Introduction to Data Warehousing and Data Mining

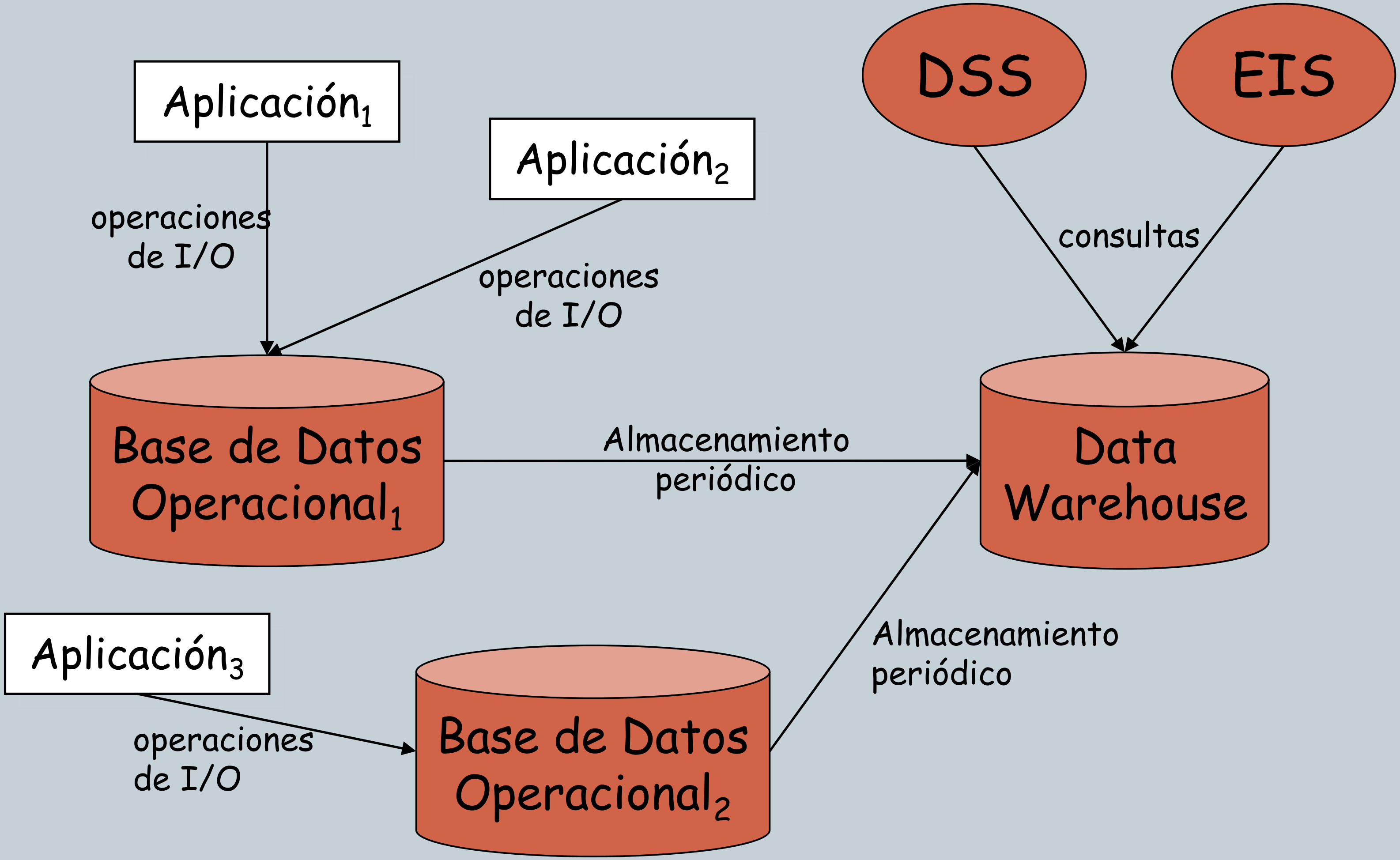
DR. MIGUEL ÁNGEL OROS HERNÁNDEZ

Agenda

SQL Avanzado

- Data Warehouse Concepts
- Data warehouse modeling
- Data cubes construction
- Data warehouse functionalities
- Introduction to Data Mining

Data Warehouse Concepts



Data Warehouse Concepts

Definición

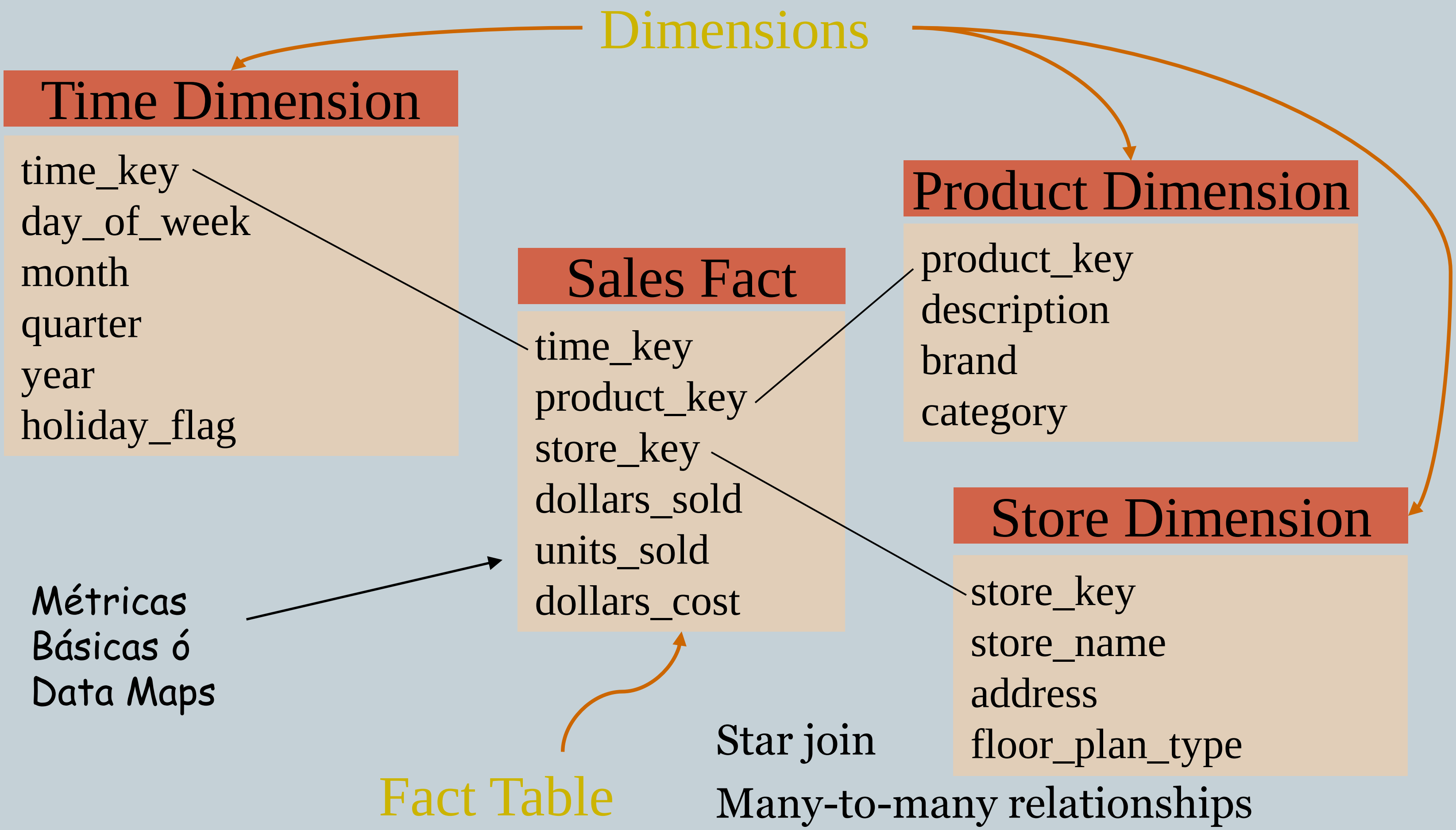
Colección de datos para apoyo a la toma de decisiones

Características

- Orientada hacia la información relevante
- Integrada
- No volátil
- Variable en el tiempo

Data Warehouse Concepts

Highly recognizable to the end user



Dimensions

Time Dimension

time_key
day_of_week
month
quarter
year
holiday_flag

Sales Fact

time_key
product_key
store_key
dollars_sold
units_sold
dollars_cost

Product Dimension

product_key
description
brand
category

Store Dimension

store_key
store_name
address
floor_plan_type

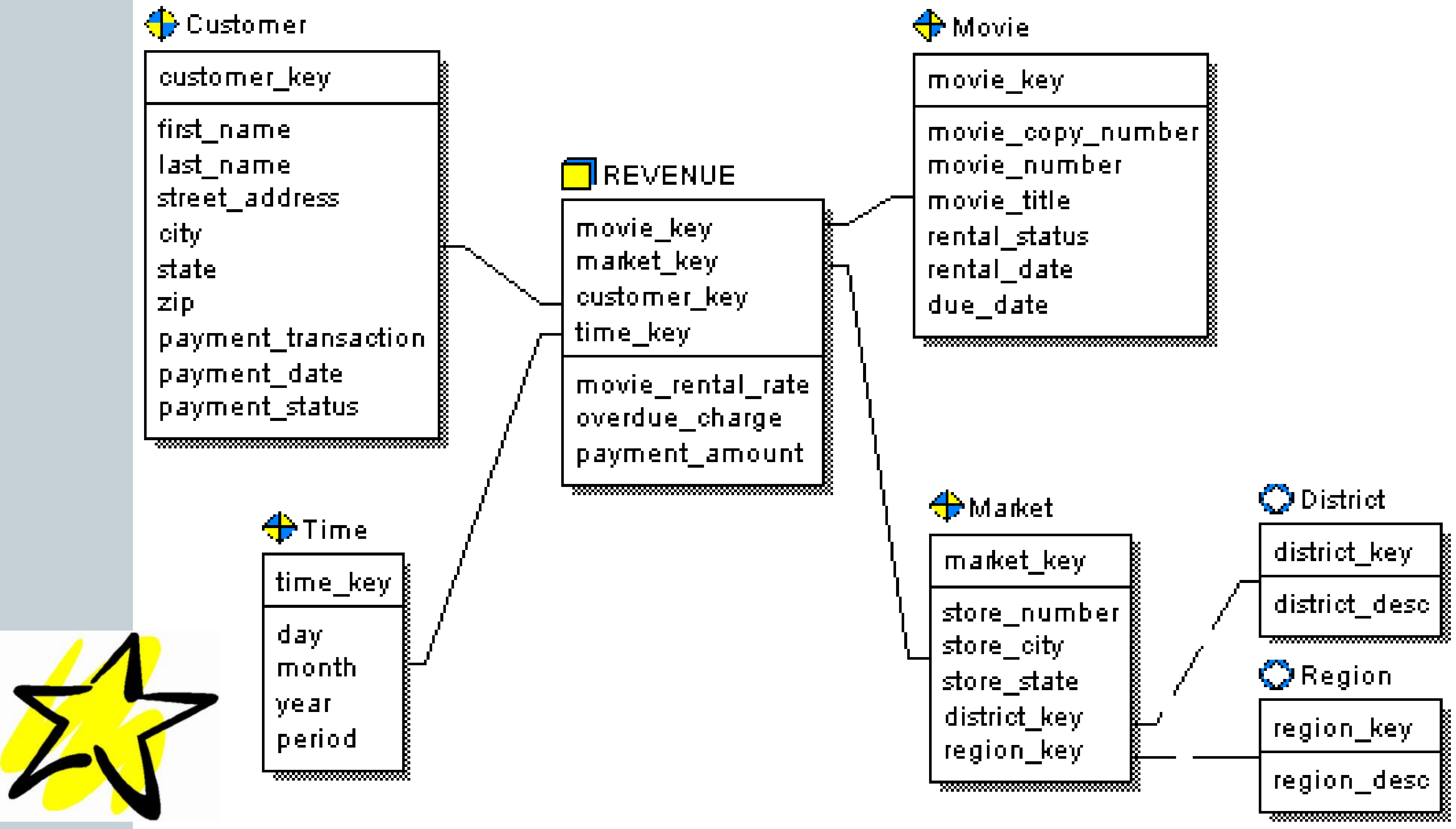
Métricas
Básicas ó
Data Maps

Fact Table

Star join

Many-to-many relationships

Data Warehouse Concepts



Customer
- customer_key
- first_name
- last_name
- street_address
- city
- state
- zip
- payment_transaction
- payment_date
- payment_status

REVENUE
- movie_key
- market_key
- customer_key
- time_key
- movie_rental_rate
- overdue_charge
- payment_amount

Movie
- movie_key
- movie_copy_number
- movie_number
- movie_title
- rental_status
- rental_date
- due_date

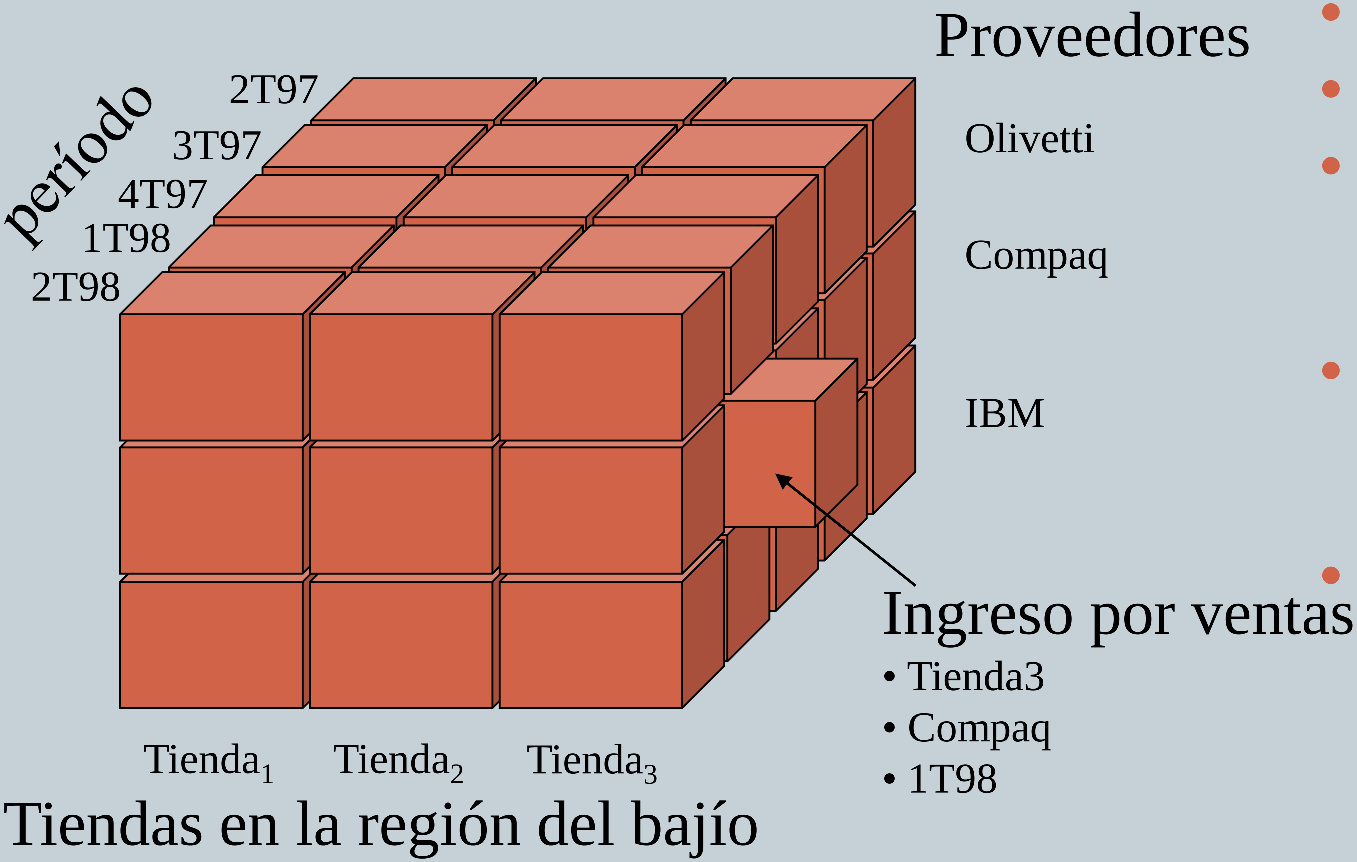
Time
- time_key
- day
- month
- year
- period

Market
- market_key
- store_number
- store_city
- store_state
- district_key
- region_key

District
- district_key
- district_desc

Region
- region_key
- region_desc

Data Warehouse Concepts



Proveedores

período

2T97

3T97

4T97

1T98

2T98

Olivetti

Compaq

IBM

Ingreso por ventas

- Tienda3
- Compaq
- 1T98

Tienda$_1$ Tienda$_2$ Tienda$_3$

Tiendas en la región del bajío

- Cubes
- Hypercubes
- Real dimensional models: 4 to 15 dimensions
- Models with only 2 or 3 dimensions are rare
- Models with 20 or more dimensions seem unjustified

Data Warehouse Concepts
Necesidades de los usuarios finales

- Muestra qué es importante
- Pregunta
  - ¿Por qué?
  - Resumen
  - Otros datos
- Desempeño

Data Warehouse Concepts
Principales conceptos

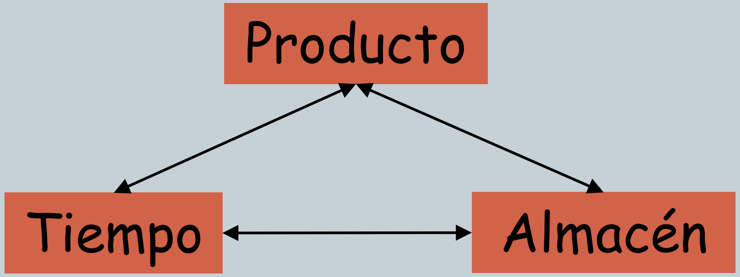
Dimensiones

Tiempo

Almacén

Producto

Interrelaciones

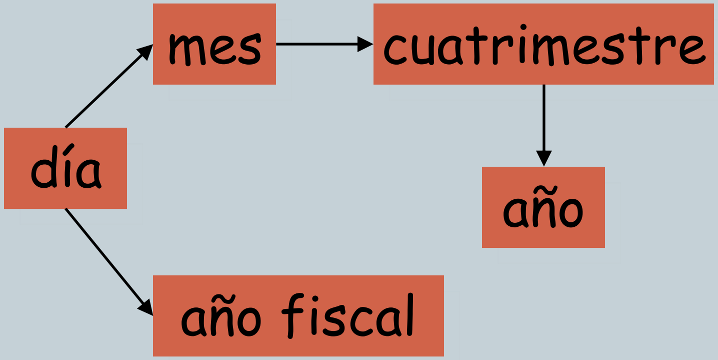


Producto

Tiempo

Almacén

Jerarquías



día

mes

cuatrimestre

año

año fiscal

Servicios

Resumen

Detalle

Agregación

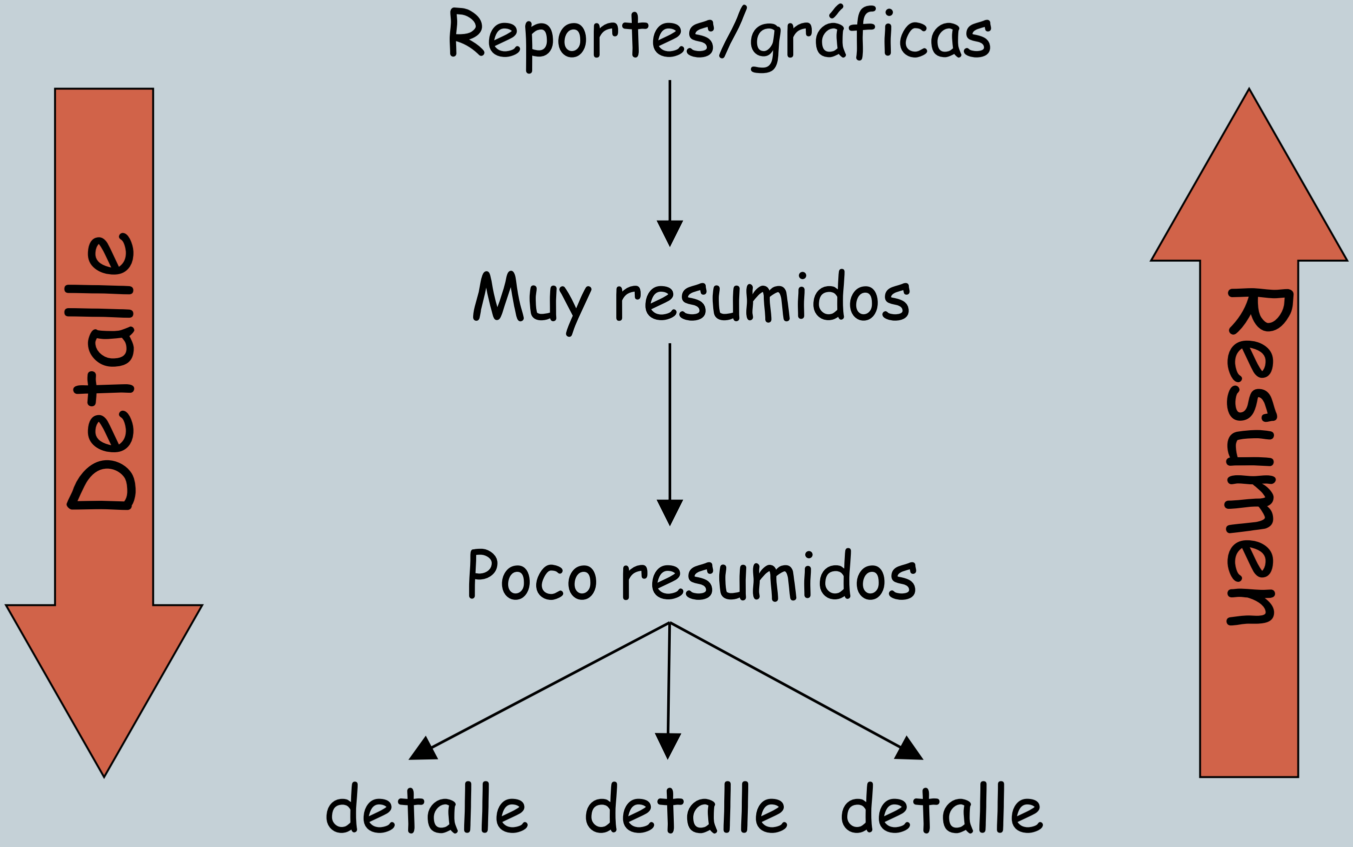
Perspectivas

Consolidación

Cálculo

Data Warehouse Concepts

Detalle y resumen



Reportes/gráficas

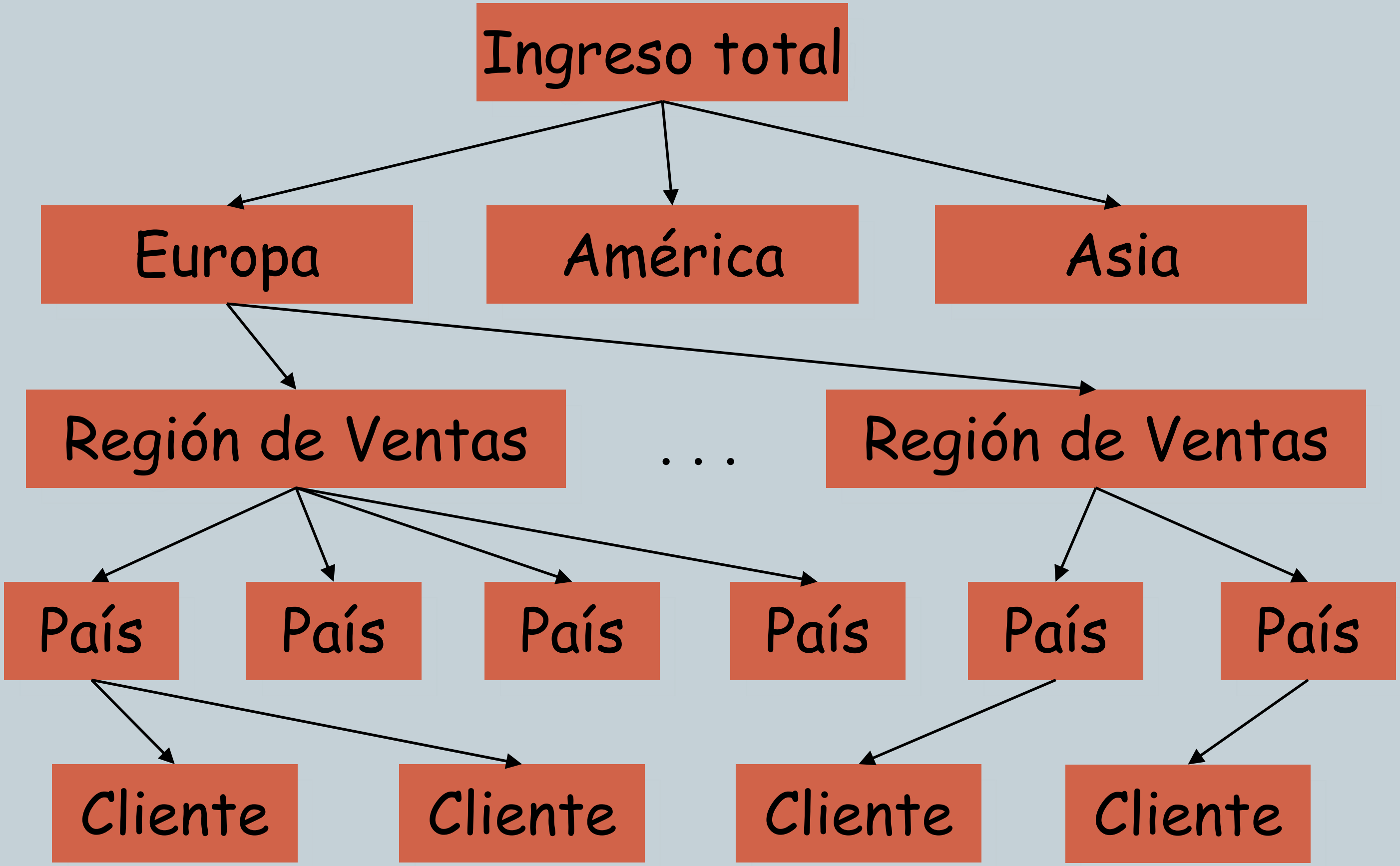
Muy resumidos

Poco resumidos

detalle detalle detalle

Data Warehouse Concepts

Ejemplo de detalle y resumen



Ingreso total

Europa

América

Asia

Región de Ventas

. . .

Región de Ventas

País

País

País

País

País

País

Cliente

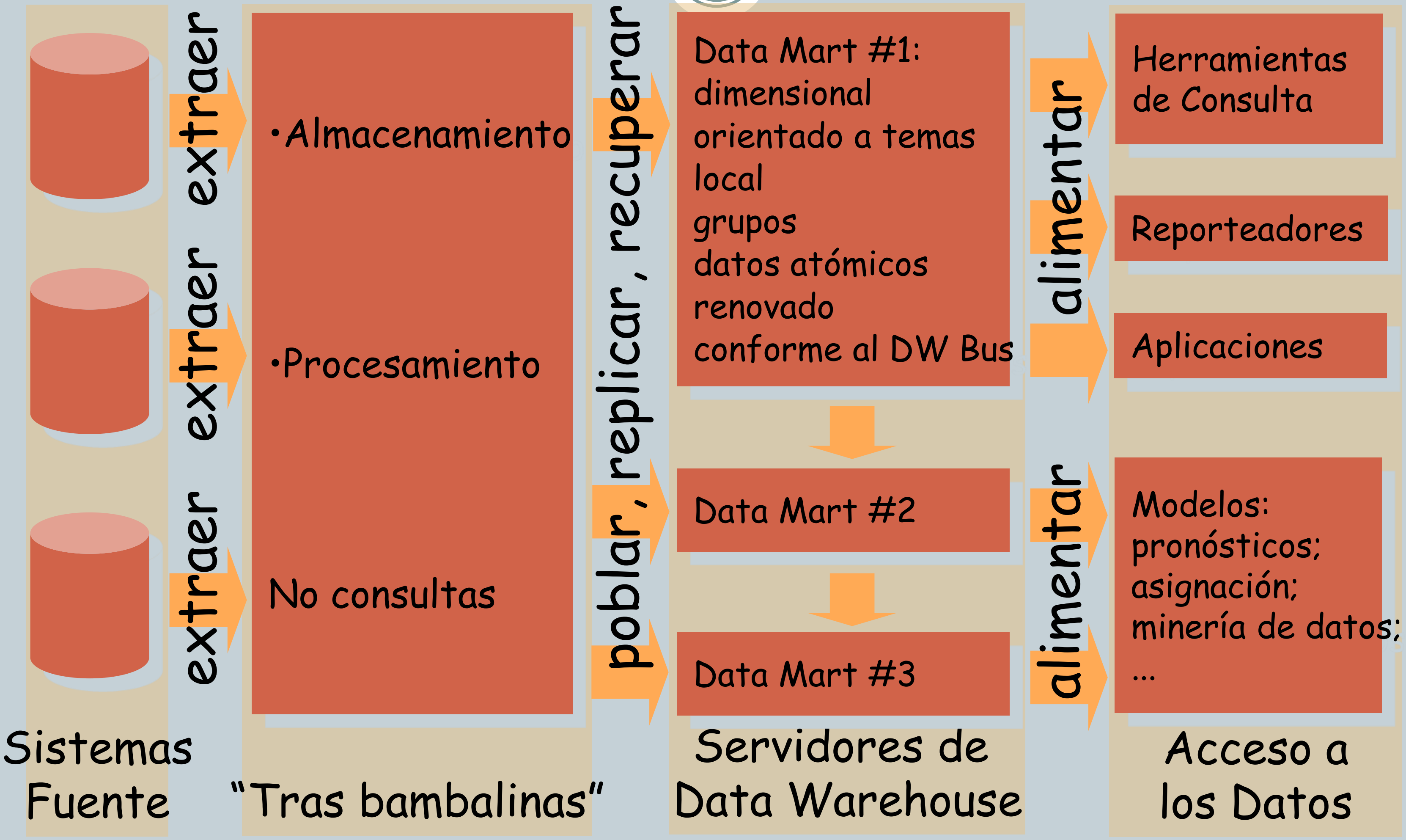
Cliente

Cliente

Cliente

Data Warehouse Concepts

Elementos básicos de la arquitectura



extraer

extraer

extraer

Sistemas Fuente

- Almacenamiento
- Procesamiento

No consultas

"Tras bambalinas"

poblar, replicar, recuperar

Data Mart #1:
dimensional
orientado a temas
local
grupos
datos atómicos
renovado
conforme al DW Bus

