

Course code : **CSE2007**  
Course title : **Database Management System**  
Module : **2**  
Topic : **2**

# Relational Database Design

# Objectives

This session will give the knowledge about

- Relational Database design
- Using ER-to-Relational Mapping
- Referential integrity constraints

# Introduction to Relational Database

The term Relational database was defined by **Edgar Codd at IBM in 1970**.

Relational database is **a collection of tables**.

In relational database, data is represented in simple two dimensional tables (**relations**) which consists of rows (**tuples**) and columns (**attributes**).

# Relational database concepts

## Relations or Tables

- A relation is defined as a set of tuples that have the same attributes. A tuple usually represents an object and information about that object.

## Base and Derived relations

- Relations that store the data are called “base relations” and in implementations are called “tables”
- Relations that do not store the data, but are computed by applying relational operations to other relations are called “derived relations” and in implementation are called “views” or “reports”.

# Relational database concepts

## Tuple / Record / Row

- A tuple / record / row holds all the information about one item or object.

## Field / Column

- A field / columns holds one piece of information about an item or object.

## Domain

- A domain describes the set of possible values for a given attribute.

# Relational database concepts

## Constraints

- A constraints allows you to restrict the domain of an attribute.
- It restricts the data that can be stored in relations
- Every attribute has an associated domain which is know as domain constraints

## Data type

- Every field in a database table is assigned a data type, which describes the kind of data that can be stored in that field.

## Primary Key

- A primary key is a field (or possible multiple fields used together) that uniquely identifies each record in the table.

# Relational database concepts

## Foreign Key

- A Foreign Key is a **key used to link two tables together**.
- A Foreign Key is a field (or collection of fields) in **one table that refers to the primary key in another table**.

## Stored Procedure

- A stored procedure is a high end tool that adds programming capability into the database.
- A stored procedure is **executable group of queries that is associated with, and generally stored in the database**.

# Relational database concepts

## Indices

- An index is one way of providing quicker access to data.
- Indices can be created on any combination of attributes on a relation.

## Relational Operations

- Queries made against the relational database, and the derived relations in the database are expressed in a relational calculus or relational algebra.
- There are eight relational operations
- First four are relational operations based on mathematical set operations and the last four involve special operation specific to RDBMS



# Relational operations

## Union

- Combines the tuples of two relations and removes all duplicate tuples from the result.

## Intersection

- Produces the set of tuples that two relations share in common.

## Difference

- Acts on two relations and produces the set of tuples from first relation that do not exist in the second relation.

# Relational operations

## Cartesian Product

- Cartesian product of two relations is a join that is not restricted by any criteria.
- It produces every tuple of the first relation being matched with every tuple of the second relation.

## Selection

- Retrieves all the tuples from a relation.

## Projection

- Retrieves the tuples without duplicate tuples from a relation.

# Relational operations

## Join

- Two relations are connected by their **common attributes**.

## Relational division

- It is the **direct opposite of the Cartesian product operator**.
- It involves essentially using the tuples of one relation (the dividend) to partition a second relation (the divisor).

## Normalization

- It is **used to eliminate the duplication of data**, which in turn prevents data manipulation anomalies and loss of data integrity.

