UTS:ENGINEERING AND INFORMATION TECHNOLOGY

UTS UNIVERSITY OF TECHNOLOGY, SYDNEY

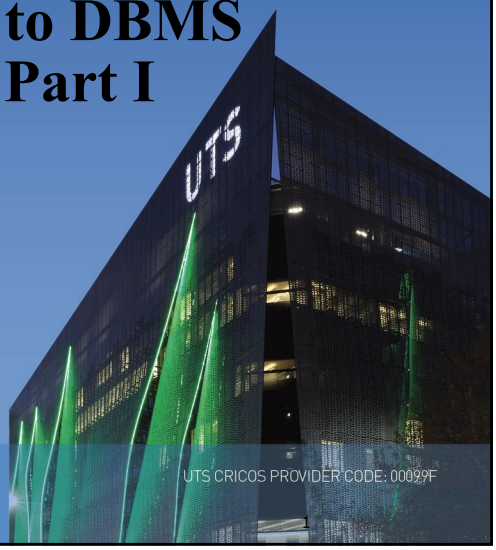
lecture 1: Introduction to DBMS and Data Modeling Part I

Modern Database Management

11th Edition, International Edition

Chapter 1 & 2: Modeling Data in the Organization

Jeffrey A. Hoffer, V. Ramesh, Heikki Topi



Innovation in practice

eng.uts.edu.au • it.uts.edu.au

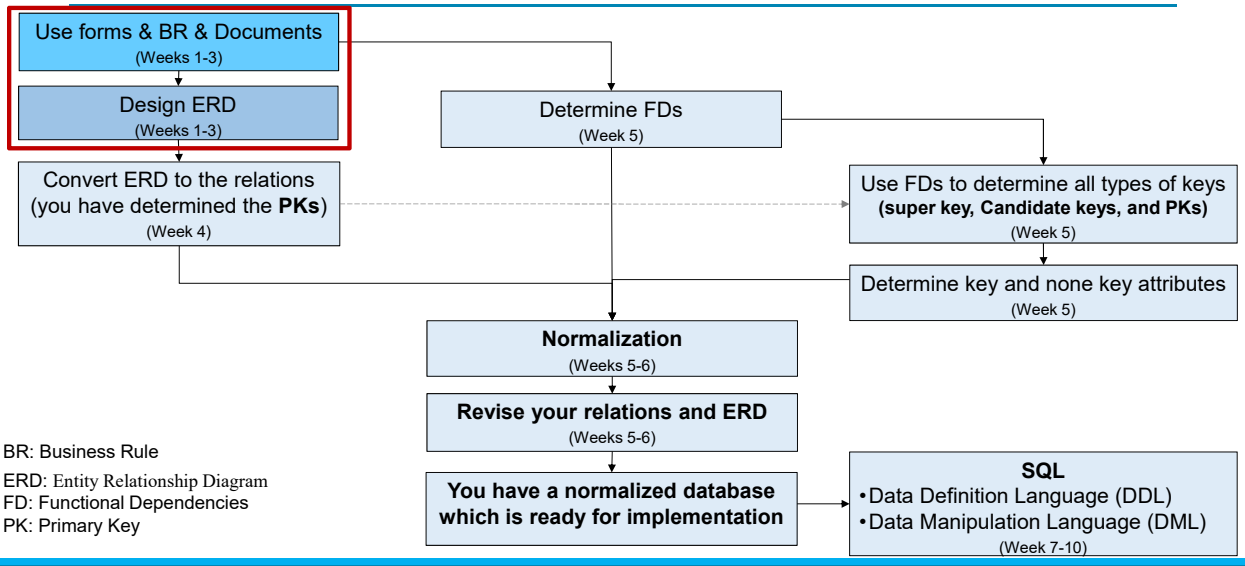
Participations and Discussions

If you have any question and you don't want to share it now, send it to us via UTSOnline/Discussion Board.

However, it is better to speak out ☺

Introduction to the subject

Subject Flowchart

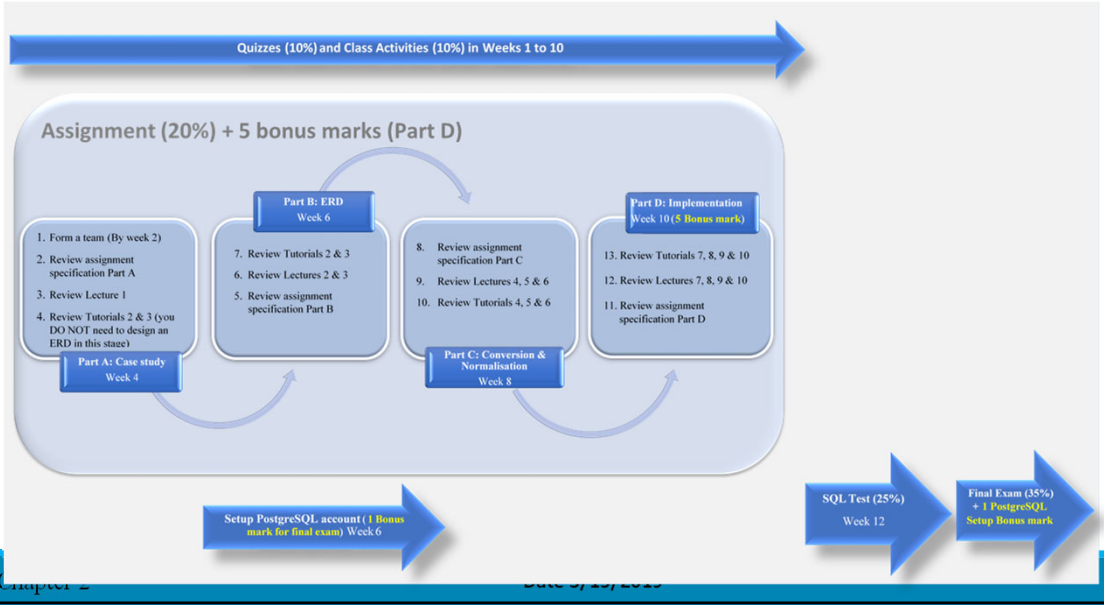


BR: Business Rule
ERD: Entity Relationship Diagram
FD: Functional Dependencies
PK: Primary Key

Subject Overview

- Design Entity Relationship Diagram (ERD)
  - Week 1: Data Modelling I (Conceptual Level)
  - Week 2: Data Modelling II (Conceptual Level)
  - Week 3: Data Modelling III (Conceptual Level)
  - Week 4: Convert ERD to Relations (Logical Level)
  - Week 5: Functional Dependencies
  - Week 5: Normalization I
  - Week 6: Normalization II
- Data manipulation
  - Week 7: Simple Query
  - Week 8: Multiple Table Queries
  - Week 9: Subquery
  - Week 10: Correlated Subquery

Assessment Chart and Knowledge Guideline



Subject Outline

UTSOnline (Learning plan, etc.)

