Database Management System (DBMS)

L-2:

Relational Data Model

Dr. Kamruddin Nur

Associate Professor, Computer Science
kamruddin.nur@gmail.com

Lecture Content

- Relational Data Model
- Relation Schema
- Relational Database Schema
- Relational Keys
- Relational integrity Constraints

Background..

- **Network and Hierarchical Database Systems**
  - Emerged in late 1960s
  - Complex Data Structures
  - No Separation between logical and physical data description
  - Navigational programming languages used

Relational Data Model

- Introduced in 1970 by E. F. Codd
- Based on Mathematical foundation
- Complete separation of logical and physical structure
- Complete data model
- solves problems with previous models
- Has become a dominant model
- Basic Terms and Concepts
  - domain, attribute, relation, tuple
  - Relation schema, instance, constraints

A Sample Relational Database

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	*CourId*	*Points*	*Dept*
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	*CourId*	*Grade*
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Relational Data Models Objectives

- To allow high degree of data independence

  - Application program must not be affected by internal data representation

- To deal with data semantics, consistency, and redundancy problems
- To enable set-oriented data manipulation language

Relational Data Structure

- **Relation:**

  A relation is a table with columns and rows of logically related data.

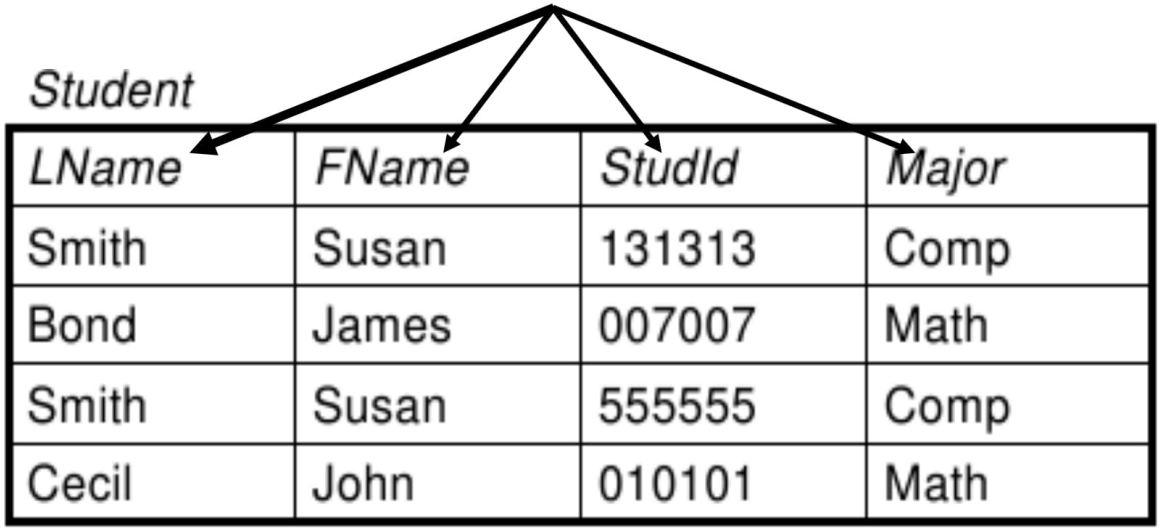
Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Relational Data Structure cont.

- **Attribute:**

  An attribute is a named column of a relation.



Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Relational Data Structure cont.

- **Domain:**

  A domain is a set of allowable values for one or more attributes.

- A set of values, $D = \{ D_i \mid i = 1,...,n \}$

  - D = Domain name

  - D_i is a domain element

  For example,

  LNameDom = { 'Susan', 'James', 'John',...}

  FNameDom = { 'Smith', 'Bond', 'Cecil',...}

  StudIdDom = {131313,007007,555555,010101,...}

  MajorDom = {Comp, Math,...}

Relational Data Structure cont.

- **Degree:**

  The degree of a relation is the number of attributes it contains.

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Here, in the degree of Student relation is 4.

Relational Data Structure cont.

- **Tuple:**

  A tuple is a row of a relation.

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Here, in the Student relation the number of tuple is 4.

Relational Data Structure cont.

- **Cardinality:**

  The Cardinality of a relation is the number of Tuples it contains.

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Here, the cardinality of Student relation is 4.

Relational Data Structure cont.

- **Relational Database:**

  A collection of normalized relations with distinct relation names.

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	*CourId*	*Points*	*Dept*
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	*CourId*	*Grade*
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Relation

- To understand the meaning of the term relation, we have to review some concepts of mathematical term.

