



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

DATABASE MANAGEMENT SYSTEM

BCSC0003

By

Suman Kumar Bhattacharyya

Assistant Professor

Computer Engineering and Application Department

Today we will discuss

- ✓ Database Interfaces
- ✓ Database Development Life Cycle (DDLC)
- ✓ Data Model using Entity Relationship Model

Database Interfaces

Database management system interface is a user interface.

It allows for the ability to input queries to a database without using the query language itself.

✓ User-friendly interfaces provide by DBMS may include the following,

1. Menu-Based Interfaces for Web Clients or Browsing
2. Forms-Based Interfaces
3. Graphical User Interface
4. Natural language Interfaces
5. Speech Input and Output
6. Interfaces for DBA

Database Interfaces

1. Menu based interfaces:- These interfaces presents the user with list of option called menus. Pull down menus are the very important technique.
2. Graphical user interfaces:- It displays a schema to the user in diagrammatic form. Most GUIs use pointing device such as a mouse.
3. Natural language interfaces:- These Interfaces accept request written in English or some other language.
4. Forms base interfaces:- These interface displays a form to each user. User can fill out all the entries to insert new data.
5. Interfaces for parametric user:- parametric user such as bank tellers, often have a small set of operations that they must perform repeatedly. For this a small set of abbreviated commands is included, with the goal of minimizing the number of keystrokes required for each request.
6. Interface for DBA:- most database systems contain privileged commands that can be used only by the dba staff.

Database Development Life Cycle

Database system is a fundamental component of the larger enterprise information system.

The database development life cycle (DDLC) is a process of designing, implementing and maintaining a database system to meet strategic or operational information needs of an organization or enterprise such as:

- Improved customer support and customer satisfaction.
- Better production management.
- Better inventory management.
- More accurate sales forecasting.

The database development life cycle (DDLC) is inherently associated with the software development life cycle (SDLC) of the information system.

Database Development Life Cycle

The different phases of database development life cycle (DDLC) in the Database Management System (DBMS) are explained below –

- Requirement analysis.
- Database design.
- Evaluation and selection.
- Logical database design.
- Physical database design.
- Implementation.
- Data loading.
- Testing and performance tuning.
- Operation.

Database Development Life Cycle

Requirement Analysis

The most important step in implementing a database system is to find out what is needed i.e what type of a database is required for the business organization, daily volume of data, how much data needs to be stored in the master files etc.

In order to collect all this information, a database analyst spends a lot of time within the business organization talking to people, end users and getting acquainted with the day-to-day process.

Database Development Life Cycle

Database Design

In this phase the database designers will make a decision on the database model that perfectly suits the organization's requirement. The database designers will study the documents prepared by the analysis in the requirement analysis stage and then start development of a system model that fulfils the needs.

Database Development Life Cycle

- **Evaluation and selection**
 - In this phase, we evaluate the diverse database management systems and choose the one which perfectly suits the requirements of the organization.
 - In order to identify the best performing database, end users should be involved.
- **Logical database design**
 - Once the evaluation and selection phase is completed successfully, the next step is logical database design.
 - This design is translated into internal model which includes mapping of all objects i.e design of tables, indexes, views, transaction, access privileges etc.,

Database Development Life Cycle

Physical Database Design

This phase selects and characterizes the data storage and data access of the database.

The data storage depends on the type of devices supported by the hardware, the data access methods.

Physical design is very vital because of bad design which results in poor performance.

Database Development Life Cycle

Implementation

Database implementation needs the formation of special storage related constructs.

These constructs consist of storage groups, table spaces, data files, tables etc.

Data Loading

Once the database has been created, the data must be loaded into the database.

The data required to be converted, if the loaded data is in a different format.

Database Development Life Cycle

Operations

In this phase, the database is accessed by the end users and application programs.