Data Mart #2

Data Mart #3

Servidores de Data Warehouse

alimentar

alimentar

Herramientas de Consulta

Reporteadores

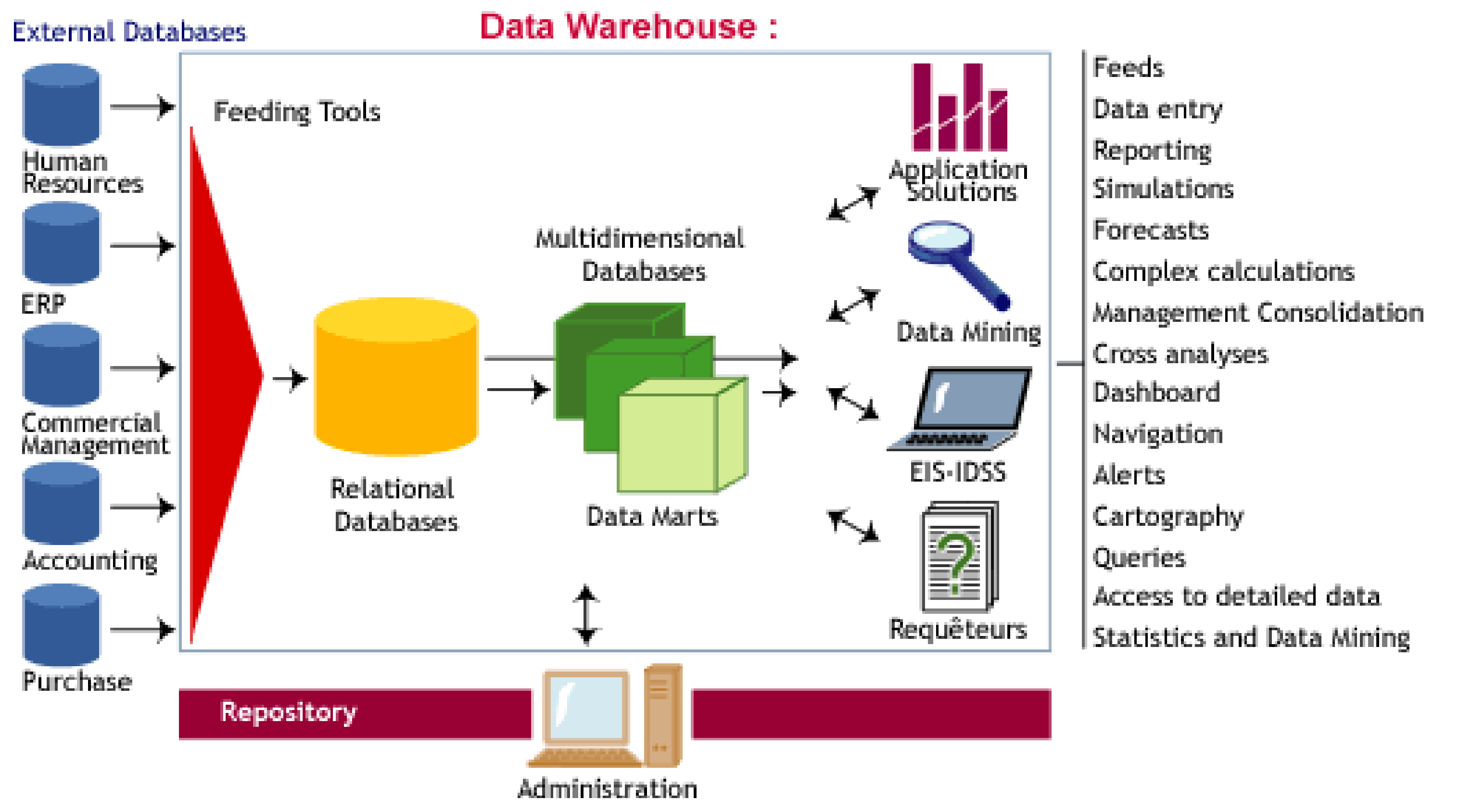
Aplicaciones

Modelos:
pronósticos;
asignación;
minería de datos;
…

Acceso a los Datos

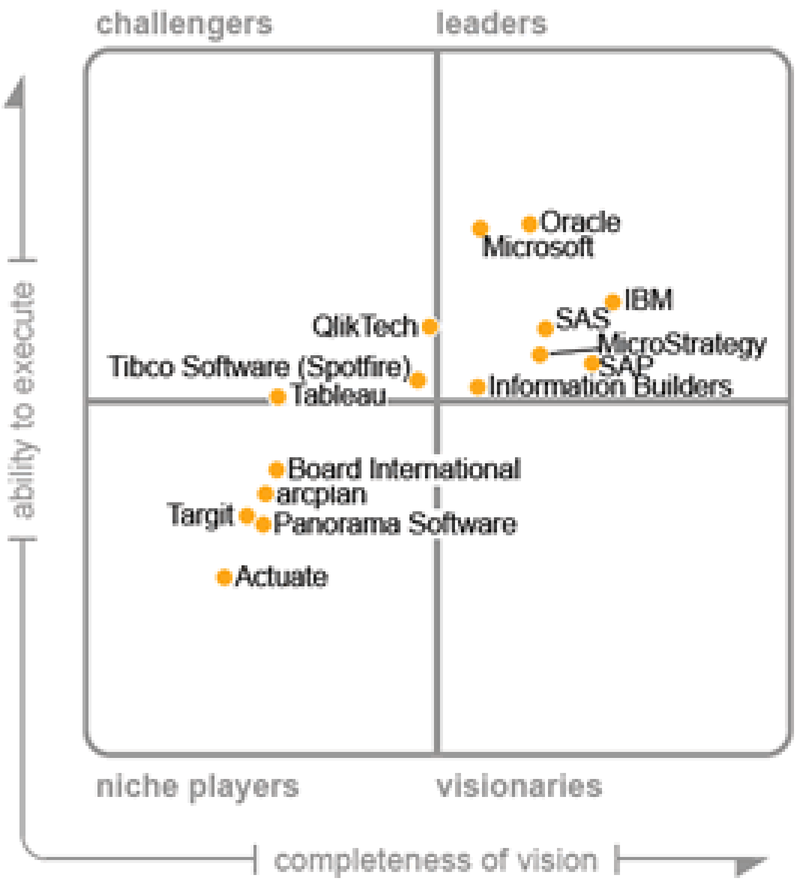
Data Warehouse Concepts

Arquitectura



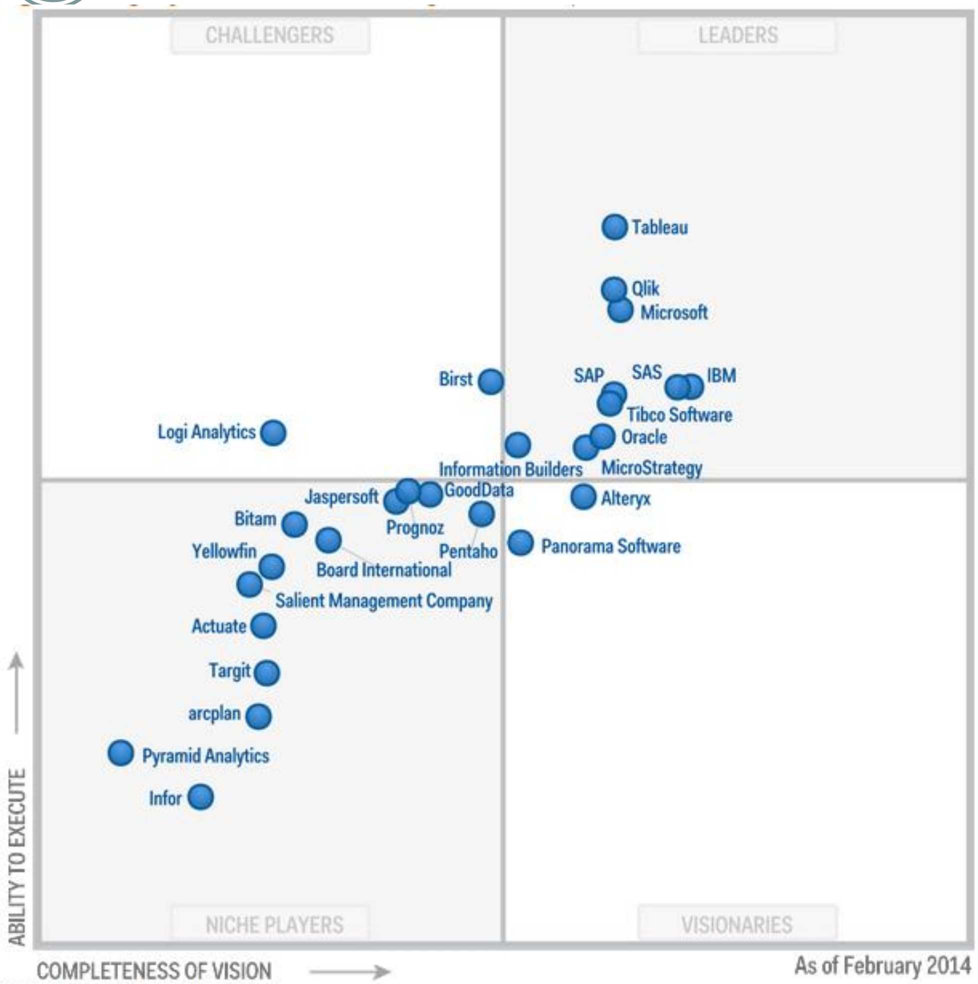
Productos comerciales

Cuadrante mágico de Gartner



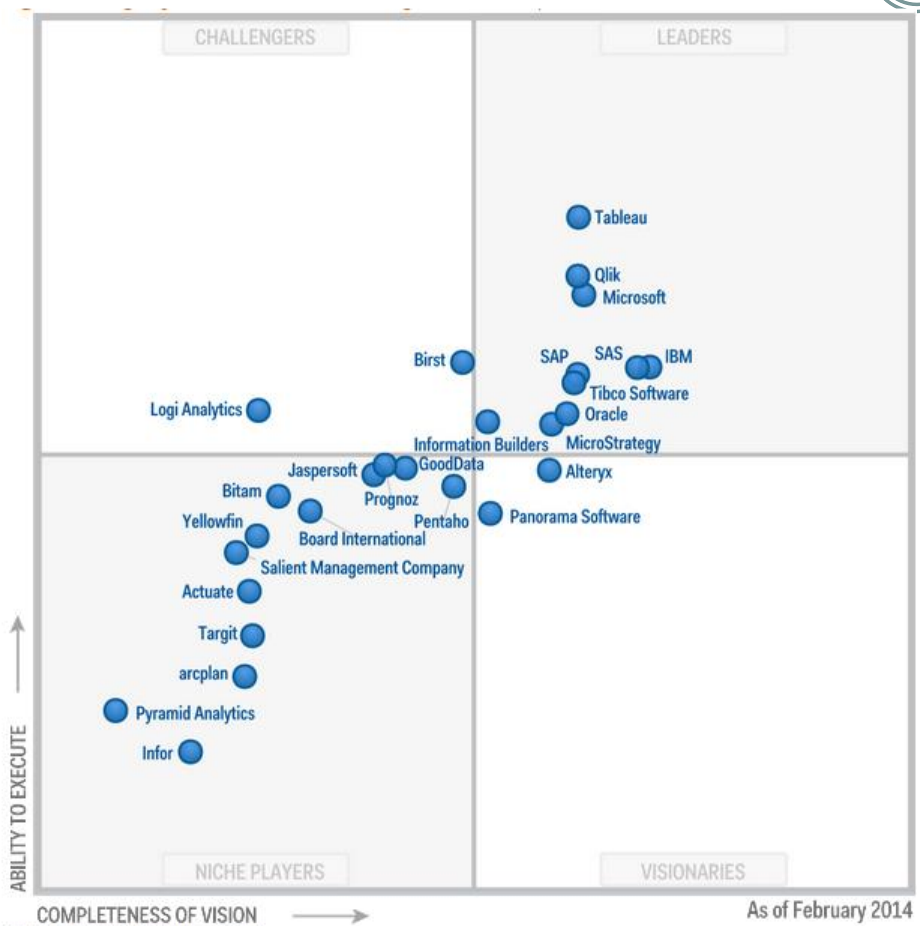
Source: Gartner (January 2010)

As of January 2010

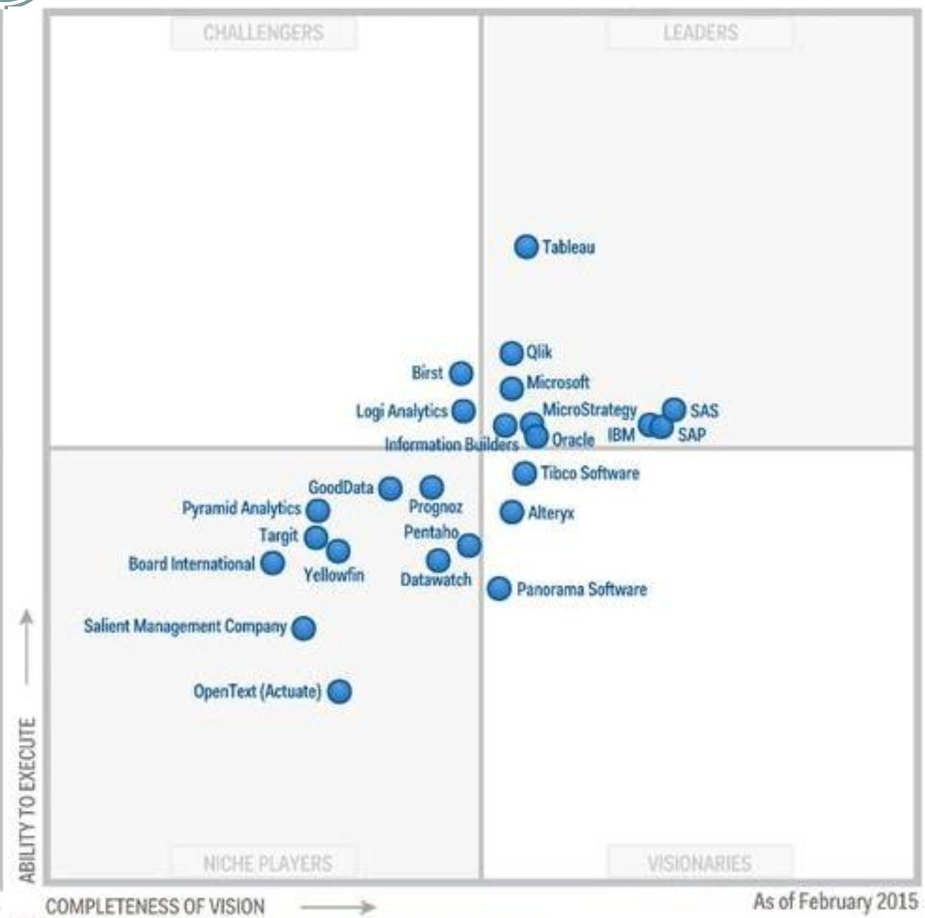


Source: Gartner (February 2014)

Productos comerciales
Cuadrante mágico de Gartner

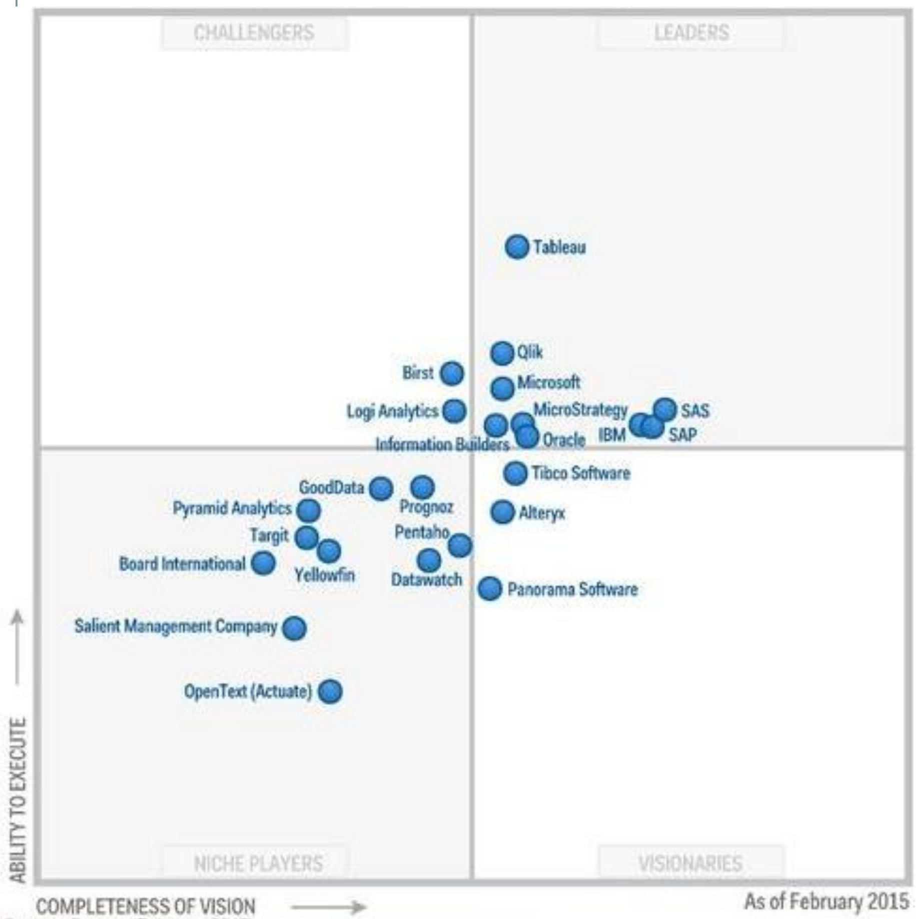


Source: Gartner (February 2014)

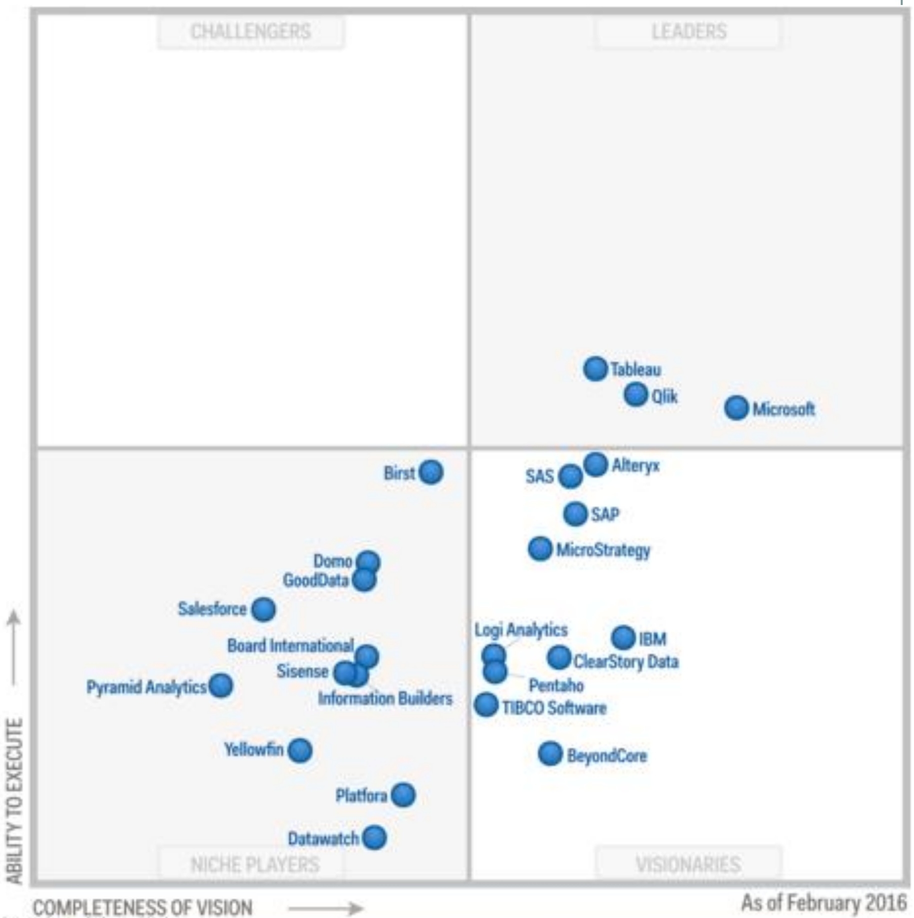


Source: Gartner (February 2015)

Productos comerciales
Cuadrante mágico de Gartner



Source: Gartner (February 2015)



Source: Gartner (February 2016)

Data Warehouse Modeling

fact table

- Numerical measurements
- Numeric, continously value and additive
- Facts
  - Additive
  - Semiadditive
  - Nonadditive
- Most fact tables are extremely sparse

Data Warehouse Modeling

Dimension tables

- Textual descriptions
- Many attributes
- Best attributes: textual, discrete and used as the source of constraints
- Short description (10 to 15 characters)
- Long description (30 to 60 characters)

Data Warehouse Modeling

Slowly Changing Dimensions (SCD)

to refer to the occasional and sporadic changes that occur to dimensional entities like product and customer

0. ignore the change
1. overwrite the changed attribute
2. add a new dimension record with a generalized key
3. add an "old valued" field

Data Warehouse Modeling

Rapidly Changing Small Dimensions

- What if changes are fast?
- Must I use a different design technique?
- Type 2 SCD

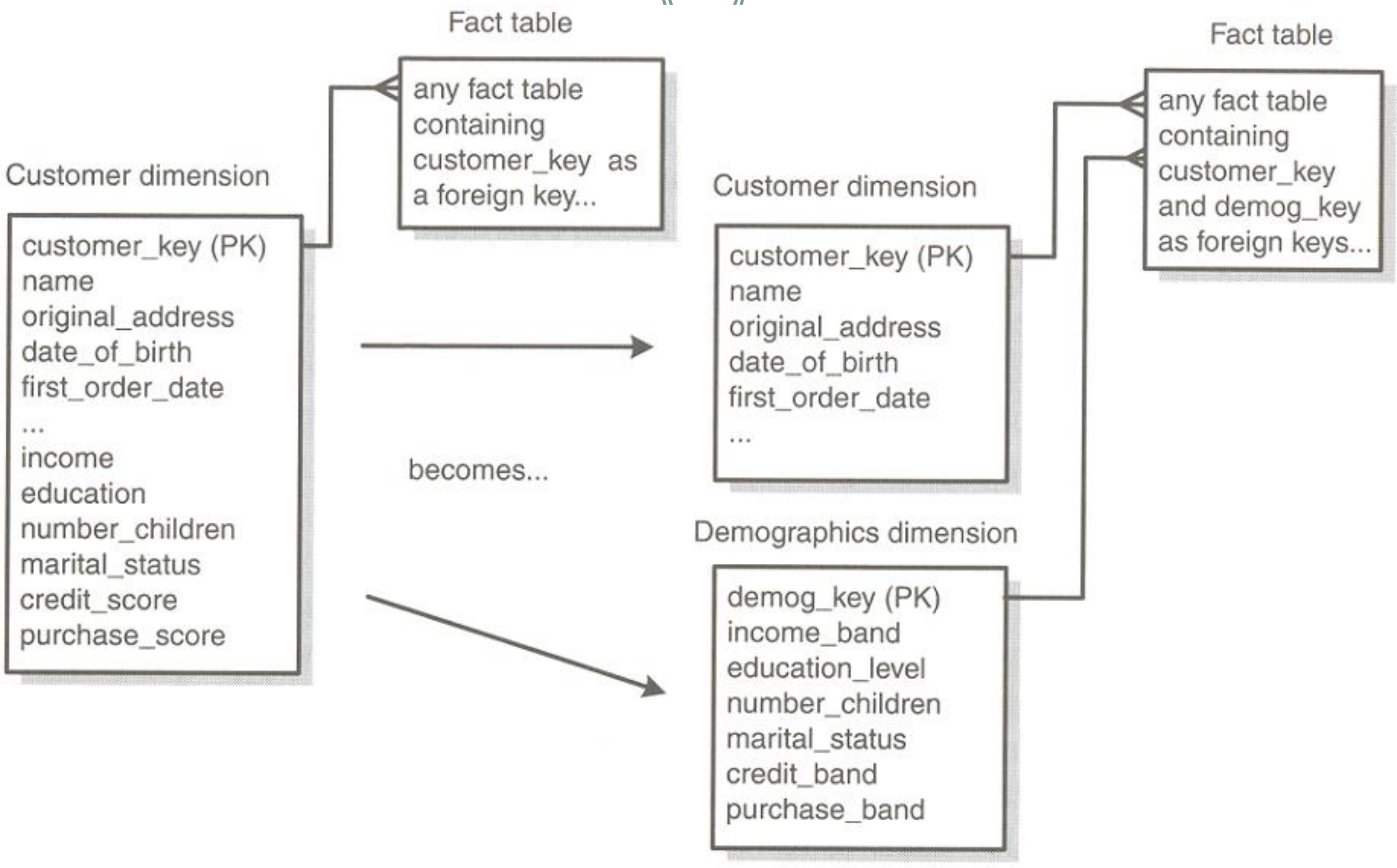
Data Warehouse Modeling

Large Dimensions

- Adopt a conservative design to keep these dimensions under control
- Do not create additional records to handle the SCD problem

Data Warehouse Modeling

The Worst Case: Rapidly Changing Monster Dimensions



Fact table

any fact table containing customer_key as a foreign key...

Customer dimension

customer_key (PK)
name
original_address
date_of_birth
first_order_date
...
income
education
number_children
marital_status
credit_score
purchase_score

becomes...

Customer dimension

customer_key (PK)
name
original_address
date_of_birth
first_order_date
...

Demographics dimension

demog_key (PK)
income_band
education_level
number_children
marital_status
credit_band
purchase_band

Fact table

any fact table containing customer_key and demog_key as foreign keys...

Data Warehouse Modeling

Kimball Methodology: Grocery store item movement

- 500 large grocery stores spread over a three-state area
- each of the stores is a typical modern supermarket with a full complement of departments including grocery, frozed foods, meat, bakery, floral, hard goods, liquor, drugs, …
- each store has roughly 60,000 individual products, *stock keeping units* (SKU)
- temporary price reductions (TPRs)

Metodología de Kimball

Steps in the design process

1. choose a *business process* to model
   - examples: orders, invoices, shipments, inventory
2. choose the *grain* of the business process
   - the grain is the fundamental, atomic level of data to be represented in the fact table for this process
   - examples: individual transactions, individual daily snapshots
3. choose the *dimensions* that will apply to each fact table record
   - examples: time, item, customer, supplier, warehouse, transaction type, and status
4. choose the *measures* that will populate each fact table record
   - typical measures are numeric additive quantities like *dollars_sold* and *units_sold*

Data Warehouse Modeling

Kimball Methodology: Steps in the design process

- choose a business process to model
  - build a daily item movement database
- choose the grain of the business process
  - the grain determines the dimensionality of the database and has a profound impact on the size of the database
  - grain: *SKU by store by promotion by day*
- choose the *dimensions* that will apply to each fact table record

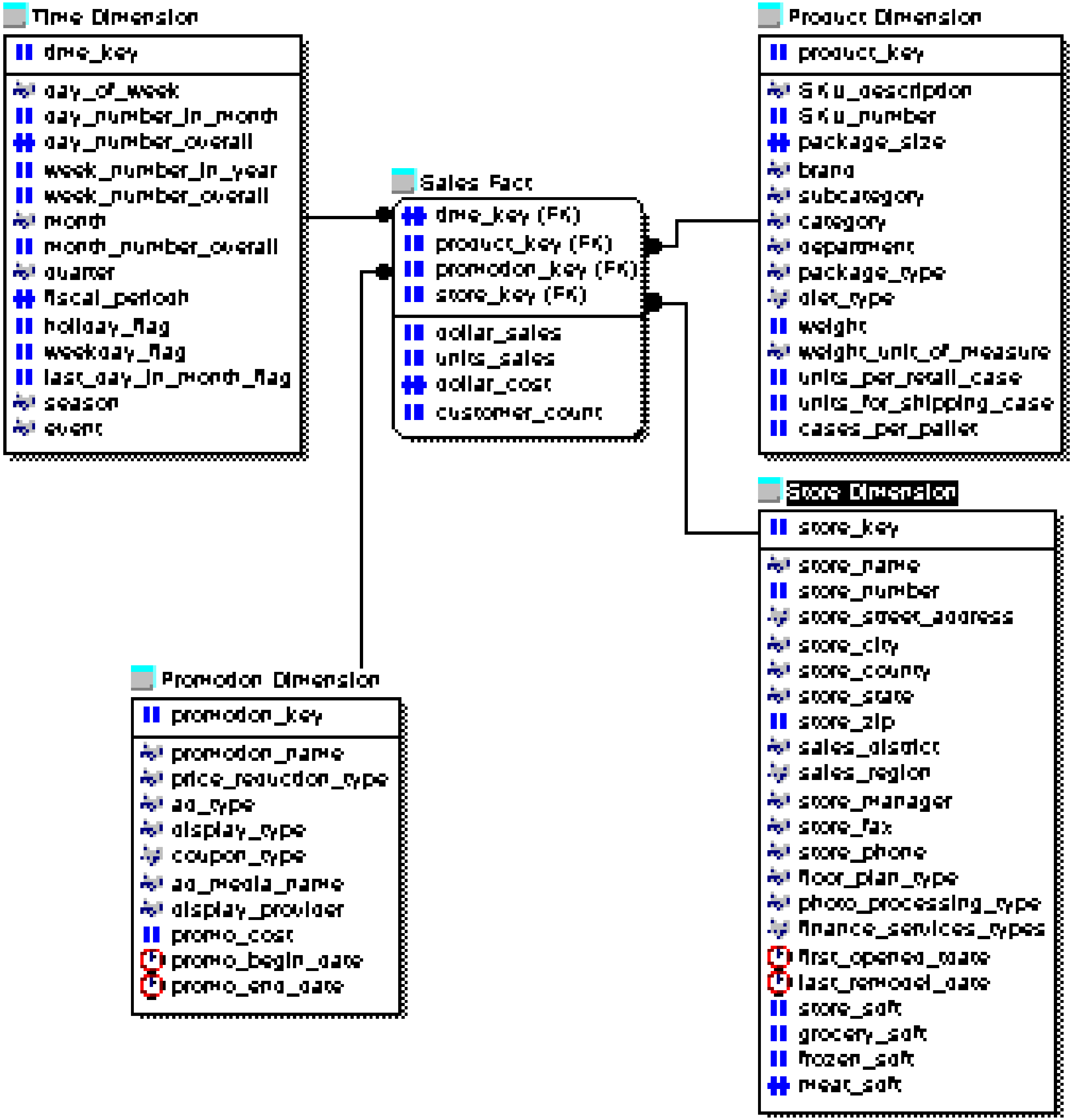
Data Warehouse Modeling

Kimball Methodology: Steps in the design process: the fact table

- dollar sales
- units sales
- dollar cost
- customer count

Data Warehouse Modeling

Kimball Methodology: the grocery store schema



Time Dimension

time_key

day_of_week
day_number_in_month
day_number_overall
week_number_in_year
week_number_overall
month
month_number_overall
quarter
fiscal_period
holiday_flag
weekday_flag
last_day_in_month_flag
season
event

Sales Fact

time_key (FK)
product_key (FK)
promotion_key (FK)
store_key (FK)

dollar_sales
units_sales
dollar_cost
customer_count

Product Dimension

product_key

SKU_description
SKU_number
package_size
brand
subcategory
category
department
package_type
diet_type
weight
weight_unit_of_measure
units_per_retail_case
units_for_shipping_case
cases_per_pallet

Store Dimension

store_key

store_name
store_number
store_street_address
store_city
store_county
store_state
store_zip
sales_district
sales_region
store_manager
store_fax
store_phone
floor_plan_type
photo_processing_type
finance_services_types
first_opened_date
last_remodel_date
store_sqft
grocery_sqft
frozen_sqft
meat_sqft

Promotion Dimension

promotion_key

promotion_name
price_reduction_type
ad_type
display_type
coupon_type
ad_media_name
display_provider
promo_cost
promo_begin_date
promo_end_date

Data Warehouse Modeling

Kimball Methodology: the promotion dimension

- causal dimension
- factors
  - lift

    whether the product under promotion experienced a gain in sales during the promotional period
  - time shifting

    whether the products under promotion showed a drop in sales after the promotion, thereby canceling the gain during the promotion
  - cannibalization

    whether the products under promotion showed a gain in sales but other products nearby on the shelf showed a corresponding decrease in sales
  - growing the market

    whether all the products in the promoted category of products experienced a net overall gain in sales taking into account the time periods before, during and after the promotion
  - profit

    whether the promotion was profitable

Data Warehouse Modeling

Kimball Methodology: The grocery store facts

- quantity sold
- dollar revenue
- dollar costs
- customer count
  - is not additive accross the product dimension
  - semiadditive
- gross profit = dollar revenue - dollar cost
- gross margin = gross profit/dollar revenue
  = 1-dollar cost/dollar revenue

Data Warehouse Modeling

Kimball Methodology: Database sizing for the grocery chain

- time dimension: 2 years × 365 days = 730 days
- store dimension: 300 stores, reporting sales each day
- product dimension: 30,000 products in each store, of which 3,000 sell each day in a given store
- promotion dimension: a sold item appears in only one promotion condition in a store on a day
- number of base fact records

  730 × 300 × 3000 × 1 = 657 million records
- number of key fields: 4; number of fact fields = 4
- base fact table size: 657 million × 8 × 4 = 21 Gb

Data Warehouse Modeling

Fact/Qualifier Modeling: Business questions

- stakeholder driven
- goal oriented
- business process oriented
- business measures based
- data source analysis
- current reporting analysis
- surrogate system analysis
- subject analysis

business questions

Data Warehouse Modeling

Fact/Qualifier Modeling: Facts, qualifiers, associations

- facts
  - discrete items of business information that (partially) satisfy the information needs of the business.
  - these are typed as descriptive or metric
- qualifiers
  - criteria, by which the facts are accessed, sorted, grouped, aggregated, filtered and presented to warehouse users
- the fact/qualifier association
  - an entry at an intersecting cell indicating that the qualifier may be used to control how the fact is used in analysis
  - association entries may record data about the association

Data Warehouse Modeling

Fact/Qualifier Modeling: The modeling process

- the matrix combines two lists derived from the information needs and their related business questions
- list of facts ≡ know list

  answers the question "what do you need to know"
- list of qualifiers ≡ by list

  answers the question "what do you want to know it by"

Data Warehouse Modeling

Fact/Qualifier Modeling: The modeling process

- stage one: mapping of business questions

  initial analysis of business questions to identify which parts of the question represent facts and which represent qualifiers

- stage two: fact analysis
  - understand the facts in terms of the way that they are to be used
  - is each fact intended to measure something, to describe something, or to identify something?

- stage three: fact refinement
  - remove redundancy in the fact set
  - combine synonymous facts to be represented as a single fact with only one name
  - remove modifying words from fact names and migrate them to be represented as qualifiers
  - maintain fact/qualifier associations throughout

Data Warehouse Modeling

Fact/Qualifier Modeling: The modeling process

- stage four: qualifier analysis
  - understand how the qualifiers relate to one another
  - are there any hierarchical relationships among the qualifiers?
  - what are the hierarchical levels of the qualifiers?
  - are there any missing levels that need to be added to the qualifier axis?

  this analysis clearly cannot be done without understanding how the qualifiers are used in the business

- stage five: qualifier refinement
  - ensure that each qualifiers is fully understood
  - ensure that associations of facts and qualifiers maintain the integrity of the structure when qualifiers are hierarchical related

Data Warehouse Modeling

Fact/Qualifier Modeling: Résumé

Fact/qualifier Analysis

metric or non-metric facts

what is the subject of each fact?

are they hierarchical?

are any levels missing?

describing qualifiers

single qualifier in multiple dimension

are the relationships amont the qualifiers?

non-hierarchical dimensions

fact and qualifiers refinement

Data Warehouse Modeling

Families of fact tables: chains

- Many businesses have logical flow that has a beginning and an end
- Product
  - Raw material production
  - Ingredient purchasing
  - Ingredient delivery
  - Ingredient inventory
  - Bill of materials
  - Manufacturing process control
  - Manaufacturing costs
  - Packaing
  - trans-shipping to warehouse
  - Finished goods inventory

Data Warehouse Modeling

Families of fact tables: chains

Product as finished good

- Finished goods inventory
- Manufacturing shipments
- Distributor inventory
- Distributor shipments
- Retail inventory
- Retail sales

Insurance companies

- Marketing
- Agent/broker sales
- Rating
- Underwriting
- Reinsuring
- Policy creation
- Claims processing
- Claims investigation
- Claims payments

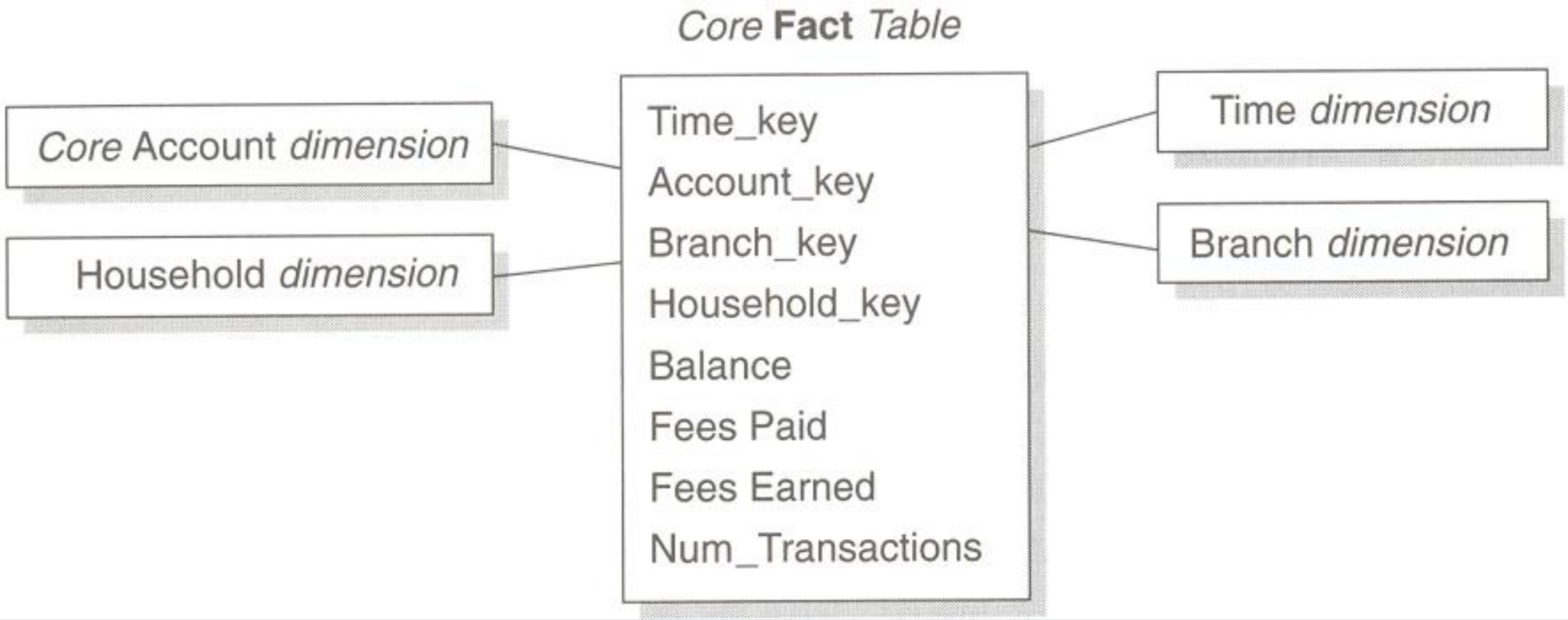
Data Warehouse Modeling

Families of fact tables: chains and circles

- Multiple fact tables are needed to support a business with many process
- Each process spawns one or more fact tables
- When the processes are naturally arranged in order, when often call this a value chain

Data Warehouse Modeling

Families of fact tables: heterogeneous product schemas



Core **Fact** *Table*

Time_key
Account_key
Branch_key
Household_key
Balance
Fees Paid
Fees Earned
Num_Transactions

