Types of Relationships

a. One-One Relationship (1-1 Relationship)
b. One-Many Relationship (1-M Relationship)
c. Many-Many Relationship (M-M Relationship)

This tech-recipe covers only **1-1 and 1-M relationship**.

1. One-One Relationship (1-1 Relationship)

One-to-One (1-1) relationship is defined as the relationship between two tables where both the tables should be associated with each other based on only one matching row. This relationship can be created using **Primary key-Unique foreign key constraints**.

With One-to-One Relationship in SQL Server, for example, a person can have only one passport. Let's implement this in SQL Server.

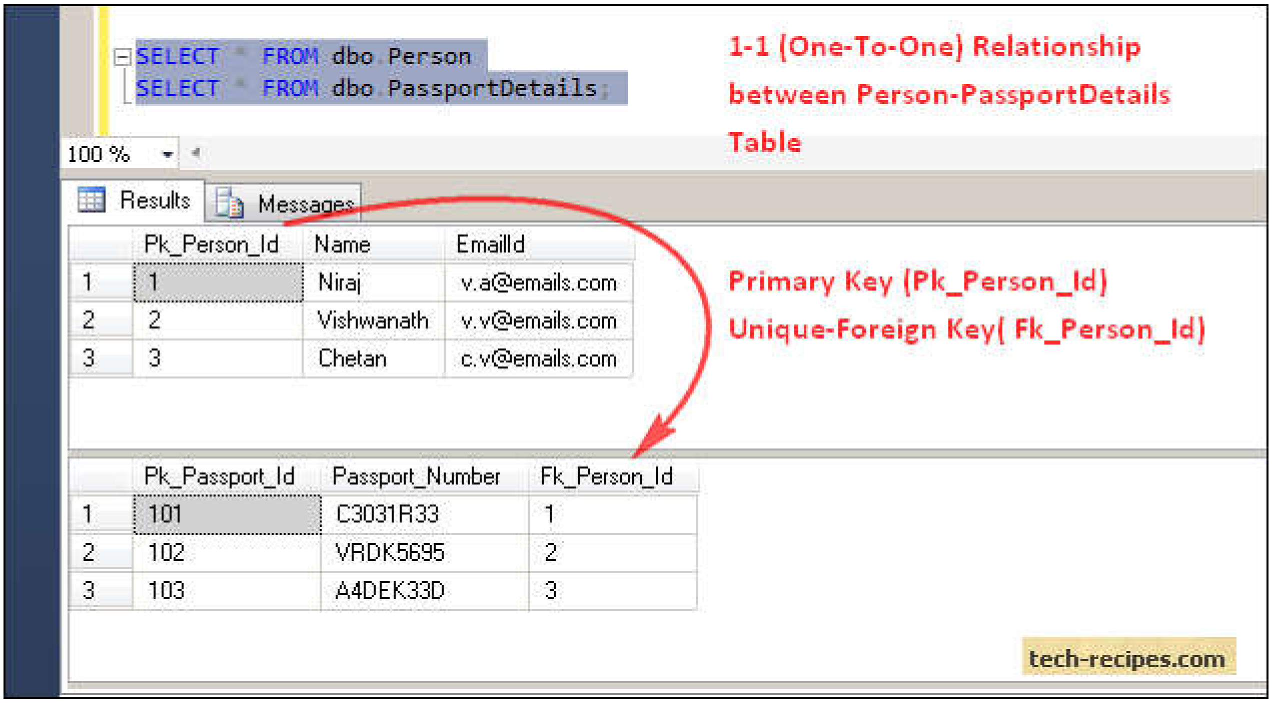
```
CREATE TABLE dbo.Person
(
Pk_Person_Id INT IDENTITY PRIMARY KEY,
Name VARCHAR(255),
EmailId VARCHAR(255),
);

CREATE TABLE dbo.PassportDetails
(
Pk_Passport_Id INT PRIMARY KEY,
Passport_Number VARCHAR(255),
Fk_Person_Id INT UNIQUE FOREIGN KEY REFERENCES dbo.Person(Pk_Person_Id)
);

INSERT INTO dbo.Person VALUES ('Niraj','v.a@emails.com');
INSERT INTO dbo.Person VALUES ('Vishwanath','v.v@emails.com');
INSERT INTO dbo.Person VALUES ('Chetan','c.v@emails.com');
GO

INSERT INTO dbo.PassportDetails VALUES (101, 'C3031R33', 1);
INSERT INTO dbo.PassportDetails VALUES (102, 'VRDK5695', 2);
INSERT INTO dbo.PassportDetails VALUES (103, 'A4DEK33D', 3);
GO

SELECT * FROM dbo.Person
SELECT * FROM dbo.PassportDetails;
```



```
SELECT * FROM dbo.Person
SELECT * FROM dbo.PassportDetails;
```