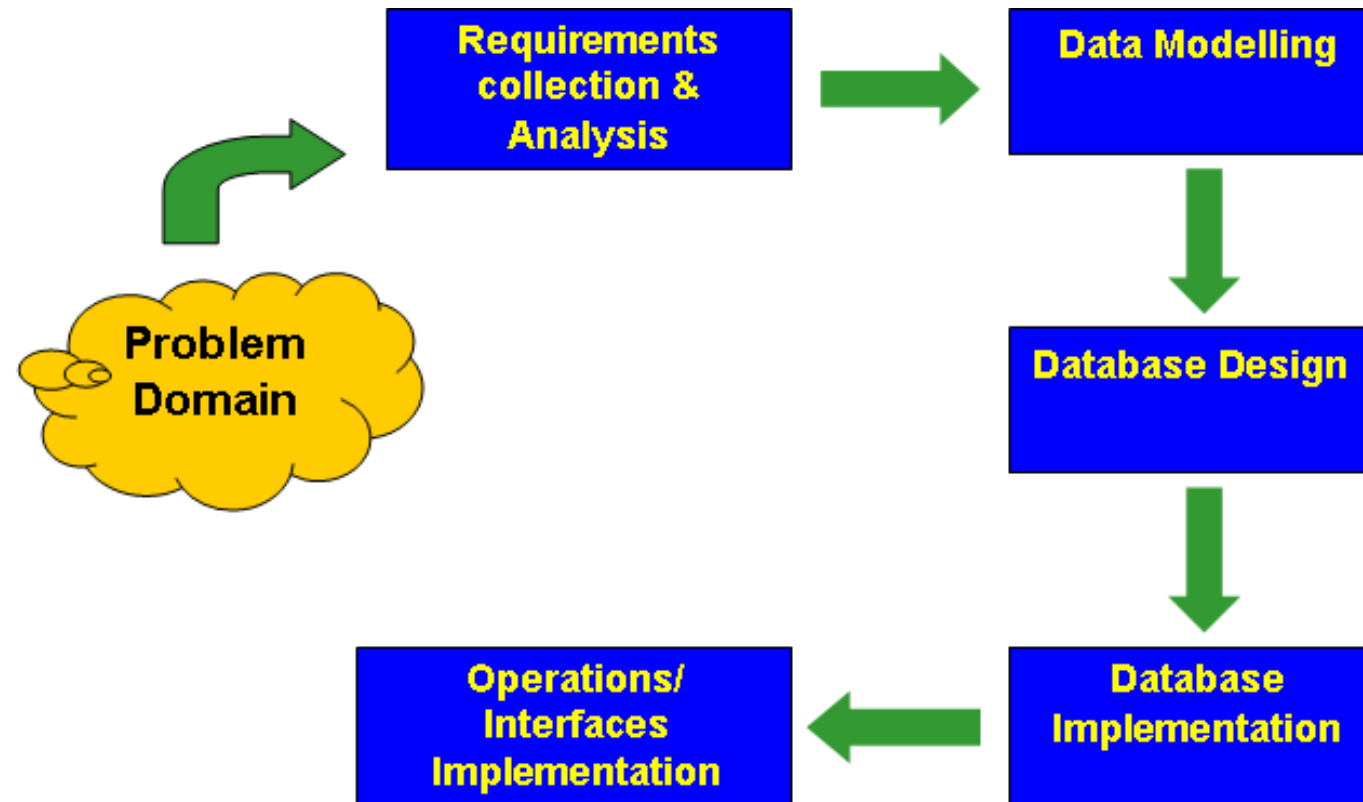
This stage includes adding of new data, modifying existing data and deletion of absolute data.

This phase provides useful information and helps management to make a business decision.

Maintenance

It is one of the ongoing phases in DDLC.

Phases of the database application development process



Data Model using Entity Relationship Model

- ✓ Data modeling is a technique to document a software system using diagrams and symbols.
- ✓ It is used to represent communication of data.
- ✓ The highest level of abstraction for the data model is called the Entity Relationship Diagram (ERD).
- ✓ It is a graphical representation of data requirements for a database.

Entity Relationship Diagram

The main value of carefully constructing an ERD is that it can readily be converted into a database structure.

There are three components in ERD.

Entities: Number of tables you need for your database.

Attributes: Information such as property, facts you need to describe each table.

Relationships: How tables are linked together.