Core Account *dimension*

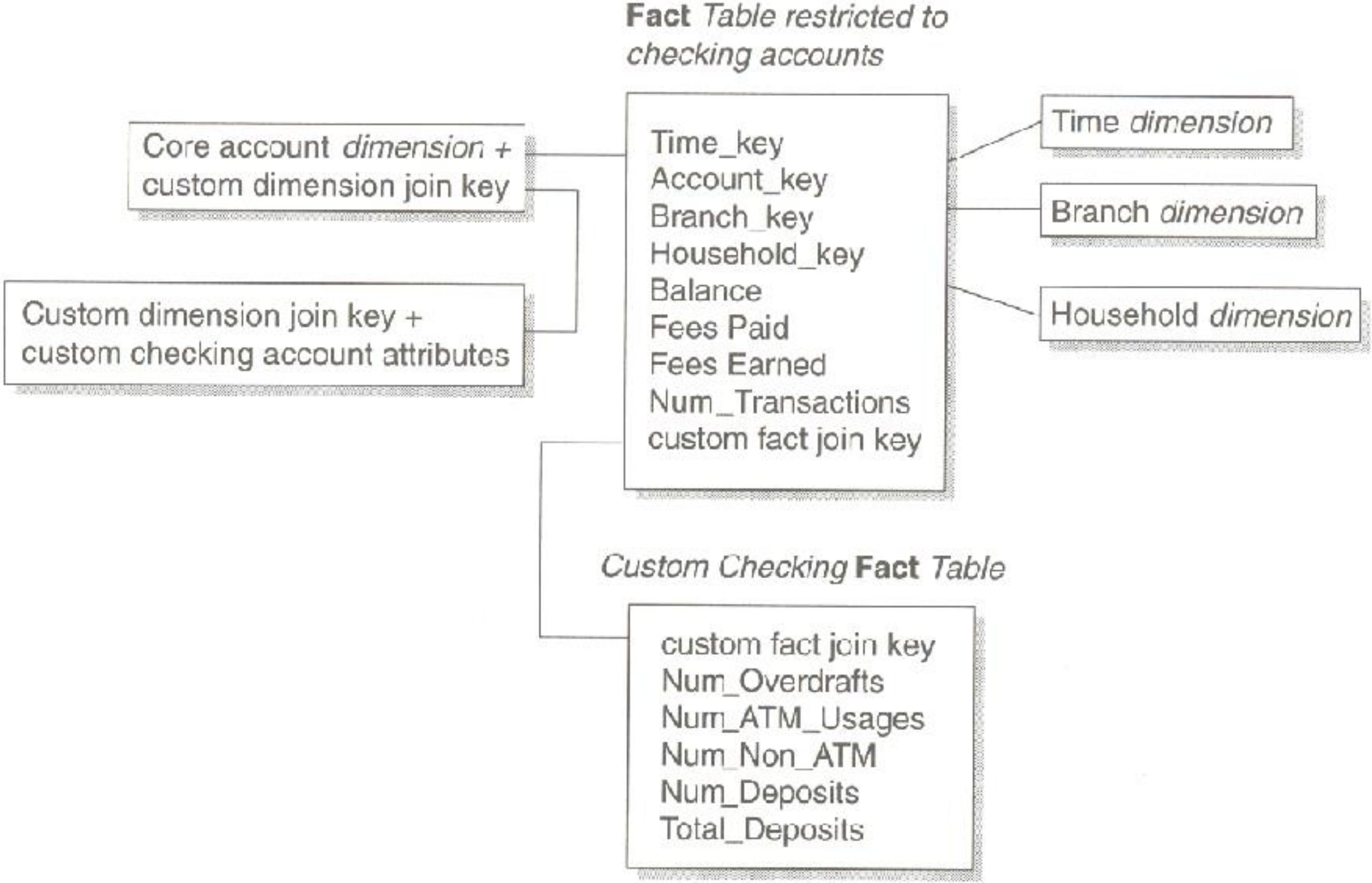
Household *dimension*

Time *dimension*

Branch *dimension*

Data Warehouse Modeling

Families of fact tables: heterogeneous product schemas



Fact *Table restricted to checking accounts*

Time_key
Account_key
Branch_key
Household_key
Balance
Fees Paid
Fees Earned
Num_Transactions
custom fact join key

Core account *dimension* + custom dimension join key

Custom dimension join key + custom checking account attributes

Time *dimension*

Branch *dimension*

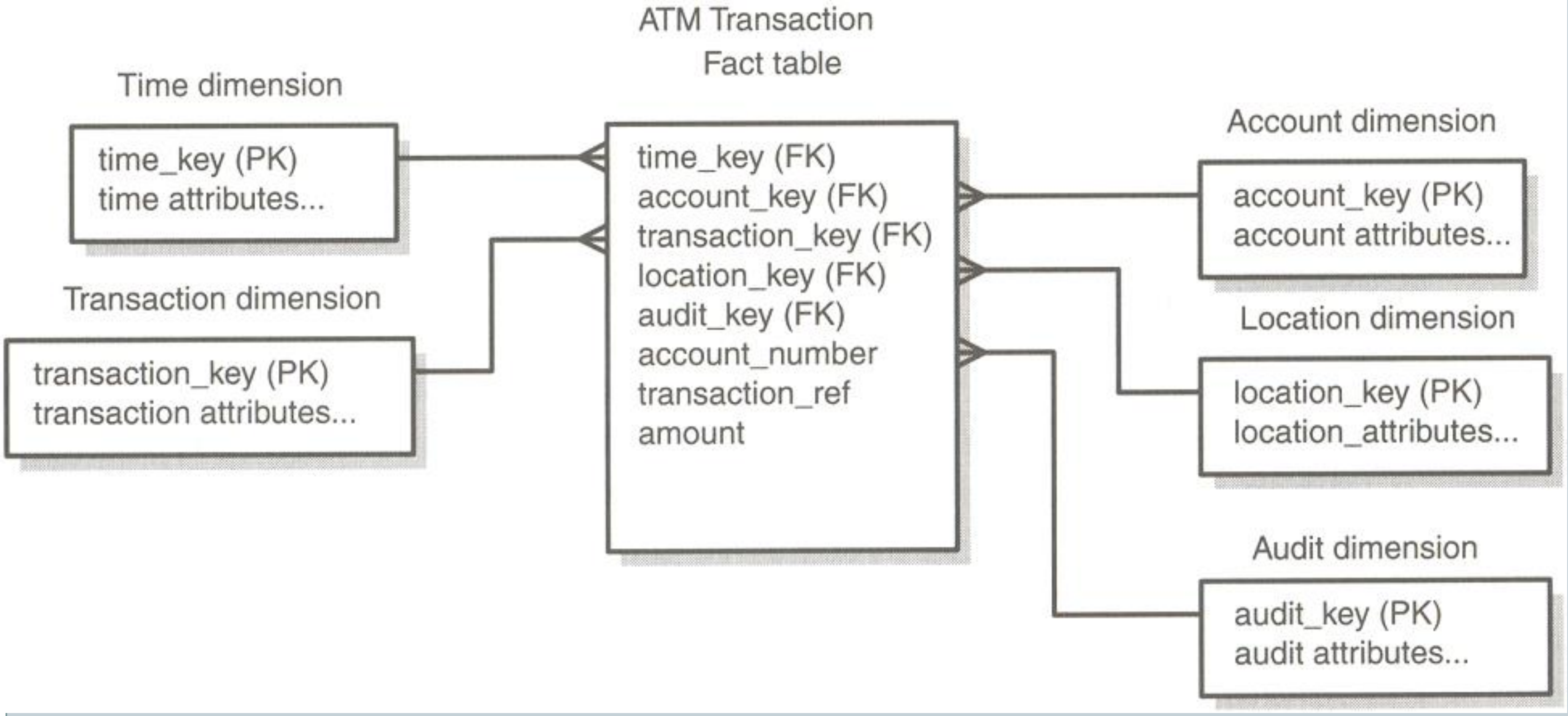
Household *dimension*

Custom Checking **Fact** *Table*

custom fact join key
Num_Overdrafts
Num_ATM_Usages
Num_Non_ATM
Num_Deposits
Total_Deposits

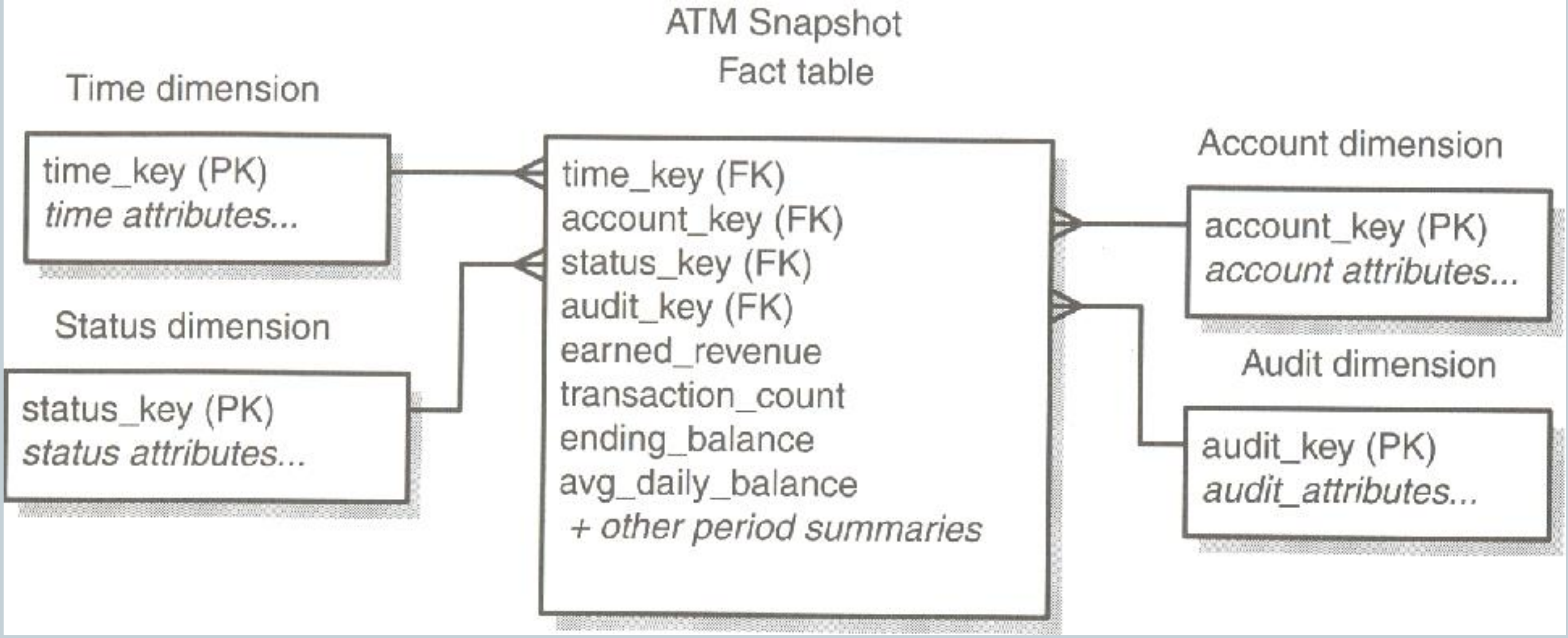
Data Warehouse Modeling

Families of fact tables: the transaction schema



Data Warehouse Modeling

Families of fact tables: the snapshot schema



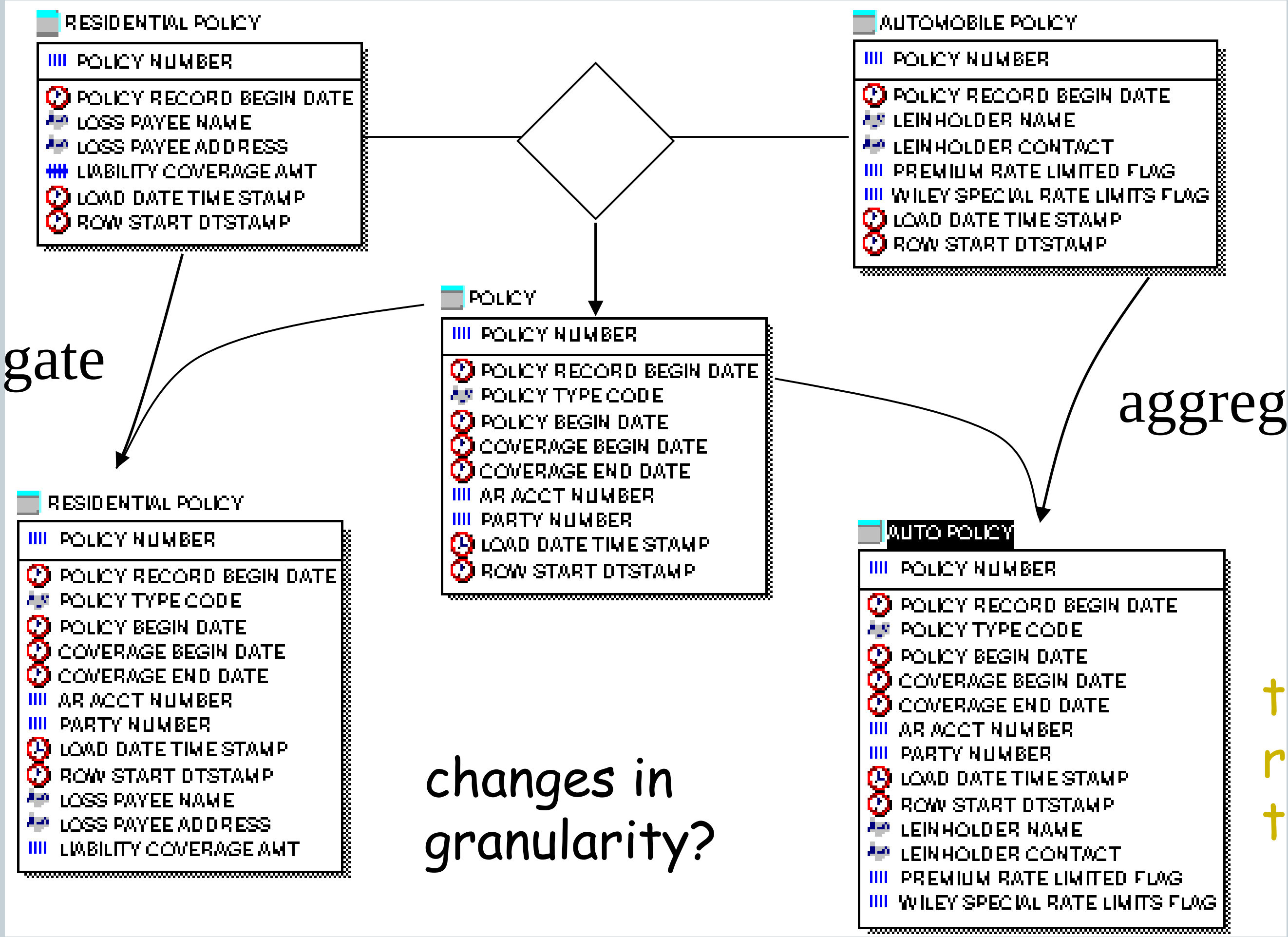
Data Warehouse Modeling

Families of fact tables: aggregates

- Improve query performance
- Stored in separate tables
- Derived from the most granular fact table in each datamart
- Each member of the family represents a particular degree of summarization

Data Warehouse Modeling

Families of fact tables: aggregates



RESIDENTIAL POLICY

- POLICY NUMBER
- POLICY RECORD BEGIN DATE
- LOSS PAYEE NAME
- LOSS PAYEE ADDRESS
- LIABILITY COVERAGE AMT
- LOAD DATE TIME STAMP
- ROW START DTSTAMP

AUTOMOBILE POLICY

- POLICY NUMBER
- POLICY RECORD BEGIN DATE
- LEINHOLDER NAME
- LEINHOLDER CONTACT
- PREMIUM RATE LIMITED FLAG
- WILEY SPECIAL RATE LIMITS FLAG
- LOAD DATE TIME STAMP
- ROW START DTSTAMP

POLICY

- POLICY NUMBER
- POLICY RECORD BEGIN DATE
- POLICY TYPE CODE
- POLICY BEGIN DATE
- COVERAGE BEGIN DATE
- COVERAGE END DATE
- AR ACCT NUMBER
- PARTY NUMBER
- LOAD DATE TIME STAMP
- ROW START DTSTAMP

RESIDENTIAL POLICY

- POLICY NUMBER
- POLICY RECORD BEGIN DATE
- POLICY TYPE CODE
- POLICY BEGIN DATE
- COVERAGE BEGIN DATE
- COVERAGE END DATE
- AR ACCT NUMBER
- PARTY NUMBER
- LOAD DATE TIME STAMP
- ROW START DTSTAMP
- LOSS PAYEE NAME
- LOSS PAYEE ADDRESS
- LIABILITY COVERAGE AMT

AUTO POLICY

- POLICY NUMBER
- POLICY RECORD BEGIN DATE
- POLICY TYPE CODE
- POLICY BEGIN DATE
- COVERAGE BEGIN DATE
- COVERAGE END DATE
- AR ACCT NUMBER
- PARTY NUMBER
- LOAD DATE TIME STAMP
- ROW START DTSTAMP
- LEINHOLDER NAME
- LEINHOLDER CONTACT
- PREMIUM RATE LIMITED FLAG
- WILEY SPECIAL RATE LIMITS FLAG

aggregate

aggregate

changes in granularity?

to facilate ready access to date

Data Warehouse Modeling

Factless Fact Tables

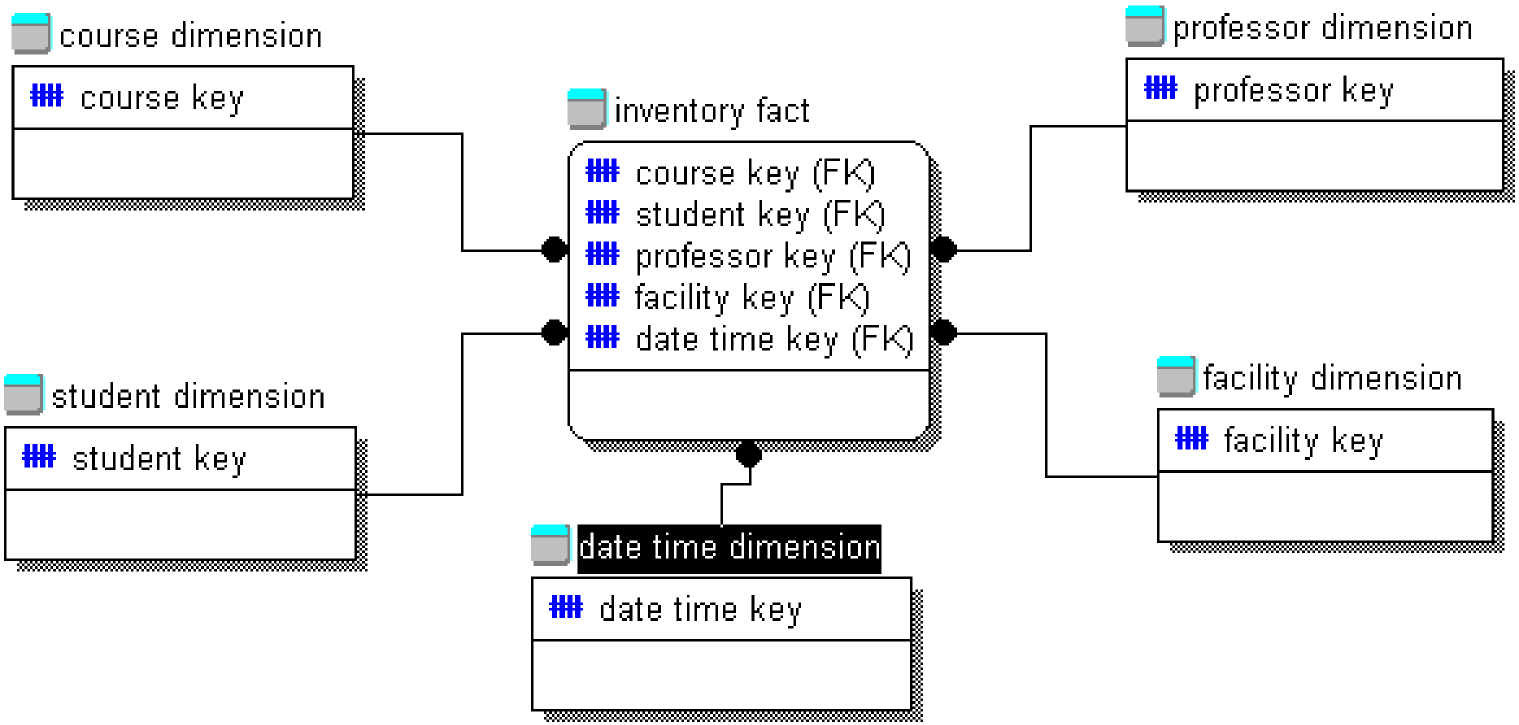
- tables without no measured facts!
- example: modeling daily class attendance at a college with a fact table
- questions
  - which courses were the most heavily attended?
  - which courses suffered the least attrition over time?
  - which facilitites in which departments where used by the most students from other departments?
  - what was the average occupancy rate of the facilities as a function of time of day?
- applications will perform mostly counts

Data Warehouse Modeling

Factless Fact Tables

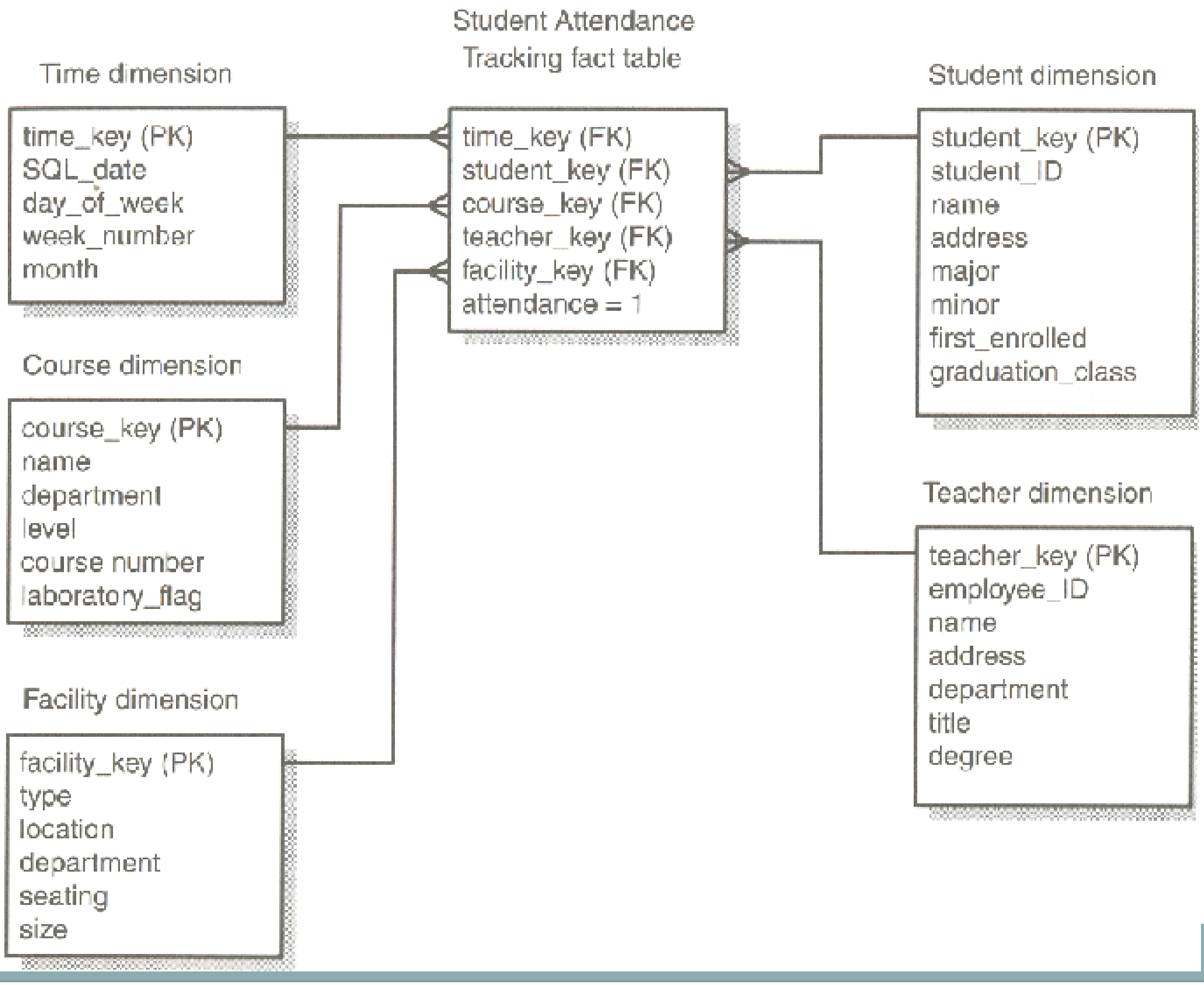
- grain: daily attendance
- SQL

```
SELECT professor, count(date_time_key)
...
GROUP BY professor
```



Data Warehouse Modeling

Factless Fact Tables



Student Attendance
Tracking fact table

time_key (FK)
student_key (FK)
course_key (FK)
teacher_key (FK)
facility_key (FK)
attendance = 1

Time dimension

time_key (PK)
SQL_date
day_of_week
week_number
month

Course dimension

course_key (PK)
name
department
level
course number
laboratory_flag

Facility dimension

facility_key (PK)
type
location
department
seating
size

Student dimension

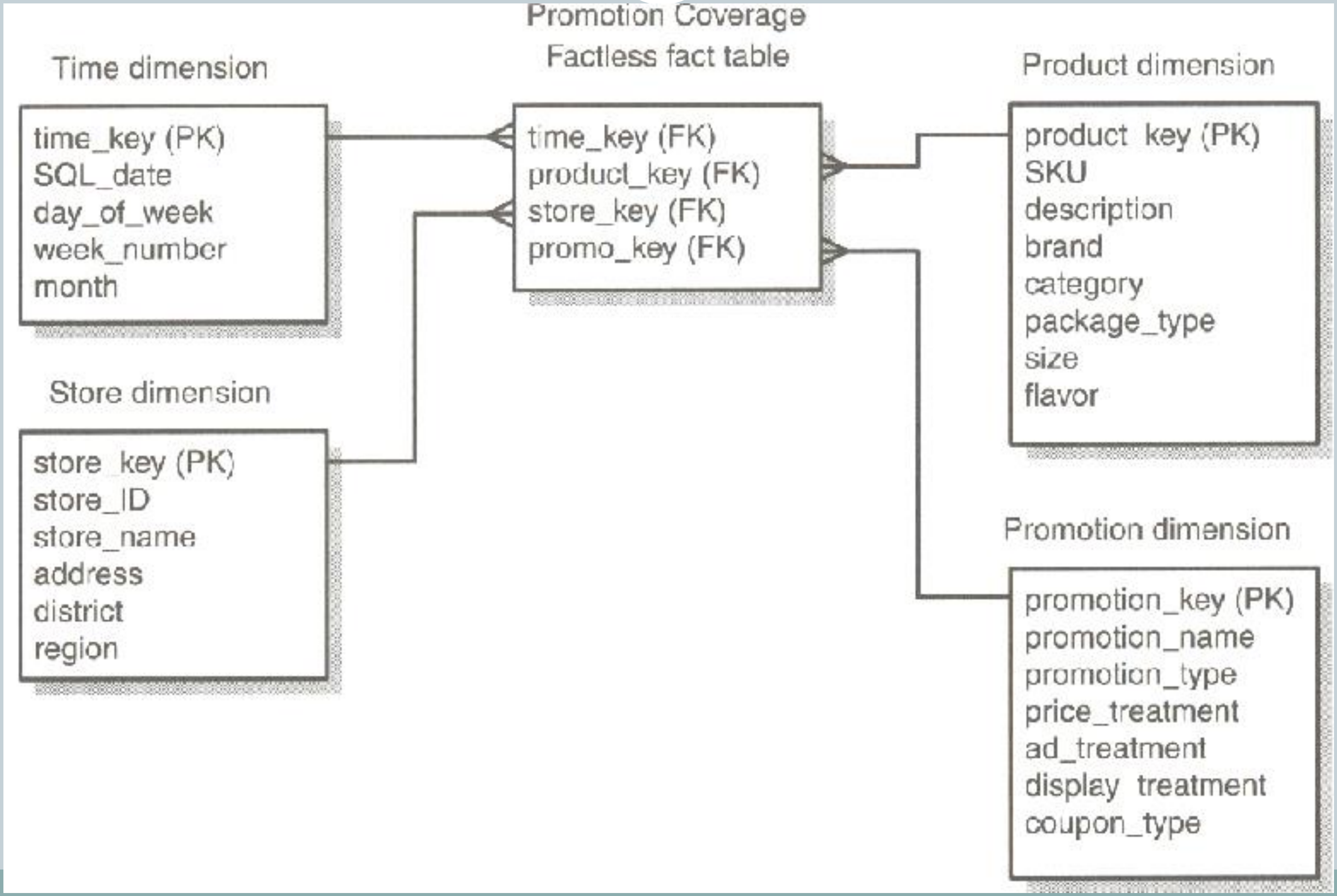
student_key (PK)
student_ID
name
address
major
minor
first_enrolled
graduation_class

Teacher dimension

teacher_key (PK)
employee_ID
name
address
department
title
degree

Data Warehouse Modeling

Factless Fact Tables



Promotion Coverage
Factless fact table

time_key (FK)
product_key (FK)
store_key (FK)
promo_key (FK)

Time dimension

time_key (PK)
SQL_date
day_of_week
week_number
month

Store dimension

store_key (PK)
store_ID
store_name
address
district
region

Product dimension

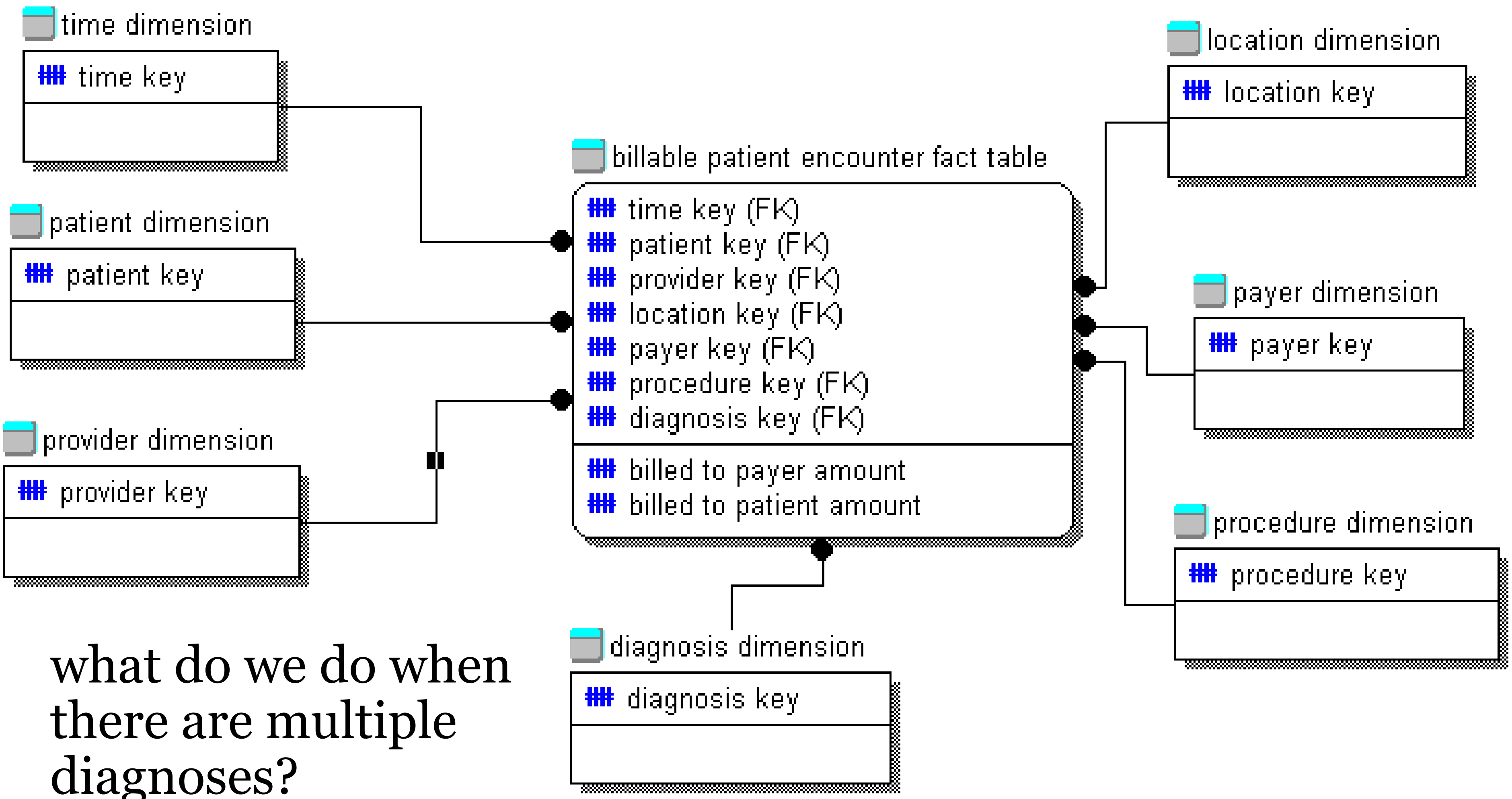
product key (PK)
SKU
description
brand
category
package_type
size
flavor

Promotion dimension

promotion_key (PK)
promotion_name
promotion_type
price_treatment
ad_treatment
display treatment
coupon_type

Data Warehouse Modeling

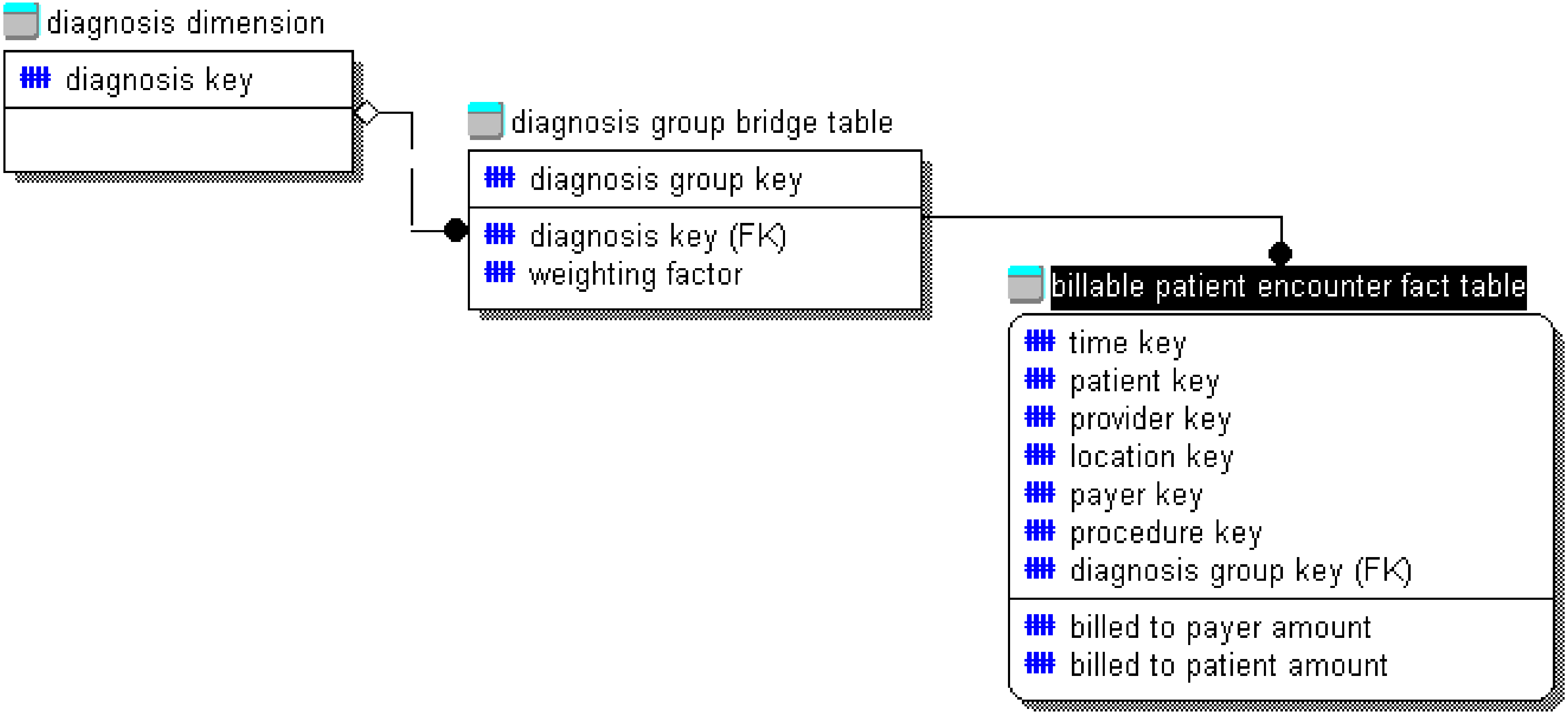
Many to many dimensions



what do we do when there are multiple diagnoses?

