Tutorialwing Courses Contact Us Articles

Sign in to Teal

Database

01 - Basic

- Intro
- File vs Database
- Data Abstraction
- Data Independence

02 - ER Model

- ER Model Concepts
- Mapping Constraints
- ER Diagram

03 - Relational Model

- Relational Data Model
- Relational Algebra
- Additional Algebra
- Anomalies in dbms
- RDBMS Introduction
- Keys in RDBMS
- Normalization in dbms
- Redundancy in DBMS
- Functional Dependency
- Types of Decomposition

04 - SQL

- Data Types in SQL
- Data Definition Language in SQL
- Data Manipulation

ER Diagram in DBMS – Componen and Notations

Like 2 Tweet

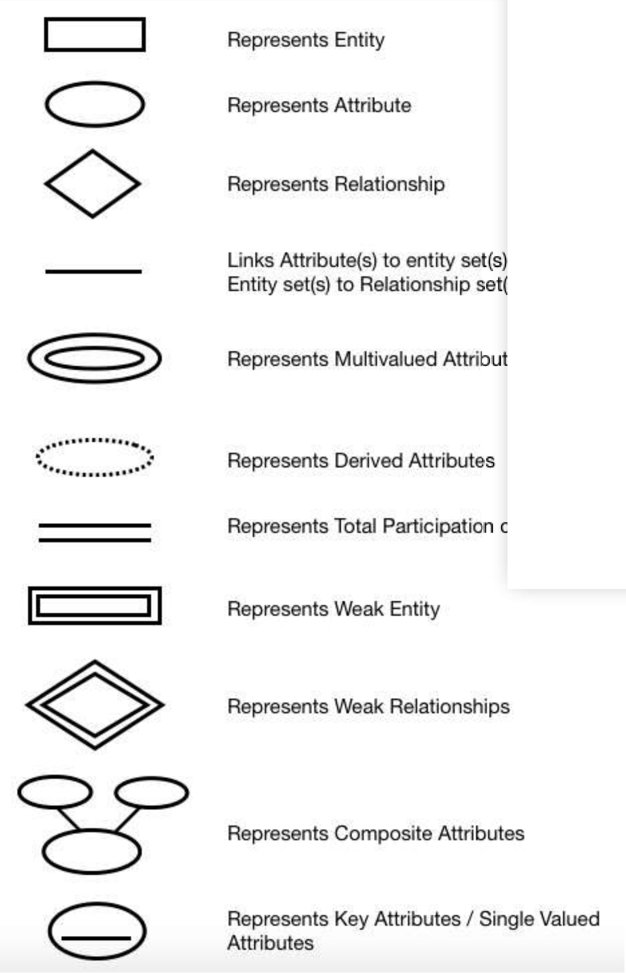
<< Prev Post

In this post, we will learn about different component, symbols ar Diagram in DBMS. We will also see some extended features on Specialization, Generalization and Aggregation.

We have already seen some basic concepts about ER model in

Different notations used for ER Diagra

Below picture clearly depicts about different notations we need DBMS. They are –


Represents Entity
Represents Attribute
Represents Relationship
Links Attribute(s) to entity set(s)
Entity set(s) to Relationship set(
Represents Multivalued Attribut
Represents Derived Attributes
Represents Total Participation c
Represents Weak Entity
Represents Weak Relationships
Represents Composite Attributes
Represents Key Attributes / Single Valued Attributes

Components of ER Diagram in dbms

As above image describes,

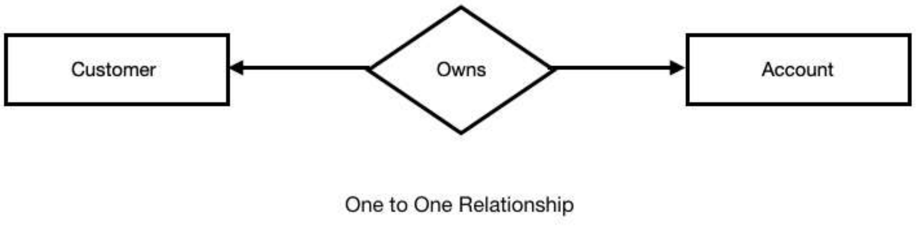
- **Rectangle** – It represents entity in the ER Model.
- **Ellipse** – It represents attribute in the ER Model.
- **Diamond** – It represents relationship between entity and attribute.
- **Line** – It links attribute(s) to entity set(s) and entity set(s) to relationship set(s).

