Data Warehousing

Data Warehouse Design

1

Data Warehouse Design

- Conceptual data modelling principles
  - abstraction
- ER modelling review
  - ER modelling
  - Relational modelling
- Dimensional modelling

2

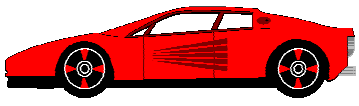
Conceptual Data Modelling Principles

- A conceptual data model is an abstraction, focusing on “important things” and hiding unnecessary detail
- Conceptual data models are intended to assist human understanding and communication about data
- A conceptual data model provides the basis for database design

3

Data Modelling Abstraction Principles

- Identification
  - What is one instance?
  - How do I identify it?



4

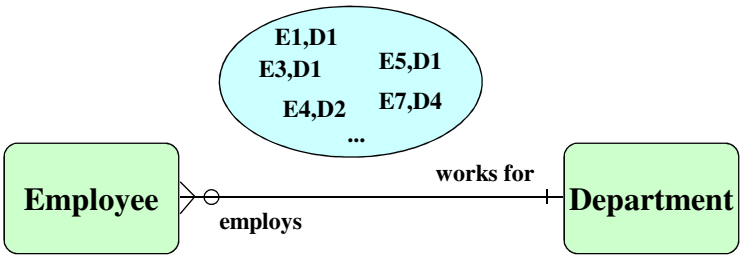
Data Modelling Abstraction Principles

- Classification
  - How will I group instances into classes?
  - What is similar about the instances in a class and what differentiates them from instances in other classes?

5

Data Modelling Abstraction Principles

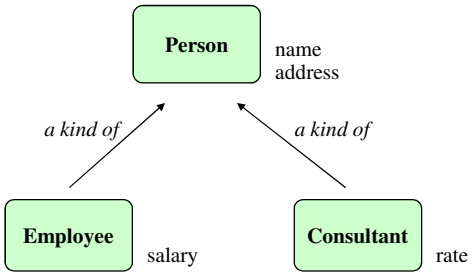
- Association
  - Which classes should I link together?
  - What is the reason (or business rule, or fact) for the link?



6

Data Modelling Abstraction Principles

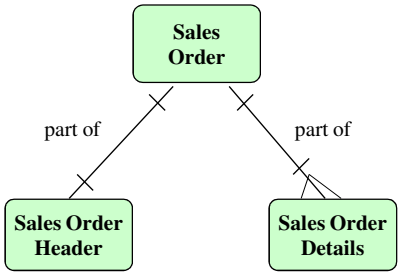
- Generalisation
  - Which classes will I group together into higher level classes (eg. generic classes)?



7

Data Modelling Abstraction Principles

- Aggregation
  - Which classes will I group together into aggregates (eg. Subject areas)?



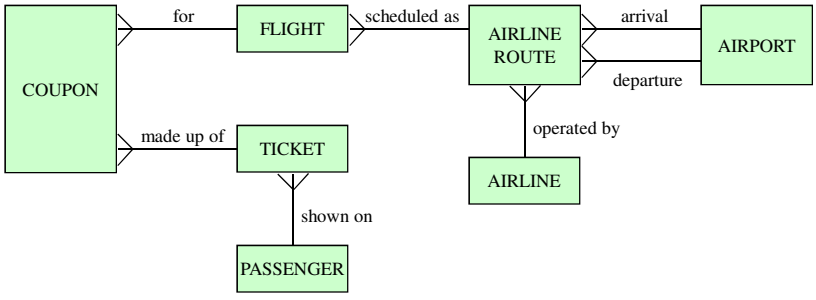
8

Entity Relationship (ER) Modelling (Review)

- first proposed by Peter Chen in 1976
- entity relationship (ER) models represent the major data classes and the business relationships between them
- ER modelling is the standard data modelling technique used in practice
- many CASE tools support ER modelling

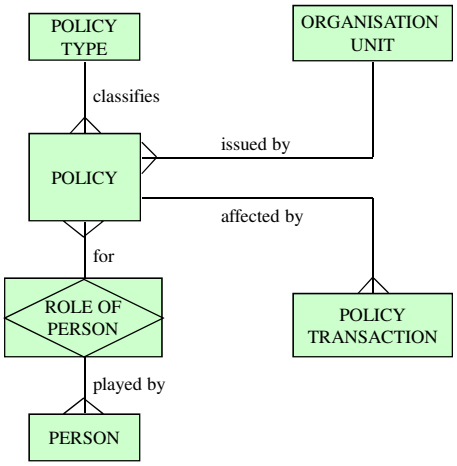
9

Example ER Model



10

Example ER Model



11

Components of ER Models

- entities
- attributes
- relationships

12

Entities

- an entity is a person, place, object, event, or concept in the user environment about which the organisation wishes to maintain data
- each entity type is represented on the ER model by a named rectangle - by convention the name is a singular noun