Data Warehouse Modeling

Bridge or helper table



diagnosis dimension

- diagnosis key

diagnosis group bridge table

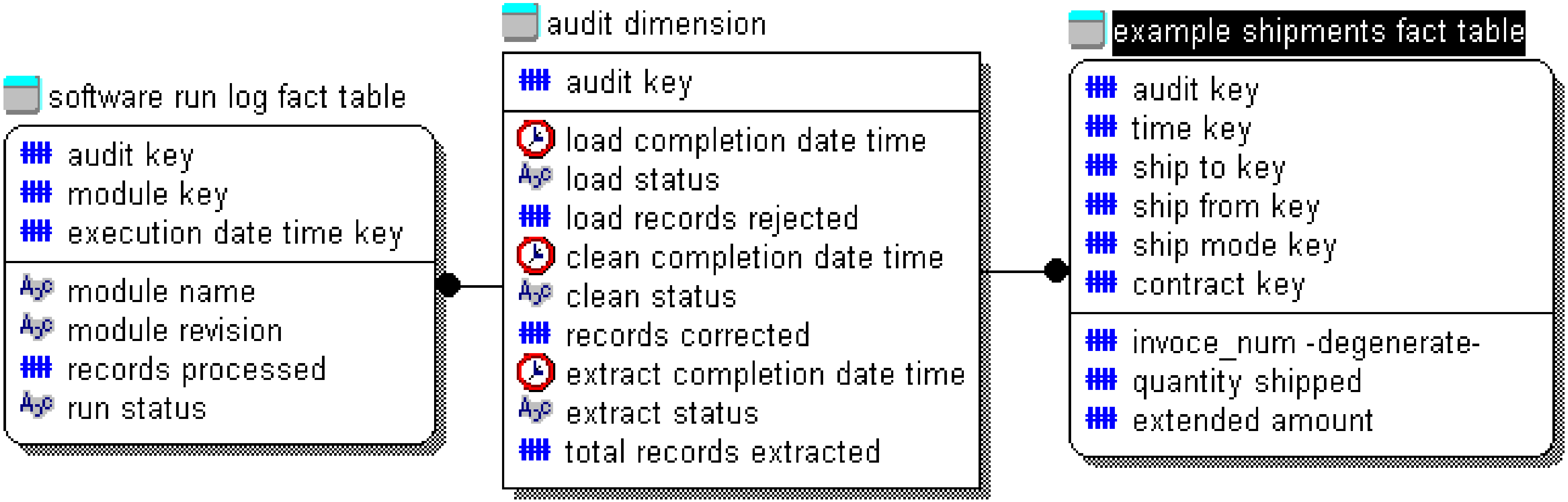
- diagnosis group key
- diagnosis key (FK)
- weighting factor

billable patient encounter fact table

- time key
- patient key
- provider key
- location key
- payer key
- procedure key
- diagnosis group key (FK)
- billed to payer amount
- billed to patient amount

Data Warehouse Modeling
Audit dimension

- constructed during the extract process in the data staging area



software run log fact table

audit key
module key
execution date time key

module name
module revision
records processed
run status

audit dimension

audit key

load completion date time
load status
load records rejected
clean completion date time
clean status
records corrected
extract completion date time
extract status
total records extracted

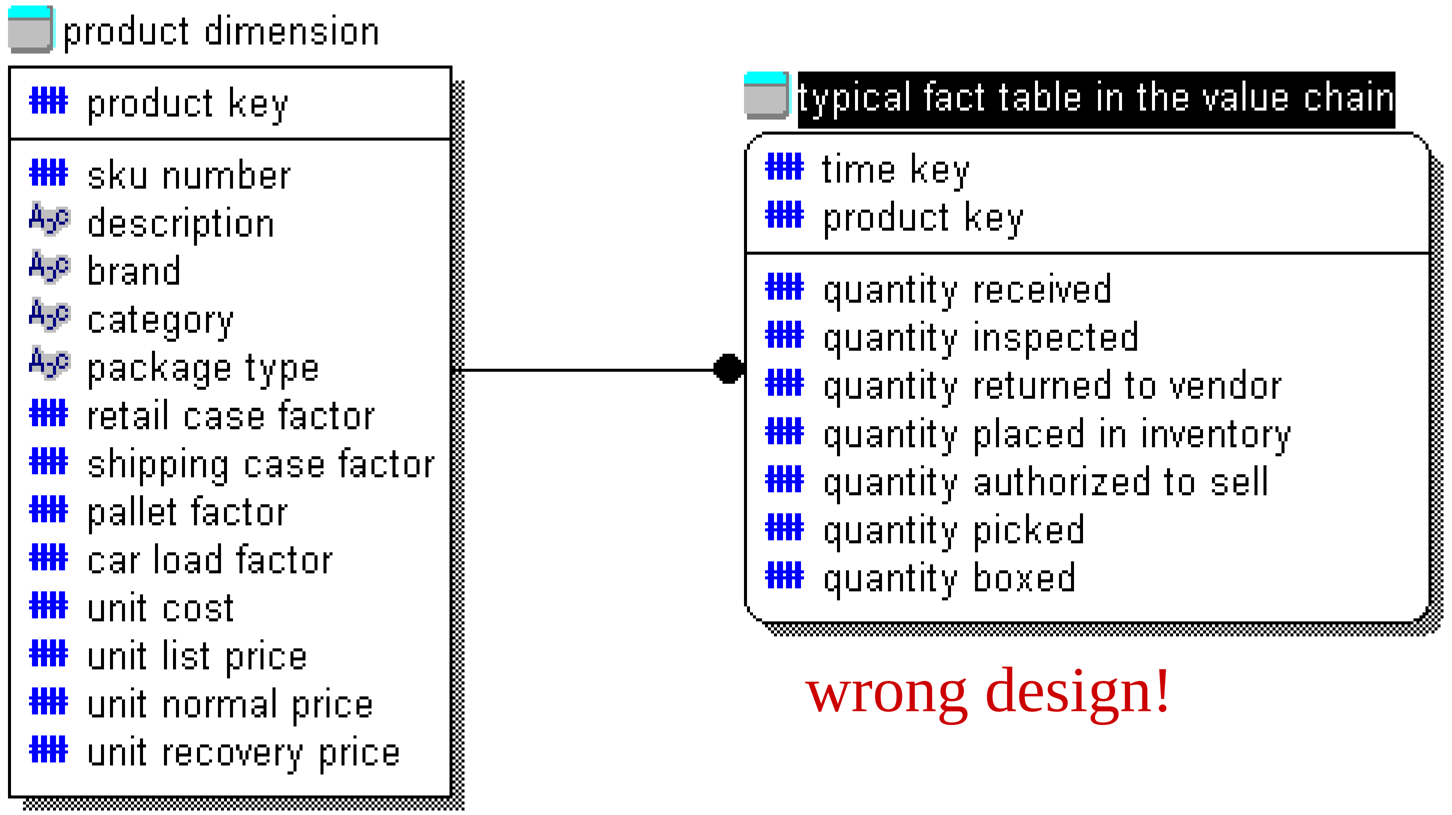
example shipments fact table

audit key
time key
ship to key
ship from key
ship mode key
contract key

invoce_num -degenerate-
quantity shipped
extended amount

Data Warehouse Modeling

Multiple units of measure



product dimension

- product key
- sku number
- description
- brand
- category
- package type
- retail case factor
- shipping case factor
- pallet factor
- car load factor
- unit cost
- unit list price
- unit normal price
- unit recovery price

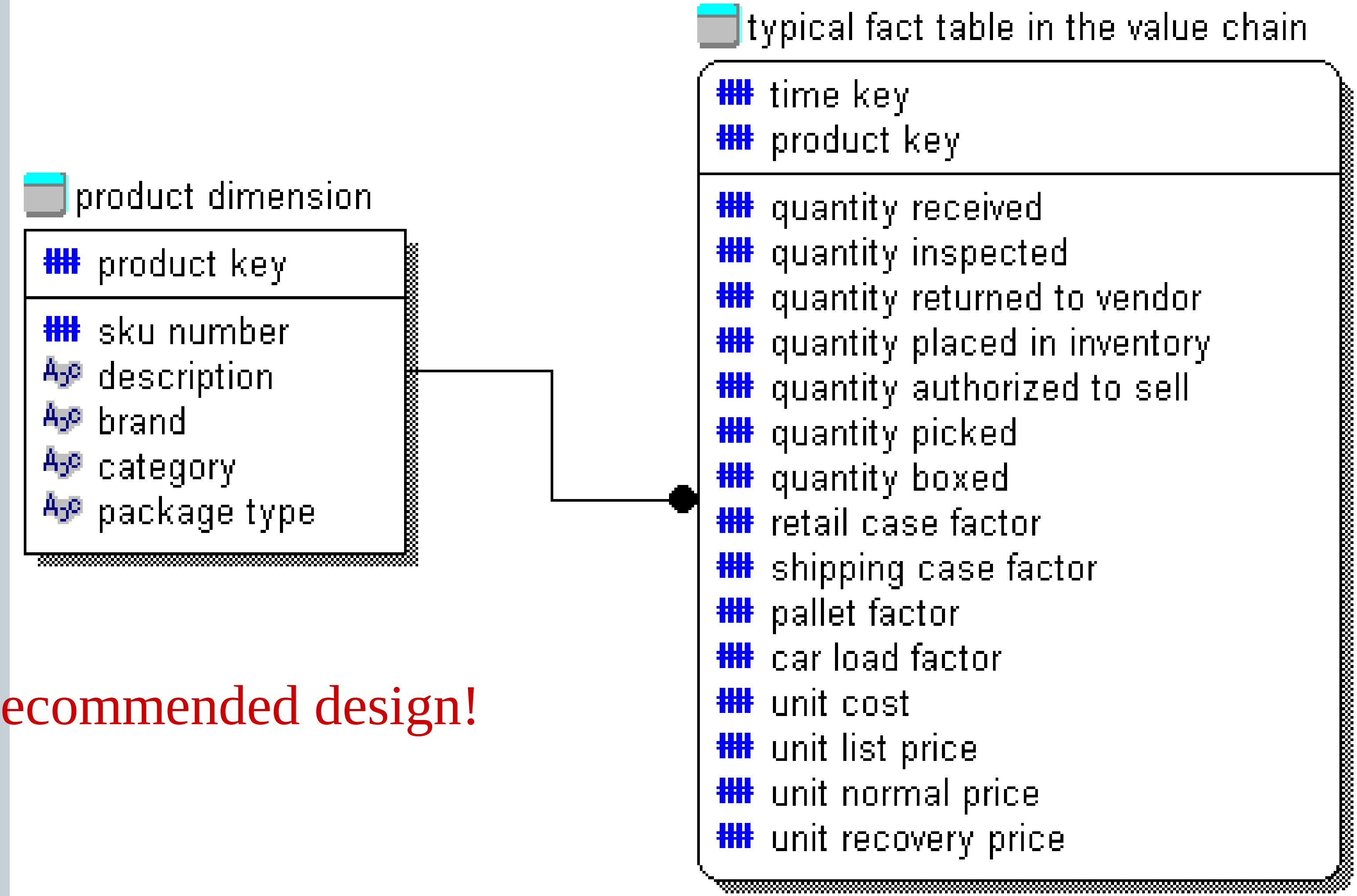
typical fact table in the value chain

- time key
- product key
- quantity received
- quantity inspected
- quantity returned to vendor
- quantity placed in inventory
- quantity authorized to sell
- quantity picked
- quantity boxed

wrong design!

Data Warehouse Modeling

Multiple units of measure



product dimension

- product key
- sku number
- description
- brand
- category
- package type

typical fact table in the value chain

- time key
- product key
- quantity received
- quantity inspected
- quantity returned to vendor
- quantity placed in inventory
- quantity authorized to sell
- quantity picked
- quantity boxed
- retail case factor
- shipping case factor
- pallet factor
- car load factor
- unit cost
- unit list price
- unit normal price
- unit recovery price

recommended design!

Tarea

- Degenerate dimensions
- Monster dimensions
- Junk dimensions
- Help for dimensional modeling
- Five alternatives for better employee dimension modeling
- Joe Caserta, “What Changed?”
- Slowly Changing Dimensions
  - http://www.kimballgroup.com/2008/08/slowly-changing-dimensions/
  - http://www.kimballgroup.com/2008/09/slowly-changing-dimensions-part-2/
  - http://www.kimballgroup.com/2013/02/design-tip-152-slowly-changing-dimensions-types-0-4-5-6-7/
  - The Data Warehouse: ETL Toolkit. Chapter 5.
- Design Tip #107 the MERGE statement for Slowly Changing Dimension Processing (http://www.kimballgroup.com/2008/11/design-tip-107-using-the-sql-merge-statement-for-slowly-changing-dimension-processing/)

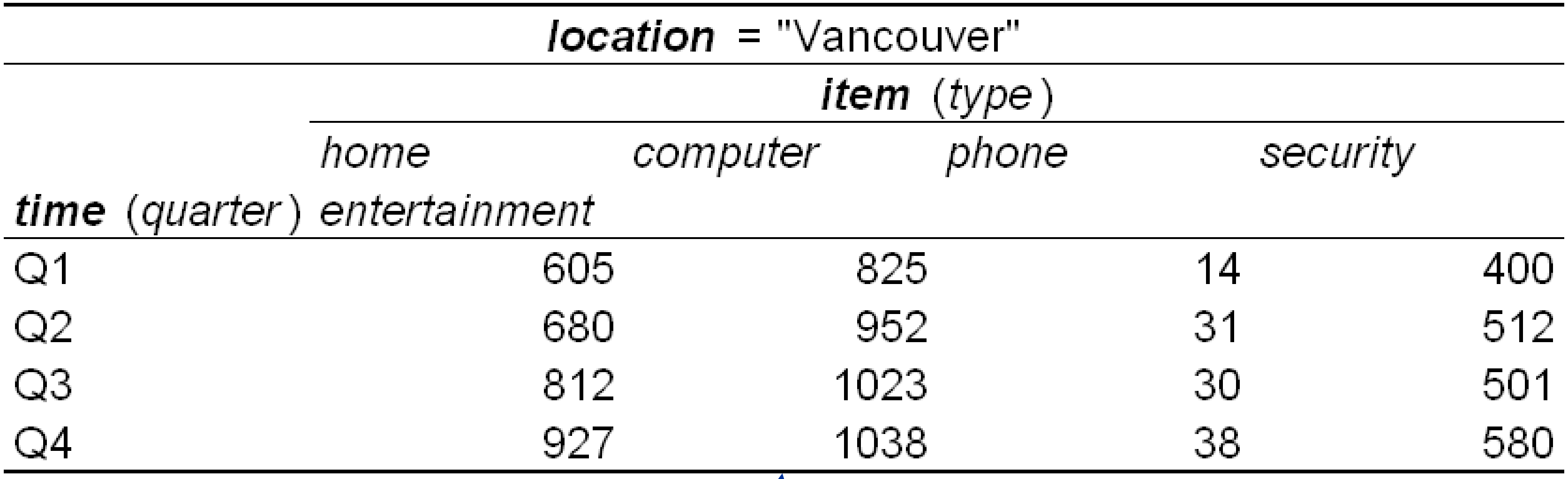
Data Warehouse Functionalities

OLTP vs OLAP

Características	OLTP	OLAP
Datos	Actuales y actualizables	Históricos y estáticos
Almacenamiento	Base de datos pequeñas y medianas (Mb y Gb)	Bases de datos grandes (Gb y Tb)
Procesos	Repetitivos	No previsibles
Estructura	Detallada	Detallada con Niveles de agregación
Usos	Soporte operacional orientado a procesos	Soporte de análisis orientado a información relevante
Unidades de ejecución	Transaccional	Consultas
Cantidad de datos	Miles	Millones
Modelo de acceso	Escritura, Lectura, elevado número de transacciones	Lectura, número de transacciones bajo o medio
Tiempo de respuesta	Segundos - minutos	Segundos – horas
Decisiones	Operativas diarias	Estratégicas
Tipos de usuario	Operativos	Administrativos
Número de usuarios	Miles	Cientos o menos

Representación multidimensional por medio de una rejilla de cuboides

a simple 2-D data cube

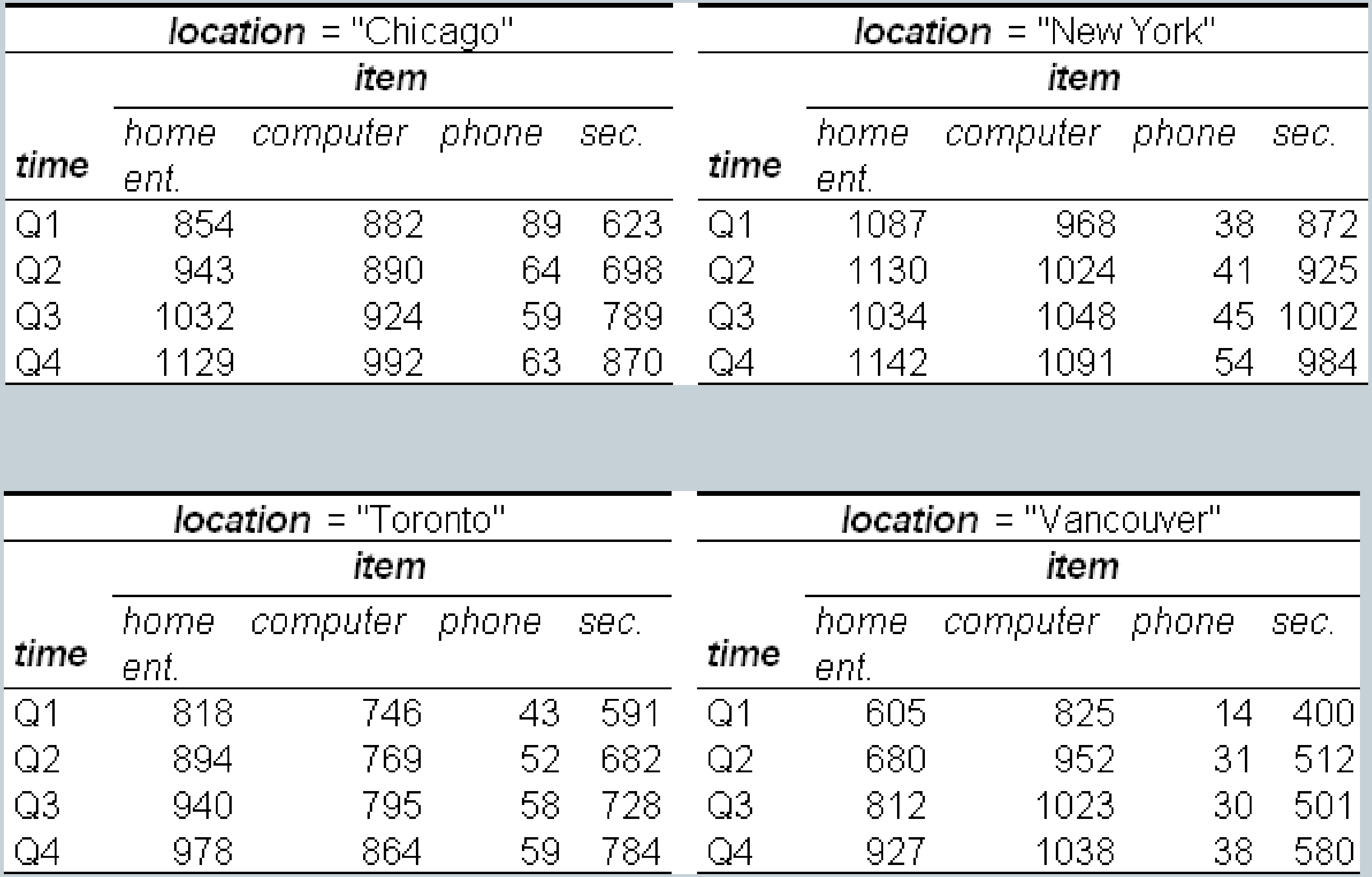


	location = "Vancouver"			
	item (*type*)			
time (*quarter*)	*home entertainment*	*computer*	*phone*	*security*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

dollars_sold (in thousands)

Representación multidimensional por medio de una rejilla de cuboides

a 3-D data cube



location = "Chicago"				
	item			
time	*home ent.*	*computer*	*phone*	*sec.*
Q1	854	882	89	623
Q2	943	890	64	698
Q3	1032	924	59	789
Q4	1129	992	63	870

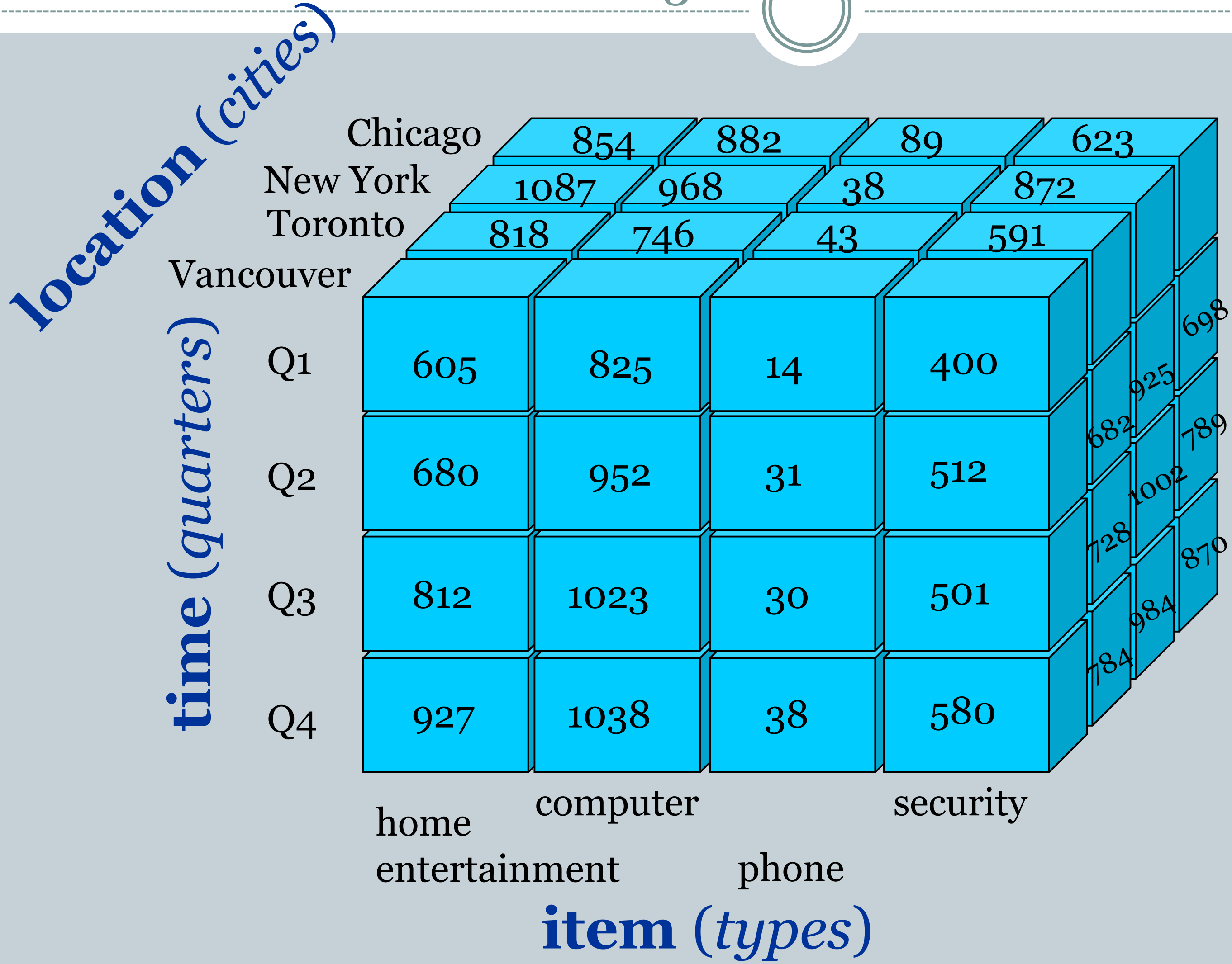
location = "New York"				
	item			
time	*home ent.*	*computer*	*phone*	*sec.*
Q1	1087	968	38	872
Q2	1130	1024	41	925
Q3	1034	1048	45	1002
Q4	1142	1091	54	984

location = "Toronto"				
	item			
time	*home ent.*	*computer*	*phone*	*sec.*
Q1	818	746	43	591
Q2	894	769	52	682
Q3	940	795	58	728
Q4	978	864	59	784

location = "Vancouver"				
	item			
time	*home ent.*	*computer*	*phone*	*sec.*
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Representación multidimensional por medio de una rejilla de cuboides

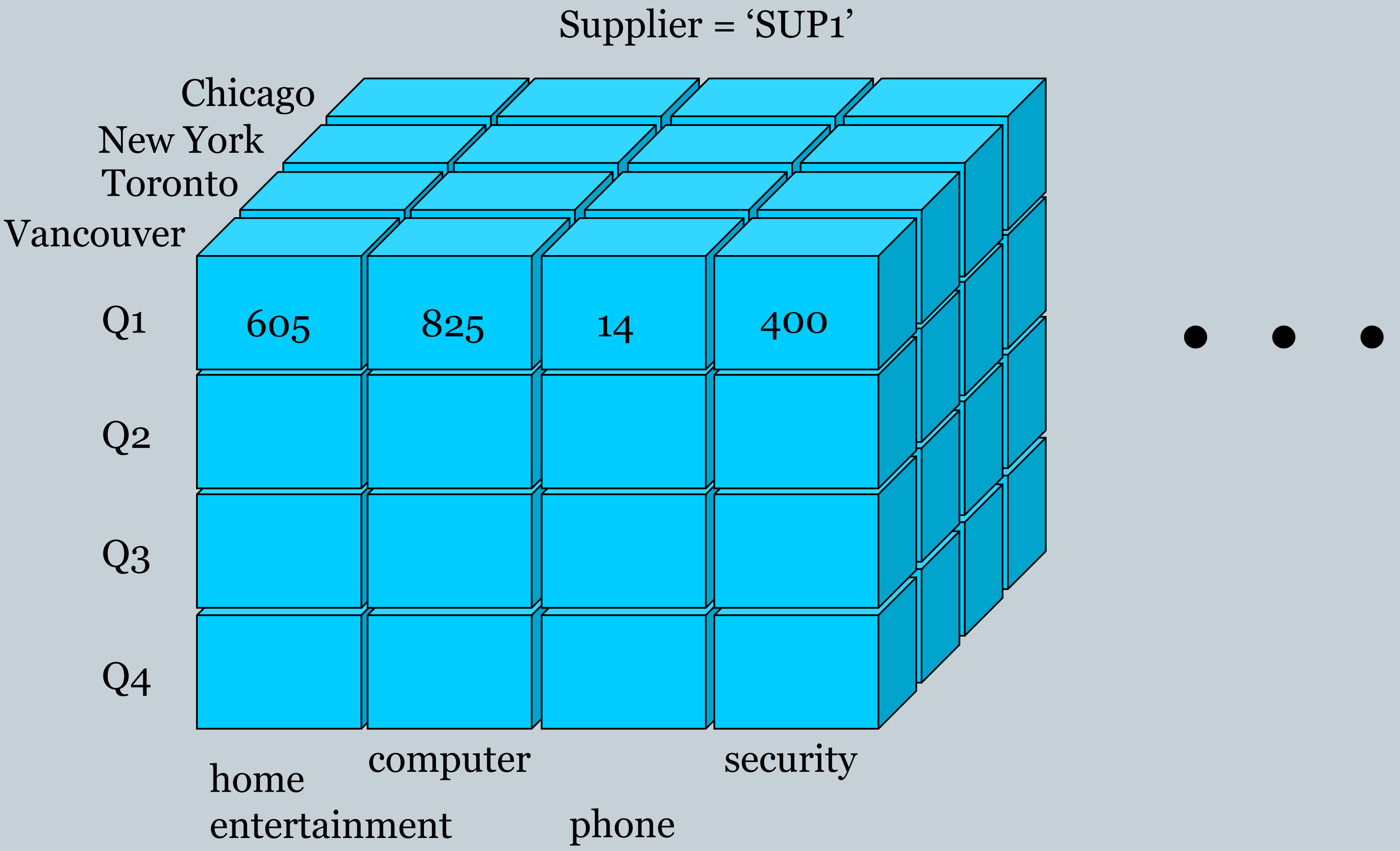
a 3-D data cube



	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Representación multidimensional por medio de una rejilla de cuboides

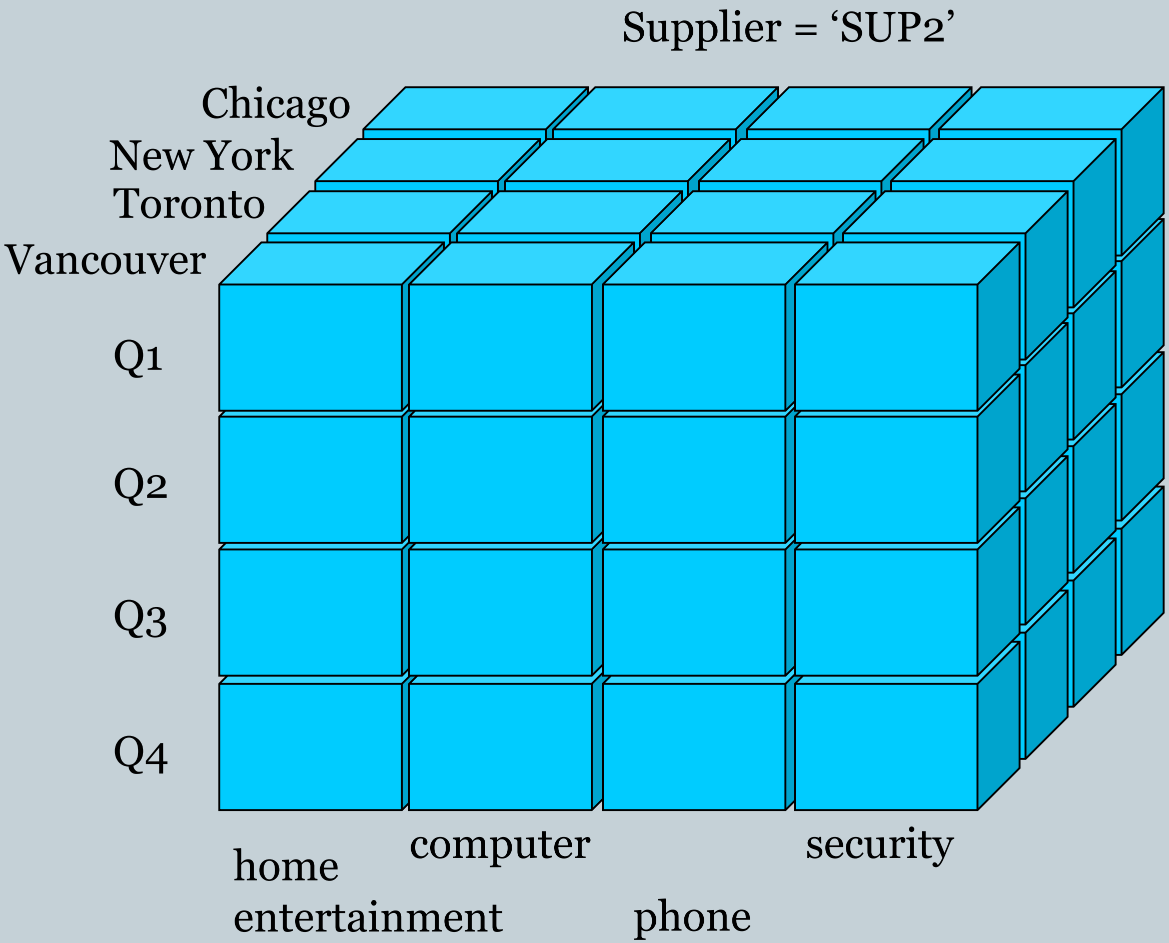
a 4-D cube as a series of 3-D cubes



Representación multidimensional por medio de una rejilla de cuboides

a 4-D cube as a series of 3-D cubes

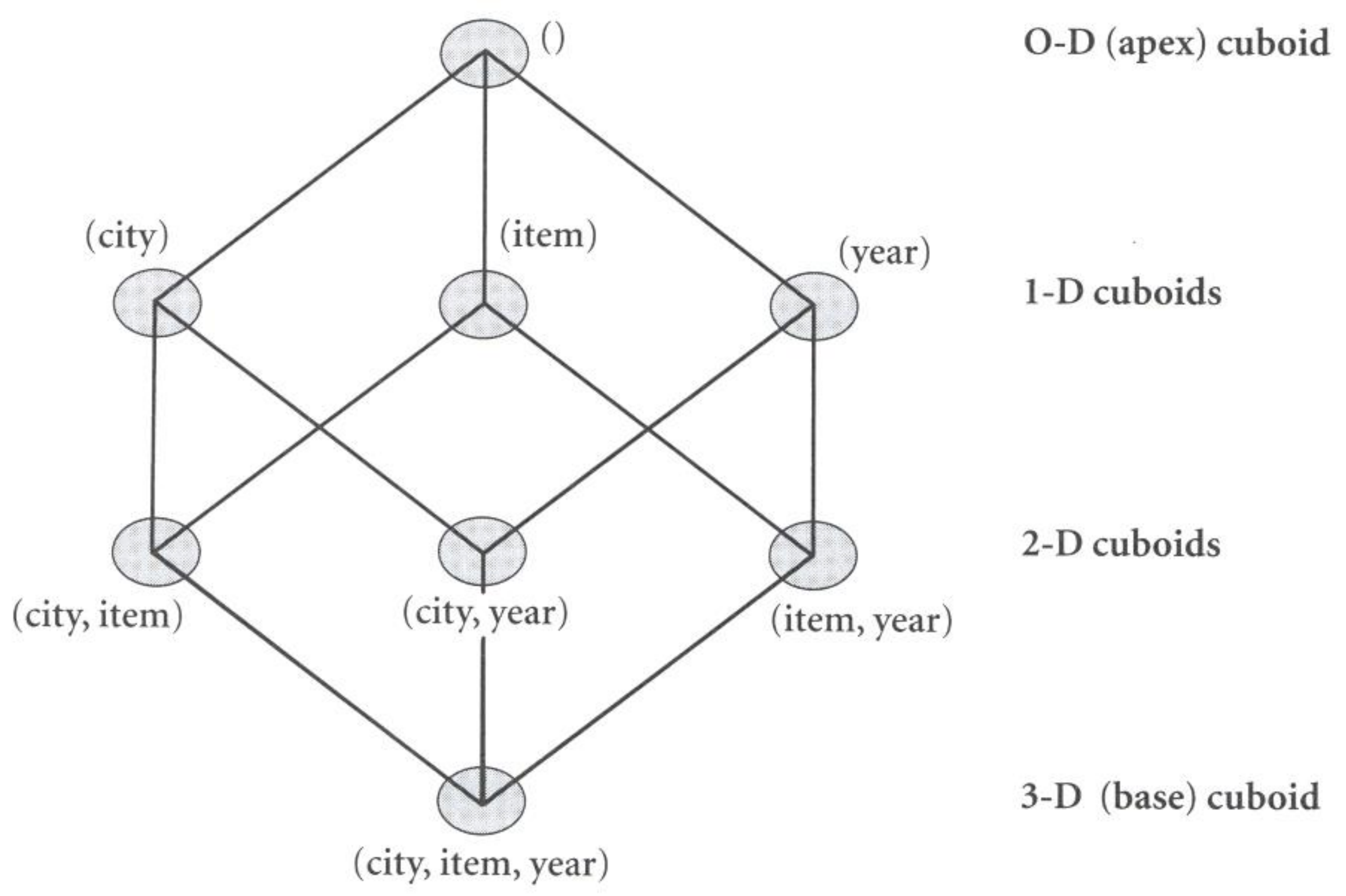
• • •



Representación multidimensional por medio de una rejilla de cuboides

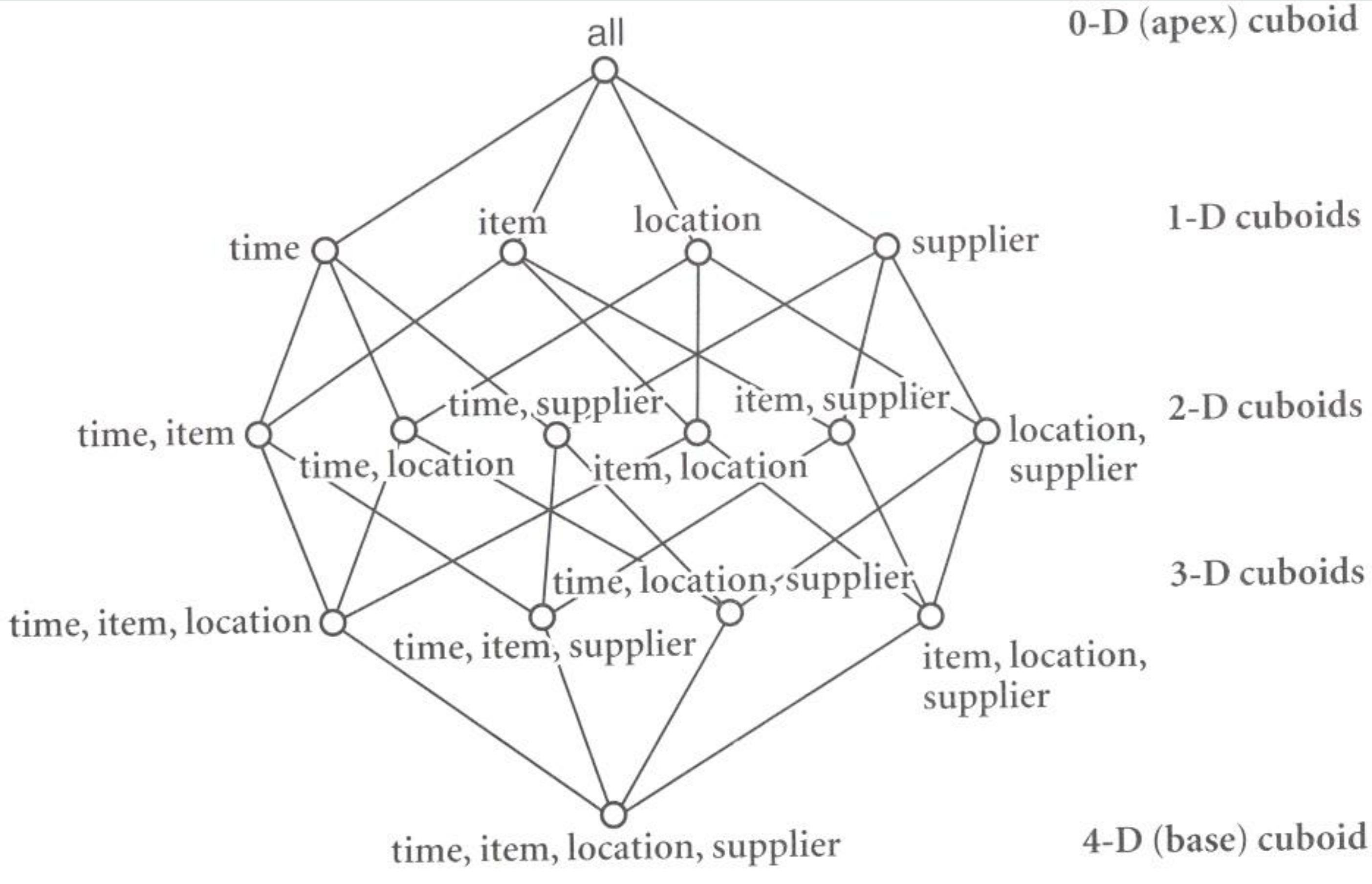
- *n*-D data data cube as a series of (*n*-1)-D cubes
- cuboide:
  - each data cube
  - data at a degree of summarization or ***group by***
- lattice of cuboids