Email structure to the subject coordinator:

The subject coordinator is more than happy to answer the emails from students that have the following requirements:

- The email is related to a **personal** issue, OR
- The information is not provided in the **subject outline** or the **assignment specification,** or is not posted in the **announcements,** OR
- There is no related forum on **discussion board.**

Your email needs to have the following information in its **title**:

- **Subject Number (31271)**
- **Subject of the email (e.g. Request for Extension)**
- **First Name & Last Name**
- **Student ID**
- **Your tutorial name (e.g. Tut1-05)**

Considering the number of students in this subject (400), I need to say that we may not be able to answer emails that do not meet these requirements.

The response to the emails that do not meet the aforementioned requirements will be "**see the email structure in Page 9 of Lecture 1**" ☺

Many thanks and kind regards,
Fahimeh

DF Learning Plan

Workshop Structure:

- The workshop starts with a collaborative lecture. You will complete some tasks during the lecture as part of your class activities. Then you will have a short break. The tutorial starts after the break. During the tutorial you will complete the rest of your class activities.
- The class activities need to be checked by tutors then submitted at the end of the workshop. Please see the 'further information' of task 2 in the subject outline for more information. You will complete the tasks in groups.
- You also need to do an open book quiz of what you have learned in the workshop. The quiz will be run after the lecture, or after the break, or at the end of the class.

Please be aware that the lecture slides with Blue title are designed for your self study.

Workshop 1 Timetable:

Activity	Duration	Comments
Lecture	2 hours	You will have 4 tasks to complete (20 minutes)
Rest	10 minutes	Have fun :)
Review	10 minutes	Please review the review questions and ask your questions if you have any
Tutorial	25 minutes	Have even more fun :D
Quiz (Open book)	5 minutes	Do your best ;)
Leave the class	10 minutes	Don't forget to review what you have learn in this class, and check the information that is provided on **UTSOnline/Learning Material/Week 1**

Lecture Objectives:

Section 1: Why Databases & Introduction to DBMS (For your information)

Section 2: Database Development (Basic Concepts)

Section 3: Database Development (Conceptual Level: ERD)

Section 1: Why Databases & Introduction to DBMS

Note: This section is designed to provide you with extra information and give you a perspective to the subject. Information in this section will not be examined.

Traditional File Processing system Creates Duplicate Data

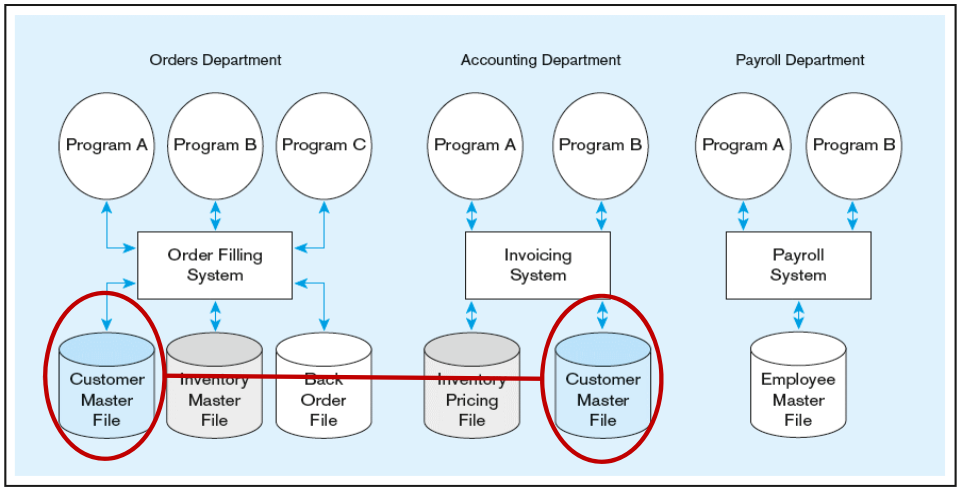
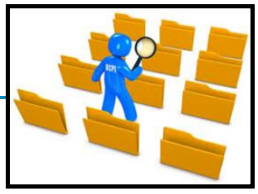


FIGURE 1-2 Old file processing systems at Pine Valley Furniture Company

Disadvantages of File Processing



- **Program-Data Dependence**
  - All programs maintain metadata for each file they use
- **Duplication of Data**
  - Different systems/programs have separate copies of the same data
- **Limited Data Sharing**
  - No centralized control of data
- **Lengthy Development Times**
  - Programmers must design their own file formats
- **Excessive Program Maintenance**
  - 80% of information systems budget

Problems with Data Dependency

- Each application programmer must maintain his/her own data
- Each application program needs to include code for the metadata of each file
- Each application program must have its own processing routines for reading, inserting, updating, and deleting data
- Lack of coordination and central control
- Non-standard file formats

Problems with Data Redundancy (Duplication of Data)

- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest problem:
  - **Data changes in one file could cause inconsistencies**
  - Compromises in ***data integrity***

Problem with Spreadsheets: Redundancy

- In a spreadsheet, each row is intended to stand on its own.
  As a result, the same information may be entered several times

- E.g. The BoyGirl
  Compare the BoyGirl spreadsheet to BoyGirl Relational database …

BoyGirl Database: Spreadsheet

NOTE: Not a good design!

BoyGirlNo	BoyName	BoyMobile	GirlName	GirlMobile
1	Adam	0414 1236	Alice	0414 1234
2	Bob	0414 1237	Bonnie	0414 1235
3	Charlie	0414 1238	Bonnie	0414 1235
4	Dennis	<null>	<null>	<null>

- **One** girl can contact **many** boys, so …
- .. we store **redundant** data about Bonnie.

A BoyGirl Relational Database

BoyNo	BoyName	BoyMobile	GirlNo
1	Adam	0414 1236	1
2	Bob	0414 1237	2
3	Charlie	0414 1238	2
4	Dennis	<null>	<null>

GirlNo	GirlName	GirlMobile
1	Alice	0414 1234
2	Bonnie	0414 1235

Primary keys

Foreign keys

Primary keys

- No redundant data about Bonnie
- Two tables with a **One-to-Many** relationship
- … linked by a **Foreign Key**

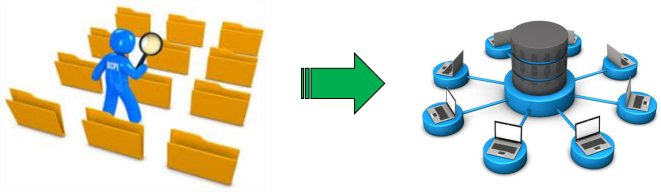
The Problem of Storing Redundant Data

- **Delete:** some but not all instances of data
- **Update:** some but not all instances of data
- **Insert:** multiple data entry can introduce inconsistency

