Unit - II Relational Data Model

RDBMS

12 Marks

Syllabus Details

2.1 Relational Structure :- Tables (Relations), Rows (Tuples), Domains, Attributes, Entities

2.2 Keys :- Super Keys, Candidate Key, Primary Key, Foreign Key.

2.3 Data Constraints :- Domain Constraints ,Referential Integrity Constraints

2.4 Entity Relationship Model : - Strong Entity set, Weak Entity set, Types of Attributes, Symbols for ER diagram, ER Diagrams

2.5 Normalization:- Functional dependencies, Normal forms: 1NF, 2NF, 3NF

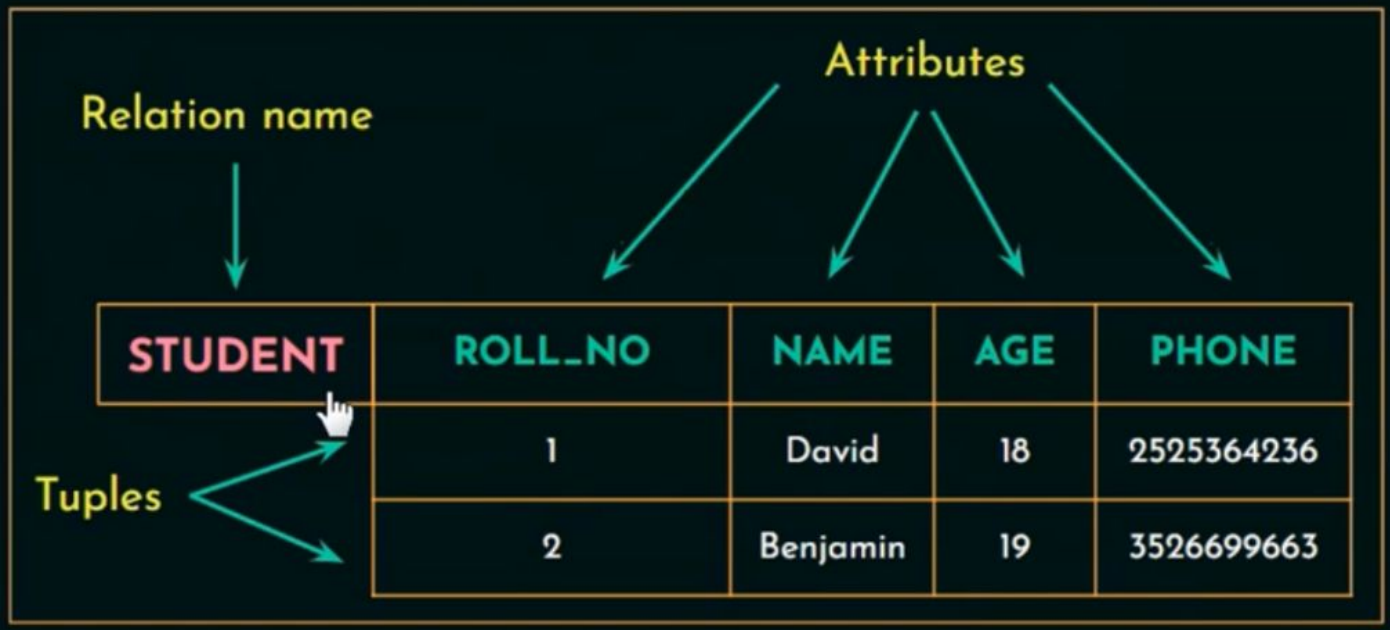
❖ Terminologies

★ Relational model represents data as a collection of tables.