	Pk_Person_Id	Name	EmailId
1	1	Niraj	v.a@emails.com
2	2	Vishwanath	v.v@emails.com
3	3	Chetan	c.v@emails.com

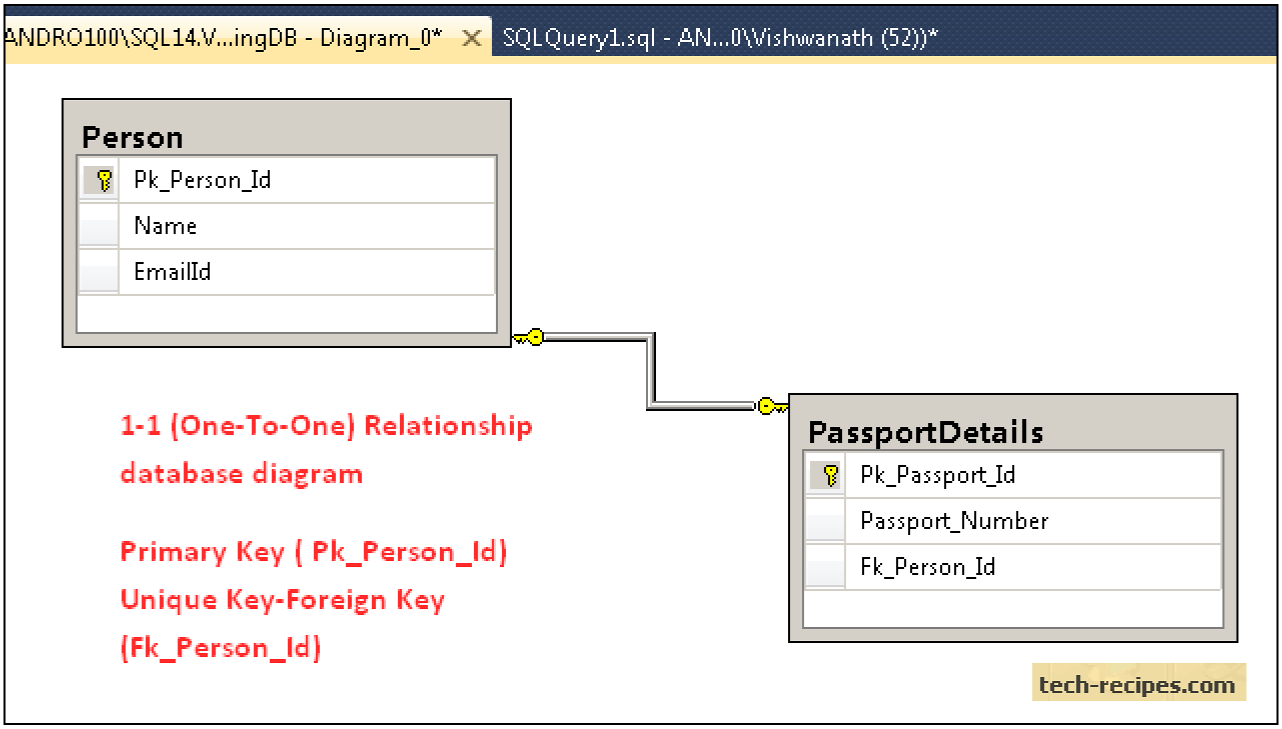
	Pk_Passport_Id	Passport_Number	Fk_Person_Id
1	101	C3031R33	1
2	102	VRDK5695	2
3	103	A4DEK33D	3

1-1 (One-To-One) Relationship between Person-PassportDetails Table

Primary Key (Pk_Person_Id)

Unique-Foreign Key(Fk_Person_Id)

One-to-One Relationship is implemented using **dbo.Person(Pk_Person_Id)** as the Primary key and **dbo.PassportDetails(fk_person_id)** as (Unique Key Constraint-Foreign Key).



ANDRO100\SQL14.V...ingDB - Diagram_0*
SQLQuery1.sql - AN...0\Vishwanath (52))*

Person
- Pk_Person_Id
- Name
- EmailId

PassportDetails
- Pk_Passport_Id
- Passport_Number
- Fk_Person_Id

1-1 (One-To-One) Relationship database diagram

Primary Key (Pk_Person_Id)

Unique Key-Foreign Key (Fk_Person_Id)

Therefore, it will always have only one matching row between the Person-PassportDetails table based on the **dbo.Person(Pk_Person_Id)-dbo.PassportDetails(Fk_Person_Id)** relationship.