- Also – very important! – multiple data entry is expensive

SOLUTION: The DATABASE Approach

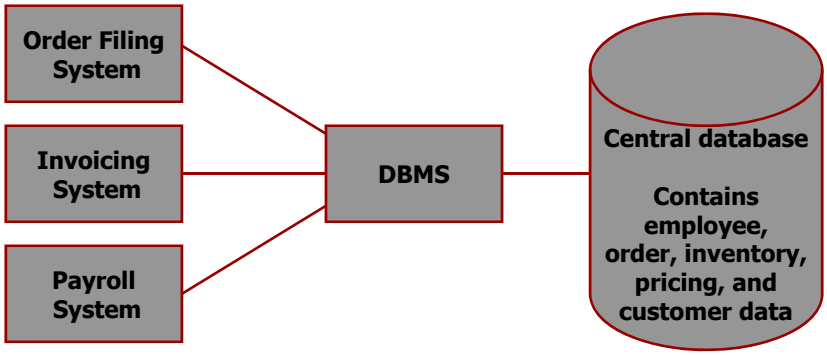
- Central repository of shared data
- Data is managed by a controlling agent
- Stored in a standardized, convenient form



Requires a Database Management System (DBMS)

Database Management System

- **A software system that is used to define, create, maintain a database, and provide controlled access to user databases**



DBMS manages data resources like an operating system manages hardware resources

Advantages of the Database Approach

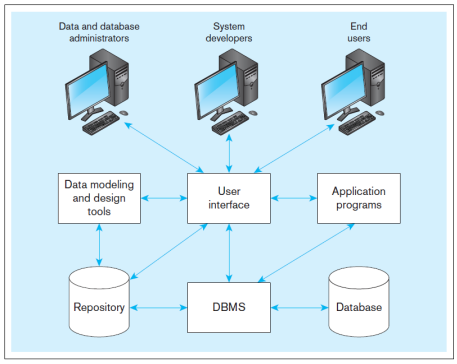
- Program-data independence
- Planned data redundancy
- Improved data consistency
- Improved data sharing
- Increased application development productivity
- Enforcement of standards
- Improved data quality
- Improved data accessibility and responsiveness
- Reduced program maintenance
- Improved decision support

Costs and Risks of the Database Approach

- New, specialized personnel
- Installation and management cost and complexity
- Conversion costs
- Need for explicit backup and recovery
- Organizational conflict

Components of the Database Environment (figure 1-5)

- **Data modeling and design tools** –automated tools used to design databases and application programs
- **Repository**–centralized storehouse of metadata
- **Database Management System (DBMS)** –software for managing the database
- **Database**–storehouse of the data
- **Application Programs**–software using the data
- **User Interface**–text, graphical displays, menus, etc. for user
- **Data/Database Administrators**–personnel responsible for maintaining the database
- **System Developers**–personnel responsible for designing databases and software
- **End Users**–people who use the applications and databases



Database Schema based on "American National Standards Institute-Standards Planning And Requirements Committee (ANSI-SPARC)"

- **External Schema**
  - User Views
  - Subsets of Conceptual Schema

  Different people have different views of the database these are the external schema
- **Conceptual Schema**
  - E-R models (Lectures 1, 2 and 3)
- **Internal Schema**
  - Logical structures (Lecture 4)
  - Physical structures

  The internal schema is the underlying design and implementation

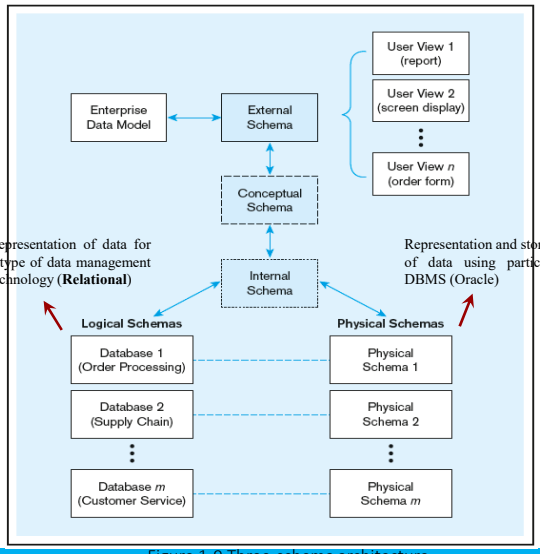


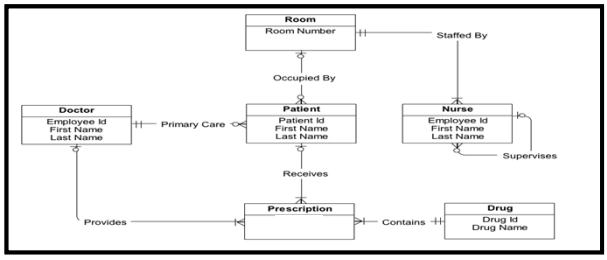
Figure 1-9 Three-schema architecture

ANSI-SPARC Architecture

External Level (User Views): A user's view of the database describes a part of the database that is relevant to a particular user. It excludes irrelevant data as well as data which the user is not authorized to access.

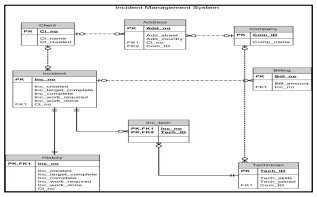
ANSI-SPARC Architecture

- **Conceptual Level:** The conceptual level is a way of describing what data is stored within the whole database and how the data is inter-related. The conceptual level does not specify how the data is physically stored.
- It is a detailed, technology-independent specification of the overall structure of organizational data (could be represented by an entity-relationship diagram (ERD)).



ANSI-SPARC Architecture

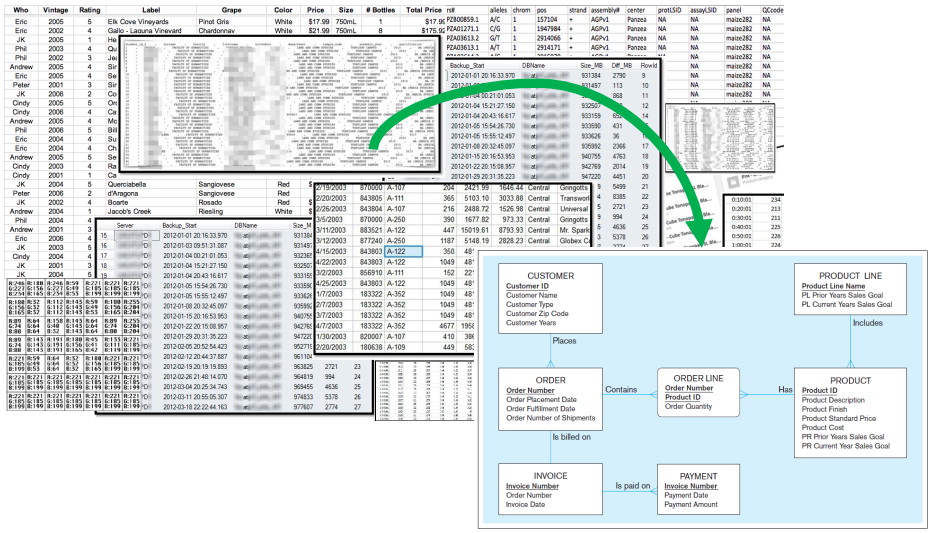
- **Internal Level:** The internal level involves how the database is physically represented on the computer system. It describes how the data is actually stored in the database and on the computer hardware.
  - **Logical data model (or schema):** Data model specific to a particular database approach, for example the relational data model. In the case of a relational data model, elements of the logical model include tables, columns, rows, primary and foreign keys, as well as constraints.