13

Attribute

- an attribute is a property or characteristic of an entity
- a CANDIDATE KEY is an attribute (or combination of attributes) that uniquely identifies an instance of an entity
- a PRIMARY KEY is selected from the candidate keys as the identifier of the entity
- one fact per attribute

14

Primary keys

- Choose a key that is stable and will not change its value over the life of each instance of the entity
- choose a key that has valid values and will not be null
- avoid keys with embedded structure
- consider substituting surrogate keys for large composite keys

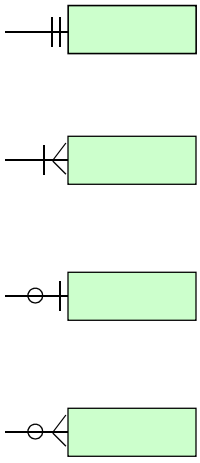
15

Relationships

- a relationship is an association between instances of one or more entity types that is of interest to the organization
- relationships are usually due to an event occurring or a natural linkage between entity instances
- they are represented using lines on the ER diagram and labeled with a verb phrase

16

Relationship Cardinalities



Mandatory 1 cardinality

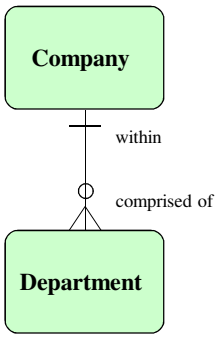
Many cardinality (1,2 …m)

Optional (0 or 1) cardinality

Optional (0 or many) cardinality

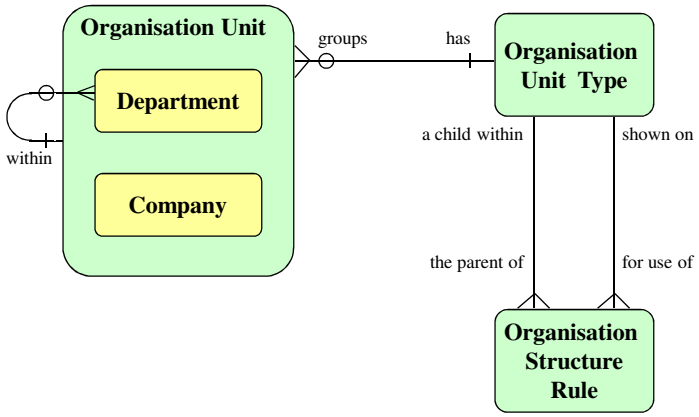
17

Example ER Model (simple)



18

Example ER Model (complex)



19

Relational Modelling (Review)

- the relational data model represents data in the form of tables (or relations)
- Important concepts are:
  - relation
  - primary key
  - functional dependency
  - foreign key
  - determinates

20

Relation

- a two dimension or "flat" file of data (loosely) corresponding to an entity

Employee

Emp#	Name	Salary	Dept
1247	Adams	24000	Finance
1982	Smith	27000	MIS
9314	Jones	33000	Finance

Employee (Emp#, Name, Salary, Dept)

21

Functional Dependency

- a functional dependency concerns values of attributes in a relation
- attribute B of a relation is functionally dependent on attribute A if each value of A has only *ONE* value of B associated with it
- Emp# → Emp-name
- Emp# → Salary
- →

22

Normalisation

- normalisation is a process for converting complex data structures into simple, stable data structures
- normalised data models are:
  - robust and stable
  - minimally redundant
  - flexible

23

Normalised Relations

- normalised relations contains a minimum amount of redundancy and allow users to insert, modify, and delete rows in tables without errors or inconsistencies (anomalies)
- Three types of anomaly are possible
  - insertion
  - deletion
  - modification

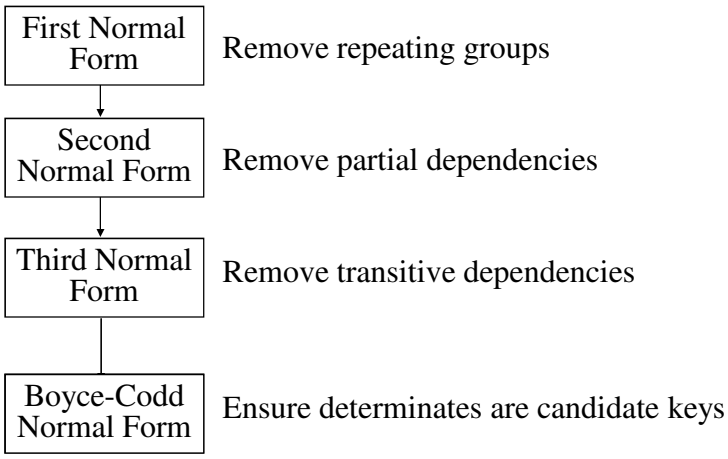
24

Normalisation

A relation is normalised if all determinates are candidate keys

25

Steps in Normalisation



26

First Normal Form

Remove Repeating Groups

repeating groups of attributes cannot be represented in a flat, two dimensional table

