**1.Introduction:**A database is a collection of 1 or more ‘relations’, where each relation is a table with rows and columns. A relation consists of

a)Relation Schema.

b)Relation Instance.

**A)Relation Schema:**The schema specifies the relation’s name, the name of each field (column, attribute) and the ‘domain’ of each field.A domain is referred to in a relation schema by the domain name and has a set of associated values.



Example:

Students (Sid: string, name: string, login: string, age: integer, gross: real)

**B)Relation Instance:**



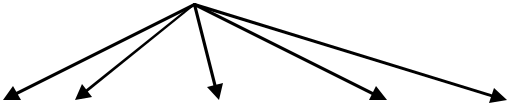
An instance of a relation is a set of Unique ‘tuples’, also called ‘records’, in which each tuple has the same number of fields as the relation schemas.

A relation instance can be thought of as a table in which each tuple is a row and all rows have the same number of fields.

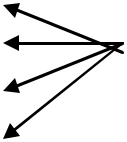
The relation instance is also called as ‘relation’.

**Example:**

Fields (Attributes, Columns)



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| sid | Name |  | login | | | |  | age | Gross | Field names |
| 1111 | Dave |  | dave@cs |  | | |  | 19 | 1.2 | Tuples (Records, Rows) |
|  |  |  |  | | |  |  |  |  |
| 2222 | Jones |  | Jones@cs | | | |  | 18 | 2.3 |
|  |  |  |  | |  | |  |  |  |  |
| 333 | Smith |  | smith@ee | | | |  | 18 | 3.4 |  |
|  |  |  |  | | |  |  |  |  |  |
| 4444 | Smith |  | smith@math | | | |  | 19 | 4.5 |  |
|  |  |  |  |  |  |  |  |  |  |  |



This example is an instance of the students relation, which consists 4 tuples and 5 fields. No two rows are identical.



Degree:The number of fields is called as ‘degree’.

This is also called as ‘arity’.



Cardinality:The cardinality of a relation instance is the number of tuples in it.

Example:

In the above example, the degree of the relation is 5 and the cardinality is 4.



Relational database:It is a collection of relations with distinct relation names.

2.**Constraints over Relations**:An integrity constraint (IC) is a condition that is specified on a relation schema and restricts the data can be stored in an instance of the relation. The constraints can be classified into 3 types as below.

1.Domain Constraints.

2.Entity Integrity Constraints.

3.Referential Integrity Constraints.

2.1) Domain constraints:-A domain is defined as the set of all unique values permitted for an attribute.

Ex:-

1. Domain of integer is all possible whole numbers.
2. Domain of day-of-week is Monday,Tuesday,-----Sunday.
3. Domain of date is set of all possible valid dates.

Usually, domain of field is type of field.

Domain Integrity constraints are mainly two types:

1.NOT NULL

2.CHECK constraints.

1.NOT NULL:- It restricts a column from having the null value. Once NOT NULL constraint is applied to column ,you can not pass null value to that column.

Relation Schema: Student(sid:number not null, name:varchar2,age:number)

Query in SQL: sql> create table student(sid number(3) NOT NULL , name varchar2(8),age number(2));

Sid column will not take the null value.

2.Check:- Check constraint is used to restrict the value of column between range. It is like condition checking before saving data into column.

Relation Schema: Student(sid:number not null check(sid>1),name :varchar3(8),age: number(2))

Query in SQL:Sql> create table student(sid number(2)NOT NULL CHECK(sid>1),name varchar2(8),age number(4));

This query will restrict sid value to be greater than zero.

2.2) Entity Integrity Constraints:-The entity integrity constraints are mainly two types,

1.Unique constraint.

2.Primary key Constraint.

1. Unique Constraint:- A unique constraint field will not have duplicate data.Unique constraint field will take the null value.

SQL Query: create table student(sid number(3)Unique key,name varchar2(7));

Relation Schema:student(sid: number Unique key,name: string)

1. Primary key:- primary key constraint uniquely identifies each record in database. A primary key allows unique value and itdoes not accept null value.

