Introduction to Entities, Attributes, and Relationships
Overview

Why conceptual modeling?

Introduction of the Key role players:

- Entities
- Attributes
- Relationships
Why Create a Conceptual Model?

- It describes exactly the information needs of the business
- It facilitates discussion
- It helps to prevent mistakes, misunderstanding
- It forms important “ideal system” documentation
- It forms a sound basis for physical database design
- It is a very good practice with many practitioners
Between Dream and Reality...
Entity Relationship Modeling

- Models business, not implementation
- Is a well-established technique
- Has a robust syntax
- Results in easy-to-read diagrams...

...although they may look rather complex at first sight
Goals of Entity Relationship Modeling

- Capture *all* required information
- Information appears *only* once
- Model *no* information that is derivable from other information already modeled
- Information is in a predictable, logical place
Database Types

Hierarchical

ER Model

Relational

Network
Entity

- An Entity is:
  - “Something” of significance to the business about which data must be known
  - A name for the things that you can list
  - Usually a noun
- Examples: objects, events
- Entities have instances
## Entities and Instances

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON</td>
<td>Mahatma Gandhi</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>2.5 x 35 mm copper nail</td>
</tr>
<tr>
<td>PRODUCT TYPE</td>
<td></td>
</tr>
<tr>
<td>EMPLOYMENT CONTRACT</td>
<td>my previous contract</td>
</tr>
<tr>
<td>JOB</td>
<td>violinist</td>
</tr>
<tr>
<td>SKILL LEVEL</td>
<td>fluent</td>
</tr>
<tr>
<td>TICKET RESERVATION</td>
<td>tonight: Hamlet in the Royal</td>
</tr>
<tr>
<td>PURCHASE</td>
<td>the CD I bought yesterday</td>
</tr>
<tr>
<td>ELECTION</td>
<td>for parliament next fall</td>
</tr>
<tr>
<td>PRINTER PREFERENCE</td>
<td></td>
</tr>
<tr>
<td>DOCUMENT VERSION</td>
<td></td>
</tr>
</tbody>
</table>
Entities and Sets

An entity represents a set of instances that are of interest to a particular business.

**JOB**

- manager
- cook
- dish washer
- porter
- piano player
- waitress
- financial controller
- waiter
Attribute

- Also represents something of significance to the business
- Is a *single valued* property detail of an entity
- Is a specific piece of information that:
  - Describes
  - Quantifies
  - Qualifies
  - Classifies
  - Specifies an entity
## Attribute Examples

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>Family Name, Age, Shoe Size, Town of Residence, Email, ...</td>
</tr>
<tr>
<td>CAR</td>
<td>Model, Weight, Catalog Price, ...</td>
</tr>
<tr>
<td>ORDER</td>
<td>Order Date, Ship Date, ...</td>
</tr>
<tr>
<td>JOB</td>
<td>Title, Description, ...</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>Amount, Transaction Date, ...</td>
</tr>
<tr>
<td>EMPLOYMENT</td>
<td>Start Date, Salary, ...</td>
</tr>
<tr>
<td>CONTRACT</td>
<td></td>
</tr>
</tbody>
</table>
Relationships

- Also represent something of significance to the business
- Express how entities are mutually *related*
- Always exist between *two* entities (or one entity *twice*)
- Always have two perspectives
- Are named at both ends
Relationship Examples

EMPLOYEES have JOBS
JOBS are held by EMPLOYEES

PRODUCTS are classified by a PRODUCT TYPE
PRODUCT TYPE is a classification for a PRODUCT

PEOPLE make TICKET RESERVATIONS
TICKET RESERVATIONS are made by PEOPLE
Employees have Jobs

**Numerical observation:**

- All EMPLOYEES have a JOB
- No EMPLOYEE has more than one JOB
- Not all JOBS are held by an EMPLOYEE
- Some JOBS are held by more than one EMPLOYEE
Entity Representation in Diagram

- Drawn as a “softbox”
- Name singular
- Name inside
- Neither size, nor position has a special meaning

During design, entities usually lead to tables.
Attributes in Diagrams

Mandatory attribute, that is, known and available for every instance.

Optional attribute, that is, unknown or unimportant to know for some instances.

EMPLOYEE
- Family Name
- Address
- Birth Date
- Shoe Size
- Email

JOB
- Title
- Description

During design, attributes lead to columns.
An employee \textit{has exactly one job.}

Jobs are \textit{held by one or more employees.}

During design, relationships lead to foreign keys.
Diagrams Are To Communicate
Characteristics Of The Relationship Line

mandatory: _______________  optional: — — — — — —
Two Perspectives

mandatory: EMPLOYEE has JOB
optional: held by
One Way

Every EMPLOYEE has exactly one JOB

mandatory: ————-    optional: — — — — —

EMPLOYEE has    held by

JOB
The Other Way

A JOB may be *held by* one or more EMPLOYEES
Reading a Relationship End

P \quad \text{split into} \quad \text{part of} \quad Q
Reading a Relationship End

P split into Q

part of
"Each P must be split into exactly one Q, 
may be split into one or more Qs."
“Each P may be split into one or more Qs”
“Each P may be split into one or more Qs”
“Each P may be *split into* one or more Qs”

“Each Q must be *may be* exactly one P one or more Ps”
“Each P may be *split into* one or more Qs”

“Each Q must be *part of* exactly one P”
Functions Drive Data

- **Business functions are always present.**
  - Explicit
  - Assumed
- **Business functions need data.**
- **An entity, attribute, or relationship may be modeled because:**
  - It is used by a business function.
  - The business need may arise in the near future.
# Weather Forecast

**January 26**

<table>
<thead>
<tr>
<th>Location</th>
<th>Temperature</th>
<th>Condition</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>København</td>
<td>1/-5</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Bremen</td>
<td>0/-3</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Berlin</td>
<td>3/-1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>München</td>
<td>5/-3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Amsterdam</td>
<td>8/3</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Bruxelles</td>
<td>4/0</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Paris</td>
<td>4/1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Bordeaux</td>
<td>7/2</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
Weather Forecast, a Solution

- CITY
  - Name
  - Geographical Position

- COUNTRY
  - Name
  - Geographical Position

- FORECAST
  - Date
  - Minimum Temperature
  - Maximum Temperature
  - Wind Force

- WEATHER TYPE
  - Icon
  - Description

- WIND DIRECTION
  - Icon
  - Description

- Subject of
- Located in
- Having
- Referring to
- Referred in
Graphical Elements of ER Diagram

- Entity
- Attribute
- Relationship

Subtype
Unique identifier
Arc
Nontransferability
Summary

- ER Modeling models information conceptually
- Based on functional business needs
- “What”, not “How”
- Diagrams provide easy means of communication
- Detailed, but not too much
Practices

- Instance or Entity
- Guest
- Reading
- Hotel
- Recipe
# Practice: Instance or Entity?

<table>
<thead>
<tr>
<th>Concept</th>
<th>E/A/I?</th>
<th>Example Instance or Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRESIDENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELLA FITZGERAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANIMAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEIGHT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>CAR</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>CAR</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>CAR</td>
<td></td>
</tr>
</tbody>
</table>
### Practice: Guest

<table>
<thead>
<tr>
<th>Address</th>
<th>Arrival Date</th>
<th>Family Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hotel</th>
<th>Room Number</th>
<th>Floor Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Room</th>
<th>Number of Beds</th>
<th>Number of Parking Lots</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price</th>
<th>TV set available?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Note:**
- Address: Input the address of the hotel.
- Arrival Date: Provide the date the guest is expected to arrive.
- Family Name: Enter the guest's family name.
- Room Number: Specify the number of the guest's room.
- Floor Number: Indicate the floor where the room is located.
- Number of Beds: State the number of beds in the room.
- Number of Parking Lots: Mention the number of parking lots available.
- Price: Specify the cost of the room.
- TV set available?: Confirm if a TV set is available in the room.
Practice: Reading

A Each EMPLOYEE may be assigned to one or more DEPARTMENTS
Each DEPARTMENT must be responsible for one or more EMPLOYEES

B Each EMPLOYEE must be assigned to one or more DEPARTMENTS
Each DEPARTMENT may be responsible for one or more EMPLOYEES

C Each EMPLOYEE must be assigned to exactly one DEPARTMENT
Each DEPARTMENT may be responsible for exactly one EMPLOYEE
Practice: Read and Comment

**PERSON**
- born in
- living in
- visitor of
- mayor of

**TOWN**
- birthplace of
- home town of
- visited by
- with mayor
Practice: Hotel

HOTEL
* Address

STAY
* Arrival Date

ROOM
* Room Number

PERSON
* Name

the lodging for

host of

in

with

of

with

guest in
| Soups | Açorda alentejana  
bread soup from Portugal |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>vegetarian</td>
<td>For 4 persons:</td>
</tr>
</tbody>
</table>
| 15 min easy | 1 onion  
4 cloves of garlic  
1 red pepper  
1 liter of vegetable broth  
4 tablespoons of olive oil  
4 fresh eggs  
1 handful of parsley or coriander  
salt, pepper  
9-12 slices of (old) bread |
| Preparation | Cut the onion into small pieces and fry together with the garlic. Wash the red pepper, cut it in half, remove the seeds and fry it for at least 15. |
Entities and Attributes in Detail
Overview

- Data compared to information
- Entities and how to track them down
- Attributes
- Subtypes and supertypes
Data Compared to Information

Data
- *Facts given from which other facts may be inferred*
- *Raw material*
- Example: Telephone Directory

Information
- *Knowledge, intelligence*
- Example: Telephone number of florist
Data

- *Modeling, Conceptual*
  Structuring data concepts into logical, coherent, and mutually related groups

- *Modeling, Physical*
  Modeling the structure of the (future) physical database

- *Base*
  A set of data, usually in a variety of formats, such as paper and electronically-based

- *Warehouse*
  A huge set of organized information
Entities

- Give the entity a unique name
- Create a formal description of the entity
- Add a few attributes, if possible
- Be aware of homonyms
- Check entity names and descriptions regularly
- Avoid use of reserved words
- Remove relationship name from entity name
Relationship Name in Entity Name

GUEST

guest of

HOTEL

host of

PERSON

guest of

ACCOMMODATION

host of
Some Background Information

“ElectronicMail (EM) wants to provide an attractive and user-friendly Web-based e-mail system. Important concepts are user and message.

An EM user has a unique address of 30 characters at most and a password supplied by the person who set up the EM user. Who the person really is, we do not know, although we ask for some additional information, such as the name, country, birth date, line of business, and a few more things.
Users must be able to send and receive mail messages. A mail message is usually a piece of straight text. A message may have attached files. An attachment is a file, like a spreadsheet, that is sent and kept with the message, but not created with our software.

Messages are kept in folders. Every user has three folders to start with: Inbox, Outbox, and Wastebasket. Additional folders can be created by the user.”
Subject: test
To: bipi, giovanni, papini@yahoo.com
Cc: myself
Bcc: 
Message text: this is a test

Attachments:
- abc.html
- xyz.doc

Type:
- Hypertext
- Word document
Some Desired Functionality

- “Users of ElectronicMail must be able to address messages to a mail list, for example, a group of e-mail addresses. The system should only keep one copy of the message sent (to save database space) plus information about whom the message was sent to.

- Users must be able to create templates for their messages. A template must be named and may contain everything a real message contains. A template may be used for new messages.
Some Desired Functionality

- Users must be able to reply to a message. By replying the user creates a new message to which the old message is added.
- Users must be able to create an alias for an e-mail address, to hide the often complex addresses behind an easy-to-remember nickname.”
Evolution of an Entity Definition

- A message is a piece of text sent by a user.
- A message is a piece of text sent by *an EM* user.
- A message is a note that is sent by an EM user. *A message does not necessarily contain text, nor a subject, etc.*
- A message is a note that is sent by an EM user *or received by an EM user or both*. A message does not necessarily contain text, nor a subject, etc.
- A message is a note that is *received* by an EM user. A message does not necessarily contain text, nor a subject, etc.
Business Functions

• “Users of ElectronicMail must be able to address messages to a mail list, for example, a group of e-mail addresses. The system should only keep one copy of the message sent (to save data base space) plus information about whom the message was sent to.

• Users must be able to create templates for their messages. A template must be named and may contain everything a real message contains. A template may be used for new messages.
Business Functions

- Users must be able to *reply* to a message. By replying the user *creates* a new message to which the old message is added.
- Users must be able to create an alias for an e-mail address, to hide the often complex addresses behind an easy-to-remember nickname.”
An Attribute...

- Always answers “of what?”
- Is the property of entity, not of relationship
- Must be single valued
- Has format, for example:
  - Character string
  - Number
  - Date
  - Picture
  - Sound
- Is an elementary piece of data
Nouns, Entities, Attributes

- “ElectronicMail (EM) wants to provide an attractive and user friendly Web-based email system. Important concepts are user and message.

- An EM USER has a unique address of 30 characters at most and a password supplied by the PERSON who set up the EM user. Who the person really is, we do not know, although we ask for some additional information, like the name, COUNTRY, birth date, line of business, and a few things more.
Nouns, Entities, Attributes

- Users must be able to send and receive mail *MESSAGES*. A mail message is usually a piece of straight *text*. A message may have attached files. An *ATTACHMENT* is a *file*, like a spreadsheet, that is sent and kept with the message, but not created with our software.

- Messages are kept in *FOLDERS*. Every user has three folders to start with: *Inbox*, *Outbox* and *Wastebasket*. Additional folders can be created by the user.”
EM Entities and Attributes

Nouns

user
address
password
person
name
country
birth date
occupation
message
text
attachment
file
folder
inbox
outbox
wastebasket

Entities/Attributes/Instances

USER
Address
Password
PERSON
Name
COUNTRY
Birth Date
Occupation
MESSAGE
Text
ATTACHMENT
File
FOLDER
Inbox
Outbox
Wastebasket

Entities with their Attributes

USER
- Address
- Password
PERSON
- Name
- Birth Date
- Occupation
COUNTRY
- Name
MESSAGE
- Text
ATTACHMENT
- Filename
FOLDER
- Name
Attribute and Entity

Attributes in one model can be entities in another.

