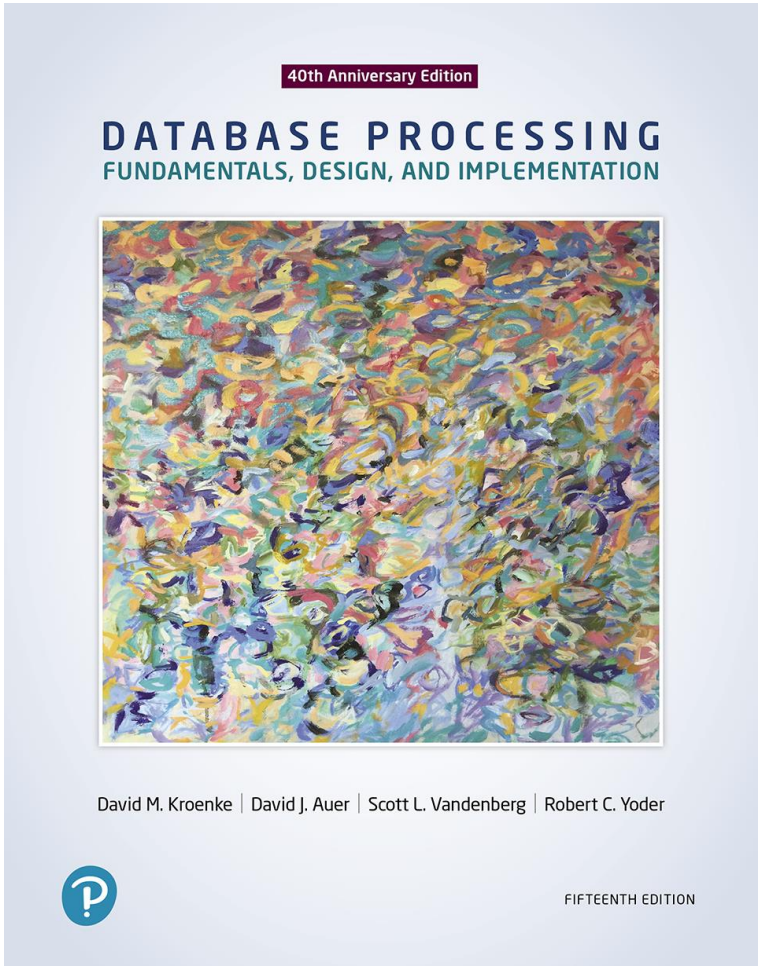
Database Processing: Fundamentals, Design, and Implementation