SQL Query:- Create table student(sid number(3)primary key ,name varchar2(7));

Relation Schema:student(sid: number primary key,name :string);

The above query create primary key on sid column.

2.3) Referential Integrity Constraints:-

The referential integrity constraint is specified between two tables .

Foreign key constraint:-A Foreign key is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) in another relation. The foreign key is used to establish and enfore link between the data in two relations to control data that can be stored in the foreign key.The table with the foreign key is called the child table(referencing), and the table with the primary key is called the referenced or parent table.

Syntax:

Create table childtable\_name(

Col1 definition,

Col2 definition,

----

Foreign key(col\_name)references parenttable\_name(primary key of parent\_table)

[On delete cascade|null]

);

Example:

Create table student(roll\_no number(2) primary key,

Sname varchar2(10),

Dob date,

Age number(2),

M\_no number(11));

Create table address(roll\_no number(2) primary key,

Village varchar2(5),

Mandal varchar2(5),

District varchar2(10),

Pin\_code number(6),

Foreign key (roll\_no)references student(roll\_no));

Note:- Without student record in student table, we can’t keep student address in address table.

3.3) **Refential Integrity constraints**:

Create table student(roll\_no number(2) primary key,

Sname varchar2(10),

Dob date,

Age number(2),

M\_no number(11));

Create table address(roll\_no number(2) primary key,

Village varchar2(5),

Mandal varchar2(5),

District varchar2(10),

Pin\_code number(6),

Foreign key (roll\_no)references student(roll\_no));

Where student is referenced Entity set(table).

Address is referencing Entity set(table).

**Insertion,Deletioin and udpdation on Referencing Entity set(Address):**

1. Sql> insert into address values(7,’vav’,’vidavalur’,’nlr’,520001);

The above insertion violates the referential integrity constraint. Because there is no student tuple with roll\_no 7 in student table.

1. Deletion does not violates the referential integrity constraint.
2. Sql > update table student set roll\_no=2 where roll=8;

The above updation violates referential integrity constraint Because there is no student tuple with roll\_no 2 in student table.

**Insertion,Deletion and udpdation on Referenced Entity set(student):**

1. Insertion on Referenced entity set does not violates referential integrity constraint.
2. Sql > delete from student where roll\_no=2;

Case 1:

If student with id 2 has already not enrolled for courses, then deletion does not violates the referential integrity constraint.

Case 2:

If student with id 2 has already enrolled for courses,then deletion violates referential integrity constraint.

In this case , There are 4 options

Option1: Delete all address rows that refers to the deleted students rows.

Option2: Disallow the deletion.

Option3 : set the roll\_no column to roll\_no of some(existing) ‘default’ student, for every address row that refers to the deleted students now..

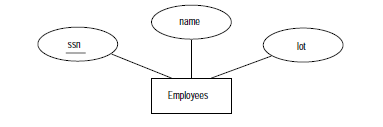
Option4: For every address row that refers to it, set the roll\_no column to null.

1. Updation may violates referential integrity constraint.

5.**Logical DB Design:ER to Relational:**We now describe how to translate an ER diagram into are collection of tables with associated constraints, i.e., a relational database schema.

5.1) Entity Set to Tables:-

* + The Entityset is mapped into Relation.
  + The attributes becomes fields of Relation.
  + We define the constraints in Relation schema.



Relational Schema:

Employees(ssn:char,name:String,lot:number,primary key(ssn))

(or)

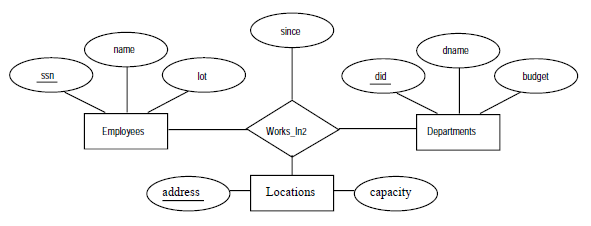
Employees(ssn:char primary key,name:string,lot:number)

5.2) Relationship set to Tables(Cardinality is M-M):

The Relationship set is mapped to relation in relation model. The relation will contain following attributes:

* Primarykey of each participating entity set.
* Descriptive attributes.

