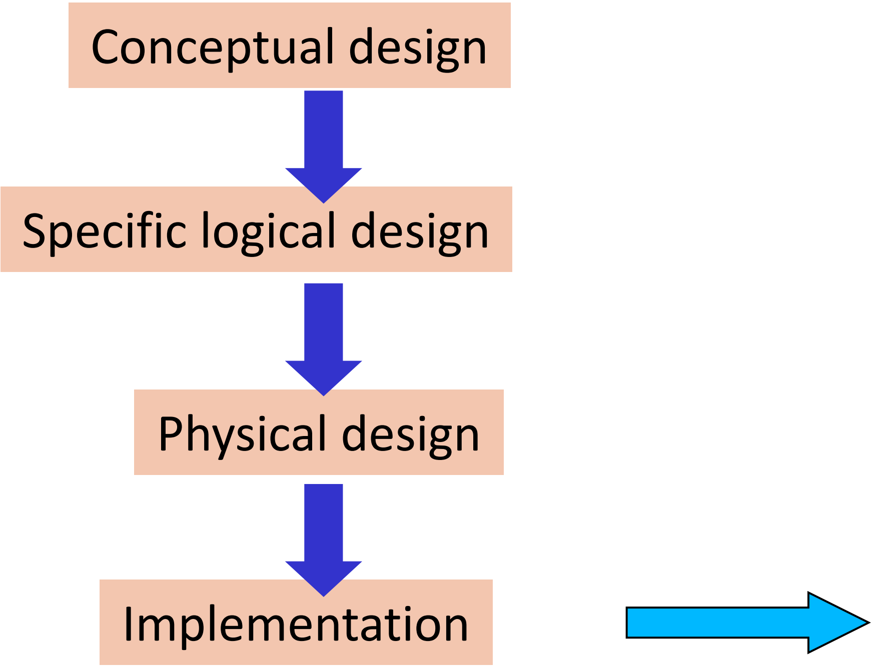
- Multidimensional modeling
- Star, snowflake, both models,
- Methodology Kimball

Datawarehouse design



- Storage management (ROLAP, MOLAP, HOLAP)
  - Big data?
- Integration (ETL ?)design

Datawarehouse design



Conceptual design

Specific logical design

Physical design

Implementation

- Integration implementation
- Analysis (OLAP) tools

Datawarehouse design

- Multidimensional model:
  - it models an activity which is subjected to analysis (**fact**) and **dimensions** that characterize the activity.
    - Composed key
  - relevant information about the event (activity) is represented by a set of indicators (**measures** or fact attributes)
  - descriptive information for each dimension is represented by a set of attributes (**dimension attributes**).
    - Simple key

Entity relationship vs Multidimensional models

- The Multidimensional Model and Entity Relationship have connections, but are different.
- application
  - ER is used for transaction systems
  - MM is used for data analysis
- structure
  - ER identifies and eliminates redundancy relations
  - MM usually includes denormalization
- use
  - ER queries are complex
  - MM queries are simple and efficient

Methodology for multidimensional modeling

- We start from:
  - Knowledge about the domain (possibly CM)
  - Data Sources
  - Indicators: user queries
- Objective: resolve user queries efficiently.
  - Methods focused on logical and physical design
  - [Kimball, 96]: Methodology of 9 steps

Methodology for multidimensional modeling

- 9-step methodology [Kimball, 96] :

1. Select the **process**
2. Select the granularity
3. Identification and conformation of the **dimensions**
4. Selection of the **facts**
5. Storing precalculated values in fact table
6. Complete the dimension tables
7. Select the duration of the database
8. Control of slowly changing dimensions
9. Select priorities and query modes

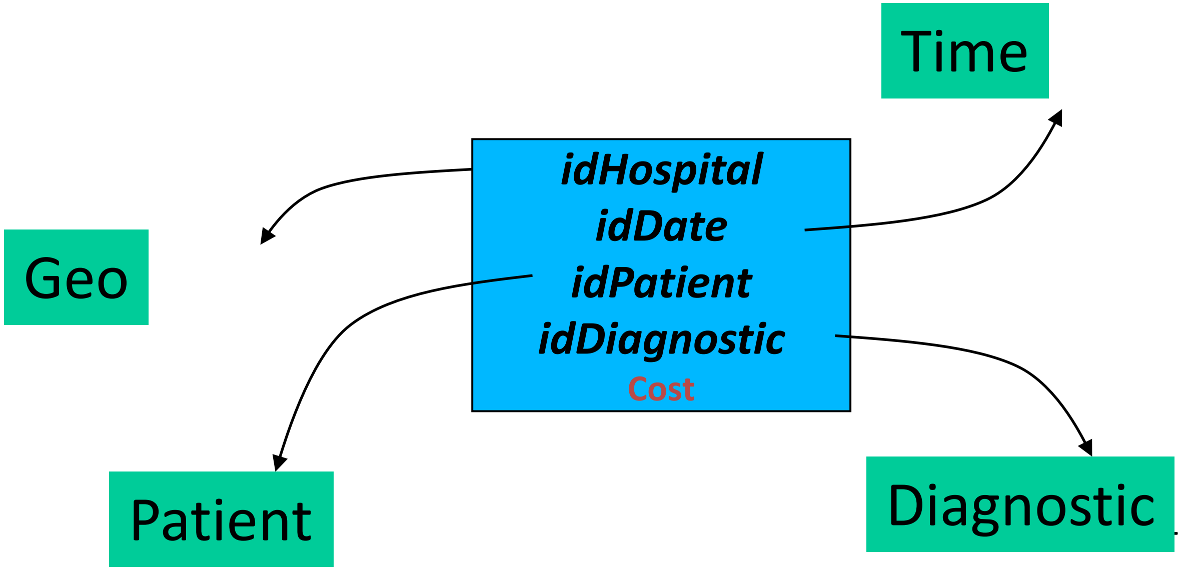
Methodology. Step 1

- Step 1: Select the process
- Process: activity objective of the datawarehouse.
- A process is supported by OLTP systems.
- Start with the most important to the organization.
- Examples: diagnoses, deceased, inventory, billing, …

Methodology. Step 2

- Step 2: Select the granularity
- Granularity: level of detail in which information is stored
- Each fact table and measures are defined
- Example: weekly cost of diagnoses in health centers

¿Days? ¿Weeks?
The level that allows the better analysis: Fine grain
¿Test performed or cost?