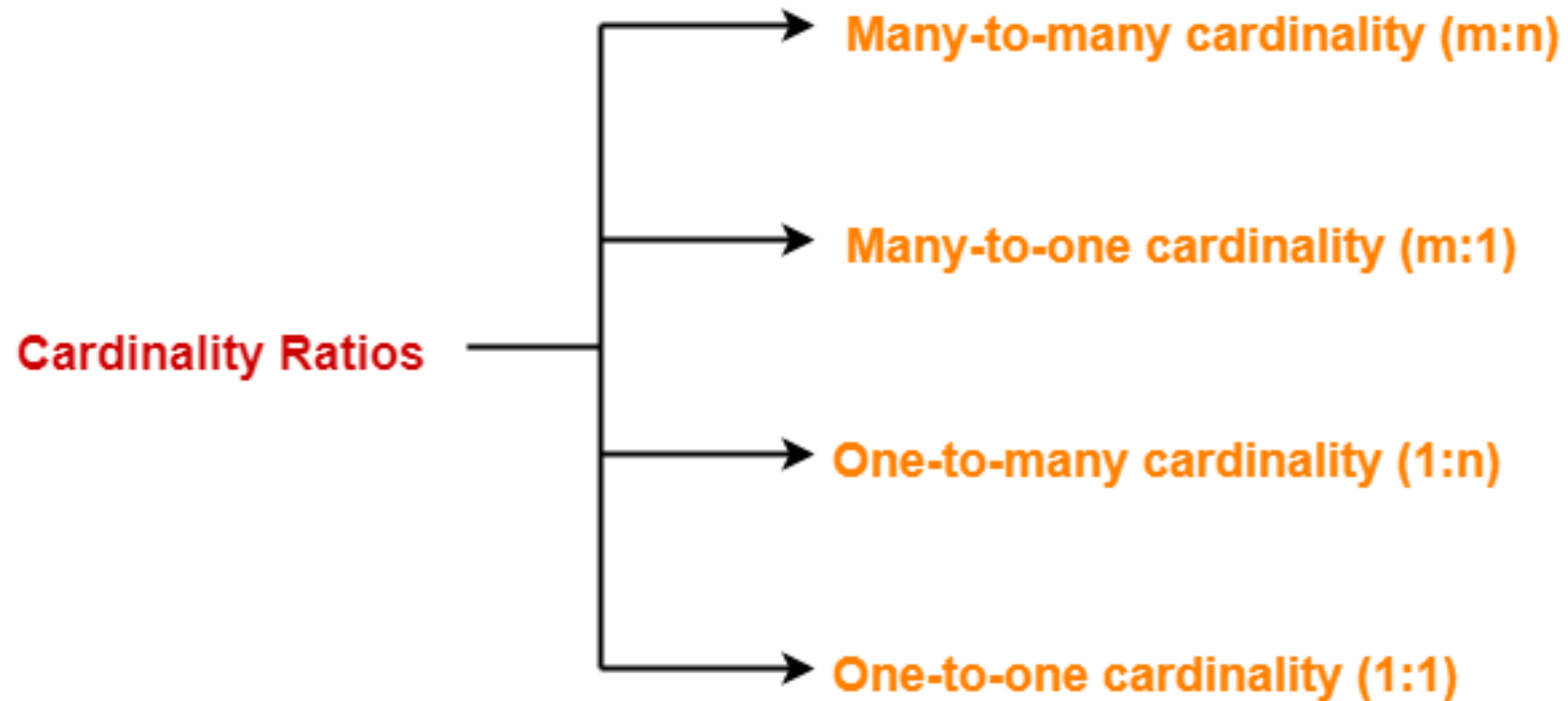
Entity

- Entities are the basic objects of ERDs.
- These are the tables of your database.
- Entity are nouns and the types usually fall into five classes: concepts, locations, roles, events or things.
- ✓ For example: students, courses, books, campus, employees, payment, projects.
- ✓ A specific example of an entity is called an instance. Each instance becomes a record or a row in a table. For example: the student John Smith is a record in a table called students.

Relationships

- Relationships are the associations between the entities.
- Verbs often describe relationships between entities.
- We will use Crow's Foot Symbols to represent the relationships.
- 4 types of relationships are discussed here.
- If you read or hear cardinality ratios, it also refers to types of relationships.

Cardinality Ratios



In this example,
One student can enroll in any number (zero or more) of courses.
One course can be enrolled by any number (zero or more) of students.



Many to Many Relationship

Cardinality Ratios

One to One Relationship (1:1)

A single entity instance in one entity class is related to a single entity instance in another entity class.

✓ For example:

Each student fills one seat and one seat is assigned to only one student.

Each professor has one office space.

Cardinality Ratios

One to Many Relationship (1:M)

A single entity instance in one entity class (parent) is related to multiple entity instances in another entity class (child)

✓For example:

One instructor can teach many courses, but one course can only be taught by one instructor.

One instructor may teach many students in one class, but all the students have one instructor for that class.

Cardinality Ratios

Many to Many Relationship (M:M)

Each entity instance in one entity class is related to multiple entity instances in another entity class; and vice versa.

For example:

Each student can take many classes, and each class can be taken by many students.

Each consumer can buy many products, and each product can be bought by many consumers.

The detailed Crow's Foot Relationship symbols can be found here. [Crow's Foot Relationship Symbols](#)

Attributes

- Attributes are facts or description of entities.
- They are also often nouns and become the columns of the table.
- ✓ For example, for entity student, the attributes can be first name, last name, email, address and phone numbers.