Suppose two sets, D_1 = {2,4}and D_2 = {1,3,5}

The Cartesian product of D_1 and D_2 is D_1 x D_2

D_1 x D_2 = Set of all ordered pairs such that first element is a member of D_1 and the second element is a member of D_2

D_1 x D_2 = {(2,1), (2,3), (2,5), (4,1), (4,3), (4,5)}

- ***Any subset of this Cartesian product is a relation.***

e.g. We could say a relation R = {(2,1), (4,1)}

Relation

- We can extend the concept Set to Domain, since Domain is a collection (set) of valid element.

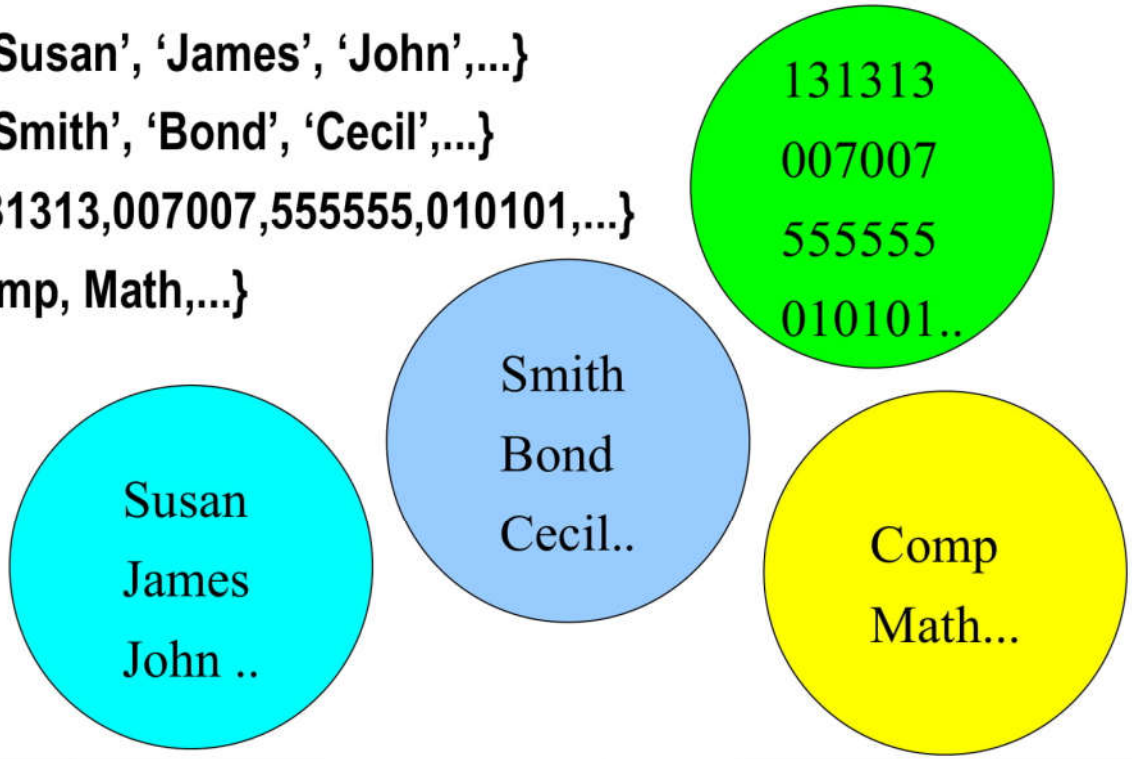
  For example,

  LNameDom = { 'Susan', 'James', 'John',...}

  FNameDom = { 'Smith', 'Bond', 'Cecil',...}

  StudIdDom = {131313,007007,555555,010101,...}

  MajorDom = {Comp, Math,...}



Database Relations

- **Relation Schema:**

  "A named relation defined by the set of attribute and domain name pairs."

  Let $A_1, A_2, \ldots\ldots, A_n$ be attributes with domains $D_1, D_2, \ldots\ldots\ldots, D_n$

  Then the set, $\{A_1:D_1, A_2:D_2, \ldots\ldots\ldots, A_n:D_n\}$ is a relation schema

- **Relational Database Schema:**

  "A set of relation schemas, each with a distinct name."

  If $R_1, R_2, \ldots\ldots\ldots, R_n$ are a set of relation schema then we can say,

  Relational Database Schema $R = \{R_1, R_2, \ldots\ldots\ldots, R_n\}$

Properties of Relations

A relation has the following properties:

- A relations has a distinct name within a relational database schema
- Each cell of a relation contains exactly one atomic (single) value
- Each attribute has a distinct name
- The value of an attribute are all from the same domain
- Each tuple is distinct; there are no duplicate tuples
- The order of attributes has no significance
- The order of tuples has no significance

Relational Keys

- Keys are special fields of a relation mainly for identification.
- Two main purposes -

  - To identify a record (tuple) uniquely in a relation. [Primary Keys]

  - To identify or relate to another relation records. [Foreign Keys]
- Keys can be simple (a single field)or composite (more than one field).

