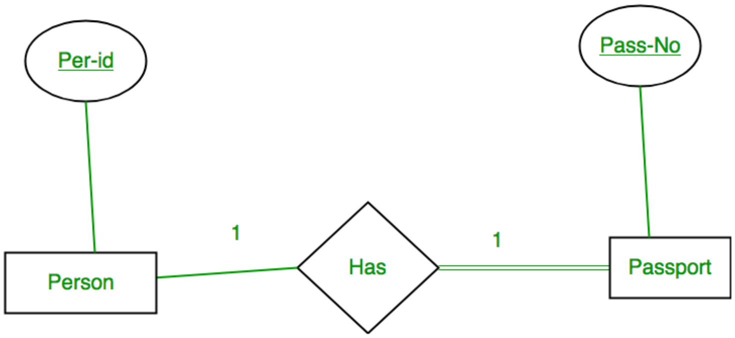
Mapping from ER Model to Relational Model

After designing the ER diagram of system, we need to convert it to Relational models.

Case 1: Binary Relationship with 1:1 cardinality with total participation of an entity



A person has 0 or 1 passport number and Passport is always owned by 1 person. So it is 1:1 cardinality with full participation constraint from Passport.

First Convert each entity and relationship to tables. Person table corresponds to Person Entity with key as Per-Id. Similarly Passport table corresponds to Passport Entity with key as Pass-No. Has Table represents relationship between Person and Passport (Which person has which passport). So it will take attribute Per-Id from Person and Pass-No from Passport.

Person	
Per-Id	**Other Person Attribute**
PR1	–
PR2	–
PR3	–

Has	
Per-Id	**Pass-No**
PR1	PS1
PR2	PS2

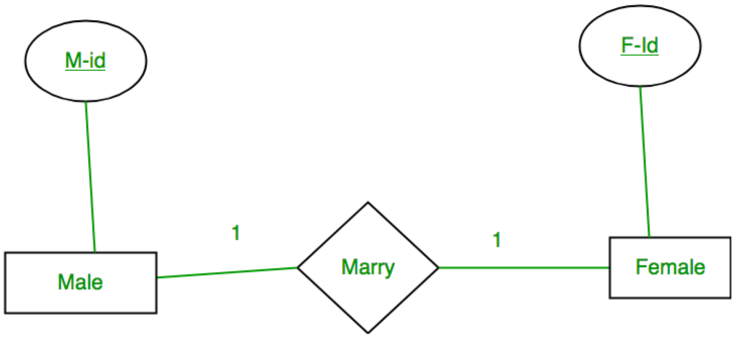
Passport	
Pass-No	**Other PassportAttribute**
PS1	–

PS2	–

As we can see from above Table, each **Per-Id** and **Pass-No** has only one entry in Has Table. So we can merge all three tables into 1 with attributes. Each Per-Id will be unique and not null. So it will be the key. Pass-No can't be key because for some person, it can be NULL.

Per-Id	Other Person Attribute	Pass-No	Other PassportAttribute

Case 2: Binary Relationship with 1:1 cardinality and partial participation of both entities



A male marries 0 or 1 female and vice versa as well. So it is 1:1 cardinality with partial participation constraint from both.

Male	
M-Id	Other Male Attribute
M1	–
M2	–
M3	–

Marry	
M-Id	F-Id
M1	F2
M2	F1

Female	
F-Id	Other FemaleAttribute
F1	–
F2	–
F3	–

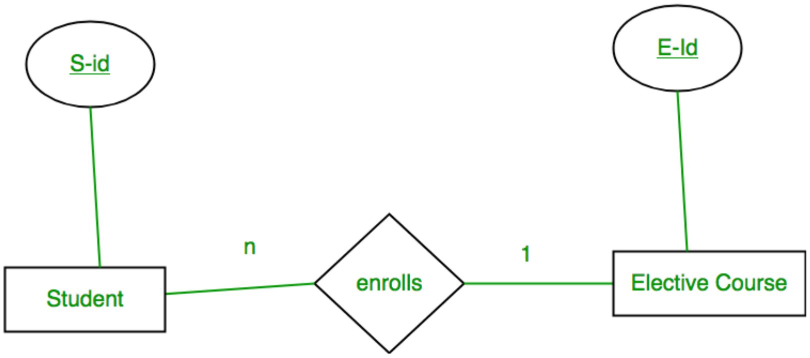
As we can see from above Table, some males and some females do not marry. If we merge 3 tables into 1, for some M-Id, F-Id will be NULL. So there is no attribute which is always not NULL. So we can't merge all three tables into 1. We can convert into 2 tables.

M-Id	Other Male Attribute	F-Id

F-Id	Other FemaleAttribute

Note: Binary relationship with 1:1 cardinality will have 2 table if partial participation of both entities in the relationship. If atleast 1 entity has total participation, number of tables required will be 1.

Case 3: Binary Relationship with n: 1 cardinality



In this scenario, every student can enroll only in one elective course but for an elective course there can be more than one student.