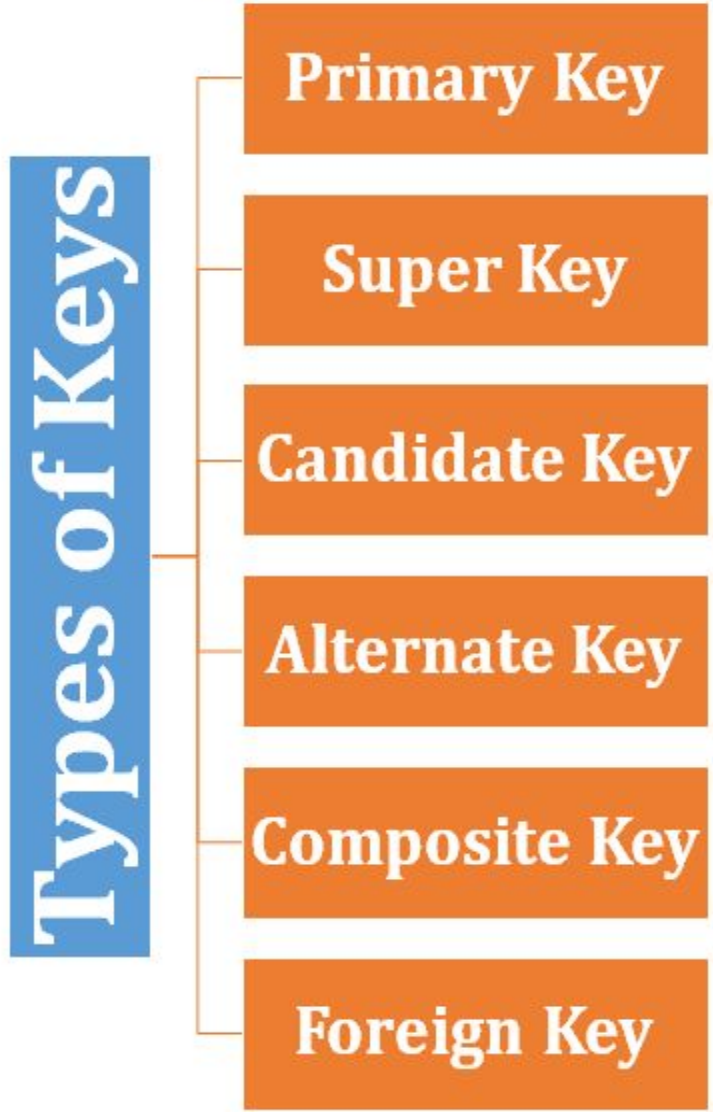
★ A table is also called a relation.

★ Each row → tuple.

★ Column headers→ attributes.

