Chapter 3
The Relational Model

The Relational Model

The model was proposed by E. F. Codd in 1970 and consists of the following three components:

- *Data Structure* - a data structure for building the database.
- *Data Manipulation* - a collection of operators that may be used to retrieve, derive or modify data stored in the database.
- *Data Integrity* - a collection of rules that implicitly or explicitly define a consistent database state or changes of states.

Data Structure

- The beauty of the relational model is the simplicity of its structure.
- Its fundamental property is that all information about the entities and their attributes as well as the relationships is presented to the user as tables (called *relations*) and nothing but tables.

- The rows of the tables may be considered records and the columns as fields.

Properties of Relations

- Each relation contains only one record type.
- Each relation has a fixed number of columns that are explicitly and uniquely named.
- No two rows in a relation are the same.
- Each item or element in the relation is atomic.
- Rows have no ordering associated with them.
- Columns have no ordering associated with them (although most commercially available systems do).

Terminology

- *Relation* - essentially a table
- *Tuple* - a row in the table
- *Attribute* - a column in the relation
- *Degree of a relation* - number of attributes in the relation. A relation with degree *N* is *N-ary* relation.
- *Cardinality of a relation* - number of tuples
- *Domain* - a set of values that an attribute is permitted to take
- *Primary key* - an attribute (or a set of attributes) that uniquely identifies each tuple and satisfies minimality

Keys in Database

- Keys are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.
- **Keys** are defined to easily identify any row of data in a table.

Example of keys

student_id	name	phone	age
1	Akon	9876723452	17
2	Akon	9991165674	19
3	Bkon	7898756543	18
4	Ckon	8987867898	19
5	Dkon	9990080080	17

Types of keys

- Super key
- Candidate key
- Primary key
- Composite key
- Alternate key

Super Key

- **Super Key** is defined as a set of attributes within a table that can uniquely identify each record within a table.
- Super Key is a superset of Candidate key.
- Id, id and name are the superkeys

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

Candidate keys

- Candidate keys are defined as the minimal set of fields which can uniquely identify each record in a table.
- It is an attribute or a set of attributes that can act as a Primary Key for a table to uniquely identify each record in that table.
- There can be more than one candidate key.
- Id, name and dept_name are the candidate keys
- Id, name is not candidate key since its not minimal