  - **Physical data model (or schema):** A set of specifications that detail how data from a logical data model are stored in a computer's secondary memory for a specific database management system.

Section 2: Database Development (Basic Concepts)

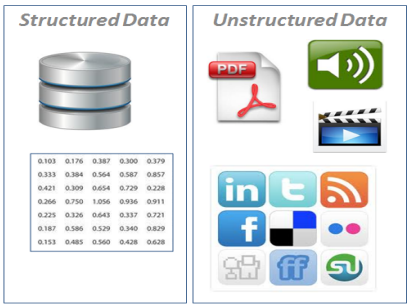
Database: Organized collection of logically related data



Data

Data: Stored representations of objects and events that have meaning and importance in the user's environment.

- Structured: numbers, text, dates
- Unstructured: images, video, documents



Metadata

Metadata: Data that describes the properties or characteristics of end-user data and the context of that data.

TABLE 1-1 Example Metadata for Class Roster

Data Item	Metadata					
Name	**Type**	**Length**	**Min**	**Max**	**Description**	**Source**
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and data context

Information

Information: Data that have been processed in such a way as to increase the knowledge of the person who uses it.

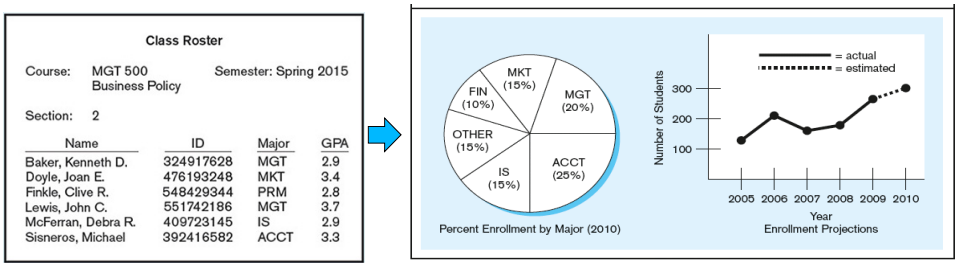


Figure 1-1a: Data

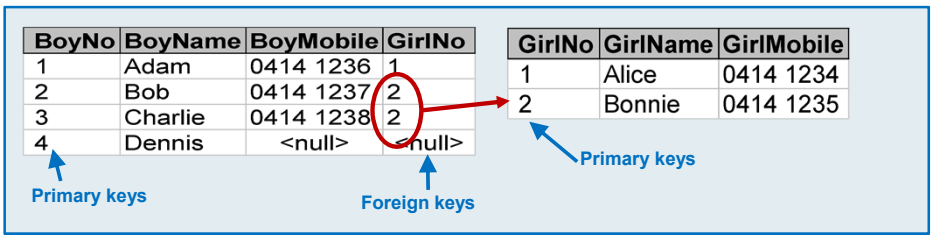
Figure 1-1b: Summarized data (Information)

Graphical displays turn data into useful information that managers can use for decision making and interpretation

Section 3: Database Development
(Conceptual Level: ERD)

Relational Database

Relational Database: a database that represents data as a collection of tables in which all data relationships are represented by common value in related tables



Enterprise Data Model

First steps in the database development process

3.1. Specifies scope and general content

3.2. Overall picture of organizational data at high level of abstraction

3.3. Business rules

3.4. Entity-relationship diagram & Descriptions of entity types

3.5. Relationships between entities

Conceptual Level

Step 3.2. Overall picture of organizational data at high level of abstraction

Figure 1-6 Example business function-to-data entity matrix (Steps 3.1 and 3.2)

The functions that the database need to support

Business Functions \ Data Entity Types	Customer	Product	Raw Material	Order	Work Center	Work Order	Invoice	Equipment	Employee
Business Planning	X	X						X	X
Product Development		X	X		X			X	
Materials Management		X	X	X	X	X		X	
Order Fulfillment	X	X	X	X	X	X	X	X	X
Order Shipment	X	X		X	X		X		X
Sales Summarization	X	X		X			X		X
Production Operations		X	X	X	X	X		X	X
Finance and Accounting	X	X	X	X	X		X	X	X

X = data entity is used within business function

Step 3.3. Business Rules (BR)

- Are **statements** that define or **constrain** some aspect of the business
- Are derived from **policies**, **procedures**, **events**, **functions**
- Assert business structure
- Control/influence business behavior
- Are expressed in terms familiar to end users
- Are **automated** through DBMS software

We use **BR** to design an organization's database, i.e. to determine **entities**, **attributes**, **attribute types** (optional/required, …), **identifiers**, **relationships** (optional/mandatory), etc.

Class Activity 1.1 (5 BRs in 5 minutes)

1. Read the case study below and write a list the business rules.

- The Online Camera Shop database keeps records of customers and their wish lists for managing market campaign.
- Every customer needs to register on the site by providing his/her personal information including name, home address, email address and contact phone number. The system will generate a unique customer number for each customer when his/her registration is successfully completed.
- Customers can create their own private wish lists, which contain their favourite cameras they wish to purchase. The system will automatically generate a unique number for a wish list of a customer in order to trace all the wish lists. The system also stores the date that the wish list of the customer has been created. Within a wish list, customer can add as many cameras as he or she likes. For each entry in a wish list, which is for one and only one camera, customer can make a brief note, for example, "this is my first choice". A camera is described by a unique camera number, a camera name and a standard price.
- A customer may be allocated to an employee of the shop. An employee is described by a unique employee no, a name and a contact phone number. In order to attract more sales, the top manager will regularly propose promotion deals for customers to buy specific cameras. Each promotion deal is described by a start date, an expiry date and a discount rate for a specific camera model. Some employees also act as supervisors for other employees.

