

CHAPTER FOUR

Logical Database Design

OUTLINES

- 👉 **Introduction**
- 👉 **Normalization**
- 👉 **Functional dependence**

Introduction

- The **relational data** model is a model that achieves complete physical data independence.
 - Applications don't need to know at all how data is organized as long as the database takes care of finding the right data when the application provides table & column names along with primary key.
- **Relation** can be viewed as a “table”. In that table, each row represents a **tuple** of data values and each column represents an **attribute**.
- **Attribute**: A column of a relation designated by name. The name associated should be

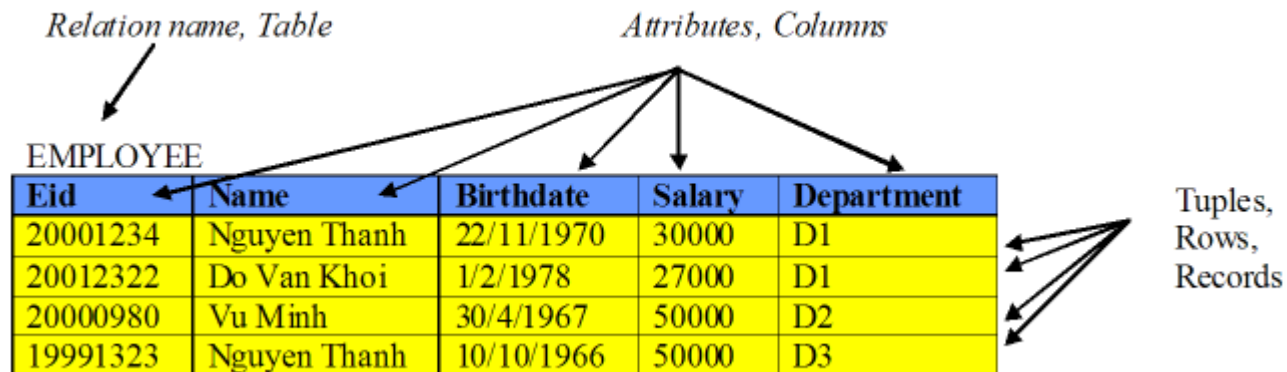
Cont...

▮ **Relation schema** denoted by R is a list of attributes $(A_1, A_2, \dots, A_n)$

▶ The degree of the relation is the number of attributes of its relation schema. **The cardinality of the relation is the number of tuples in the relation.**

▶ Example: relation, relation schema & attribute

Cont...



Cont...

□ Logical database design:

- ▶ It is the process of translating the conceptual design into a logical database design that can be implemented on a chosen DBMS
 - Input: conceptual model (ERD)
 - Output: relational schema, normalized relations
- ▶ A relation is a named, two-dimensional table of data. Table consists of rows (records) and columns (attributes or fields)
- ▶ Correspondence of Relation with E-R Model
 - ▶ Relations (tables) correspond with entity types.
 - ▶ Columns correspond with attributes
 - ▶ Rows correspond with entity instances

Cont...

□ **NOTE:** The word *relation* (in relational database) is NOT the same as the word *relationship* (in E-R model)

□ Requirements for a table to qualify as a relation are:

- ▶ Every table must have a unique name
- ▶ Every attribute value must be atomic
 - not multi-valued, not composite
- ▶ Every row must be unique
 - can't have two rows with exactly the same values for all their fields
- ▶ Attributes (columns) in tables must have unique names
- ▶ The order of the columns must be irrelevant
- ▶ The order of the rows must be irrelevant

Cont...

▢ Steps to convert ER-Diagram to Relational Table

- ▶ Rule 1: Entity names will automatically be table names
- ▶ Rule 2: Map ATTRIBUTES to COLUMNS
 - ▶ Atomic or single valued or derived or stored attributes will be columns
 - ▶ Decompose composite attributes into child attributes and will be used as columns of the table.
 - ▶ Multi-valued attributes will be mapped to a new table where the primary of the main table will be posted for cross referencing.
- ▶ Rule 3: Relationship will be mapped by using foreign key. Foreign key is a primary or candidate key of one relation used to create association between tables.

Cont...