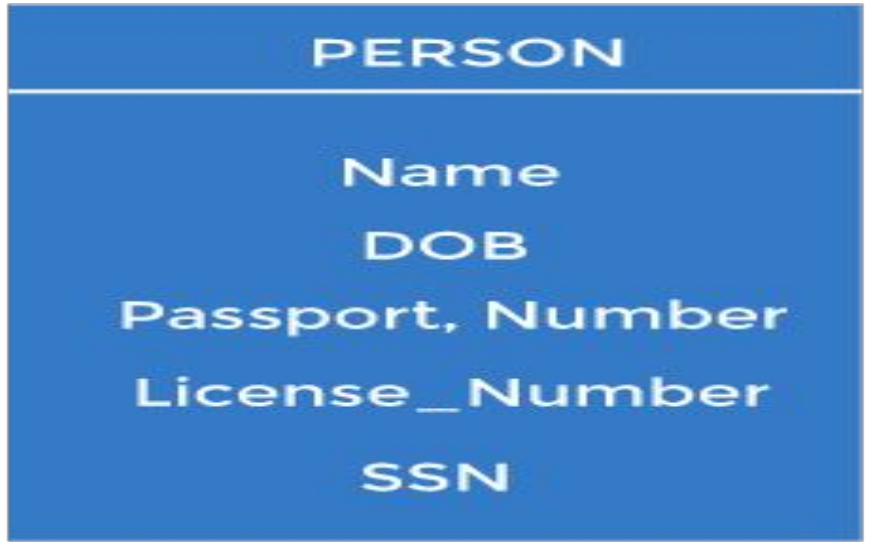


Key Concepts



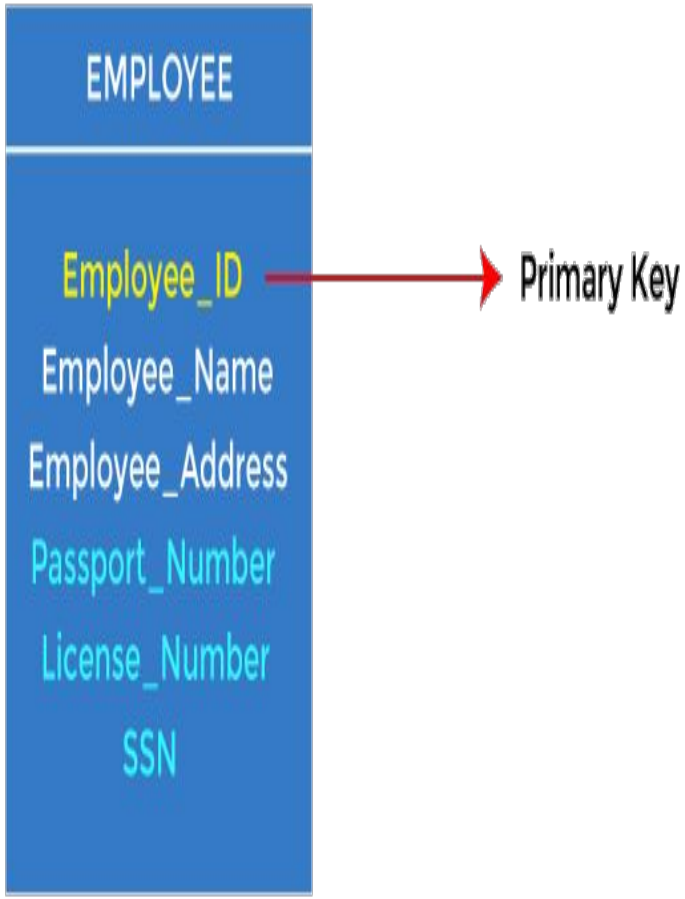
- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

For example, ID is used as a key in the Student table because it is unique for each student. In the PERSON table, passport_number, license_number, SSN are keys since they are unique for each person. **Social Security Number**



1. Primary key

- It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.
- Primary Key cannot be a null value and should be unique
- In the EMPLOYEE table, ID can be the primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.
- For each entity, the primary key selection is based on requirements and developers.

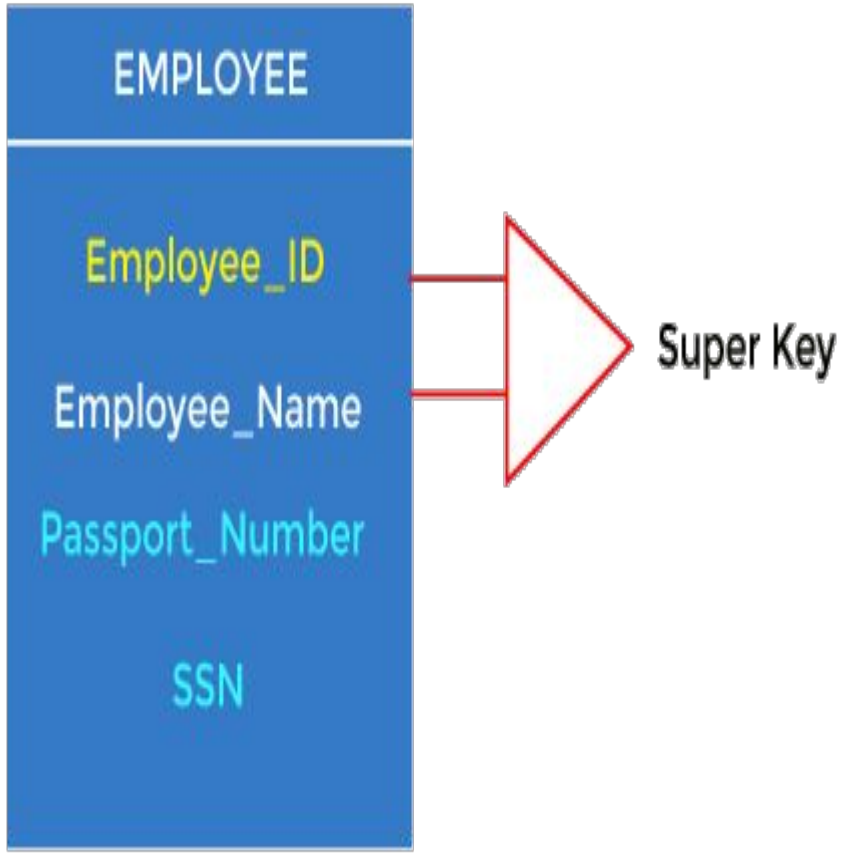


2. Candidate key

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.
- **For example:** In the **EMPLOYEE table**, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.