Construcción del cubo de datos por MOLAP



Representación multidimensional por medio de una rejilla de cuboides

lattice of cuboids



Operaciones OLAP

- *slice-and-dice* queries
- *drill-down* and *roll-up* queries
- *drill-accross* queries

  combines cubes that share one or more dimensions
- *drill-through* queries

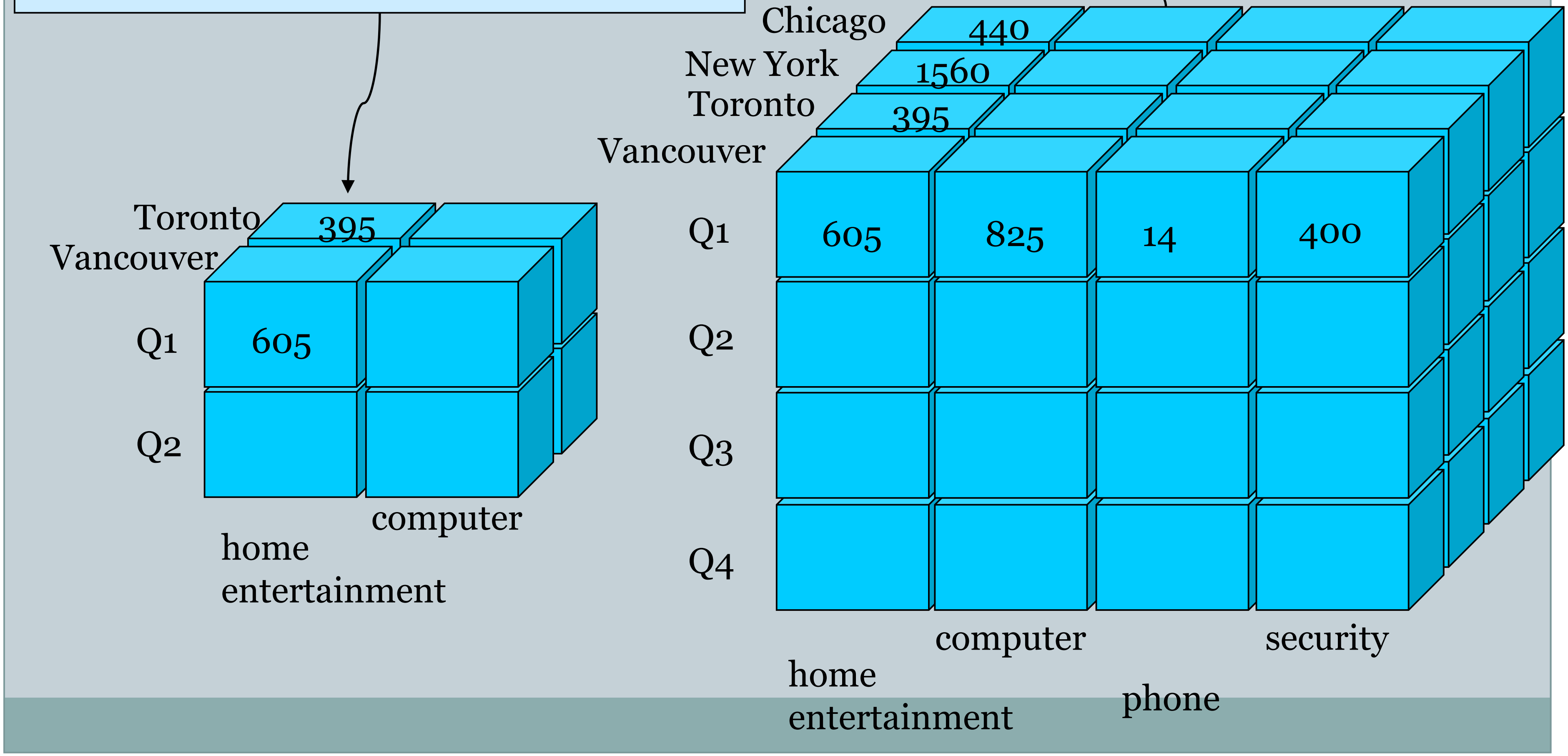
  make use of relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables
- *ranking* (*top n* / *bottom n*) queries
- *rotating (pivoting)*

  a cube allows users to see the data grouped by other dimensions

Operaciones OLAP

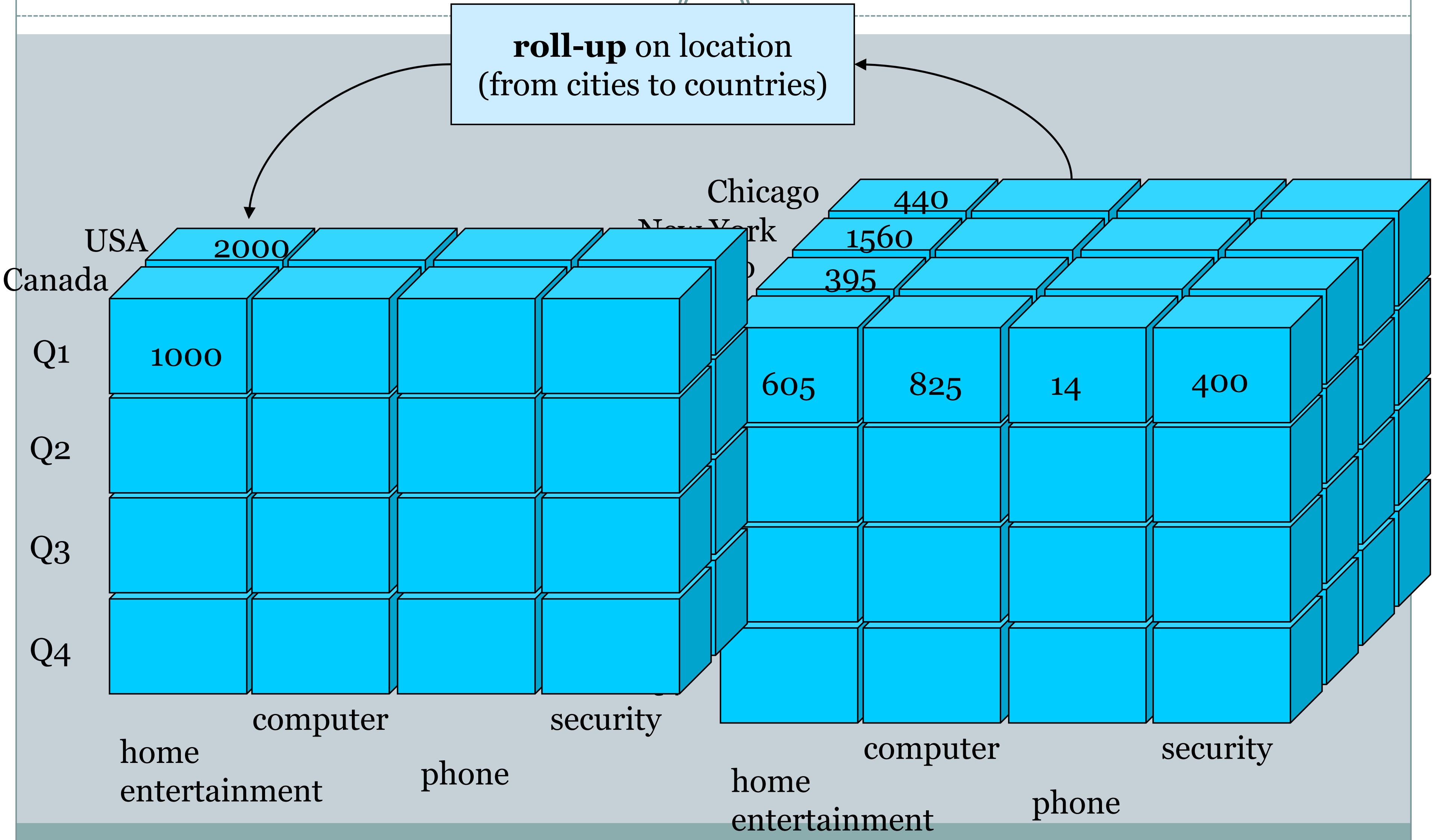
dice

dice for
(location = “Toronto” or “Vancouver”)
and (time = “Q1” or “Q2”) and
(item = “home entertainment” or “computer”



Operaciones OLAP

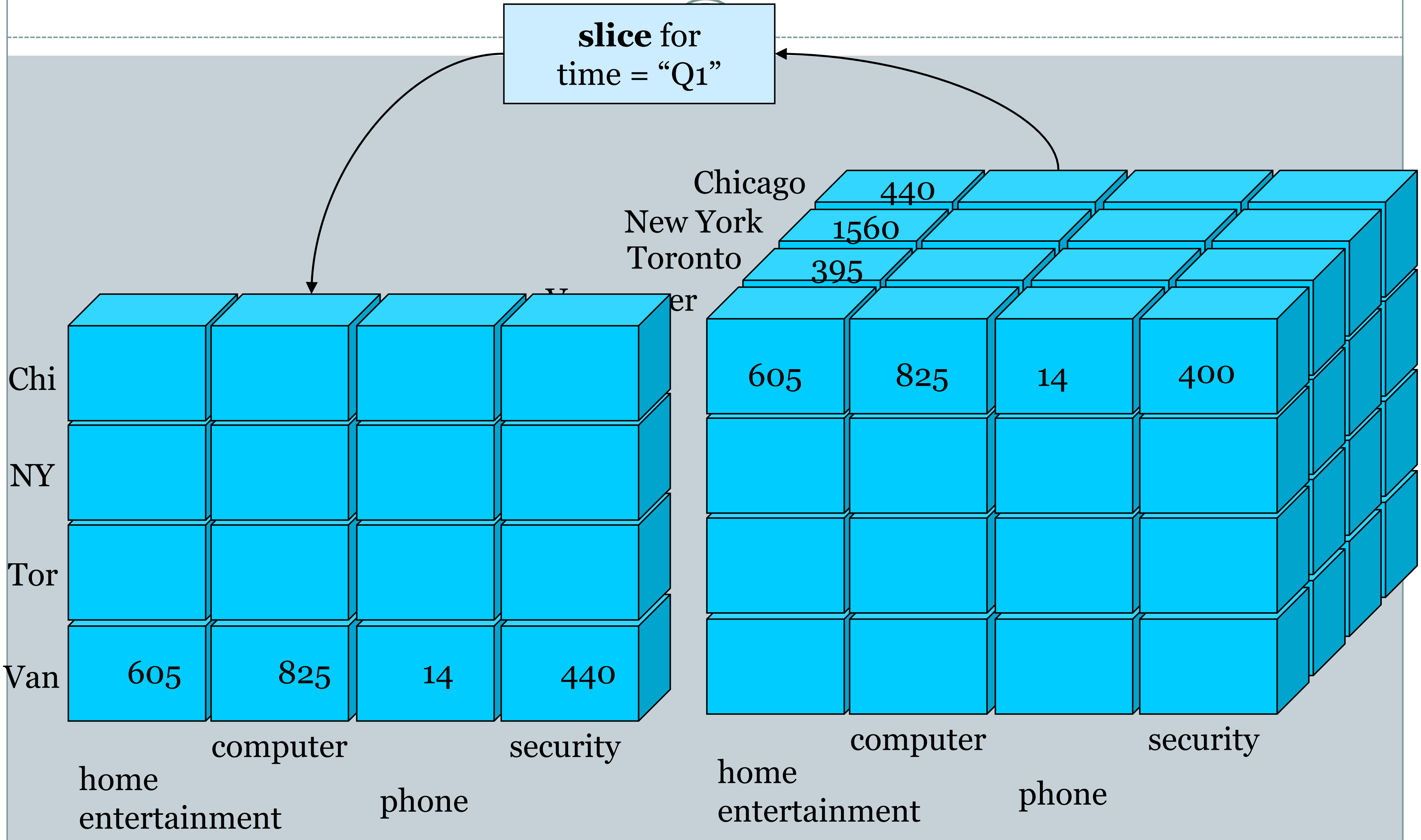
roll-up



roll-up on location
(from cities to countries)

Operaciones OLAP

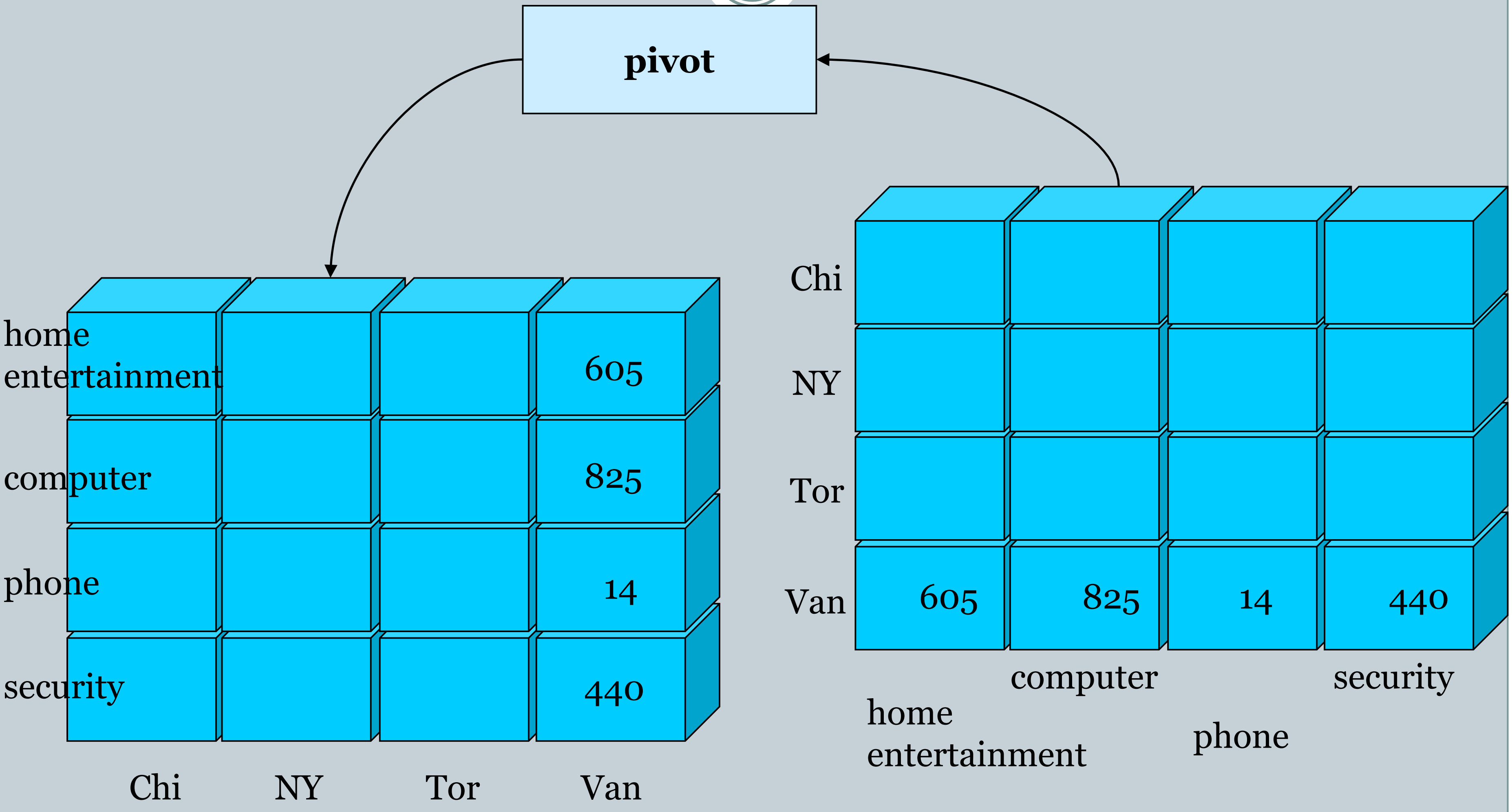
slice



slice for
time = "Q1"

Operaciones OLAP

pivot



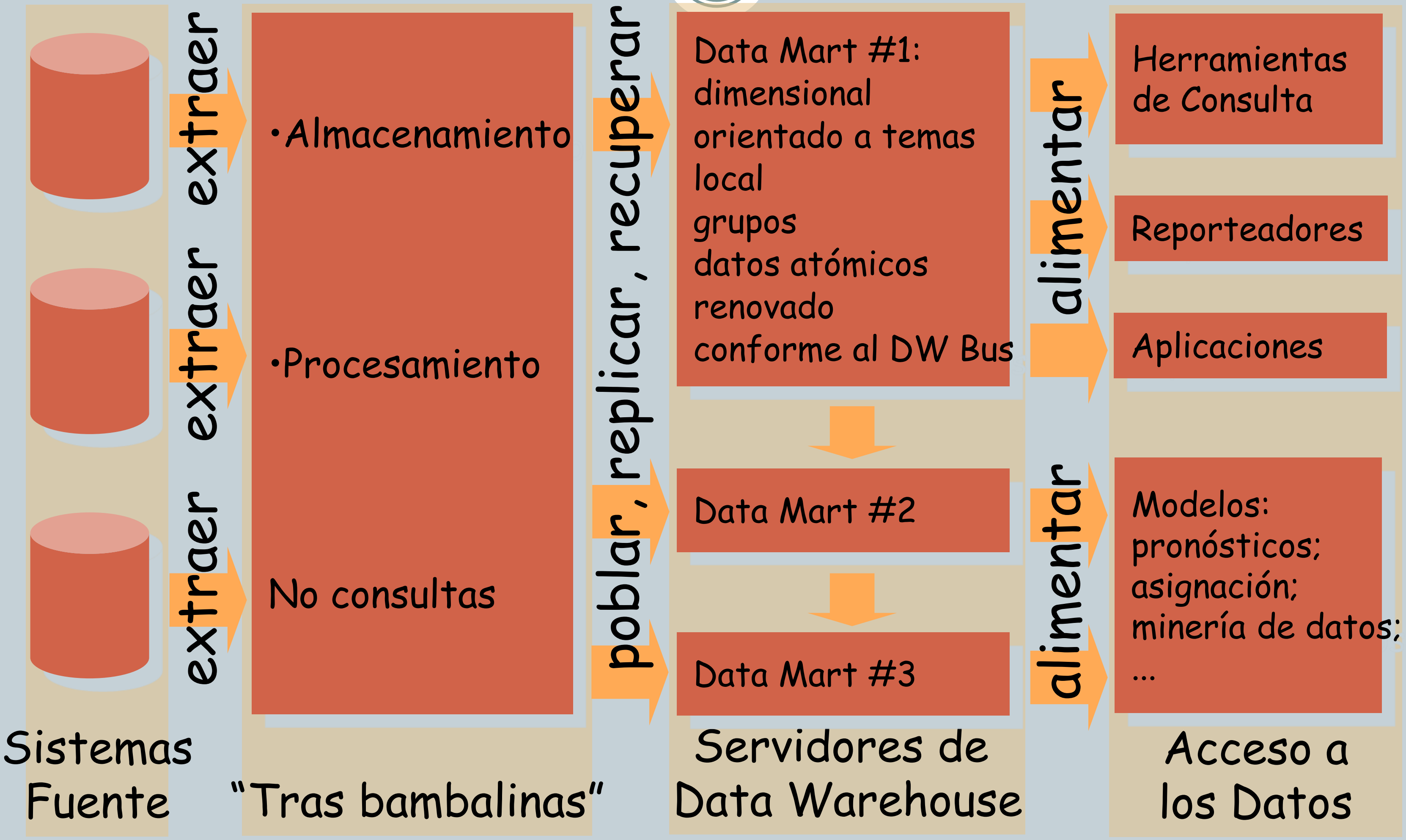
pivot

	Chi	NY	Tor	Van
home entertainment				605
computer				825
phone				14
security				440

	home entertainment	computer	phone	security
Chi				
NY				
Tor				
Van	605	825	14	440

Data Warehouse Concepts

Elementos básicos de la arquitectura



extraer

extraer

extraer

- Almacenamiento

- Procesamiento

No consultas

poblar, replicar, recuperar

Data Mart #1:
dimensional
orientado a temas
local
grupos
datos atómicos
renovado
conforme al DW Bus

Data Mart #2

Data Mart #3

alimentar

alimentar

Herramientas de Consulta

Reporteadores

Aplicaciones

Modelos:
pronósticos;
asignación;
minería de datos;
…

Sistemas Fuente

“Tras bambalinas”

Servidores de Data Warehouse

Acceso a los Datos

Data Warehouse Functionalities

ETL Process

- **E**xtract, **T**ransform, **L**oad
- Proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos, limpiarlos y cargarlos a otra base de datos, datamart o data warehouse para analizar o en otro sistema operacional para apoyar un proceso de negocio

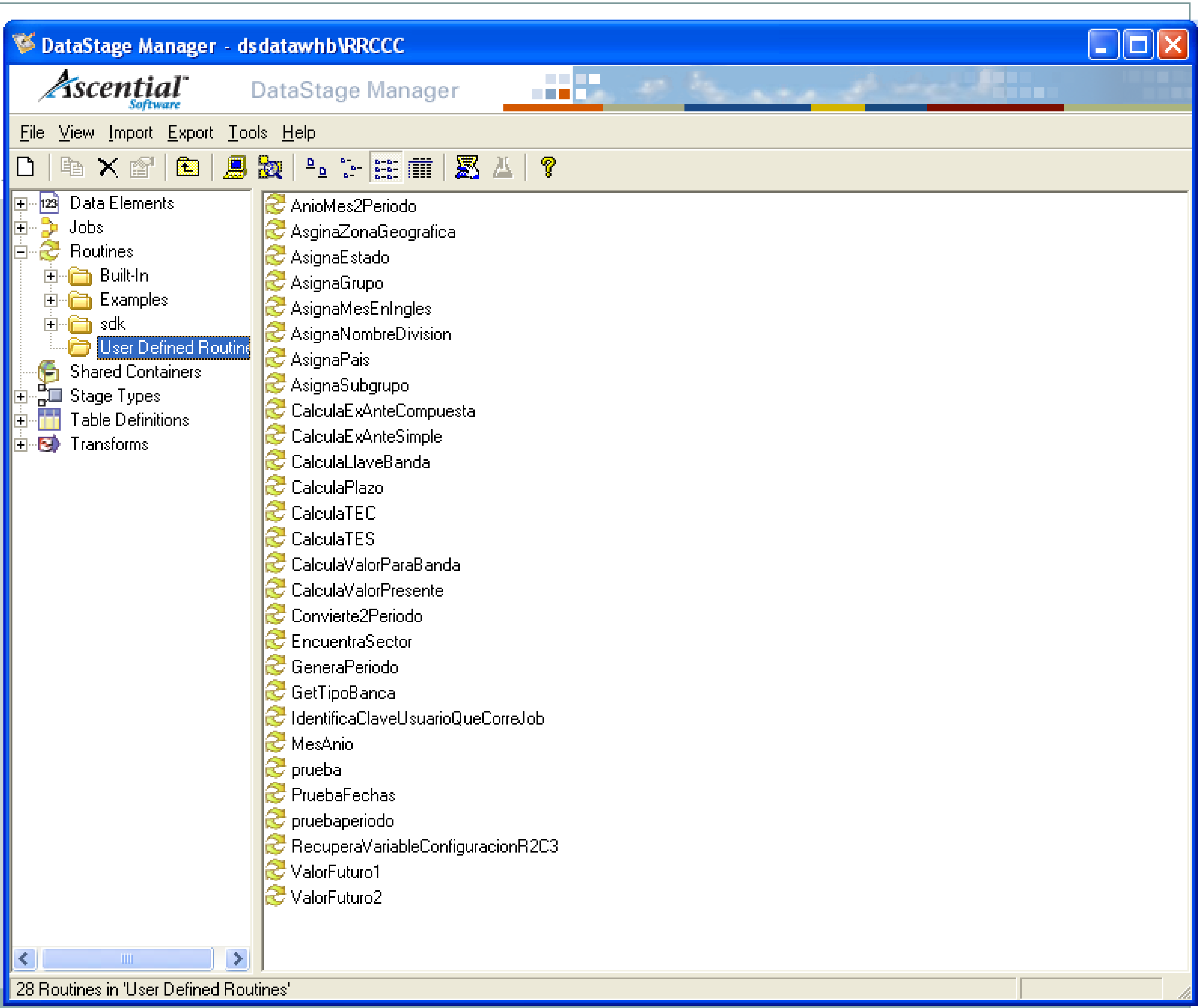
Data Warehouse Functionalities

ETL: Data Staging Area

- The construction site for the data warehouse
- Data Staging Storage Types
    - Flat files
    - Relational tables
    - Propietary structures used by data staging tools
- Many data staging tools are designed to work with relational databases

Data Warehouse Functionalities

General Data Staging Requirements: Metadata driven

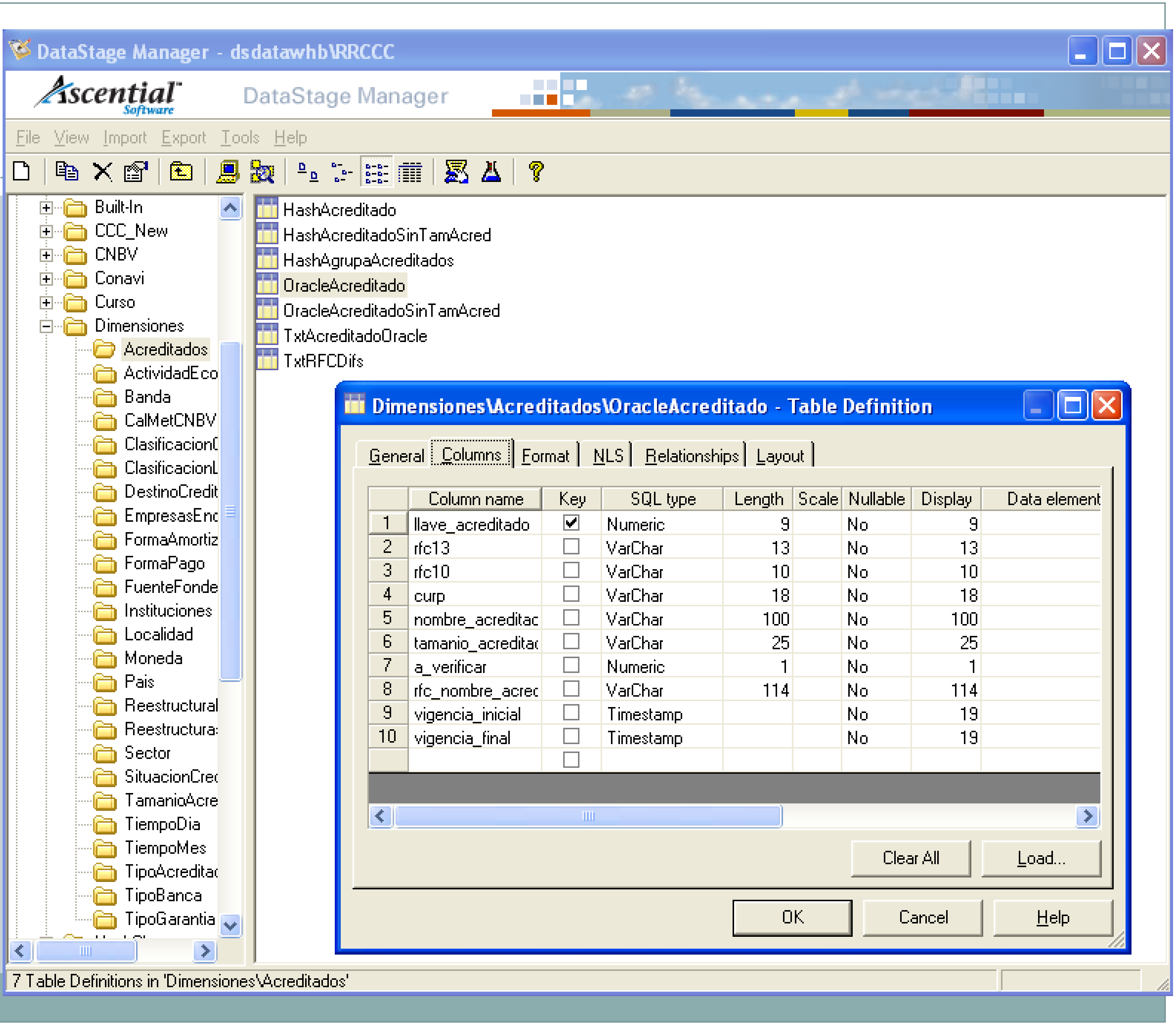


AnioMes2Periodo
AsginaZonaGeografica
AsignaEstado
AsignaGrupo
AsignaMesEnIngles
AsignaNombreDivision
AsignaPais
AsignaSubgrupo
CalculaExAnteCompuesta
CalculaExAnteSimple
CalculaLlaveBanda
CalculaPlazo
CalculaTEC
CalculaTES
CalculaValorParaBanda
CalculaValorPresente
Convierte2Periodo
EncuentraSector
GeneraPeriodo
GetTipoBanca
IdentificaClaveUsuarioQueCorreJob
MesAnio
prueba
PruebaFechas
pruebaperiodo
RecuperaVariableConfiguracionR2C3
ValorFuturo1
ValorFuturo2

28 Routines in 'User Defined Routines'

Data Warehouse Functionalities

General Data Staging Requirements: Metadata driven


DataStage Manager - dsdatawhb\RRCCC
Ascential Software
DataStage Manager
File View Import Export Tools Help

Built-In
CCC_New
CNBV
Conavi
Curso
Dimensiones
Acreditados
ActividadEco
Banda
CalMetCNBV
Clasificacion(
ClasificacionL
DestinoCredit
EmpresasEnc
FormaAmortiz
FormaPago
FuenteFonde
Instituciones
Localidad
Moneda
Pais
Reestructural
Reestructura:
Sector
SituacionCre(
TamanioAcre
TiempoDia
TiempoMes
TipoAcreditac
TipoBanca
TipoGarantia

HashAcreditado
HashAcreditadoSinTamAcred
HashAgrupaAcreditados
OracleAcreditado
OracleAcreditadoSinTamAcred
TxtAcreditadoOracle
TxtRFCDifs

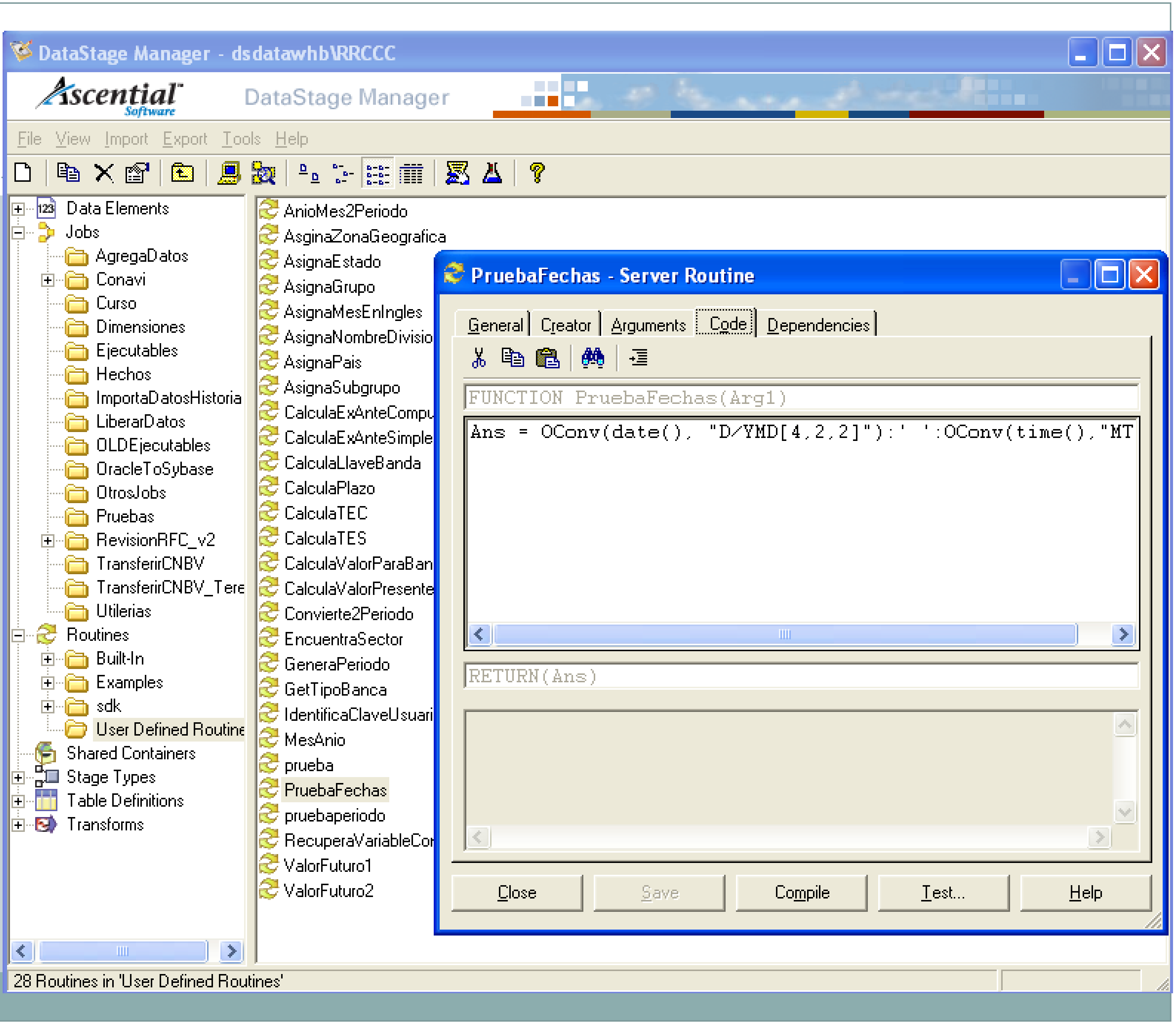
Dimensiones\Acreditados\OracleAcreditado - Table Definition