GARMENT
Name
Price

CURRENCY
PRICE
NAME
LANGUAGE
# Redundancy

<table>
<thead>
<tr>
<th>COMMODITY</th>
<th>Name</th>
<th>Price exclusive VAT</th>
<th>Price inclusive VAT</th>
<th>VAT %</th>
</tr>
</thead>
</table>

Copyright © Oracle Corporation, 2002. All rights reserved.
A Subtype ...

- Inherits all attributes of supertype
- Inherits all relationships of supertype
- Usually has its own attributes or relationships or business functions
- Is drawn within supertype
- Never exists alone
- May have subtypes of its own
- Is also known as “Subentity”
Subtype: Example
Subtype: Rules

Subtypes of the same entity must be:

• Exhaustive:
  Every instance of a supertype is also instance of one of the subtypes.

and

• Mutually exclusive:
  Every instance of the supertype is of one and only one subtype.

Name subtypes adequately:

A
B
C
NON-B
OTHER A
Subtypes: Three Levels

- COMPOSITION
  - Subject
  - Cc
  - Bcc
  - Text

- MESSAGE

- OTHER
  - COMPOSITION
    - Name
  - DRAFT
  - TEMPLATE
More on Subtypes

Subtypes *always* exist...

![Subtype Diagram](chart1.png)

... but do not all make sense

![Subtype Diagram](chart2.png)
Summary

Entities
- Nouns in texts
- Tangible, intangible, events

Attributes
- Single-valued qualifiers of entities

Subtypes
- Inherit all attributes and relationships of supertype
- May have their own attributes and relationships
Practices

- Books
- Moonlight Coffees
- Shops
- Subtypes
- Schedule
- Address
1. I have just finished writing a book. It’s a novel about justice and power.
2. We have just published this book. The hard cover edition is available now.
3. Did you read that new book on Picasso? I did. It’s great!
4. If you like you can borrow my book.
5. I have just started translating this book into Spanish. I use the modern English text as a basis and not the original, which is 16th century.
6. I ordered that book for my parents.
7. Yes, we have that book available. You should find it in Art books.

8. A second printing of the book War and Peace is very rare.

9. I think My name is Asher Lev is one of the best books ever written. Mine is autographed.

10. I want to write a book on entity relationship modeling when I retire.
Summary

- Moonlight Coffees is a fast growing chain of high quality coffee shops with currently over 500 shops in 12 countries of the world. Shops are located at first-class locations, such as major shopping, entertainment and business areas, airports, railway stations, museums. Moonlight Coffees has some 9,000 employees.

Products

- All shops serve coffees, teas, soft drinks, and various kinds of pastries. Most shops sell nonfoods, like postcards and sometimes even theater tickets.
Summary

Financial

Shop management reports sales figures on a daily basis to Headquarters, in local currency. Moonlight uses an internal exchange rates list that is changed monthly. Since January 1, 1999, the European Community countries must report in Euros.

Stock

Moonlight Coffees is a public company; stock is traded at NASDAQ, ticker symbol MLTC. Employees can participate in a stock option plan.
Shop List

Shoplist, ordered to date opened  page 4

181 The Flight, JFK Airport terminal 2, New York, USA, 212.866.3410, Airport, 12-oct-97

182 Hara, Kita Shinagawa, Tokyo, JP, 3581.3603/4, Museum, 25-oct-97

183 Phillis, 25 Phillis Rd, Atlanta, USA, 405.867.3345, Shopping Centre, 1-nov-97

184 JFK, JFK Airport terminal 4, New York, USA, 212.866.3766, Airport, 1-nov-97
Shop List

185 VanGogh, Museumplein 24, Amsterdam, NL, 76.87.345, Museum, 10-nov-97

186 The Queen, 60 Victoria Street, London, UK, 203.75.756, Railway Station, 25-nov-97

187 Wright Bros, JFK Airport terminal 1, New York, USA, 212.866.9852, Airport, 6-jan-98

188 La Lune, 10 Mont Martre, Paris, FR, 445 145 20, Entertainment, 2-feb-98
Subtypes

- **DISABLED PERSON**
  - DEAF
  - BLIND
  - OTHER DISABLED PERSON

- **CAR**
  - STATION WAGON
  - SEDAN

- **BUILDING**
  - HOUSE

- **HOTEL**
  - ROOM WITH BATH
  - OTHER ROOM

- **DOG**
  - DOMESTIC ANIMAL
  - MAMMAL
van Gogh, Museumplein, Amsterdam

<table>
<thead>
<tr>
<th>Schedule Oct 12 - Oct 18</th>
<th>prepared by Janet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift</td>
<td>Mon</td>
</tr>
<tr>
<td>Annet S</td>
<td></td>
</tr>
<tr>
<td>Annet B</td>
<td>1</td>
</tr>
<tr>
<td>Dennis</td>
<td>2</td>
</tr>
<tr>
<td>Jürgen</td>
<td></td>
</tr>
<tr>
<td>Kiri</td>
<td></td>
</tr>
<tr>
<td>Wil</td>
<td></td>
</tr>
</tbody>
</table>
Practice: Address (1/2)

Rheingasse 123
53111 Bonn
Germany

34 Oxford Road
Reading
Berkshire RG1 8JS
UK

1020 Maple Drive
Kirkland WA 98234
USA
Practice: Address (2/2)

P.O. Box 66708
Nairobi
Kenya

c/o Mrs Smith
Maude Street
Sandton
Johannesburg 2144
South Africa
Relationships in Detail
Overview

- Relationships
- Ten different relationship types
- Nontransferability
- Relationships that seem to have attributes
- Rules of Normalization
Establishing a Relationship

- Determine the existence of a relationship
- Choose a name for the relationship from both perspectives
- Determine optionality
- Determine degree
- Determine nontransferability
Establishing the Relationship

MESSAGE

sending

receiving

USER

replying
Relationship Names

MESSAGE

USER

sent by

sender of

sent to

receiver of

replied to by

reply of
Naming the Relationship

A MESSAGE is received by a USER

A USER is receiver of a MESSAGE
Optionality
Optionality

No: — — — — —  Yes: ————

• *Must* every MESSAGE be received by a USER? *Yes*
• *Must* every USER be receiver of a MESSAGE? *No*
Mandatory 1: Mandatory m

- Every A must be *split into* at least one B
- Every B must be *part of* exactly one A
Degree

MESSAGE

with <5

ATTACHMENT

written by

USER

author of

received by

receiver of

reply of

replied to by
Degree

One: ___________ One or more: ___________

MESSAGE

\[\text{received by}\]

\[\text{receiver of}\]

USER

• *Can a MESSAGE be received by more than one USER?* Yes
• *Can a USER be the receiver of more than one MESSAGE?* Yes
Nontransferability
Relationship Types
1:m

(a) 

(b) 

(c) 

(d)
Relationship Types

m:1
Relationship Types
m:m

(e)

(f)

(g)
Relationship Types
m:m

USER

part of

LIST

consists of
Relationship Types
1:1

(h) ___________________________

(i) ___________________________

(j) ___________________________
1:1 Relationships
Roles

PERSON
* Name

acting as

PATIENT
* Blood Type

role of

acting as

EMPLOYEE
* Job

role of

acting as
1:1 Relationships
Process

DRAFT  

basis for  

MESSAGE

result of
Redundant Relationships

- Location of a person's hometown is redundant.
- Location of a person's birth is redundant.

Diagram:
- COUNTRY
  - location of TOWN
  - located in
- TOWN
  - hometown of PERSON
  - living in
- PERSON
  - born in
Relationships and Attributes

- An attribute can hide a relationship
- Relationship can be “downgraded” to attribute
Attribute Compared to Relationship

- Easy model
- Fewer tables
- No join

- Value control
- List of values
- Other relationships

```
ATTACHMENT
  * Type
  * Content

ATTACHMENT TYPE
  * Name

with

of
```
Attribute or Entity

- NAME
- SALARY
- GENDER
- TEAM

EMPLOYEE

* Id

- JOB
- BADGE
- NATIONALITY
- ADDRESS
Attribute Compared to Relationship

- There is no such thing as a foreign key attribute.
- Usually, the attribute name should not contain an entity name.

Message:
- * Message-Id
- * Text
- * Folder Name

Folder:
- * Name

The diagram shows the relationship between a **MESSAGE** and a **FOLDER** with operations **containing** and **placed in**.
Relationship Compared to Attribute

MESSAGE

```
addressed to
```

USER

```
addressee of
```

MESSAGE

```
* Addressee
```

USER

MESSAGE

```
o Addressee
```

USER

```
addressed to
addressee of
```
m:m Relationships May Hide Something

CUSTOMER
  * Id
  * Name

PRODUCT
  * Code
  * Name

buyer of

bought by
Quantity Is Attribute of ...

CUSTOMER
* Id
* Name
 Quantity

PRODUCT
* Code
* Name

buyer of

bought by

CUSTOMER
* Id
* Name

PRODUCT
* Code
* Name
 Quantity

buyer of

bought by
Attribute of Relationship?

CUSTOMER
* Id
* Name

PRODUCT
* Code
* Name

buyer of

bought by

Quantity
New Entity ORDER

CUSTOMER
- * Id
- * Name

PRODUCT
- * Code
- * Name

ORDER
- *Quantity Sold

CUSTOMERS
<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sanchez</td>
</tr>
<tr>
<td>2</td>
<td>Lowitch</td>
</tr>
<tr>
<td>3</td>
<td>Yomita</td>
</tr>
</tbody>
</table>

PRODUCTS
<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jeans</td>
</tr>
<tr>
<td>2</td>
<td>Shirt</td>
</tr>
<tr>
<td>3</td>
<td>Tie</td>
</tr>
</tbody>
</table>

ORDERS
<table>
<thead>
<tr>
<th>Ctr_id</th>
<th>Pdt_code</th>
<th>Quantity_sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Multiple PRODUCTS for an ORDER

CUSTOMER
- *Id*
- *Name*

PRODUCT
- *Code*
- *Name*

ORDER
- *Id*
- *Date*

Quantity
Another New Entity: ORDER ITEM

CUSTOMER
* Id
* Name

PRODUCT
* Code
* Name

ORDER HEADER
* Id
* Date

ORDER ITEM
* Quantity Sold

with
of
with
with
for
for
## Tables

### CUSTOMERS

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sanchez</td>
</tr>
<tr>
<td>2</td>
<td>Lowitch</td>
</tr>
<tr>
<td>2</td>
<td>Yomita</td>
</tr>
</tbody>
</table>

### ORDER_HEADERS

<table>
<thead>
<tr>
<th>Id</th>
<th>Ctr_id</th>
<th>Date_ordered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>25-MAY-1999</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>25-MAY-1999</td>
</tr>
</tbody>
</table>

### ORDER_ITEMS

<table>
<thead>
<tr>
<th>Ohd_id</th>
<th>Pdt_code</th>
<th>Quantity_sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### PRODUCTS

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jeans</td>
</tr>
<tr>
<td>2</td>
<td>Shirt</td>
</tr>
<tr>
<td>3</td>
<td>Tie</td>
</tr>
</tbody>
</table>
Resolving m:m Relationship

- Create new intersection entity
- Create two m:1 relationships, derive optionality
- Remove m:m relationship
Resolving m:m Relationship

- Create new intersection entity
- Create two m:1 relationships, derive optionality
- Remove m:m relationship
Resolving m:1 Relationship

PERSON classified as external

classification of Type

classified as internal

CUSTOMER
Resolving m:1 Relationship

PERSON

with

external

CUSTOMER TYPE

for

CLASSIFICATION

in

internal

with
## Normalization Rules

<table>
<thead>
<tr>
<th>Normal Form Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Normal Form</td>
<td>All attributes are single valued.</td>
</tr>
<tr>
<td>Second Normal Form (2NF)</td>
<td>An attribute must be dependent upon entity’s entire unique identifier.</td>
</tr>
<tr>
<td>Third Normal Form (3NF)</td>
<td>No non-UID attribute can be dependent on another non-UID attribute.</td>
</tr>
</tbody>
</table>

“A normalized entity-relationship data model automatically translates into a normalized relational database design”

“Third normal form is the generally accepted goal for a database design that eliminated redundancy”
First Normal Form in Data Modeling

USER
# Name
* Person Name
* Message Receive Date
  o Message Subject
  o Message Text

All attributes must be single-valued.

RECEIVED MESSAGE
# Receive Date
  o Subject
  o Text

received by
receiver of

USER
# Name
* Person Name
Second Normal Form in Data Modeling

An attribute must be dependent upon its entity’s entire unique identifier.
No non-UID attribute can be dependent upon another non-UID attribute.
Summary

- Relationships express how entities are connected.
- Initially relationships often seem to be of type m:m.
- Finally relationships are most often of type m:1.
- Relationships can be resolved into:
  - Two new relationships
  - One intersection entity
- Third Normal form is generally accepted standard.
Practices

• Read the Relationship
• Find a Context
• Name the Intersection Entity
• Receipt
• Moonlight P&O
• Price List
• EMail
• Holiday
Practice: Read the Relationship

ALU \( \text{of} \) BRY

PUR \( \text{bazooned in} \) YOK

KLO \( \text{bilought in} \) HAR

\( \text{with} \)

\( \text{bazooned by} \)

\( \text{glazoed with} \)
Find a Context (1)
Find a Context (2)
Find a Context (3)
Find a Context (4)
Practice: Name the Intersection Entity

PRODUCT sold by DEPARTMENT STORE

PERSON crewing SAILBOAT crewed by

INTERPRETER fluent in LANGUAGE spoken by
Practice: Receipt

Served by: Dennis

Till: 3 Dec 8, 4:35 pm

-----------------------------------------------

CAPPUCC M  3.60
  * 2  7.20
CREAM       .75
  * 2  1.50
APPLE PIE   3.50
BLACKB MUF  4.50

<SUB>       16.70
tax 12%      2.00
<TOTAL>     18.70

========

CASH        20.00
RETURN      1.30

-----------------------------------------------

Hope to serve you again
@MOONLIGHT COFFEES
25 Phillis Rd, Atlanta
Practice: Moonlight P&O

- All Moonlight Coffee employees work for a department such as “Global Pricing” or “HQ”, or for a shop. All employees are at the payroll of one of our country organizations. Jill, for example, works as a shop manager in London; Werner is a financial administrator working for Accounting and is located in Germany.

- All shops belong to one country organization (“the countries”). There is only one country organization per country. All countries and departments report to HQ, except HQ itself.