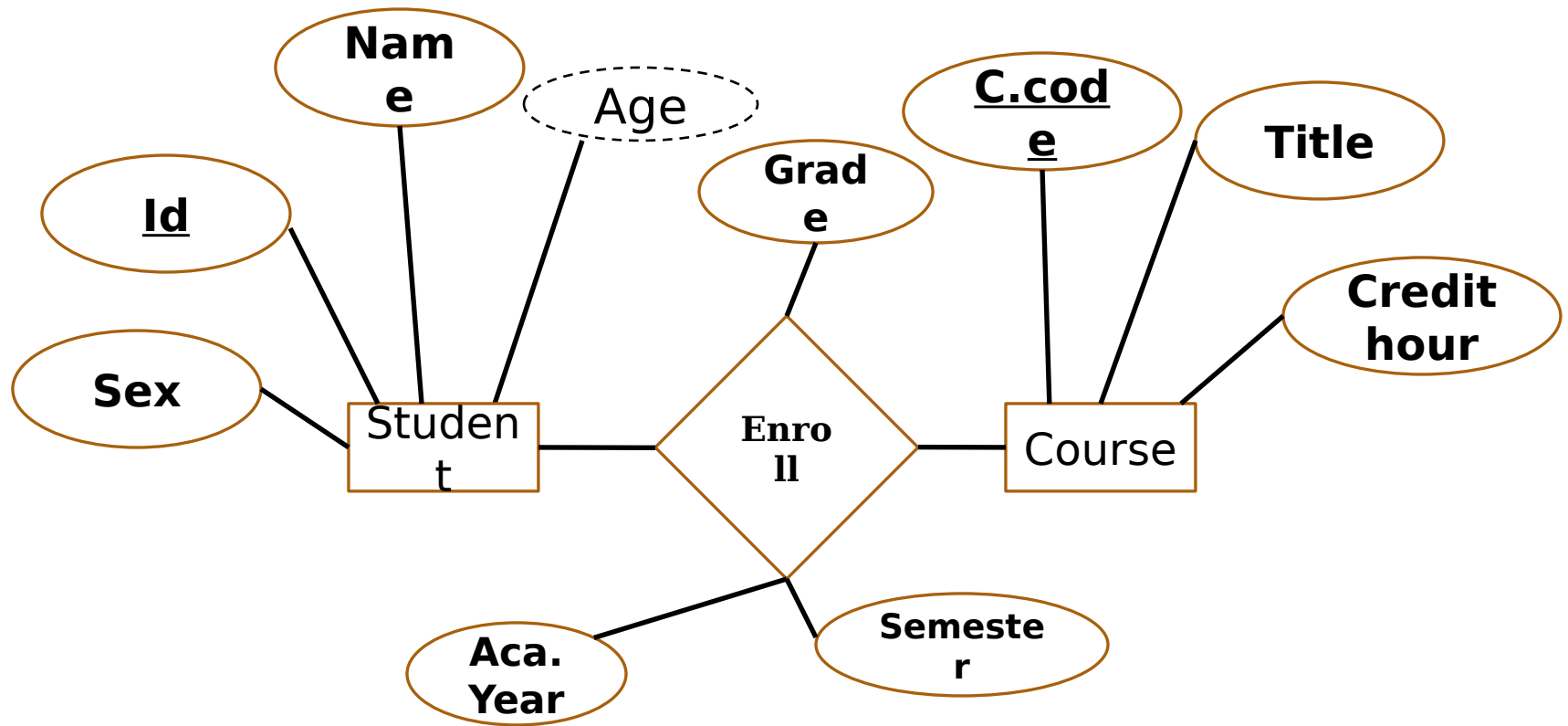
- ▶ ***One-to-One cardinality:*** Post the primary or candidate key of one table in to the other as foreign key.
 - ▶ ***Example:*** a relationship between **Employee** and **Department** where Employee **manages** a department.
 - ▶ The cardinality is **one -to- one** as one Employee will manage only one department and one department will have one manager. Here, the Primary key of Employee can be posted to the Department or the Primary key of the Department can be posted to the Employee as Foreign key.

Cont...

- ▶ **One-to-Many cardinality:** Post the primary or candidate key from the “One” side as foreign key attribute to the “Many” side.
 - ▶ **Example:** for a relationship “**Belongs to**” between Employee(Many) and Department(One). Here, the Primary key of the one side which is Department should be posted to the many side which is Employee as foreign key.
- ▶ **Many-to-Many cardinality:** for a relationships having many-to-many cardinality, one has to create a new table(which is the association entity) and post primary key from the participant entities as foreign key attributes in the new table along with some additional attributes.

Cont...

❖ Example: convert the ff. ER-diagram to Relational table?



Cont...

▢ Answer:

- Mapping Student Entity to relational table

Student

Primary key

<u>Id</u>	FName	MName	LName	Sex	Age
-----------	-------	-------	-------	-----	-----

Mapping Course Entity to relational table

Course

Primary key

- Mapping Course Entity to relational table

<u>C.code</u>	Title	Credit Hour
---------------	-------	-------------

Enroll

Foreign keys

<u>Id</u>	<u>C.code</u>	Semester	Aca.year	Grade
-----------	---------------	----------	----------	-------

Cont...