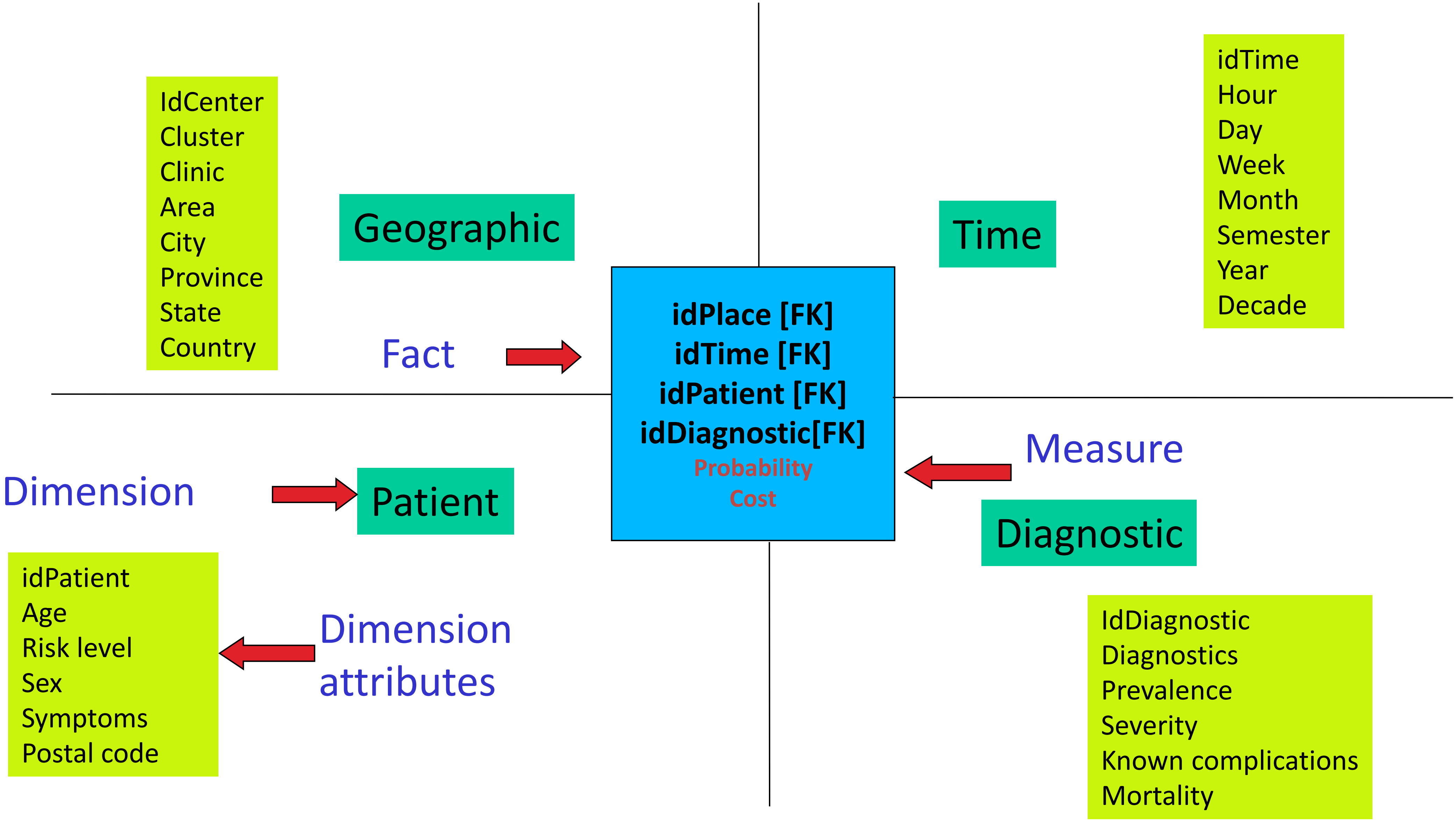


Methodology. Step 3

- Step 3: Identification and conformation of the dimensions
- After learning the facts, the dimensions and attributes are defined
- Dimension: characterization of facts at a level of detail chosen.
- They are descriptive and are query parameters
- Hierarchies between dimension attributes

IdCenter
Cluster
Clinic
Area
City
Province
State
Country

Methodology. Step 3



IdCenter
Cluster
Clinic
Area
City
Province
State
Country

Geographic

Fact

idPlace [FK]
idTime [FK]
idPatient [FK]
idDiagnostic[FK]
Probability
Cost

Time

idTime
Hour
Day
Week
Month
Semester
Year
Decade

Dimension

Patient

Measure

Diagnostic

idPatient
Age
Risk level
Sex
Symptoms
Postal code

Dimension attributes

IdDiagnostic
Diagnostics
Prevalence
Severity
Known complications
Mortality

Methodology. Step 3

- Dimensions define the interest areas for analyzing the facts:
  - Age
  - Age ranges
  - Sex
  - Risk level
  - Nacionality

idPatient
Age
Risk level
Sex
Symptoms
Postal code

- Special attention is paid to time and space

Methodology. Step 3

- Common attributes in the **time** dimension:
  - Day number, month number, year number, number of weeks
  - day of the month (1 .. 31): allows comparisons on the same day in different months (sales by 1 month).
  - weekday (Monday ... Sunday)
  - month-end or Weekend indicators allows comparisons on the last day of the month or day weekend in different months.
  - quarter (1 .. 4): allows analysis of a specific quarter in different years.
  - holiday indicator: allows analysis on days adjacent to a holiday.
  - season (spring, summer, autumn, winter)
  - special event indicator (football, elections, earthquake, ...)

Methodology. Step 4

- Step 4: Select the facts.
- Fact: analyzed information stored in the fact table.
- Useful facts are: Numerical, additives, or in general facts that can be aggregated
  - Because normally large sets of the fact table are queried
- Select events to the granularity of information chosen
  - Cost of the treatment
  - Cost of the tests
  - Deceased status
  - Total daily detected

Methodology. Step 4

- Facts
  - Additives: they can be added in all dimensions.
    - Activity data are usually additives.
    - Eg: sales, units, money.
  - Semi-Additive: they can be added only in some dimensions.
    - Intensity data are not usually additives.
    - Eg: Stock,
    - Some dimensions may be added, but not in time.
    - Eg: total existing stocks.
  - Non-Additives: they can not be added in any dimension.
    - Eg: temperature, unit price, percentage, ...
    - Can be aggregated by average values
    - You can also include dummy variables (0.1) to indicate occurrence.

Methodology. Step 4

- Fact Tables:
  - Transactional: represent detailed events in space time.
    - Maximum level of exploration.
  - Factless: contain no measures, only the occurrence of certain events.
    - Use to establish relations between dimensions.
    - Eg students to class attendance, negative analysis (products that are not sold).
  - Snapshot: Each row is an instant of time. Describe the state of the facts in a particular moment in time. Normally includes semi-additive and non-additive.
    - They are often taken at predefined intervals.
    - Also cumulative.

Methodology. Step 5-7

- Step 5: Store precalculated values in fact table
  - Include derived attributes that may be useful (see non-additive)
  - For example: differences, decomposed values (eg numerator and denominator) or calculated (amount as the price * units)
- Step 6: Complete the dimension tables
  - Add textual descriptions to the dimensions.
  - Intuitive and understandable
- Step 7: Select of the duration of the database
  - Set date from which store data.
  - It depends on the problem: valid data, necessary data, data not available, ...

Methodology. Step 8-9

- Step 8: Control slowly changing dimensions (SCD)
  - When an attribute changes value but not the key
  - Example: change in marital status, professional category, ….
  - Some solutions to slowly changing dimensions:
    - Type 1: overwrite a changed dimension attribute.
    - Type 2: create a new dimension record.
      - Current value (active, valid, …) + validity date
      - **Note**: Business key are repeated. *Handle queries with care.*
    - Type 3: establish an alternate attribute, so that both the old and the new are accessible.
      - Limited number of changes?
    - Type 4: mini-dimension (also historical table)
    - Also combinations: types 4, 5 (4+1), 6 (1+2+3), 7
- Step 9: Select the priorities and query modes (physical design)

Step 8: Control slowly changing dimensions

TABLA DE HECHOS

MEDIDAS	PROYECTO	INVESTIGADOR
1	P1	I1
2	P2	I1
3	P3	I1
4	P4	I1

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Categoría	Facultad
I1	Manolo	M. Campos	AYD	Informática

TIPO 1: REESCRIBIR CATEGORÍA

TIPO 2: NUEVO REGISTRO. MANTENER BK

TABLA DE HECHOS

MEDIDAS	PROYECTO	INVESTIGADOR
1	P1	I1
2	P2	I2
3	P3	I3
4	P4	I3

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Categoría	Facultad	FECHA_INI	FECHA_FIN	VIGENTE
I1	Manolo	M. Campos	AYD	Informática	2005	2009	N
I2	Manolo	M. Campos	CD	Informática	2010	2016	N
I3	Manolo	M. Campos	TU	Informática	2017	-	Y