- Employees can work part time. Lynn has had an 80% assignment for Product Development since the 1st September. Before that she had a full-time position.
## Practice: Price List

<table>
<thead>
<tr>
<th>Item</th>
<th>small</th>
<th>medium</th>
<th>large</th>
</tr>
</thead>
<tbody>
<tr>
<td>regular coffee</td>
<td>2.25</td>
<td>2.90</td>
<td>3.50</td>
</tr>
<tr>
<td>cappuccino</td>
<td>2.90</td>
<td>3.60</td>
<td>4.20</td>
</tr>
<tr>
<td>café latte</td>
<td>2.60</td>
<td>3.20</td>
<td>3.90</td>
</tr>
<tr>
<td>special coffee</td>
<td>3.10</td>
<td>3.70</td>
<td>4.40</td>
</tr>
<tr>
<td>espresso</td>
<td>2.25</td>
<td>2.90</td>
<td>3.50</td>
</tr>
<tr>
<td>coffee of the day</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>decaffeinated</td>
<td>.25</td>
<td>.50</td>
<td>.75</td>
</tr>
<tr>
<td>black tea</td>
<td>2.25</td>
<td>2.90</td>
<td>3.50</td>
</tr>
<tr>
<td>infusions</td>
<td>2.60</td>
<td>3.20</td>
<td>3.90</td>
</tr>
<tr>
<td>herbal teas</td>
<td>2.90</td>
<td>3.60</td>
<td>4.20</td>
</tr>
<tr>
<td>tea of the day</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>decaffeinated</td>
<td>.25</td>
<td>.50</td>
<td>.75</td>
</tr>
<tr>
<td>milk</td>
<td>1.25</td>
<td>1.90</td>
<td>2.50</td>
</tr>
<tr>
<td>soft drinks</td>
<td>2.25</td>
<td>2.90</td>
<td>3.50</td>
</tr>
<tr>
<td>soda water</td>
<td>2.25</td>
<td>2.90</td>
<td>3.50</td>
</tr>
<tr>
<td>mineral water</td>
<td>2.90</td>
<td>3.60</td>
<td>4.20</td>
</tr>
<tr>
<td>apple pie</td>
<td></td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>strawberry cheesecake</td>
<td></td>
<td></td>
<td>3.50</td>
</tr>
<tr>
<td>whole wheat oats muffin with almonds</td>
<td></td>
<td></td>
<td>3.90</td>
</tr>
<tr>
<td>blackberry muffin</td>
<td></td>
<td></td>
<td>4.50</td>
</tr>
<tr>
<td>fruitcake</td>
<td></td>
<td></td>
<td>4.50</td>
</tr>
<tr>
<td>cake of the day</td>
<td></td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>additional whipped cream</td>
<td></td>
<td></td>
<td>.75</td>
</tr>
</tbody>
</table>

Sales Tax included

September 16

25 Phillis Road, Atlanta

Visit us at [www.moonlight.com](http://www.moonlight.com)

Copyright © Oracle Corporation, 2002. All rights reserved.
Practice: E-Mail

- FOLDER
  - containing
  - placed in
- USER
  - written by
  - author of
  - part of
  - receiver of
- LIST
  - consists of
- COMPOSITION
  - reply of
- MESSAGE
  - received by
- OTHER COMPOSITION
  - containing
  - reply to
  - replied to by
- ATTACHMENT
  - containing
  - with <5
- ATT. TYPE
  - containing

ORACLE
Practice: Holiday

“Paul and I hiked in the USA. Eric and I hiked in France and we rented a car in the USA last year.”

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>TRANSPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Car</td>
</tr>
</tbody>
</table>

Paul and I hiked in the USA. Eric and I hiked in France and we rented a car in the USA last year.
Practice: Normalize an ER Model

ENROLLMENT
- grade code
- teacher number
- grade
- description
- course name

COURSE
- course number
- course name
- teacher number
- department code
- department name
- teacher name

STUDENT
- #* student id
- last name
- first name

for

completed with

assigned
4

Constraints
Overview

• Unique Identifiers
• Arcs
• Domains
• Various other constraints
Rembrandt
## Identification and Representation

<table>
<thead>
<tr>
<th>Name</th>
<th>Initials</th>
<th>Birthdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAPINI</td>
<td>G.</td>
<td>02-FEB-1954</td>
</tr>
<tr>
<td>HIDE</td>
<td>T.M.</td>
<td>11-JUN-1961</td>
</tr>
<tr>
<td>PAPINI</td>
<td>G.</td>
<td>02-FEB-1945</td>
</tr>
<tr>
<td>BAKER</td>
<td>S.J.T.</td>
<td>24-SEP-1958</td>
</tr>
</tbody>
</table>

G. Papini, please?
Unique Identifier Examples

**JOB**
Name

**COMPUTER IN NETWORK**
IP Address

**TELEPHONE**
Country code, Area code, Telephone number

**EMPLOYEE**
Employee number or Name, Initials, Birth Date

**MAIL LIST**
Name, Owner
Unique Identifier

ORDER

by

responsible for

CUSTOMER

# Family Name

o Initials

# Address

o Telephone

Indicates Unique Identifier

Indicates Unique Identifier
Unique Identifiers

- USER # Name
- MAIL LIST # Name
- ROOM # No
- FLOOR # No
- HOTEL # Name

- part of
- owner of
- contains
- owned by
Multiple Relationship UID

USER # Name
- part of
  - contains
    - LIST # Name

USER # Name
- owner of
  - owned by
    - LIST # Name

USER # Name
- referred to
  - containing
    - LIST ITEM

LIST # Name
- contains
  - containing
    - LIST ITEM

LIST # Name
- contains
  - containing
    - LIST ITEM
Well-defined Unique Identifiers
Incorrect Unique Identifiers
Information-Bearing Codes

54.0.093.81

Product Group
In Production?
Factory
Sequence Number

PRODUCT
# Code
* In Production?
* Sequence No

PRODUCT GROUP
# Code

FACTORY
# Id
“A contract consists of contract components; these are standard conditions or customized conditions”
Exclusive Arc
Possible Arc Constructs
Some Incorrect Arc Constructs

- The arc “belongs” to one entity
- Relationships in the arc must be of the same optionality
- Arcs must contain at least two relationships

An arc may be correct, but is quite difficult to implement ...
## Some Incorrect Arc Constructs

<table>
<thead>
<tr>
<th>Number of Valid Relationships in Arc Per Entity Instance</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="538x36" alt="Arc Diagram" /> ({n})</td>
<td>(n)</td>
<td>(n)</td>
</tr>
<tr>
<td><img src="538x36" alt="Arc Diagram" /> ({n})</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><img src="538x36" alt="Arc Diagram" /> ({n})</td>
<td>0</td>
<td>(n)</td>
</tr>
<tr>
<td><img src="538x36" alt="Arc Diagram" /> ({n})</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Arc or Subtype

USER

owned by

LIST

is referred to
contains

LIST ITEM

is referred to
referring to

USER

owned by

ADDRESS

contains

LIST

is referred to
referring to

LIST ITEM

referring to
Arc and Subtypes

1

A

R

P

Q

2

A

P

Q

3

A

B

C

P

Q

4

A

B

C

R

P

Q

5

A

B

C

R

P

Q
Subtypes Hide Relationships in Arc

- Every A is either a B or a C
- Every B is an A
- Every C is an A

- Every A must be a B or be a C
- Every B must be an A
- Every C must be an A
Value sets

- YESNO
  - Code
  - Description

- GENDER
  - Code
  - Description

- WEEKDAY
  - Code
  - Description

- CODE TYPE
  - Id
  - Name
  - Max Length of Description

- CODE
  - Code
  - Description
Other Constraints: Range Check

- EMPLOYEE
  * Name
  * Address

- JOB
  * Title
  * Minimum Salary
  * Maximum Salary
  - between

- EMPLOYMENT
  * Start Date
  - End Date
  * Salary

- referring to
  - for

- with
Other Constraints: State Value Transition

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>* Name * Address * Current Marital Status</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Possible Marital Status Transitions</th>
<th>Sin</th>
<th>Mar</th>
<th>Wid</th>
<th>Div</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Married</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Widowed</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divorced</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domestic Partnership</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Possible transitions: from Single to Married, from Married to Widowed, from Widowed to Divorced, from Divorced to Domestic Partnership.
Conditional Relationship

CONTRACT
# Id * Standard Indicator

consists of

part of

referring to

STANDARD CONDITION

basis for

based on

in

CUSTOMIZED CONDITION

in

referring to

CONTRACT COMPONENT
Boundaries

unrelated entity

and possible implementation

EXTERNALS

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Value added tax %</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Maximum available Space per Mail User in Mbyte</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>Maximum level of Nested Mail Folders</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Maximum level of Nested Mail Lists</td>
<td>16</td>
</tr>
</tbody>
</table>
Summary

- **Identification**
  - Can be a real problem in the real world
  - Models cannot overcome this
- **Entities must have at least one Unique Identifier**
- **Unique Identifiers consist of attributes or relationships or both**
- **Arcs**
- **Many types of constraint are not represented in ER model**
Practices

- Identification Please
- Identification
- Moonlight UID
- Tables
- Modeling Constraints
Practice: Identification Please

- A city
- A contact person for a customer
- A train
- A road
- A financial transaction
- An Academy Award (Oscar)
- A painting
- A T.V. show
Practice: Identification 1
Practice: Identification 2
Practice: Identification 3

A
* Xx
B
# Yy
C
# Zz

with

of

D
# Id
Practice: Identification 4
Practice: Identification 5
Practice: Identification 6

PERSON

MALE
# Seqno

son of

partner in

with husband

MARRIAGE
# Start Date

FEMALE
# Name
# Birth Date

mother of

partner in

with wife
Practice: Moonlight UID
“In a relational database system, data is stored in tables. Tables of a database user must have a unique name. A table must have at least one column. A column has a unique name within the table. A column must have a data type and may be Not Null.

Tables can have one primary key and any number of unique keys. A key contains one or more columns of the table. A column can be part of more than one key.
Practice: Table 1

A table can have foreign keys. A foreign key always connects one table with another. A foreign key consists of one or more columns of the one table that refers to key columns of the other table. “The sequence of columns within the key and foreign key is important.”
Practice: Table 2

```
FOREIGN KEY
#  Name
with

ASSOCIATION
#  Seqno
for

from
with

to

referenced
in

with

TABLE
#  Name

with

from

in

COLUMN
#  Name
*  Data Type
o  Not Null

for

in

in

USAGE
#  Seqno

for

KEY
#  Name

with

of

PRIMARY

with

of

UNIQUE

with

of
```

Practice: Table 2
Practice: Constraints 1
Practice: Constraints 2

- USER
  - # Name
  - owned by
  - owner of

- LIST
  - # Name
  - owned by
  - owner of

- NICKNAME
  - # Alias
  - owned by
  - owner of
Practice: Constraints 3

FOLDER Name

with subfolder

within

owned by

owner of

USER # Name
Modeling Change
Overview

- Date and time
- Modeling change over time
- Prices change
- Journaling
Change and Time

- Every update means loss of information.
- Time in your model makes the model more complex.
- There are often complex join conditions.
- Users can work in advance.
- When do you model date/time as an entity?
- What constraints do arise?
- How do you handle journaling?
Entity DAY

DAY
  # Date
  * Public Holiday Indicator

starts on

first day of

TASK ASSIGNMENT
  * Duration in Hours

for

in

of

with

TASK
  # Id

EMPLOYEE
  # Name

Copyright © Oracle Corporation, 2002. All rights reserved.
Modeling Change

```plaintext
EMPLOYEE
# Id

COUNTRY
# Name

of

in

for

as

ASSIGNMENT
# Start Date
# End Date
```
Even a Country Has a Life Cycle

Even a Country Has a Life Cycle

EMPLOYEE
# Id

of

COUNTRY
# Name
# Start Date
* End Date

in

life cycle attributes

ASSIGNMENT
# Start Date
o End Date

for

as
Products and Prices

PRODUCT
# Id
* Name

with

PRICE
* Price in $
# Start date
○ End Date

PRICE = PRICED PRODUCT = HISTORICAL PRICE
### What Price to Pay?

<table>
<thead>
<tr>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td># Id</td>
</tr>
<tr>
<td>* Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Price in $</td>
</tr>
<tr>
<td># Start date</td>
</tr>
<tr>
<td>° End Date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORDER HEADER</th>
</tr>
</thead>
<tbody>
<tr>
<td># Id</td>
</tr>
<tr>
<td>* Order Date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORDER ITEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Quantity Ordered</td>
</tr>
</tbody>
</table>

**Diagram:**
- **PRODUCT** with **PRICE** referred by **ORDER HEADER** with **ORDER ITEM**
- **PRODUCT** with **PRICE** referring to **ORDER HEADER**

---

Copyright © Oracle Corporation, 2002. All rights reserved.
Price List Search

PRICE LIST
- # Id
- * Start Date
- ◦ End Date

PRODUCT
- # Id
- * Name

ORDER HEADER
- # Id
- * Order Date

PRICED PRODUCT
- * Price in $
Order for Priced Products

**PRICE LIST**
- # Id
- * Start Date
- ◦ End Date

**PRODUCT**
- # Id
- * Name

**PRICED PRODUCT**
- * Price in $

**ORDER HEADER**
- # Id
- * Order Date

**ORDER ITEM**
- * Quantity Ordered

*with* refers to

*referred by* refers to

*referring to*

*of* refers to
Negotiated Prices

- **PRICE LIST**
  - # Id
  - * Start Date
  - ◦ End Date
  - with

- **PRODUCT**
  - # Id
  - * Name
  - with

- **PRICED PRODUCT**
  - * Price in $
  - on
  - of

- **ORDER HEADER**
  - # Id
  - * Order Date
  - with

- **ORDER ITEM**
  - * Quantity Ordered
  - referring to
  - * Negotiated Price
  - of
Journaling

PAYMENT
  * Date Paid
  * Amount in $

by
to

PAYMENT
  * Date Paid
  * Amount in $

with

AMOUNT
MODIFICATION
  * Old Amount in $
  * Modified by
  * Date Modification

of
to

by
Summary

- Consider the need for keeping old values
- Time in your model is complicated:
  - Implicit versions
  - References
- Journaling
Practices

- Shift
- Strawberry Wafer
- Bundles
- Product Structure
## Practice: Shift

**Museumplein, Amsterdam, March 21**

<table>
<thead>
<tr>
<th>Shift</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>6:30</td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>23:00</td>
<td>-</td>
</tr>
<tr>
<td>Tue</td>
<td>7:00</td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>23:00</td>
<td>-</td>
</tr>
<tr>
<td>Wed</td>
<td>7:00</td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>23:00</td>
<td>-</td>
</tr>
<tr>
<td>Thu</td>
<td>7:00</td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>23:00</td>
<td>-</td>
</tr>
<tr>
<td>Fri</td>
<td>7:00</td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>16:00</td>
<td>20:30</td>
<td>24:00</td>
<td>-</td>
</tr>
<tr>
<td>Sat/Sun</td>
<td>8:00</td>
<td>11:30</td>
<td>15:00</td>
<td>18:00</td>
<td>21:00</td>
</tr>
<tr>
<td></td>
<td>11:30</td>
<td>15:00</td>
<td>18:00</td>
<td>21:00</td>
<td>24:00</td>
</tr>
</tbody>
</table>
Practice: Strawberry Wafer

- Prices are at the same level within a country; prices are determined by the Global Pricing Department. Usually the prices for regular, global products are re-established once a year.