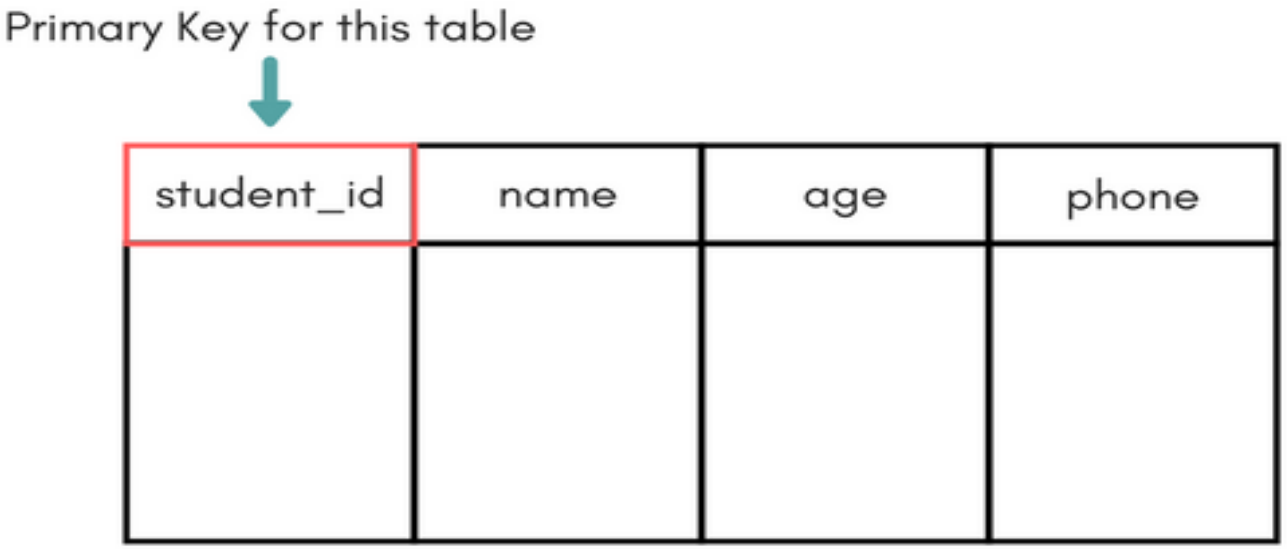
Properties of candidate keys

- A candiate key can never be NULL or empty. And its value should be unique.
- There can be more than one candidate keys for a table.
- A candidate key can be a combination of more than one columns(attributes).

Primary key

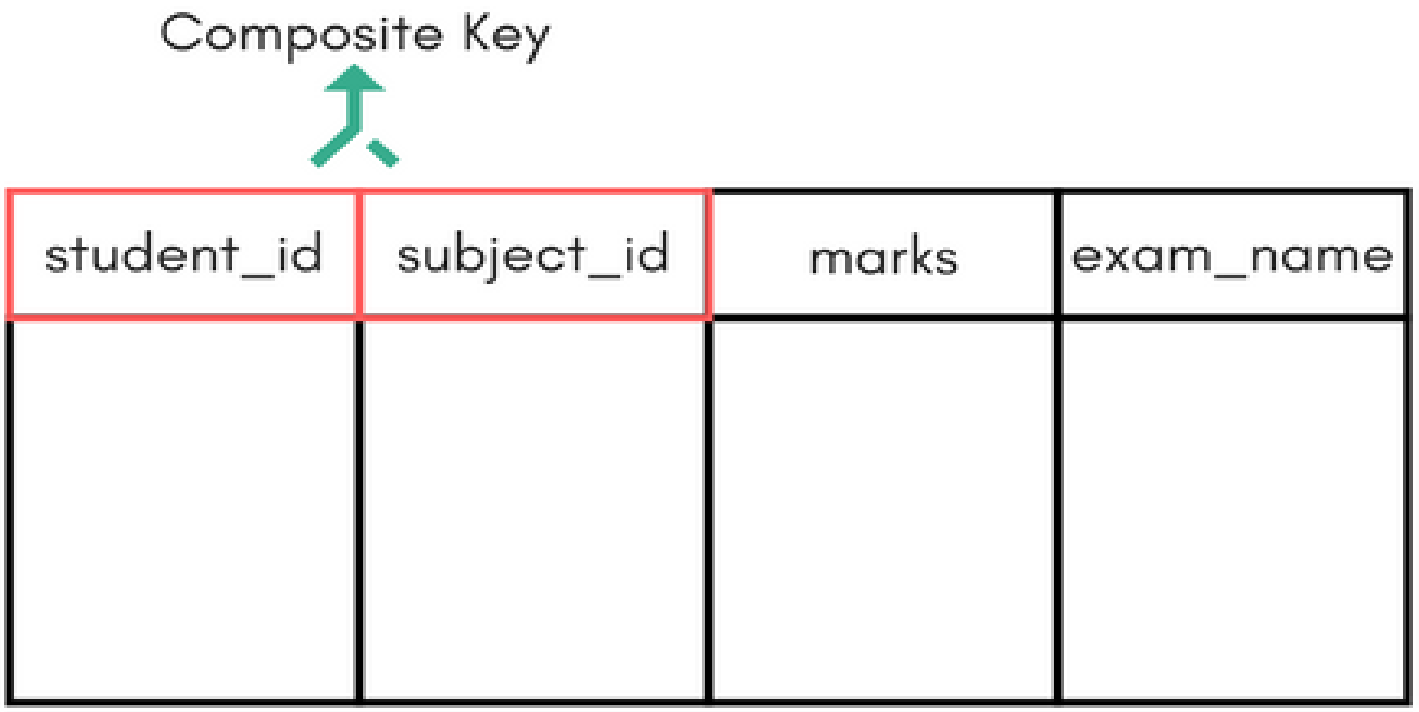
- Primary key is a candidate key that is most appropriate to become the main key for any table.
- It is a key that can uniquely identify each record in a table.
- Id is the **primary key** for the instructor relation.