15th Edition



Chapter 3

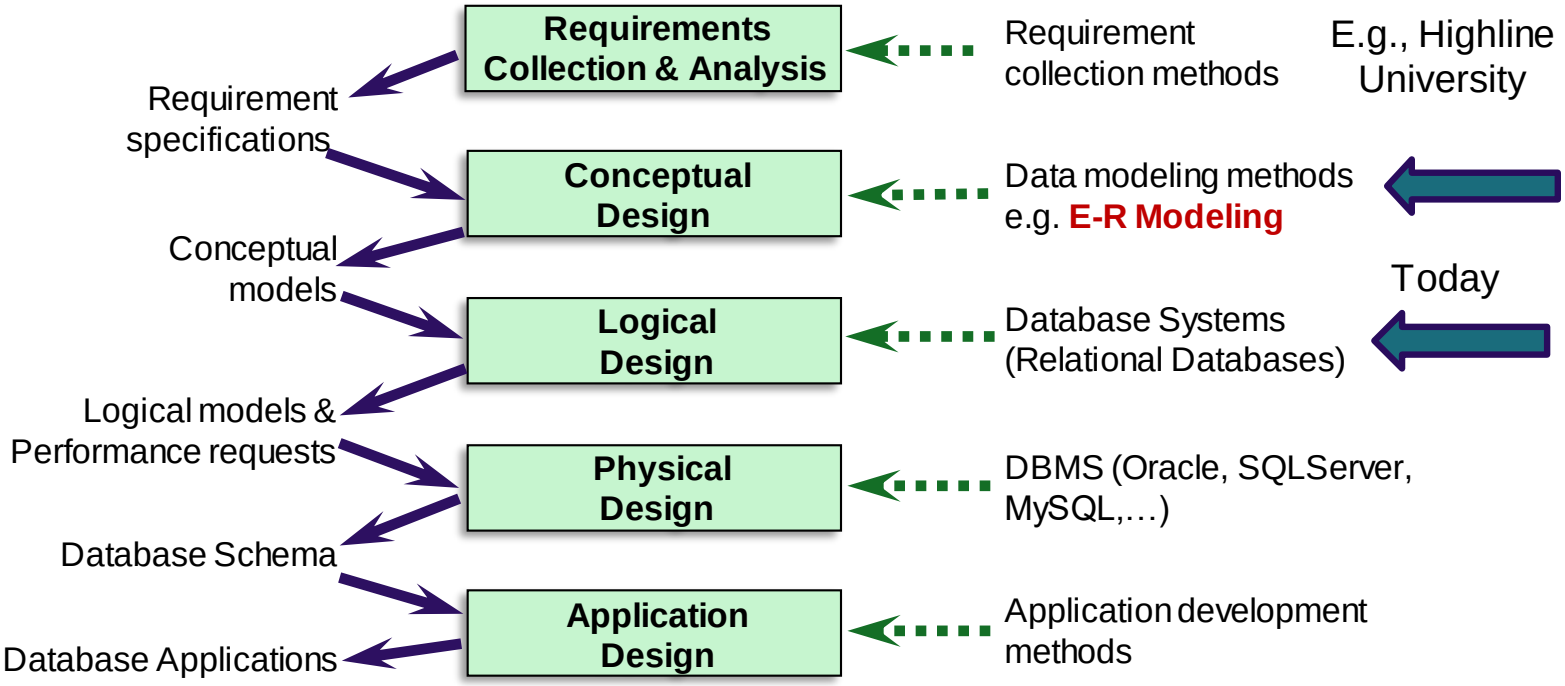
Logical Design:

Relational Model

Learning Objectives

- Understand basic relational data model terminology
- Understand characteristics of relations
- Identify primary, candidate, and composite keys
  - Hint: These will come from the entities in the conceptual model
- Identify possible insertion, deletion, and update anomalies in a relation
  - Basic commands for insertion, deletion, and update
- Learn how to transform a conceptual model into a logical model
  - E.g. entity-relationship model to relational model
  - Coming

Database Design and Implementation



Constructs:

Conceptual design – entities, relationships, attributes, min/max cardinalities [High level of abstraction]
Relational design -- relations (1 construct)

Recall from Session 1: How we described a database.

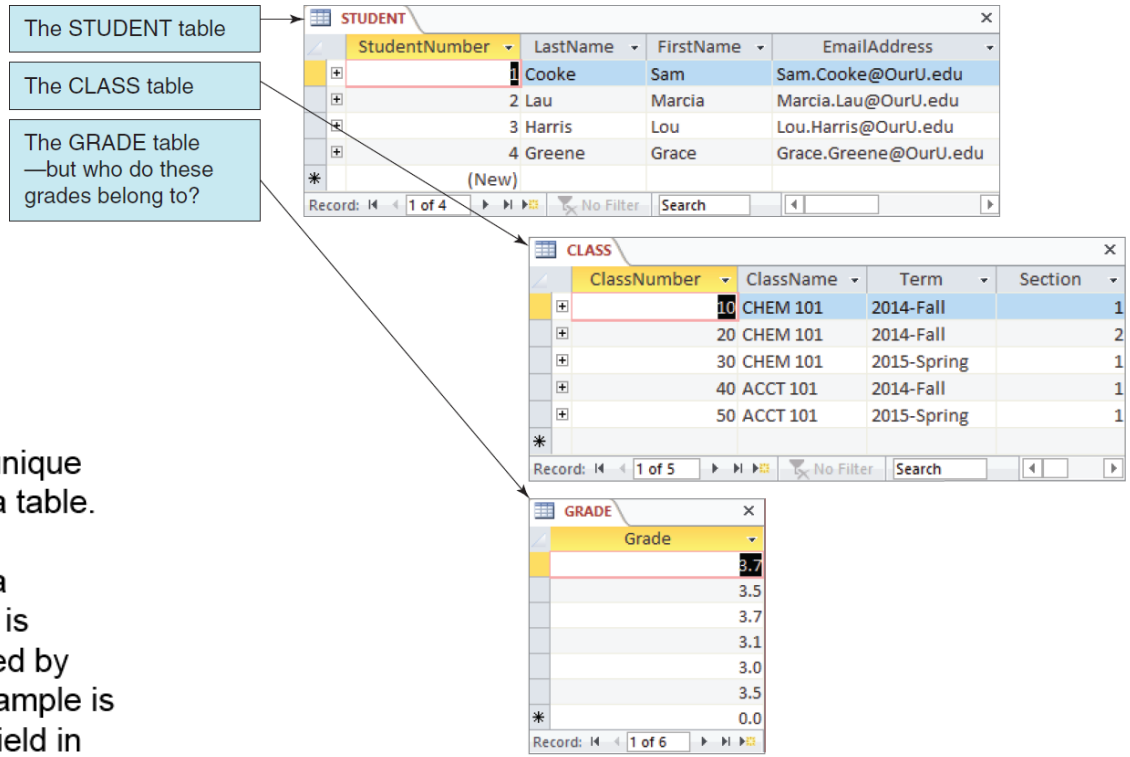
A database is a set of data that has a regular structure and is organized in such a way that a computer can easily retrieve the desired results.

