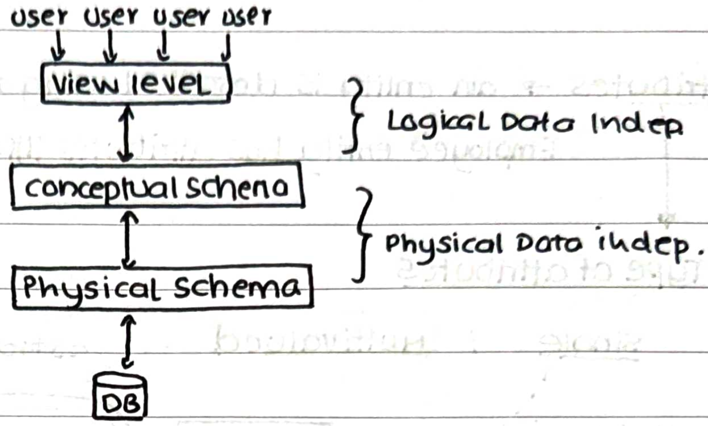
Database Management System

- Data Independence



- Keys (Primary, Candidate, Foreign)

Primary Key : unique, non-null attribute which uniquely identifies a row.

Candidate Key : set of unique(not neccessarily non-null) attributes which can become primary key.

Foreign Key : attribute / set of attributes that refers to the p.k. of same as some other table. (establishing a relation b/w 2 tables)

(maintains referential integrity → Meaning data insertion, del., upd. operations in the referenced and referencing tables must be done such that data is consistent everywhere)

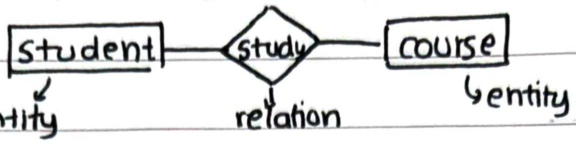
Super key : superset of any candidate key.

- Steps Involved in DB Design

Requirement Analysis (user's needs)
↓
Conceptual Design (High Level ER diagram)
↓
Logical Design (Tables)
↓
Schema Refinement (Normalization)
↓
Physical Design (Indices)
↓
Security Design (Access control)

- Entity-Relationship Model (ER)

entity → an object in real world (a noun)



attributes → an entity is described using a set of attributes.

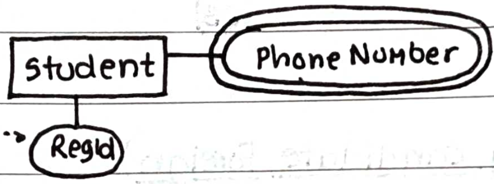
Employee entity has attributes like name, salary, social security number.

Type of attributes

(I) Single / Multivalued

a student may have multiple numbers

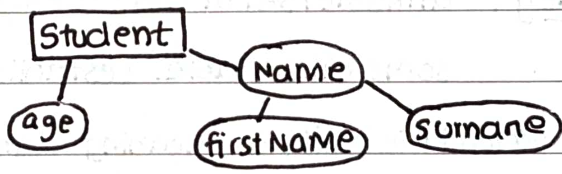
a student will only have a single regⁿ ID.



(II) simple / composite

cannot be further divided (age)

can be further divided (Name = firstName + surname)



(III) Stored / Derived → represented by (age)

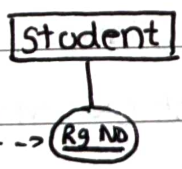
stored in DB as asked from user (DOB)

calculated from stored attributes (age) → (Present Date - DOB)

(IV) Key / Not Key

unique attribute (PK)

represented as

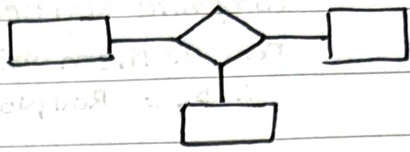


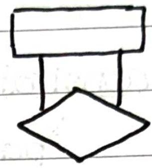
(a key is a minimal set of attributes whose value uniquely identify an entity in the set.)

