Man is a knot, a web, a mesh into which relationships are tied. Only those relationships matter.

Saint-Exupéry
Data modeling

- A technique for modeling data
- A graphical representation of a database
- The goal is to identify the facts to be stored in the database
- Data modeling is a partnership between the client and analyst
The building blocks

- Entity
- Attribute
- Relationship
- Identifier
Data model quality

- A well-formed data model
- A high fidelity image
A well-formed data model

- Construction rules obeyed
- No ambiguity
  - All entities, attributes, relationships, and identifiers are defined
  - Names are meaningful to the client
A high fidelity image

- Faithfully describes the world it is supposed to represent
- Relationships are of the correct degree
- Data model is complete, understandable, and accurate
- The data model makes sense to the client
Quality improvement

- Is the level of detail correct?
- Are all exceptions handled?
- Is the model accurate?
Pure geography

Can a nation have more than one capital?
Can a city be the capital of more than one state?
Geography revised

NATION
  *natname

ADMIN UNIT
  *unitname
  unittype
  unitpop
  unitarea

ADMIN UNIT-CITY
  unitcaptype

CITY
  *cityid
  cityname
  citypop
  cityarea
  natcaptype

National capital
Family matters - take 1

Can we generalize?
Family matters - take 2
What about couples who are not officially married but have cohabited for an extended period?
Family matters - take 4
Family matters - take 5

PERSON
- *id
- dob
- fname
- oname
- lname
- gender

MARRIAGE
- *marriageno
- marriagestatus
- begindate
- enddate
Bookish matters - take 1

LIBRARY
*libname

BOOK
*callno
ISBN
booktitle
duedate

BORROWER
*borrowerid
...

Should copyno be an attribute of book?
This model records only the current borrower of a copy of a book
History - take 1

Can an employee work in multiple departments?
History - take 2

How do we keep track of an employee’s pay checks?
How is an instance of PAYSLIPLINE identified?
History - take 4
A ménage à trois for entities - take 1

Where do we store information about the lease?
A ménage à trois for entities - take 2

Why is start date part of the composite primary key?
Planning and doing - take 1

- EMPLOYEE
  - *empno
  - ...
- ACTIVITY
  - planned hours
- PROJECT
  - *projectid
  - ...
- DAILY WORK
  - *workdate
  - actual hours
Planning and doing - take 2

- **EMPLOYEE**
  *empno
  ...

- **ACTIVITY**

- **PROJECT**
  *projectid
  ...

- **DAILY WORK**
  *workdate
  planned hours
  actual hours
<table>
<thead>
<tr>
<th>Cardinality</th>
<th>Modality</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,1</td>
<td>Optional</td>
<td>There can be zero or one instances of the entity relative to the other entity</td>
</tr>
<tr>
<td>0,n</td>
<td></td>
<td>There can be zero or many instances of the entity relative to the other entity</td>
</tr>
<tr>
<td>1,1</td>
<td>Mandatory</td>
<td>There is exactly one instance of the entity relative to the other entity</td>
</tr>
<tr>
<td>1,n</td>
<td></td>
<td>The entity must have at least one and can have many instances relative to the other entity</td>
</tr>
</tbody>
</table>
Minimalist approach

Focus has been on identifying the basic cardinality (1:m or m:m?)

Now add greater precision

There must be 1 instance

Learn the basics and then add more detail
Modality

- Also known as optionality
- Cardinality indicates the range of instances in a relationship
- Modality defines the minimum number of instances
- Cardinality and modality are linked
Modality and Cardinality

- Optional entity
  - Cardinality is 0
  - O

- Mandatory entity
  - Cardinality is 1
  - 1 of 1:m
Modality

Can a lineitem exist without a sale?