- Prices and availability for local specialties are determined by the individual shops. For example, the famous Norwegian Vafler med Jordbær (a delicious wafer with fresh strawberries) is only available in summer. Its price depends on the current local market price of fresh strawberries.
<table>
<thead>
<tr>
<th>Item</th>
<th>klein</th>
<th>middel</th>
<th>groot</th>
</tr>
</thead>
<tbody>
<tr>
<td>gewone koffie</td>
<td>60</td>
<td>90</td>
<td>120</td>
</tr>
<tr>
<td>cappuccino</td>
<td>90</td>
<td>110</td>
<td>140</td>
</tr>
<tr>
<td>koffie verkeerd</td>
<td>75</td>
<td>100</td>
<td>130</td>
</tr>
<tr>
<td>speciale koffies</td>
<td>99</td>
<td>125</td>
<td>150</td>
</tr>
<tr>
<td>espresso</td>
<td>60</td>
<td>95</td>
<td>110</td>
</tr>
<tr>
<td>koffie van de dag</td>
<td>45</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>caffeine vrij</td>
<td>5</td>
<td>10</td>
<td>15 toeslag</td>
</tr>
<tr>
<td>zwarte thees</td>
<td>60</td>
<td>100</td>
<td>120</td>
</tr>
<tr>
<td>vruchten thees</td>
<td>75</td>
<td>110</td>
<td>130</td>
</tr>
<tr>
<td>kruiden thees</td>
<td>80</td>
<td>120</td>
<td>140</td>
</tr>
<tr>
<td>dag thee</td>
<td>50</td>
<td>85</td>
<td>100</td>
</tr>
<tr>
<td>caffeine vrij</td>
<td>5</td>
<td>10</td>
<td>15 toeslag</td>
</tr>
<tr>
<td>frisdranken</td>
<td>60</td>
<td>100</td>
<td>130</td>
</tr>
<tr>
<td>diverse sodas</td>
<td>60</td>
<td>100</td>
<td>130</td>
</tr>
<tr>
<td>mineraal water</td>
<td>75</td>
<td>120</td>
<td>140</td>
</tr>
<tr>
<td>appel taart</td>
<td></td>
<td></td>
<td>180</td>
</tr>
<tr>
<td>brusselse wafel</td>
<td></td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>portie chocolade bonbons</td>
<td></td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>koekje van eigen deeg</td>
<td></td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>portie slagroom</td>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

inclusief BTW 16 September
Practice: Bundles(1)

A SweetTreat™ consists of a large soft drink plus cake of the day.
A BigBox™ consist of a large coffee of the day plus two cakes of the day.
A SuperSweetTreat™ consists of a SweetTreat™ plus whipped cream (on the cake).
A FamilyFeast™ consists of two BigBoxes™ plus two SweetTreats™ plus a small surprise.

A DecafPunch™ consists of a regular decaffeinated coffee or a regular decaffeinated tea, plus a blackberry muffin.
Practice: Bundles(2)

PRODUCT GROUP
#  Name

classification
for

classified
as

PRODUCT
#  Id
*  Name
Practice: Product Structure

+ Products
  + Drinks
    + Coffees
      Regular
      Cappuccino
      Café Latte
      + Special Coffee
    + Teas
      + Black
      Chinese
      Indian
      English
      + Infusions
      + Herbal
    + Soft drinks
      + Juices
        Orange
        Grape
      + Waters
      + Sodas
    + Dairy Products
  + Foods
    + Pastry
    + Candy Bars
    + Local Specialties
  + Non Foods
    + Merchandise
      CDs
      + Stationary
    Other
      + Tickets
      + Art
Advanced Modeling Topics
Overview

• Patterns
• Drawing conventions
• Generic modeling
Patterns

- Similar structure
- Similar rules and constraints?
Patterns: Master–Detail

- **Characteristic:** *consists of*
  An instance of B only exists in the context of an A
- **Metaphor:** Master – Detail
Pattern: Basket

- **Characteristic:**
  container for various types of items

- **Items may be of different types**

- **Metaphor: Shopping Basket**

  - A
  - B
  - X
  - Y
  - Z
Patterns: Hierarchy

- **Characteristic:** manager of / subordinate of
- **Additional constraints to guard hierarchical nature**
- **Metaphor:** Mother–Child
Patterns: Chain

- Characteristic: *preceded by / followed by*
- Sequence is important
- Metaphor: Elephants
Patterns: Network

- **Characteristic: pairs**
  Every A can be connected to every A (sometimes: to every other A)
- **Metaphor: Web Document with Hyperlinks**
Bill of Material

PRODUCT
# Code

product of
with
part in
in

COMPOSITION
* Quantity Needed

PRODUCTS

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>914.53</td>
<td>AAAAAAAA</td>
</tr>
<tr>
<td>914.54</td>
<td>AAA</td>
</tr>
<tr>
<td>914.55</td>
<td>BBBBBBBBB</td>
</tr>
<tr>
<td>914.56</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>DDDDDD</td>
</tr>
</tbody>
</table>

COMPOSITIONS

<table>
<thead>
<tr>
<th>Prod_code</th>
<th>Part_code</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>854.01</td>
<td>604.18</td>
<td>1</td>
</tr>
<tr>
<td>854.01</td>
<td>604.19</td>
<td>1</td>
</tr>
<tr>
<td>854.01</td>
<td>914.54</td>
<td>2</td>
</tr>
<tr>
<td>914.54</td>
<td>914.55</td>
<td>1</td>
</tr>
<tr>
<td>914.54</td>
<td>914.56</td>
<td>1</td>
</tr>
<tr>
<td>914.54</td>
<td>915.12</td>
<td>1</td>
</tr>
</tbody>
</table>
Bill of Material - Example

- 854.01
- 914.54
- 914.54
- 604.18
- 604.19
- 914.55
- 914.56
Symmetric Relationship

<table>
<thead>
<tr>
<th>Group_id</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S₁</td>
</tr>
<tr>
<td>1</td>
<td>S₂</td>
</tr>
<tr>
<td>2</td>
<td>S₃</td>
</tr>
<tr>
<td>2</td>
<td>S₄</td>
</tr>
<tr>
<td>3</td>
<td>S₅</td>
</tr>
<tr>
<td>3</td>
<td>S₆</td>
</tr>
</tbody>
</table>
Patterns: Roles

- **Characteristic:** *is / is* 1:m (or 1:1) relationships
- **Metaphor:** Person–Many Hats
  (not necessarily concurrently...)

![Diagram showing A connected to P and Q, with multiple hats on a head.](image)
Roles

- PERSON
  - PRESIDENT
    - appointed
      - MINISTER
        - appointed
          - PARTY LEADER
    - appointing
      - COUNTRY
      - DEPARTMENT
  - PARTY

ROLE
- PERSON
- ROLE
- ROLE TYPE
Fan Trap

- **Characteristic:** ring of m:m related entities
- **Metaphor:** ABC Combination
Fan Trap Resolved

AB functions as list of values

BC functions as list of values
Patterns: Data Warehouse

- Characteristic: *multidimensional*, many, many detail instances
- Metaphor: *star* model
  Stars may be strangely shaped:
- Snowflake model
Not important *which* convention you choose, as long as you follow one of them.
Use Conventions Sensibly

But:
Readability first
Model Readability

- Takes space
- Subject to taste
Generic Modeling

MANUFACTURER
- Name

FILM
- Asa

TRIPOD
- Height

LENS
- Focal Distance

CAMERA BODY
- Weight

ARTICLE
- Weight
- Focal Distance
- Height
- Asa Number
- ...

ARTICLE TYPE
Generic Modeling

ARTICLE TYPE
* Definition Prop1
o Definition Prop2
o Definition Prop3
o Definition Prop4
...

ARTICLE
o Property1
o Property2
o Property3
o Property4
o Property5
o Property6
o Property7
o Property8

MANUFACTURER
* Name
Generic Model

ARTICLE TYPE

ARTICLE

PROPERTY

ARTICLE PROPERTY VALUE

o Value
Generic

having some kind of relationship with

having some kind of relationship with

THING
More Generic

THING

ASSOCIATION
More Generic Plus
Best of Two Worlds

CUSTOMER

ORDER HEADER

ORDER ITEM

ˈdown to earthˈ

ˈgenericˈ

ARTICLE TYPE

ARTICLE

PROPERTY

ARTICLE PROPERTY VALUE

ORACLE
Summary

Patterns
- Show similarities
- Invent your wheel only once

Generic models
- Reduce the number of entities dramatically
- Are more complex to implement
- Are very flexible
- Are usually the best choice in unstable situations
Practices

- Patterns
- Data Warehouse
- Argos and Erats
- Synonym
Practice: Patterns

- Model of moves in a chess game
- Model of tenders (quotations)
- Model of recipes
- Model of all people involved in college: students, teachers, parents, ...
- Rentals in a video shop
- Model of phases in a process
Practice: Data Warehouse

- What is the sales volume in $ of coffee last month compared with the coffee sales volume same month last year?
- What is the sales volume in $ of coffee per head in Japan compared with the average coffee sales volume in the Moonlight countries around the world?
- What is the growth of the sales volume in $ of coffee in Sweden compared with the growth of sales volume of all products in the same geographical area? What is the growth in local currency?
Practice: Data Warehouse

- What was the total sales volume in $ of coffee last month, compared with the total coffee sales volume in the same month last year, for the shops that have been open for at least 18 months?
- What is the growth of the sales volume in $ of nonfoods compared to that of foods?
- What is the best day of the week for total sales in the various countries? How is that related to the average? Is the best day of the week dependent on the type of location?
Practice: Data Warehouse

- What products are most profitable per country? Globally?
- Does the service level (#employees per 1000 items sold) have influence on sales?
Practice: Argos and Erats

"Erats have names that are unique. Erats can have argos. Argos have names as well. The name of an argo must be unique within the erat it belongs to. Erats mutually have rondels. There are only a few different types of rondels. Erats can have one or more ubins. A ubin always consists of one or more argos of the erat, one or more rondels of the erat, or combinations of the two."
Practice: Synonym

practice - exercise
order - command
entity - being
order - sequence
order - arrangement
command - demand
Mapping the Entity Model
Overview

- Why use design modeling?
- Introduction to the components:
  - Tables
  - Columns
  - Constraints
- Basic Mapping
- Complex mapping
Why Create a Data Design Model?

- Closer to the implementation solution
- Facilitates discussion
- Ideal model can be adapted to an RDBMS model
- Sound basis for physical database design
Presenting Tables

Table: EMPLOYEES

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Address</th>
<th>Birth_date</th>
<th>Dpt_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>PAGE</td>
<td>12, OXFORD ST</td>
<td>03-03-66</td>
<td>10</td>
</tr>
<tr>
<td>349</td>
<td>PAPINI</td>
<td>53, HAYES AVE</td>
<td>10-08-77</td>
<td>20</td>
</tr>
<tr>
<td>785</td>
<td>GARRET</td>
<td>08-12-55</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

Table diagram: EMPLOYEES
Transformation Process

Conceptual Model

Relational Model
Terminology Mapping

ANALYSIS  
ER Model

ENTITY  
Attribute

Primary UID  
Secondary UID

Relationship

Business Constraints

DESIGN  
Physical Design

TABLE  
Column

Primary Key  
Unique Key

Foreign Key  
Check Constraints
General Naming Topics

Decide on a convention for:

- Table names
- Special characters (%, *, #, -, space, …)
- Table short names
- Column names
- Primary and Unique Key Constraint names
- Foreign Key Constraint names
- Foreign Key Column names
Naming Restrictions with Oracle

- Table and column names:
  - Must start with a letter
  - May contain up to 30 alphanumeric characters
  - Cannot contain space or some special characters such as “!”
- Table names must be unique within a schema
- Column names must be unique within a table
Basic Mapping for Entities

1 - Entities

Table Name: EMPLOYEES
Short Name: EPE

EMPLOYEE

EMPLOYEES (EPE)
Basic Mapping for Attributes

1 - Entities
2 - Attributes

Table Name: EMPLOYEES
Short Name: EPE

EMPLOYEES (EPE)

<table>
<thead>
<tr>
<th></th>
<th>Id</th>
<th>Name</th>
<th>Address</th>
<th>Birth_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Id</td>
<td>Name</td>
<td>Address</td>
<td>Birth_date</td>
</tr>
<tr>
<td>o</td>
<td>Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Birth_date</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Mapping for Unique Identifiers

1 - Entities
2 - Attributes
3 - Unique identifiers

Table Name: EMPLOYEES
Short Name: EPE

<table>
<thead>
<tr>
<th>EMPLOYEES (EPE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
</tr>
<tr>
<td>uk₁</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>uk₁</td>
</tr>
</tbody>
</table>
Rules for Relationships