1. Create two Tables (Table A & Table B) with the Primary Key on Both the tables.
2. Create Foreign key in Table B which references the Primary key of Table A.
3. Add a Unique Constraint on the Foreign Key column of Table B.

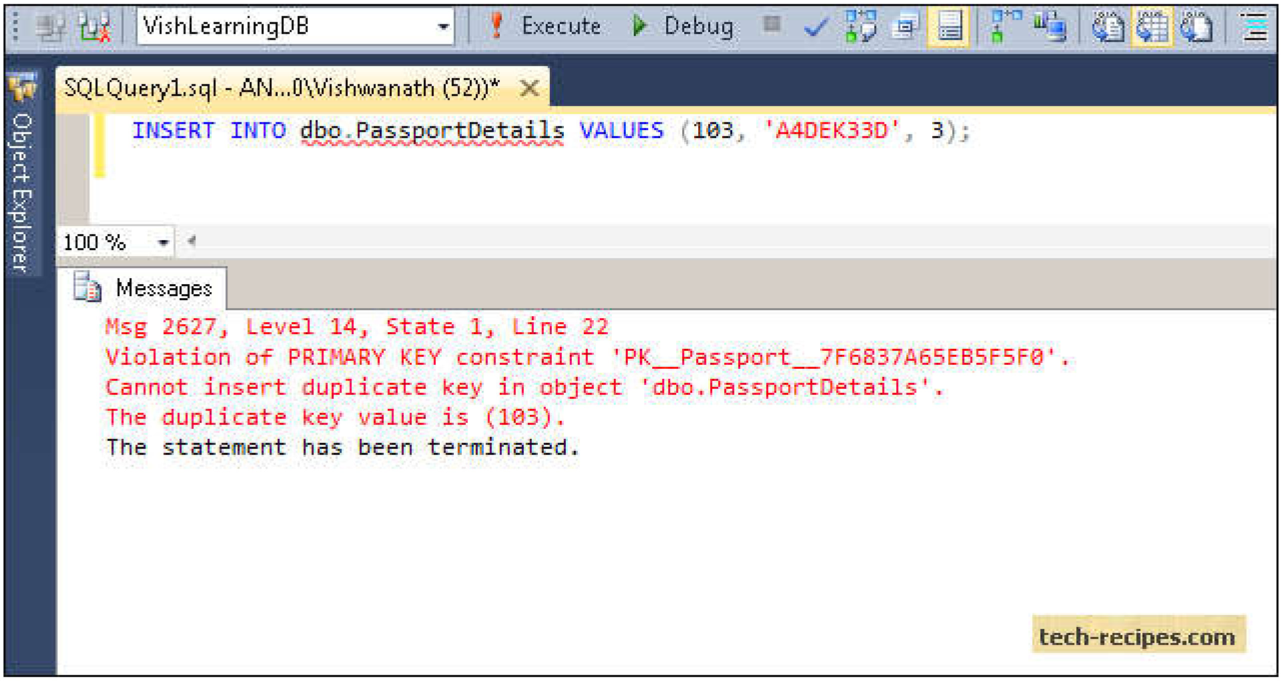
What happens if we try to insert passport details for the same fk_person_id which already exists in the passportDetails table?

We get an error of Unique key violation.

```
Msg 2627, Level 14, State 1, Line 1

Violation of UNIQUE KEY constraint 'UQ__Passport__04554C334F12BBB9'.

Cannot insert duplicate key in object 'dbo.PassportDetails'. The duplicate
key value is (3).
The statement has been terminated.
```



```
INSERT INTO dbo.PassportDetails VALUES (103, 'A4DEK33D', 3);
```

```
Msg 2627, Level 14, State 1, Line 22
Violation of PRIMARY KEY constraint 'PK__Passport__7F6837A65EB5F5F0'.
Cannot insert duplicate key in object 'dbo.PassportDetails'.
The duplicate key value is (103).
The statement has been terminated.
```

2. One-Many Relationship (1-M Relationship)

The One-to-Many relationship is defined as a relationship between two tables where a row from one table can have multiple matching rows in another table. This relationship can be created using **Primary key-Foreign key relationship**.

In the One-to-Many Relationship in SQL Server, for example, a book can have multiple authors. Let's implement this in SQL Server.

