DATABASE DESIGN BCNS1103C

Session 2: An Introduction to Relational Database System and Relational Integrity

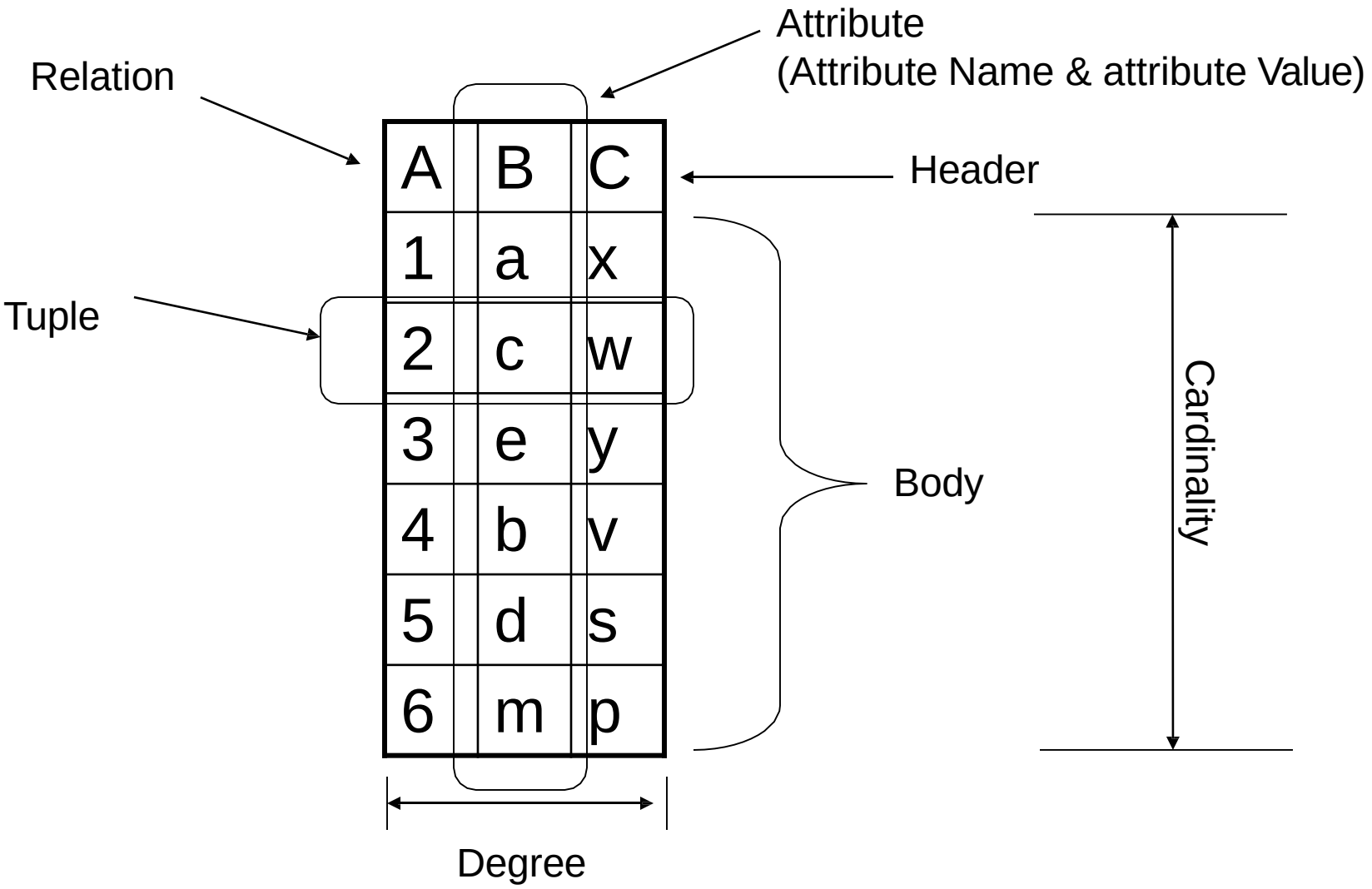
Objectives

- This session covers the following topics:
  - Understanding the three parts of the Relational Model
  - Explain various concepts in the definition of the relational data structure, and the properties of a relation as a mathematical set.
    - Entity (3 types), Attributes (4 Types), Relationship (3 Types)
    - Cardinality Constraint (4 Types),
    - Business Rules,
    - Null Values,
    - Keys (6 Types)
    - Constraint (6 Types)
  - Understand the importance and objectives of data integrity in a relational database.
  - Describe the three types of relational integrity and which rule deals with what problem.
  - Describe how to maintain referential integrity.
  - Understand the need for application-specific integrity in addition to relational integrity.
  - Procedure for Designing Relational Databases

Three Parts of the Relational Model

- Structural Part
  - The type of structures (Building blocks) from which the database is constructed.
- Manipulative Part
  - The operations which are used for retrieving and updating data in the database
- Integrity Part
  - The rules that all valid databases must obey