EMPLOYEE
# Id
* Name
o Address
* Birth Date

DEPARTMENT
# Id
* Name

EMPLOYEES (EPE)
pk
* Id
* Name
m Dpt_id
f Dpt_id

DEPARTMENTS (DPT)
pk
* Id
* Name

fk1 = epe_dpt_fk
fk2 = epe_epe_fk
Mapping 1:m Relationships

Diagram showing the mapping of a 1:m relationship between two entities: XS and Y_id. The diagram illustrates how a single entity XS can be related to multiple instances of Y_id through a foreign key (fk) relationship.
Mapping Barred and Nontransferable Relationships

\[ fk = y_x_fk \]
Mapping Cascade Barred Relationships

A
# Id
* C1

B
# Id
* C2

C
# Id
* C3

D
# Id
* C4

AS (A)
pk
* * C1

BS (B)
pk
* * C2
fk,pk
* * A_id

CS (C)
pk
* * C3
fk,pk
* * B_id
fk,pk
* * B_a_id

DS (D)
pk
* * C4
fk
* * C_id
fk
* * C_b_id
fk
* * C_a_id

fk = b_a_fk
fk = c_b_fk
fk = d_c_fk
Mapping m:m Relationships

\[ \text{fk1} = xy_x_fk \]
\[ \text{fk2} = xy_y_fk \]
Mapping 1:1 Relationships

Choose which side for FK for other cardinalities
Mapping Arcs

Explicit implementation

LIST ITEMS (LIM)

- \( pk, fk_1 \)
- \( * \)
- \( X_id \)
- \( 0 \)
- \( Usr_id \)
- \( 0 \)
- \( Als_id \)

ALIASES (ALS)

- \( pk \)
- \( * \)
- \( Id \)

ALIASES (ALS)

- \( * \)
- \( Name \)

USERS (USR)

- \( pk \)
- \( * \)
- \( Id \)
- \( Name \)

Explicit implementation

\( fk1 = lim_x_fk \)

\( fk2 = lim_usr_fk \)

\( fk3 = lim_als_fk \)

+ check constraint
Mapping Subtypes

Variety of implementation choices

- Supertype
- Subtype
- Both Supertype and Subtype (“Arc”)
Supertype Implementation

- Mandatory discriminator column
- Additional constraints

PS (P)

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>Xxx</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>Yyy</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>Zzz</td>
</tr>
<tr>
<td>fk₁</td>
<td>*</td>
<td>A_id</td>
</tr>
<tr>
<td>fk₂</td>
<td>o</td>
<td>B_id</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>P_type</td>
</tr>
</tbody>
</table>
Subtype Implementation

P
# Id
* Xxx

Q
o Yyy

R
* Zzz

L
# Id

K
# Id

A
# Id

B
# Id

QS (Q)

Q
pk
* *
Id
* 
Xxx

o
Yyy

fk
* 
A_id

RS (R)

R
pk
* *
Id
* 
Xxx

fk
* *
Zzz

fk
* *
A_id

fk
* *
B_id

q_a_fk

fk_1=r_a_fk

fk_2=r_b_fk
Supertype and Subtype (Arc) Implementation

Supertype (P):
- # Id
- * Xxx
- Q
  - o Yyy
- R
  - * Zzz
- L
  - # Id

Subtype (Q):
- pk
- fk
- fk_1
  - uk
  - 1
- fk_2
  - uk
  - 2

Subtype (R):
- pk
- fk
- 1
- B_id

Relational Schemata (RS):
- pk
- fk
- 1
- B_id

Query Schemata (QS):
- pk
- fk
- 1
- Yyy

Implementation:
- fk_1 = p_q_fk
- fk_2 = p_r_fk
- fk_3 = p_a_fk
- r_b_fk
Storage Implication

1 table
2 tables
3 tables
Storage Implication
Supertype Implementation

discriminator column

cols P ↔ cols Q ↔ cols R

rows Q

rows R
Storage Implication
Subtype Implementation
Storage Implication
Supertype and Subtype (Arc) Implementation
Summary

- Relational concepts
- Naming rules convention
- Basic mapping
- Complex mapping
Practice

- Mapping basic Entities, Attributes and Relationships
- Mapping Supertype
- Quality Check
  Subtype Implementation
- Quality Check
  Supertype and Subtype (Arc) Implementation
- Mapping Primary Keys and Columns
Practice: Mapping basic Entities, Attributes and Relationships

**EMPLOYEE**
- # Id
- * First Name
- * Last Name
- * Date of Birth
- * Home Phone

**DEPARTMENT**
- # Id
- * Name
- * Location

**Relationships**
- **assigned to**
- **responsible for**
# Practice: Mapping Supertype

## DEPARTMENT

<table>
<thead>
<tr>
<th># Id</th>
<th>* Name</th>
<th>* Head Count</th>
</tr>
</thead>
</table>

## HQ

- Address

## COUNTRY ORGANIZATION

- # Tax Id Number

## DEPARTMENTS ( )
Partial ER model Moonlight

- **COUNTRY**
  - # Code

- **PRODUCT GROUP**
  - # Name

- **PRODUCT**
  - # Name
  - # Code
  - Size

- **PRICE LIST**
  - # Start Date
  - # End Date

- **GLOBAL PRICE**
  - * Amount

- **LOCAL**
  - # Name

- **SHOP**
  - # No
  - * Name
  - * Address
  - * City

For

With

With

Partial ER model Moonlight
**Practice: Quality Check**

**Subtype Implementation**

<table>
<thead>
<tr>
<th>GLOBAL_PRODUCTS (GPT)</th>
<th>LOCAL_PRODUCTS (LPT)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pk</strong></td>
<td><strong>pk</strong></td>
</tr>
<tr>
<td>*</td>
<td>#</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td><strong>Shop_no</strong></td>
</tr>
<tr>
<td><strong>Pgp_name</strong></td>
<td><strong>Pgp_name</strong></td>
</tr>
</tbody>
</table>

- `lpt_shop_fk`
Practice: Quality Check
Arc Implementation

**PRODUCTS (PDT)**

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>fk₁</td>
<td>*</td>
<td>Pgp_name</td>
</tr>
<tr>
<td>fk₂</td>
<td>*</td>
<td>Gpt_code</td>
</tr>
<tr>
<td>fk₃</td>
<td>*</td>
<td>Lpt_name</td>
</tr>
</tbody>
</table>

FKs:
- \( fk₁ = \text{pdt}_\text{pgp}_\text{name} \)
- \( fk₂ = \text{pdt}_\text{gpt}_\text{code} \)
- \( fk₃ = \text{pdt}_\text{lpt}_\text{name} \)

**GLOBAL_PRODUCTS (GPT)**

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td></td>
<td>Size</td>
</tr>
</tbody>
</table>

FK:
- \( gpt\_pgp\_fk \)

**LOCAL_PRODUCTS (LPT)**

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk₁</td>
<td>o</td>
<td>Shp_no</td>
</tr>
<tr>
<td>fk₁</td>
<td>*</td>
<td>Pgp_name</td>
</tr>
</tbody>
</table>

FKs:
- \( fk₁ = \text{shp}_\text{lpt}_\text{fk} \)
- \( fk₂ = \text{pgp}_\text{lpt}_\text{fk} \)
Practice: Mapping Primary Keys and Columns

<table>
<thead>
<tr>
<th>GLOBAL_PRICES (     )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Denormalized Data
Overview

- Denormalization
- Benefits
- Types of denormalization
Denormalization Overview

Denormalization

- Starts with a “normalized” model
- Adds “redundancy” to the design
- Reduces the “integrity” of the design
- Application code added to compensate
Denormalization Techniques

- Storing Derivable Values
- Pre-joining Tables
- Hard-Coded Values
- Keeping Details with Master
- Repeating Single Detail with Master
- Short-Circuit Keys
Storing Derivable Values

Add a column to store derivable data in the “referenced” end of the foreign key.
EMail Example of Storing Derivable Values

Before

STORE derivable column in the ‘referenced’ end of the foreign key.

After

MESSAGES (MSE)

pk * Id
* Subject
* Text
* Number_of_times_received
Pre-Joining Tables

Before

Add the non_key column to the table with the foreign key.

After
EMail Example of Pre-Joining Tables

Before

<table>
<thead>
<tr>
<th>FOLDERS (FDR)</th>
<th>RECEIVED_MESSAGES (RME)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
<td>pk, fk</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Id</td>
<td>Mse_id</td>
</tr>
<tr>
<td>Name</td>
<td>Flr_id</td>
</tr>
<tr>
<td></td>
<td>Date_received</td>
</tr>
</tbody>
</table>

Create a table with all the frequently queried columns.

After

<table>
<thead>
<tr>
<th>RECEIVED_MESSAGES (RME)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>Mse_id</td>
</tr>
<tr>
<td>Flr_id</td>
</tr>
<tr>
<td>Date_received</td>
</tr>
<tr>
<td>Fdr_Name</td>
</tr>
</tbody>
</table>
Hard-Coded Values

Before

\[
\begin{array}{ccc}
\text{A} & \text{B} \\
pk & \text{pk} & \text{fk} \\
* & \ast & * \\
Id & \ast & \ast \\
Type & \text{Id} & \text{A\_i}
\end{array}
\]

Remove the foreign key and hard code the allowable values and validation in the application.

After

\[
\begin{array}{ccc}
\text{B} \\
pk & \ast \\
\ast & \ast \\
Id & \ast \\
A\_Type & \text{A\_Type}
\end{array}
\]
Email Example of Hard-Coded Values

Before

**BUSINESS_TYPES**

pk | * | Id  
---|---|-----
   |   | Name

**USERS (USR)**

pk | * | Id  
---|---|-----
fk | * | Bte_id  
   |   | Per_name

Hard code the allowable values and validation in the application.

After

**USERS (USR)**

pk | * | Id  
---|---|-----
   |   | Business_type  
   |   | Per_name
Keeping Details with Master

Before

Add the repeating detail columns to the master table.

After
EMail Example Keeping Detail with Master

Before

<table>
<thead>
<tr>
<th>USERS (USR)</th>
<th>STORAGE_QUOTAS (SQA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, *</td>
<td>* Usr_Id</td>
</tr>
<tr>
<td>pk</td>
<td>* Storage_type</td>
</tr>
<tr>
<td>*</td>
<td>* Allocated</td>
</tr>
<tr>
<td>*</td>
<td>* Available</td>
</tr>
</tbody>
</table>

Add the repeating detail columns to the master table.

After

<table>
<thead>
<tr>
<th>USERS (USR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*</td>
</tr>
</tbody>
</table>
Repeating Current Detail with Master

Before

Add a column to the master to store the most current details.

After

* Current_price
**EMAIL Example of Repeating Single Detail with Master**

**Before**

<table>
<thead>
<tr>
<th>MESSAGES (MSE)</th>
<th>ATTACHMENTS (ATT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>pk, fk</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Id</td>
<td>Id</td>
</tr>
<tr>
<td>Subject</td>
<td>Mse_id</td>
</tr>
<tr>
<td>Text</td>
<td>Name</td>
</tr>
</tbody>
</table>

Add a column to the master to store the most current details.

**After**

<table>
<thead>
<tr>
<th>MESSAGES (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>Id</td>
</tr>
<tr>
<td>First_attachment_name</td>
</tr>
<tr>
<td>Subject</td>
</tr>
<tr>
<td>Text</td>
</tr>
</tbody>
</table>
**Short-Circuit Keys**

Before

Create a new foreign key from the lowest detail to the highest master.

After

Create a new foreign key from the lowest detail to the highest master.
EMail Example of Short-Circuit Keys

Before

CREATE A NEW FOREIGN KEY FROM THE LOWEST DETAIL TO THE HIGHEST MASTER.

After
Add an *end date* column to speed up queries so that they can use a *between* operator.
Example of End Date Column

Create an extra column derivable End_date column.

Before

PRODUCTS (PDT)

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
</table>

PRICES (PCE)

<table>
<thead>
<tr>
<th>pk, fk</th>
<th>*</th>
<th>Pdt_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>*</td>
<td>Start_date</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>Price</td>
</tr>
</tbody>
</table>

After

PRICES (PCE)

<table>
<thead>
<tr>
<th>pk, fk</th>
<th>*</th>
<th>Pdt_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>*</td>
<td>Start_date</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>Price</td>
</tr>
<tr>
<td>o</td>
<td>*</td>
<td>End_date</td>
</tr>
</tbody>
</table>
Current Indicator Column

Before

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>pk, fk</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Id</td>
<td>A_id</td>
</tr>
<tr>
<td></td>
<td>Start_date</td>
</tr>
</tbody>
</table>
```

Add a column to represent the most current record in a long list of records.

After