Relations → association among entities.

relation represented by 

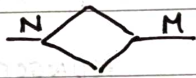
relations are usually 'verbs.'

ternary relationships → 

recursive relation^n →  → for ex. 

Here an employee will report to another employee only.

mapping cardinalities → many to many (N to M)

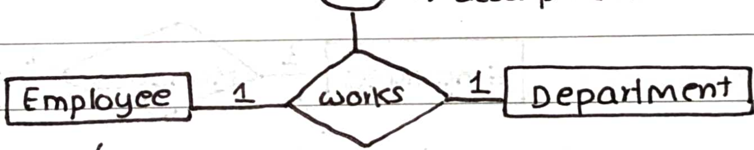


one to one (1 to 1)

one to many (1 to N)

 --> descriptive attributes of a relation.

For example,

ER ⇒ 

Tabular ⇒

PK

EID	Ename	age

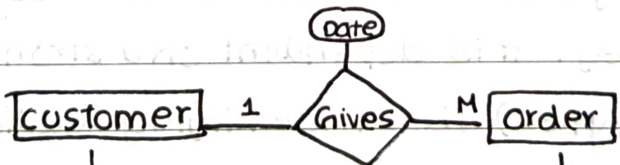
FK FK

EID	DID

PK

DID	DName	DLoc

either of them can be PK because all values are unique since the rel^n is 1 to 1.

ER ⇒ 

Tabular ⇒

PK

CID	Name	City

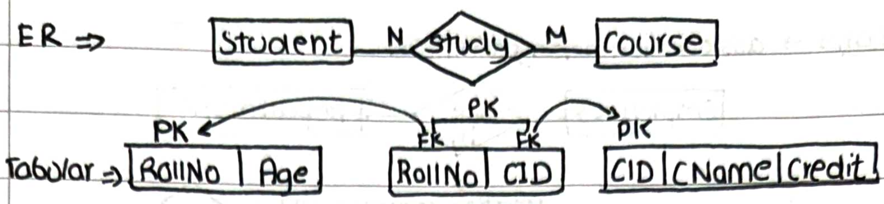
FK PK FK

CID	OID	Date

PK

OID	ItemName	cost

here OID will be the primary key because related to one CID there can be N orders, but related to one OID, there can only be one CID, ie. one customer.



Here both Roll No and CID will form a composite key because on their own, none of them uniquely identifies a row.

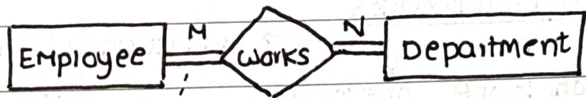
∴ PK = Roll No, CID

Participation constraints

Total part.

partial part.

<u>Total</u>: the part. of an entity set E in a relationship R is total if every entity in E participates in atleast one relation in R. For ex.:



representing total part. meaning that each employee should work in <u>atleast one</u> dept, and a dept. must have <u>atleast one</u> working employee

<u>Partial</u>: If only some entities in E participate in R. For ex.:

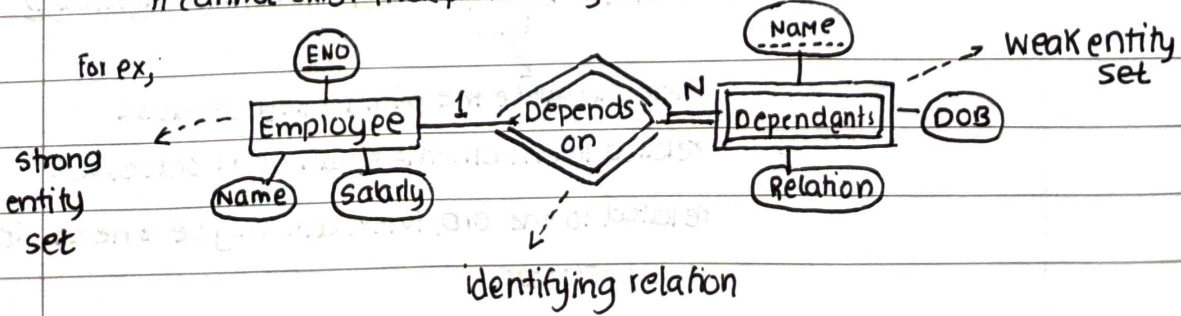
representing partial part., meaning that not all employees manage a dept.