Solution to Class Activity 1.1

A Good Business Rule Is:

- **Declarative**– what, not how
- **Precise–** clear, agreed-upon meaning
- **Atomic**– one statement
- **Consistent**– internally and externally
- **Expressible**– structured, natural language
- **Distinct**– non-redundant
- **Business-oriented**– understood by business people

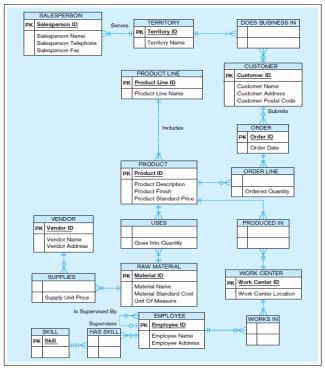
A Good Data Name Is:

- Related to business, not technical, characteristics
- Meaningful and self-documenting
- Unique
- Readable
- Composed of words from an approved list
- Repeatable
- Written in standard syntax

Steps 3.4 & 3.5: E-R Model Constructs (Conceptual Level)

Data Model: A graphical system used to capture the nature and relationships among data made up **entities**, **attributes** and **relationships**.

- The most common data modeling representation is entity relationship diagram (ERD)



- **Entities**
  - Entity– person, place, object, event, concept
  - Entity Type– collection of entities
- **Attributes**
  - Properties or characteristics of an entity or relationship type
- **Relationships**
  - Relationship instance–link between entities
  - Relationship type– category of relationship…link between entity types

Entities

Entities

- **Entity** – a person, a place, an object, an event, or a concept in the user environment about which the organization wishes to maintain data (often corresponds to a **table**)
- **Entity type** – a collection of entities that share common properties or characteristics … An entity type is a template for entity instances
- **Entity instance** – A single occurrence of an entity type (often corresponds to a **row** in a table)

Entities: E-R Model Constructs (Entities, Attributes and Instances)

NOTE: This is not a good design!

EMPLOYEE
Employee_Number
Employee_F_Name
Employee_L_Name
Address (Street, City
State, Zip_Code)
Date_Hired
Birth_Date
Skills
Years Employed

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Hire	Birth_Date	Skills	Year_Employed
1123	Sara	Brown	UTS	1/1/2014	1/1/1985	C++, Java	5
1456	Jake	Cooper	32/50 …	5/8/2013	7/8/1990	C#, Java, Python	6
7892	Fahimeh	Ramezani	12/97 …	2/3/2013	8/7/1987	Python, C++	6
8764	Ricky	Romanous	45/34 …	2/3/2015	4/3/1982	C++, C#	4

Entities: E-R Model Constructs (Entities, Attributes and Instances)

Address

Entity type: EMPLOYEE

Attributes	Attribute Data Type	Example Instance	Example Instance
Employee Number	CHAR (10)	642-17-8360	534-10-1971
Name	CHAR (25)	Michelle Brady	David Johnson
Street	CHAR (30)	100 Pacific Avenue	450 Redwood Drive
City	CHAR (20)	San Francisco	Redwood City
State	CHAR (2)	CA	CA
Zip Code	CHAR (9)	98173	97142
Date Hired	DATE	03-21-1992	08-16-1994
Birth Date	DATE	06-19-1968	09-04-1975

FIGURE 2-3 Entity type EMPLOYEE with two instances

Entities: Entity Characteristics

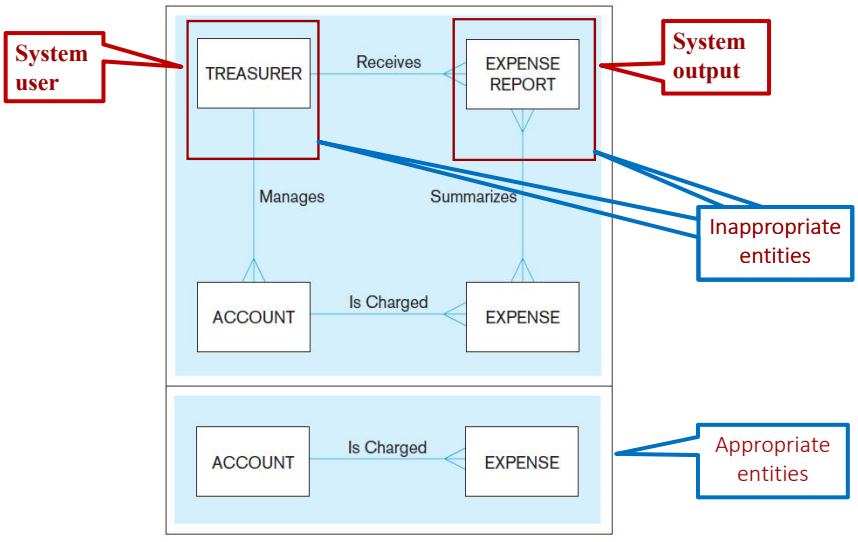
➢**Should be:**

- An object that will have **many instances** in the database
- An object that will be composed of **multiple attributes**
- An object that we are trying to **model**

➢**Should not be:**

- A **user** of the database system
- An **output** of the database system (e.g., a report)

Entities: Example of inappropriate entities (Figure 2-4)