	Column name	Key	SQL type	Length	Scale	Nullable	Display	Data element
1	llave_acreditado	☑	Numeric	9		No	9	
2	rfc13	□	VarChar	13		No	13	
3	rfc10	□	VarChar	10		No	10	
4	curp	□	VarChar	18		No	18	
5	nombre_acreditac	□	VarChar	100		No	100	
6	tamanio_acreditac	□	VarChar	25		No	25	
7	a_verificar	□	Numeric	1		No	1	
8	rfc_nombre_acrec	□	VarChar	114		No	114	
9	vigencia_inicial	□	Timestamp			No	19	
10	vigencia_final	□	Timestamp			No	19	
		□						

7 Table Definitions in 'Dimensiones\Acreditados'

Data Warehouse Functionalities

General Data Staging Requirements: Metadata driven



```
FUNCTION PruebaFechas(Arg1)
Ans = OConv(date(), "D/YMD[4,2,2]"):' ':OConv(time(),"MT
RETURN(Ans)
```

28 Routines in 'User Defined Routines'

Data Warehouse Functionalities

Data Staging (or Back Room) Services

- Extract Services
- Data Transformation Services
- Data Loading Services
- Data Staging Job Control Services

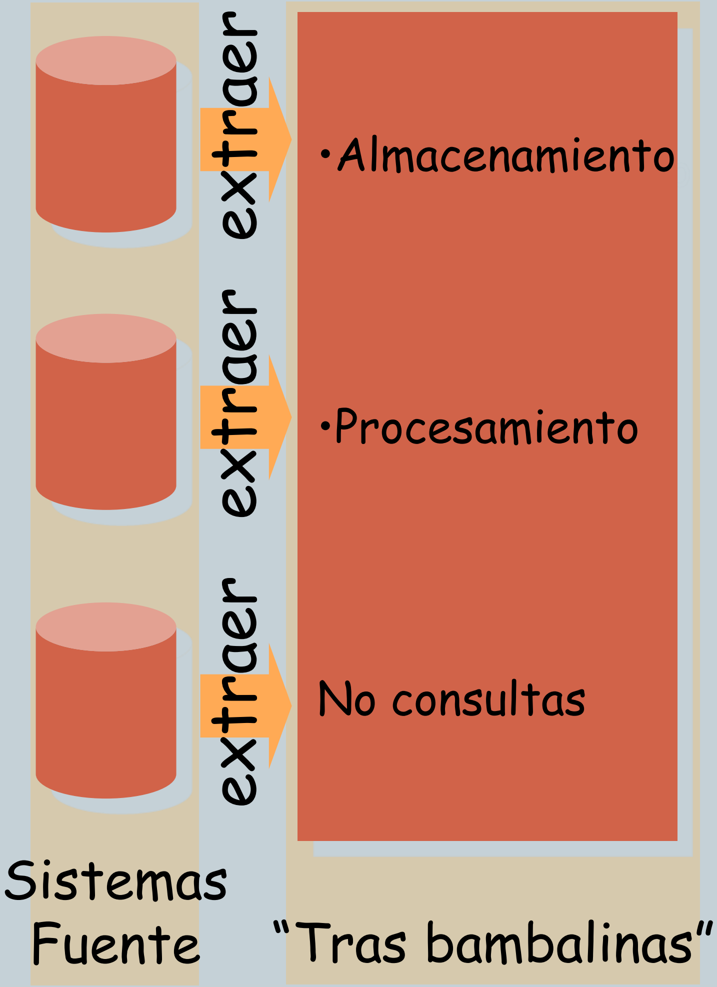
Data Warehouse Functionalities

Incrementa extraction: Extract Services

- Multiple Sources
- Code Generation
- Multiple Extract Types
- Replication
- Compression/Decompression

Data Warehouse Functionalities

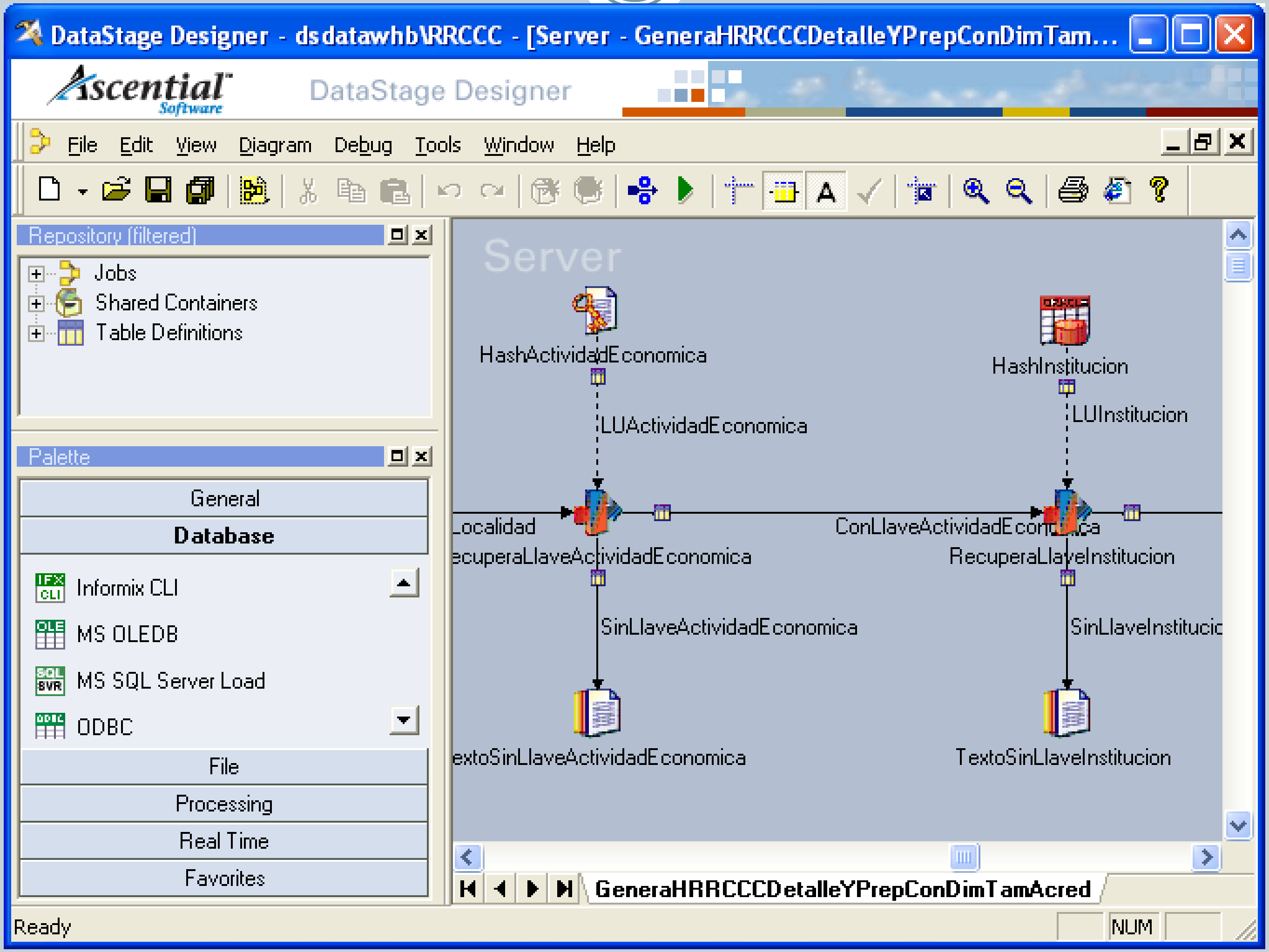
Incremental Extraction:Extract Services: Multiple Sources



- Multiple systems
- Multiple data stores
- Multiple plataforms

Data Warehouse Functionalities

Data Staging (or Back Room) Services



Data Warehouse Functionalities

Incremental Extraction: Extract Services: Multiple Extract Types

- Incremental loads
  - Based on a transaction date or some kind of indicator flag in the source system
  - Metadata: date of the last load
- Transaction events
  - All new transactions
  - Update records
  - Delete records
- Full Refresh

Data Warehouse Functionalities

Incremental extraction: Extract Services: Replication

- Continuously update a table during the day
- Valuable

  Multiple load process depend on access to update versions of the conformed dimension tables

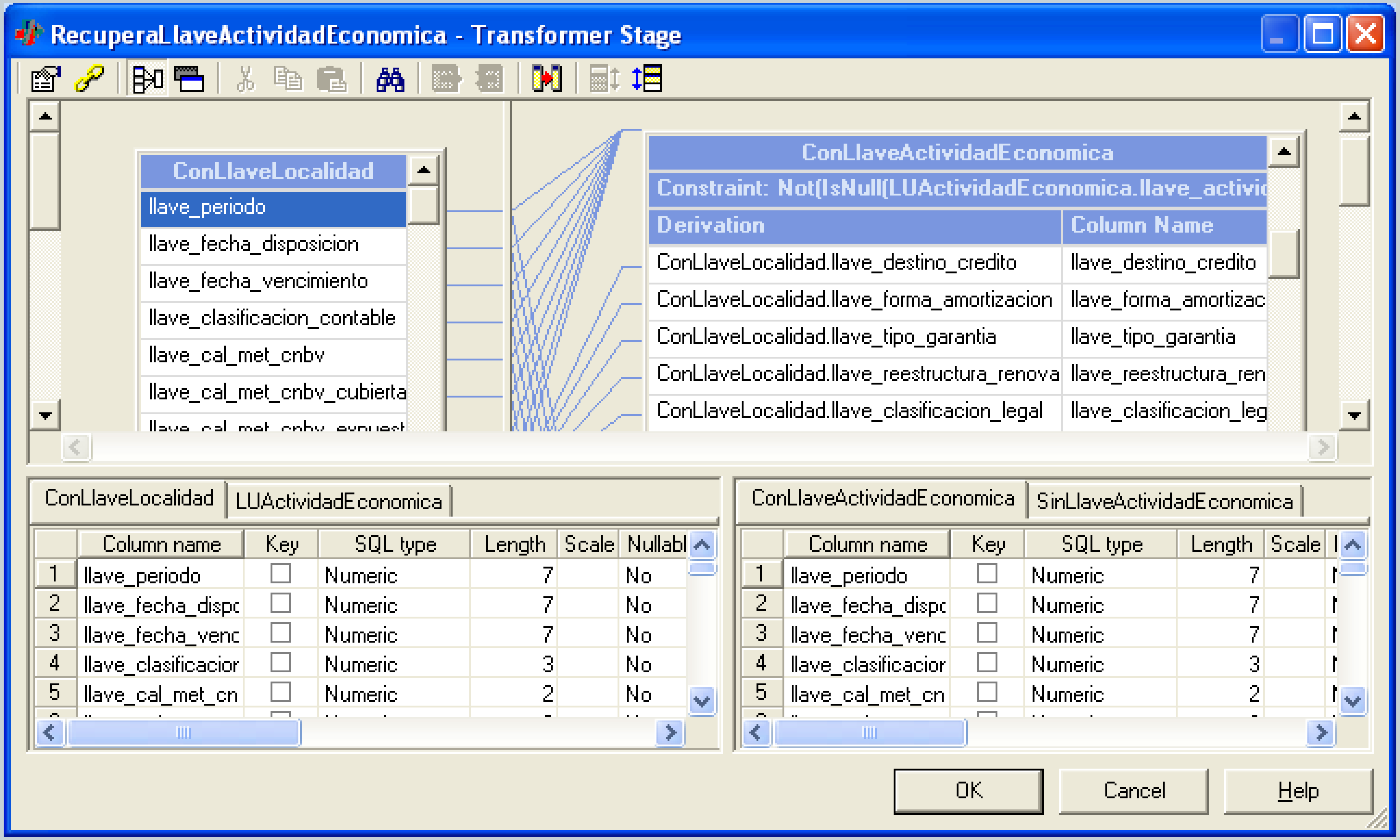
Data Warehouse Functionalities

Data Transformation Services

- Integration
- Slowly changing dimension maintenance
- Referential integrity checking
- Denormalization and renormalization
- Cleansing, deduping, merge/purge
- Data type conversion
- Calculation, derivation, allocation
- Aggregation
- Data content audit
- Data lineage audit
- Tool- or analysis-specific transformation
- Null values
- Pre- and post-step exists

Data Warehouse Functionalities

Data Transformation Services



RecuperaLlaveActividadEconomica - Transformer Stage

ConLlaveLocalidad
llave_periodo
llave_fecha_disposicion
llave_fecha_vencimiento
llave_clasificacion_contable
llave_cal_met_cnbv
llave_cal_met_cnbv_cubierta
llave_cal_met_cnbv_expuest

ConLlaveActividadEconomica

Constraint: Not(IsNull(LUActividadEconomica.llave_activic

Derivation	Column Name
ConLlaveLocalidad.llave_destino_credito	llave_destino_credito
ConLlaveLocalidad.llave_forma_amortizacion	llave_forma_amortizac
ConLlaveLocalidad.llave_tipo_garantia	llave_tipo_garantia
ConLlaveLocalidad.llave_reestructura_renova	llave_reestructura_ren
ConLlaveLocalidad.llave_clasificacion_legal	llave_clasificacion_leg

ConLlaveLocalidad | LUActividadEconomica

	Column name	Key	SQL type	Length	Scale	Nullabl
1	llave_periodo	□	Numeric	7		No
2	llave_fecha_dispc	□	Numeric	7		No
3	llave_fecha_venc	□	Numeric	7		No
4	llave_clasificacior	□	Numeric	3		No
5	llave_cal_met_cn	□	Numeric	2		No

ConLlaveActividadEconomica | SinLlaveActividadEconomica

	Column name	Key	SQL type	Length	Scale	I
1	llave_periodo	□	Numeric	7		I
2	llave_fecha_dispc	□	Numeric	7		I
3	llave_fecha_venc	□	Numeric	7		I
4	llave_clasificacior	□	Numeric	3		I
5	llave_cal_met_cn	□	Numeric	2		I

Data Warehouse Functionalities

Data Transformation Services: Integration

- Generation
  - Surrogate keys
  - Maping keys from keys one system to another
  - Mapping codes into full descriptions
- Maintainance
  - Master key lookup table

Data Warehouse Functionalities

Data Loading Services

- Support for multiple targets
- Load optimization
- Entire load process support

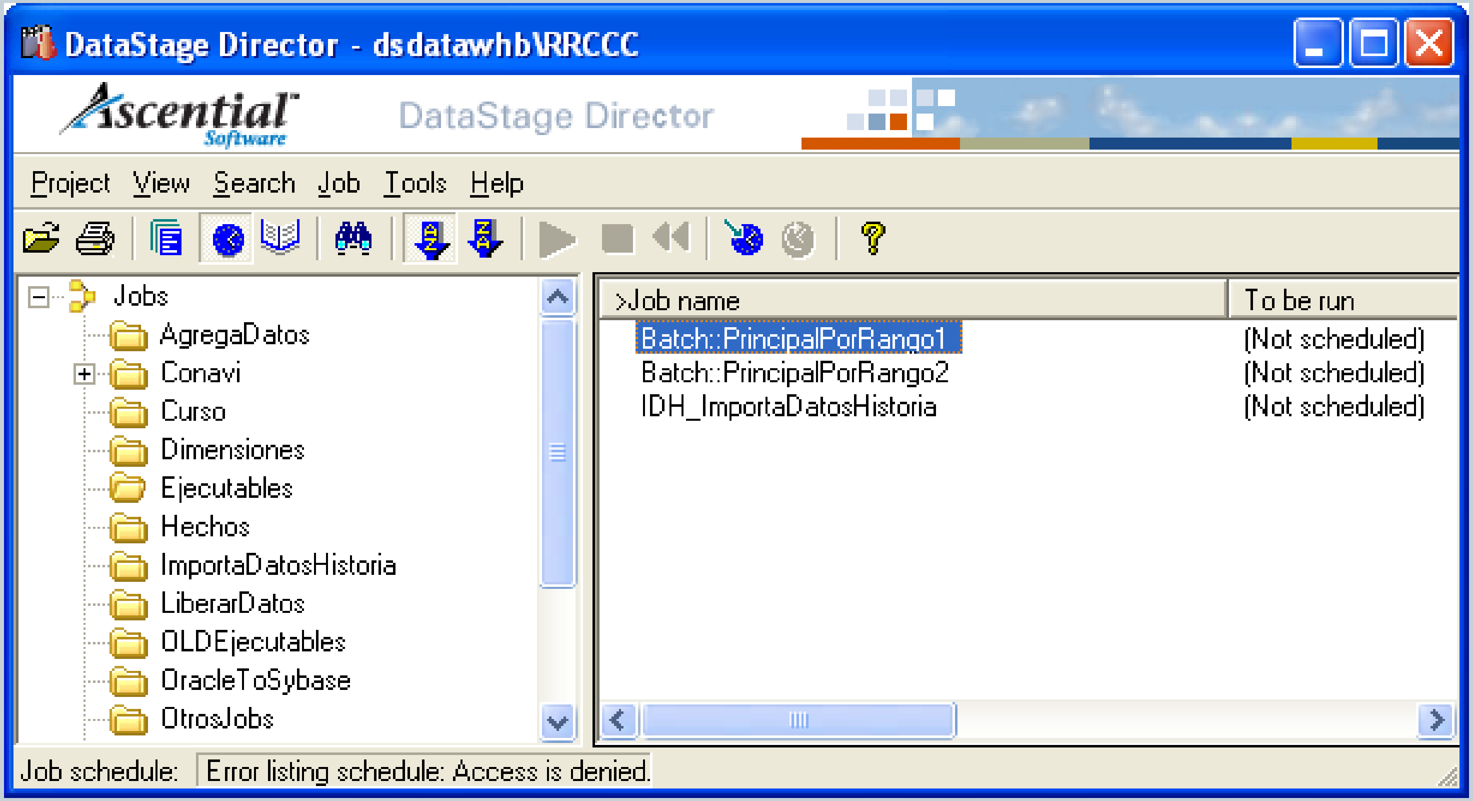
Data Warehouse Functionalities

Control: Data Staging Job Control Services

- Job definition
- Job scheduling
- Monitoring
- Logging
- Exception handling
- Error handling
- Notification

Data Warehouse Functionalities

Control: Data Staging Job Control Services: Scheduling


DataStage Director - dsdatawhb\RRCCC
Ascential Software
DataStage Director
Project View Search Job Tools Help

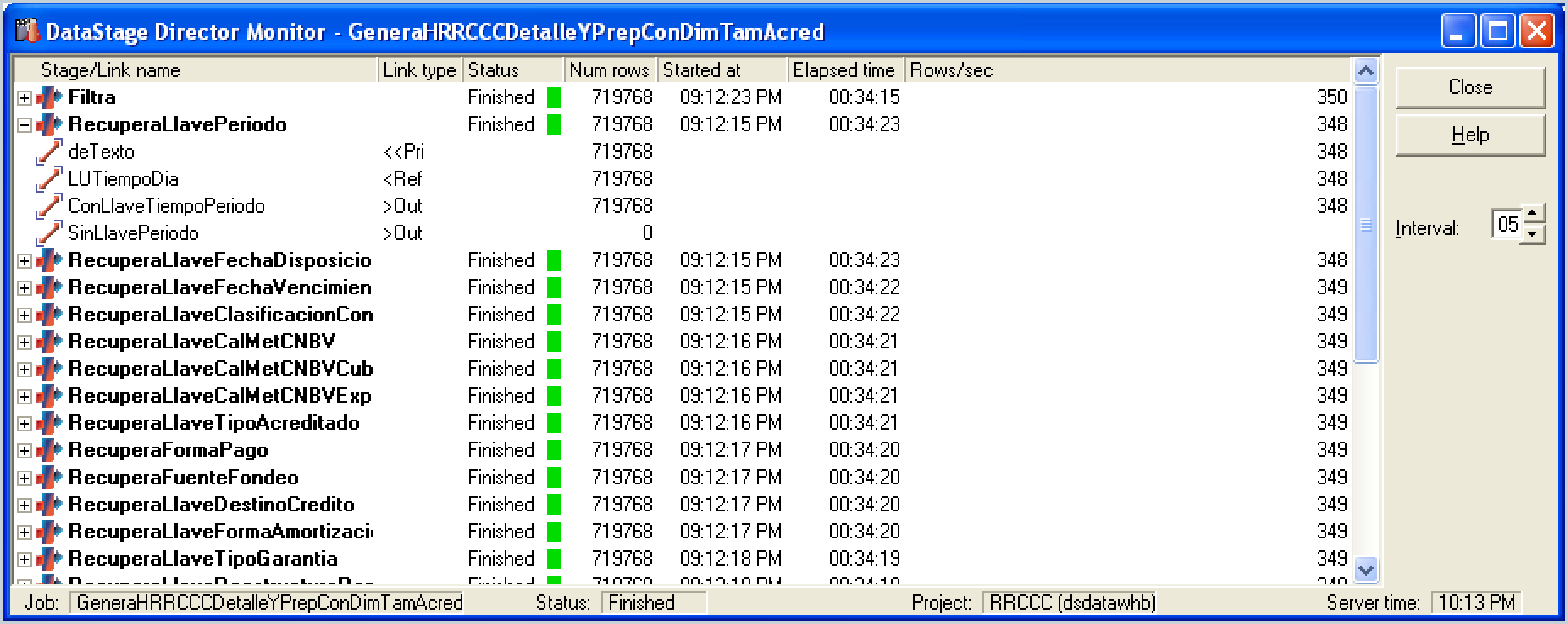
- Jobs
  - AgregaDatos
  - Conavi
  - Curso
  - Dimensiones
  - Ejecutables
  - Hechos
  - ImportaDatosHistoria
  - LiberarDatos
  - OLDEjecutables
  - OracleToSybase
  - OtrosJobs

>Job name	To be run
Batch::PrincipalPorRango1	(Not scheduled)
Batch::PrincipalPorRango2	(Not scheduled)
IDH_ImportaDatosHistoria	(Not scheduled)

Job schedule: Error listing schedule: Access is denied.

Data Warehouse Functionalities

Control: Data Staging Job Control Services: Monitoring



DataStage Director Monitor - GeneraHRRCCCDetalleYPrepConDimTamAcred

Stage/Link name	Link type	Status	Num rows	Started at	Elapsed time	Rows/sec
Filtra		Finished 🟩	719768	09:12:23 PM	00:34:15	350
RecuperaLlavePeriodo		Finished 🟩	719768	09:12:15 PM	00:34:23	348
deTexto	<<Pri		719768			348
LUTiempoDia	<Ref		719768			348
ConLlaveTiempoPeriodo	>Out		719768			348
SinLlavePeriodo	>Out		0			
RecuperaLlaveFechaDisposicio		Finished 🟩	719768	09:12:15 PM	00:34:23	348
RecuperaLlaveFechaVencimien		Finished 🟩	719768	09:12:15 PM	00:34:22	349
RecuperaLlaveClasificacionCon		Finished 🟩	719768	09:12:15 PM	00:34:22	349
RecuperaLlaveCalMetCNBV		Finished 🟩	719768	09:12:16 PM	00:34:21	349
RecuperaLlaveCalMetCNBVCub		Finished 🟩	719768	09:12:16 PM	00:34:21	349
RecuperaLlaveCalMetCNBVExp		Finished 🟩	719768	09:12:16 PM	00:34:21	349
RecuperaLlaveTipoAcreditado		Finished 🟩	719768	09:12:16 PM	00:34:21	349
RecuperaFormaPago		Finished 🟩	719768	09:12:17 PM	00:34:20	349
RecuperaFuenteFondeo		Finished 🟩	719768	09:12:17 PM	00:34:20	349
RecuperaLlaveDestinoCredito		Finished 🟩	719768	09:12:17 PM	00:34:20	349
RecuperaLlaveFormaAmortizaci		Finished 🟩	719768	09:12:17 PM	00:34:20	349
RecuperaLlaveTipoGarantia		Finished 🟩	719768	09:12:18 PM	00:34:19	349

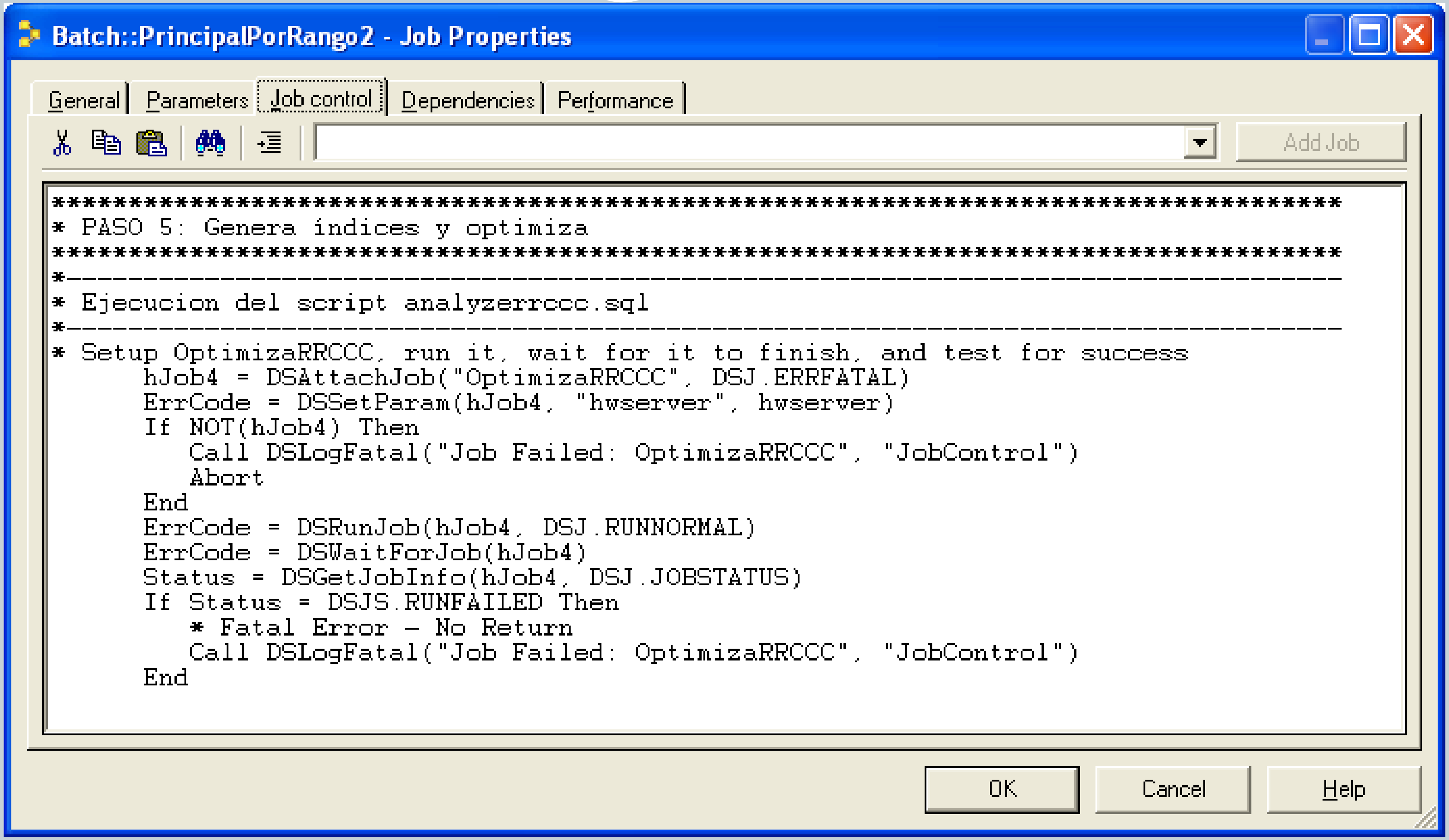
Interval: 05

Job: GeneraHRRCCCDetalleYPrepConDimTamAcred Status: Finished Project: RRCCC (dsdatawhb) Server time: 10:13 PM

Data Warehouse Functionalities

Control: Data Staging Job Control Services:

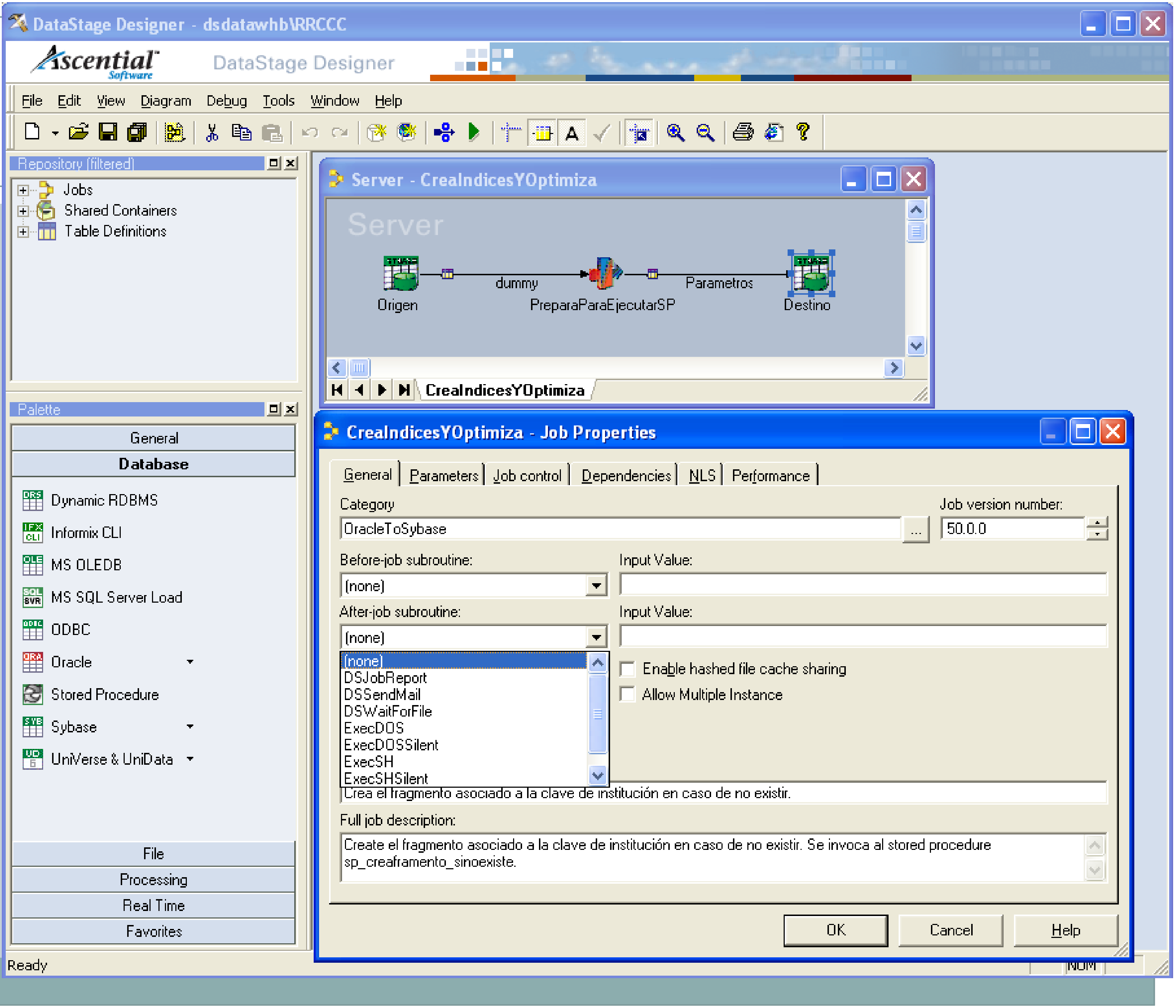
Error handling



```
************************************************************************************
* PASO 5: Genera indices y optimiza
************************************************************************************
*-----------------------------------------------------------------------------------
* Ejecucion del script analyzerrccc.sql
*-----------------------------------------------------------------------------------
* Setup OptimizaRRCCC, run it, wait for it to finish, and test for success
      hJob4 = DSAttachJob("OptimizaRRCCC", DSJ.ERRFATAL)
      ErrCode = DSSetParam(hJob4, "hwserver", hwserver)
      If NOT(hJob4) Then
         Call DSLogFatal("Job Failed: OptimizaRRCCC", "JobControl")
         Abort
      End
      ErrCode = DSRunJob(hJob4, DSJ.RUNNORMAL)
      ErrCode = DSWaitForJob(hJob4)
      Status = DSGetJobInfo(hJob4, DSJ.JOBSTATUS)
      If Status = DSJS.RUNFAILED Then
         * Fatal Error - No Return
         Call DSLogFatal("Job Failed: OptimizaRRCCC", "JobControl")
      End
```

Data Warehouse Functionalities

Control: Data Staging Job Control Services: Notification



DataStage Designer - dsdatawhb\RRCCC

Server - CreaIndicesYOptimiza

CreaIndicesYOptimiza - Job Properties

Category: OracleToSybase

Job version number: 50.0.0

Before-job subroutine: (none)

Input Value:

After-job subroutine: (none)

Input Value:

□ Enable hashed file cache sharing

□ Allow Multiple Instance

Crea el fragmento asociado a la clave de institución en caso de no existir.

Full job description:

Create el fragmento asociado a la clave de institución en caso de no existir. Se invoca al stored procedure sp_creaframento_sinoexiste.