Weak Entity sets

It is an entity set that does not have enough attributes to form a primary key. It is dependent on a strong entity set (with a defined primary key) via an identifying relation.

<u>NOTE</u>: A weak entity set always has a <u>total part</u>. in the identifying relation as it cannot exist independently.

For ex,



both the weak entity and the identifying relation are represented by double lines and the partial key (identifies weak entities related to some owner)) is denoted by underlining with dotted line.

There can be an employee without a dependent, but there will be no record of a Dependent in the system unless the dependent is related to some employee

- Normalization

It is a technique to remove/reduce redundancy from a table. It removes insertion, deletion and updation anomalies from the table.

insertion anomaly : when certain attributes cannot be inserted indivisually into the table, w/o the presence of other attributes.

deletion anomaly : when we delete a record that may contain attributes that shouldn't be deleted.

updation anomaly : results in data inconsistency from redundancy and partial update.

1st Normal Form (1NF)

A table is in 1NF if

(I) domain of each attribute consists of only atomic values.

(II) cells of table does not contain multivalued or composite entries.

For example,

ENO.	Name	Dependents
103	Rahul	Raghav/Seema
104	Amit	Anil/Manoj/Geeta

NOT IN 1NF

option 1 (1NF)

ENO.	Name	Dep.
103	Rahul	Raghav
103	Rahul	Seema
104	Amit	Anil
104	Amit	Manoj
104	Amit	Geeta

option 2 (1NF)

Em.	Name	Dep1	Dep2	Dep3
103	Rahul	Raghav	-	-
104	Amit	Anil	Manoj	Geeta

option 3 (1NF)

ENO.	Name
103	Rahul
104	Amit

ENO.	Dependent
103	Raghav
103	Seema
104	Anil
104	Manoj
104	Geeta.

1 # Functional Dependency

$X \rightarrow Y$ Means X determines (uniquely) Y, or Y is uniquely determined by X. If you know 'X', then there is only one 'Y' to match.

Trivial f.d. $\Rightarrow$ if $X \rightarrow Y$, then $X \cap Y \neq \Phi$.

NonTrivial f.d. $\Rightarrow$ if $X \rightarrow Y$, then $X \cap Y = \Phi$.

properties of functional Dependencies:

(i) Reflexivity : If $Y \subseteq X$, then $X \rightarrow Y$

(ii) Augumentation : if $X \rightarrow Y$, then $XZ \rightarrow YZ$

* (iii) Transitivity : If $X \rightarrow Y$ and $Y \rightarrow Z$, then $\underline{X \rightarrow Z}$

* (iv) Union : If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$

* (v) Decomposition : If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

(vi) Pseudo transitivity : If $X \rightarrow Y$ and $WX \rightarrow Z$, then $WX \rightarrow Z$

(vii) composition : If $X \rightarrow Y$ and $Z \rightarrow W$, then $XZ \rightarrow YW$.

Closure Method to find the set of candidate keys

A^+ (ie. A closure) is the set of all the attributes that are determined by A. If A can determine all the attributes in the set, then A is a candidate key.

NOTE : A candidate key is a set of minimal attribute(s), that can determin all other attributes. If A is a candidate key, then AX is a super key (X is another attribute of the set), not a candidate key as A was self sufficient and candidate keys are minimal.

For example, given R(ABCDE) and FD = {$A \rightarrow B$, $BC \rightarrow D$, $E \rightarrow C$, $D \rightarrow A$}. We have to find the set of candidate keys.