- Order-Item (Order#, Customer#, (Item#, Desc, Qty)

- Order-Item (Order#, Item#, Desc, Qty)
- Order (Order#, Customer#)

27

Second Normal Form

Remove Partial Dependencies

a non-key attribute cannot be identified by part of a composite key

- Order-Item (Order#, Item#, Desc, Qty)

Order-Item (Order#, Item#, Qty)

- Item (Item#, Desc)

28

Partial Dependency Anomalies

Order-Item

Order#	Item#	Desc	Qty
27	873	nut	2
28	402	bolt	1
28	873	nut	10
30	495	washer	50

- *UPDATE* change item desc in many places
- *DELETE* data for last item lost when last order for that item is deleted
- *CREATE* cannot add new item until it is ordered

29

Solution to these Anomolies

Order-Item

Order#	Item#	Qty
27	873	2
28	402	1
28	873	10
30	495	50

delete last order for item, but item remains

Item

Item#	Desc
873	nut
402	bolt
495	washer

add new item at any time

change item description in one place

30

Third Normal Form

Remove Transitive Dependencies

a non-key attribute cannot be identified

by another non-key attribute

- Employee (Emp#, Ename, Dept#, Dname)

(look for foreign keys and their attributes)

- Employee (Emp#, Ename, Dept#)
- Department(Dept#, Dname)

31

Transitive Dependency Anomolies

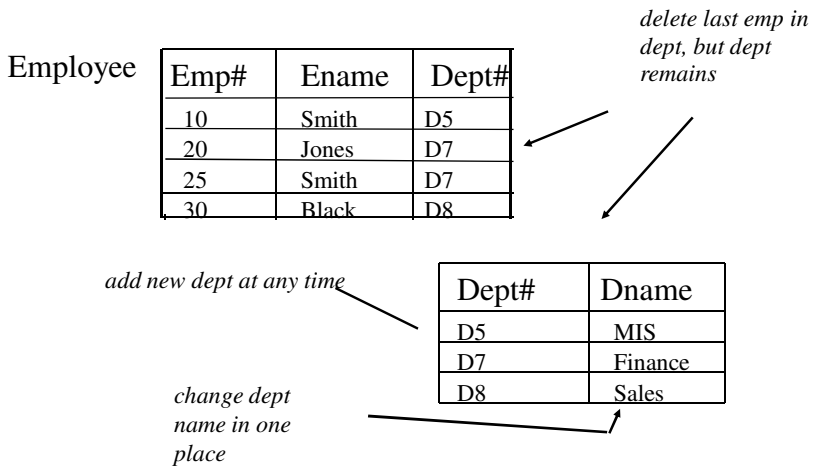
Employee

Emp#	Ename	Dept#	Dname
10	Smith	D5	MIS
20	Jones	D7	Finance
25	Smith	D7	Finance
30	Black	D8	Sales

- *UPDATE* change dept name in many places
- *DELETE* data for dept lost when last employee for that dept is deleted
- *CREATE* cannot add new dept until an employee is allocated to it

32

Solution to these Anomolies



33

Boyce-Codd Normal Form

Every determinant must be a candidate key

a non-key attribute cannot determine the values of another non-key attribute

- Allocation (Student#, Subject, Teacher)

- Assignment (Student#, Teacher)
- Department(Teacher, Subject)

34

Boyce-Codd Anomolies

Allocation

Stud#	Subject	Teach
123	Physics	Einstein
123	Music	Mozart
456	Biology	Darwin
789	Physics	Bohr
999	Physics	Einstein

- *UPDATE* student changing subject may lose teacher for that subject
- *DELETE* deleting student may lose teacher for that subject
- *CREATE* cannot assign a teacher to a subject until a student takes it

35

Solution to these Anomolies

Stud#	Teacher
123	Einstein
123	Mozart
456	Darwin
789	Bohr
999	Einstein

delete student but subject remains

Teacher	Subject
Einstein	Physics
Mozart	Music
Darwin	Biology
Bohr	Physics

Change enrolment without losing teacher/subject

assign teacher to subject

36

Relational Modelling

- mainly used for database design in transaction processing, operational systems
- used for detailed enterprise data warehouse design
- design is robust, flexible and support data integrity