```
<table>
<thead>
<tr>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
</tr>
<tr>
<td>pk</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>A_Id</td>
</tr>
<tr>
<td>Start_date</td>
</tr>
<tr>
<td>Current_indicator</td>
</tr>
</tbody>
</table>
```
Example of Current Indicator Column

Before

PRODUCT (PDT)

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Id</th>
<th>Name</th>
</tr>
</thead>
</table>

PRICES (PCE)

<table>
<thead>
<tr>
<th>pk,fk</th>
<th>*</th>
<th>Pdt_id</th>
<th>Start_date</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Add a column to represent the most current record, in a long list of records.

After

PRICES (PCE)

<table>
<thead>
<tr>
<th>pk,fk</th>
<th>*</th>
<th>Pdt_id</th>
<th>Start_date</th>
<th>Price</th>
<th>Current_indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>o</td>
</tr>
</tbody>
</table>
Create a column to represent the hierarchy level of a record.
Example of Hierarchy Level Indicator

Create a column to represent the hierarchy level of a record.
Denormalization Summary

Denormalization Techniques

- Storing Derivable Information
  - End Date Column
  - Current Indicator
  - Hierarchy Level Indicator

- Pre-Joining Tables
- Hard-Coded Values
- Keeping Detail with Master
- Repeating Single Detail with Master
- Short-Circuit Keys
Practices

- Name that Denormalization
- Triggers
- Denormalize Price Lists
- Global Naming
## Practice: Name that Denormalization (1/3)

### WEEKDAYS (WDY)
```
| pk |   | Code | Name |
```

### SHIFTS (SFT)
```
| pk | fk | No | Wdy_code | Start_time | End_time | Wdy_name |
```
Practice: Name that Denormalization (2/3)
Practice: Name that Denormalization (3/3)
Practice: Triggers (1/6)
## Practice: Triggers (2/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHR</td>
<td>Insert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Order_total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OIM</td>
<td>Insert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Ohr_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Item_total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Practice: Triggers (3/6)
## Practice: Triggers (4/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCN</td>
<td>Insert</td>
<td>Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Address</td>
<td></td>
<td>other cols</td>
</tr>
<tr>
<td>EPE</td>
<td>Insert</td>
<td>Lcn_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Lcn_address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Practice: Triggers (5/6)
### Practice: Triggers (6/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDT</td>
<td>Insert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCE</td>
<td>Insert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Pdt_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start_date</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>End_date</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Curr_price_Ind</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Practice: Denormalize Price Lists

- Speed up performance for queries on Amount.
- Insert new price lists before their effective date.

<table>
<thead>
<tr>
<th>PRICE_LISTS (PLT)</th>
<th>GLOBAL_PRICES (GPE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
<td>*</td>
</tr>
<tr>
<td>pk</td>
<td>*</td>
</tr>
<tr>
<td>pk, fk</td>
<td>*</td>
</tr>
<tr>
<td>pk, fk</td>
<td>Plt_cty_code</td>
</tr>
<tr>
<td>*</td>
<td>Amount</td>
</tr>
</tbody>
</table>

 questões
Practice: Global Naming

Track a corporate name for each product.
Database Design Considerations
Overview

• Oracle specific Design Considerations
• Data Integrity Issues
• Performance Considerations
• Storage Issues
Why Adapt Data Design?

- User Expectations
  - Volumes
  - Hardware
  - Network
  - O.S.

Initial design

- Adapted Physical Design

- Oracle specifics
Oracle Data Types

Depending on:
- Domains
- Storage issue
- Performance
- Use

Select a data type for columns:
- Character
- Number
- Date
- Large Objects
Suggested Column Sequence

- Primary key columns
- Unique Key columns
- Foreign key columns
- Mandatory columns
- Optional columns

Large object columns *always* at the end
Primary Keys

CREATE TABLE countries
( code NUMBER(6) NOT NULL,
  name VARCHAR2(25) NOT NULL,
  currency NUMBER (10,2) NOT NULL
);
ALTER TABLE countries
ADD CONSTRAINT cty_pk PRIMARY KEY (code);

Constraint *and* Index name
Primary Keys

Choosing the Right Key

• Simplicity
• Ease of use
• Performance
• Size
• Meaningless
• Stability
Artificial Keys

- **AS (A)**
  - pk
  - * Id
  - C1

- **BS (B)**
  - pk
  - * Id
  - C2

- **CS (C)**
  - pk
  - * Id
  - C3

- **XS (X)**
  - pk
  - * Id
  - D_id
  - C5

- **DS (D)**
  - uk
  - fk
  - * A_id
  - B_id
  - C_id
  - C4
  - C5

  - pk
  - * Id

  - fk
  - * D_id

  - fk
  - * x_d_fk

- **fk₁ = d_a_fk**
- **fk₂ = d_b_fk**
- **fk₃ = d_c_fk**
- **fk = x_d_fk**
Sequences

CREATE SEQUENCE sequence_name
INCREMENT BY number
START WITH number
MINVALUE number
MAXVALUE number
CACHE number / NOCACHE
CYCLE | NOCYCLE;
## Foreign Key Behavior

<table>
<thead>
<tr>
<th></th>
<th>Delete</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restrict</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Cascade</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Default Nullify</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

✔️ Supported by Oracle through declaration
Indexes

- **Performance**

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBERT</td>
<td>2655</td>
</tr>
<tr>
<td>ALFRED</td>
<td>3544</td>
</tr>
<tr>
<td>ALICE</td>
<td>7593</td>
</tr>
<tr>
<td>ALLISON</td>
<td>3456</td>
</tr>
<tr>
<td>ALVIN</td>
<td>8642</td>
</tr>
<tr>
<td>ALPHONSO</td>
<td>2841</td>
</tr>
</tbody>
</table>

- **Uniqueness**
Choosing Indexes

- **B*tree**
  - aba .1.2.5
  - abb .1.3.5
  - abc .1.1.5
  - bba .1.4.5
  - ...

- **Bitmap**
  - X 0 1 0
  - Y 1 0 0
  - Z 0 1 0 1

- **Reverse**
  - aba .1.2.5
  - abb .1.4.5
  - bba .1.3.5
  - cba .1.1.5
  - ...

- **I.O.Table**
  - C1 | C2
  - aba | X
  - abb | Z
  - abc | Y
  - bba | Z
  - bbc | X

---

Copyright © Oracle Corporation, 2002. All rights reserved.
Which Columns to Index?

- Primary key columns and Unique Key columns (Up to Version 6)
- Foreign Key columns
- When significant better performance can be observed in SELECT statements

Avoid indexing:

- Small tables
- Columns frequently updated
When Can Indexes be Used?

- When referenced in a Where clause or Order By
- When the Where clause does not include some operators
- When the optimizer decides
- With hints in the SQL statement
Partitioning Tables and Indexes

<table>
<thead>
<tr>
<th>CUSTOMERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUSTOMERS_R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUSTOMERS_R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Views

- Restricting access
- Presentation of data
- Isolate applications from data structure
- Save complex queries
- Simplify user commands
Reasons for Views

• **Advantages**
  – Dynamic views
  – Present denormalized data from normalized tables
  – Simplify SQL statements

• **Disadvantages**
  – May affect performances
  – Restricted DML in some cases
Old Fashioned Design

- Unique index
- Views with “Check option” clause
- Generic Arc implementation
Generic Arc Implementation

Diagram showing relationships between entities A, X, Y, and an entity AS (A) with attributes "Table_name" and "Fk_id". The diagram illustrates the topology of the arcs and related entities.
Distributed Database

Different physical databases appear as one logical database.
Benefits of Distributed Databases

- Resilience
- Reduced line traffic
- Location transparency
- Local autonomy
- Easier growth path

but

- Increased, distributed, complexity
Database Structure

DATABASE

resides in

TABLESPACE

consists of

container of

residence of

DATA FILE

consists of

part of

DATA BLOCK

resides in

TABLE OR INDEX PARTITION

located in

part of

USED

FREE

SEGMENT

sliced in

part of

INDEX SEGMENT

consists of

part of

TABLE SEGMENT

sliced in

part of
Summary

- Data Types
- Primary, Foreign, and Artificial Keys
- Indexes
- Partitioning
- Views
- Distributed design
Practices

- Data Types
- Artificial Keys
- Product Pictures
## Data Types (1)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Suggested Data Type</th>
<th>Your Choice Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRIES</td>
<td>Code</td>
<td>Varchar2(2)</td>
<td></td>
</tr>
<tr>
<td>CURRENCIES</td>
<td>Code</td>
<td>Varchar2(3)</td>
<td></td>
</tr>
<tr>
<td>EXCHANGE_RATES</td>
<td>Month</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rate</td>
<td>Number(8,4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start_date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>End_date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>PRICE_LISTS</td>
<td>Name</td>
<td>Char(8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code</td>
<td>Char(10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>Number(4,2)</td>
<td></td>
</tr>
<tr>
<td>PRODUCTS</td>
<td>Pdt_type</td>
<td>Number(1)</td>
<td></td>
</tr>
<tr>
<td>PRODUCT_GROUPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODUCTS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Data Types (2)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Your Choice Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL_PRICES</td>
<td>Amount</td>
<td></td>
</tr>
<tr>
<td>LOCAL_PRICES</td>
<td>Start_date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>End_date</td>
<td></td>
</tr>
<tr>
<td>SHOPS</td>
<td>Amount</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>City</td>
<td></td>
</tr>
</tbody>
</table>
## Solution: Instance or Entity?

<table>
<thead>
<tr>
<th>Concept</th>
<th>E/A/I?</th>
<th>Example Instance or Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PRESIDENT</strong></td>
<td>E</td>
<td>Lincoln, Washington, Gorbachev</td>
</tr>
<tr>
<td><strong>ELLA FITZGERALD</strong></td>
<td>I</td>
<td>STAR, SINGER, PERSON</td>
</tr>
<tr>
<td><strong>DOG</strong></td>
<td>E</td>
<td>Snoopy</td>
</tr>
<tr>
<td><strong>ANIMAL</strong></td>
<td>E</td>
<td>Cat, Dog, ...</td>
</tr>
<tr>
<td><strong>HEIGHT</strong></td>
<td>A</td>
<td>PERSON, BUILDING, ...</td>
</tr>
<tr>
<td><strong>TYPE OF TRANSPORT</strong></td>
<td>E</td>
<td>CAR</td>
</tr>
<tr>
<td><strong>Number of Wheels</strong></td>
<td>A</td>
<td>CAR</td>
</tr>
<tr>
<td><strong>My current car</strong></td>
<td>I</td>
<td>CAR</td>
</tr>
</tbody>
</table>
Solution: Guest

GUEST

HOTEL

ROOM

Address
Arrival Date
Family Name
Room Number
Floor Number
Number of Beds
Number of Parking Lots
Price
TV set available?
Solution: Reading

A Each EMPLOYEE may be assigned to one or more DEPARTMENTS
Each DEPARTMENT must be responsible for one or more EMPLOYEES

B Each EMPLOYEE must be assigned to one or more DEPARTMENTS
Each DEPARTMENT may be responsible for one or more EMPLOYEES

C Each EMPLOYEE must be assigned to exactly one DEPARTMENT
Each DEPARTMENT may be responsible for exactly one EMPLOYEE
Practice: Read and Comment

PERSON

born in

TOWN

birthplace of

living in

home town of

visitor of

visited by

mayor of

governed by (as mayor)
Solution: Hotel

- HOTEL
  * Address

- ROOM
  * Room Number

- STAY
  * Arrival Date

- PERSON
  * Name

- Located in
- Taking place in
- Endured by
- Owned by
- Known by
- Favored by
- Aware of
- In favor of
- Location for
- Enduring
**Acorda alentejana**

**bread soup from Portugal**

**vegetarian**

**15 min**

**easy**

**for 4 persons:**
- 1 onion
- 4 cloves of garlic
- 1 red pepper
- 1 liter of vegetable broth
- 4 tablespoons of olive oil
- 4 fresh eggs
- 1 handful of parsley or coriander
- salt, pepper
- 9-12 slices of (old) bread

**preparation**

Cut the onion into small pieces and fry together with the garlic. Wash the red pepper, cut it in half, remove the seeds and fry it for at least 15
**Acorda alentejana**
bread soup from Portugal

<table>
<thead>
<tr>
<th>Vegetarian</th>
<th>15 min</th>
<th>Easy</th>
</tr>
</thead>
</table>

For 4 persons:
- 1 onion
- 4 cloves of garlic
- 1 red pepper
- 1 liter of vegetable broth
- 4 tablespoons of olive oil
- 4 fresh eggs
- 1 handful of parsley or coriander
- Salt, pepper
- 9-12 slices of (old) bread

**Preparation**

Cut the onion into small pieces and fry together with the garlic. Wash the red pepper, cut it in half, remove the seeds and fry it for at least 15
Solution: Recipe

A RECIPE must be classified in exactly one RECIPE GROUP.

A RECIPE GROUP may be classification for one or more RECIPES.

A RECIPE must be prepared with one or more INGREDIENTS.

An INGREDIENT may be used in exactly one RECIPE.
Solution: Books
Solution: Books

- MANUSCRIPT
- TRANSLATION
- TITLE
- EDITION
- PRINTING
- VERSION
- COPY
- LANGUAGE
Solution: Moonlight

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY</td>
<td>PRODUCT</td>
</tr>
<tr>
<td>COUNTRY COMMUNITY</td>
<td>PRODUCT GROUP</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>PRICE</td>
</tr>
<tr>
<td>DEPARTMENT</td>
<td>SALE</td>
</tr>
<tr>
<td>DRINK</td>
<td>SHOP</td>
</tr>
<tr>
<td>EMPLOYEE</td>
<td>STOCK OPTION</td>
</tr>
<tr>
<td>EXCHANGE RATE</td>
<td>STOCK PRICE</td>
</tr>
<tr>
<td>FOOD</td>
<td>TICKER SYMBOL</td>
</tr>
<tr>
<td>JOB</td>
<td></td>
</tr>
<tr>
<td>LOCATION TYPE</td>
<td></td>
</tr>
<tr>
<td>PASTRY</td>
<td></td>
</tr>
</tbody>
</table>
Solution: Shops

SHOP
- Number
- Name
- Location
- City
- Telephone
- Open Date

COUNTRY
- Code
- Name

LOCATION TYPE
- Description

LOCATION
- Name

or

SHOP
- Number
- Name
- Telephone
- Open Date

COUNTRY
- Code
- Name

LOCATION
- Name

LOCATION TYPE
- Description
Solution: Subtypes

DISABLED PERSON
- DEAF AND BLIND
- DEAF, NOT BLIND
- BLIND, NOT DEAF
- OTHER DISABLED

CAR
- STATION WAGON
- SEDAN
- OTHER

BUILDING
- HOUSE
- OTHER

HOTEL ROOM
- ROOM WITH BATH
- OTHER ROOM

DOG
- DOMESTIC ANIMAL
- MAMMAL

beyond repair
Solution: Schedule (1/2)
Solution: Schedule (2/2)

SCHEDULE ENTRY
* Monday Shift
* Tuesday Shift
* Wednesday Shift
* Thursday shift
* Friday Shift
* Saturday Shift
* Sunday Shift

for with

within with

with
Solution: Address

1. PERSON
   Address text line1
   Address text line2
   Address text line3
   Address text line4

2. PERSON
   Street or “PO Box” Indicator
   Street
   House Number
   City
   Post Code
   Province or State

   ADDRESS TYPE
   COUNTRY
   COUNTRY
Solution: Read the Relationship

Every ALU must be of exactly one BRY
Every BRY may be with one or more ALUS

Every PUR may be bazooned in one or more YOKS
Every YOK may be bazooned by one or more PURS

Every KLO must be bilought in one or more HARS
Every HAR may be glazoed with exactly one KLO
Solution: Find a Context

- COUNTRY
  - birthplace of
  - location of

- PERSON
  - born in

- TOWN
  - located in

- PERSON
  - mother of
  - son of

- FEMALE
  - mother of

- MALE
  - son of

- PERSON
  - most popular movie star in

- COUNTRY
  - birthplace of

- PERSON
  - born in

- COUNTRY
  - birthplace of

- PERSON
  - with most popular movie star

- COUNTRY
  - birthplace of
Solution: Name the Intersection Entity

PERSON

SAILBOAT

CREW MEMBERSHIP
  * Role

PRODUCT

DEPARTMENT
STORE

SALE
  * Date

INTERPRETER

LANGUAGE

FLUENCY
  * Score

in

of

at

with
### Solution: Receipt

**Served by:** Dennis

**Till:** 3 Dec 8, 4:35 pm

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Price</th>
<th>Subtotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPPUCC M</td>
<td>3.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREAM</td>
<td>.75</td>
<td>* 2</td>
<td>1.50</td>
</tr>
<tr>
<td>APPLE PIE</td>
<td>3.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLACKB MUF</td>
<td>4.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **<SUB>** 16.70
- **tax 12%** 2.00
- **<TOTAL>** 18.70

---

**CASH** 20.00

**RETURN** 1.30

---

Hope to serve you again.

@MOONLIGHT COFFEES

25 Phillis Rd, Atlanta
Solution: Moonlight P&O (1)

All Moonlight Coffee employees work for a department such as “Global Pricing” or “HQ”, or for a shop. All employees are at the payroll of one of our country organizations. Jill, for example, works as a shop manager in London; Werner is a financial administrator working for Accounting and is located in Germany …
... All shops belong to one country organization ("the countries"). There is only one country organization per country. All countries and departments report to HQ, except HQ itself ...
Solution: Moonlight P&O (3)

Employees can work part time. Lynn has had an 80% assignment for Product Development since the 1st September. Before that she had a full-time position.
Solution: Moonlight P&O (4)

4a: -

COUNTRY ORGANIZATION

4b

with

EMPLOYEE

with

for

PAYROLL ENTRY

* Start Date

for

4c

DEPARTMENT

* Name

with

HQ

reporting to

report of

REPORT OF

for

COUNTRY ORGANIZATION

in

SHOP

* Name

* No

* Name

4d

with

located in

belongs to

of

of

4e: -
Solution: Price List

- COUNTRY
  - Currency

- SHOP
  - Name
  - Address
  - City

- PRICE LIST
  - Start Date
  - End Date

- PRODUCT GROUP
  - Short Name

- PRODUCT
  - Size

- PRICE
  - Amount

- coffees, teas, foods, nonfoods, supplements
Solution: EMail

1. USER
   - owner of
   - owned by
   - referred to
   - for

2. NICK NAME
   - containing
   - within

3. USER
   - is
   - forwarded with
   - containing
   - within

4. FOLDER
   - containing
   - of
   - owned by
   - of
   - of
   - of
   - of

   LIST
   - containing
   - of
   - owned by
   - of
   - of
   - of
   - of
Solution: Holiday (1)

“Paul and I hiked in the USA. Eric and I hiked in France and we rented a car in the USA last year”.

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>TRANSPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Car</td>
</tr>
</tbody>
</table>

Paul and I hiked in the USA. Eric and I hiked in France and we rented a car in the USA last year.
Solution: Holiday (2)

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>COMPANION</th>
<th>TRANSPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Eric</td>
<td>Boots</td>
</tr>
<tr>
<td>France</td>
<td>Eric</td>
<td>Car</td>
</tr>
<tr>
<td>USA</td>
<td>Eric</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Eric</td>
<td>Car</td>
</tr>
<tr>
<td>USA</td>
<td>Paul</td>
<td>Boots</td>
</tr>
</tbody>
</table>
Solution: Holiday (3)

<table>
<thead>
<tr>
<th>COUNTRY</th>
<th>COMPANION</th>
<th>TRANSPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Eric</td>
<td>Boots</td>
</tr>
<tr>
<td>USA</td>
<td>Paul</td>
<td>Car</td>
</tr>
<tr>
<td>USA</td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>Paul</td>
<td></td>
</tr>
</tbody>
</table>
Practice: Normalize an ER Model: Solution
Solution: Identification 1

1. A # Xx
   B * Yy
   C # Zz

2. A
   C # Code
   B # Id

3. A * Xx
   B # Yy
   C # Zz

   with

   D # Id

   of
Solution: Identification 2

4 P
# Id

5 P
# Name

PERSON
MALE
# Seqno
son of
partner in
with husband

FEMALE
# Name
# Birth Date
mother of
partner in
with wife

MARRIAGE
# Start Date
Solution: Moonlight UID
Solution: Table

USER

FOREIGN KEY
#  Name

TABLE
#  Name

KEY
#  Name
  PRIMARY
  UNIQUE

COLUMN
#  Name
  *  Data Type
  o  Not Null

ASSOCIATION
#  Seqno

USAGE
#  Seqno

owner of

in

from

to

with

for

referenced in

in

of

of

for
Solution: Constraints

1. EMPLOYEE # Name managed by OTHER EMPLOYEE
2. USER # Name owned by ALIAS
3. FOLDER # Name with subfolder within USER # Name

Owner of:
- CEO
- OTHER EMPLOYEE
- ALIAS
- LIST
- NICKNAME

Manager of:
- CEO
- OTHER EMPLOYEE

with subfolder within USER # Name

owned by
- CEO
- USER # Name
- ALIAS
- LIST
- NICKNAME
Solution: Shift

- SHOP
  - Name

- CALENDAR DAY
  - Date
  - Public Holiday Indicator

- WEEKDAY
  - Code
  - Name

- SHIFT
  - No
  - Start Time
  - End Time

- SCHEDULE
  - Start Date
  - End Time

- of starting
- on
- with
- or
- on

SHOP with CALENDAR DAY of starting
WEEKDAY on SHOP SCHEDULE
SHIFT with SCHEDULE
Solution: Moonlight Pricing
DecafPunch (DP) = \{Coffee (C) or Tea (T)\} and \{Blackberry Muffin (BM)\}

Consider this as:
AS1 = (C or T)
DP = (AS1 and BM)

<table>
<thead>
<tr>
<th>Code</th>
<th>And_or</th>
<th>Pg_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td>..</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>AS1</td>
<td>OR</td>
<td>....</td>
</tr>
<tr>
<td>DP</td>
<td>AND</td>
<td>.....</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prod_code</th>
<th>Using_code</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS1</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>AS1</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>DP</td>
<td>AS1</td>
<td>1</td>
</tr>
<tr>
<td>DP</td>
<td>BM</td>
<td>1</td>
</tr>
</tbody>
</table>
A fixed number of levels

PRODUCT CLASS # Name

or generic:

with

within

LEVEL1

LEVEL2

LEVEL3

LEVEL4

LEVEL5

PRODUCT

2

product

+ coffees
+ teas
+ foods
+ nonfoods
+ supplements
+ bundles
## Patterns

<table>
<thead>
<tr>
<th>Model</th>
<th>Chain or Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model of moves in a chess game</td>
<td></td>
</tr>
<tr>
<td>Model of quotations</td>
<td></td>
</tr>
<tr>
<td>Model of recipes</td>
<td></td>
</tr>
<tr>
<td>Model of all people involved in college:</td>
<td></td>
</tr>
<tr>
<td>students, teachers, parents, ...</td>
<td></td>
</tr>
<tr>
<td>Rentals in a video shop</td>
<td></td>
</tr>
<tr>
<td>Model of phases in a process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M/D or Basket</td>
</tr>
<tr>
<td></td>
<td>Bill of Material</td>
</tr>
<tr>
<td></td>
<td>Roles</td>
</tr>
<tr>
<td></td>
<td>M/D or Basket</td>
</tr>
<tr>
<td></td>
<td>Chain or Network</td>
</tr>
</tbody>
</table>
## Patterns

| Model of moves in a chess game | Chain or Network |
| Model of quotations | M/D or Basket |
| Model of recipes | Bill of Material |
| Model of all people involved in college: students, teachers, parents, ... | Roles |
| Rentals in a video shop | M/D or Basket |
| Model of phases in a process | Chain or Network |
Moonlight Data Warehouse

YEAR # No
  with in
  with
  QUARTER
    with in
    assigned to
    with within
    MONTH # No
      with
      with
      WEEK # No
        with in
        with
        PRODUCT CLASS
          with of
          within
          PRODUCT # Id
            with
            of
            in
            of
            GEOGRAPHY
              with
              within
              * Number of Inhabitants
              * Number of Employees
            with
            of
            of
            SALES VOLUME
              * Quantity
              * Value in Local Currency
              * Value in $
Solution: Argos and Erats

Constraint not shown:
A ubin always consists of one or more argos of the erat, one or more...
Solution: Synonym

MEANINGS

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Action, actual doing ...</td>
</tr>
<tr>
<td>2</td>
<td>Regular arrangement ...</td>
</tr>
<tr>
<td>3</td>
<td>Order, command ...</td>
</tr>
<tr>
<td>4</td>
<td>A vehicle with two wheels</td>
</tr>
</tbody>
</table>

WORDS

<table>
<thead>
<tr>
<th>Word</th>
<th>Mng_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practice</td>
<td>d</td>
</tr>
<tr>
<td>Exercise</td>
<td>1</td>
</tr>
<tr>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Sequence</td>
<td>2</td>
</tr>
<tr>
<td>Arrangement</td>
<td>2</td>
</tr>
<tr>
<td>Order</td>
<td>2</td>
</tr>
<tr>
<td>Command</td>
<td>3</td>
</tr>
<tr>
<td>Demand</td>
<td>3</td>
</tr>
<tr>
<td>Bike</td>
<td>3</td>
</tr>
</tbody>
</table>

MEANING

<table>
<thead>
<tr>
<th># Id</th>
<th>* Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

shared by

with

synonym

homonym
Solution: Mapping basic Entities, Attributes and Relationships

**DEPARTMENTS (DPT)**
- **pk**: *  
- **Id**:  
- **Name**:  
- **Location**: 

**EMPLOYEES (EPE)**
- **pk**: *  
- **Id**:  
- **Dpt_id**:  
- **Last Name**:  
- **First Name**:  
- **Home Phone**:  

Relationship: `epe_dpt_fk`
Solution: Mapping Supertype

<table>
<thead>
<tr>
<th>DEPARTMENTS (DPT)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pk</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Diagram: Dpt_dpt_fk
Solution: Mapping Supertype