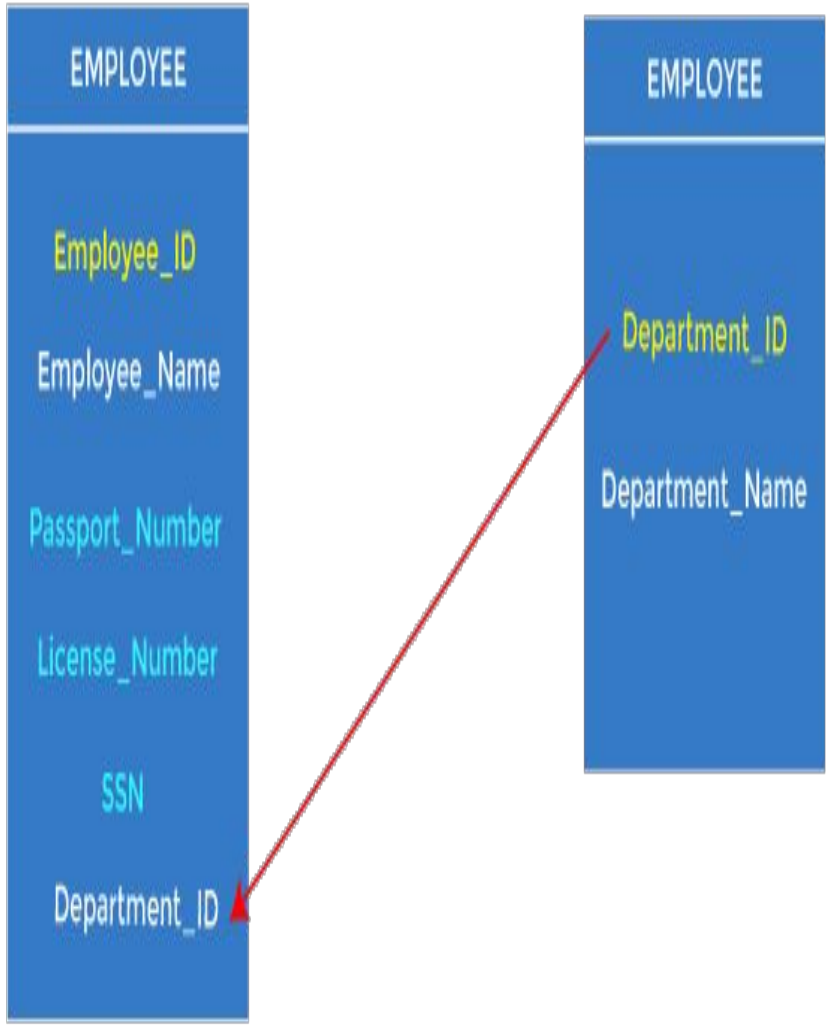
3. Super Key

- Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.
- **For example:** In the EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME), the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key
- The super key would be EMPLOYEE-ID (EMPLOYEE_ID, EMPLOYEE-NAME), etc.



4. Foreign key

- Foreign keys are the column of the table used to point to the primary key of another table.
- Every employee works in a specific department in a company, and employee and department are two different entities. So we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.
- We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.
- In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.



DBMS

- DBMS stores data as file
- Data elements need to access individually.
- No relationship between data
- Normalization is not present
- DBMS does not support distributed database.
- It deals with small quantity of data.
- Data redundancy is common in this model.
- It supports single user.
- Low software and hardware necessities.
- Eg:-Examples: XML, Window Registry, etc.

RDBMS

- RDBMS stores data in tabular form.
- Multiple data elements can be accessed at the same time.
- Data is stored in the form of tables which are related to each other
- Normalization is present.
- RDBMS supports distributed database.
- It deals with large amount of data.
- Keys and indexes do not allow Data redundancy.
- It supports multiple users.
- Higher software and hardware necessities.
- Eg;-Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc.