37

Dimensional Modelling

- popularised by Ralph Kimball in the 1990s
- based on the multi-dimensional model of data and designed for retrieval-only databases
- very simple, intuitive, and easily-understood structure
- also known as star schema design

38

Dimensional Modelling

- A dimensional model consists of
  - a fact table
  - several dimensional tables
  - hierarchies in the dimensions
  - aggregate tables
- Essentially a "de-normalised" data model

39

Fact Table

- A fact table contains the actual business measures (additive), called facts
- Also contain foreign keys for dimensions

40

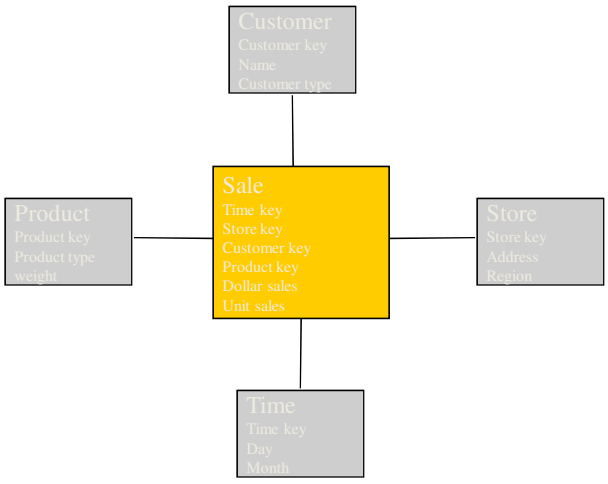
Fact Table - example

- Actual data might look like this
- Granularity, or level of detail, is a key issue

Time-id	*Store-id*	*Cust-id*	*Prod-id*	*Dollar sales*	*Unit Sales*
T100	S303	C101	P98	$120,000	5,000
T101	S303	C256	P98	$240000	10,000
T102	S387	C101	P10	$456,000	27,899
T100	S234	C400	P56	$100,200	5,600

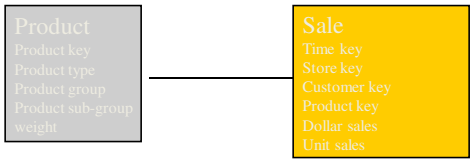
41

Dimension Tables



42

Dimension Hierarchies



Product group
- **Product subgroup**
  - **Product**

43

Dimension Table - example

- Actual data might look like this
- Hierarchy evident in data

Prod-id	*Prod-Name*	*Prod-Group*	*Prod-Subgroup*	*Weight*
P10	Hammer	Hardware	Tool	5kg
P56	10cm Nails	Hardware	Nails	1kg
P98	Plastic Pipe	Plumbing	Pipe	1kg

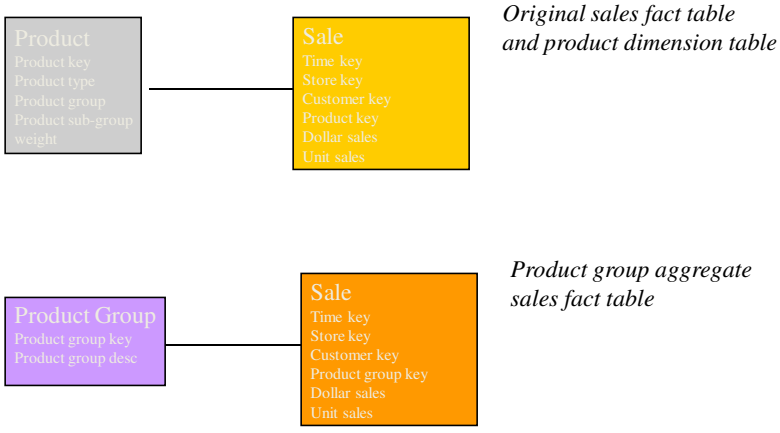
44

Aggregate Tables

- Most data warehouses are very large - aggregate tables improve performance
- An aggregate table summarises base level fact table records
- Aggregate facts reside either in new aggregate fact tables or in the original fact table

45

Aggregate Tables



46

Designing a Dimensional Model

- Choose a Business Process
- Choose the grain of the fact table
- Choose the dimensions
- Choose the measured facts
  (usually numeric, additive quantities)
- Complete the dimension tables

(Kimball, 1996)

47

Data Modelling in Data Warehouse Design

- entity relationship modelling and subject areas
  – data warehouse architecture
- relational modelling
  – data warehouse detailed design
- dimensional modelling
  – data mart design / data warehouse design

48