Structure – structure of relational data model dictates how data stored in the database implementation. Well-known characteristics of relational databases enables database developers to create databases with needed, proper structure.

Organized -- rules for transforming a conceptual model into a relational model with appropriate structure

What was wrong with this example (from Session 1: The Nature of Data)

Data in Tables



Recall:

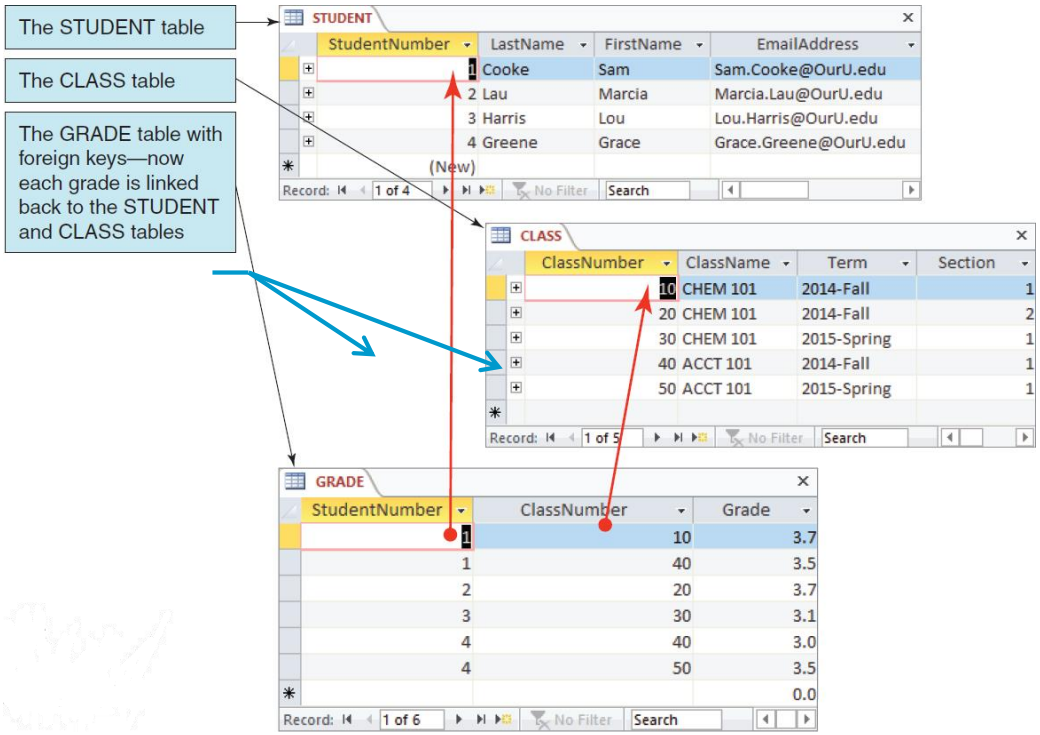
A **Primary Key** is a unique identifier field within a table.