Report Form Example

Example Report Form

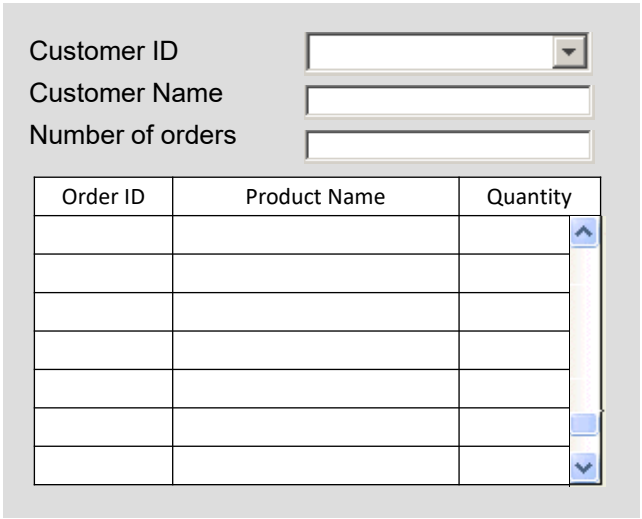
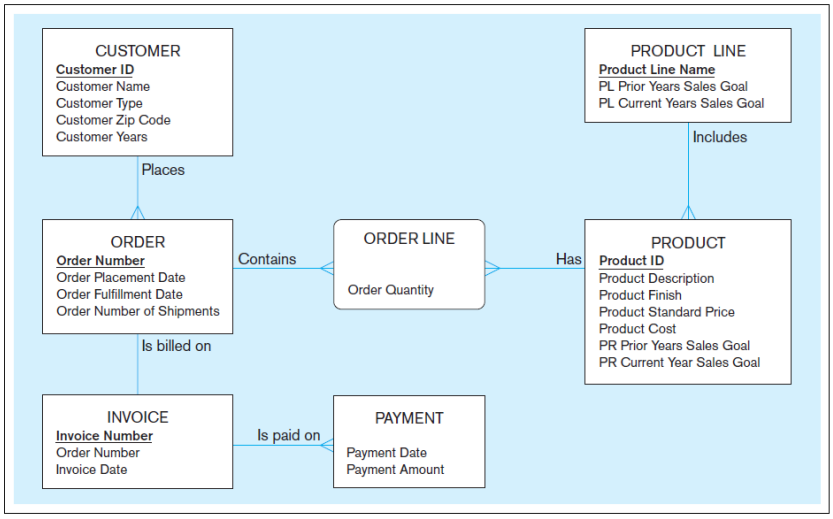
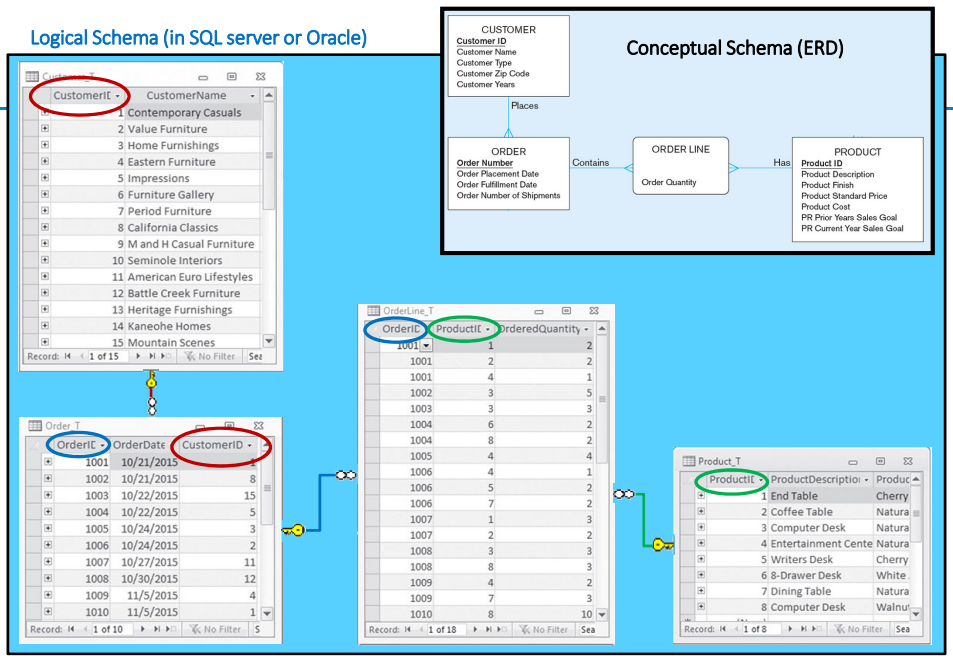
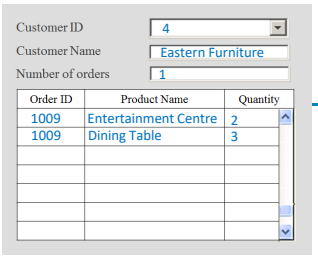


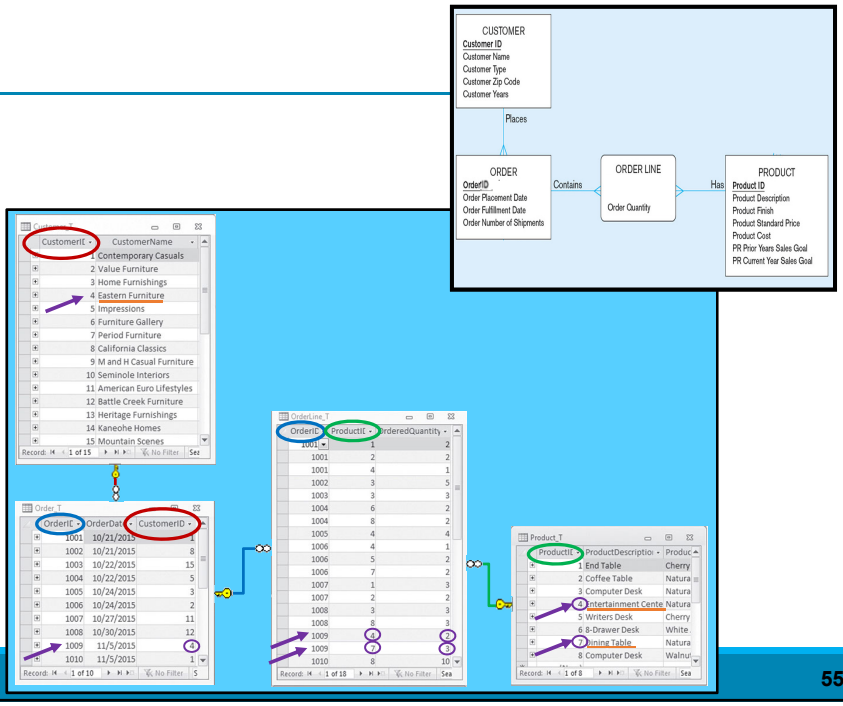
Figure 1-15 Project data model for Home Office product line marketing support system







The information of a form can be retrieved from different tables



Example: The overall perspective to the database designer role

We also provided you the following information through this example:

Example of a cconceptual schema (ERD) (Lectures 1, 2 and 3) →

that is converted to a logical schema (Lectures 4) →

then creating a related user interface/report form
(Programmer/database designer role)

Attributes

Attributes

- **Attribute**–property or characteristic of an entity or relationship type (often corresponds to a **field** in a table)
- Classifications of attributes:
  - **Required** versus **Optional** Attributes
  - **Simple** versus **Composite** Attribute
  - **Single-Valued** versus **Multivalued Attribute**
  - **Stored** versus **Derived** Attributes
  - **Identifier** Attributes

Attributes: Required vs. Optional Attributes

Entity type: STUDENT

Attributes	Attribute Data Type	Required or Optional	Example Instance	Example Instance
Student ID	CHAR (10)	Required	876-24-8217	822-24-4456
Student Name	CHAR (40)	Required	Michael Grant	Melissa Kraft
Home Street	CHAR (30)	Required	314 Baker St.	1422 Heft Ave
Home City	CHAR (20)	Required	Centerville	Miami
Home State	CHAR (2)	Required	OH	FL
Home Zip Code	CHAR (9)	Required	45459	33321
Major	CHAR (3)	Optional	MIS	

Required – must have a value for every entity (or relationship) instance with which it is associated

Optional – may not have a value for every entity (or relationship) instance with which it is associated

Attributes: Example of Required vs. Optional Attributes

- Is BoyNo a required attribute?
- Is GirlMoblie a required attribute?

How we can answer these questions?