Data Types in SQL

1.CHAR (length)

2.VARCHAR (length)

3.BOOLEAN

4.SMALLINT

5.INTEGER or INT

6.DECIMAL or DEC

7.NUMERIC

8.FLOAT

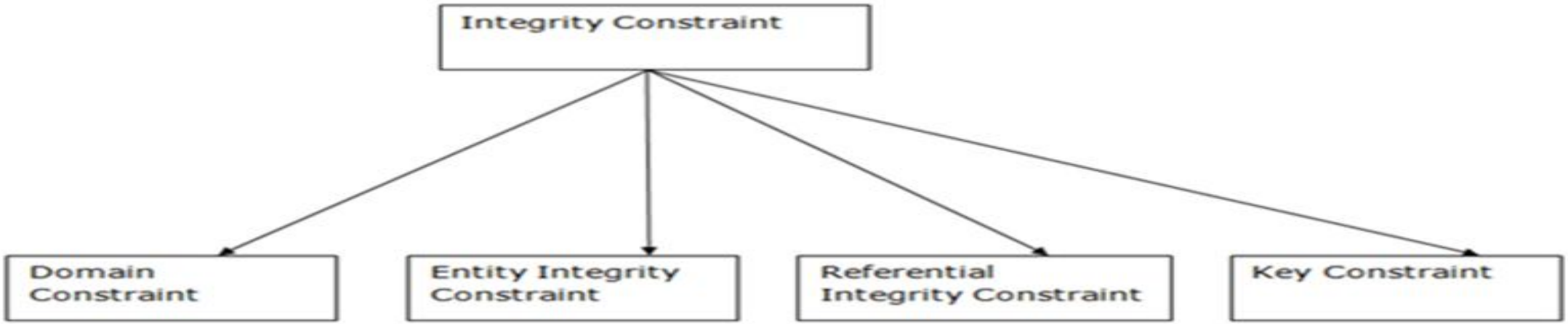
9.DATE

10.TIME

Integrity Constraints

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to guard against accidental damage to the database.

Types of Integrity Constraint



1. Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

Example:

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

- The entity integrity constraint states that primary key value can't be null.
- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.

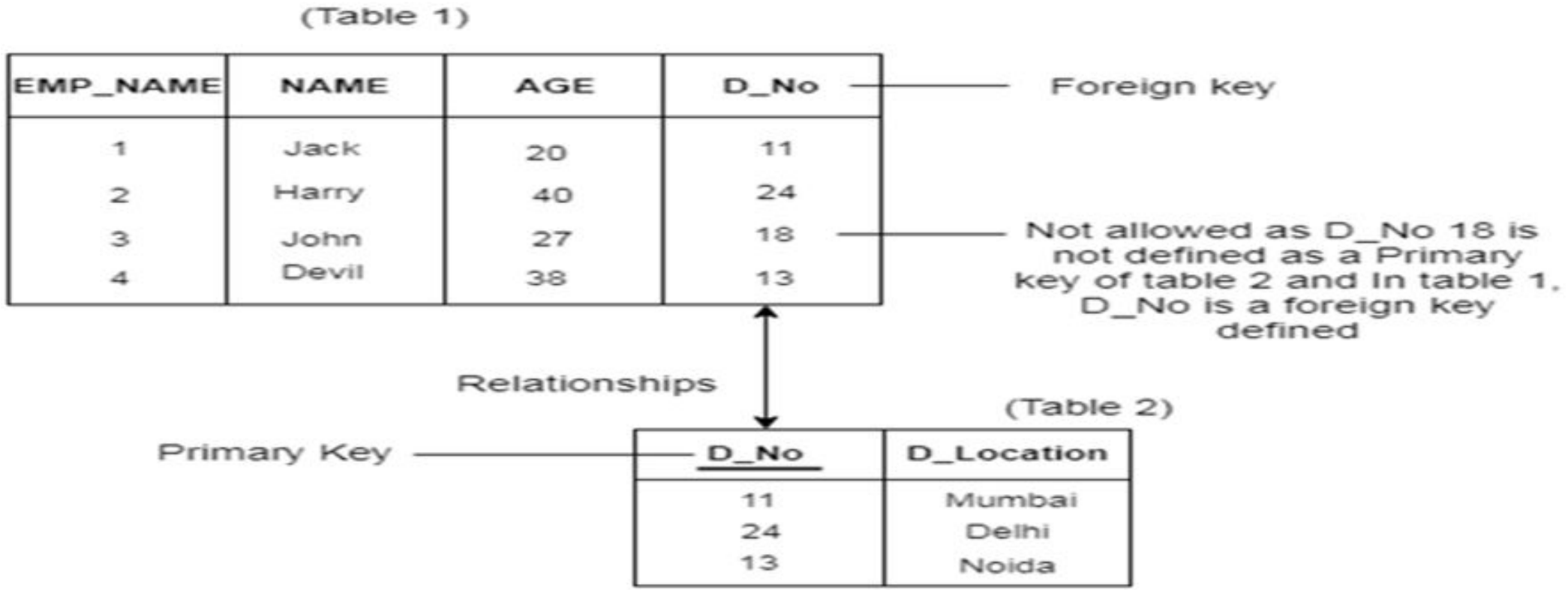
EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