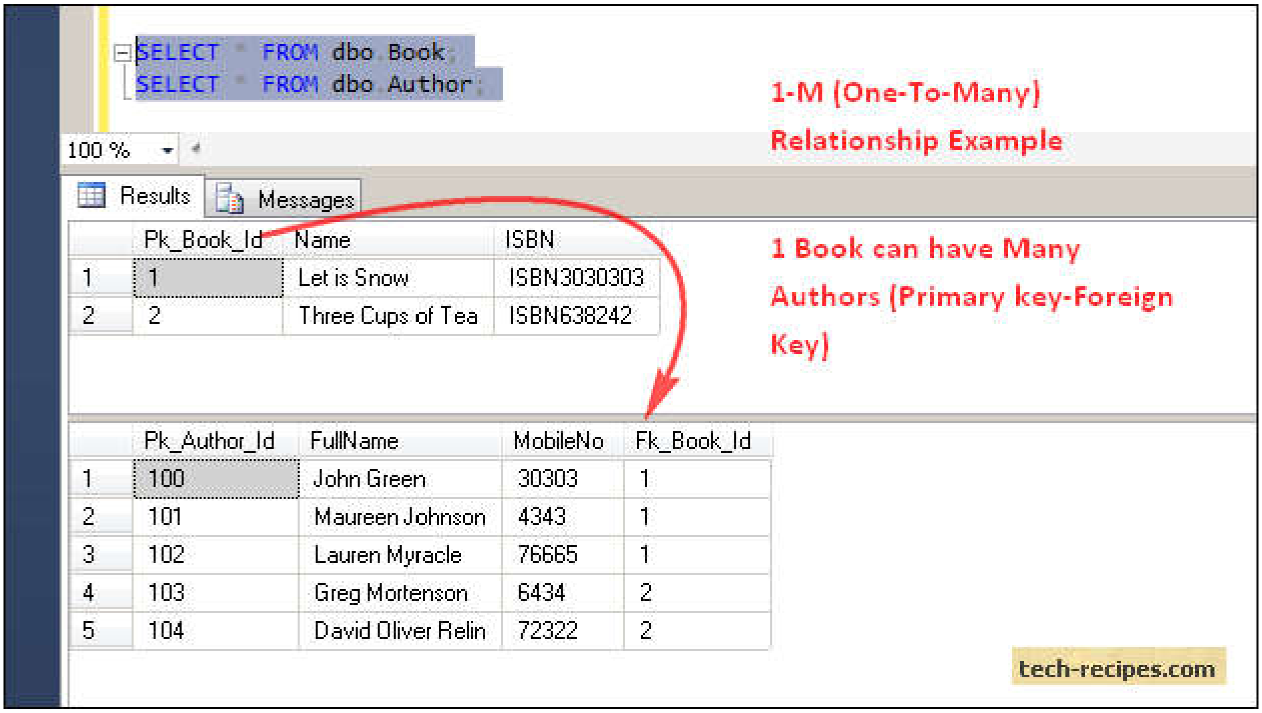
```sql
CREATE TABLE dbo.Book
(
Pk_Book_Id INT PRIMARY KEY,
Name VARCHAR(255),
ISBN VARCHAR(255)
);

CREATE TABLE dbo.Author
(
Pk_Author_Id INT PRIMARY KEY,
FullName     VARCHAR(255),
MobileNo     CHAR(10),
Fk_Book_Id   INT FOREIGN KEY REFERENCES Book(Pk_Book_Id)
);

INSERT INTO Book VALUES (1, 'Let is Snow', 'ISBN3030303');
INSERT INTO Book VALUES (2, 'Three Cups of Tea','ISBN638242');
GO

INSERT INTO dbo.Author VALUES(100,'John Green','30303',1);
INSERT INTO dbo.Author VALUES(101,'Maureen Johnson','4343',1);
INSERT INTO dbo.Author VALUES(102,'Lauren Myracle','76665',1);
INSERT INTO dbo.Author VALUES(103,'Greg Mortenson','6434',2);
INSERT INTO dbo.Author VALUES(104,'David Oliver Relin','72322',2);
GO

SELECT * FROM dbo.Book;
SELECT * FROM dbo.Author;
```



```sql
SELECT * FROM dbo.Book;
SELECT * FROM dbo.Author;
```