TIPO 3: NUEVO ATRIBUTO

TABLA DE HECHOS

MEDIDAS	PROYECTO	INVESTIGADOR
1	P1	I1
2	P2	I1
3	P3	I1
4	P4	I1

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Categoría AC	Categoría	Facultad	FECHA_INI	FECHA_FIN
I1	Manolo	M. Campos	TU	CD	Informática	2005	2009

TIPO 4: MINIDIMENSION

TIPO 5: MINIDIMENSION +OUTRIGGER

TABLA DE HECHOS

MEDIDAS	PROYECTO	INVESTIGADOR	ESTADO
1	P1	I1	E1
2	P2	I1	E2
3	P3	I1	E3
4	P4	I1	E3

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Facultad	FECHA	ESTADO
I1	Manolo	M. Campos	Informática	2005	E1

MINIDIMENSION ESTADO

PK	Categoría ACTUAL
E1	AYD
E2	CD
E3	TU

Methodology: Errors / guidelines (Kimball)

- Error 1: Not to share dimensions between different fact tables.
  - Unify master files. Sex {'M', 'F'}.
- Error 2: Not to unify the facts from different fact tables
  - Although coming from different departments of the company and other computer systems. Example: retail and enterprise sale.
- Error 3: Ignore the aggregate tables and always add more hardware to address performance issues.
- Error 4: Forget the highest level of detail in the entity-relationship model.
  - Maximum detail in 3 areas: staging, relational and dimensional.
- Error 5: Mix facts of different granularity on the same fact table.
  - It is better to create tables that contain precalculated aggregates for common queries. Each granularity in a separate table.

Methodology: Errors / guidelines (Kimball)

- Error 6: Create a dimensional model to solve a particular report.
- Error 7: Adding dimensions to a fact table before setting its granularity.
  - The fact table only contains FK and measures.
  - No decompose the dimensions in the fact table.
- Error 8: Create "smart keys" to relate a dimension table to a fact table.
  - Key numbers are auto-increment (even for the time dimension)
  - Why:
    - Heterogeneous data sources keep their own primary key.
    - Changes in source applications should not affect the dw.
    - Performance (storage size and comparison speed).

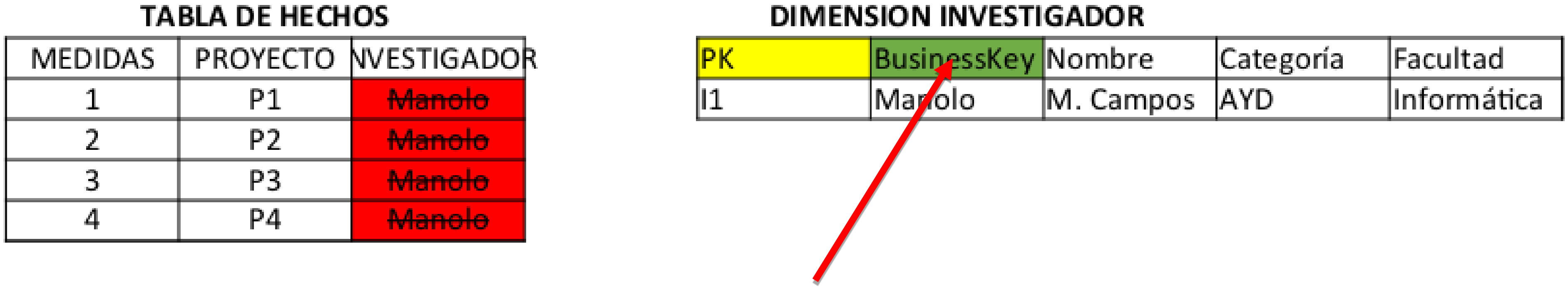


TABLA DE HECHOS

MEDIDAS	PROYECTO	NVESTIGADOR
1	P1	~~Manolo~~
2	P2	~~Manolo~~
3	P3	~~Manolo~~
4	P4	~~Manolo~~

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Categoría	Facultad
I1	Manolo	M. Campos	AYD	Informática

Methodology: Errors / guidelines (Kimball)

- Error 9: Not to face slowly changing dimensions.
- Error 10: Splitting hierarchies and hierarchy levels into multiple dimensions.

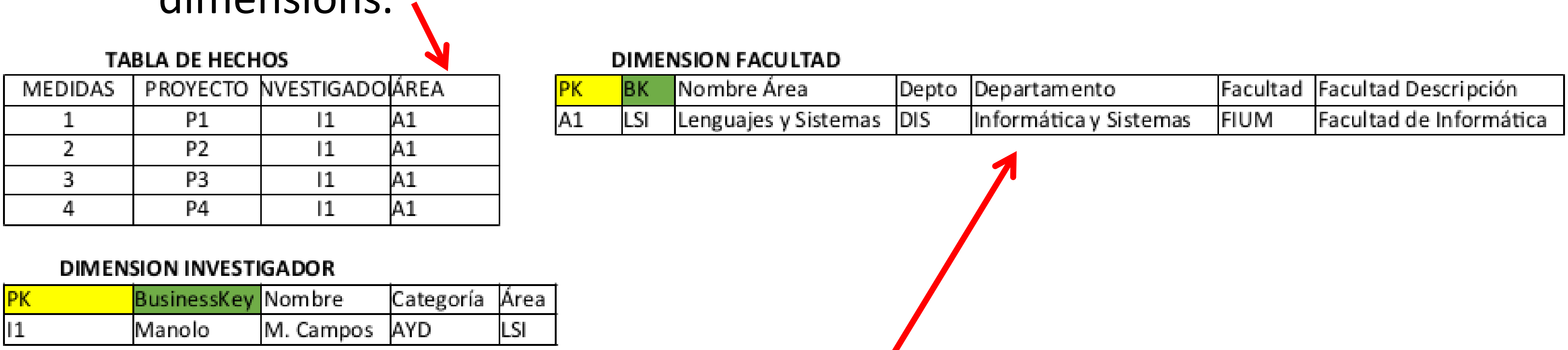


TABLA DE HECHOS

MEDIDAS	PROYECTO	NVESTIGADO	ÁREA
1	P1	I1	A1
2	P2	I1	A1
3	P3	I1	A1
4	P4	I1	A1

DIMENSION FACULTAD

PK	BK	Nombre Área	Depto	Departamento	Facultad	Facultad Descripción
A1	LSI	Lenguajes y Sistemas	DIS	Informática y Sistemas	FIUM	Facultad de Informática