Relational Data Structures



An Example

CUSTOMER

CUST_ID	Name	Credit Status	Business_Type	etc
1	Smith	Excellent	Retail	-
2	Jones	Good	Wholesale	-
3	Brown	Fair	Multiple	-

ORDER

O_ID	O_DATE	CUST_ID	etc
100	01/02/2011	2	-
101	04/02/2011	1	-
102	12/02/2011	2	-

Entity Defined

An entity is:

- "Something" of significance to the business about which data must be known
- A name for a set of similar things that you can list
- Usually a noun
- Examples: objects, events, people
- Entities have instances.
- An instance is a single occurrence of an entity.

Entities and Instances

Entities	Instances
PERSON	Mahatma Gandhi, George Washington
PRODUCT	Nike Air Jordan, Gibson Les Paul Custom
PRODUCT TYPE	Shoe, Video Game
JOB	Electrician, IT Technician
SKILL LEVEL	Beginner, Expert
CONCERT	U2 at the Palladium, Beyoncé at the Greek Theatre L.A.
ANIMAL	Dog, Cat
CAR	Volkswagen Beetle, Toyota Corolla

Entities and Instances

- A Dalmatian, a Siamese cat, a cow and a pig are instances of ANIMAL
- A convertible, a sedan and a station wagon are instances of CAR
- Some entities have many instances and some have only a few
- Entities can be:
  - Tangible, like PERSON or PRODUCT
  - Intangible, like SKILL LEVEL
  - An event, like CONCERT

Entities and Instances

- Is DOG an instance or an entity?

