C-2 31-10-20

Background:
Network and Hierarchical Database Systems

It was emerged in late 1960s. and it was

i. complex Data structures

ii. No separation between logical and physical data structure.

iii. Navigational programming language used.
↳ It means how we will fetch the data need to tell by instruction. (how to fetch, what to fetch)

Relational Data Model → Relational Algebra

1970 by E.F Codd i. It based on Mathmatical foundation

ii. There is complete separation of logical and physical data structure.

iii. complete data model

iv. It solves the problem with previous model.

v. has become a dominant model.

* Basic terms & concepts (Domain, attribute, relation, tuple, Relation schema, instance, constraints)

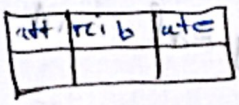
objectives –

* high degree of data independence. In other words data is not dependent on any application

* bring consistency, semantics and out redundancy problems.

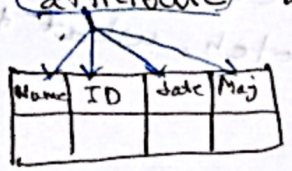
* Operation become set oriented rather than particular.

Relation: A relation is a table, or entity with columns and rows of logically related data.



Attribute: (Column Name)

An attribute is a named column of a relation.



attribute 없으면?
⇒ Relation 없다!

Domain: (Allowed values) (Every attribute has a set of Domain)

A domain is a set of allowable values for one or more attributes.

A set of values $D = \{D_i \mid i = 1, \ldots, n\}$ where D is domain name and D_i is a domain element.

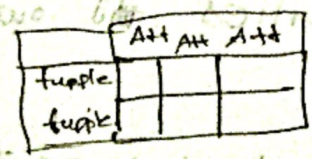
Example: LName Dom = {'Susan', 'James', 'John', ...}

* we cant enter anything which is not in Domain.

Degree = # of cols (Number of attribut a relation contains)

Cardinality = # of rows or tupples.

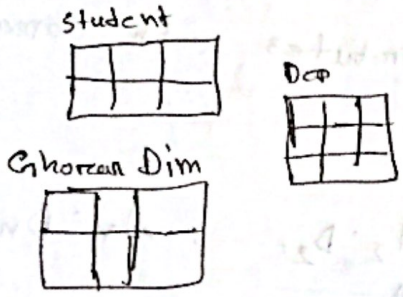
Tupples: A row is called as tupple in a relation.



Relational Database:

↳ It is a process to create mathematically accurate Database.

A collection of normalized relations with distinct relation names.



** Relation / Table / entity

Express by mathematical term:

Let, two sets, $D_1 = \{2, 4\}$ and $D_2 = \{1, 3, 5\}$

The cartesian product of D_1 and D_2 is $D_1 \times D_2$

$D_1 \times D_2$ = Set of all ordered pairs such that first element is a member of D_1 and last is a member of D_2.

$D_1 \times D_2 = \{(2,1), (2,3), (2,5), (4,1), (4,3), (4,5)\}$

↳ Any subset of this product is a relation

example, $R = \{(2,1), (4,1)\}$

Relation Schema: (combination of Attribute and Domain) ~~(A:D1, D2)~~

A named relation defined by the set of attribute and domain name pairs.

Let $A_1, A_2 \ldots, A_n$ be attributes with domains

$D_1, D_2 \ldots, D_n$

Then the set, $\{A_1 : D_1, A_2 : D_2, \ldots, A_n : D_n\}$ is a relation schema

Relational Database Schema:

A set of relation schemas, each with distinct name.

if $R_1, R_2 \ldots, R_n$ are a set of relation schema then we can say,

Relational Database Schema $R = \{R_1, R_2, \ldots, R_n\}$

simply Collection of Relation Schema

Relational Keys:

Keys are special fields of a relation mainly for identification.

Here we have tow main purposes-

i. To identify a record (tuple) uniquely in a relation. [Primary Key]

ii. To identify or relate to another relation records. [Foreign Key]

** Keys can be simple (a single field) or comosite (more then one field).

(Set of all primary key)

Superkey: (Combination of all uniquly identifier)

An attribute or set of attributes, that uniquly identifies a tuple within a reelation

** A superkey may contain additional attributes that are not necessary for unique identification.

candidate Key:

A superkey such that no proper subset is a superkey within the reelation.

* A candidate key is a superkey with minimal attributes.

OR

~~Keys as candidate~~ keys that are candidate for superkey.

In a table it may seems there is one or more primary key then these are candidate primary key. बनावायन प्राथमी

Relational Integrity constraints

An attribute can have a valid value of its Domain. Thats why we have to consider two important integrity rules-

1. Entity Integrity: In a base (physical database) relation, no attribute of a primary key can be null.

ii. Referential Integrity constraints:

if a foreign key exist in a relation, then there can be two things. one is the foreign key value must match a candidate key value of some tupple in its home relation and or other thing is the foreign key value must be wholly null.