- **Double lines** – It indicates total participation of an entity in a relationship set.
- **Double Rectangle** – It represents weak entity set.
- **Double Diamonds** – It represents weak relationships.
- **Multiple ellipses connected to single ellipse using lines** – It represents composite attribute
- **Ellipse with line inside it** – It represents single values attributes

Connectivity of a Relationship –

It means how many instances of one entity type are linked with how many entity of another entity set. Different types of connectivity of a relationship are –

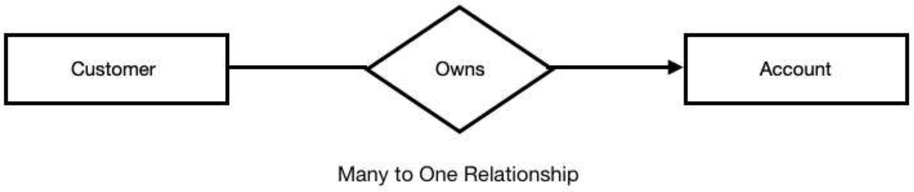
(A) **one-to-one relationship** – If the relationship is one-to-one i.e. each customer has one account and each account belongs to one customer, then this is represented as follows. –



One to One Relationship

Type – 1

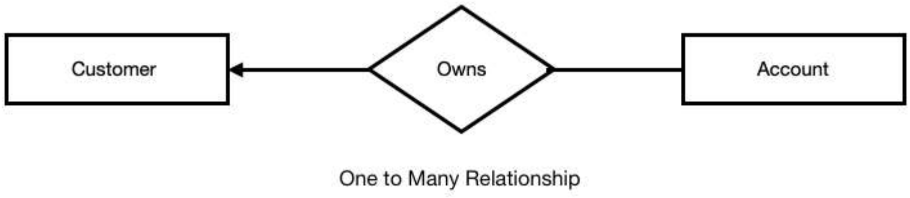
(B) **many-to-one relationship** – If this relationship is many-to-one, i.e. many customers can share one account then this is represented as follows.



Many to One Relationship

Type – 2

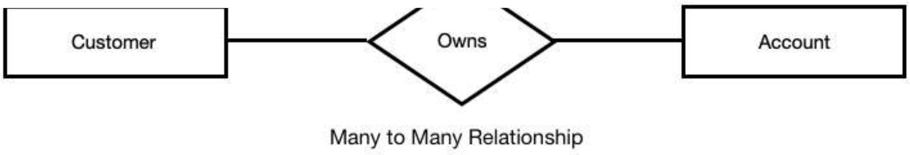
(C) **one-to-many relationship** – If the relationship is one-to-many, i.e. one customer may have any number of accounts, but one account belongs to only one customer then this cardinality is represented as –



One to Many Relationship

Type – 3

(D) **many-to-many relationship** – If relationship is many-to-many, i.e. one customer may have any number of accounts and one account can be shared by any number of customers, then this is represented as follows.



Type – 4

2. Extended ER Features

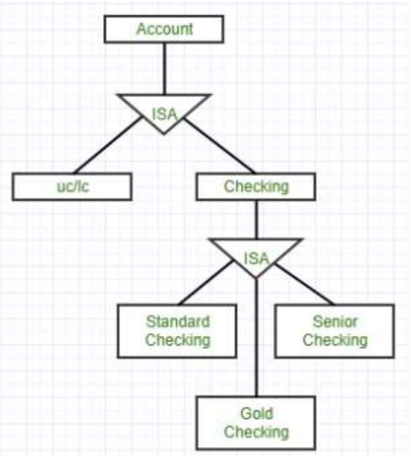
Now, we will study about some of the extended features of ER Diagrams – Specialization, Generalization and Aggregation.

2.1 Specialization

It is the process of designating sub grouping within an entity set.
For example: Consider an account entity which can be classified into
a. Saving
b. Checking

Checking can be further classified into
a. Standard checking
b. Gold checking
c. Senior checking

In E-R diagram, specialization is depicted by a triangle component labeled ISA.



Specialisation in DBMS

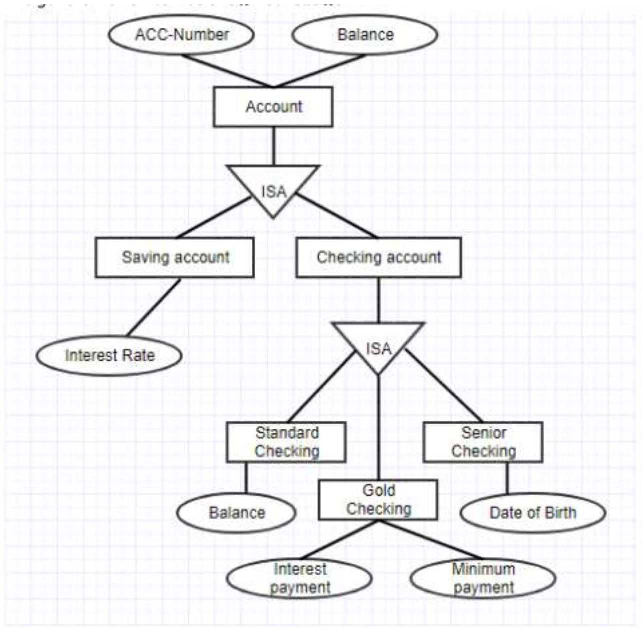
Account is higher-level entity and saving and checking are lower level entities and lower to this level are standard, senior and gold checking.