Relational Keys cont.

- **Superkey:**

  "An attribute or set of attributes, that uniquely identifies a tuple within a relation."

  [- A super key uniquely identifies, however, a superkey may contain additional attributes that are not necessary for unique identification.
  - We are interested in identifying superkeys that contain minimum number of attributes necessary for unique identification!!!]

- **Candidate Key:**

  "A superkey such that no proper subset is a superkey within the relation "

  [- A candidate key is a superkey with minimal attributes.]

Relational Keys cont.

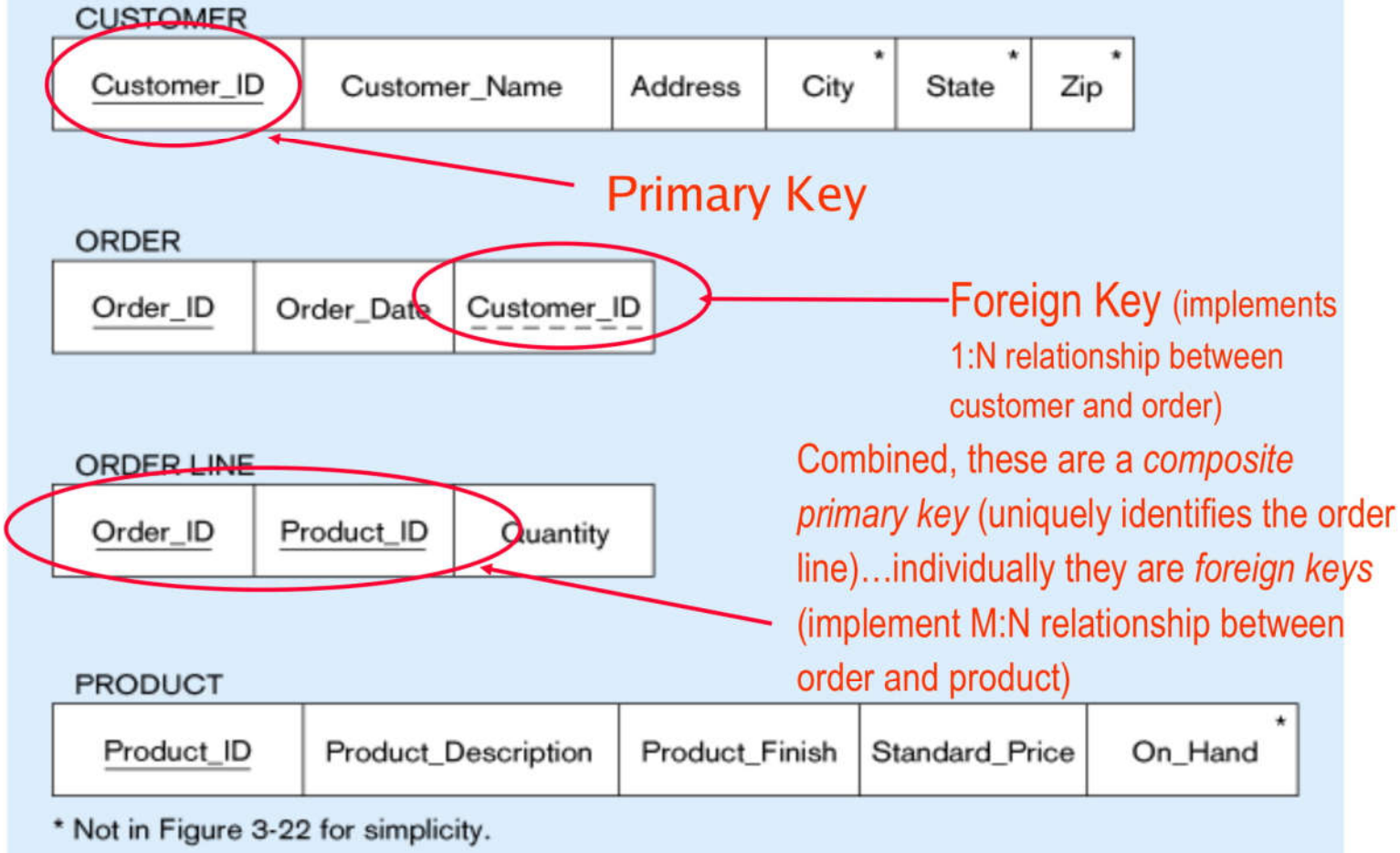
- **Primary Key**:

  "The candidate key that is selected to identify tuples uniquely within the relation."

- **Foreign Key**:

  "An attribute, or a set of attributes, within one relation that matches the candidate key of some (possibly the same) relation ."

Primary Key & Foreign Key cont.