BoyNo	BoyName	BoyMobile	GirlNo
1	Adam	0414 1236	1
2	Bob	0414 1237	2
3	Charlie	0414 1238	2
4	Dennis	<null>	<null>

Primary keys

Foreign keys

GirlNo	GirlName	GirlMobile
1	Alice	0414 1234
2	Bonnie	0414 1235

Primary keys

Attributes: Simple vs. Composite Attributes

Composite attribute – An attribute that has meaningful component parts (attributes)

The address is broken into component parts

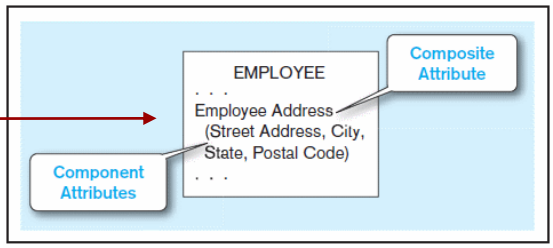


Figure 2-7 A **composite** attribute

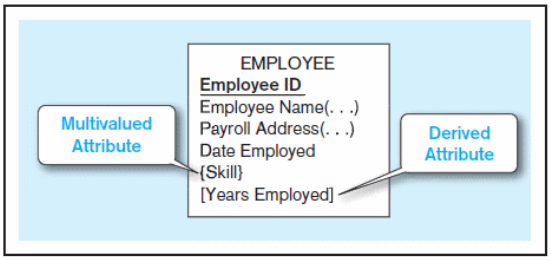
Attributes: Multi-valued and Derived Attributes

Multivalued – may take on more than one value for a given entity (or relationship) instance

Derived – values can be calculated from related attribute values (not physically stored in the database)

Figure 2-8 Entity with **multivalued** attribute (Skill) and **derived** attribute (Years Employed)

Multivalued
an employee can have more than one skill



Derived
Calculated from date employed and current date

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Hire	Birth_Date	Skills	Year_Employed
1123	Sara	Brown	UTS	1/1/2014	1/1/1985	C++, Java	5
1456	Jake	Cooper	32/50 …	5/8/2013	7/8/1990	C#, Java, Python	6
7892	Fahimeh	Ramezani	12/97 …	2/3/2013	8/7/1987	Python, C++	6
8764	Ricky	Romanous	45/34 …	2/3/2015	4/3/1982	C++, C#	4

Attributes: Identifiers (Keys)

- **Identifier (Key) –** an attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
- **Simple** (one attribute) versus **Composite** (combination of attributes) Identifier
- **Candidate Identifier –** an attribute (or combination of attributes) that could be an identifier/key (i.e. satisfies the requirements for being an identifier)

Candidate Keys

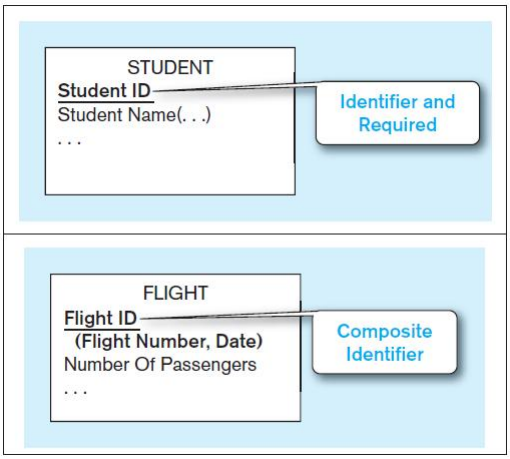
One of the candidate keys will become the primary key

Employee_Number	Employee_F_Name	Employee_L_Name	Address	Date_Hire	Birth_Date	Skills	Year_Employed
1123	Sara	Brown	UTS	1/1/2014	1/1/1985	C++, Java	5
1456	Jake	Cooper	32/50 …	5/8/2013	7/8/1990	C#, Java, Python	6
7892	Fahimeh	Ramezani	12/97 …	2/3/2013	8/7/1987	Python, C++	6
8764	Ricky	Romanous	45/34 …	2/3/2015	4/3/1982	C++, C#	4

Note: Names are not a good choice to be a PK.

Attributes: Simple and Composite Identifier Attributes (Figure 2-9)

(a) Simple identifier attribute

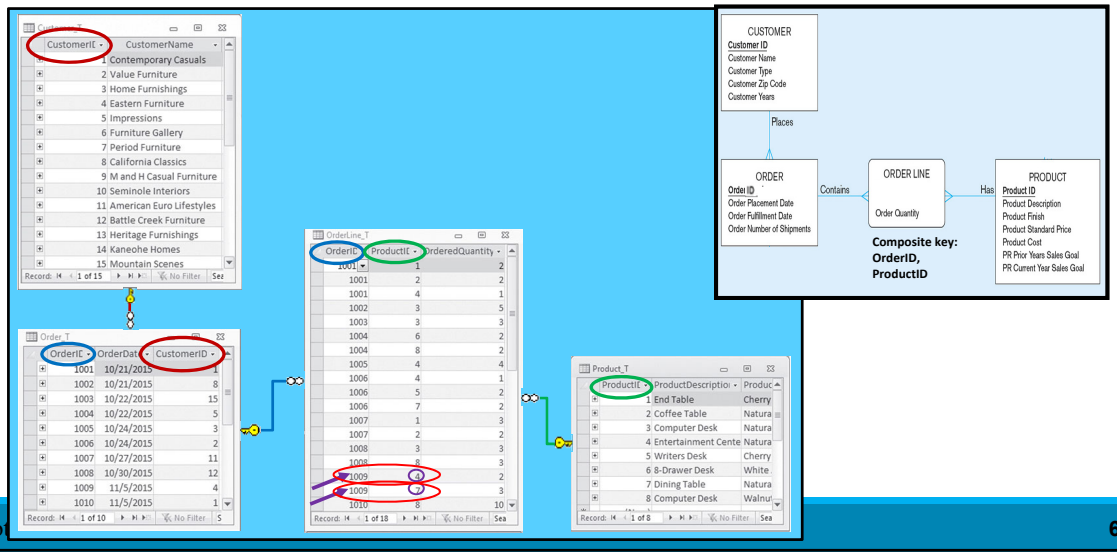


(b) Composite identifier attribute

The identifier is boldfaced and underlined

Attributes: Example of Simple/Composite Identifier

Business Rule: any order can be related to many products (related to OrderLine_T)



Attributes: Naming Attributes

- Name should be a singular noun or noun phrase
- Name should be unique
- Name should follow a standard format
  - + e.g. **[Entity type name { [Qualifier] }] Class**
- Similar attributes of different entity types should use the same qualifiers and classes

Class Activity 1.2 (2 minutes)