Student	
S-Id	Other Student Attribute
S1	–
S2	–
S3	–
S4	–

Enrolls	
S-Id	E-Id
S1	E1
S2	E2
S3	E1
S4	E1

Elective_Course	
E-Id	Other Elective CourseAttribute

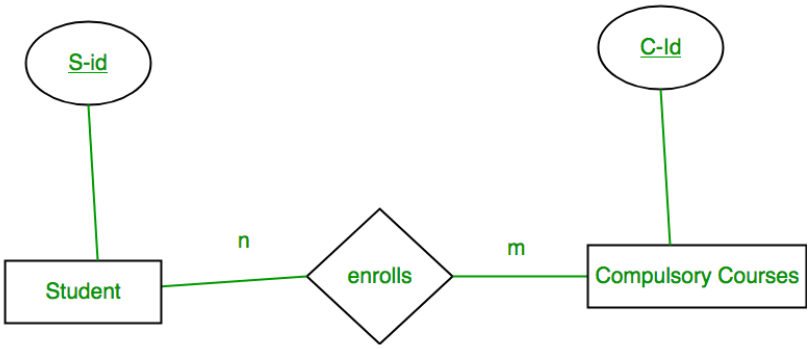
E1	–
E2	–
E3	–

As we can see from above Table, S-Id is not repeating in Enrolls Table. So it can be considered as a key of Enrolls table. Both Student and Enrolls Table's key is same; we can merge it as a single table.

S-Id	Other Student Attribute	E-Id

E-Id	Other Elective CourseAttribute

Case 4: Binary Relationship with m: n cardinality



In this scenario, every student can enroll in more than 1 compulsory course and for a compulsory course there can be more than 1 student.

Student	
S-Id	Other Student Attribute
S1	–
S2	–
S3	–
S4	–

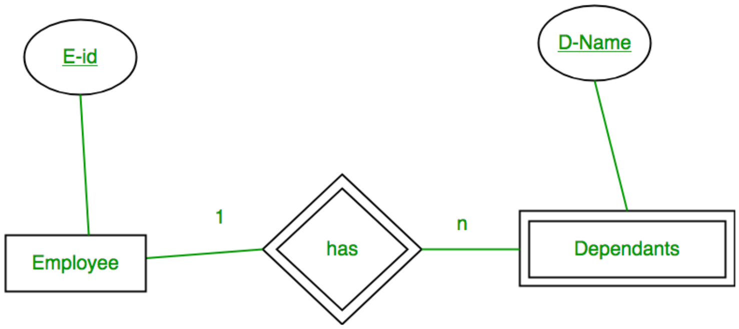
Enrolls	
S-Id	C-Id
S1	C1
S1	C2
S3	C1

S4	C3
S4	C2
S3	C3

Compulsory_Courses	
C-Id	Other Compulsory CourseAttribute
C1	–
C2	–
C3	–
C4	–

As we can see from above Table, S-Id and C-Id both are repeating in Enrolls Table. But its combination is unique; so it can be considered as a key of Enrolls table. All tables' keys are different, these can't be merged.

Case 5: Binary Relationship with weak entity



In this scenario, an employee can have many dependents and one dependent can depend on one employee. A dependent does not have any existence without an employee (e.g; you as a child can be dependent of your father in his company). So it will be a weak entity and its participation will always be total. Weak Entity does not have key of its own. So its key will be combination of key of its identifying entity (E-Id of Employee in this case) and its partial key (D-Name).

Employee	
E-Id	Other Employee Attribute
E1	–
E2	–
E3	–

Has	
E-Id	D-Name
E1	RAM
E1	SRINI
E2	RAM
E3	ASHISH

Dependents		
D-Name	E-Id	Other DependentsAttribute
RAM	E1	–
SRINI	E1	–
RAM	E2	–
ASHISH	E3	–

As we can see from above Table, E-Id, D-Name is key for **Has** as well as Dependents Table. So we can merge these two into 1.

E-Id	Other Employee Attribute

D-Name	E-Id	Other DependentsAttribute

Case 6: Mapping of Multivalued Attributes

1. **Identify the Multivalued Attribute**:
    - Determine which attribute of an entity can have multiple values.
    - Example: `Employee` entity has a multivalued attribute `PhoneNumbers`.
2. **Create a New Entity**:
    - Create a new entity to represent the multivalued attribute.
    - This new entity will hold the multiple values.
    - Example: Create a new entity called `EmployeePhoneNumbers`.
3. **Create a Relationship**:
    - Establish a relationship between the original entity and the new entity.
    - This relationship is typically one-to-many, where the original entity has one occurrence but can be associated with multiple occurrences of the new entity.
    - Example: Establish a one-to-many relationship between `Employee` and `EmployeePhoneNumbers`.
4. **Define Primary Keys**:
    - The primary key of the original entity will be a foreign key in the new entity to maintain the relationship.
    - The new entity will have its own primary key, which could be a composite key if needed.

- Example: The primary key of `Employee` (e.g., `EmployeeID`) will be a foreign key in `EmployeePhoneNumbers`. `PhoneNumber` is an attribute of `EmployeePhoneNumbers` which, combined with `EmployeeID`, forms a composite primary key if needed.

Case 7: Mapping of N-ary Relationship

1. **Identify the Participating Entities**:
    - Determine all the entities involved in the N-ary relationship.
    - Example: `Employee`, `Project`, `Department`.
2. **Create a New Relationship Table**:
    - Create a new table to represent the N-ary relationship.
    - The table will include foreign keys referencing the primary keys of all participating entities.
    - Example: Create a table `WorksOn` to represent the ternary relationship.
3. **Include Additional Attributes**:
    - If the relationship has any attributes, include these in the new table.
    - Example: `Hours` (representing the number of hours an employee works on a project in a department).
4. **Define Primary Key**:
    - The primary key of the new table is usually a composite key composed of the primary keys of the participating entities.
    - Example: The primary key of `WorksOn` could be the combination of `EmployeeID`, `ProjectID`, and `DepartmentID`.