Relational Integrity Constraints

An attribute can have a valid / allowed value of its Domain.

That's why we have to consider two important integrity rules -

- Entity Integrity

- Referential Integrity

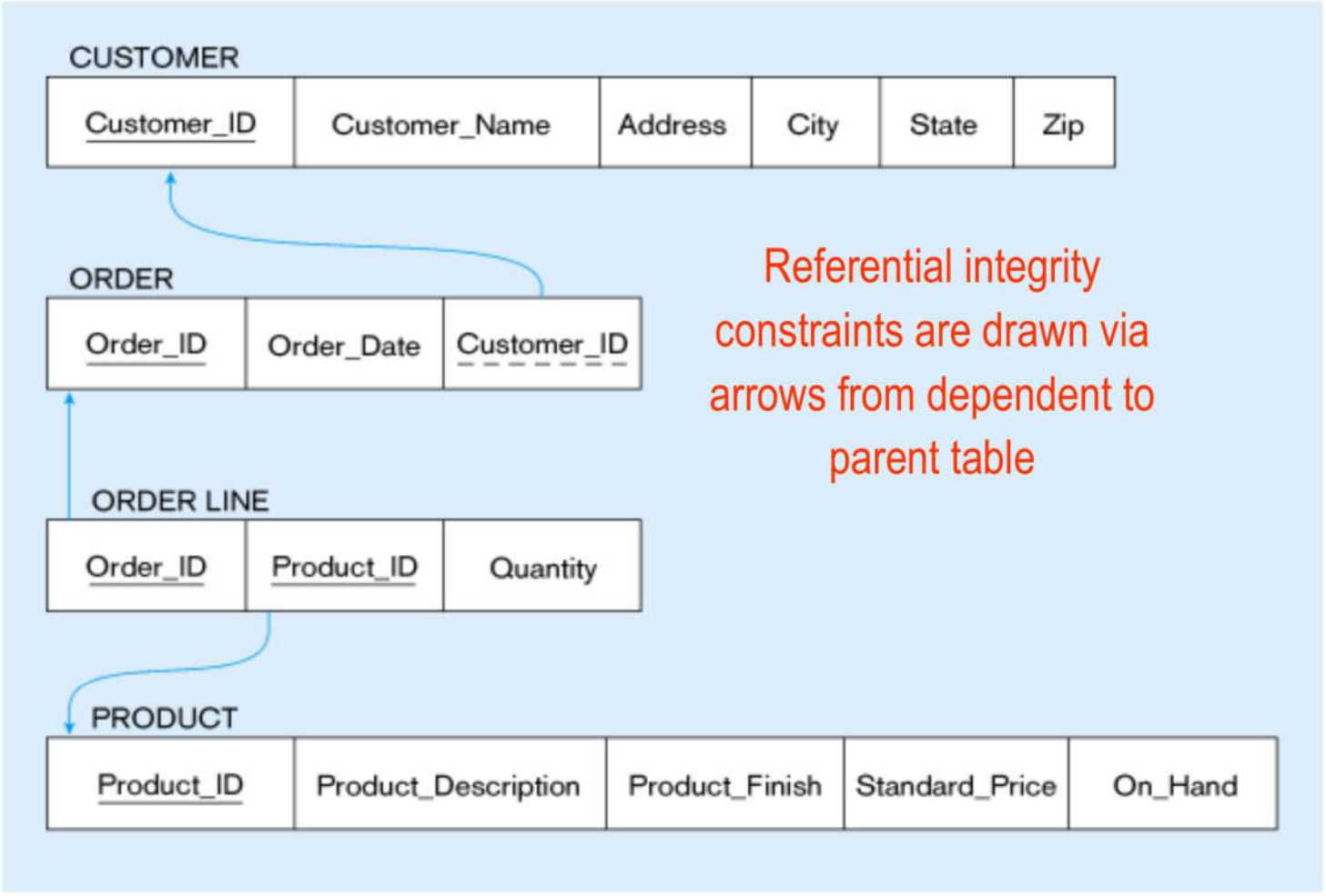
- **Entity Integrity Constraint**:

  "In a base (physical database) relation, no attribute of a primary key can be null."

- **Referential Integrity Constraint**:

  "If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null."

Relational Integrity Constraints



Relational Integrity Constraints

Is there any relational integrity violation in the relational database below?

Student

LName	*FName*	*StudId*	*Major*
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

PName	*CourId*	*Points*	*Dept*
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	*CourId*	*Grade*
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Summary

- Relational Data Model
- Relation Schema
- Relational Database Schema
- Relational Keys : Superkeys, Candidate Keys, Primary keys, Foreign keys
- Relational integrity Constraints : Entity Integrity Constraints, Referential Integrity Constraints