- The Goal of **relational schema design** is to **avoid update anomalies** and **redundancy**. Anomalies are errors or inconsistencies resulting from redundancy.
 - ▶ **Insertion anomaly**: adding new rows forces user to create duplicate data
 - ▶ **Deletion anomaly**: deleting rows may cause a loss of data that would be needed for other future rows
 - ▶ **Modification anomaly**: changing data in a row forces changes to other rows because of duplication

Data normalization

- ▮ **Normalization** is the process of organizing a database into a structure that preserves the integrity of the stored data and minimizes redundant data.
- ▮ It is the process of decomposing relations with anomalies to produce smaller, ***well-structured*** relations
- ▮ It is a formal process for deciding which attributes should be grouped together in a relation
 - ▶ **Objective:** Primarily a tool to validate and improve a logical design so that it satisfies certain constraints that ***avoid unnecessary data duplication and redundancies in order to help eliminate data anomalies.***

Cont...

- **Database normalization** is a series of steps followed to obtain a database design that allows for consistent storage and efficient access of data in a relational database.
 - ▶ These steps reduce data redundancy and the risk of data becoming inconsistent.
- Normalization is expected to test for:
 - ▶ No repeating elements or groups of elements
 - ▶ No partial dependencies on a composite keys
 - ▶ No dependencies on non-key attributes

Cont...

▣ Normalization stages

- ▶ First normal form (1NF)
- ▶ Second normal form (2NF)
- ▶ Third normal form (3NF)

▣ 1NF (First normal form): A relation in which every attribute value is atomic (i.e. the intersection of each row and column contains one and only one value).

- No multi-valued attributes.
- The first normal form provides the foundation for the second and third forms.

Cont...

Example for 1NF:

Un-normalized data

EmpID	First Name	Last Name	Skill	Skill type	School
10	AA	BB	SQL	Data base	AU
			C++	Programmin g	AAU
11	XX	YY	SQL	Data base	AU
12	BB	CC	Java	Programmin g	AU
			Oracl e	Data base	Jima
			Cisco	Networking	AU

Cont...

First Normal Form(1NF):

✎ Remove all repeating groups. Distribute the multi-valued attributes into different rows and identify a unique identifier for the relation.

<u>EmpID</u>	First Name	Last Name	<u>Skill ID</u>	Skill	Skill type	School
10	AA	BB	1	SQL	Data Base	AU
10	AA	BB	2	C++	Programming	AAU
11	XX	YY	3	SQL	Data Base	AU
12	BB	CC	4	Java	Programming	AU
12	BB	CC	5	Oracle	Data base	Jima
12	BB	CC	6	Cisco	Networking	AU

Cont...

- The resulting table is in 1NF since it contains atomic values at the intersection of each row and column.
- Any problem you observed in the above table?

❖ Second Normal Form(2NF)

- Second normal form is based on the concept of full functional dependency.
- Functional dependency describes the relationship between attributes in a relation.
- ***For example:*** If A and B are attributes of relation R, B is functionally dependent on A (denoted $A \twoheadrightarrow B$), if each value of A is associated with exactly one value of B.

Cont...

- ▶ A relation is in second normal form (2NF), if every non-primary key attribute is functionally dependent on the whole primary key.
- ▶ Thus no non-primary key attribute is functionally dependent on part of the primary key.
- ▶ A table (relation) is in 2NF, if
 - ▶ It is in 1NF
 - ▶ The primary key consists of only one attribute.
 - ▶ Every non-primary key attributes are functionally dependent on the entire primary key.

Cont...

- ▮ To convert a relation to second normal form, we need to remove the functional dependency that violates the rule.
- ▮ To do that, the relation needs to be decomposed in to new relation.
- ▮ **Example:**

R = (studID, studName, courseNo, courseName, Grade, deptName, Dhead)

- Given a student_ID, we can determine the student name
 - But given a student name we cannot determine the student_ID
- Given a course_ID, we can determine the course name.

$F = \{ \text{studID}, \text{courseNo} \rightarrow \text{Grade}$
 $\text{studID} \rightarrow \text{studName},$
 $\text{courseNo} \rightarrow \text{courseName} \}$

Cont...

- Are all the student-course relation dependent on composite keys (studID, courseNo)?
 - *No, because it hold true only for Grade:*
 $\{ \text{studID}, \text{courseNo} \rightarrow \text{Grade} \}$
 - Others are partially dependent on composite key attributes.
- If an attribute which is not a member of the primary key is dependent on some part of the primary key (if we have composite primary key) then that attribute is **partially dependent** on the primary key.

