RDBMS Concepts

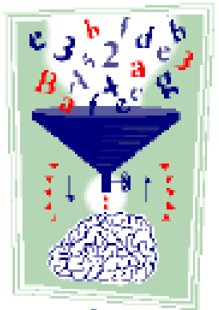
Session 1

RDBMS Concepts

Session Objectives

- Understand the importance of data and database
- List data models
- Define DBMS
- RDBMS
- ER Diagram
- Normalization/De-Normalization

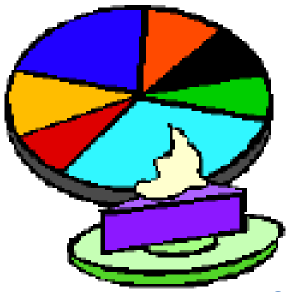
The Role of Data

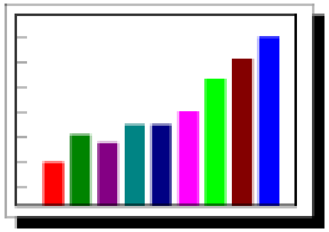

To Identify

To Explain

To Plan

To Administer


To Monitor


To Analyze

To Implement

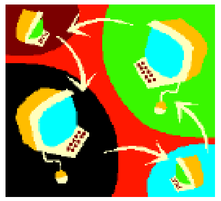
To Decide

Data Models

- Flat-file model

- Hierarchical model

- Network model



- Relational model

Hierarchical model

- A hierarchical data model is a data model in which the data is organized into a tree-like structure

- A hierarchical data model is a data model in which the data is organized into a tree-like structure.

- The structure allows repeating information using parent/child relationships:
  - Each parent can have many children but each child only has one parent.
  - All attributes of a specific record are listed under an entity type.
- In a database, an entity type is the equivalent of a table:
  - Each individual record is represented as a row
  - And an attribute as a column.
- Entity types are related to each other using 1: N mapping, also known as one-to-many relationships.
- The most recognized example of hierarchical model database is IMS designed by IBM.

Network Model

The network model is a database model conceived as a flexible way of representing objects and their relationships.

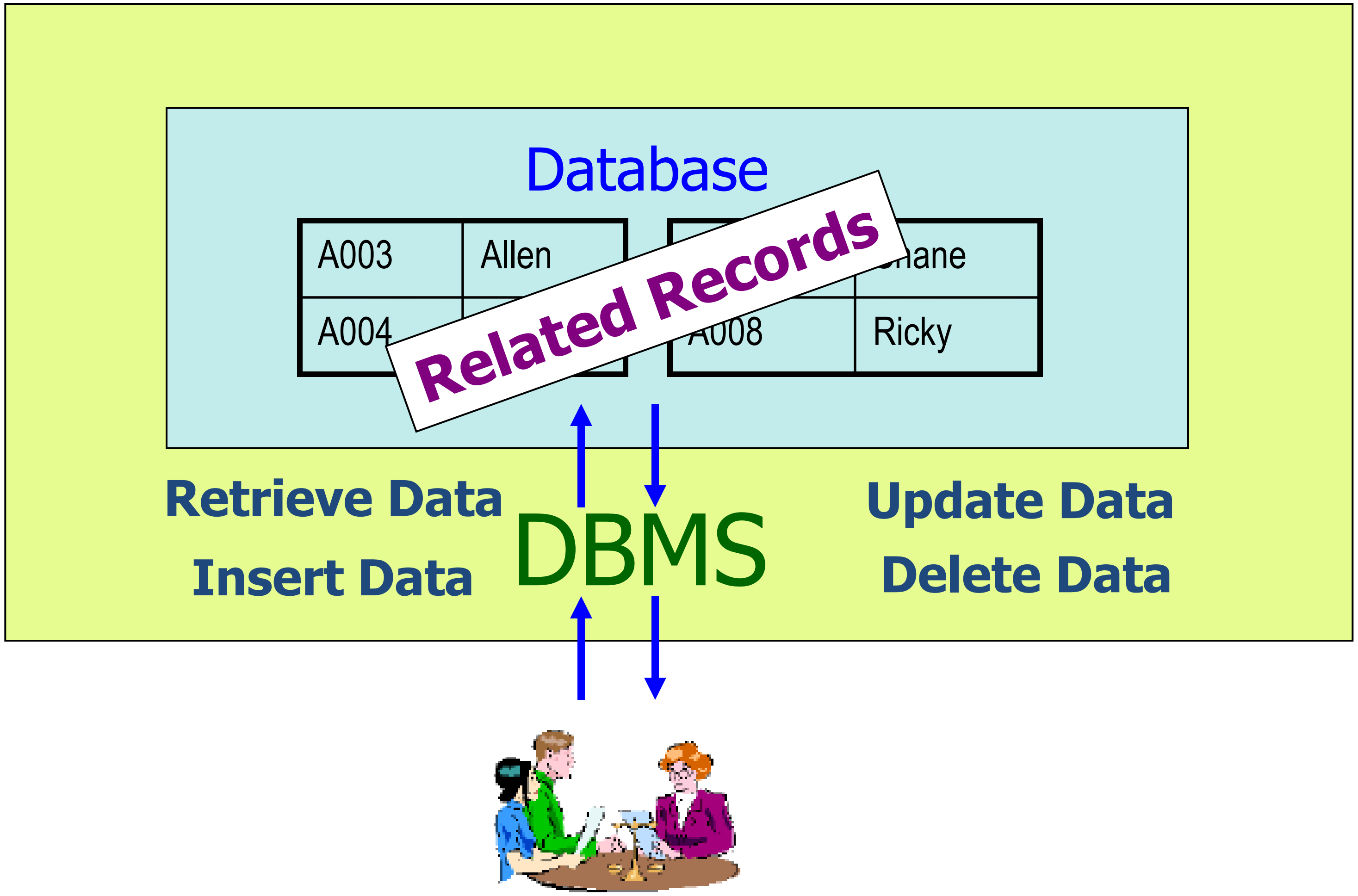
The network model's original inventor was Charles Bachman, and it was developed into a standard specification published in 1969 by the Conference on Data Systems Languages (CODASYL) Consortium.