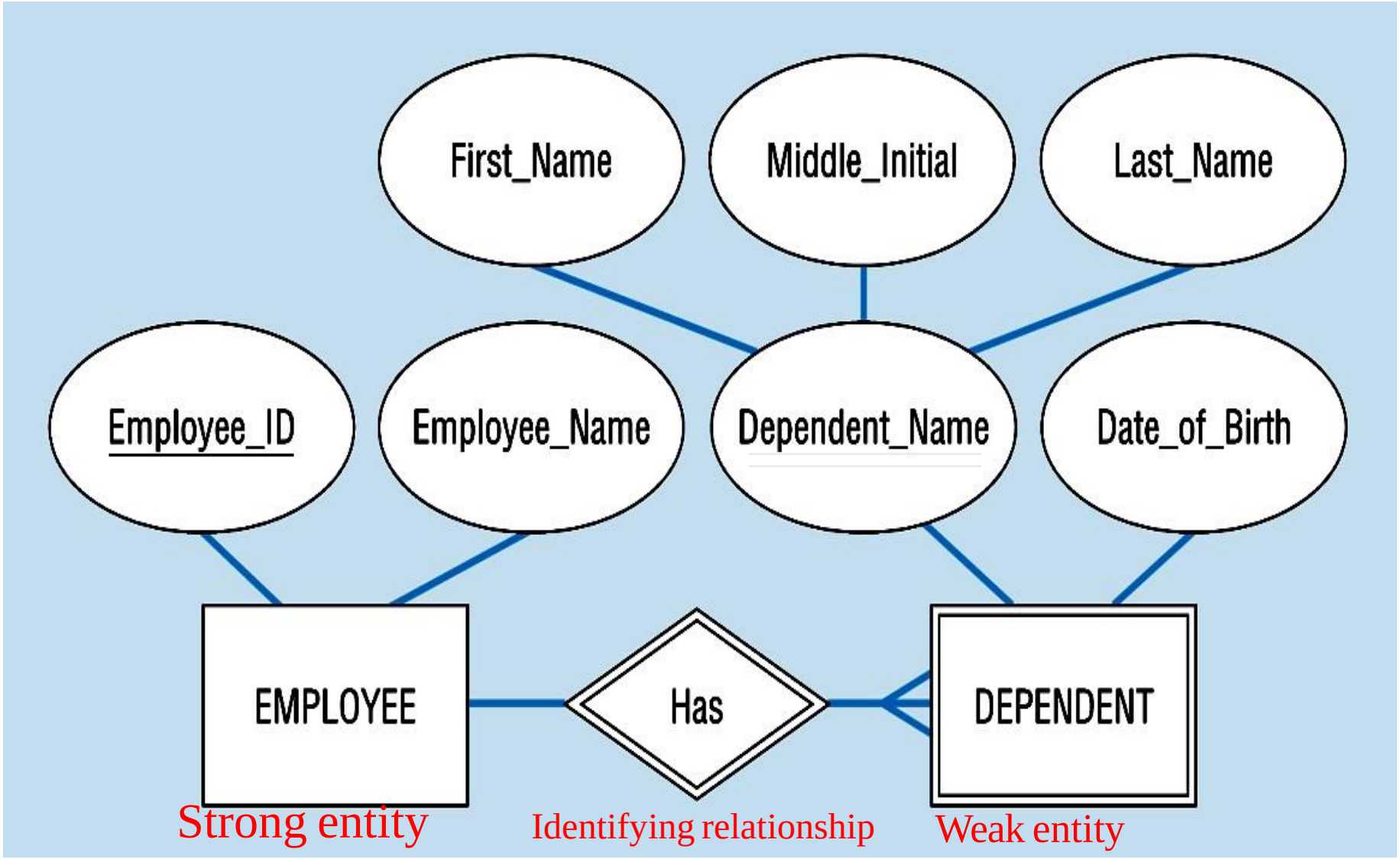
Entities and Instances

- Is DOG an instance or an entity?
- It depends:
  - If we consider many different kinds of animals, it makes sense to think of the entity ANIMAL to include instances DOG, CAT, HORSE and so on.
  - But what if we run a dog-breeding business? We will need to keep data on many different breeds of dog, but not on other species of animal.
  - For a dog-breeder, it is more natural to think of an entity DOG to include instances TERRIER, POODLE, LABRADOR and so on.