A **Surrogate Key** is a primary key field that is automatically assigned by the computer. An example is the StudentNumber field in the STUDENT table above.

Course: focuses on how to design good databases

Why is this better? (from Session 1: The Nature of Data)

Characteristic of Databases: Related Relations (Tables)
Challenge: proper and efficient design



KROENKE AND AUER - DATABASE PROCESSING, 14th Edition © 2016 Pearson Education, Inc.

Figure 3-1: How Many Tables?

ORDER_ITEM

	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	201000	1	300.00	300.00
2	1000	202000	1	130.00	130.00
3	2000	101100	4	50.00	200.00
4	2000	101200	2	50.00	100.00
5	3000	100200	1	300.00	300.00
6	3000	101100	2	50.00	100.00
7	3000	101200	1	50.00	50.00

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

SKU_ITEM

	OrderNumber	SKU	Quantity	Price	SKU_Description	Department	Buyer
1	1000	201000	1	300.00	Half-dome Tent	Camping	Cindy Lo
2	1000	202000	1	130.00	Half-dome Tent Vestibule	Camping	Cindy Lo
3	2000	101100	4	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
4	2000	101200	2	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers
5	3000	100200	1	300.00	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
6	3000	101100	2	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
7	3000	101200	1	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers

Answer: The number of tables (relations) is based on the conceptual model.

Figure 3-2: A Very Strange Table!

PRODUCT_BUYER

	BuyerName	SKU_Managed	CollegeMajor
1	Pete Hansen	100100	Business Administration
2	Pete Hansen	100200	Business Administration
3	Pete Hansen	100300	Business Administration
4	Pete Hansen	100400	Business Administration
5	Pete Hansen	100500	Business Administration
6	Pete Hansen	100600	Business Administration
7	Nancy Meyers	101100	Art
8	Nancy Meyers	101100	Info Systems
9	Nancy Meyers	101200	Art
10	Nancy Meyers	101200	Info Systems
11	Cindy Lo	201000	History
12	Cindy Lo	202000	History
13	Cindy Lo	203000	History
14	Jenny Martin	301000	Business Administration
15	Jenny Martin	301000	English Literature
16	Jenny Martin	302000	Business Administration
17	Jenny Martin	302000	English Literature

To understand why this is a very odd table, suppose that **Nancy Meyers** is assigned a new, **SKU 101300**! What addition should we make to this table? One or two rows?

But First--

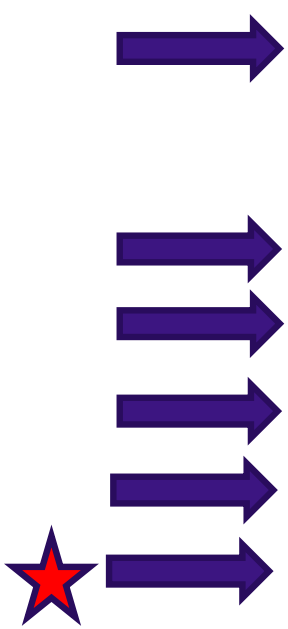
- We need to understand:
  - Structure of relational model
  - Terminology related to relational model

A little history: Relational Model

- Created by Engineer: E.F. Codd, 1970
  - Used mathematics: Standard model for commercial DBMS products
- "Stood the test of time"

Why? based on mathematics: "relational algebra"

Figure 3-3
Important Relational Model Terms



Important Relational Terms
Relation
Functional Dependency
Determinant
Candidate Key
Composite Key
Primary Key
Surrogate Key
Foreign Key
Referential integrity constraint
Normal forms
Multivalued dependency

Figure 3-4: Characteristics of Relations

Relational DBMS products store data about entities in relations, which are a special type of table.

A **relation** is a two-dimensional table that has the following characteristics:

Learn these characteristics!

Characteristics of Relations
Rows contain data about an entity.
Columns contain data about attributes of the entities.
All entries in a column are of the same kind.
Each column has a unique name.
Cells of the table hold a single value.
The order of the columns is unimportant.
The order of the rows is unimportant.
No two rows may be identical.