1-M (One-To-Many) Relationship Example

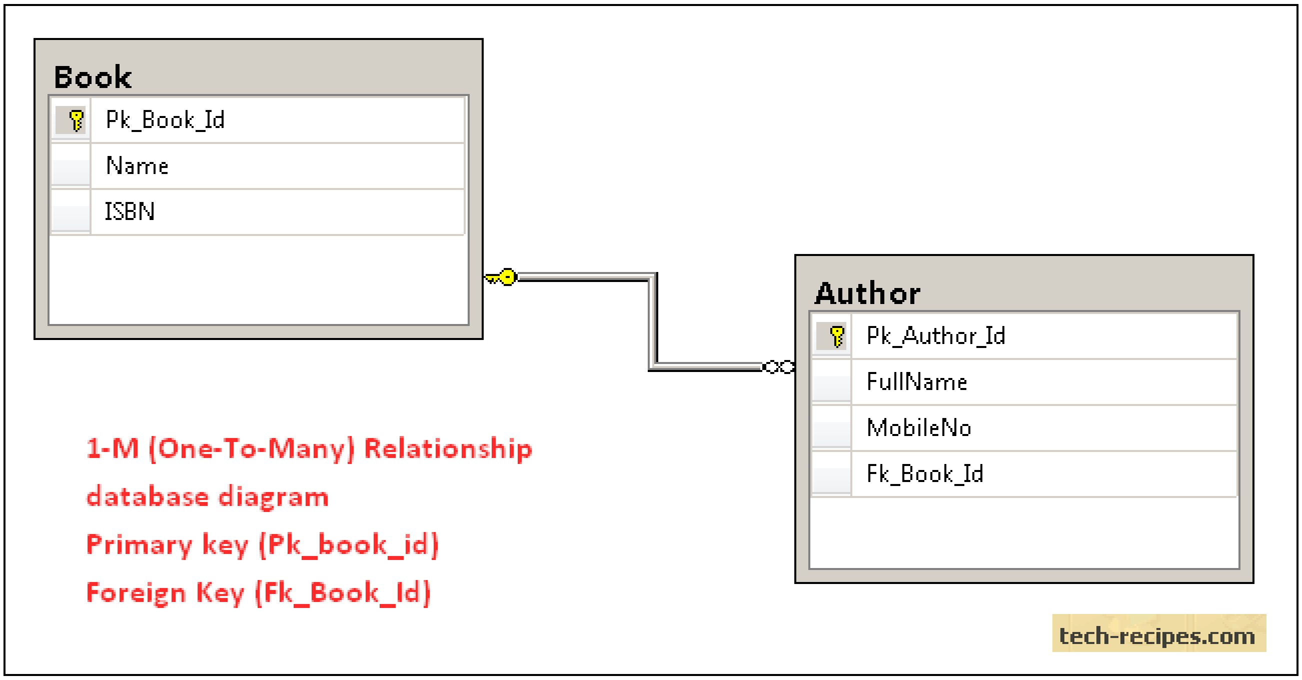
	Pk_Book_Id	Name	ISBN
1	1	Let is Snow	ISBN3030303
2	2	Three Cups of Tea	ISBN638242

1 Book can have Many Authors (Primary key-Foreign Key)

	Pk_Author_Id	FullName	MobileNo	Fk_Book_Id
1	100	John Green	30303	1
2	101	Maureen Johnson	4343	1
3	102	Lauren Myracle	76665	1
4	103	Greg Mortenson	6434	2
5	104	David Oliver Relin	72322	2

tech-recipes.com

One-to-Many Relationship is implemented using **dbo.Book(Pk_Book_Id) as the Primary Key and dbo.Author (Fk_Book_Id) as (Foreign Key)**. Thus, it will always have only One-to-Many (One Book-Multiple Authors) matching rows between the Book-Author table based on the dbo.Book (Pk_Book_Id)-dbo.Author(Fk_Book_Id) relationship.



1-M (One-To-Many) Relationship
database diagram
Primary key (Pk_book_id)
Foreign Key (Fk_Book_Id)

1. Create two Tables (Table A & Table B) with the Primary Key on both the tables.
2. Create a Foreign key in Table B which references the Primary key of Table A.

3. Many-Many Relationship

```
/*Many to Many*/
CREATE TABLE teamPlayer
(
playerID INT NOT NULL,
teamID INT NOT NULL,
PRIMARY KEY(playerID, teamID),

constraint fk_teamPlayer__Player
foreign key(playerID) references Player(personID),

constraint fk_teamPlayer__Team
foreign key(teamID) references Team(teamID)

);
```

```
one book many authors
one to many
```

key	fk
1	3
2	3
3	3
4	3

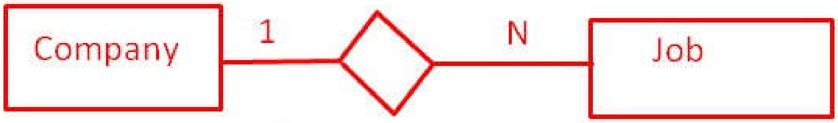
many to many

user comments vine

key

u1	v1
u1	v2
u1	v3
u1	v4
u2	v3
u2	v4
u2	v5
u2	v6

Kazuvanje



една копманија содржи повеќе огласи

еден оглас припаѓа на една копманија