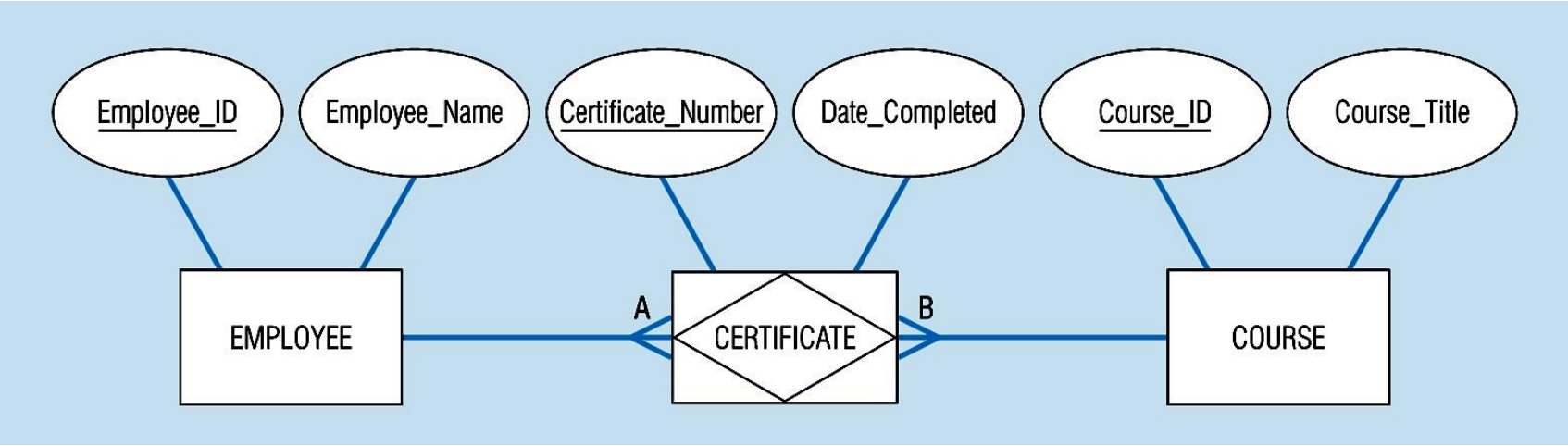
Entities

- Entity - anything about which data are to be collected and stored
  - Employee, Supplier, Parts etc
- 3 Types of Entities:
  - Strong Entity
  - Weak Entity
  - Associative Entity

Strong and weak entities



An associative entity (CERTIFICATE)



Associative entity involves a rectangle with a diamond inside. Note that the many-to-many cardinality symbols face toward the associative entity and not toward the other entities

Attributes

- Attribute - a characteristic of an entity
  - Customer: Cust_no, Cust_name etc
- 4 Types of Attributes:
  - Single or simple attribute
  - Multi-valued Attribute
  - Composite Attribute
  - Derive Attribute

What is an Attribute?

- Like an entity, an attribute represents something of significance to the business.
- An attribute is a specific piece of information that helps:
  - Describe an entity
  - Quantify an entity
  - Qualify an entity
  - Classify an entity
  - Specify an entity
- An attribute has a single value.

Attributes

- Attributes have values. An attribute value can be a number, a character string, a date, an image, a sound, etc.
- These are called "data types" or "formats." Every attribute stores one piece of data of one specific data type.

Entities	Attributes
CUSTOMER	family name, date of birth, shoe size, town of residence, email
CAR	model, weight, catalog price
ORDER	order date, ship date
JOB	title, description
TRANSACTION	amount, transaction date
EMPLOYMENT CONTRACT	start date, salary

Attributes

- What is the data type of each attribute in CUSTOMER?
- For example: family name is a character string. Attributes are single-valued. Each attribute can have only one value (at any point in time) for each instance of the entity.

Entities	Attributes
CUSTOMER	family name, age, shoe size, town of residence, email
CAR	model, weight, catalog price
ORDER	order date, ship date
JOB	title, description
TRANSACTION	amount, transaction date
EMPLOYMENT CONTRACT	start date, salary

Attributes

- Some attributes (such as age) have values that constantly change.
- These are called volatile attributes.
- Other attributes (such as order date) will rarely change, if ever.
- These are nonvolatile attributes.
- If given a choice, select the nonvolatile attribute.
- For example, use birth date instead of age.

Attributes

- Some attributes must contain a value—these are mandatory attributes.
- For example: in most businesses that track personal information, name is required.
- Other attributes may either contain a value or be left null—these are optional attributes.
- For example: cell phone number is often optional except in mobile or wireless applications.

Attributes

- Example: Email address could be a mandatory attribute for EMPLOYEE in an email application, but an optional attribute for CUSTOMER in an online catalog.