Generalization

It is a containment relationship that exists between a higher level entity set and one or lower level entity sets.
Attributes of higher-level entity set and of lower level entity set are common. This commonality is expressed by generalization.
For example: Attributes of account entity set are {acc-no, bal} Attributes of saving account are {acc-no, bal, interest rates} Attributes of checking account are {acc-no, balance, overdraft amount}
Attributes of standard checking account are {Acc-no, bal, overdraft amount, number-check}
Attributes of gold checking account are j8 {Acc-no, bal, overdraft amount, inter-pay, min-

date of birth}

The generalization can be shown as follows



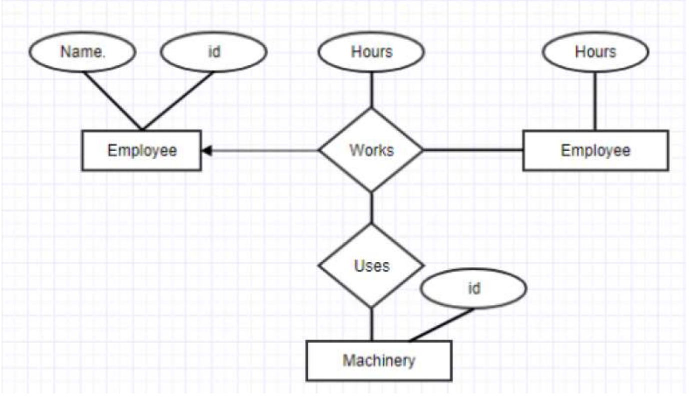
Generalization in dbms

Generalization is a simple inversion of specialization. Specialization emphasizes difference among several entities within the set by creating distinct lower level entity set. Generalization is used to emphasize the similarities among lower level entity sets and to hide the differences.

Aggregation

Aggregation is an abstraction through which relationship are treated as higher-level entities.
It helps to express relationship among relationships. E.g. consider the database about employees who work on a particular project and use a number of different machines in their work.

E-R diagram to represent the database without using aggregation is



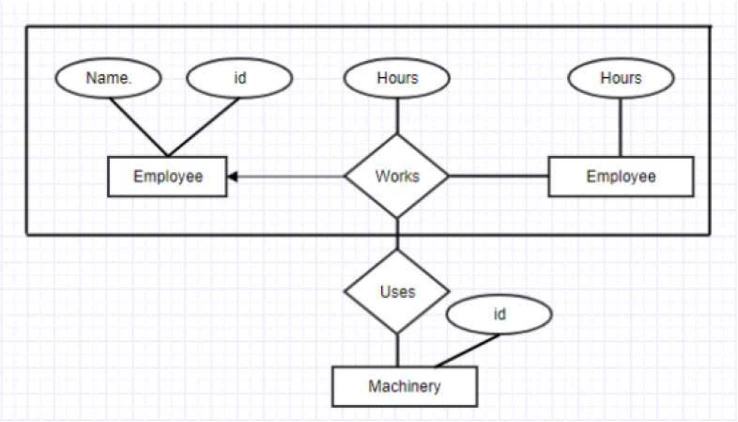
Example – 1

Here, work relation allocates projects to employee.

One employee may work on any number of projects and on one project any number of employees may be working. Work relationship relates employee and project correctly. Uses relation, which is ternary relationship, cannot relate machines, projects and

and the employees working on that project.

To represent this relation correctly, consider the relation work as one entity and machinery as another entity.



Example – 2

Here, attributes of work relationship are {emp-id, no, hours} and primary key {emp-id, no}.
Uses is relationship which relates work and machinery, whose attributes are {emp-id, no, machine-id}
Primary key – {emp-id, no, machine-id}

Attribute Inheritance

The attributes of higher-level entity set are inherited by the lower level entity sets.
A lower level entity set also inherits participation in relationship sets in which its higher-level entity set participates.
For example: Attributes of account entity are inherited by saving account, checking account.

<< Prev Post Next Post >>

RECOMMENDED POSTS

- Logical And Physical Data Independence in DBMS
- DBMS Tutorial | Database Management System Tutorial
- Basic Concepts of ER Model in DBMS
- Anomalies in DBMS – Insert, Update, Delete Anomalies
- Problems Caused By Redundancy in DBMS
- Normalization in DBMS With Example
- RDBMS Introduction, EF Codd Rules With Examples
- DBMS Relational Algebra Examples With Solutions
- Difference Between Traditional File System And Database System
- Different Keys in RDBMS With Examples

LEAVE A REPLY

You must be logged in to post a comment.

- Android With Kotlin
- Android With Java
- Kotlin
- Python
- C
- Math
- Database
- Computer Networks
- Operating System

- Python Program to Find Sum of N Natural Numbers using Recursion
- Python Program to Find Sum of N Natural Numbers
- Python Program to Check if Number is Positive or Negative
- Python Program to Find if Given Year is Leap Year or Not
- Python Program to Find Largest among N numbers

- About Us
- Contact Us
- Privacy Policy

JOIN US AT

- Facebook
- Twitter
- Youtube

NEWSLETTER

Name

Email

Subscribe