Where the hierarchical model structures data as a tree of records, with each record having one parent record and many children, the network model allows each record to have multiple parent and child records, forming a lattice structure.

Some Well-known Network Databases

- Turbo IMAGE
- IDMS(Integrated Database Management System)
- RDM Embedded
- RDM Server

What is a DBMS?



Benefits of DBMS

- The amount of **redundancy** in stored data can be **reduced**
- **No more inconsistencies** in data
- Stored **data** can be **shared**
- **Standards** can be set and followed
- **Data integrity** can be maintained
- **Data Security** can be implemented

Relational Model

The purpose of the relational model is to provide a declarative method for specifying data and queries:

we directly state

what information the database contains and

what information we want from it,

and let the **database management system software** take care of describing data structures for storing the data and retrieval procedures for getting queries answered.

IBM's original implementation of Codd's ideas was System R.

There have been several commercial and open source products based on Codd's ideas, including

IBM's DB2,

Oracle Database,

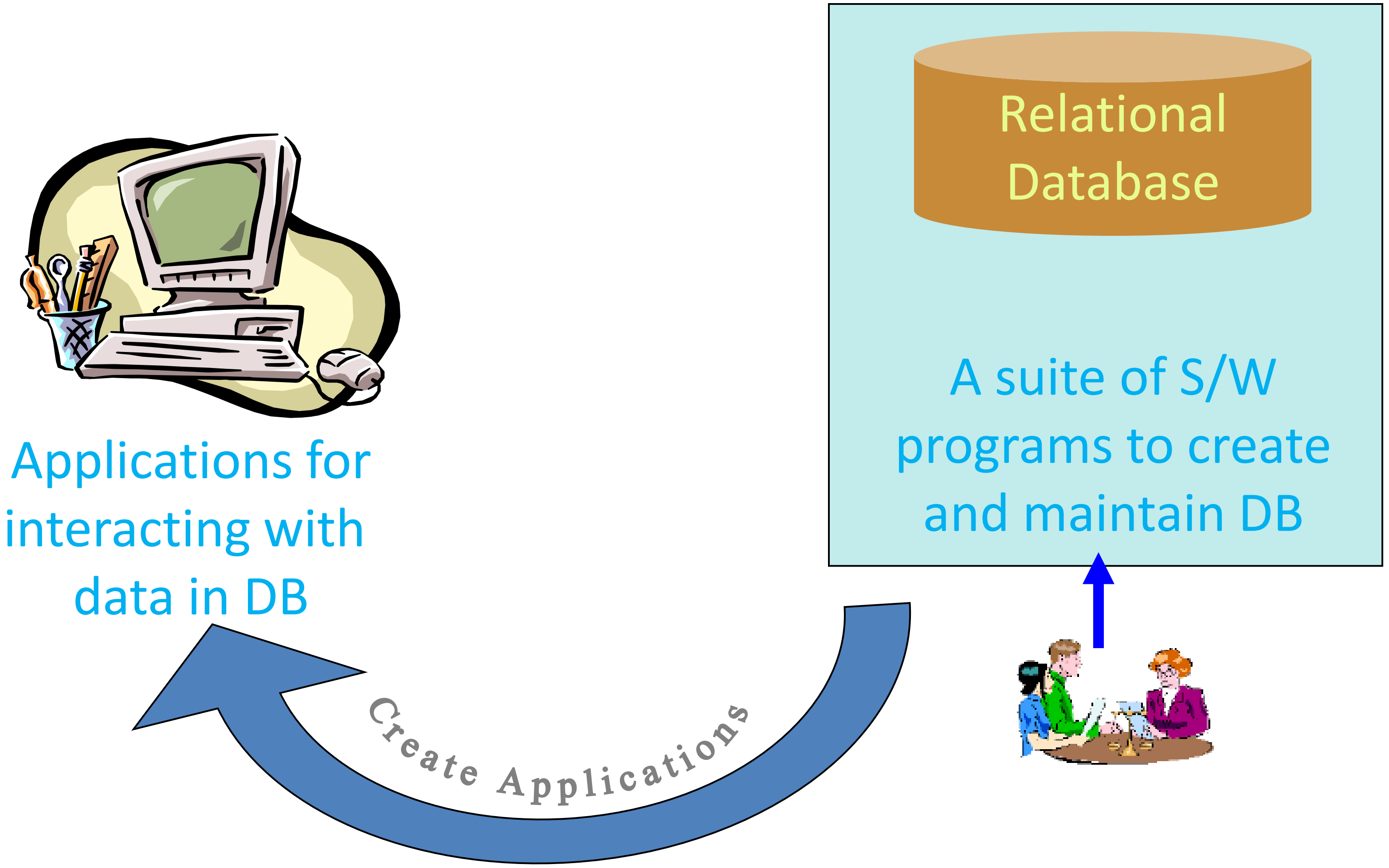
Microsoft SQL Server,

PostgreSQL,

MySQL, and many others.

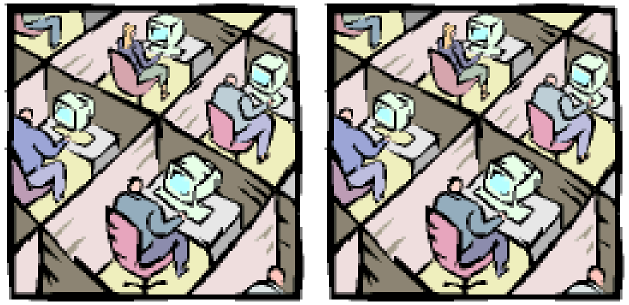