Attributes

- If we were to model a Human Resource system, we would have an entity to store data for each worker called EMPLOYEE.
- What attributes does EMPLOYEE have?
- Give one or two examples of the values that each EMPLOYEE attribute might contain.

Identifiers

- An EMPLOYEE has a unique identifier (UID).
- A UID is either a single attribute or a combination of multiple attributes that distinguishes one employee from another.
- How do you find a specific employee that works for the company?
- What information uniquely identifies one EMPLOYEE?

Identifiers

- Think about all the students in the classroom.
- Each student is described by several traits or attributes.
- Which attribute or attributes allow you to pick a single student from the rest of the class?
- That is the student's UID.

Cardinality Constraint

- Constraint - a restriction placed on the data
  - Optional One
  - Optional Many
  - Mandatory One
  - Mandatory Many

Cardinality Constraint

- **Optionality of a Relationship**

1.Mandatory Relationship: A mandatory relationship specifies that each instance from an entity must be related to another instance. This is represented by a straight line.

2. **Optional Relationship:** An optional relationship specifies that each instance from an entity may be related to another instance. This is represented by a dashed line.

Relationship

- Relationship - describes an association among entities
  - One-to-One (1:1) relationship
  - One-to-many (1:M) relationship
  - many-to-Many (M:N) relationship

Drawing Relationships

- When drawing a Relationship using the Barker's notation the following rules need to be respected:
  1. A relationship can exist between a maximum of two entities.
  2. A relationship can exist on the same entity.
  3. A relationship has two perspectives.
  4. Both perspectives of a relationship must be labelled.

Drawing Relationships

- When drawing a Relationship using the Barker's notation the following steps need to be taken:
  1. Determine the entities affected by the relationship.
  2. Determine the optionality of the relationship.
  3. Determine the degree of the relationship.
  4. Label the perspectives of the relationship.

Business Rules

- Brief, precise, and unambiguous descriptions of a policies, procedures, or principles within a specific organization
- Apply to any organization that stores and uses data to generate information
- Description of operations that help to create and enforce actions within that organization's environment

Business Rules (continued)

- Must be rendered in writing
- Must be kept up to date
- Sometimes are external to the organization
- Must be easy to understand and widely disseminated
- Describe characteristics of the data as viewed by the company

Business Rules (continued)

- Example:
  - A customer may generate many invoices
  - An invoice is generated by only one customer
- Note: a business rule establish entities, relationships and constraints
- The above rules establish 2 entities CUSTOMER and INVOICE and a 1:M relationship between those two entities.

Business Rules (continued)

- Example:
  - A training session cannot be scheduled for fewer than 10 employees or for more than 30 employees
- This business rule establish a constraint: no fewer than 10 people and no more than 30 people; two entities EMPLOYEE and TRAINING; and a relationship between EMPLOYEE and TRAINING

Discovering Business Rules

Sources of Business Rules:

- Company managers
- Policy makers
- Department managers
- Written documentation
  - Procedures
  - Standards
  - Operations manuals
- Direct interviews with end users

Translating Business Rules into Data Model Components

- Standardize company's view of data
- Constitute a communications tool between users and designers
- Allow designer to understand the nature, role, and scope of data
- Allow designer to understand business processes
- Allow designer to develop appropriate relationship participation rules and constraints
- Promote creation of an accurate data model

Discovering Business Rules (continued)

- Generally, nouns translate into entities
- Verbs translate into relationships among entities
- Relationships are bi-directional

NULL

- Nulls:
  - No data entry (does not mean zero or blank space)
  - Not permitted in primary key
  - Emp_Initial may be null
  - Should be avoided in other attributes
  - Indicates poor database design!!
  - Can represent
    - An unknown attribute value
    - A known, but missing, attribute value
    - A “not applicable” condition

NULL

- Nulls:
  - Can create problems when functions such as COUNT, AVERAGE, and SUM are used
  - Can create logical problems when relational tables are linked