Important concept: Domain Integrity Constraint

- Requirement that all of the values in a column are of the same kind -- ***domain integrity constraint***.
- Term domain means a grouping of data that meets a specific type definition.
  - **FirstName** could have a domain of names such as *Albert, Bruce, Cathy, David, Edith*, and so forth.
  - All values of **FirstName** *must* come from the names in that domain.
- Columns in different relations may have the same name.

Note: Understanding the domain is part of the work needed to structure a good relational database. Different than semantics of "application domain."

Figure 3-5: Sample EMPLOYEE Relation

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-1102
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Note: An employee relation comes from an employee entity in a conceptual model.
The employee relational "represents" the employee entity.

Figure 3-6 Tables That are Not Relations: Multiple Entries per Cell

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-2102 Fax: 518-834-9911 Home: 518-723-8795
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101 Fax: 518-834-9912 Home: 518-723-7654
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Question: Why might this be a problem when we want a database that has structure?

Figure 3-7 Tables That Are Not Relations: Table with Required Row Order

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-2102
				Fax:	518-834-9911
				Home:	518-723-8795
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101
				Fax:	518-834-9912
				Home:	518-723-7654
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102

Question: What is the "required row order" here?

Figure 3-8
A Relation with Values of Varying Length

EmployeeNumber	FirstName	LastName	Department	EmailAddress	Phone	Comment
100	Jerry	Johnson	Accounting	JJ@somewhere.com	518-834-1101	Joined the Accounting Department in March after completing his MBA. Will take the CPA exam this fall.
200	Mary	Abernathy	Finance	MA@somewhere.com	518-834-2101	
300	Liz	Smathers	Finance	LS@somewhere.com	518-834-2102	
400	Tom	Caruthers	Accounting	TC@somewhere.com	518-834-1102	
500	Tom	Jackson	Production	TJ@somewhere.com	518-834-4101	
600	Eleanore	Caldera	Legal	EC@somewhere.com	518-834-3101	
700	Richard	Bandalone	Legal	RB@somewhere.com	518-834-3102	Is a full-time consultant to Legal on a retainer basis.

Question: Is the comment column valid? Why or why not?

Figure 3-9 Alternative Terminology

- Although not all tables are relations, the terms *table* and *relation* are often used interchangeably (unfortunately).
- The following sets of terms are equivalent:

Table	Column	Row
Relation	Attribute	Tuple
File	Field	Record

This course: First two rows.

Keys [Revisited, same as entities]

- A **key** is a combination of one or more columns that is used to identify particular rows in a relation.
- A **composite key** is a key that consists of two or more columns.

tdwi.org

The Entity (Relation) Integrity Constraint

- **Entity integrity constraint -- t**he primary key must have unique data values inserted into every row of a table.

- The phrase *unique data values* implies that this column is NOT NULL, and does not allow a NULL value in any row.

Note: The concept of a null value is important.
To be revisited.

Foreign Keys (1 of 2)

Foreign key -- column or composite of columns that is the primary key of a table other than the one in which it appears.

The term arises because it is a key of a table *foreign* to the one in which it appears as the primary key.

Foreign key – concept for relations

*Understand how/why [Next time]

Foreign Keys (2 of 2)

Text: The primary keys of the relations are underlined and any foreign keys are in *italics in text.* (Italics not required for course work).