Most of these use the SQL data definition and query language .

RDBMS


Relational Database
A suite of S/W programs to create and maintain DB
Applications for interacting with data in DB
Create Applications

Users of RDBMS

Database Administrators


End Users

Application Programmers

Relational model and application to databases

- A **type** as used in a typical relational database might be the set of integers, the set of character strings, the set of dates, or the two Boolean values *true* and *false*, and so on.
- The corresponding **type names** for these types might be the strings "int", "char", "date", "Boolean", etc. It is important to understand, though, that relational theory does not dictate what types are to be supported;
- nowadays provisions are expected to be available for *user-defined* types in addition to the *built-in* ones provided by the system.
- **Attribute** is the term used in the theory for what is commonly referred to as a **column**
- **table** is commonly used in place of the theoretical term **relation**
- A **tuple** is basically the same thing as a **row**

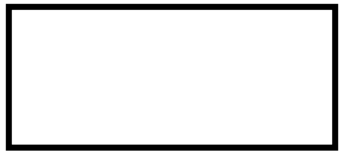
Entity-Relationship Data Model

Elements of E-R Model

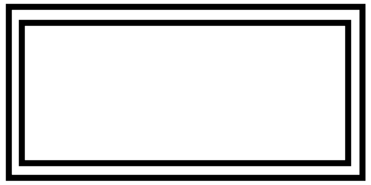
In the E/R model, the structure of data is represented graphically, as an “entity-relationship diagram,” using three Principal element types:

- Entity Sets.
  - An entity is an abstract object of some sort, and a collection of similar entities forms an entity set.
- Attributes
  - The properties of the entities in the set.
- Relationship
  - The connections among two or more entity Sets

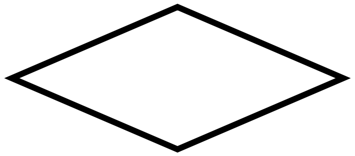
Notation Guide



- ENTITY TYPE

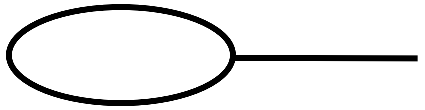


- WEAK ENTITY TYPE

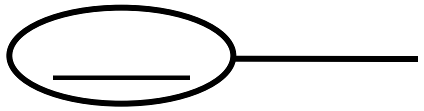


- RELATIONSHIP TYPE

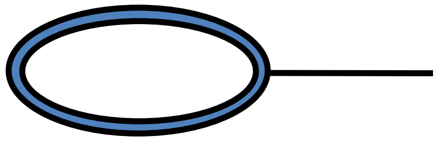
Notation Guide



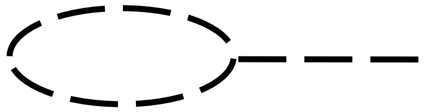
- ATTRIBUTE



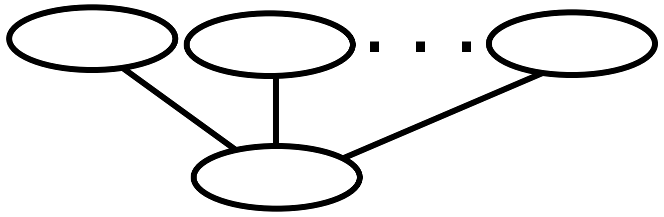
- KEY ATTRIBUTE



- MULTIVALUED ATTRIBUTE



- DERIVED ATTRIBUTE

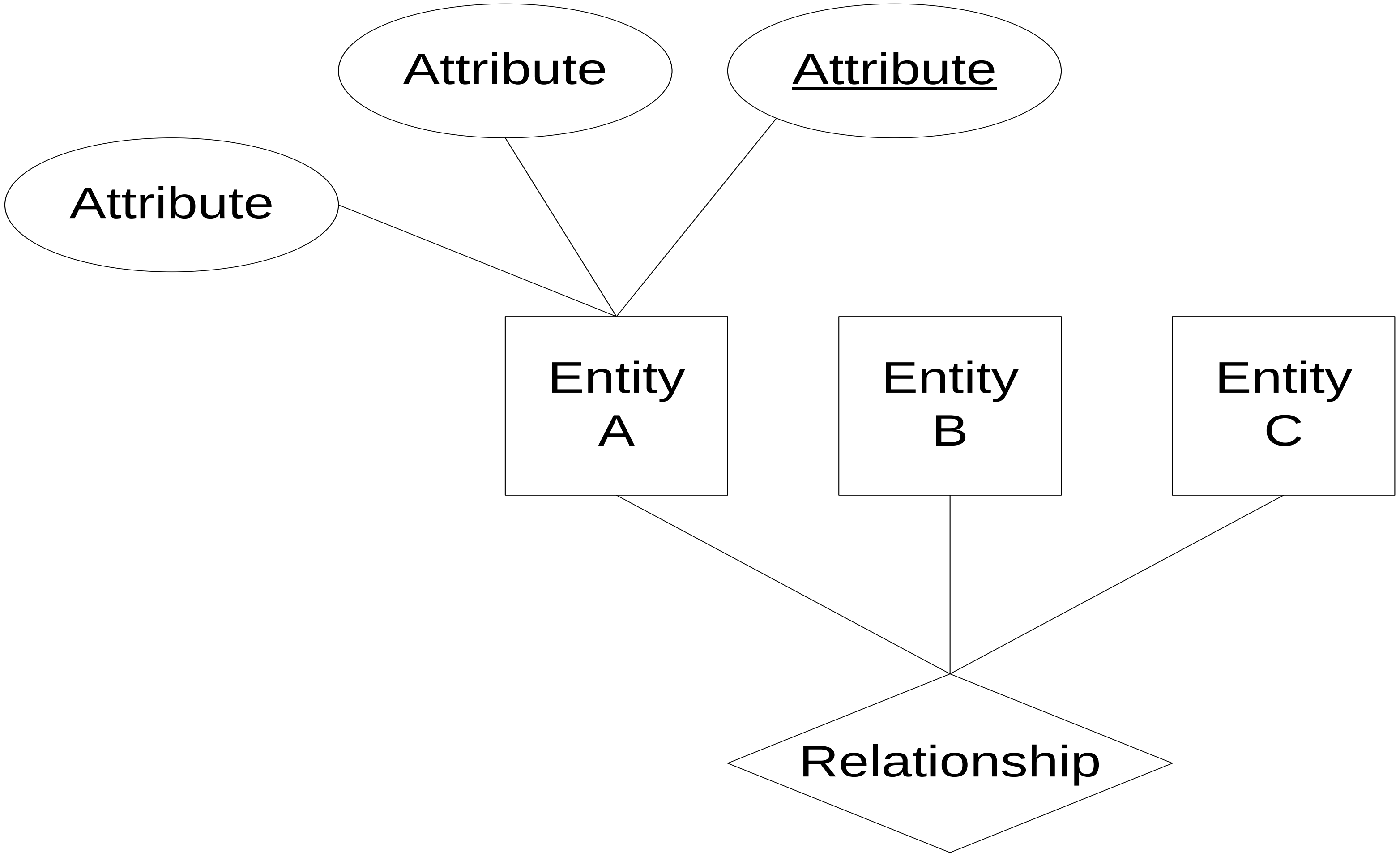


- COMPOSITE ATTRIBUTE

Example of Elements of E-R Model

- Entity Sets
  - Departments
  - Professors
  - Students
  - Administrators
- Attributes
  - Name of Departments, Phone No., Address...
  - Name, SSN, Address of Professors...
- Relationship
  - Students and Professors are under a certain department
  - Admin manage the campus/ departments

Example of the 3 elements in E/R Diagram



Classification of Constraints

1. Keys
2. Single-value constraints
3. Multi-valued constraints
4. Mapping Cardinalities and Participation Constraints

Key in the E/R Model

- Super key
- Candidate key
- Unique key
- Primary key

Keys

Superkey

A superkey is an attribute or set of attributes that uniquely identifies rows within a table; in other words, two distinct rows are always guaranteed to have distinct superkeys.

{Employee ID, Employee Address, Skill} would be a superkey for the "Employees' Skills" table; {Employee ID, Skill} would also be a superkey.

Keys

Candidate key

A candidate key is a minimal superkey, that is, a superkey for which we can say that no proper subset of it is also a superkey.

{Employee Id, Skill} would be a candidate key for the "Employees' Skills" table.

Keys

- Unique key
  - A **unique key** is a candidate key to uniquely identify each row in a table
  - The values in a unique key columns may or may not be NULL

Keys

Primary key

Most DBMSs require a table to be defined as having a single unique key, rather than a number of possible unique keys.