```
DEPARTMENTS (DPT)

| pk | * | Id  |
|------------------------------|
|    |   | Name|
|------------------------------|
|    |   | Dpt_type|
|------------------------------|
|    |   | Headcount|
|------------------------------|
|    |   | Address|
|------------------------------|
|    |   | Dpt_id_reporting_to|
```

`dpt_dpt_fk`
Solution: Quality Check
Subtype Implementation

GLOBAL_PRODUCTS (GPT)

| pk | *  | Code  
|    | o  | Size  
| fk | *  | Pgp_name |

LOCAL_PRODUCTS (LPT)

| pk | *  | Name  
| pk,fk | * | Shp_no  
| fk  | *  | Pgp_name |
Solution: Quality Check both Supertype and Subtype Implementation

<table>
<thead>
<tr>
<th>PRODUCTS (PDT)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>* Code</td>
</tr>
<tr>
<td>fk₁</td>
<td>* Pgp_name</td>
</tr>
<tr>
<td>fk₂, uk₁</td>
<td>o Gpt_code</td>
</tr>
<tr>
<td>fk₃, uk₂</td>
<td>o Lpt_name</td>
</tr>
<tr>
<td>fk₃, uk₂</td>
<td>o Lpt_shp_no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GLOBAL_PRODUCTS (GPT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
</tr>
<tr>
<td>* Code</td>
</tr>
<tr>
<td>o Size</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCAL_PRODUCTS (LPT)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>* Name</td>
</tr>
<tr>
<td>pk, fk</td>
<td>* Shp_no</td>
</tr>
</tbody>
</table>

fk₁=pdt_pgp_fk
fk₂=pdt_gpt_fk
fk₃=pdt_lpt_fk
lpt_shp_fk
Solution: Primary Keys and Columns

GLOBAL_PRICES (GPE)

<table>
<thead>
<tr>
<th>pk, fk₁</th>
<th>*</th>
<th>Plt_cty_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk₁</td>
<td>*</td>
<td>Plt_start_date</td>
</tr>
<tr>
<td>pk, fk₂</td>
<td>*</td>
<td>Gpt_code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Amount</td>
</tr>
</tbody>
</table>

fk₁ = gpe_plt_fk

fk₂ = gpe_gpt_fk
Practice: Triggers (1/6)
Solution: Triggers (2/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHR</td>
<td>Insert</td>
<td></td>
<td>Y</td>
<td>Order_total := 0</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Id</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Order_total</td>
<td>Y</td>
<td>prevent update</td>
</tr>
<tr>
<td>OIM</td>
<td>Insert</td>
<td></td>
<td>Y</td>
<td>recalculate Order_total</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>Y</td>
<td>recalculate Order_total</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Ohr_id</td>
<td>Y</td>
<td>recalculate Order_total</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Item_total</td>
<td>Y</td>
<td>recalculate Order_total</td>
</tr>
</tbody>
</table>
Practice: Triggers (3/6)

LOCATIONS (LCN)

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>Address</td>
</tr>
</tbody>
</table>

EMPLOYEES (EPE)

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>fk</td>
<td>*</td>
<td>Lcn_id</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Lcn_address</td>
</tr>
</tbody>
</table>
## Solution: Triggers (4/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCN</td>
<td>Insert</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Address</td>
<td>Y</td>
<td>Cascade to Employees</td>
</tr>
<tr>
<td></td>
<td></td>
<td>other cols</td>
<td>Y</td>
<td>If pk updated than extended cascade</td>
</tr>
<tr>
<td>EPE</td>
<td>Insert</td>
<td></td>
<td>Y</td>
<td>Set correct Lcn_address</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Lcn_id</td>
<td>Y</td>
<td>Set correct Lcn_address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lcn_address</td>
<td>Y</td>
<td>Prevent update</td>
</tr>
</tbody>
</table>
Practice: Triggers (5/6)

PRODUCTS (PDT)
- pk
- * Id
- * Name

PRICES (PCE)
- pk
- * Pdt_id
- * Start_date
- o End_date
- * Curr_price_ind
### Solution: Triggers (6/6)

<table>
<thead>
<tr>
<th>Table</th>
<th>Trg Type</th>
<th>Column</th>
<th>Needed?</th>
<th>What should it do?</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDT</td>
<td>Insert</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PCE</td>
<td>Insert</td>
<td></td>
<td>Y</td>
<td>Prevent overlap in price periods</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Pdt_id</td>
<td>Y</td>
<td>Set Curr_price_ind to NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start_date</td>
<td>Y</td>
<td>Re-evaluate Curr_price_ind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>End_date</td>
<td>Y</td>
<td>Re-evaluate Curr_price_ind</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Curr_price_Ind</td>
<td>Y</td>
<td>Prevent update by user</td>
</tr>
</tbody>
</table>
Solution: Denormalize Price Lists

**GLOBAL_PRICES (GPE)**

<table>
<thead>
<tr>
<th>pk, fk</th>
<th>*</th>
<th>Plt_start_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
<td>*</td>
<td>Plt_cty_code</td>
</tr>
<tr>
<td>pk, fk</td>
<td>*</td>
<td>Amount</td>
</tr>
<tr>
<td>pk, fk</td>
<td>*</td>
<td>Current_indicator</td>
</tr>
</tbody>
</table>

**PRICE_LISTS (PLT)**

<table>
<thead>
<tr>
<th>pk</th>
<th>*</th>
<th>Start_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
<td>*</td>
<td>Cty_code</td>
</tr>
<tr>
<td>pk, fk</td>
<td>*</td>
<td>End_date</td>
</tr>
</tbody>
</table>

**GLOBAL_PRICES** and **PRICE_LISTS** are denormalized to avoid querying the same price information multiple times.
Solution: Denormalize Global Naming

**PRODUCTS (PDT)**
- pk (primary key)
- Code
- Size
- Corporate_Name

**PRODUCT_NAME (PNE)**
- pk (primary key)
- Pdt_code
- Lge_code
- Name

**LANGUAGES (LGE)**
- pk (primary key)
- Code
- Name
### Solution: Data Types (1)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Suggested Data Type</th>
<th>Your Choice Data Type?</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRIES</td>
<td>Code</td>
<td>Varchar2(2)</td>
<td>Char(2)</td>
</tr>
<tr>
<td>CURRENCIES</td>
<td>Code</td>
<td>Varchar2(3)</td>
<td>Char(3)</td>
</tr>
<tr>
<td>EXCHANGE_RATES</td>
<td>Month</td>
<td>Date</td>
<td>Numberlicate(2,2)</td>
</tr>
<tr>
<td>PRICE_LISTS</td>
<td>Start_date</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>PRODUCT_GROUPS</td>
<td>End_date</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>PRODUCTS</td>
<td>Name</td>
<td>Char(8)</td>
<td>Char(8)</td>
</tr>
<tr>
<td></td>
<td>Code</td>
<td>Char(10)</td>
<td>Char(10)</td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>Number(4,2)</td>
<td>Number(4,2)</td>
</tr>
<tr>
<td></td>
<td>Pdt_type</td>
<td>Number(1)</td>
<td>Number(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Solution: Data Types (2)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Your Choice Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL_PRICES</td>
<td>Amount</td>
<td>Number(15,3)</td>
</tr>
<tr>
<td></td>
<td>Start_date</td>
<td>Date</td>
</tr>
<tr>
<td></td>
<td>End_date</td>
<td>Date</td>
</tr>
<tr>
<td>LOCAL_PRICES</td>
<td>Amount</td>
<td>Number(15,3)</td>
</tr>
<tr>
<td>SHOPS</td>
<td>Name</td>
<td>Varchar2(50)</td>
</tr>
<tr>
<td></td>
<td>Address</td>
<td>Varchar2(50)</td>
</tr>
<tr>
<td></td>
<td>City</td>
<td>Varchar2(50)</td>
</tr>
</tbody>
</table>
Solution: Product Pictures

