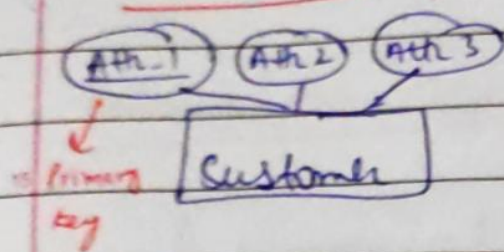


Section 7

Conversion of ER Diagrams to Table



Converts to

| Attr-1 | Attr-2 | Attr-3 |
|--------|--------|--------|
| | | |

Entity Set

Relational Table

Entity Set converts to table

Attributes converts to columns of table

For every simple attribute, create a column of the table

The key attribute will become primary key of the table

However, composite attribute, multi-valued attribute and derived attribute need special treatment.

Handling of Composite Attribute

Composite attributes are those which can be split into components

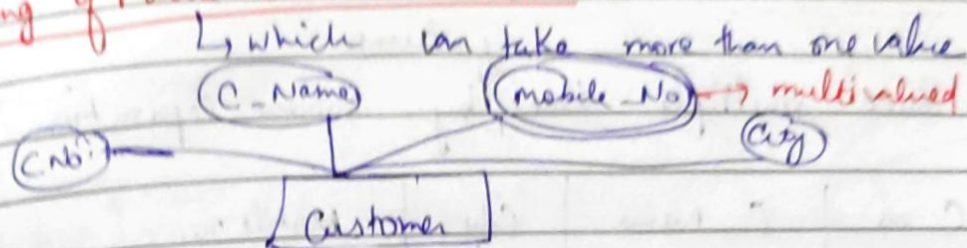
For eg: name as first_name and last_name

In order to convert an entity set to table, the composite attributes are flattened out by creating a separate attribute for each component of composite attribute

Create the columns for branches and ignore the root.

2 separate columns will be created for first_name & last_name and no need of creating name column

Handling of Multivalued Attributes



Possible Solution

Solution 1

→ Create a column for each possible value of attribute for a second

| C.No. | C_Name | City | Mobile No. 1 | Mobile No. 2 | Mobile No. 3 |
|-------|--------|------|--------------|--------------|--------------|
| | | | 321 | 156 | 759 |

Problems with this Solution

- 1) Various operations like retrieval, updation, deletion of data becomes complex
- 2) There will be many rows with NULL values
- 3) Simplicity of relational model is lost

eg If we want to mobile_no - 759, then we have to check all 3 columns to find 759.

```
SELECT Customer_name  
FROM customer
```

```
WHERE
```

```
mobile_no. 1 = 759
```

```
OR
```

```
mobile_no. 2 = 759
```

```
OR
```

```
mobile_no. 3 = 759
```


Solution 2

→ Store multiple values ^{of column} by comma separation

| C no. | C Name | City | Mobile no. |
|-------|--------|------|---------------|
| | | | 321, 193, 367 |

↓
Since there are commas
⇒ Not an integer

Problem

→ Integrity of data is lost

★

Solution - 3

correct

→ Create separate table for multi value attribute
→ The primary key for this multivalued attribute table will be primary key of entity table + multivalued attribute

| eg | ^{Primary key} C-no. | cust name | City | ^{foreign key} C-no. | ^{Primary key} Mobile no. |
|----|---------------------------------|-----------|------------|---------------------------------|--------------------------------------|
| | 1 | A | Delhi | 1 | 868 |
| | 2 | B | Chennai | 1 | 754 |
| | 3 | C | Chandigarh | 2 | 392 |
| | | | | 3 | 869 |
| | | | | 3 | 985 |

Customers

Cust_Mobile_no.1

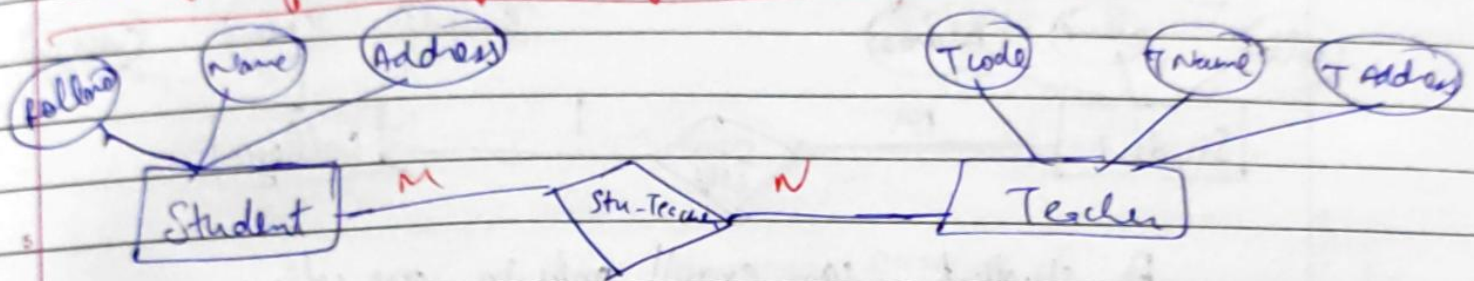
#

Handling of Derived Attribute

→ No need to create column for derived attribute
→ No need to create column for age, its value will be derived from DOB

→ SELECT (SYSDATE - DOB) / 365 AGE FROM Customers
 ↓ today's date ↓ Subtract ↓ to get no. of years