Can an item exist without a lineitem?
Modality

- Does every department have a boss?
- Is every employee a department boss?
Modality

DEPT
*deptname
depthfloor
depthphone

EMP
*empno
empfname
empsalary

employee's boss

department's boss

Modality

Why is it optional for a monarch to have a successor?
Modality

- Optional for a product to have components
- Optional for a product to be a component
- Every assembly must have products
Modality

- Adds additional information to a data model
- If a relationship is mandatory then add a constraint
  - Could be
    - Referential integrity constraint
    - Application logic
Entity types

- Independent
- Dependent
- Associative
- Aggregate
- Subordinate
Independent

- Often a starting point
- Prominent in the client's mind
- Often related to other independent entities

**NATION**
- *nation code*
- *nation name*
- *exchange rate*

**STOCK**
- *stock code*
- *stock firm*
- *stock price*
- *stock quantity*
- *stock dividend*
- *stock PE*
Dependent

- Relies on another entity for its existence and identification
- Can become independent if given an arbitrary identifier
Associative

- A by-product of an m:m relationship
- Typically between independent entities
- Can store current or historical data
- Can become independent if given an arbitrary identifier

Diagram:

- DIVISION
  *divname

- DEPARTMENT
  *deptname ...

- POSITION
  *begdate
  *enddate
  *posttitle

- EMPLOYEE
  *empid ...

+ relationships between entities.
Aggregate

- Created from several different entities that have a common prefix or suffix
- Commonly used with addresses or names
Subordinate

An entity with data that can vary among instances
Generalization

A relationship between a more general element and a more specific element

- ANIMAL
  - *animalid
  - animaldob
  - ...

- SHEEP
  - fleeceweight
  - ...

- HORSE
  - furlongspeed
  - ...
Generalization

- Map with one table for each entity
- For each of the subtype entities the primary key is that of the supertype entity
- You must also make this column a foreign key so that a subtype cannot be inserted without the presence of the matching supertype
UML aggregation

Aggregation is a part-whole relationship between two entities
UML shared aggregation

One entity owns another entity, but other entities can own that entity as well.
UML composite aggregation

One entity exclusively owns the other entity
Data model contraction

\[\text{ART COLLECTION}\]

\[\text{PAINTING}\quad \text{SCULPTURE}\quad \text{CERAMIC}\]

\[\text{ART}\quad \text{art type}\]
Hints on data modeling

- The model will expand and contract
- Invent identifiers where necessary
- Identifiers should have only one purpose – identification
- A data model does not imply ordering
- Create an attribute if ordering of instances is required
- An attribute’s meaning must be consistent
Names and addresses

The query test

- If an attribute has parts, are any of the parts ever likely to appear in a query?

Have an understanding on representing names and addresses in a data model
A US zip code is CHAR(5) because leading zeroes are displayed
- Boston MA 02201

Full US zip is CHAR(10)
- 30602-6273

VARCHAR(20) probably covers all countries
Hints on data modeling

- Single instance entities are OK
- Select names carefully
- Synonyms—different words have the same meaning
  - Get clients to settle on a common word or use views
- Homonyms—same word has different meanings
  - Clarify to avoid confusion
- Naming associative entities
  - Concatenate entity names if there is no obvious real world name
Hints on data modeling

- Uncover all exceptions
- Label relationships to avoid ambiguity
- Keep the data model well-formed and accurate
Meaningful identifiers

An identifier is meaningful when some attributes of the entity can be inferred from the identifier’s value.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognizable and rememberable</td>
<td>Identifier exhaustion</td>
</tr>
<tr>
<td>Administrative simplicity</td>
<td>Reality changes</td>
</tr>
<tr>
<td></td>
<td>Loss of meaningfulness</td>
</tr>
</tbody>
</table>
Recommendation

Nothing, however, is lost and much is gained by using **non-meaningful** identifiers.

Non-meaningful identifiers serve their sole purpose well:
- To uniquely identify an entity.

Attributes are used to describe the characteristics of the entity.

A clear distinction between the role of identifiers and attributes creates fewer data management problems.
The seven habits of highly effective data modelers

- Immerse
- Challenge
- Generalize
- Test
- Limit
- Integrate
- Complete
Key points

- A high-fidelity data model handles all exceptions
- Identifiers need identify only an instance
- Data modeling skills take time to develop