3. Referential Integrity Constraints

- A referential integrity constraint is specified between two tables.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.



4. Key constraints

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

Draw ER diagram for Car Insurance Company that has a set of customer each of whose having one or more car. Each car associated with it zero to any number of record accidents

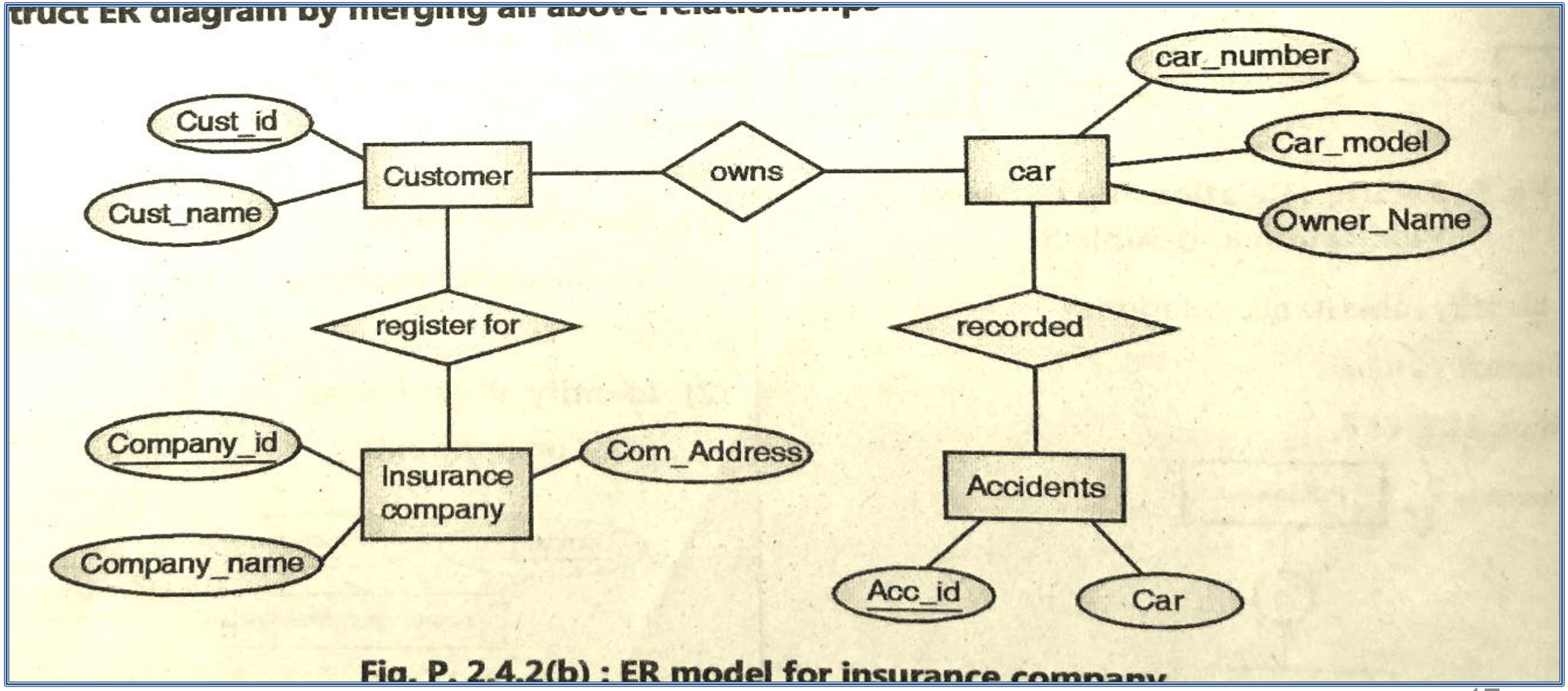