- Primary key of Student table which stores the details of a student



Composite Key

- Key that consists of two or more attributes that uniquely identify any record in a table is called **Composite key**.
- But the attributes which together form the **Composite key** are not a key independently or individually.



Score Table - To save scores of the student for various subjects.

- **Secondary or Alternative key**
  - The candidate key which are not selected as primary key are known as secondary keys or alternative keys.
- **Non-key Attributes**
  - **Non-key** attributes are the attributes or fields of a table, other than **candidate key** attributes/fields in a table.
- **Non-prime Attributes**
  - **Non-prime** Attributes are attributes other than **Primary Key attribute(s).**.

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

STUDENT_COURSE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

Table 2

Identify superkey, candidate key, primary key, foreign key, alternate key

Integrity

For example, if a relation CLASS (s-id, lecturer, cno) has (s-id, lecturer) as the primary key then allowing tuples like the following leads to ambiguity:

3123	NULL	IT100
NULL	Smith	IT100

Since the primary key is the tuple identifier, changing it needs very careful controls. Codd has suggested three possible approaches:

Integrity

- Only a select group of users be authorised to change primary key values.
- Updates on primary key values be banned. If it was necessary to change a primary key, the tuple would first be deleted and then a new tuple with new primary key value but same other attributes' values be inserted.
- A different command for updating primary keys be made available.

Domains

Although some commercial database systems do not provide facilities for specifying domains, domains could be specified as below:

create	DOMAIN	Name1	CHAR(10)
create	DOMAIN	S-id	INTEGER
create	DOMAIN	Name2	CHAR(10)

Note that Name1 and Name2 are both character strings of length 10 but they now belong to different (semantic) domains.

Domains

It is important to denote different domains to

- constrain unions, intersections, differences, and equi-joins of relations.
- let the system check if two occurrences of the same database value denote the same real world object.

Nulls and Missing Information

The issue of Null values in a database is extremely important since information is often incomplete in the real world. A large number of issues arise. For example, if two NULL values are compared, should they be considered equal? How should NULL values be treated when aggregate operators are used?

When information is missing from a database, the DBA must consider:

(1)What kind of information is missing?
(2)What is the main reason for it being missing?

Nulls and Missing Information

The missing information may be *missing-but-applicable* or it may be *missing-and-inapplicable*. The other types of missing information is possible e.g. *missing - not defined* or *missing - does not exist*.

It has been suggested that different type of *marks* be used to represent different types of missing information, for example, *A-mark* and *I-mark*.

Nulls

Codd suggests use of three-value logic to solve some of the difficulties that arise because of missing information. For example, let us pose the following queries:

find all students from Germany.
find all students from outside Germany.

We may assume that the two queries will include all the students but they may not if some information is missing.

Advantages of the relational model

Codd in his 1970 paper indicated that there were four major objectives in designing the relational data model.

- *Data Independence* - to provide a sharp and clear boundary between the logical and physical aspects of database management.

Advantages

- *Simplicity* - to provide a simpler structure than were being used at that time. A simple structure is easy to communicate to users and programmers and a wide variety of users can interact with a simple model.
- *Set-processing* - to provide facilities for manipulating a set of records at a time so that programmers are not operating on the database record by record.
- *Sound Theoretical Background* - to provide a theoretical background for database management field.

Important Questions

- What are keys?
- What are their types?
- Given a relation identify the types of keys
- What are nulls
- What are the advantages of relational model
- What are selection, project and cartesian product
- Numericals on relational algebra