DIMENSION INVESTIGADOR

PK	BusinessKey	Nombre	Categoría	Área
I1	Manolo	M. Campos	AYD	LSI

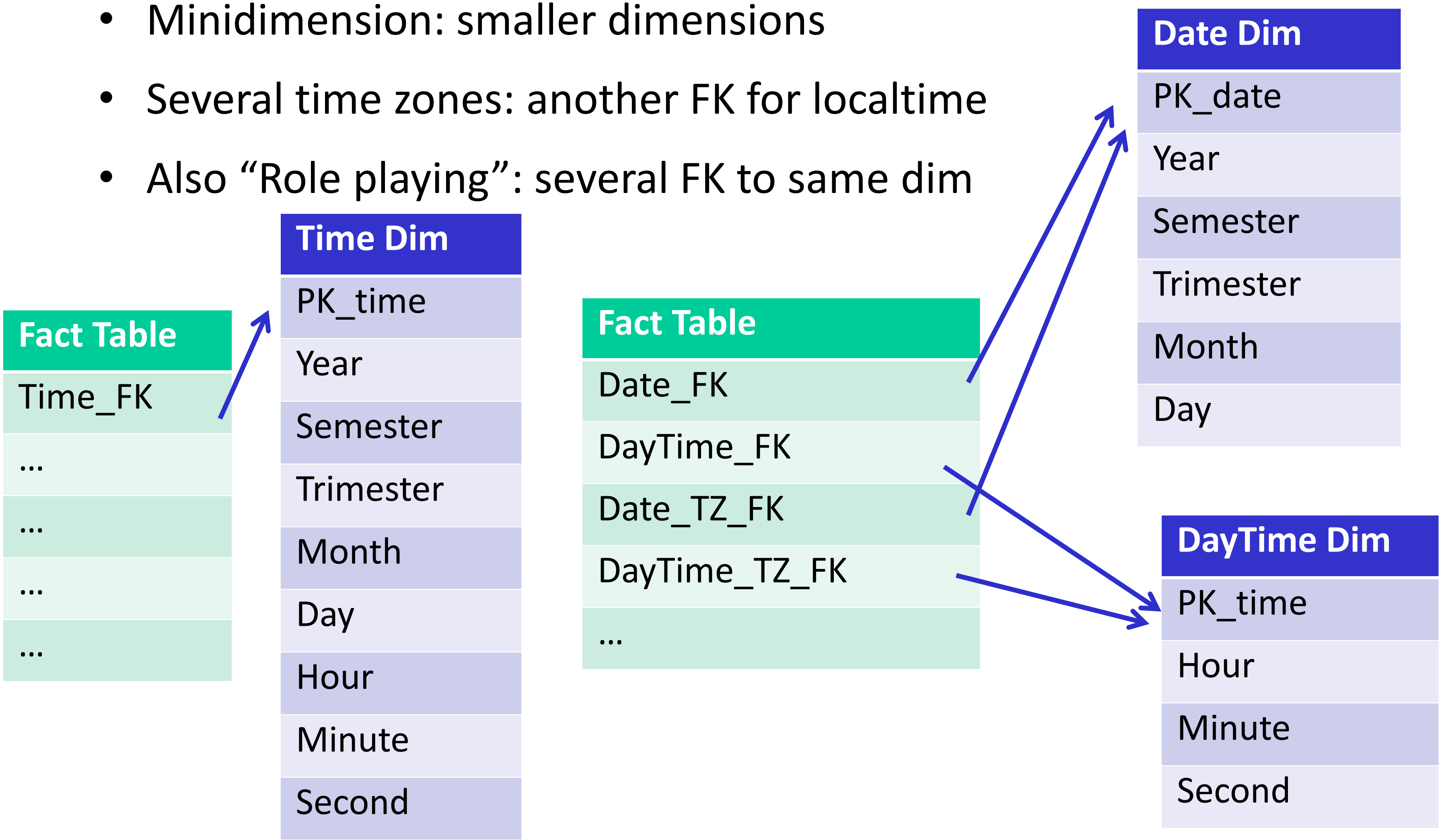
- Error 11: Shorten the descriptions in the dimension tables with the intention of reducing the space required.
  - The dimensions are the interface that users have to browse.
  - They take up little space in relation to the facts.
- Error 12: Include text attributes in a fact table, if done with the intention of filtering or grouping.

DW Design techniques

- Date and time
  - Minidimension
  - Several time zones
- (Shrunken) Rollup dimension
- Junk dimension
- Degenerate dimension
- Bridge table
- Variable depth hierarchies
- Outrigger dimension
- Snowflake dimension (normalization)

DW Design techniques

- Date and time
  - Minidimension: smaller dimensions
  - Several time zones: another FK for localtime
  - Also “Role playing”: several FK to same dim



Fact Table
Time_FK
…
…
…
…

Time Dim
PK_time
Year
Semester
Trimester
Month
Day
Hour
Minute
Second

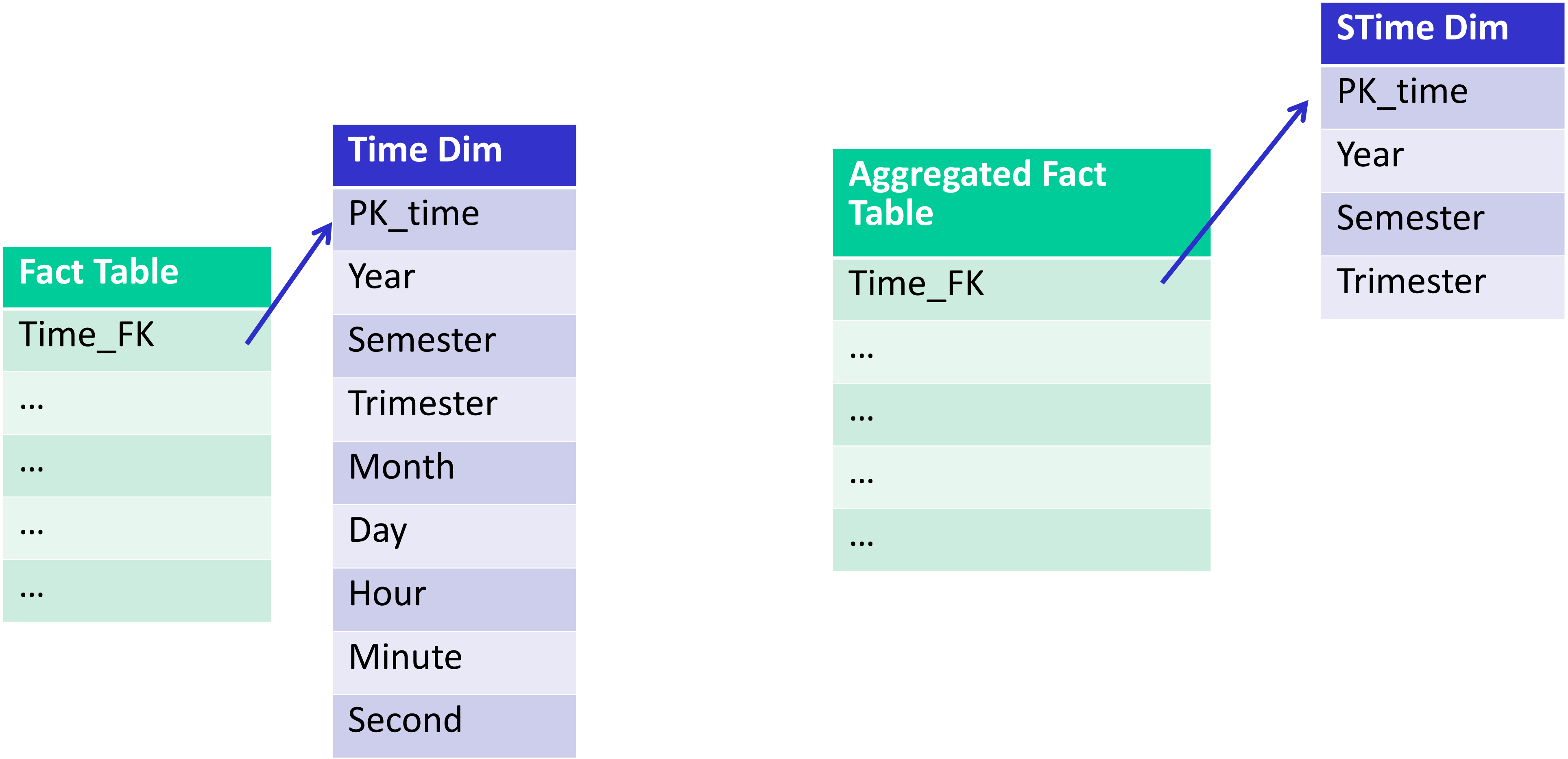
Fact Table
Date_FK
DayTime_FK
Date_TZ_FK
DayTime_TZ_FK
…