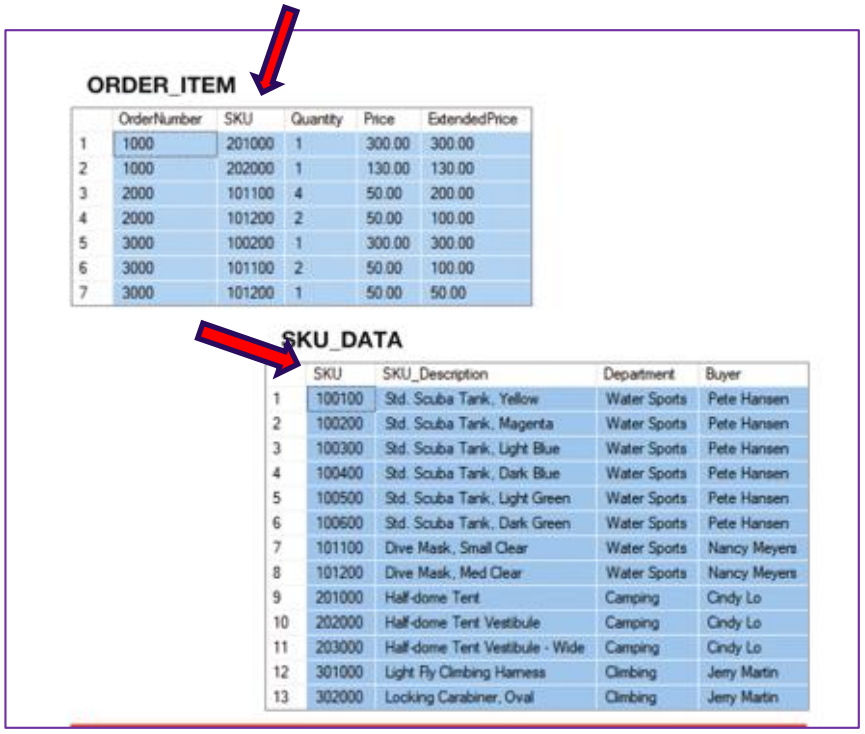
DEPARTMENT (DepartmentName, BudgetCode, ManagerName)
EMPLOYEE (EmployeeNumber, EmployeeLastName, EmployeeFirstName, *DepartmentName*)

Interpretation

Referential Integrity Constraint

Referential integrity constraint: statement that limits values of the foreign key to those already existing as primary key values in corresponding relation:

SKU in ORDER_ITEM must exist in SKU in SKU_DATA



ORDER_ITEM

	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	201000	1	300.00	300.00
2	1000	202000	1	130.00	130.00
3	2000	101100	4	50.00	200.00
4	2000	101200	2	50.00	100.00
5	3000	100200	1	300.00	300.00
6	3000	101100	2	50.00	100.00
7	3000	101200	1	50.00	50.00

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

Foreign Key with Referential Integrity Constraint

SKU_DATA (**SKU**, SKU_Description, Department, Buyer)

ORDER_ITEM (**OrderNumber**, ***SKU***, Quantity, Price, ExtendedPrice)

Where ORDER_ITEM.SKU must exist in SKU_DATA.SKU

SKU_DATA

	SKU	SKU_Description	Department	Buyer
1	100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
2	100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
3	100300	Std. Scuba Tank, Light Blue	Water Sports	Pete Hansen
4	100400	Std. Scuba Tank, Dark Blue	Water Sports	Pete Hansen
5	100500	Std. Scuba Tank, Light Green	Water Sports	Pete Hansen
6	100600	Std. Scuba Tank, Dark Green	Water Sports	Pete Hansen
7	101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
8	101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
9	201000	Half-dome Tent	Camping	Cindy Lo
10	202000	Half-dome Tent Vestibule	Camping	Cindy Lo
11	203000	Half-dome Tent Vestibule - Wide	Camping	Cindy Lo
12	301000	Light Fly Climbing Harness	Climbing	Jerry Martin
13	302000	Locking Carabiner, Oval	Climbing	Jerry Martin