$A^+ = \{A, B\}$ *(since E never comes on RHS, it will always be a part of every candidate key)

$(AE)^+ = \{A, E, C, B, D\}$

$(DE)^+ = \{A, D, C, B, E\}$ $\therefore$ $C = \{AE, DE, BE\}$

$(BE)^+ = \{A, B, C, D, E\}$ set of candidate keys.

Canonical covers of FDs.

These are minimal/optimal representations of FDs by eliminating 'extraneous attributes.'

RULES : (i) RHS of every FD is a single attribute
(ii) closure of F_c is equal to closure of F
(iii) F_c is minimal

for example, find F_c for $F = \{ \overset{(1)}{AB \rightarrow C}, \overset{(2)}{A \rightarrow B}, \overset{(3)}{A \rightarrow C}, \overset{(4)}{B \rightarrow C}, \overset{(5)}{A \rightarrow B} \}$

Remove (2) as (2) & (4) are same.

Since A can determine C in (3), we don't need B. ie. we can remove (1).

(3) is redundant due to transitivity b/w (5) and (4). So we remove (3).

$\therefore F_c = \{ A \rightarrow B, B \rightarrow C \}$.

Equivalence of 2 sets of FDs, E & F.

Two sets of FDs are equal if E covers F ie. every dependency of F can be inferred from E, and F covers E ie. every dependency of E can be inferred from F.

For example, $E = \{ A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H \}$

$F = \{ A \rightarrow CD, E \rightarrow AH \}$

(I) Check if E covers F :

Take all FDs of F, one at a time :

$A \rightarrow CD \Rightarrow$ take A^+ using FDs in E. A^+ in $E = \{A, C, D\}$ ✓

$E \rightarrow AH \Rightarrow$ take E^+ using FDs in E. E^+ in $E = \{E, A, D, H\}$ ✓

$\therefore$ E covers F.

(II) Check if F covers E :

Take all FDs of E, one at a time :

$A \rightarrow C \Rightarrow$ Take A^+ in F. A^+ in $F = \{A, C, D\}$ ✓

$AC \rightarrow D \Rightarrow$ Take AC^+ in F. $AC^+ = \{A, C, D\}$ ✓

$E \rightarrow DA \Rightarrow$ Take E^+ in F. $E^+ = \{E, A, D, H, C\}$ ✓

$E \rightarrow H \Rightarrow$ Take E^+ in F. $E^+ = \{E, A, D, H, C\}$ ✓

$\therefore$ F covers E.

$\therefore$ E and F are equivalent.

2nd Normal Form (2 NF)

A table is in 2 NF if:

(i) it is in 1 NF.

(ii) No non-prime attributes (which are not a part of candidate key), should be dependent upon a part of the candidate key, ie. if P → NP dependency exists, then it should be the whole of the candidate key, not a part of it. (no partial dependency exists)

3rd Normal Form (3NF)

(i) Table should be in 2NF.

(ii) No non-prime attribute should derive another non-prime attribute.
(no transitive dependency exists)
↓
(either LHS is candidate key OR RHS is a prime attibute)

BCNF (Boyce Codd NF) → Special case of 3 NF.

(i) Table should be in 3 NF

(ii) Every attribute should be derived only from candidate keys (or superkeys), ie. LHS should only have candidate keys.
↓
(

Decompositions

Ideal decomposition of a table should be : lossless & dependency preservin

Lossy Decomposition
↓
When R is decomposed into R1, R2 (say) and we try to join R1 and R2 again using a natural join (R1 x R2), we donot obtain the original R table if the decomposition was lossy.

Lossless Decomposition

If
- $R_1 \cup R_2 = R$ (retrieve original table)
- $R_1 \cap R_2 \neq \phi$ (Two generated tables should atleast have one common column)
- The common column (attribute) should be a candidate key (or superkey) in atleast one of the new tables.

Dependency Preservation → refer to gate samsher