Date Dim
PK_date
Year
Semester
Trimester
Month
Day

DayTime Dim
PK_time
Hour
Minute
Second

DW Design techniques

- (Shrunken) Rollup dimension
  - For Aggregatted fact tables



Fact Table

Time_FK

…

…

…

…

Time Dim

PK_time

Year

Semester

Trimester

Month

Day

Hour

Minute

Second

Aggregated Fact Table

Time_FK

…

…

…

…

STime Dim

PK_time

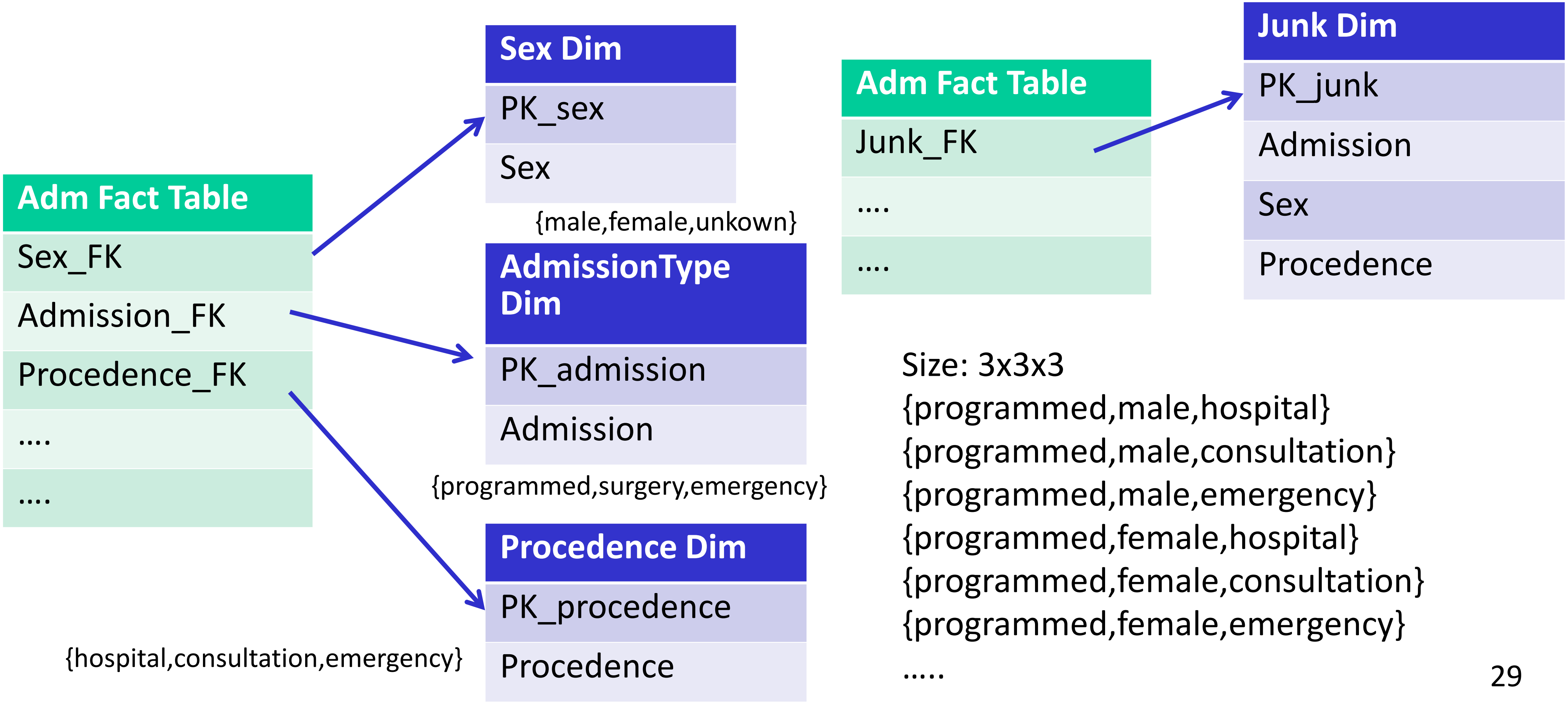
Year

Semester

Trimester

DW Design techniques

- Junk dimension
  - Fact tables with many dimensions with few values
  - Junk dimension holds cartesian product of small dimensions



Adm Fact Table
Sex_FK
Admission_FK
Procedence_FK
….
….

Sex Dim
PK_sex
Sex
{male,female,unkown}

AdmissionType Dim
PK_admission
Admission
{programmed,surgery,emergency}

Procedence Dim
PK_procedence
Procedence
{hospital,consultation,emergency}

Adm Fact Table
Junk_FK
….
….

Junk Dim
PK_junk
Admission
Sex
Procedence

Size: 3x3x3
{programmed,male,hospital}
{programmed,male,consultation}
{programmed,male,emergency}
{programmed,female,hospital}
{programmed,female,consultation}
{programmed,female,emergency}
…..

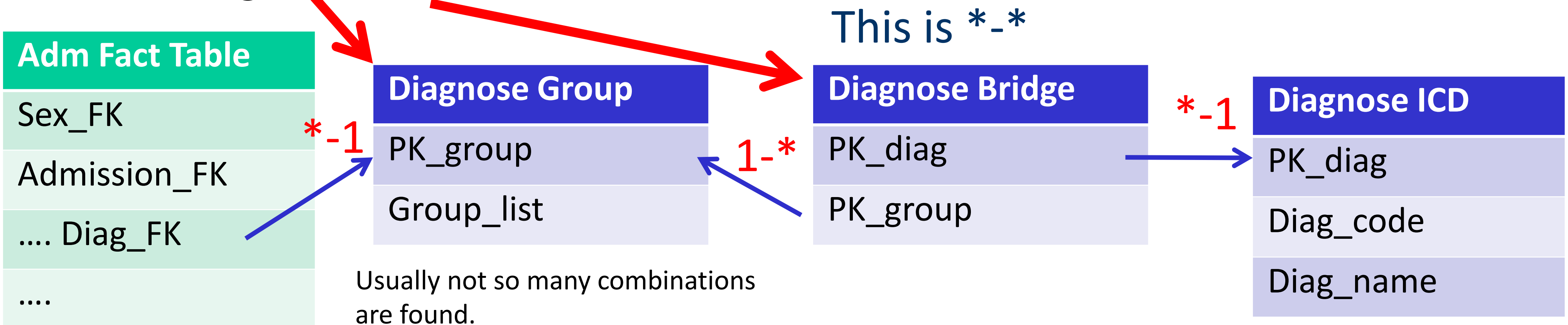
DW Design techniques

- Degenerate dimension
  - Only contains the PK. Eg: episode number, order id, …
  - Store it in the fact table (number)
  - Useful for grouping
  - **Not for filtering**

Adm Fact Table
Sex_FK
Admission_FK
Procedence_FK
….
Episode_number:bigint
….

DW Design techniques

- Problem M-N (*-*) associations between fact and dimensions
  - Eg: admission has several discharge diagnoses
- Is this the right granularity? Solved in fact-tables!!!
  - Change granularity to diagnose? Another fact table?
- Standard solution: “Bridge table” with 2 additional tables
  - Group table: to keep association 1-* between fact and dimension
  - Bridge table between fact-table and dimension or dimension and values



This is *-*

Adm Fact Table
- Sex_FK
- Admission_FK
- .... Diag_FK
- ....
- ....