2. Which definition is NOT correct?

a. Attribute is property or characteristic of an entity or relationship type

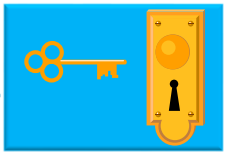
b. Business rules are derived from policies, procedures, events, functions

c. System outputs can be appropriate entities

d. None of them

Relationship

Key Fields



- Keys are special fields that serve two main purposes:

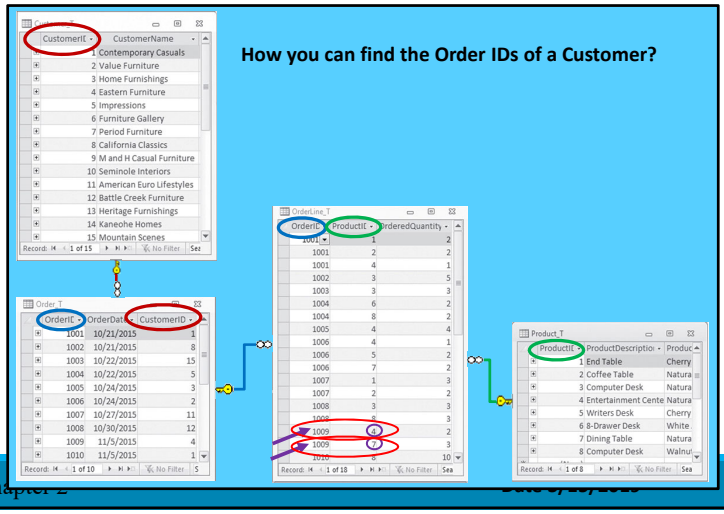
  - ***Primary keys*** are **unique** identifiers of the relation.
    - Examples include employee numbers, social security numbers, etc.
    - This guarantees that all rows are unique.

  - ***Foreign keys*** are identifiers that enable a **dependent** relation to refer to its **parent** relation

- Keys can be ***simple*** (a single field) or ***composite*** (more than one field).
- Keys usually are used as indexes to speed up the response to user queries *(more on this in Chapter 5).*

Primary key/Foreign key

Business Rule: any order can be related to many products (related to OrderLine_T)

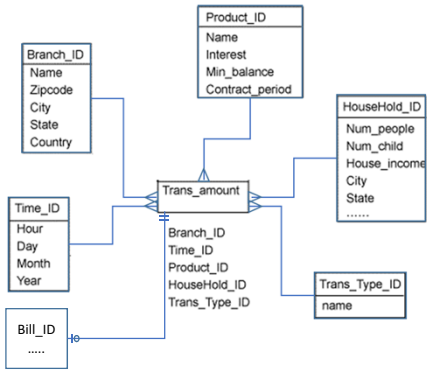


If two relations (tables) have a relationship, then the PK of the parent relation will be a FK in the dependent relation …

Question: What would be the FK in a relation (table) that need to have a relationship with OrderLine_T?

Surrogate Key for Transaction entity

We can keep "BranchID, TimeID, ProductID, HouseHoldID and TransTypeID" unique and use system generated ID (Surrogate key) as well.



Then the FK of Bill entity will be the "Surrogate key" not "BranchID, TimeID, ProductID, HouseHoldID, TransTypeID"

Criteria for Identifiers

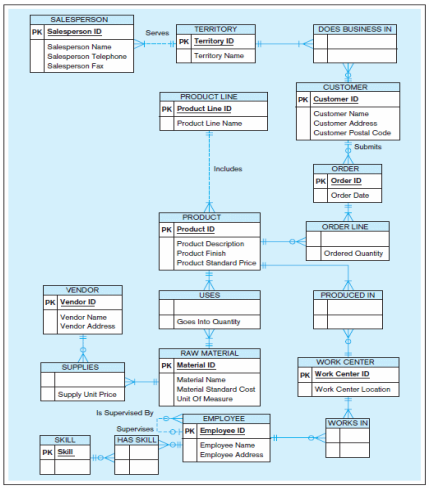
- Choose Identifiers that
  - Will not change in value
  - Will not be null
- Avoid intelligent identifiers
  - e.g., containing locations or people that might change
- Substitute new, simple keys for long, composite keys (create a Surrogate Key).

Relationship (Step 3.5)

- **Relationship instance:**

  link between entities (corresponds to **primary key-foreign key** equivalencies in related tables)

- **Relationship type:**
  category of relationship…link between entity types



Class Activity 1.3 (5 minutes)

3. Use the following BR to determine related entities, attributes, attribute types (optional/required, …), identifiers, relationships, etc.

- **BR1:** Every customer **needs to** register on the site by providing his/her personal information including name, home address, email address and contact phone number.
- **BR2:** The system will generate a unique customer number for each customer when his/her registration is successfully completed.

- **Entities**:
- **Attributes**:
- **Attribute types**:
- **Identifier**:
- **Relationships**:

Class Activity 1.4 (5 minutes)

4. Provide an entity that has composite identifier. Draw the correspond table to the entity with some sample data that shows the composite identifier uniquely identifies each row (individual instance) of the table. The table should have at least 3 rows of data.

- **Answer:**
- Entity Name:
- Composite Identifier:

Summary

- ✓ Define terms
- ✓ Understand basic concepts: data, metadata, and information.
- ✓ Understand importance of data modeling
- ✓ First steps in the database development process
- ✓ Business rules
- ✓ Write good names and definitions for entities, relationships, and attributes

Next Lecture …

3. Modeling Relationships:

3.1. Relationship Types vs. Relationship Instances

3.2. Degree of Relationships

3.3. Cardinality of Relationships

3.4. Multiple Relationships Between Entities

3.5. Multivalued Attributes Can be Represented as Relationships

3.6. Relationships Can Have Attributes

3.7. Associative Entity– Combination of Relationship and Entity

3.8. Identifying Relationship – Weak and Strong Entities

Notations

4.1. Basic E-R Notation

4.2. Crow's foot Notation

Message from previous students ☺
Angelo Athanasiou (DF Grade HD)

- **Why read the text book:**
  The modern database management textbook covers everything more in-depth than the lectures and will greatly help with understanding any areas that are unclear, the textbook is also available from the UTS library so students don't have to pay to access it. Older editions of the textbook can also be obtained for free and contain the same relevant information.

- **What to learn:**
  Learn how a relational database uses relations, cardinality, etc. because if you don't understand those concepts early on the subject won't be as clear as it progresses.

  Learn how SQL statements affect a database and what they do, as it is important to understand **how they work** instead of just understanding what they do, such as knowing why a certain output is given instead of just knowing what to do to get a certain output.

- **To aid with the transition from ERD to SQL,**
  Microsoft Access can be used to understand how things work as you can view the ERD, as well as use SQL to gain output. What I like about using Microsoft access to help people visualize is because you can use QbE to compare how a query would be undertaken in SQL.
  Links: How to use the Query By Example (QBE) grid | lynda.com tutorial:
  https://www.youtube.com/watch?v=X9vyzpdUWHs