Attributes

✓ Primary Key

Primary Key* or identifier is an attribute or a set of attributes that uniquely identifies an instance of the entity.

For example, for a student entity, student number is the primary key since no two students have the same student number. We can have only one primary key in a table. It identify uniquely every row and it cannot be null.

Attributes

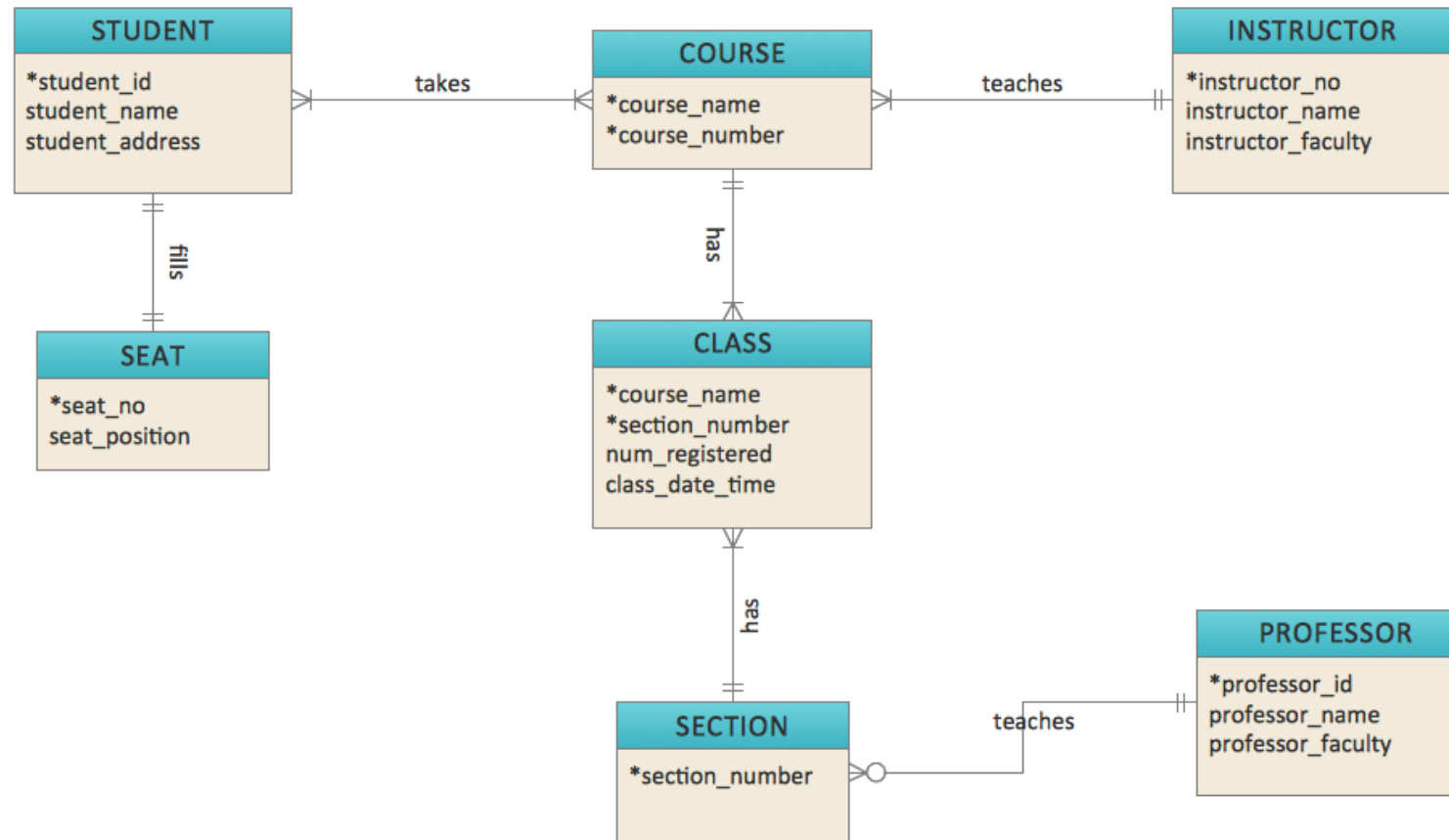
✓ Foreign key

A foreign key+ (sometimes called a referencing key) is a key used to link two tables together.

Typically you take the primary key field from one table and insert it into the other table where it becomes a foreign key (it remains a primary key in the original table).

We can have more than one foreign key in a table.

Example,



Thanks