# Characteristics of Relational Model

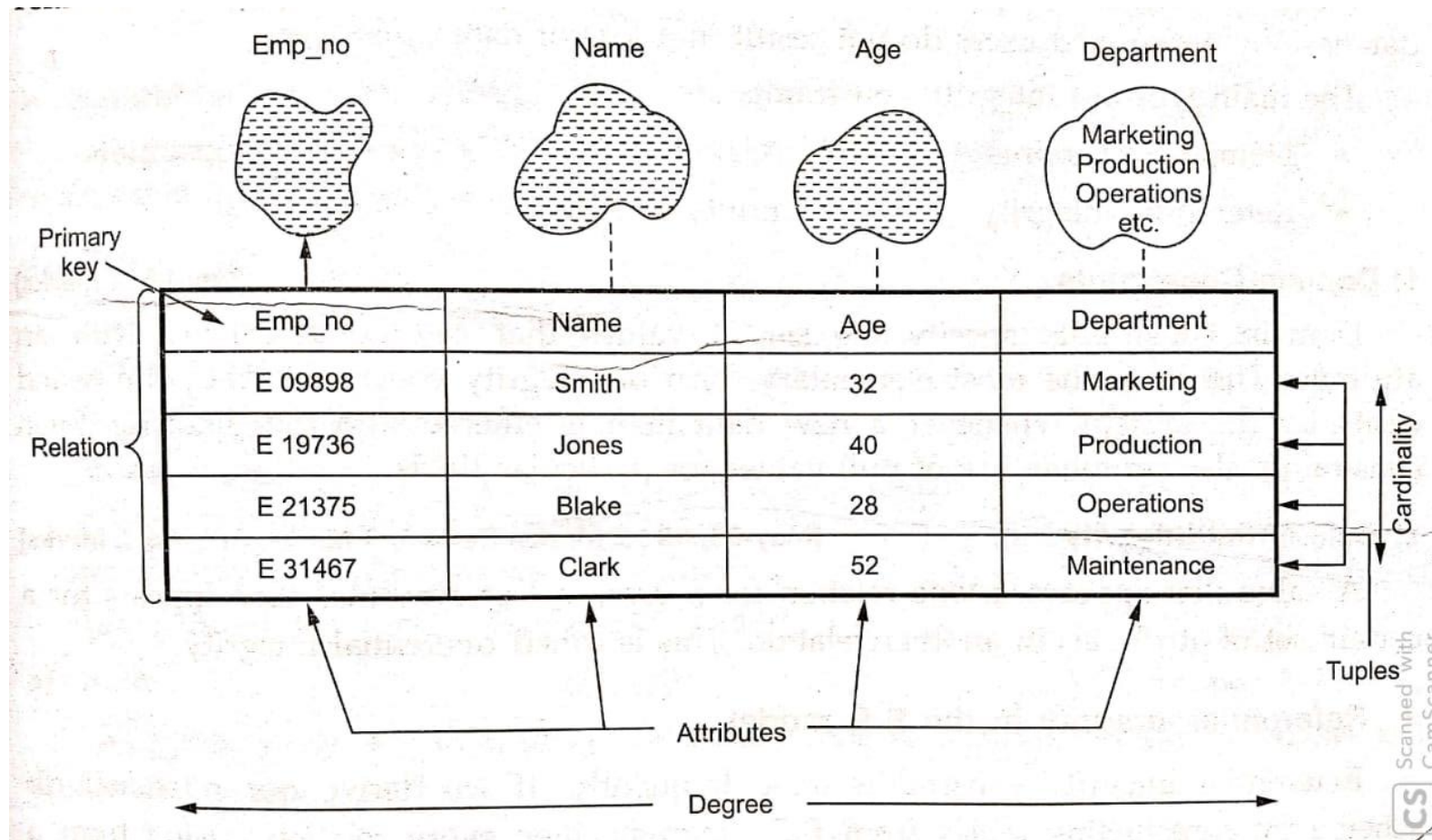
- It eliminates all parent and child relationships and represented the data as a table.
- Relational model table consists of rows and columns
- Each table is an individual entity and there is no physical relationship between tables
- All relational databases supports query languages like SQL
- Relational model of data is based on set theory and the user interface with the relational models is non-procedural.

# Components of Relational Model

The relational model consists of the following three basic components

- Data structure
- Data integrity
- Data manipulation

# Data Structure



# Data Integrity

Integrity constraints provide a means of ensuring that changes made to the database by authorized for ensuring data consistency.

The types of integrity constraints are

- Domain constraints
- Referential integrity
- NULLs
- Entity integrity
- Enterprise constraints

## Domain constraints

- Domain constraints specifies the set of values that can be associated with an attribute.
- These are the most elementary form of the integrity constraint.
- They are easily tested by the system whenever a new data item is entered into the database.
- Constrains also prohibits the use of null values for particular fields.



# Referential integrity

Referential integrity can be defined as a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

**Primary Table**

CompanyId	CompanyName
1	Apple
2	Samsung

**Related Table**

CompanyId	ProductId	ProductName
1	1	iPhone
15	2	Mustang

Associated Record ✓

Orphaned Record ✗

## Referential integrity in SQL

```
CREATE TABLE deposit (  
    branch_name varchar2(20),  
    acc_no number(20) not null,  
    cust_name varchar2(20),  
    balance number(5),  
    PRIMARY KEY (acc_no),  
    FOREIGN KEY (branch_name) REFERENCES branch  
    FOREIGN KEY (cust_name) REFERENCES customer);
```

# NULLs, Entity and Enterprise Constrains

## NULL Constrains:

NULLs represents a value for an attribute that is currently unknown or is not applicable for this tuple. NULLs are a way to deal with incomplete or exceptional data.

## Entity Constraints:

In a base relation no attributes of a primary key can be null.

## Enterprise constrain:

The additional rules or constrains specified by the database administrators are known as enterprise constrains. Example: no of students in a class should be 65.

# Summary

This session will give the knowledge about

- Relational Database design
- Using ER-to-Relational Mapping
- Referential integrity constraints