Name – Akshay Patwal

father's Name – Mohan Singh Patwal

University Roll No. – 2015147

Subject – Database Management Systems

Subject Code – TCS 503

Sem – 5th

Course – B.Tech (DS & AI)

Type – Regular

Akshay

(a) Keys can be used to reduce data redundancy in database. (foreign key) & (essentially primary key)

Data redundancy leads to data anomalies and data corruption and should be avoided when creating a 'Relational database consisting of several entities. Database normalization prevents redundancy and makes the best possible useage of storage.

In order to minimize redundancy while designing database, we put data across multiple tables.

Essentially, primary & foreign keys are used as a way to constraint or link Related data in database. This ensures that data remains consistent & that the database contains no reductant data.

eg → If you delete a table (or even a row in a table) that other table rely on, the reductant data is removed.

Akshay

database contains no redundant data.

No data redundancy can't be completely eliminated when the database approach is used. It cannot be totally removed from the database, though there needs to be controlled redundancy to reduce chances of destructive anomalies.

i) Super key is an attribute that is used to uniquely identify all attributes in a relation. All super keys are not candidate keys but all candidate keys are super keys. This is because candidate keys can be NULL.

ii) Both primary keys and candidate keys are used to get records from table. These keys are also used to create relationship between tables. All primary keys are candidate keys as we know that Primary key is minimal super key and a super key can be candidate key.

iii) Primary key = 0
As in all ~~four~~ four rows / in table every column has a repeated value.

∴ Candidate key = 0
Super key = 3 (As super key is a group of single or multiple key which is identity row in table.)

—Akshay

1) Two-Tier Database Architecture

Here, the application logic is either burried inside the user interface on the client or within the database on the server (or both). The user system interface is usually located in the user's desktop environment & dbms services are usually in a server that is a more powerful machine that services many clients.

2) Three-Tier Database Architecture

The application logic lives in the middle-tier, it is separated from the data and the user interface. They are more robust, scalable & flexible and can integrate data from multiple sources. In three tier architecture, a middle tier was added in between the user system interface client environment and dbmserver environment.

Two-tier Database Architecture	Three-tier Database Architecture
1) Client-Server Architecture.	Web based application.
2) Consists of two layers: client Tier & Data Tier	Consists of three layers: Client layer, Business layer & data layer.
3) Easy to build and maintain.	Complex to build & maintain.
4) Runs slower.	Runs faster.

Akshay

5) Less secure, as direct interaction.

More secured as client can't communicate with database.

6) Performance loss with increase in users rapidly.

Better than 2-tier architecture in performance.

7) Example:

Contact Management System created using MS-Access or Railway Reservation System.

Example:

Designing registration form which contains tent, bou, label or a large website.

1) Total Participation:-

•) It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.

•) That is why, it is also called as mandatory participation.

•) Total participation is represented using a double line between the entity set and relationship set.

```
  /\          _____
 /  \        |        |
<  R  >======|   E    |
 \  /        |_____|
  \/
```
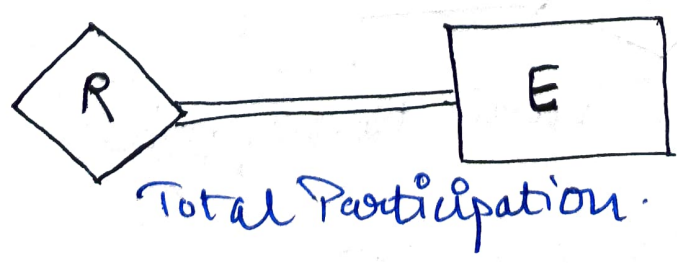
Total Participation.

Hes'

Example :

```
┌─────────┐           ╱◇╲
│ student │═════════╱Enrolled╲───────────┌────────┐
└─────────┘         ╲   in   ╱            │ Course │
                      ╲◇╱                 └────────┘
```
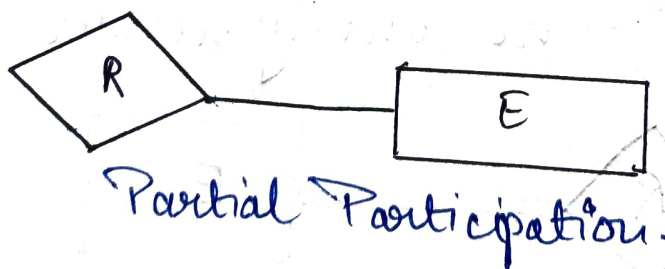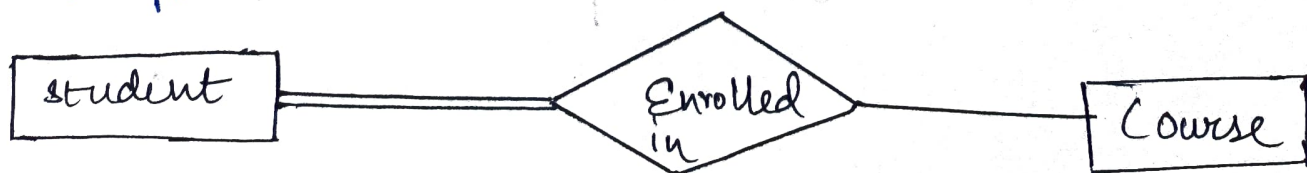
Here,

- •) Double line between the entity set "student" & relationship set "Enrolled in" signifies total participation.
- •) It specifies that each student must be enrolled in at least one course.

②) Partial Participation :-

- •) It specifies that each entity set may or may not participate in the relationship instance in that relationship set.

- •) That is why, it is also called as optional participation.

- •) Partial participation is represent using a single line between the entity set and relationship set.

```
        ╱◇╲
      ╱  R  ╲
      ╲     ╱────────┌─────────┐
        ╲◇╱          │    E    │
                     └─────────┘
        Partial Participation.
```

Example :

```
┌─────────┐              ╱◇╲
│ student │───────────╱Enrolled╲──────────┌────────┐
└─────────┘           ╲   in   ╱          │ Course │
                        ╲◇╱               └────────┘
```

Akshay

Here,

.) single line between the entity set "course" and relationship set "Enrolled in" signifies partial participation.

.) It specifies that there might exist some courses for which no enrollments are made.

Ahsly