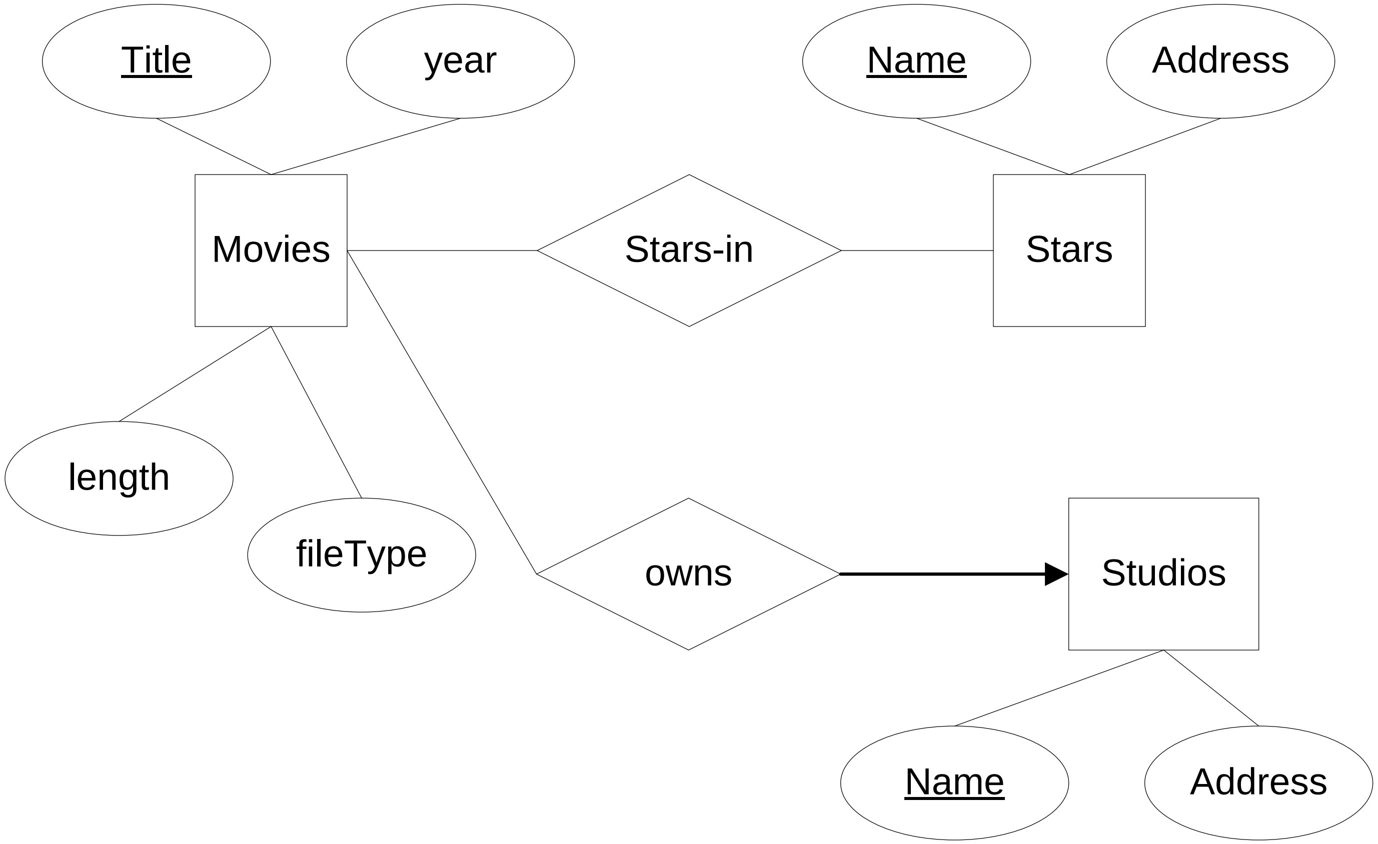
A primary key is a key which the database designer has designated for this purpose.

The values in a Primary key columns Must not be NULL.

An Index is automatically created for Primary Key.

Single/Multi-valued attributes

- Single-valued attributes are attributes that only have a single value for a particular entity.
- Multi-valued attributes refers to items that are not singled-value and Null valued. For example, consider an employee entity set with the attribute phone-number. An employee may have zero, one, or several phone numbers; different employee may have different numbers of phones.


Title
year
Name
Address
Movies
Stars-in
Stars
length
fileType
owns
Studios
Name
Address

Mapping Cardinalities or Cardinality ratios

- Express the number of items to which another item can be associated via a relationship set
- Are most useful in describing binary relationship sets. For a binary relationship set R between entity sets A and B, the mapping cardinality must be one of the following:
  - One to One
  - One to Many
  - Many to One
  - Many to Many

Participation Constraints

The participation of an entity set E in a relationship set R is said to be total, if every item in E participates in at least one relationship in R. If only some items in E participate in relationship R, the participation of entity set E in relationship R is said to be partial.

Normalization

Normalization is a systematic way of ensuring that

- a database structure is suitable for general-purpose querying
- Process of removing redundancies from incoming data
- free of certain undesirable characteristics—insertion, update, and deletion anomalies—that could lead to a loss of data