*-1

Diagnose Group
- PK_group
- Group_list

1-*

Diagnose Bridge
- PK_diag
- PK_group

*-1

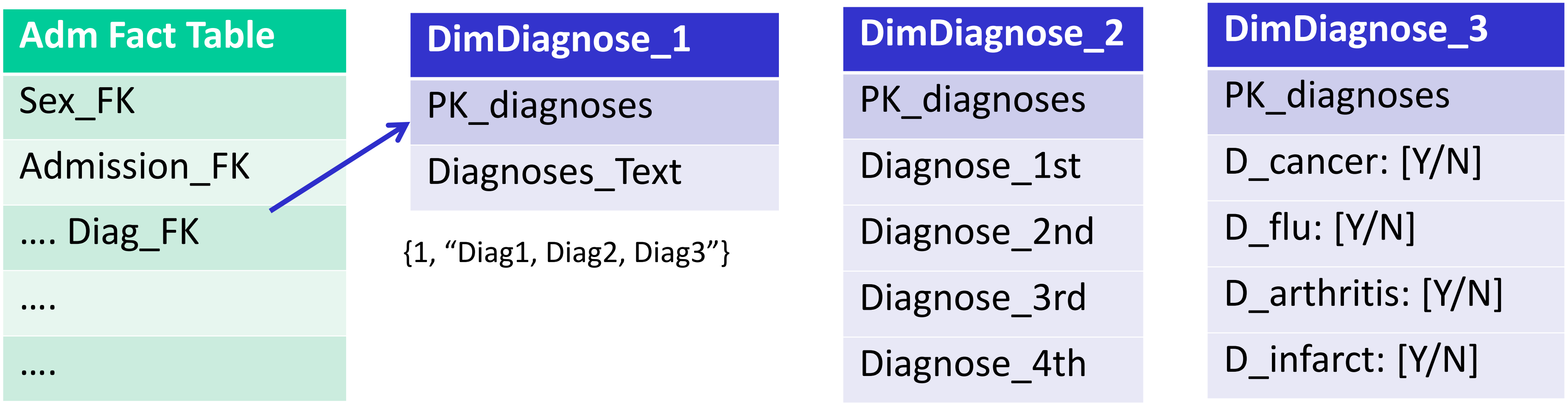
Diagnose ICD
- PK_diag
- Diag_code
- Diag_name

Usually not so many combinations are found.
{1, “Diag1, Diag2, Diag3”}
{2, “Diag1, Diag3”}

{1,”Diag1”, “Diagnose for …”}
{2,”Diag2”, “Diagnose for …”}

DW Design techniques

- Alternatives:
  - String concatenation: “diag1 # diag 2” (Pathstring)
    - Text processing in query time?
  - Multiple attributes in the dimension. Eg. Diagnose1: diag1, diagnose2: diag2.
    - Are they sorted? Order is important? First value? Second value?
  - Limited number of attributes? -> Multiple dummy attributes: Diagno



Adm Fact Table
Sex_FK
Admission_FK
…. Diag_FK
….
….

DimDiagnose_1
PK_diagnoses
Diagnoses_Text

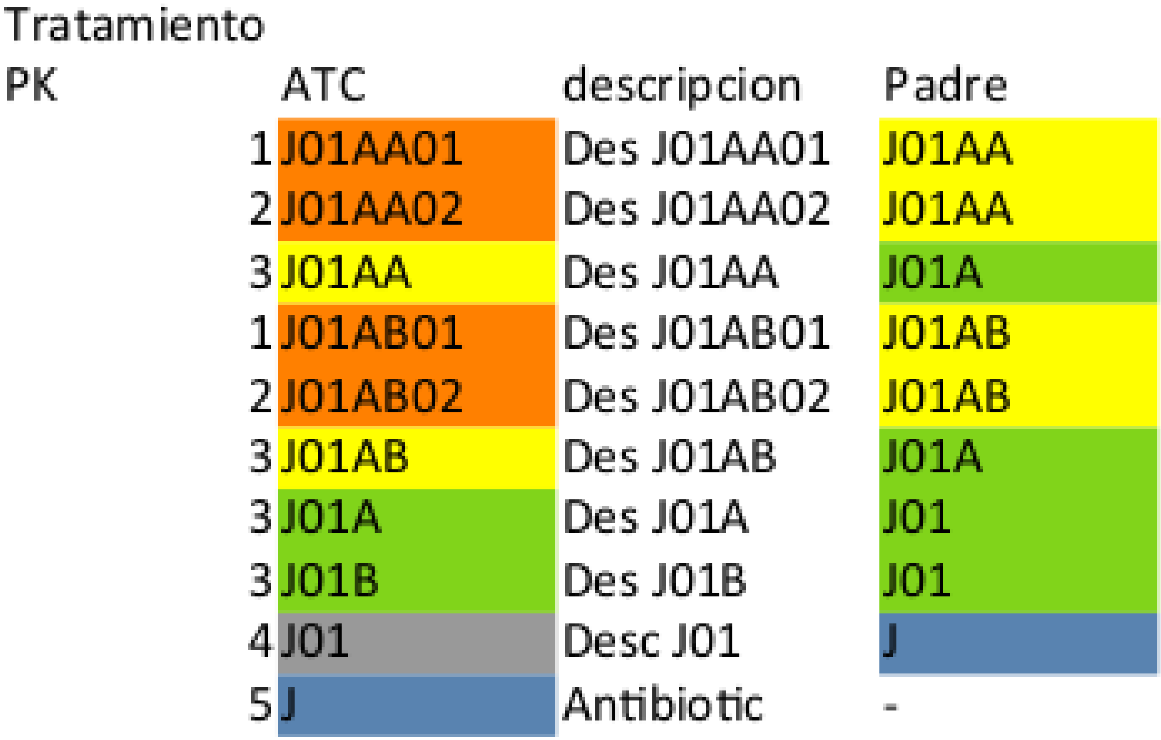
{1, “Diag1, Diag2, Diag3”}

DimDiagnose_2
PK_diagnoses
Diagnose_1st
Diagnose_2nd
Diagnose_3rd
Diagnose_4th

DimDiagnose_3
PK_diagnoses
D_cancer: [Y/N]
D_flu: [Y/N]
D_arthritis: [Y/N]
D_infarct: [Y/N]