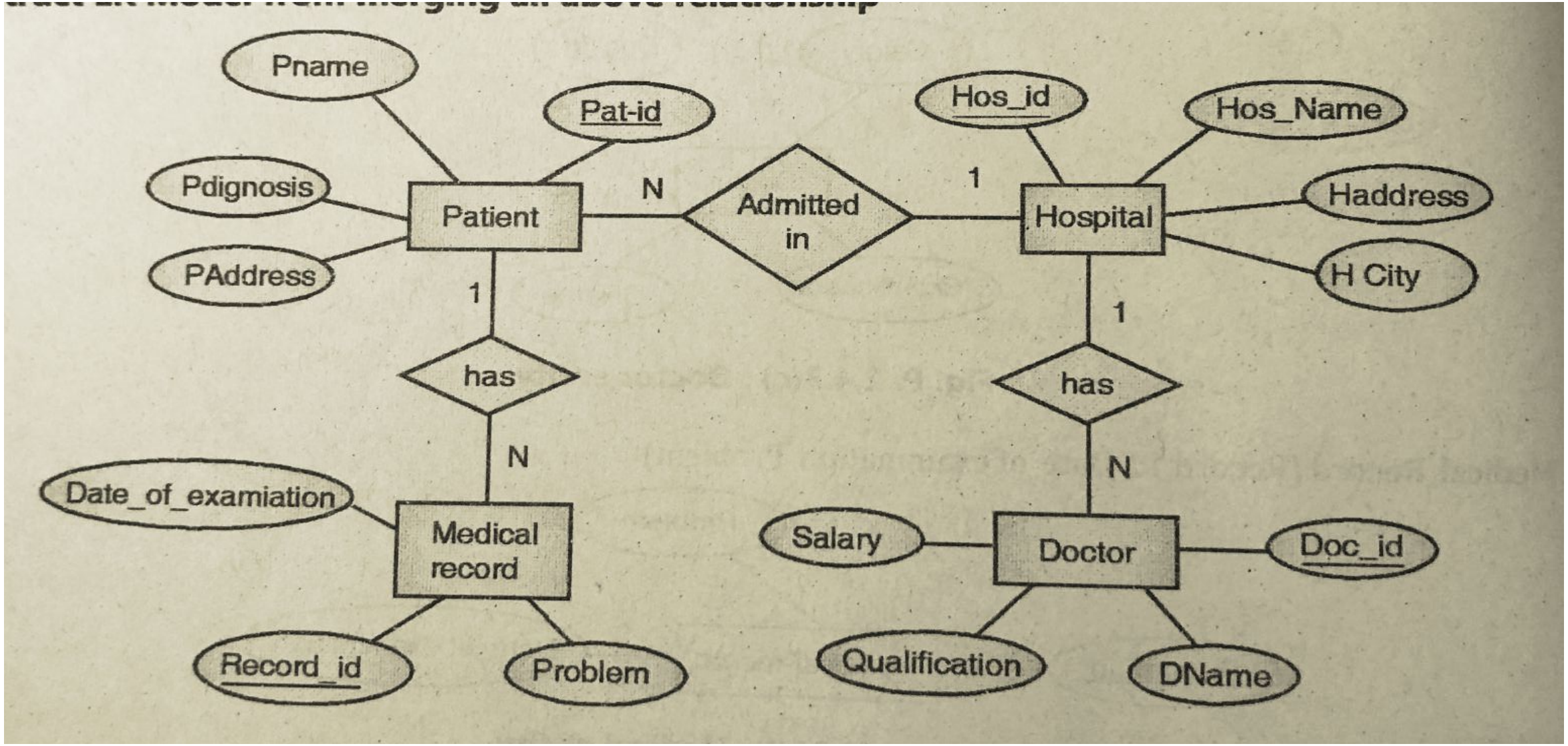


Fig. P. 2.4.2(b) : ER model for insurance company

Draw ER diagram for hospital with a set of patients and the set of medical doctors associated with each patient a record of various text and examination conducted



Domain:

- A domain is defined as the set of all unique values permitted for an attribute.The domain of a database attribute is the set of all attribute values for that attribute.