```
<table>
<thead>
<tr>
<th>PRODUCTS (PDT)</th>
<th>TEXT_DOCUMENTS (TDT)</th>
<th>BINARY_DOCUMENTS (BDT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk, fk</td>
<td>Pdt_code</td>
<td>pk, fk</td>
</tr>
<tr>
<td>pk</td>
<td>,O</td>
<td>pk, fk</td>
</tr>
<tr>
<td>Code</td>
<td>Info_type Information</td>
<td>Pdt_code</td>
</tr>
<tr>
<td></td>
<td>Information</td>
<td>Information</td>
</tr>
<tr>
<td></td>
<td>varchar2(3)</td>
<td>varchar2(3)</td>
</tr>
<tr>
<td></td>
<td>varchar2(20)</td>
<td>varchar2(20)</td>
</tr>
<tr>
<td></td>
<td>CLOB</td>
<td>BLOB</td>
</tr>
</tbody>
</table>
```


Normalization
Overview

• Table Normalization
• Normal Forms of Tables
Why Normalize?

- An Entity Model is not always available as a starting point for design.
- To reduce redundant data in existing design.
- To increase integrity of data, and stability of design.
- To identify missing tables, columns and constraints.

Note: Third normal form is the generally-accepted goal for a database design that eliminates redundancy.
## Recognize Unnormalized Data

<table>
<thead>
<tr>
<th>USER</th>
<th>USER</th>
<th>MSE</th>
<th>REC_</th>
<th>_ID</th>
<th>NAME</th>
<th>_ID</th>
<th>DATE</th>
<th>SUBJECT</th>
<th>TEXT</th>
<th>SRVR</th>
<th>SERVER</th>
<th>_ID</th>
<th>_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>Smith</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td>3786</td>
<td>IMAP05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>Smith</td>
<td>54098</td>
<td>07/12</td>
<td>Promotions</td>
<td>I like to.</td>
<td>3786</td>
<td>IMAP05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>Smith</td>
<td>54445</td>
<td>10/06</td>
<td>Next Assignment</td>
<td>Your next.</td>
<td>3786</td>
<td>IMAP05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td>6001</td>
<td>IMAP08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
<td>54512</td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can you...</td>
<td>6001</td>
<td>IMAP08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
<td>54660</td>
<td>12/01</td>
<td>Jogging Today?</td>
<td>Can you...</td>
<td>6001</td>
<td>IMAP08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td>9988</td>
<td>EMEA01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
<td>54554</td>
<td>03/17</td>
<td>Stock Quote</td>
<td>The latest</td>
<td>9988</td>
<td>EMEA01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td>2201</td>
<td>EMEA09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
<td>54512</td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can we ...</td>
<td>2201</td>
<td>EMEA09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Normalization Rules

<table>
<thead>
<tr>
<th>Normal Form Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Normal Form (1NF)</td>
<td>The table must express a set of unordered, two-dimensional tables. The table cannot contain repeating groups.</td>
</tr>
<tr>
<td>Second Normal Form (2NF)</td>
<td>The table must be in 1NF. Every non-key column must be dependent on all parts of the primary key.</td>
</tr>
<tr>
<td>Third Normal Form (3NF)</td>
<td>The table must be in 2NF. No non-key column may be functionally dependent on another non-key column. “Each non-primary key value MUST be dependent on the key, the whole key, and nothing but the key.”</td>
</tr>
</tbody>
</table>
Converting to First Normal Form

1. Remove repeating group from the base table.
2. Create a new table with the PK of the base table and the repeating group.
## First Normal Form - Single Record

### USERS

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>USER_NAME</th>
<th>MSE_REC_ID</th>
<th>DATE</th>
<th>SUBJECT</th>
<th>TEXT</th>
<th>SRVR_ID</th>
<th>SERVER_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>Smith</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is...</td>
<td>3786</td>
<td>IMAP05</td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
<td>54512</td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can you...</td>
<td>6001</td>
<td>IMAP08</td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is...</td>
<td>9988</td>
<td>EMEA01</td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is...</td>
<td>9988</td>
<td>EMEA01</td>
</tr>
</tbody>
</table>

### SRVR

<table>
<thead>
<tr>
<th>SRVR_ID</th>
<th>SERVER_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>3786</td>
<td>IMAP05</td>
</tr>
<tr>
<td>6001</td>
<td>IMAP08</td>
</tr>
<tr>
<td>9988</td>
<td>EMEA01</td>
</tr>
</tbody>
</table>
First Normal Form - Repeating Groups

**RECEIVED_MESSAGES (1NF)**

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>MSE_ID</th>
<th>REC_ID</th>
<th>DATE</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>54098</td>
<td>07/12</td>
<td>Promotions</td>
<td>I like to.</td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>54445</td>
<td>10/06</td>
<td>Next Assignment</td>
<td>Your next.</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54512</td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can you...</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54660</td>
<td>12/01</td>
<td>Jogging Today?</td>
<td>Can you...</td>
<td></td>
</tr>
<tr>
<td>7773</td>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td></td>
</tr>
<tr>
<td>54512</td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can we ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54554</td>
<td>03/17</td>
<td>Stock Quote</td>
<td>The latest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54101</td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54101</td>
<td>05/07</td>
<td>Lunch?</td>
<td>Can we ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**USERS (1NF)**

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>NAME</th>
<th>SRVR_ID</th>
<th>SERVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>Smith</td>
<td>3786</td>
<td>IMAP05</td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
<td>6001</td>
<td>IMAP08</td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
<td>9988</td>
<td>EMEA01</td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
<td>9988</td>
<td>EMEA01</td>
</tr>
</tbody>
</table>
Converting to Second Normal Form

1. Determine which non-key columns are not dependent upon the table’s entire primary key.
2. Remove those columns from the base table.
3. Create a second table with those columns and the columns from the \( PK \) that they are dependent upon.
Tables Already in Second Normal Form

Is the USERS table already in 2NF?
Convert to Second Normal Form

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>MSE_ID</th>
<th>REC_ID</th>
<th>DATE</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>54101</td>
<td></td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>2301</td>
<td>54098</td>
<td></td>
<td>07/12</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>2301</td>
<td>54445</td>
<td></td>
<td>10/06</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>5607</td>
<td>54101</td>
<td></td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>5607</td>
<td>54512</td>
<td></td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>5607</td>
<td>54660</td>
<td></td>
<td>12/01</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54098</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54445</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54660</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54554</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MESSAGES** (2NF)

<table>
<thead>
<tr>
<th>MSE_ID</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>54101</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>54098</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>54445</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>54512</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54660</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54554</td>
<td>Stock Quote</td>
<td>The latest</td>
</tr>
</tbody>
</table>

**USER_ID** ---- 2301

**MESSAGE** (2NF)

<table>
<thead>
<tr>
<th>MSE_ID</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>54101</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>54098</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>54445</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>54512</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54660</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54554</td>
<td>Stock Quote</td>
<td>The latest</td>
</tr>
</tbody>
</table>

**RECEIVED_MESSAGES** (1NF)

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>MSE_ID</th>
<th>REC_ID</th>
<th>DATE</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>54101</td>
<td></td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>2301</td>
<td>54098</td>
<td></td>
<td>07/12</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>2301</td>
<td>54445</td>
<td></td>
<td>10/06</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>5607</td>
<td>54101</td>
<td></td>
<td>05/07</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>5607</td>
<td>54512</td>
<td></td>
<td>06/07</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>5607</td>
<td>54660</td>
<td></td>
<td>12/01</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54098</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54445</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54660</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54554</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MESSAGES** (2NF)

<table>
<thead>
<tr>
<th>MSE_ID</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>54101</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>54098</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>54445</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>54512</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54660</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54554</td>
<td>Stock Quote</td>
<td>The latest</td>
</tr>
</tbody>
</table>
Converting to Third Normal Form

Remove any columns that are dependent upon another non-key column:

1. Determine which columns are dependent upon another non-key column.
2. Remove those columns from the base table.
3. Create a second table with those columns and the non-key columns that they are dependent upon.
Tables Already in Third Normal Form

No non-key column can be functionally dependent upon another non-key column.

RECEIVED_MESSAGES (2NF)

MESSAGES (2NF)

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>MSE_ID</th>
<th>REC_ID</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>54101</td>
<td>05/07</td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>54098</td>
<td>07/12</td>
<td></td>
</tr>
<tr>
<td>2301</td>
<td>54445</td>
<td>10/06</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54101</td>
<td>05/07</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54512</td>
<td>06/07</td>
<td></td>
</tr>
<tr>
<td>5607</td>
<td>54660</td>
<td>12/01</td>
<td></td>
</tr>
<tr>
<td>7773</td>
<td>54101</td>
<td>05/07</td>
<td></td>
</tr>
<tr>
<td>7773</td>
<td>54554</td>
<td>03/17</td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>54101</td>
<td>05/07</td>
<td></td>
</tr>
<tr>
<td>0022</td>
<td>54512</td>
<td>06/07</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>SUBJECT</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>54101</td>
<td>Meeting Today</td>
<td>There is.</td>
</tr>
<tr>
<td>54098</td>
<td>Promotions</td>
<td>I like to</td>
</tr>
<tr>
<td>54445</td>
<td>Next Assignment</td>
<td>Your next</td>
</tr>
<tr>
<td>54512</td>
<td>Lunch?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54660</td>
<td>Jogging Today?</td>
<td>Can you..</td>
</tr>
<tr>
<td>54554</td>
<td>Stock Quote</td>
<td>The latest</td>
</tr>
</tbody>
</table>

Are these two tables in third normal form? Why?
Converting to Third Normal Form

**USERS**

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>USER_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>Smith</td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
</tr>
</tbody>
</table>

**MAIL_SERVER**

<table>
<thead>
<tr>
<th>MAIL_SRVR_ID</th>
<th>SRVR_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>3786</td>
<td>IMAP05</td>
</tr>
<tr>
<td>6001</td>
<td>IMAP08</td>
</tr>
<tr>
<td>9988</td>
<td>EMEA01</td>
</tr>
<tr>
<td>9988</td>
<td>EMEA01</td>
</tr>
</tbody>
</table>

**SRVR**

<table>
<thead>
<tr>
<th>SRVR_ID</th>
<th>SRVR_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2301</td>
<td>Smith</td>
</tr>
<tr>
<td>5607</td>
<td>Jones</td>
</tr>
<tr>
<td>7773</td>
<td>Walsh</td>
</tr>
<tr>
<td>0022</td>
<td>Patel</td>
</tr>
</tbody>
</table>
Summary

1NF  The table must express a set of unordered, two-dimensional tables. The table cannot contain repeating groups.

2NF  The table must be in 1NF. Every non-key column must be dependent on all parts of the primary key.

3NF  The table must be in 2NF. No non-key column may be functionally dependent on another non-key column.

An entity relationship model transforms into normalized data design.