

Relational Model

Rajeev Srivastava
rajeevs@cdac.in

ToC

- Example of Relation
- Tuples
- Cardinality
- Attribute
- Degree/Arity
- Domains
- Relation vs. Table
- Definition of a Relation
- Candidate Key
- Super Key
- Primary Key
- Composite Key
- Foreign Key
- Integrity Rules

Example of Relation

Consider the relation EMPLOYEE represented by the following table:

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode |
|---------|------|-------|-------|----------|-------------|--------|----------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Tuples in a Relation

A relation is a set of tuples; each row here is a tuple :

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode | |
|---------|------|-------|-------|----------|-------------|--------|----------|---|
| | | | | | | | | 1 |
| | | | | | | | | 2 |
| | | | | | | | | 3 |
| | | | | | | | | 4 |
| | | | | | | | | 5 |
| | | | | | | | | 6 |
| | | | | | | | | 7 |

Cardinality of a Relation

No of Tuples in a Relation at a point in time.

Cardinality = 7

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode | |
|---------|------|-------|-------|----------|-------------|--------|----------|---|
| | | | | | | | | 1 |
| | | | | | | | | 2 |
| | | | | | | | | 3 |
| | | | | | | | | 4 |
| | | | | | | | | 5 |
| | | | | | | | | 6 |
| | | | | | | | | 7 |

Attribute in a Relation

An attribute represents a quality/information about an entity.

A tuple consists of Attribute values.

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode |
|---------|------|-------|-------|----------|-------------|--------|----------|
| | | | | | | | |
| | | | | | | | |

Degree (=Arity) of a Relation

A degree or arity of a Relation is the number of attributes in it.

Degree = 8

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode |
|---------|------|-------|-------|----------|-------------|--------|----------|
| | | | | | | | |
| | | | | | | | |

Domains

Each attribute has a domain associated with it.

Attribute values in a relation are restricted to the values from its domain.

| EmpCode | Name | Desig | Grade | JoinDate | BasicSalary | Gender | DeptCode |
|---------|------|-------|-------|----------|-------------|--------|----------|
| | | | | | | | |
| | | | | | | | |

DESIG

PE

TO

STO

DEPT

ACCO

PURC

COUR

Consider the Employee relation defined as :

```
create table EMPLOYEE(  
    EmpCode    integer(4),  
    Name       char(30),  
    Desig      char(4),  
    Grade      integer(4),  
    JoinDate   date,  
    Basic      integer(7),  
    Gender     char(1),  
    DeptCode   char(4) )
```

Domains of Attributes of Employee Relation :

| | |
|----------|--------------------------------------|
| EmpCode | set of all 4-digit numbers |
| Name | set of all 30-alpha characters |
| Desig | set of all designation codes |
| Grade | set of all grade values |
| JoinDate | set of all dates (in a given range) |
| Basic | set of all possible values for basic |
| Gender | set {'M','F', 'T'} |
| DeptCode | set of all dept codes |

A Relation *may be represented as a* Table *where*

| Relation | Table |
|--------------|--|
| Tuple | Row/Record |
| Attribute | Column |
| Degree/Arity | No of Columns in the table |
| Cardinality | No of Rows in the table |
| Domain | Pool of acceptable values for a column |
| Primary Key | Unique Identifier |

But, a Relation is not a Table, because :

- A table has an inherent order for rows; there is no concept of order for tuples in a relation.
- A relation must have a primary key; a table need not have an identifier.
- The tuples in a relation must be unique; there is no such restriction for tables

Relation (R) : Observations

- R is set of tuples.
- R is time-variant.
- R cannot have duplicate tuples.
- Tuples are unordered.
- Attribute values are atomic.
- R is a subset of the Cartesian Product of a set of domains.

Candidate Key & Super Key

A set of attributes is said to be **Candidate Key** if and only if it satisfies the following time-independent properties:

Uniqueness property: No two distinct tuples have the same value for the key.

Minimality property: None of the attributes of the key can be discarded from the key without destroying the uniqueness property.

A **Super Key** is a Non-Minimal Candidate Key.

Candidate Key?

```
create table EMPLOYEE(  
    EmpCode      integer(4),  
    Name         char(30),  
    Desig        char(4),  
    Grade        integer(4),  
    JoinDate     date,  
    Basic        integer(7),  
    Gender       char(1),  
    DeptCode     char(4) )
```

Candidate Key?

```
create table EMPLOYEE (  
    EmpCode      integer(4),  
    Name         char(30),  
    Desig        char(4),  
    Grade        integer(4),  
    JoinDate     date,  
    Basic        integer(7),  
    Gender       char(1),  
    DeptCode     char(4),  
    Email        char(100),  
    MobileNo     char(16) )
```


Primary Key

is a candidate key that have following **two qualities** -

- Uniquely identifies a tuple in a relation
- Must NOT be NULL

**Should be selected from candidate keys such that it never/rarely changes.*

Primary Key?

```
create table EMPLOYEE (  
    EmpCode      integer(4),  
    Name         char(30),  
    Desig        char(4),  
    Grade        integer(4),  
    JoinDate     date,  
    Basic        integer(7),  
    Gender       char(1),  
    DeptCode     char(4),  
    Email        char(100),  
    MobileNo     char(16) )
```

Composite Key

- A candidate key with two or more attributes that uniquely identifies the tuple in a Relation.
- Also called as compound key

Composite Primary Key

- A primary key which is a composite key is called as Composite Primary Key.

Can we have more than one primary key in a table?

No. We can not.

People who are new to RDBMS are often get confused with the restriction of single Primary Key in a table and the concept of Composite Key. To clarify the same -

A table can have only one Primary Key.

The Primary Key can be defined on a single column or more than one columns. If the Primary Key is defined using more than one columns, it is known as a Composite Key (or Composite Primary Key).

Therefore, a Composite Key in a table does not mean that there are more than one Primary Keys in the table. Instead, a Composite Key uses more than one columns to define a (Single) Primary Key.

Foreign Key

- A Foreign Key is a set of attributes in one relation whose values are required to match one of the values of the primary key of the same or different relation.
- There can be more than one foreign key in a given relation.

Identify a relation in any system/business, define its Attributes, Domain for each attribute and find out Primary, Key, Foreign Keys, Candidate Keys, Super Key in the relation.

Foreign Key(s)?

create table EMPLOYEE(

| | |
|----------------|--------------|
| <u>EmpCode</u> | integer(4), |
| Name | char(30), |
| Desig | char(4), |
| Grade | integer(4), |
| JoinDate | date, |
| Basic | integer(7)), |
| Gender | char(1), |
| DeptCode | char(4)) |

create table DEPT(

| | |
|-----------------|------------|
| <u>DeptCode</u> | char(4), |
| DeptName | char(30), |
| Location | char(10)) |

Integrity Rules

Entity Integrity: Implemented through Primary Key

“No Attribute participating in the primary key of a relation may accept null values”

Guarantees that each tuple will have a unique identity.

Referential Integrity: Implemented through Foreign Key

“Values of the foreign key (a) must be either null, or (b) if non-null, must match with the primary key value of some tuple of the ‘parent’ relation. The reference can be to the same relation”

*Foreign Key is also known as Reference/Referential key.