Cont...

▮ **In the above example,**

{ studID → studName } --- (partial dependency)

{ courseNo → courseName, deptName, Dhead } --- (partial dependency)

{ deptName → Dhead } (Transitive dependency)

▮ **To convert the above example to 2NF**

- ▶ Start with 1NF format
- ▶ Write each key component on separate line
- ▶ Write dependent attributes after each key
- ▶ Each component is new table

Cont...

- ▶ Student_Course (studID, courseNo, Grade)
- ▶ Student (studID, Name)
- ▶ Course (courseNo, courseName, deptName, Dhead)

□ ***Third Normal Form (3NF)***

- ▶ 3NF is based on the concept of **transitive dependency**.
- ▶ Transitive Dependency is a condition where
 - ▶ A, B and C are attributes of a relation such that if $A \twoheadrightarrow B$ and $B \twoheadrightarrow C$, then C is transitively dependent on A through B. (Provided that A is not functionally dependent on B or C).

Cont...

- A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.
 - Identify the primary key in the 2NF relation.
 - Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their dominant.
- In the above example,

{ deptName $\rightarrow$ Dhead } (Transitive dependency)

Cont...

▮ 3rd normal form conversion results

- (studID, courseNo, Grade)
- (studID, Name)
- (courseNo, courseName)
- (deptName, Dhead)

Cont...

□ **Exercise:** check the following table whether it satisfied all normalization forms or not?

ID	FNam e	LName	Dep	year	Dormitory
R/ 1001/07	AA	BB	Comp	1	101
R/ 2001/06	AX	BY	IT	2	102
R/ 3001/05	YY	CC	Comp	3	103
R/ 1002/07	QQ	RR	IT	1	101

Cont...

Summary of Data Normalization

▮ First Normal Form (1NF)

- removes repeating groups

▮ Second Normal Form (2NF)

- removes partial dependencies on the primary key

▮ Third Normal Form (3NF)

- removes transitive dependencies on the primary key

Functional dependence

- The logical association between data items that point the database designer in the direction of a good database design are referred to as **determinant** or **dependent** relationship .
- Two data items A and B are said to be in a determinant or dependent relationship if certain values of data item B always appears with certain values of data item A.
 - If the data item A is the determinant data item and B is the dependent data item then the direction of the association is from A to B and not vice versa.

Cont...

- ▶ The essence of this idea is that if the existence of something call it A, implies that B must exist and have a certain value , then we say that **“B is functionally dependent on A”**.
- ▶ We also often express this idea by saying that “A determines B”, or “B is a function of A”, or “A functionally governs B”
- ▶ $X \twoheadrightarrow Y$ holds if whenever two tuples have the same value for X, they must have the same value for Y.
- ▶ The notation $X \twoheadrightarrow Y$ is read as “Y is functionally dependent on X”

Cont...

- ▶ In general, a functional dependency is a relationship among attributes.
- ▶ In relational database, we can have a determinant that governs one or other attributes or several other attributes.
- ▶ Dependency can be classified into partial and full dependencies.
- ▶ **Partial Dependency:**
 - ▶ if an attribute which is not a member of the primary key is dependent on some part of the primary key (if we have composite primary key) then that attribute is partially dependent on the primary key.
 - ▶ Let $\{A, B\}$ is the primary key and C is not key attributes, then if $\{A, B\} \twoheadrightarrow C$ and $B \twoheadrightarrow C$ then C is partially dependent on $\{A, B\}$

Cont...

► Full dependency:

- If an attribute which is not a member of the primary key is not dependent on some part of the primary key but the whole key (if we have composite primary key) then that attribute is fully functionally dependent on the primary key.
- Let $\{A,B\}$ is the primary key and C is not key attribute, then if $\{A,B\} \twoheadrightarrow C$ and $B \not\rightarrow C$ and $A \not\rightarrow C$ does not hold then C fully functionally dependent on $\{A,B\}$

Cont...

► Transitive Dependency

- In mathematics and logic, a transitive relationship is a relationship of the following form: “if A implies B, and B implies C, then A implies C”.
- Example: if Mr. X is a human, and every human is an animal , then Mr. X must be an animal.
- In transitive dependency, if A functionally governs B and B functionally governs C, then A functionally governs C.
- i.e.: $\{(A \twoheadrightarrow B) \text{ and } (B \twoheadrightarrow C)\} = = > A \twoheadrightarrow C$

Cont...

End!