Data Warehouse Functionalities

Control: Back Room Asset Management

- Backup and Recovery
  - High performance
  - Simple administration
- Archive and Retrieval
- Backup and Archive Planning
  - Determine an appropiate backup process
  - Implement the process
  - Practice
- Extract and Load Security Issues
- Future Staging Services
  - Transaction Processing Support
  - Active Source System Participation
  - Data Push
  - Object-Oriented Systems

Data Warehouse Functionalities

Data Quality and Cleansing: Data Improvement: common problems

- Inconsistent or incorrect uses or codes and special characters (gender field: “M”, “F”, “m”, “f”, “y”, “n”, “u” and blank)
- A single field is used for unofficial or undocumented purposes
- Overloaded codes
- Evolving data
- Missing, incorrect, or duplicates values

Data Warehouse Functionalities

Data Quality and Cleansing: Data Improvement: an approach to improving the data

- Where there are alternatives, identify the highest quality source system: the organization’s system of record
- Examine the source to see how bad it is

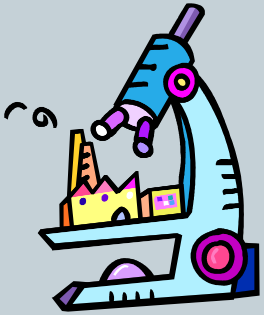
```
Select my_attribute, count(*) from source_table
Group by my_attribute order by 1
```

- Upon scanning this list, you will inmediately find minor variations in spelling
- Raise problems with the steering commitee
- Fix problems at the source if at all possible
- Fix some problems during data staging
- Don’t fix all the problems
- Use data cleansing tools against the data, and use trusted source for correct values like address
- Work with the source system owners to help them institute regular examination and cleansing of the source systems
- If it’s politically feasible, make the source systems team responsible for a clean extract

Data Warehouse Functionalities

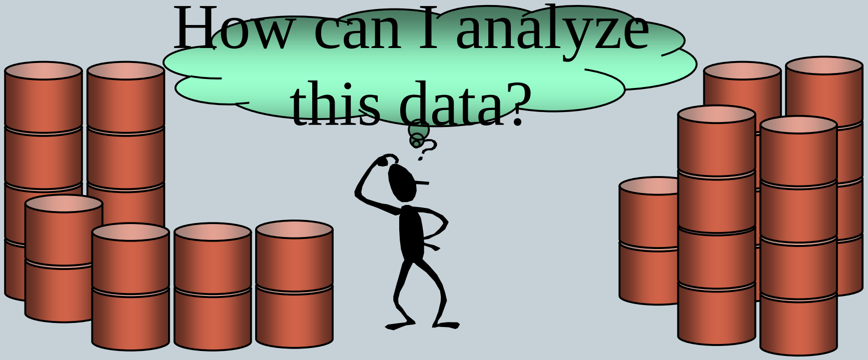
Data Quality and Cleansing: Data Quality Assurance

- Is the data you are about to load correct?
- The basic data staging audit information tells us we have the right number of rows, and referential integrity checking tells us everything matches up; but how do we know if the contents are right?



Introduction to Data Mining

- Objetivo: extraer información oculta o analizar datos mediante técnicas estadísticas
- Fuentes de información: datos de la empresa
- Responder a preguntas
  - empresariales a priori no planteadas
  - consumidoras de tiempo para ser resueltas
- Apoyo para la toma de decisiones de la alta dirección
- Técnicas
  - Agrupamiento (clustering)
  - Redes neuronales
  - Árboles de decisión
  - Reglas de asociación
  - …



Introduction to Data Mining

ejemplos

- Negocios
  - Hábitos de compra en supermercados
  - Patrones de fuga
  - Fraudes
  - Recursos humanos
- Comportamiento en internet
- Terrorismo
- Juegos
- Ciencia e ingeniería
  - Genética
  - Ingeniería eléctrica
  - Análisis de gases

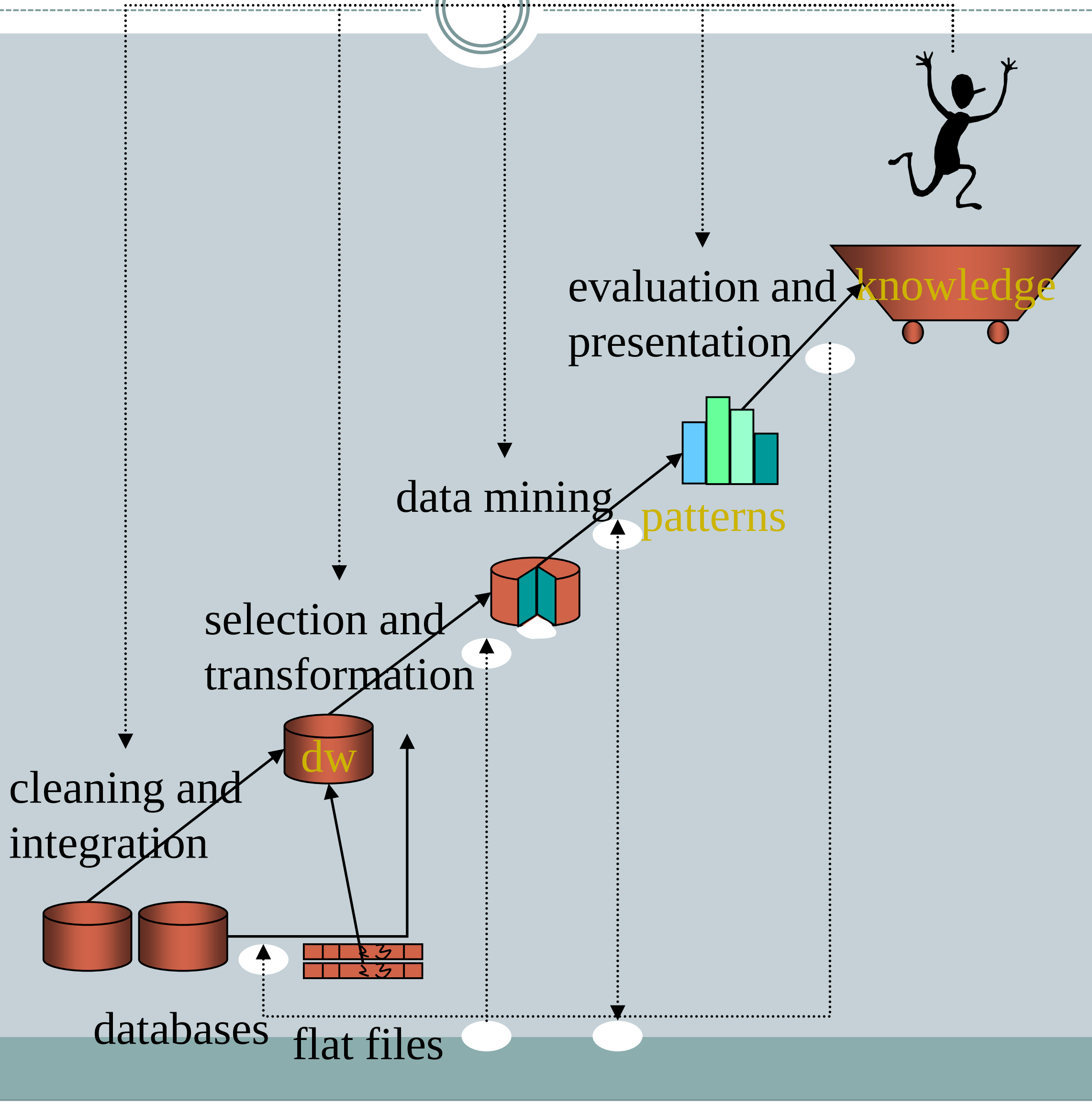
Introduction to Data Mining

data mining – on what kind of data?

- relational database
- data warehouses
- transactional databases
- advanced databases systems (object-oriented and object-relational databases, spatial databases, time-series databases, text databases, multimedia databases)
- flat files
- world wide web

Introduction to Data Mining

data mining as a step in the process of knowledge discovery



Introduction to Data Mining

from data warehousing to data mining

Data warehouse usage

- Initially
  - Generating reports
  - Answering predefined queries
- Progressively
  - analyze summarized and detailed data
- Later
  - Strategic purposes
  - Performing multidimensional analysis and sophisticated slice-and-dice operations
- Finally
  - Knowledge discovery and strategic decision making using data mining

Classified tools for data warehousing

- Access and retrieval tools
- Database reporting tools
- Data analysis tools
- Data mining tools

Introduction to Data Mining

from data warehousing to data mining: kinds of data warehouse aplications

- Information processing
- Analytical processing
- Data mining

Conceptos básicos de minería de datos

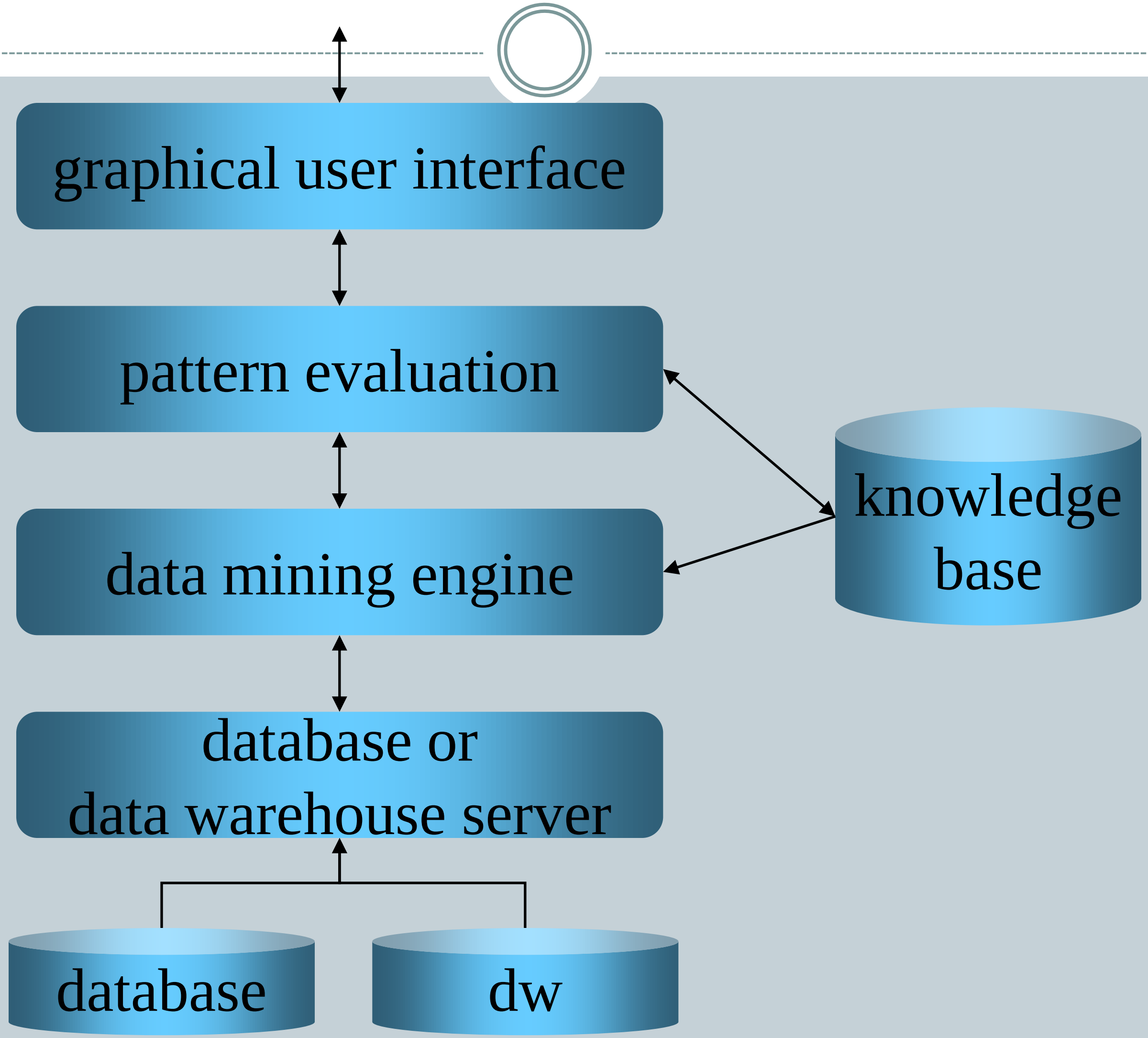
from OLAP to OLAM

On-Line Analytical Mining (OLAM)

- integrates OLAP with data mining and mining knowledge in multimensional databases

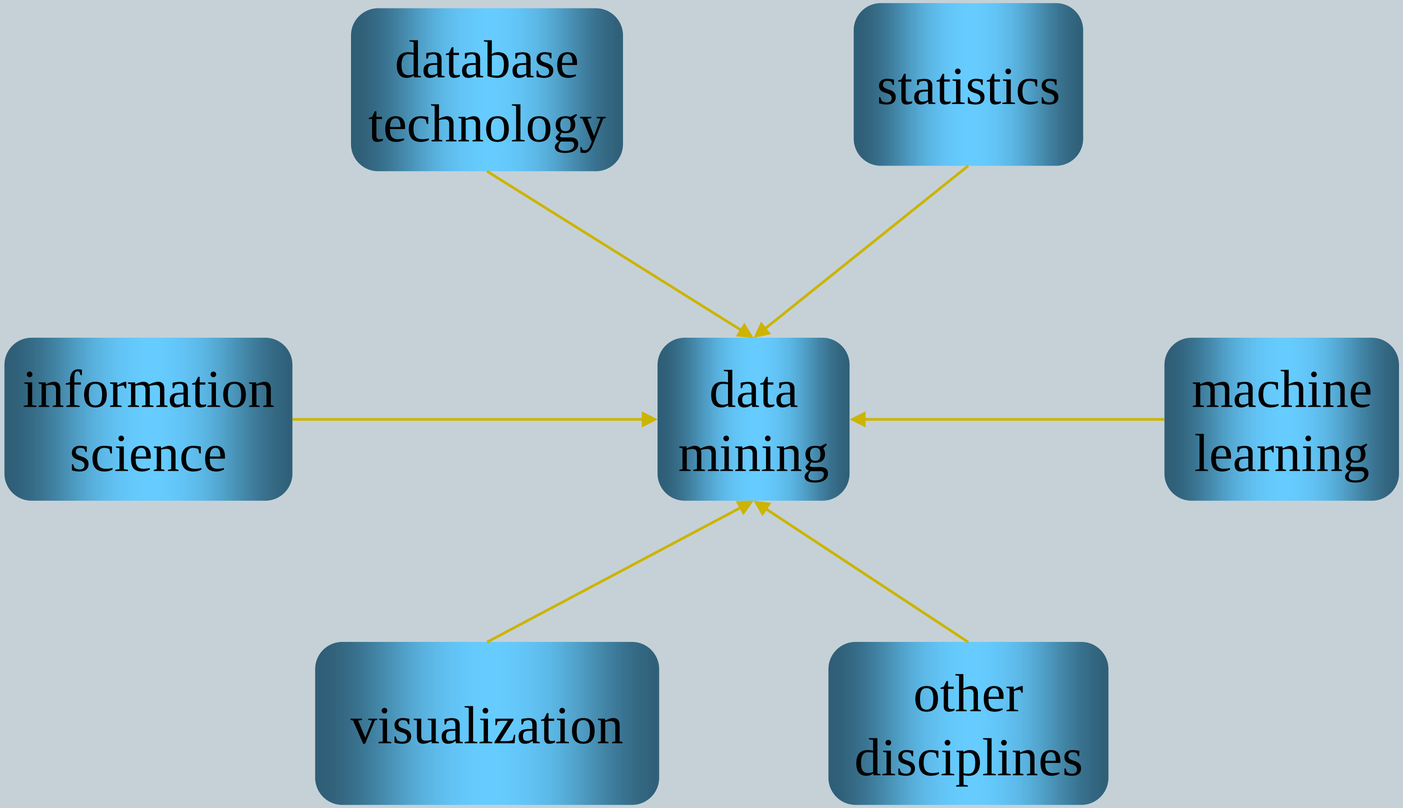
Introduction to Data Mining

architecture of a typical data mining system



Introduction to Data Mining

data mining as confluence of multiple disciplines



database technology

statistics

information science

data mining

machine learning

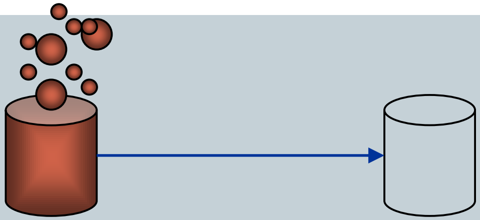
visualization

other disciplines

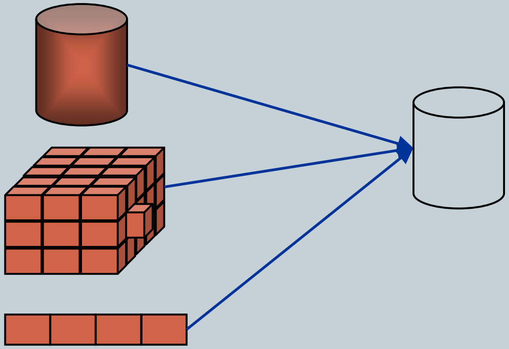
Preprocesamiento de datos

data preprocessing techniques

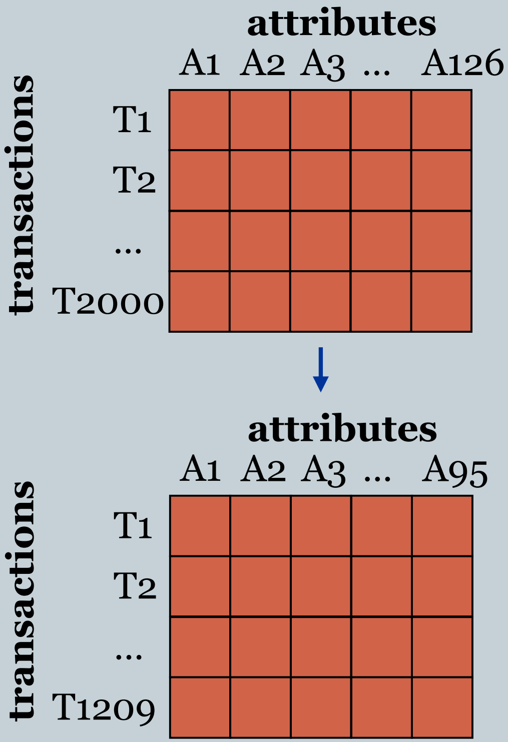
Data cleaning



Data integration



Data reduction



Data transformation

-2,32,100,59,48 → -0.02, 0.32, 1.00, 0.59, 0.48

Preprocesamiento de datos

data preprocessing techniques: Data cleaning

Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data

Preprocesamiento de datos

data preprocessing techniques: Data cleaning: Missing values

1. Ignore the tuple
2. Fill in the missing value manually
3. Use a global constant to fill in the missing value
4. Use the attribute mean to fill in the missing value
5. Use the attribute mean for all samples belonging to the same class as the given tuple
6. Use the most probable value to fill in the missing value

Methods 3 to 6 bias the data

Preprocesamiento de datos

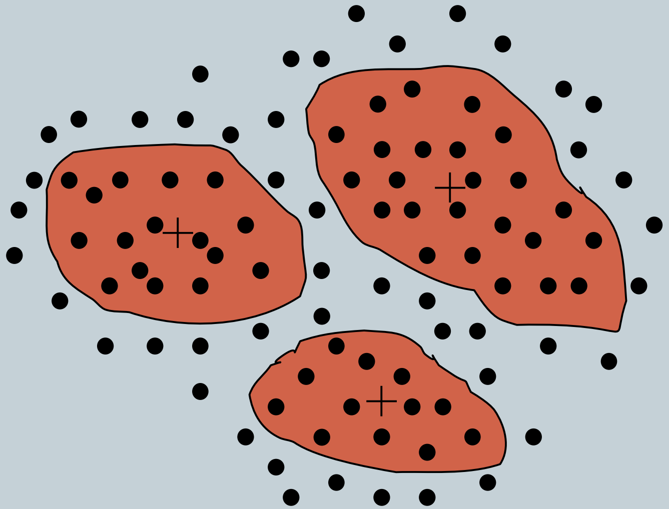
data preprocessing techniques: Data cleaning: Noisy data: Binning

- Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34
- Partition into (equidepth) bins:
  - Bin 1: 4, 8, 15
  - Bin 2: 21, 21, 24
  - Bin 3: 25, 28, 34
- Smoothing by bin means:
  - Bin 1: 9, 9, 9
  - Bin 2: 22, 22, 22
  - Bin 3: 29, 29, 29
- Smoothing by bin boundaries:
  - Bin 1: 4, 4, 15
  - Bin 2: 21, 21, 24
  - Bin 3: 25, 25, 34

Preprocesamiento de datos

data preprocessing techniques: Data cleaning: Noisy data: Clustering

- Outliers may be detected by clustering, where similar values are organized into groups, or "clusters."
- Intuitively, values that fall outside of the set of clusters may be considered outliers



Preprocesamiento de datos

data preprocessing techniques: Data cleaning: Noisy data: Regression

- Data can be smoothed by fitting the data to a function, such as with regression
- *Linear regression* involves finding the "best" line to fit two variables, so that one variable can be used to predict the other
- *Multiple linear regression* is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface

Preprocesamiento de datos

data preprocessing techniques: Data transformation

- The data are transformed or consolidated into form appropiate for mining

- Techniques
  - ○ *Smoothing*. for removing the noise from data
  - ○ *Aggregation*. summary or aggregation operations applied to the data
  - ○ *Generalization of data*. concept hierarchies
  - ○ *Normalization*. the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0 or 0.0 to 1.0
  - ○ *Attribute construction* (or *feature construction*). new attributes are constructed and added from the given set of attributes to help the mining process

Preprocesamiento de datos

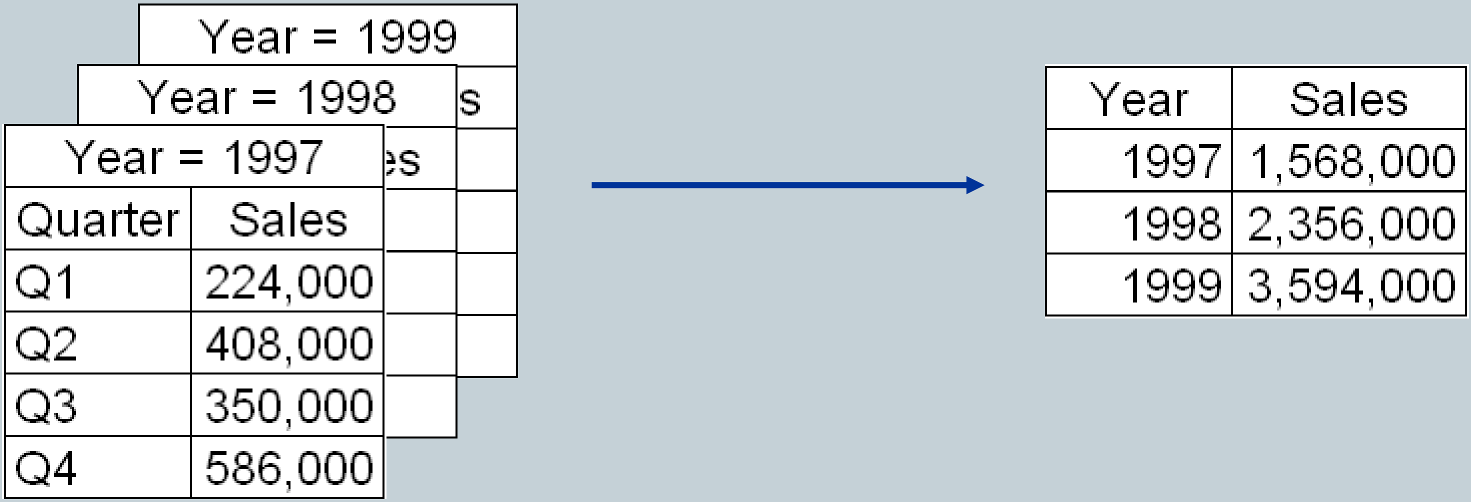
data preprocessing techniques: Data reduction techniques

- Can be applied to obtain a reduced representation of data set that is much smaller in volume, yet closely maintains the integrity of the original data
- That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results
- Strategies
  - Data cube aggregation
  - Dimension reduction
  - Data compression
  - Numerosity reduction
  - Discretization and concept hierarchy generation

Preprocesamiento de datos

data preprocessing techniques: Data reduction techniques: data cube aggregation

where aggregation operations are applied to the data in the construction of a data cube



Year = 1999

Year = 1998

Year = 1997

Quarter	Sales
Q1	224,000
Q2	408,000
Q3	350,000
Q4	586,000

Year	Sales
1997	1,568,000
1998	2,356,000
1999	3,594,000

Preprocesamiento de datos

data preprocessing techniques: Data reduction techniques: Dimensionaly reduction: basic heuristic methods

Forward selection

Initial attribute set:
{A1, A2, A3, A4, A5, A6}

Initial reduced set:
{}
→ {A1}
→ {A1, A4}
→ Reduced attribute set:
{A1, A4, A6}

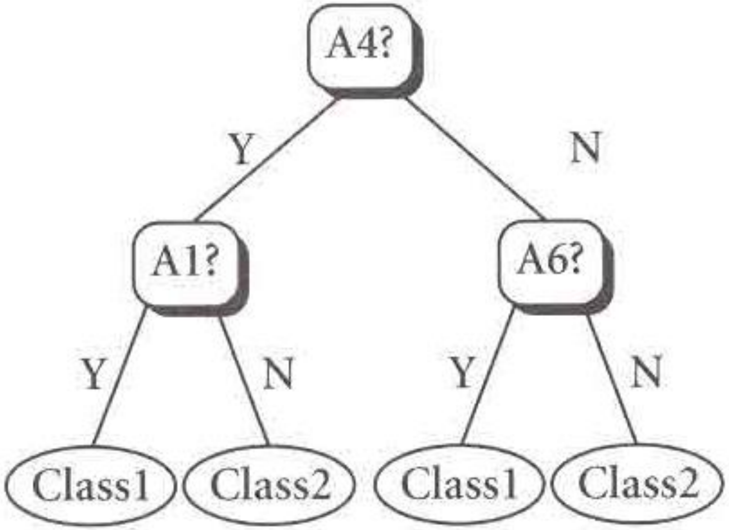
Backward elimination

Initial attribute set:
{A1, A2, A3, A4, A5, A6}

→ {A1, A3, A4, A5, A6}
→ {A1, A4, A5, A6}
→ Reduced attribute set:
{A1, A4, A6}

Decision tree induction

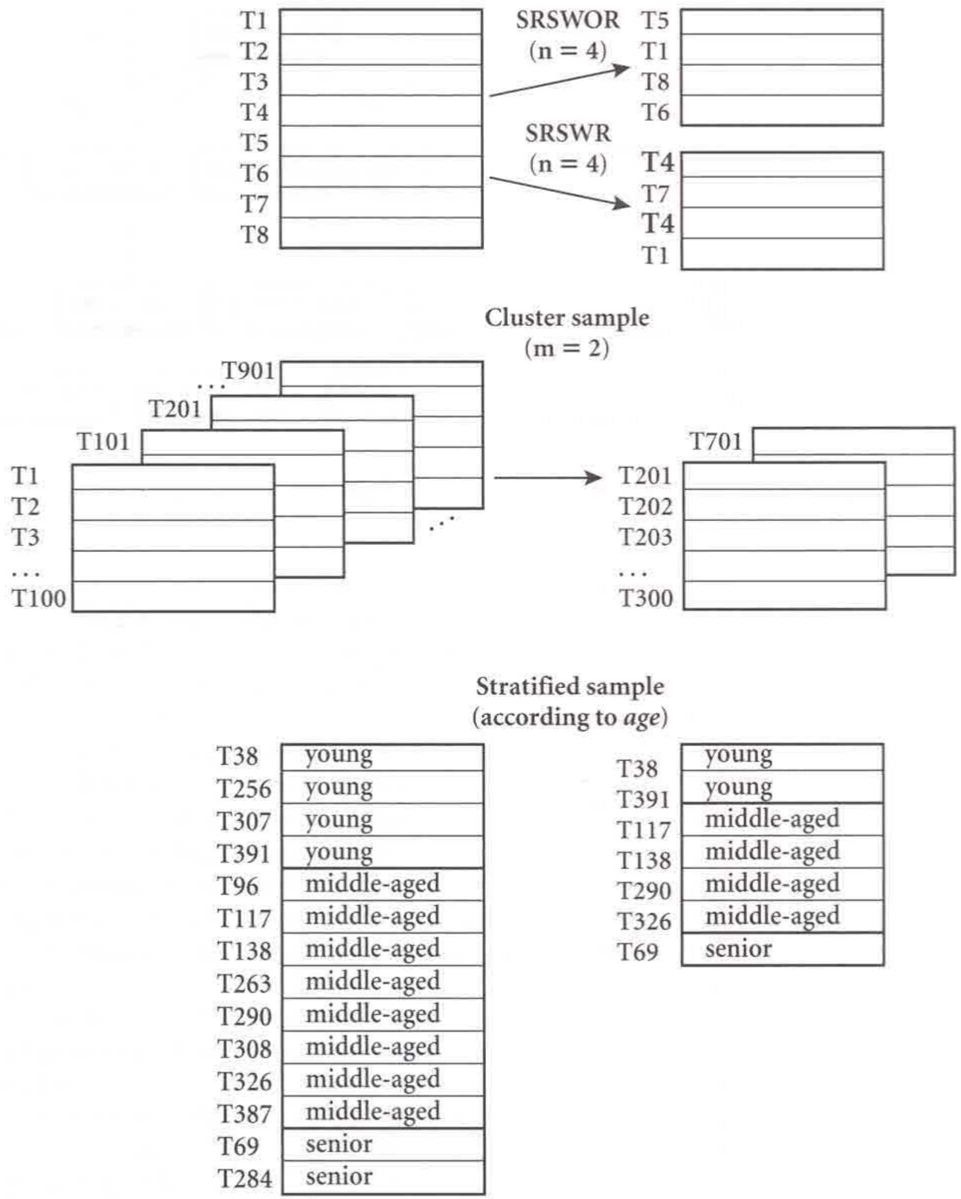
Initial attribute set:
{A1, A2, A3, A4, A5, A6}



→ Reduced attribute set:
{A1, A4, A6}

Preprocesamiento de datos

data preprocessing techniques: Data reduction techniques: Numerosity reduction: sampling



Stratified sample
(according to *age*)

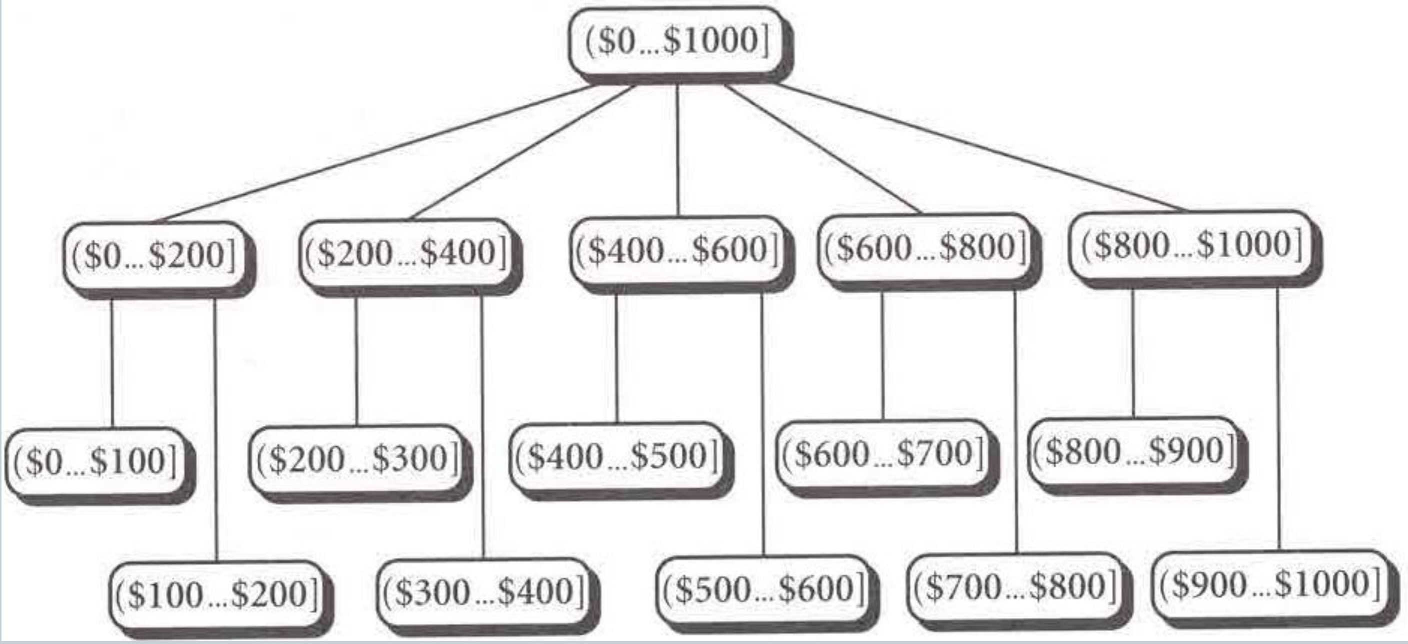
T38	young
T256	young
T307	young
T391	young
T96	middle-aged
T117	middle-aged
T138	middle-aged
T263	middle-aged
T290	middle-aged
T308	middle-aged
T326	middle-aged
T387	middle-aged
T69	senior
T284	senior

T38	young
T391	young
T117	middle-aged
T138	middle-aged
T290	middle-aged
T326	middle-aged
T69	senior

Preprocesamiento de datos

data preprocessing techniques: Data reduction techniques: Discretization and Concept hierarchy generation

- raw data values for attributes are replaced by ranges or higher conceptual levels
- Discretization techniques can be used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals



Preprocesamiento de datos

data preprocessing techniques: Data reduction techniques: Discretization and Concept hierarchy generation: 3-4-5 rule

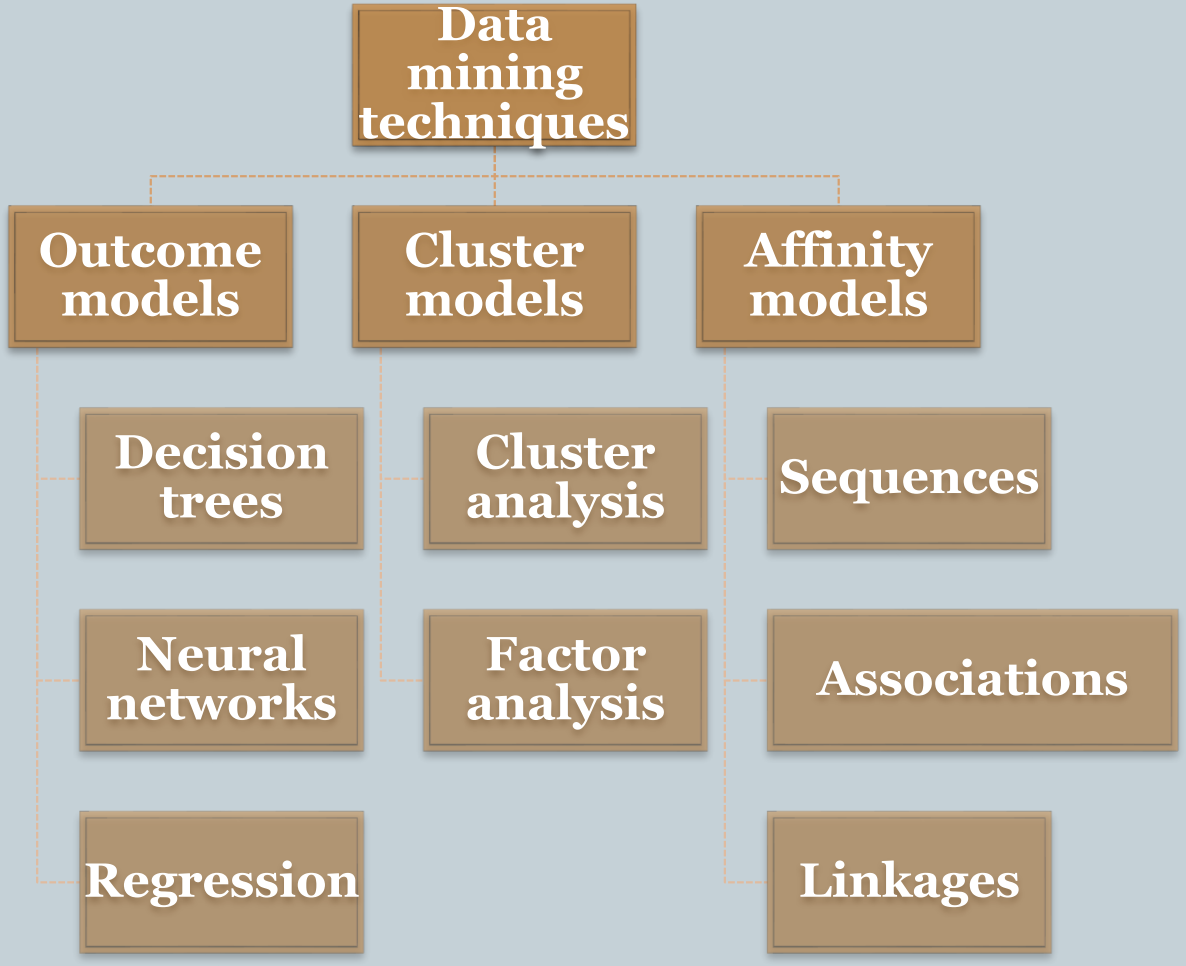
- Used to segment numeric data into relatively uniform “natural” intervals