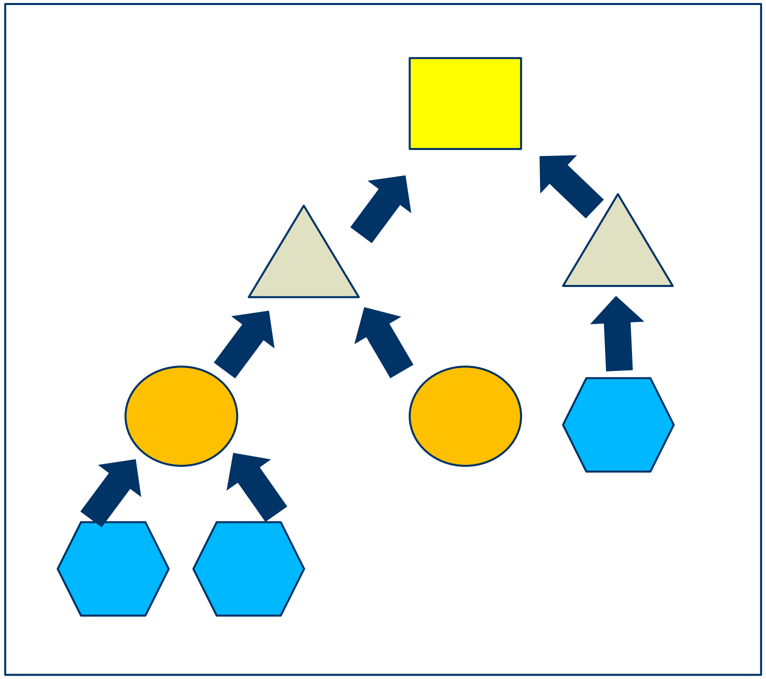
DW Design techniques

- Variable depth hierarchies
  - Recursive queries in SQL and OLAP are limited



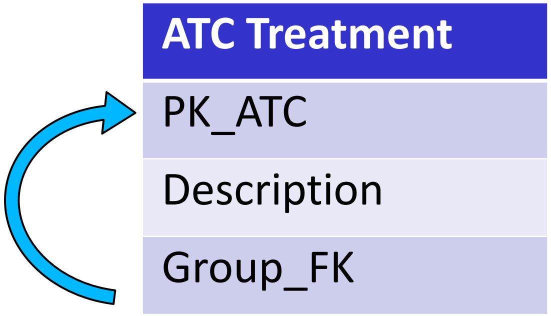
Tratamiento

PK	ATC	descripcion	Padre
1	J01AA01	Des J01AA01	J01AA
2	J01AA02	Des J01AA02	J01AA
3	J01AA	Des J01AA	J01A
1	J01AB01	Des J01AB01	J01AB
2	J01AB02	Des J01AB02	J01AB
3	J01AB	Des J01AB	J01A
3	J01A	Des J01A	J01
3	J01B	Des J01B	J01
4	J01	Desc J01	J
5	J	Antibiotic	-



DW Design techniques

- Solutions:
  - Busines decisión: Not all levels apply: “Ceuta” and “Melilla” are not “Province”. Use business significative value
  - Pathstring with complete path in hierarchy (same as with bridge tables)
  - Slightly ragged: if range is small, force fix depth (same as with bridge tables)
  - Bridge table with depth level (closure)
    - Foreing key to dimension + depth attribute

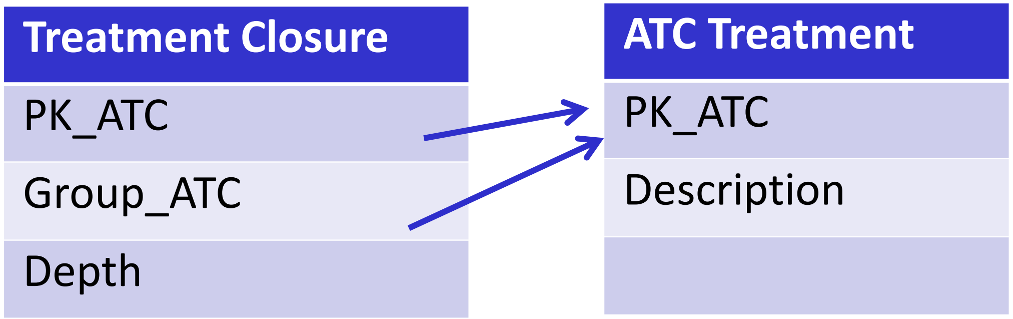


ATC Treatment

PK_ATC

Description

Group_FK



Treatment Closure

PK_ATC

Group_ATC

Depth

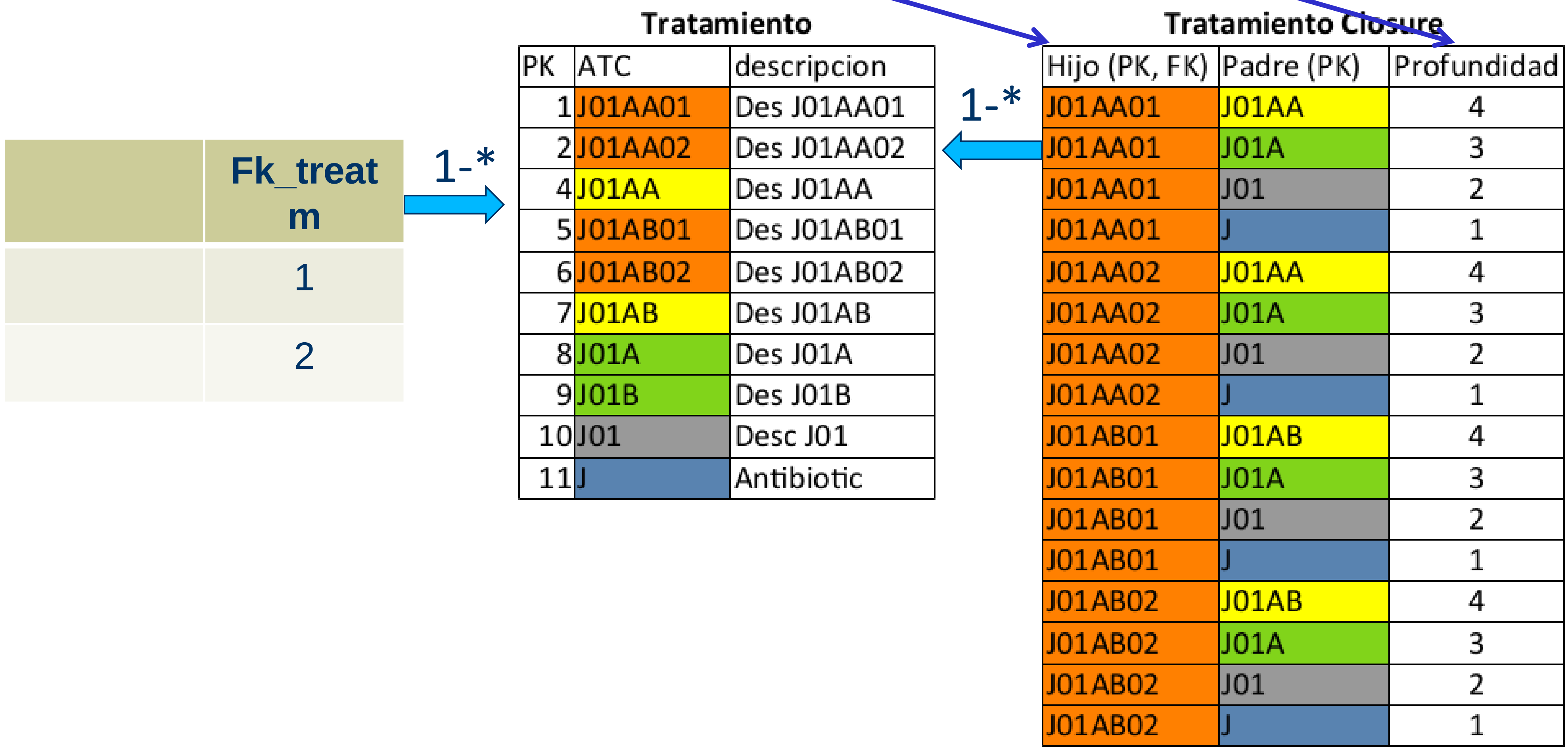
ATC Treatment

PK_ATC

Description

DW Design techniques

- Bridge table: additional table.
- Foreing key to dimension + depth attribute
- Combined PK