Insert anomalies

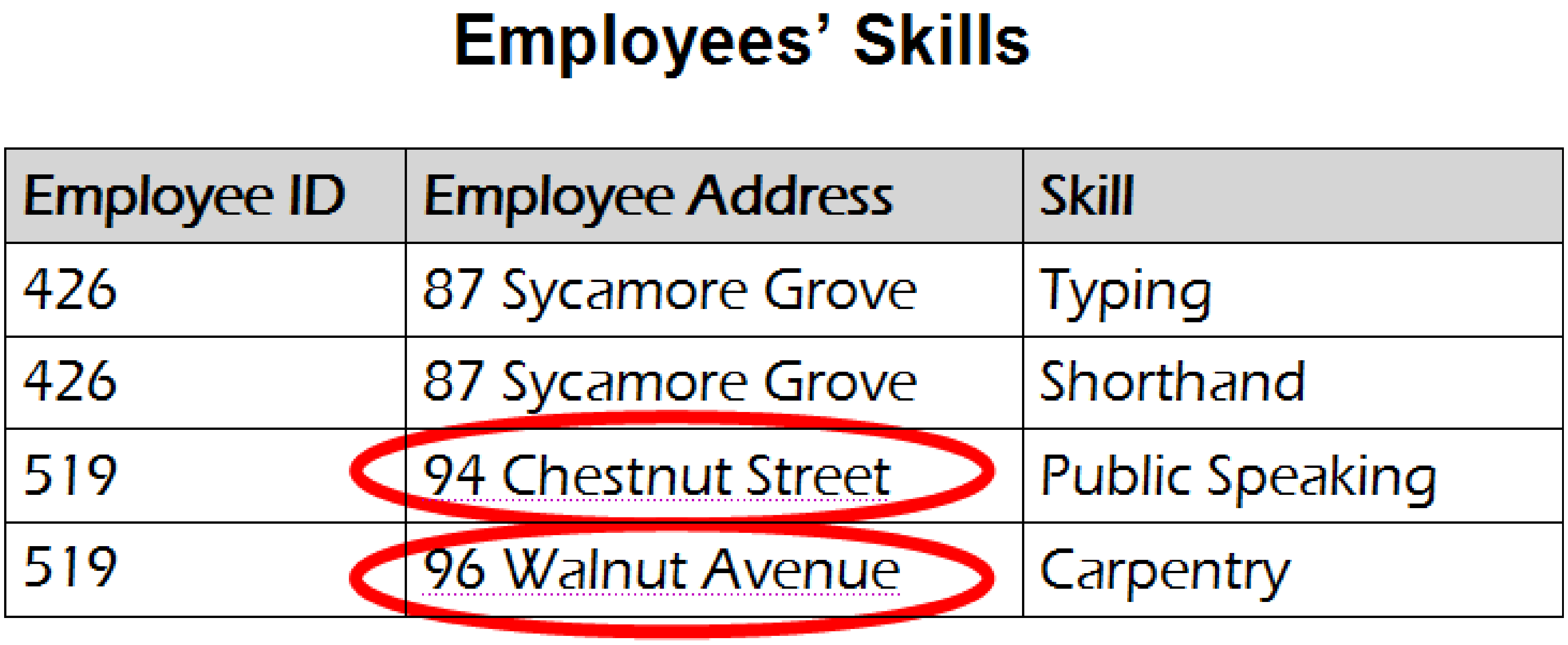
Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201

424	Dr. Newsome	29-Mar-2007	?

An **insertion anomaly**. Until the new faculty member, Dr. Newsome, is assigned to teach at least one course, his details cannot be recorded.