  ($51,263.98, $60,872.34) -> ($50,000, $60,000]

- The rule partitions a given range of data into 3, 4,or 5 relatively equiwidth intervals, recursively and level by level, based on the value range at he most significant digit

Técnicas de minería de datos

types of data mining techniques



Data mining techniques

Outcome models

Cluster models

Affinity models

Decision trees

Neural networks

Regression

Cluster analysis

Factor analysis

Sequences

Associations

Linkages

Técnicas de minería de datos

- Análisis preliminar de datos usando query tools
  - Aplicación de una consulta SQL para rescatar algunos aspectos visibles antes de aplicar las técnicas
- Técnicas de visualización
  - Aptas para ubicar patrones en un conjunto de datos
- Redes neuronales artificiales
  - Modelos predecibles, no lineales que aprenden a través de entrenamiento
- Reglas de asociación
  - Establecimiento de asociaciones en base a perfiles de los clientes

Técnicas de minería de datos

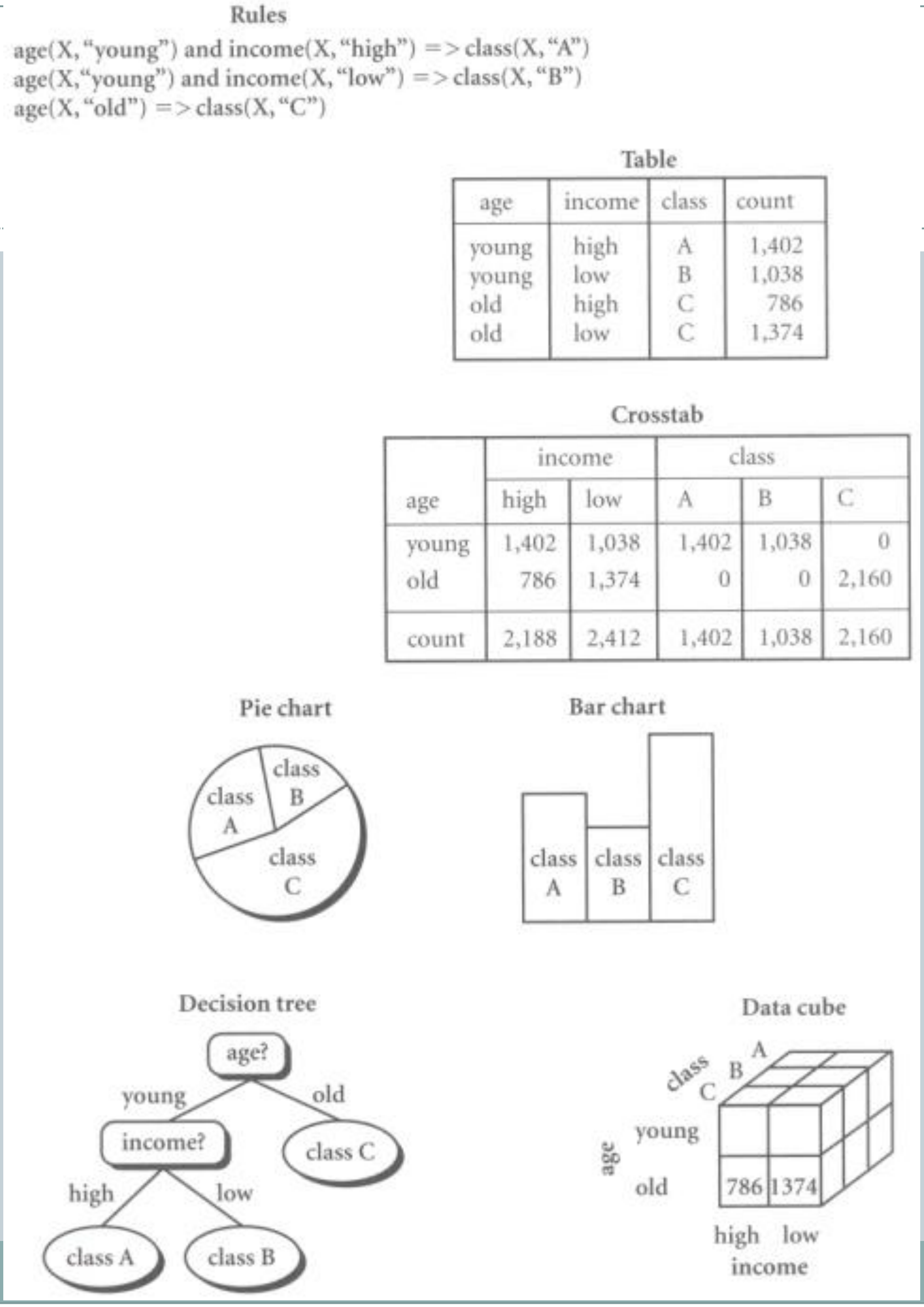
- Algoritmos genéticos

  Técnicas de optimización que usan procesos tales como combinaciones genéticas, mutaciones, etc.

- Redes bayesianas
  - Determinación de relaciones causales que expliquen un fenómeno según los datos contenidos en la base de datos
  - Usadas principalmente para realizar predicciones
- Árboles de decisión
  - Estructuras que representan conjuntos de decisiones
  - Generan reglas para la clasificación de los datos

Técnicas de minería de datos

forms of presenting and visualizing the discovered patterns



Rules

age(X, "young") and income(X, "high") => class(X, "A")
age(X,"young") and income(X, "low") => class(X, "B")
age(X, "old") => class(X, "C")

Table

age	income	class	count
young	high	A	1,402
young	low	B	1,038
old	high	C	786
old	low	C	1,374

Crosstab

	income		class		
age	high	low	A	B	C
young	1,402	1,038	1,402	1,038	0
old	786	1,374	0	0	2,160
count	2,188	2,412	1,402	1,038	2,160

Pie chart

Bar chart

Decision tree

Data cube

Técnicas de minería de datos

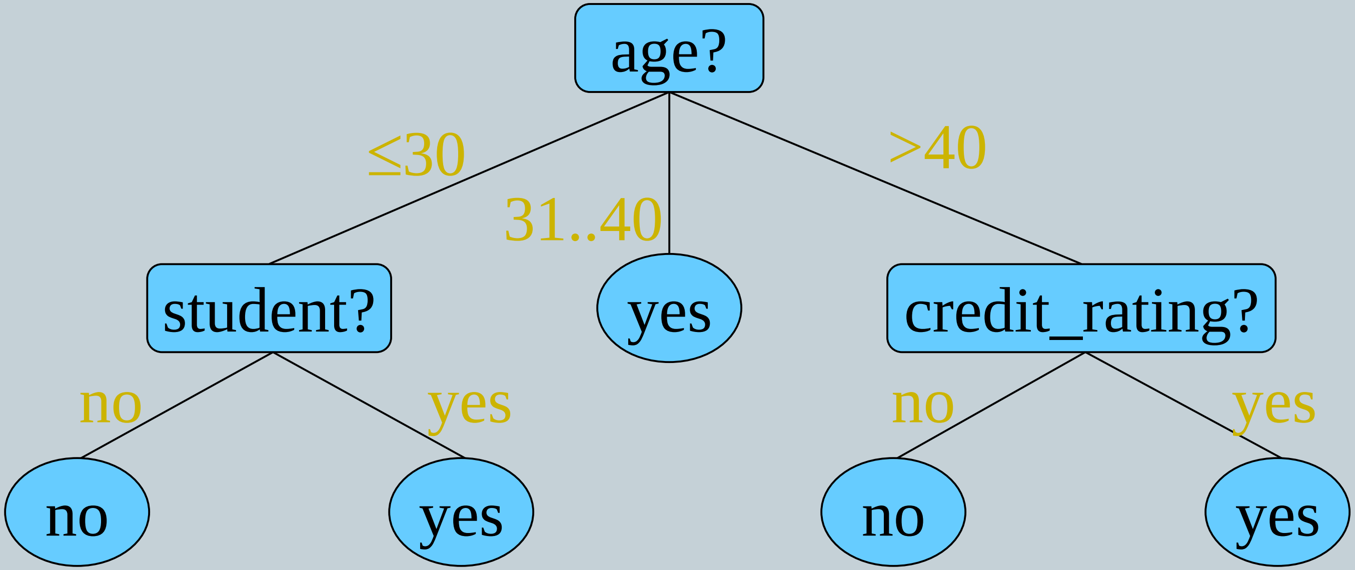
kinds of patterns

- concept/class description: characterization and discrimination
- association analysis
- classification and prediction
- cluster analysis
- outlier analysis
- evolution analysis

Técnicas de minería de datos

kinds of patterns: classification and predictions:decision trees

decision tree: flow-chart like tree structure, where each node denotes a test on an attribute value, each branch represent an outcome of the test, and tree leaves represent class of class distributions. Decision trees can be easily converted to classification rules



Técnicas de minería de datos

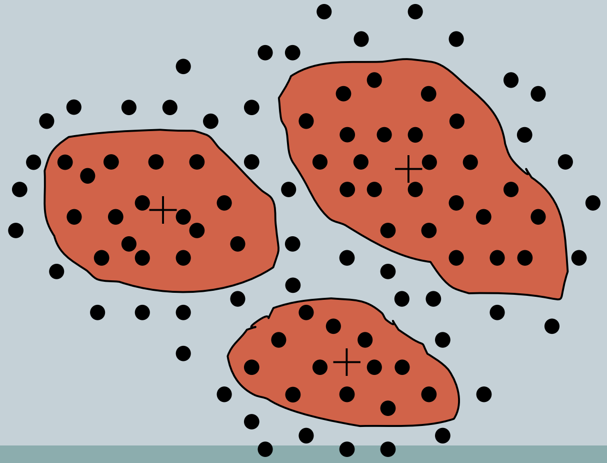
kinds of patterns: classification and predictions: neural networks

- ***neural networks***: used for classification, is typically a collection of neuron-like processing units with weighted connections between units
- classification and predictions may need to be preceed by ***relevance analysis***, which attempts to identify attributes that do not contribute to the classification or prediction process. These attributes can then be excluded
- example
  - sales manager
  - kinds of response: good response, bad response, no response
  - descriptive features of the items: price, brand, place_made, type, category
  - goal: derive a model for each of the three classes
  - the resulting decision tree may help to understand the impact of the given sales campaign and design a more effective campaign for the future

Técnicas de minería de datos

kinds of patterns: cluster analysis

- clustering can also facilitate taxonomy formation
- example
  - cluster analysis can be perfomed on All Electronics customer data in order to identify homogeneous subpopulations of customers
  - these clusters represent individual target groups for marketing
  - A 2-D plot of customer data with respect to customer locations in a city



Técnicas de minería de datos

kinds of patterns: evolution analysis

- describes and models regularities or trends for objects whose behavior changes over time
- although this may include characterization, discrimination, association, classification, or clustering of time-related, distinct features of such an analysis include time-series data analysis, sequence or periodicity pattern matching, and similarity-based data analysis
- example
  - major stock market (time-series) data of the last several years available
  - wishes to invest in shares of high-tech industrial companies
  - a data mining study to stock exchange data may identify stock evolution regularities for overall stocks and for the stocks of particular companies
  - such regularities may help predict future trends in stock market prices, contributing to the decision making regarding stock investments

Técnicas de minería de datos

kind of patterns: are all of the patterns interesting?

- a data mining system has the potential to generate thousands or even millions of patterns, or rules
- only a small fraction of the patterns potentially generated would actually be of interest to any given user
- a pattern is interesting if
  - it is easily understood by humans
  - valid on new or test data with some degree of certainty
  - potentially useful
  - novel
  - it validates a hypothesis that the user sought to confirm
- an interesting pattern represents *knowledge*

Técnicas de minería de datos

kind of patterns: what makes a pattern interesting?

- objective measures of pattern interestingness
  - rule support (X⇒Y)
    - represents the percentage of transactions from a transaction database that the given rule satisfies
    - P(X∪Y) where X∪Y indicates that a transaction contains both X and Y
  - rule confidence (X⇒Y)
    - assesses the degree of certainty of the detected association
    - P(Y|X), the probability that a transaction containing X also contains Y
- subjective measures of pattern interestingness
  - based on user beliefs in the data
  - find patterns interesting if they are unexpected or offer strategic information on which the user can act (*actionable patterns*)
  - *expected patterns* can be interesting if they confirm a hypothesis that the user wished to validated

Técnicas de minería de datos

the classification process: learning

training data

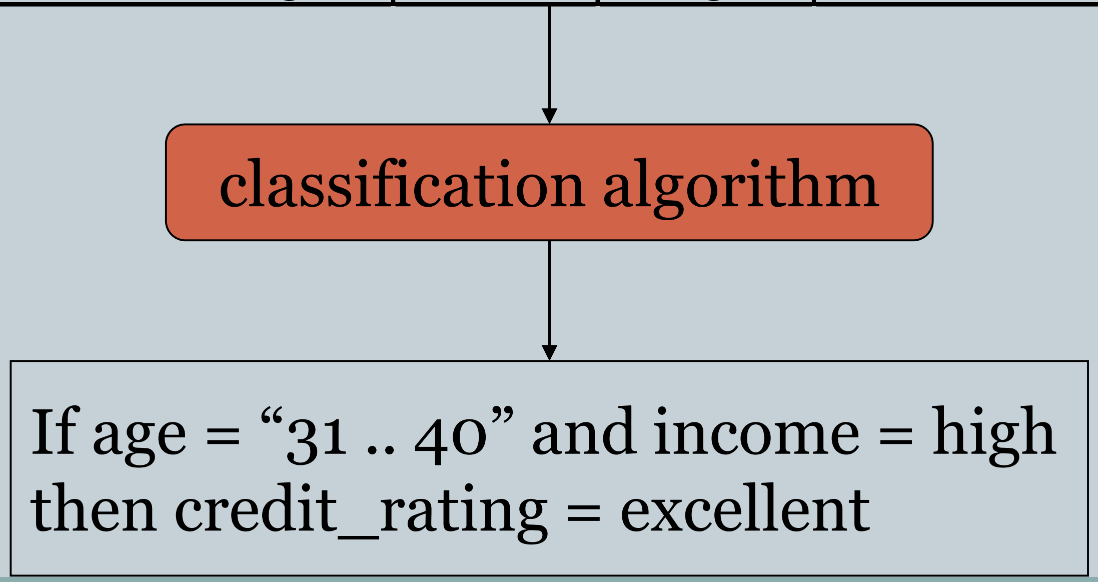
name	age	income	credit_rating
Sandy Jones	<= 30	low	fair
Bill Lee	<= 30	low	excellent
Courtney Cox	31 .. 40	high	excellent
Susan Sarandon	> 40	med	fair
Claire Chazal	> 40	med	fair
Renée Beauregard	31 .. 40	high	excellent

class label attribute

predefined classes

samples, examples, objects

supervised learning



classification algorithm

If age = "31 .. 40" and income = high then credit_rating = excellent

classification rules

Técnicas de minería de datos

the classification process

test data

name	age	income	credit_rating
Franck Silvestre	> 40	high	fair
Cathy Roubineau	<= 30	low	fair
Yanick Noah	31 .. 40	high	excellent

classifier accuracy:
holdout method, ...

classification rules

accuracy of a model

prediction?

new data

(Sandra Bulock, 31 .. 40, high)
credit_rating?→ excellent

classification and regression: typical prediction problems

Técnicas de minería de datos

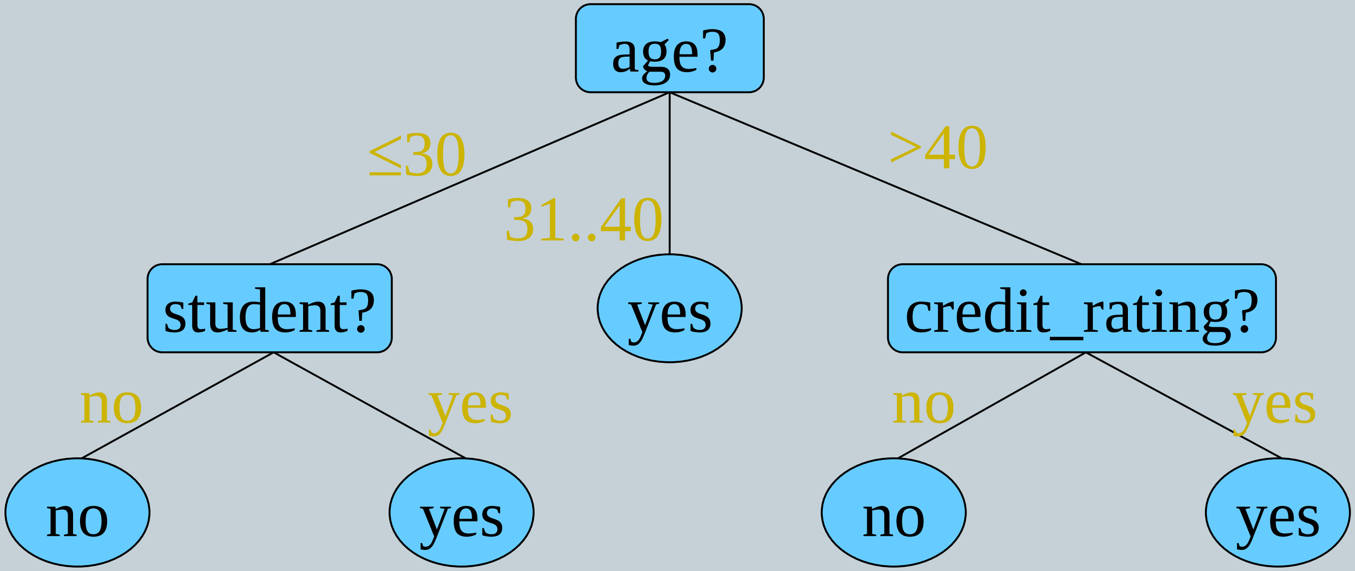
classification and prediction: examples

- database of customers: name, age, income, occupation, and credit rating
- mailing list used to send out promotional literature: new products and upcoming price discounts
- customer classification: whether or not they have purchased a computer
- supposition: new customers are added to the database
- goal: notification of only those new customers (whose are likely to purchase a new computer) of an upcoming compute sale

Clasificación por árboles de decisión

issues regarding classification and prediction

decision tree



Clasificación por árboles de decisión

issues regarding classification and prediction

- *attribute selection measure*: a heuristic for selecting the attribute tha will best separate the samples into individual classes (*information gain*, *measure of the goodness of split*)
- the attribute with the highest information gain (or greatest *entropy* reduction) is chosen as the test attribute for the current node

expected informatio n need to classify a given sample

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where

S : set of s data samples

C_i : class i (for $i = 1, \ldots, m$)

p_i : probability that an arbitrary sample belongs to class C_i and

is equal to s_i / s

entropy

$$E(A)=\sum_{j=1}^{v}\frac{s_{1j}+\ldots+s_{mj}}{s}I\left(s_{1j},\ldots,s_{mj}\right)$$

where

A : attribute

a_i : value of A, $(i=1,\ldots,v)$

S_i : partition, $(i=1,\ldots,v)$

s_{ij} : the number of samples of class C_i in Sj

$$I\left(s_{1j},s_{2j},\ldots,s_{mj}\right)=-\sum_{i=1}^{m}p_{ij}\log_2\left(p_{ij}\right)$$

where

$p_{ij}=s_{ij}\big/\left|s_j\right|$, probability that a sample in S_j belongs to C_i

enconding informatio n

$$Gain(A)=I\left(s_1,s_2,\ldots,s_m\right)-E\left(A\right)$$

Clasificación por árboles de decisión

issues regarding classification and prediction

RID	age	income	student	credit_rating	class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31..40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31..40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31..40	medium	no	excellent	yes
13	31..40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Clasificación por árboles de decisión

issues regarding classification and prediction

class label attribute: *buys_computer* = {*yes*, *no*} ⇒ *m* = 2

- C_1=yes, C_2=no;
- 9 samples for C_1 and 5 samples for class C_2⇒ s_1=9, s_2=5
- expected information

$$I(s_1, s_2) = I(9{,}5) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Clasificación por árboles de decisión

issues regarding classification and prediction

entropy for each attribute

for $age = "<= 30"$:

$s_{11} = 2 \qquad s_{21} = 3 \qquad I(s_{11}, s_{21}) = 0.971$

for $age = "31..40"$:

$s_{12} = 4 \qquad s_{22} = 0 \qquad I(s_{12}, s_{22}) = 0$

for $age = "> 40"$:

$s_{13} = 3 \qquad s_{23} = 2 \qquad I(s_{13}, s_{23}) = 0.971$

$$E(age) = \frac{5}{14} I(s_{11}, s_{21}) + \frac{4}{14} I(s_{12}, s_{22}) + \frac{5}{14} (s_{13}, s_{23}) = 0.694$$

$$Gain(age) = I(s_1, s_2) - E(age) = 0.246$$

$Gain(income) = 0.029$

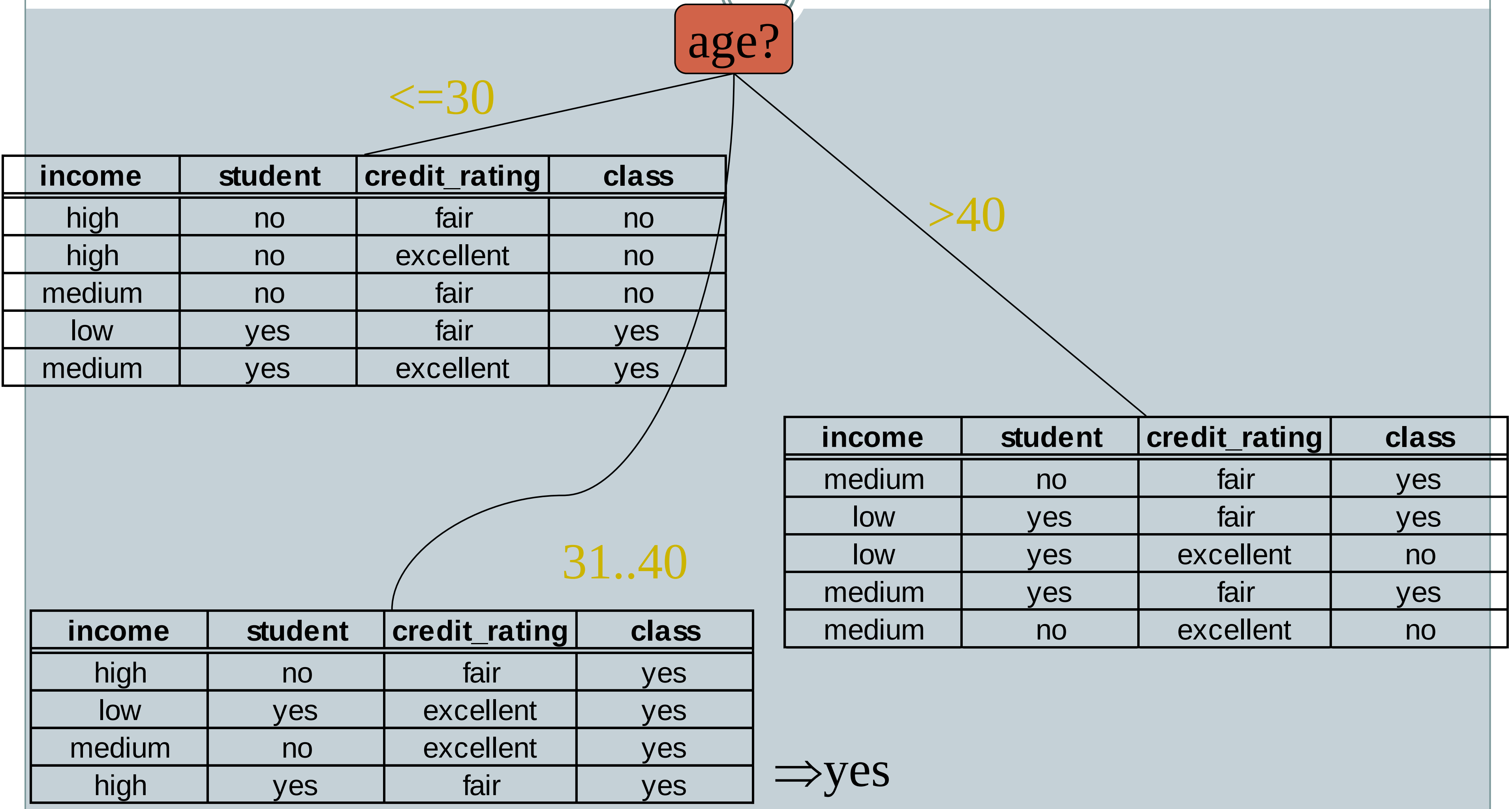
$Gain(student) = 0.151$

$Gain(credit\_rating) = 0.048$

which attribute is selected as the test attribute?

Clasificación por árboles de decisión

issues regarding classification and prediction



age?

<=30

income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

31..40

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

⇒yes

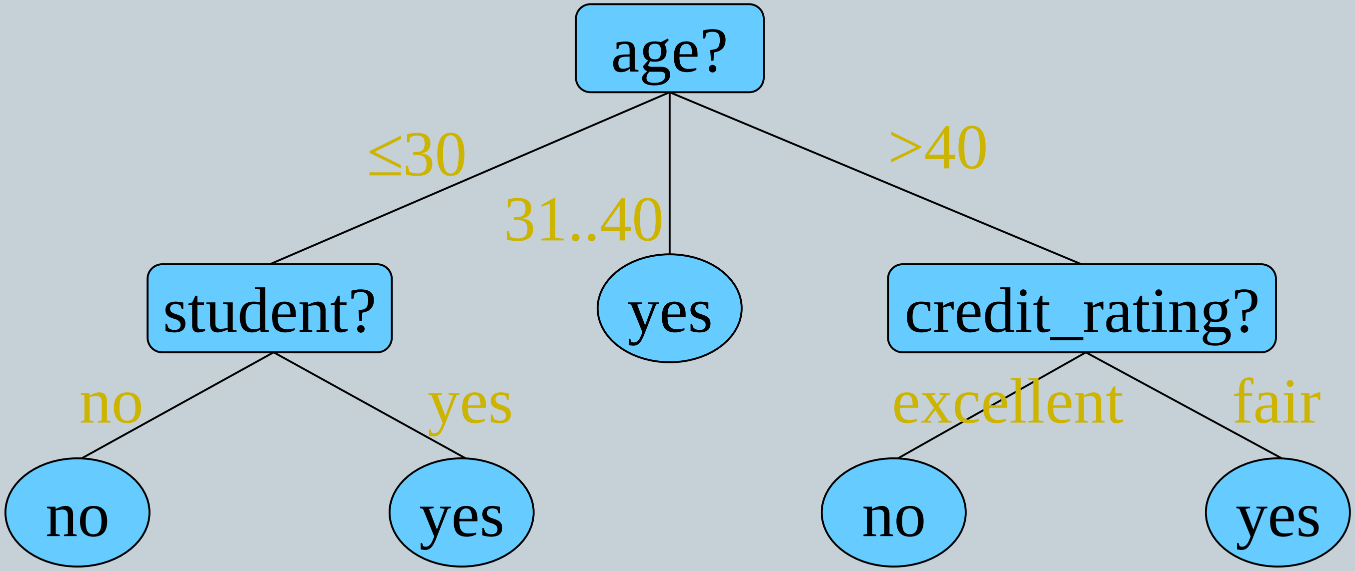
>40

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Clasificación por árboles de decisión

issues regarding classification and prediction

final decision tree



Clasificación por árboles de decisión

issues regarding classification and prediction

- ***tree pruning***: when a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers
- ***prepruning***
  - a tree is "pruned" by halting its construction early (e.g. by deciding not to further split or partition the subset of training samples a give mode)
  - upon halting, the node becomes a leaf
  - the leaf may hold the most frequent class among the subset samples or the probability distribution of those samples
- ***postpruning***
  - removes branches from "fully grown" tree
  - a tree node is pruned by removing its branches
  - the *cost complexity* pruning algorithm is an example of the postpruning approach

Clasificación por árboles de decisión

issues regarding classification and prediction

Example: extracting classification rules from decision trees

```
IF age=“<=30” AND student=“no” THEN buys_computer=“no”

IF age=“<=30” AND student=“yes”THEN
  buys_computer=“yes”

IF age=“31..40” THEN buys_computer=“yes”

IF age=“>40” AND credit_rating=“excellent”
  THEN buys_computer=“no”

IF age=“>40” AND credit_rating=“fair”
  THEN buys_computer=“yes”
```

Classification and prediction

bayesian classification

- bayesian classifiers are statistical classifiers
- they can predict class membership probabilities, such as the probability that a given sample belongs to a particular class
- based on Bayes Theorem
- simple Bayesian classifier = naive bayesian classifier comparable in performance with decision tree and neural network classifiers

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

where

X : data sample whose class label is unknown

H : some hypothesis such as that the data sample X belongs to a specified class C

$P(H|X)$: the probability that the hypothesis H holds given the observed data sample X,

posterior probability of H conditioned on X

$P(H)$: a *priori probability*

Classification and prediction

bayesian classification: Bayes Theorem example

- world of data samples: fruits described by their color and shape
- suppose that X is red and round, and that H is the hypothesis that X is apple
- $P(H|X)$ reflects the confidence that X is an apple given that we have seen that X is red and round
- $P(H)$ is the probability that any given data sample is an apple, regardless of how the data sample looks
- $P(X|H)$ is the posterior probability of X conditioned on H; it is the probability that X is red and round given that we know that it is true that X is an apple.

Classification and prediction

naive (or simple) bayesian classification

$X = (x_1, x_2, \ldots, x_n)$: n-dimensional feature vector

n measurements

$A_1, A_2, \ldots, A_n$: n attributes

$C_1, C_2, \ldots, C_m$: m classes

The classifier will predict that *X* belongs to the class having the highest posterior probability, conditioned on *X*. The naive Bayesian classifier assigns an unknown sample *X* to the class C_i if and only if

$$P(C_i | X) > P(C_j | X) \quad \text{for } 1 \le j \le m, j \ne i$$

$$maximize \;\; P(C_j | X) : \text{maximum posteriori hypothesis}$$

$$P(C_i | X) = \frac{P(X | C_i) P(C_i)}{P(X)}$$

Classification and prediction

naive (or simple) bayesian classification

$P(X)$: constant for all classes, thus only

$P(X|C_i)P(C_i)$ need be maximized

If the class prior probabilities are not unknown, then

$P(C_1) = P(C_2) = \ldots = P(C_m)$; we would therefore

maximize $P(X|C_i)$

otherwise we

maximize $P(X|C_i)P(C_i)$

the class prior probabilities may be estimated by $P(C_i) = \frac{s_i}{s}$

where s_i is the number of training samples of class C_i, and

s is the total number of training samples

Classification and prediction

naive (or simple) bayesian classification

Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$.

To reduce computation, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the sample, that is, there are no dependence relationships among the attributes

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

the probabilities $P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$ can be estimated from the training samples

Classification and prediction

naive (or simple) bayesian classification

(a) If A_k is categorical, then $P(x_k | C_i) = \frac{s_{ik}}{s_i}$ where s_{ik} is the number of the training samples of class C_i having the value x_k for A_k, and s_i is the number of training samples belonging to C_i

(b) If A_k is continuous-valued, then the attribute is typically assumed to have a Gaussian distributi on so that

$$P(x_k | C_i) = g(x_k, \mu C_i, \sigma C_i) = \frac{1}{\sqrt{2\pi\sigma C_i}} e^{-\frac{(x_k - \mu C_i)^2}{2\sigma_{C_i}^2}}$$

where $g(x_k, \mu C_i, \sigma C_i)$ is the Gaussian (normal) density function attribute A_k, while μC_i and σC_i are the mean and standard deviation, respectively, given the values for attribute A_k for training samples of class C_i

Classification and prediction

naive (or simple) bayesian classification

In order to classify an unknown sample X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i

Sample X is then assigned to the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ for } 1 \leq j \leq m,\ j \neq i$$

In other words, it is assigned to the class C_i for which $P(X|C_i)P(C_i)$ is the maximum

Classification and prediction

naive (or simple) bayesian classification

RID	age	income	student	credit_rating	class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31..40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31..40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31..40	medium	no	excellent	yes
13	31..40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Classification and prediction

naive (or simple) bayesian classification

$$X = (age = "<= 30", income = "medium", student = "yes", credit\_rating = "fair")$$

maximize $P(X|C_i)P(C_i)$ for $i = 1,2$

the prior probability of each class, can be computed based on the training examples

$$P(buys\_computer = "yes") = \frac{9}{14} = 0.643$$

$$P(buys\_computer = "no") = \frac{5}{14} = 0.357$$

Classification and prediction

naive (or simple) bayesian classification

To compute $P(X|C_i)$, for $i = 1,2$, we compute the following conditional probabilities

$$P(age = "<= 30" | buys\_computer = "yes") = 2/9 = 0.222$$

$$P(age = "<= 30" | buys\_computer = "no") = 3/5 = 0.600$$

$$P(income = "medium" | buys\_computer = "yes") = 4/9 = 0.444$$

$$P(income = "medium" | buys\_computer = "no") = 2/5 = 0.400$$

$$P(student = "yes" | buys\_computer = "yes") = 6/9 = 0.667$$

$$P(student = "yes" | buys\_computer = "no") = 1/5 = 0.200$$

$$P(credit\_rating = "fair" | buys\_computer = "yes") = 6/9 = 0.667$$

$$P(credit\_rating = "fair" | buys\_computer = "no") = 2/5 = 0.400$$

Classification and prediction

naive (or simple) bayesian classification

$$P\left(X\left|buys\_computer = "yes"\right.\right) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P\left(X\left|buys\_computer = "no"\right.\right) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

$$P\left(X\left|buys\_computer = "yes"\right.\right)P\left(buys\_computer = "yes"\right) = 0.044 \times 0.643 = 0.028$$

$$P\left(X\left|buys\_computer = "no"\right.\right)P\left(buys\_computer = "no"\right) = 0.019 \times 0.357 = 0.007$$

Therefore, the naive Bayesian classifier predicts $buys\_computer = "yes"$ for sample X

Classification and prediction

genetic algorithms

- attempt to incorporate ideas of natural evolution
- an initial population is created consisting of randomly generated rules
- each rule can be represented by a string of bits
- simple example
  - suppose that samples in a given training set are described by two Boolean attributes, A_1 and A_2, and that there are two clases, C_1 and C_2
  - the rule “IF A_1 AND NOT A_2 THEN C_2” can be encoded as the bit string “100”, where the two leftmost bits represent attributes A_1 and A_2, respectively, and the righmost bit represent the class
  - the rule “IF NOT A_1 AND NOT A_2 THEN C_1” can be encoded as the bit string “001”
- if the attribute has *k* values, where $k>2$, then *k* bits may be used to encode the attribute’s values; classes can be encoded in a similar fashion

Classification and prediction

genetic algorithms

- Based on the notion of survival of the fittest, a new population is formed to consist of the *fittest* rules in the current population, as well as offsprint of the rules
- the *fitness* of a rule

  is assessed by its classification accurary on a set of training samples
- *offspring*
  - are created by applying genetic operators (crossover, mutation)
  - *crossover*

    substrings from pairs of rules are swapped to forme new pairs of rules
  - *mutation*

    randomly selected bits in rule's string are inverted
- the process of generating new populations based on prior populations of rules continues until a population P "evolves" where each rule in P satisfies a prespecified fitness threshold

Fin