Q.) What are the properties of Entity called as :- Attributes

CH-2:- DB DESIGN AND ER MODEL:-

Data Modeling:- A model is a representation of reality that retains only carefully selected essential details.
Logical org. of data for optimum info. extraction & data manipulation.

Why Model?

- To understand & identify essential data elements.
- To produce a representation that can be transformed into a schema

Data Modeling is to be able to use DB to store data, it should be designed in an efficient manner. The 1st step in designing a DB is data modeling.

It enables a DB designer to create a model that represents the way in which info. is likely to be organized in the DB.

There are 2 major ~~categories~~ methodologies used to create a data Model:-

- ER approach &
- Object model

⇒ Our pri. focus is the design of the DB

⇒ **DB Design**:-

- Foundation of a successful info. system should promote:-
  - data integrity
  - Prevent data redundancies & anomalies
- Must yield a DB that is efficient in its provision of data access. It serves the needs of the info. system.

⇒ **DB design Strategies**:-

- Top-down design (e.g:- ER modeling)
  Identify entities / data sets
  Define attributes / data elements for each entity.
- ~~Dot~~ Bottom-up design (e.g:- normalization)
  Identify attributes
  Group them together to define entities

- Centralized design :-
  - Small no. of obj. & procedures
  - Single design process
- Decentralized design :-
  - Large no. of entities with complex relations & ofd.
  - multiple parallel design of subsystems & aggregation.

— Subparts of Top-down & Bottom-up design

→ **Steps in design Process :-**

i) Requirements analysis (where the org. is presently working & where they want to work in future)
ii) Conceptual DB design
iii) Logical DB design] — format of data to store
iv) Schema refinement (str. defined)
v) Physical DB design (how actually data stored in physical servers)
vi) Appl^n & Security design

→ **Feasibility Analysis :-** (Is the design actually possible in real life?)

- Technological feasibility :-
  - What H/w, S/w & additional would be needed?
  - What is available in-house? What has to be purchased?
  - How will the new system be integrated?
- Operational feasibility :- (day to day activities)
  - who will design the system?
  - ,, ,, maintain ,, ,, ?
  - ,, ,, do training or help-desk support?
  - Can the available ~~person~~ personnel provide the time?
- Economic feasibility :-

  Expected cost of the overall project?
  - S/w, H/w, appl^n development, staff-time
  - Hidden cost (unforeseen)

  Benefits
  - How soon expected?

Steps to Design ER Diagrams :-

Step1 :- Identify the strong & weak entity sets

Step2 :- Identify the Relevant Attributes

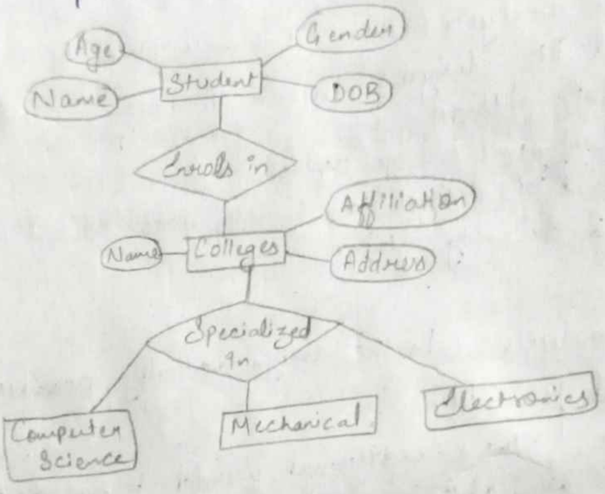
The next step is to get all the attributes that are most applicable for each entity set. Do this work by considering each entity set in mind & also the type of attributes.

Step 3:- Identify the Relationship Sets

In this step we need to find all the meaningful relationship sets among possible entity sets. This step is very tricky, as redundant relationships may lead to complicated design & in turn a bad implementation.

Step 4 :- Indentify the Cardinality (no. of tuples) Ratio & Participation Constraints.

Sample :-



Design Issues :-

1. Use of Entity Sets Vs. Attributes :-

- Choice mainly depends on the str. of the enterprise being modeled, & on the semantics associated with the attribute
- What is an entity & Attribute?
- When to represent a value as an attribute?
- ,, ,, ,, ,, ,, ,, a separate entity-set?
- Representing as a separate entity-set allows details to be added later.

2.
- It is difficult to examine if an obj. can be best expressed by an entity set or relationship set.
- To understand & determine the right use, the user need to designate a relationship set for describing an action that occurs in-b/w the entities.

3. Use of Binary Vs. n-ary Relationship Sets :-

Generally, the relationships described in the DB are binary relationships. However, non-binary relationships can be represented by several binary relationships.

For. eg:- we can create & represent a ternary relationship 'parent' that may relate to a child, his father, as well as his mother.

4 Placing Relational Attributes:-

The Cardinality ratios can become an affective measure in the placement of the relationship attributes. So, it is better to associate the attributes of one-to-one or one-to-many relationship sets with any participating entity sets.

Weak Entity Sets:-

- It is one that can only exist when owned by another one.

Example:- a ROOM can only exist in a BUILDING.
On the other hand, a TIRE might be considered as a strong entity bcoz it also exist without being attached to a CAR.

- Unlike Strong entity, a weak entity does not have any pri. key. It instead has a partial discriminator key.
- A weak entity is represented by a double rectangle
- The combination of discriminator & pri. key of the strong entity set makes it possible to uniquely identify all entities of the weak entity set.