	Fk_treat m
	1
	2

Tratamiento

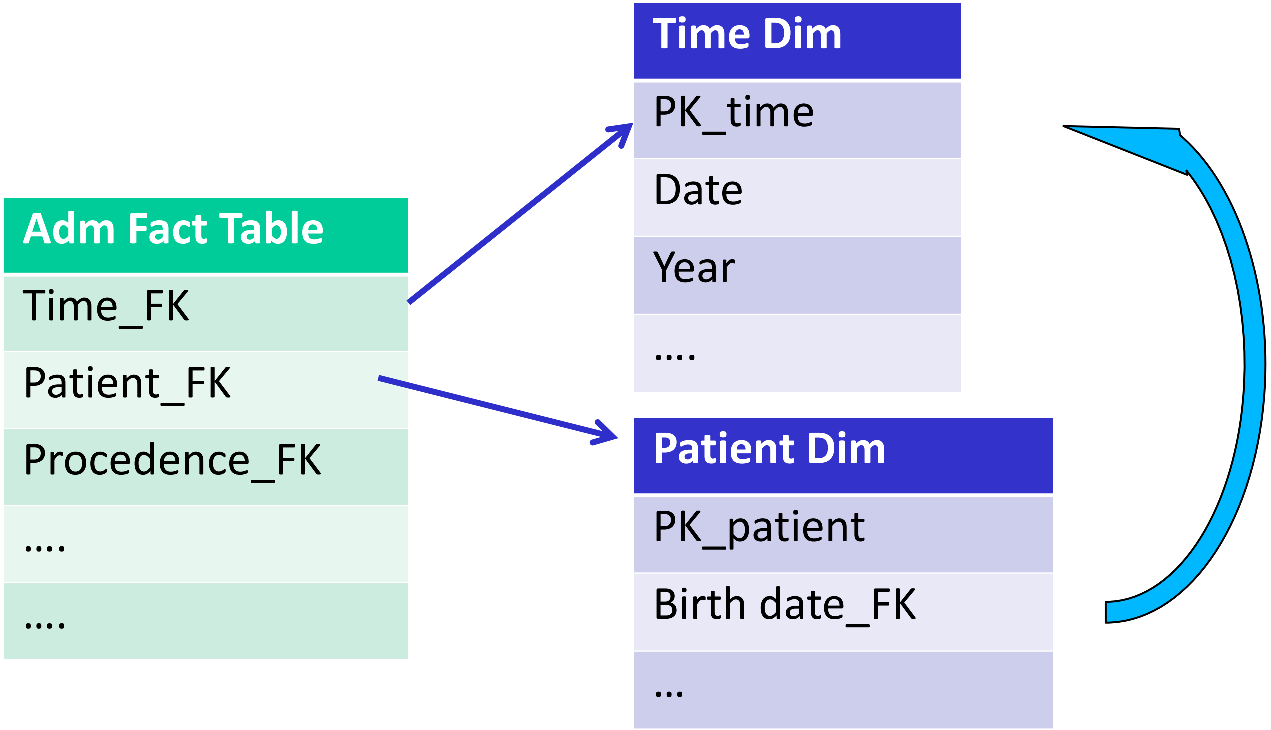
PK	ATC	descripcion
1	J01AA01	Des J01AA01
2	J01AA02	Des J01AA02
4	J01AA	Des J01AA
5	J01AB01	Des J01AB01
6	J01AB02	Des J01AB02
7	J01AB	Des J01AB
8	J01A	Des J01A
9	J01B	Des J01B
10	J01	Desc J01
11	J	Antibiotic

Tratamiento Closure

Hijo (PK, FK)	Padre (PK)	Profundidad
J01AA01	J01AA	4
J01AA01	J01A	3
J01AA01	J01	2
J01AA01	J	1
J01AA02	J01AA	4
J01AA02	J01A	3
J01AA02	J01	2
J01AA02	J	1
J01AB01	J01AB	4
J01AB01	J01A	3
J01AB01	J01	2
J01AB01	J	1
J01AB02	J01AB	4
J01AB02	J01A	3
J01AB02	J01	2
J01AB02	J	1

DW Design techniques

- Outrigger dimension
  - Exception!!
  - When a dimension has a FK to another dimension.
  - Eg.: “Registered user” and “Unregistered user”
    - “Birth date” links to “Date Dimension”



Adm Fact Table
Time_FK
Patient_FK
Procedence_FK
….
….

Time Dim
PK_time
Date
Year
….

Patient Dim
PK_patient
Birth date_FK
…

Normalization

- Snowflake dimensions
  - When a hierarchical relationship in a dimension table is normalized, low-cardinality attributes appear as secondary tables connected to the base dimension table by an attribute key.
  - It represents hierarchical data accurately, yet a flattened denormalized dimension table contains exactly the same information as a snowflaked dimension.
  - Only in big dimension tables!
  - But you should avoid snowflakes because:
    - it is difficult for business users to understand and navigate snowflakes.
    - They can also negatively impact query performance.

Normalization

DIMENSIÓN ESTRUCTURA

PK	BK	Nombre Área	Depto	Departamento	Depto ...	Facultad	Facultad Descripción	Fac ...
A1	LSI	Lenguajes y Sistemas	DIS	Informática y Sistemas	[varios]	FIUM	Facultad de Informática	[varios]
A2	ISA	Informática y Automática	DIS	Informática y Sistemas	[varios]	FIUM	Facultad de Informática	[varios]

Replicated data

DIMENSION ÁREA

PK	BK	Nombre Área	Depto
A1	LSI	Lenguajes y Sistemas	DIS
A2	ISA	Informática y Automática	DIS

FK

DIMENSION DEPTO

Depto	Departamento	Depto ...	Facultad
DIS	Informática y Sistemas	[varios]	FIUM

FK

DIMENSION FACULTAD

Facultad	Facultad Descripción	Fac ...
FIUM	Facultad de Informática	[varios]

No replicated, but join needed

Normalization

- A set of conditions on table structure that improves maintenance. Normalization removes processing anomalies:
  - Update
  - Inconsistent Data
  - Addition
  - Deletion
- All attributes depend on the key, the whole key and nothing but the key.
  - 1NF Keys and no repeating groups
  - 2NF No partial dependencies
  - 3NF No transitive dependencies

1st Normal Form

- Table has a primary key
- Table has no repeating groups

A multivalued attribute is an attribute that may have several values for one record

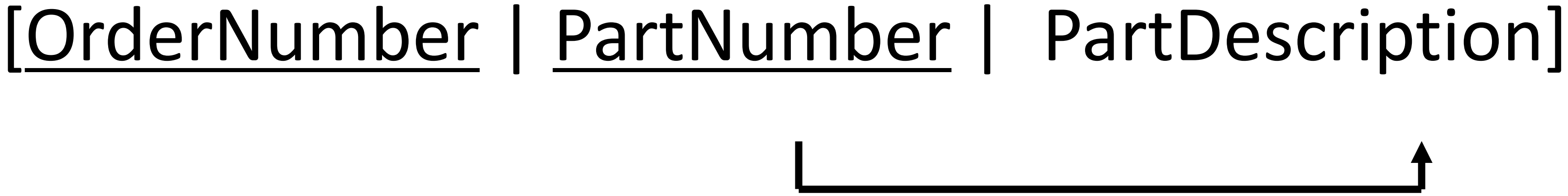
A repeating group is a set of one or more multivalued attributes that are related

2nd Normal Form

- No partial dependencies

  No attribute depends on only some of the attributes of a concatenated key.

Order-Part



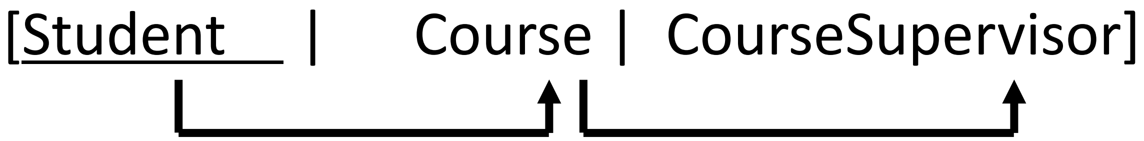
[OrderNumber | PartNumber | PartDescription]

Create a new table with PartNumber key.

3rd Normal Form

- 3rd Normal Form: no transitive dependencies

Transitive dependency means that a <u>non-key</u> attribute depends on another <u>non-key</u> attribute(s).



[<u>Student</u> | Course | CourseSupervisor]

Let’s suppose only one course per student.

This definition says nothing about dependencies that involve the key.

Create a new “Course” table with “Course” and “supervisor”