Handling of Many to Many Relationships



→ A student can have M Teachers

A Teacher can teach any no of Students

→ To depict this many to many relationship in table

→ If we add Tcode in student table, then soln will fail because a single student may have multiple teachers (similar to case of multiple mobile no.s)

→ So, the appropriate soln is

To create a separate table to depict the student-teacher relationship having rollno. & Tcode as its primary key

Rule

For every many to many type of relationship, create a separate table for relationship involving primary key of all participating entity sets

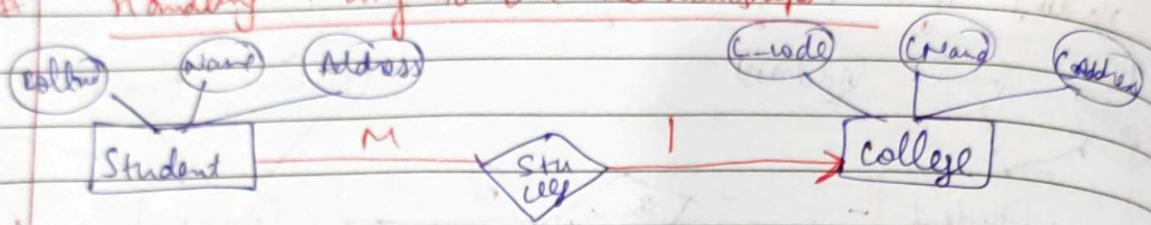
| Rollno | Name | Address |
|--------|------|---------|
|--------|------|---------|

| Rollno | T-code |
|--------|--------|
|--------|--------|

| Tcode | Tname | Taddress |
|-------|-------|----------|
|-------|-------|----------|

Foreign key

Handling Many to One Relationship



→ A student can enroll only in one college but a college can have any number of students

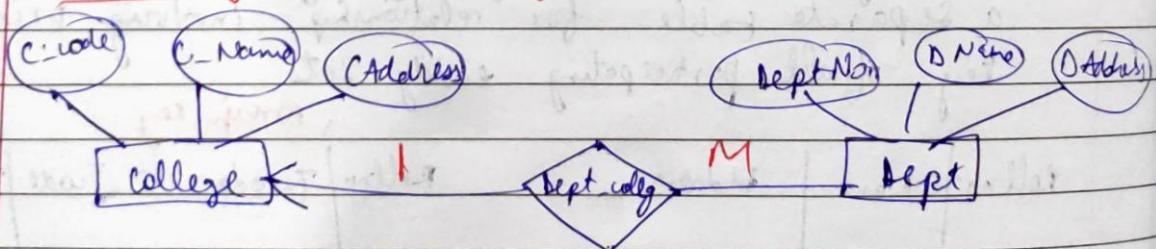
→ Rule:

- For every many to one type of relationship, there is no need to create a separate table for relationship
- Copy the primary key of one entity set towards many entity set

| | | | |
|---------|------|---------|--------|
| Rollno. | Name | Address | C-code |
| | | | |
| | | | |
| | | | |

| | | |
|--------|--------|-----------|
| C-code | C-Name | C-Address |
| | | |
| | | |
| | | |

Handling one to Many Relationship



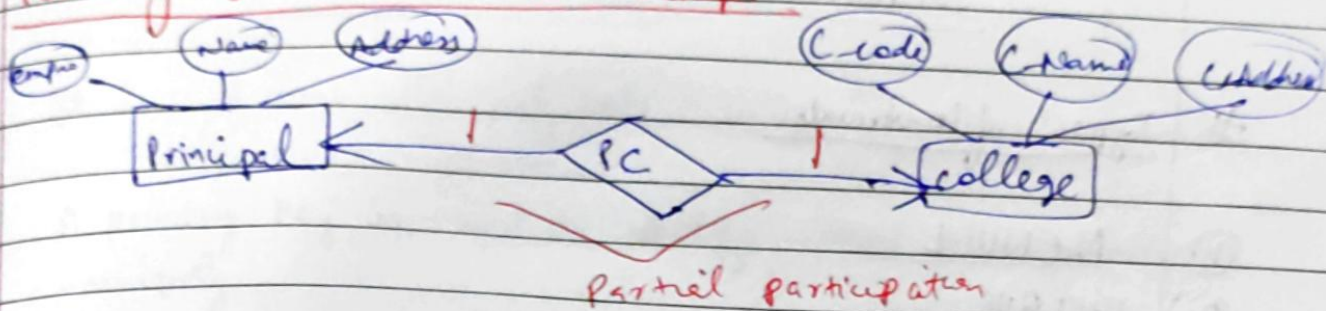
→ Every department belongs to one single college
one college can have many departments

→ Rule → Exactly same as Many to one relationship

| | | |
|--------|--------|-----------|
| C-code | C-Name | C-Address |
| | | |
| | | |
| | | |

| | | |
|---------|--------|-----------|
| Dept No | D-Name | D-Address |
| | | |
| | | |
| | | |

Handling one to one Relationships



→ Principle can administer only one college
one college can have only one Principle

→ Rule :

→ For one to one Relationship, there is no need to create a separate table for relationship

→ Copy the primary key of any of one entity set towards another entity set

→ There are 2 possible solⁿs.

either copy C-code in Principal table
or copy emp no. in college table

★★ Rule in case of one to one relationships having Partial-Total Participation



→ If there is an entity set with total participation, always copy primary key of one entity set towards an entity set having total participation

→ one value k_i primary key total value m copy k_{m+1}