Pri. key of Weak Entity Set = Its own discriminator + Pri. key of Strong Entity Set

Example:- In a University, a course is a strong entity, & a course offering can be modeled as a weak entity.
The discriminator of course offering (subparts of course) would be semester & section no. (if)

Extended ER features:-

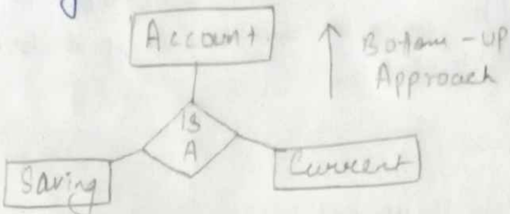
As the complexity of data is increasing, it became more & more difficult to use the traditional ER model for DB modeling. Hence some improvements or enhancements were made to the existing ER Model to make it able to handle the complex appl^{n} better.

- Generalization
- Specialization
- Aggregation

These 3 are used for data (summary) abstraction in which abstraction mechanism is used to hide details of a set of obj.

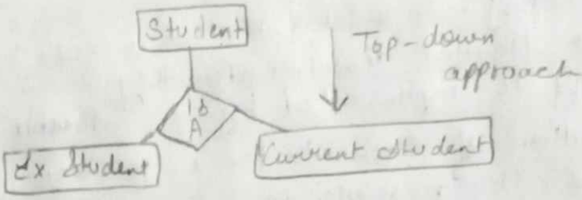
- Generalization:-

It is a bottom-up approach in which 2 lower level entities combine to form a higher level entity. In this, the higher-level entity can also combine with other lower level entity to make further higher level entity.



- Specialization:- opp. to generalization.

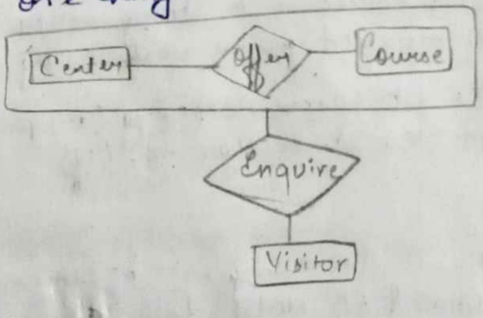
It is a top-down approach in which one higher level entity can be broken down into 2 lower-level entity. In ~~spect~~ this, a higher level entity may not have any lower-level entity sets, it's possible.



- Aggregation:-

A process when relation b/w 2 entities is treated as a single entity

It is a process in which a single entity alone is not able to make sense in a relationship so the relationship of 2 entities acts as one entity.



Relational DB:-

RDBMS is used to manage Relational DB. Relational DB is a collection of organized set of tables related to each other, & from which data can be accessed easily. These DB is the most used DB

Tuple:- row/record

Attribute:- column

Attribute Domain:- When an attribute is defined in a relation (table), it is defined to hold only a certain type of values, which is known as Attribute domain

⇒ **Relation Schema** :- It describes the str. of the relation, with the name of the relation (~~at~~ name of table), its attributes & their names & type.

⇒ **Relational Integrity Constraints** :-
Every relation has some conditions that must hold for it to be a valid relation. These conditions are called Relational Integrity Constraints. There are 3 main integrity constraints -

- Key constraints.
- Domain constraints
- Referential integrity constraints.

- Key Constraints :- In a relation with a key attribute, no 2 tuples can have identical values for key attributes.
  A key attribute can not have NULL values.
  — Pri. key ; Unique key
- Domain Constraints :-
  Attributes have specific values in real-world scenario.
  e.g:- age can only be +ve int. ~~the~~
  Every attribute is bound to have a specific range of values.
  — Check constraint.
- Referential Integrity Constraints :-
  Works on the concept of Foreign keys.
  If a relation refers to a key attribute of a diff. or same relation, then that key element must exist.

Relational Algebra Operators :- (Set theory of mth & Base of SQL)

- SELECT Operation :- works on tuples
  Notation :- $\sigma_p(r)$
  p is called the selection predicate
  Example :- $\sigma_{CITY=\text{"AMBALA"}}$ (STUDENT)
  σ = Select operator
  r = relation
- PROJECT Operation :- works on attributes
  Notation :- $\Pi_{A1,A2,\ldots,Ak}(r)$
  where A_1, A_2 are attribute names & r is the relation name.
  Π = Project operator
  Example :- $\Pi_{CITY,CLASS}$ (STUDENT)
  Duplicates are removed

— Unary Operators

- UNION Operation :- Binary operator

Notation :- r U s

For r U s to be valid.

1. r, s must have the same arity (same no. of attributes)

2. The attribute domains must be compatible

r & s are 2 relations.

Example :- To find all the customers with either an account or a loan.

$$\Pi_{customer-name}(depositor) \cup \Pi_{customer-name}(borrower)$$

Duplicates not allowed.

- SET DIFFERENCE Operation :-

Notation :- r − s

& This opd. must be taken b/w compatible relations.

r & s must have the same arity.

attribute domains of r & s must be compatible.

→ All the values of r which are not present in s.

- CARTESIAN PRODUCT :-

Notation :- r X S

Applying this on 2 relations that is one two sets of tuples, it will take every tuple one by one from the left set (relation) & will pair it up with all the tuples in the right set (relation)

- RENAME Operator :-

Allows us to name, & ∴ , to refer to, the results of relational-algebra expressions.

Allows us to refer to a relation by more than one name

Example :- $\rho_x(E)$

returns the expression E under the name X.