Update anomalies

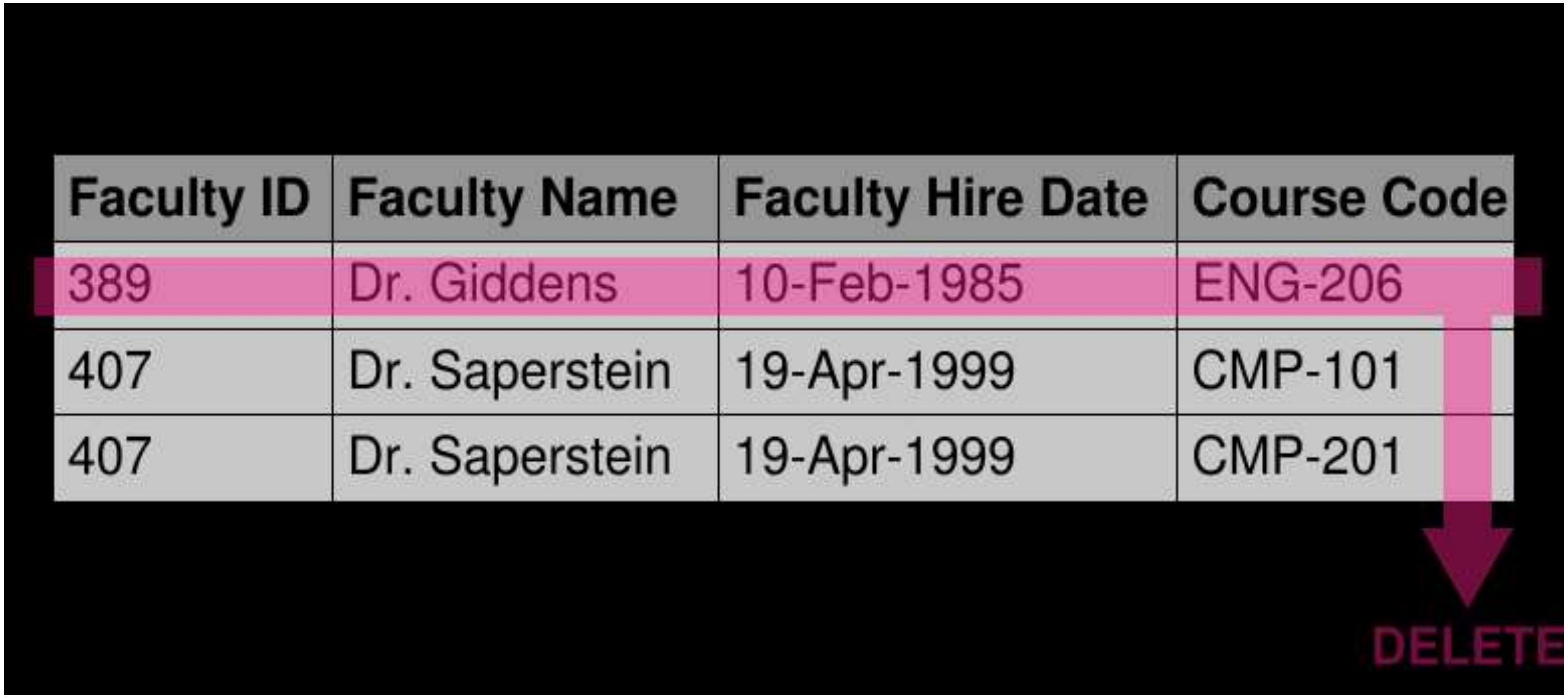


Employees' Skills

Employee ID	Employee Address	Skill
426	87 Sycamore Grove	Typing
426	87 Sycamore Grove	Shorthand
519	94 Chestnut Street	Public Speaking
519	96 Walnut Avenue	Carpentry

An **update anomaly**. Employee 519 is shown as having different addresses on different records

deletion anomaly



Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201

A **deletion anomaly**. All information about Dr. Giddens is lost when he temporarily ceases to be assigned to any courses.

Normalization: Some definitions

Functional dependency

Attribute B has a functional dependency on attribute A (i.e., $A \longrightarrow B$) if, for each value of attribute A, there is exactly one value of attribute B

Full functional dependency

An attribute is fully functionally dependent on a set of attributes X if it is

functionally dependent on X, and

not functionally dependent on any proper subset of X.

{Employee Address} has a functional dependency on {Employee ID, Skill}, but not a *full* functional dependency, because it is also dependent on {Employee ID}.

Transitive dependency

A transitive dependency is an indirect functional dependency, one in which $X \rightarrow Z$ only by virtue of $X \rightarrow Y$ and $Y \rightarrow Z$.

Normalization

To begin the normalization process, we start by moving from zero normal form to 1st normal form.

The definition of 1st normal form

- there are no repeating groups
- all the key attributes are defined
- all attributes are dependent on the primary key

Normalization

The definition of 2nd normal form

A table is in 2nd normal form if

- it's in 1st normal form
- it includes no partial dependencies

(where an attribute is dependent on only a part of a primary key).

Normalization

The definition of 3rd normal form

A table is in 3rd normal form if

- It's in 2nd normal form
- It contains no transitive dependencies (where a non-key attribute is dependent on another non-key attribute).

Characteristics of Normalized databases

- Each table has a key field
- There are no repeating fields
- Each table contains information about a single entity
- Each field in a table depends on the key field
- All non-key fields are mutually independent

When to use Normalization?

- Data is large and scattered
- No defined group of data
- Data is too complicated

Drawbacks of Normalization

- Complex join queries to access data in multiple tables
- Poor performance of the database

Solution

Denormalize

Thank you