

Entity type & Entity set

A Entity type defines a collection of entities that have same attributes

ex: A company wants to store Employee's data with same attributes. Here the Entity type is "Employee"

(con)

A company want to store similar information concerning each of the employees.

* The collection of all entities of a Particular type in the database at any point in time is called an "entity set".

Primary Key attribute of an Entity type:-

An entity type usually has an attribute whose values are distinct for each individual entity in the entity set. Such an attribute is called a key attribute

* In ER diagram each primary key attribute has its name "underlined"

→ An entity type with no primary key is called a "Weak entity type"

* Value sets of Attributes

Each simple attribute of an entity type is associated with a value set, which specifies the set of values that may be used or assigned to the attribute for each individual entity.

for: "value set of age of employee is 15 to 70".

Mathematically an attribute A of the entity type E whose value set set X V can be defined as a function from E to power set $P(V)$ of V

$$A: E \rightarrow P(V) \text{ of } V$$

Composite attributes

A attribute that can be divided into smaller subparts which represents more basic attributes with independent meanings.

ex:- address

Simple attributes

Attributes that are not divisible are called simple.

Stored vs Derived attributes

Some cases two or more attribute values are related. for example age and Birthdate.

The age attribute is called derived attributes because it is derivable from the Birth-date attribute, which is called stored attribute.

Complex attributes:

combination of composite and multivalued attributes.

* we can represent arbitrary nesting by grouping components of a composite attribute b/w parentheses and separating components with commas and by displaying multivalued attributes b/w braces {}.

for example if a person can have more than one residence and each residence can have a single address and multiple phones, an attribute Address-phone for a person can be specified as shown

{ Address-phone (Address (st-no, AP-no, ...), { phone (Area-code, Phonenumbers) }) }

Multivalued attribute

An entity may have multiple values for that attribute. for example, color of a car or previous Degree of a student. Denoted as {color} or {previous Degrees}

We refer to the value of attribute A for entity e as $A(e)$.

for a composite attribute A the value set v is the Cartesian product of $P(v_1), P(v_2), \dots, P(v_n)$ where $v_1, v_2, v_3, \dots, v_n$ are value sets of simple attributes that form A

$$v = P(v_1) \times P(v_2) \times \dots \times P(v_n)$$

Constraints on Relationship Types

We can distinguish two main types of relationship constraints: cardinality ratio and participation.

Cardinality Ratio:-

The cardinality ratio for a binary relationship specifies the maximum number of relationship instances that an entity can participate in.

An example of a 1:1 binary relationship is *Manages* which relates a department entity to the employee who manages that department.

Participation Constraints:-

The constraint specifies the minimum number of relationship instances that each entity can participate in, and is sometimes called *min. cardinality constraint*.

There are two types

- ① total ② partial

$(=)$

$(-)$

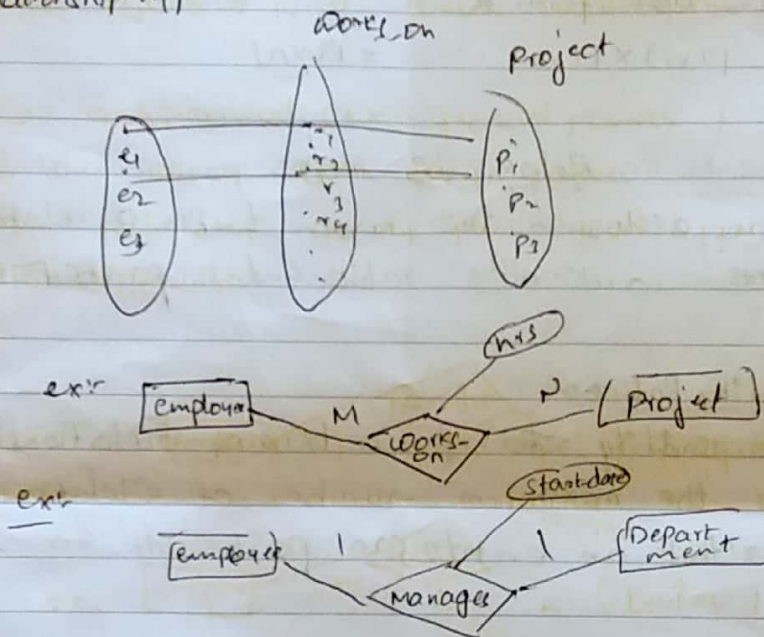
every entity of particular entity type should participate in relation.

ex:- If a company policy states that every employee must work for a department then employee entity type can exist only if it participates in at least one works for relationship instance.

Attributes of Relationship Types

Relationship type can also have attributes, similar to those of entity types.

for example to record the number of hours per week that an employee works on a project, we can include an attribute hours for the works-on relationship type.



Weak Entity Types

entity type that do not have key attributes of their own are called weak entity types.

ex: dependency (ex: driving license)

Entities belonging to a weak entity type are identified by being related to specific entities from another entity type in combination with one of their attribute values. We call this other entity type the "identifying (or) owner entity type" and we call the relationship type that relates a weak entity type to its owner the "identifying relationship".

* A weak entity type always have total participation because a weak entity cannot be identified without an owner entity. ex: A person entity can't exist unless it is related to a person entity.

Database:-

one fundamental characteristic of the database approach is that it provides some level of "data abstraction".

Data abstraction generally refers to the suppression of details of data organization and storage, and the highlighting of the essential features for an improved understanding of data.

Data Model

★ A data model - a collection of concepts that can be used to describe the structure of a database - provides the necessary means to achieve this abstraction.

★ "Structure of a database" we mean data types, relationships, and constraints that apply to the data.

★ Most data models also include a set of basic operations for specifying retrievals and update of database.

★ Many data models have been proposed, which we can categorize according to the types of concepts they use to describe the database structure.

~~Conceptual data models~~
"Conceptual data models" use concepts such as entities, attributes and relationships.

★ "Entity-Relationship model" - a popular high-level conceptual data model.

Schema:-

"In data model, it is important to distinguish b/w the description of the database and the database itself"

The description of database is called database schema.

"The schema in a relational database refers to the tables, fields and also the relationship b/w the fields and tables"

"for more info refers page 34 in Textbook (Soft copy)"

Notes

A Schema diagram^(tables) displays only some aspects of schema such as names of record types and data items, and some types of constraints.

Schema doesn't show neither the data type of each data item nor relationships among the various fields.

note

* data in the database at a particular moment in time is called database state or snapshot

* each object (group of tables) in schema is called schema construct.

ex:- STUDENT construct will contain the set of individual student entities or its instances.

* The DBMS is partly responsible for ~~ensuring~~ ensuring that every state is valid state - that is, a state that satisfies the structure and constraints specified in schema.

Hence specifying a correct schema to the DBMS is extremely important and the schema must designed with utmost care.

* The DBMS stores the description of the schema constructs and constraints also called the meta-data in the DBMS catalog so that DBMS software can refer to schema whenever it needs to.

Note's

We will refer to the cardinality ratio and participation constraints taken together, as the structural constraints of a relationship type.