SKU_ITEM

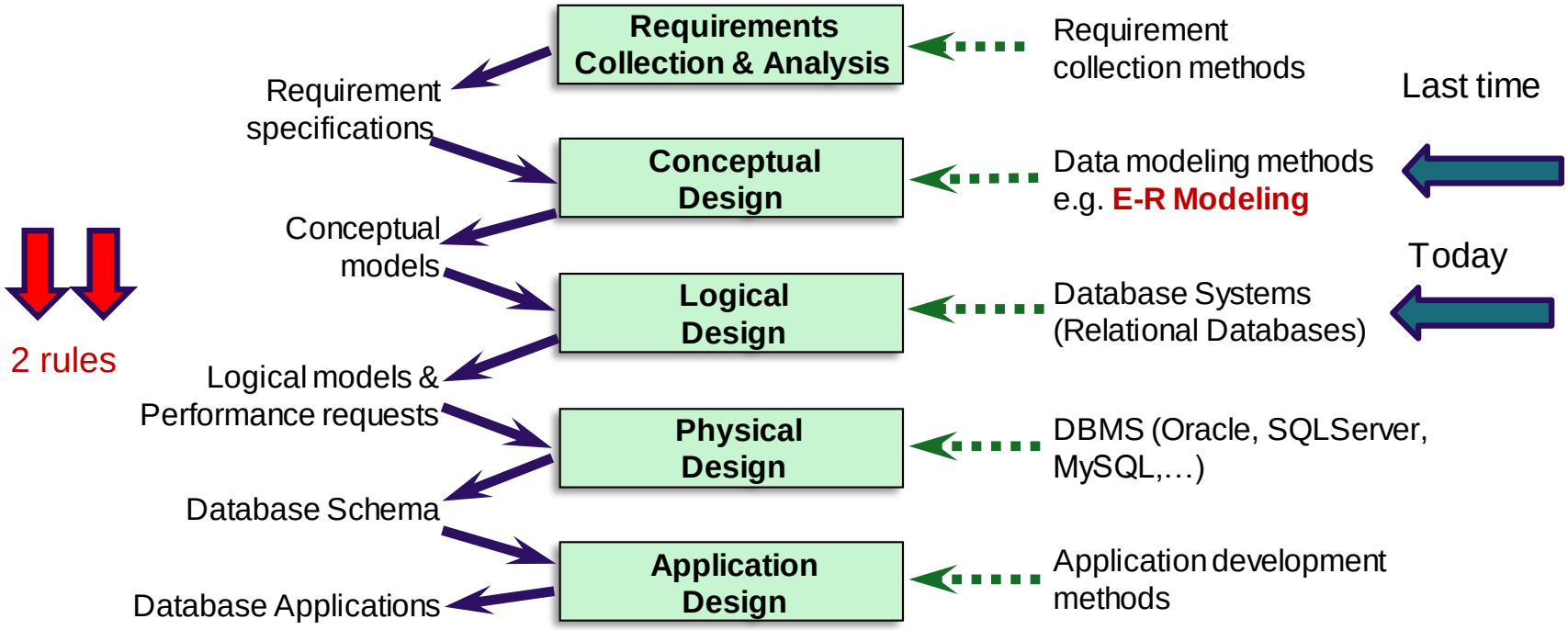
	OrderNumber	SKU	Quantity	Price	SKU_Description	Department	Buyer
1	1000	201000	1	300.00	Half-dome Tent	Camping	Cindy Lo
2	1000	202000	1	130.00	Half-dome Tent Vestibule	Camping	Cindy Lo
3	2000	101100	4	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
4	2000	101200	2	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers
5	3000	100200	1	300.00	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
6	3000	101100	2	50.00	Dive Mask, Small Clear	Water Sports	Nancy Meyers
7	3000	101200	1	50.00	Dive Mask, Med Clear	Water Sports	Nancy Meyers

Underline primary key

Database Integrity (Important)

- Three constraints:
  - **domain integrity constraint**
  - **entity integrity constraint**
  - **referential integrity constraint**
- These three constraints, taken as a whole, create **database integrity**, which means that the data in a database will be useful, meaningful data.

Transform Conceptual Model to Logical Model



Transformation to Logical Model

Two rules: transform conceptual model (entity-relationship model) to a logical model (relational model).

Called: Transformation / conversion / mapping rules. [Important and will be revisited.]

Rule 1: Each entity becomes a relation (table) with same key

Rule 2: Each relationship represented by foreign key or separate relation. Depends on min/max cardinalities of relationship.

- 1:N – Represented by foreign key
  - (0,1) and (0/1,N) – foreign key (can have null value)
  - (1,1) and (0/1,N) – foreign key (cannot have null value)
- N:M – Represented by creating separate relation
  - (0/1,N) and (0/1,M)
  - Key is concatenation (joining together) of keys of the two entities
  - Relationship attributes become non-keys

1:1 – foreign key

- (1,1) and (1,1) – foreign key in either relation (decide based upon usage)
- (0,1) and (1,1) [or (1,1) and (0,1)] – use foreign key such that null values not allowed.

Conclusion

- Introduction to relational data model
- Important terminology
- Translation rules for conceptual to logical models
- [More on this coming]