Keys

Key Fields: establish relationships among records in different tables

- Five main types of key fields:
  - Primary / natural keys
  - candidate keys
  - surrogate keys
  - foreign keys
  - composite keys

Data Integrity – the Objective

- Data integrity in a relational database is concerned with three aspects of the data in a database:
  - Accuracy
  - Correctness
  - Validity

Relational Integrity

- Integrity is enforced via integrity rules
- There are three main types of integrity specified in the relational model:
  - Domain Integrity
  - Entity Integrity
  - Referential Integrity

Domain Integrity

- Concerned with the values that may be contained within a particular attribute column
- The domain integrity rules states:

  "Every attribute is required to satisfy the constraint that its values are drawn from the relevant domain"

Entity Integrity

- Concerned with primary keys
- The entity integrity rules states:

  "Every base relation must have a primary key and no component of that primary key is allowed to accept null values"
- Null values
- Function of primary keys
- Implication of accepting nulls

Referential Integrity

- Concerned with foreign keys
- The referential integrity rules states:

  "The database must not contain any unmatched foreign key values"
- Referencing v/s Referenced
- Domain Requirement
- Possible cases for null values

Maintaining Referential Integrity

- Referential Integrity rule itself does not specify how incorrect database states can be avoided
- Foreign Key Processing rules
- Alternatives when dealing with deletion and update
  - Restricted
  - Cascades
  - Nullifies

Application-Specific or User Defined Integrity

- General Integrity Rules
- The above three relational integrity rules
- Additional application-specific or user defined rules
- Only apply to a particular application/database

Constraint

- Rules that restrict the data values that you can enter into a field in a database table
  - Primary Key Constraint
  - Foreign Key Constraint
  - Check Constraint
  - Not Null Constraint
  - Unique Constraint
  - Default !

The Relational Model

- Developed by Codd (IBM) in 1970
- Considered ingenious but impractical in 1970
- Conceptually simple
- Computers lacked power to implement the relational model
- Today, microcomputers can run sophisticated relational database software

The Entity Relationship Model

- Widely accepted and adapted graphical tool for data modeling
- Introduced by Chen in 1976
- Graphical representation of entities and their relationships in a database structure

The Entity Relationship Model (continued)

- Entity relationship diagram (ERD)
  - Uses graphic representations to model database components
  - Entity is mapped to a relational table
- Entity instance (or occurrence) is row in table
- Entity set is collection of like entities
- Connectivity labels types of relationships
  - Diamond connected to related entities through a relationship line

The Entity Relationship Model (continued)

The basic Chen ERD

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

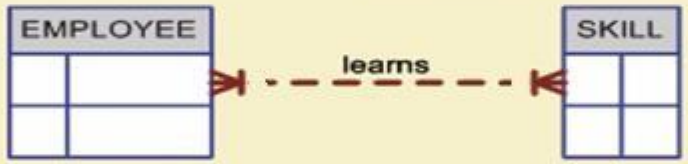
A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

The Entity Relationship Model (continued)

The basic Crow's foot ERD

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

Properties of a Relation

- There are no duplicate tuples
- All the columns name must be different
- Tuples are unordered (from top to bottom)
- Attributes are unordered (from left to right)
- All attributes values are atomic

Procedure for Designing a Relational Database

- Identify data and user needs (System Analysis)
- Construct E-R Model
- Normalize the data
- Transform the normalized E-R model to a relational one.

Summary

- In this session we covered the following topics:
  - The three parts of the Relational Model
  - Explain various concepts in the definition of the relational data structure, and the properties of a relation as a mathematical set.
    - Entity (3 types), Attributes (4 Types), Relationship (3 Types)
    - Cardinality Constraint,
    - Business Rules,
    - Null Values
  - Keys , Data Integrity
  - Constraint (4 Types)
  - Relational Model & ERD
  - Procedure for Designing Relational Databases

Question & Answer?