For example:

TableName: STUDENT

RollNumber	StudentName	StudentDepartment	StudentGender
0123	Harshita	CSE	Female
0124	Kunal	Civil	Male

- A field/Column for gender may have the domain(male,female) where those three values are the only permitted entries in that column.
- A domain of date is the set of all possible entries in that column.
- A domain of day of week is monday ,tuesday-saturday.

Normalization:

If a database design is not perfect, it may contain anomalies, which are like a bad dream for any database administrator. Managing a database with anomalies is next to impossible.

- **Update anomalies:** If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when we try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values. Such instances leave the database in an inconsistent state.

- **Deletion anomalies:** We tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.

- **Insert anomalies:** We tried to insert data in a record that does not exist at all.

Normalization is a method to remove all these anomalies and bring the database to a consistent state.

Normalization

- **Normalization-** It is a database design technique which is used to organize the tables in such a manner that it should **reduce redundancy and dependency** of data.

- **Types of Normalization-**

First Normal Form

- First Normal Form is defined in the definition of relations (tables) itself.
- This rule defines that all the attributes in a relation must have atomic domains.
- The values in an atomic domain are indivisible units.

Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

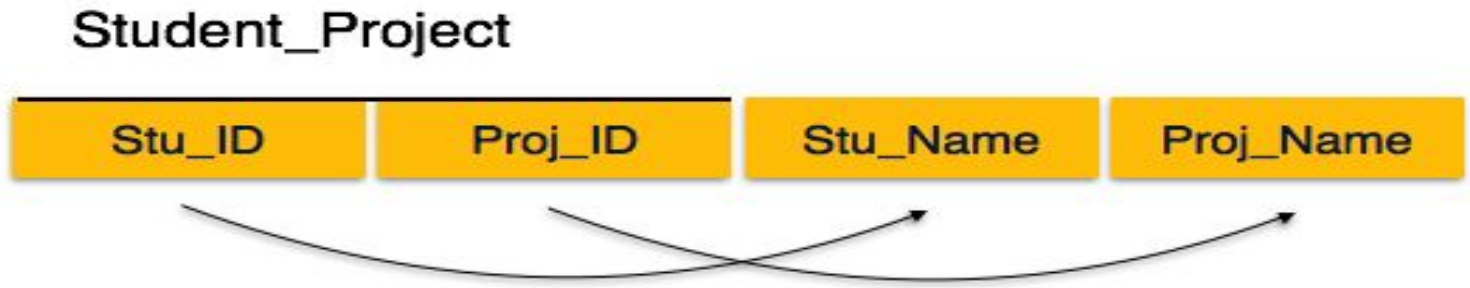
We rearrange the relation (table) as below, to convert it to First Normal Form.

- Each attribute must contain only a **single value** from its pre-defined domain.

Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

Second Normal Form

- Before we learn about the second normal form, we need to understand the following:
  - Prime attribute: An attribute, which is a part of the candidate-key, is known as a prime attribute.
  - Non-prime attribute: An attribute, which is not a part of the primary-key, is said to be a non-prime attribute.



We see here in Student_Project relation that the prime key attributes are Stu_ID and Proj_ID. According to the rule, non-key attributes, i.e. Stu_Name and Proj_Name must be dependent upon both and not on any of the prime key attribute individually.
But we find that Stu_Name can be identified by Stu_ID and Proj_Name can be identified by Proj_ID independently.
This is called partial dependency, which is not allowed in Second Normal Form.

Second Normal Form

- We see here in Student_Project relation that the prime key attributes are Stu_ID and Proj_ID.
- According to the rule, non-key attributes, i.e. Stu_Name and Proj_Name must be dependent upon both and not on any of the prime key attribute individually.
- But we find that Stu_Name can be identified by Stu_ID and Proj_Name can be identified by Proj_ID independently.
- This is called partial dependency, which is not allowed in Second Normal Form.
- We broke the relation in two as depicted in the above picture. So there exists no partial
- dependency.

Third Normal Form

- For a relation to be in Third Normal Form, it must be in Second Normal form
- We find that in the above Student_Detail relation, Stu_ID is the key and only prime key attribute.
- We find that City can be identified by Stu_ID as well as Zip itself.
- Neither Zip is a super key nor is City a prime attribute.
- Additionally, Stu_ID → Zip → City, so there exists transitive dependency.
- To bring this relation into third normal form, we break the relation into two relations
- as follows –