The set of non-descriptive attributes becomes a primary key. Every non-descriptive attributes refers the primary key in Referenced table.



**Relation Schema:-**

Works-In2(ssn:char,

did:number,

address:string,

since:date,

Primary key(ssn,did,address),

Foreign key(ssn)references employees(ssn),

foreign key(address)references locations(address),

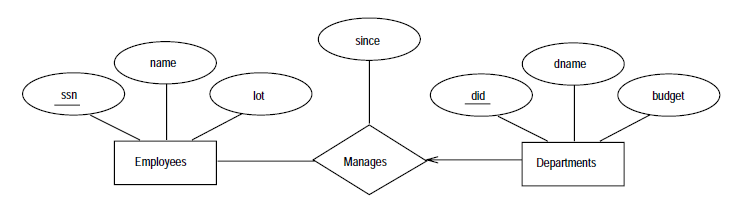
foreign key(did) references departments(did))

5.3)Relationship set to Tables (cardinality 1-M):

Approach1: The Relationship set is mapped to relation in relation model. The relation will contain following attributes:

1. Primary key of each participating entity set.
2. Descriptive attributes.

Every non-descriptive attribute refers to primary key of participating entity set. Define field of Entity set which has cardinality 1 as primary key.



Approach2:

Draw backs in Approach1:-

* We should create new table for relationship set.
* To get appropriate Information, we may write queries on more than one table. So query execution becomes slowness.

In second approach, we don’t need to create separate table for relationship set. The descriptive attributes of relationship set and primary key of another entity set is kept in the table corresponding to entity set with key.

Relation Schema: Dept\_mgr( did: number,

Dname: String,

Budget :real,

Ssn :String,

Since: date,

Primary key(did),

Foreign key(ssn)references employees)

Drawback in second approach:

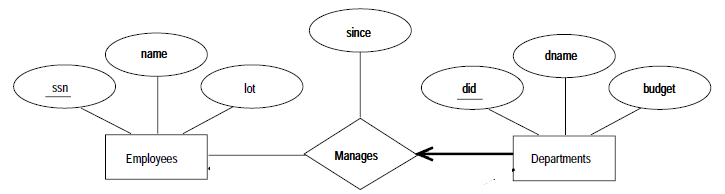
---------------------------------------

1. Space is wasted.

EX: if department has not manager, SSN stores NULL values.

5.4) Translating Relationship sets with participation constraints:-

The descriptive attributes of relationship set and primary key of another entity set is kept in the table corresponding to entity set with key. Define the NOT NULL constraint in every non-descriptive field definition of Total participation Entity set.



RelationSchema:Department(did:numberPrimarykey,

dname:string,

budget: number,

since:date,

ssn:number not null,

foreign\_key(ssn) References employee)

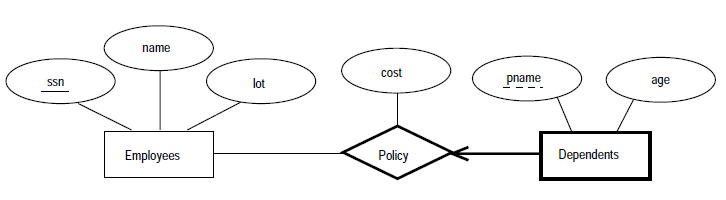
5.5) Translating weak Entity sets:-We create single relation for both weak relationship set and weak entity set.

It contains following attributes.

* All attributes of weak entity set.
* All descriptive attributes.
* Primary key of strong entity set.

Primary key of strong entity set and partial key together forms primary key for new table(Relation).

Foreign key refers to primary key of strong entity set.

RelationSchema:Dep\_policy(cost:number,

pname:string,

age:number,

ssn:number,

primary key(pname,ssn),

Foreign key(ssn) references employees)