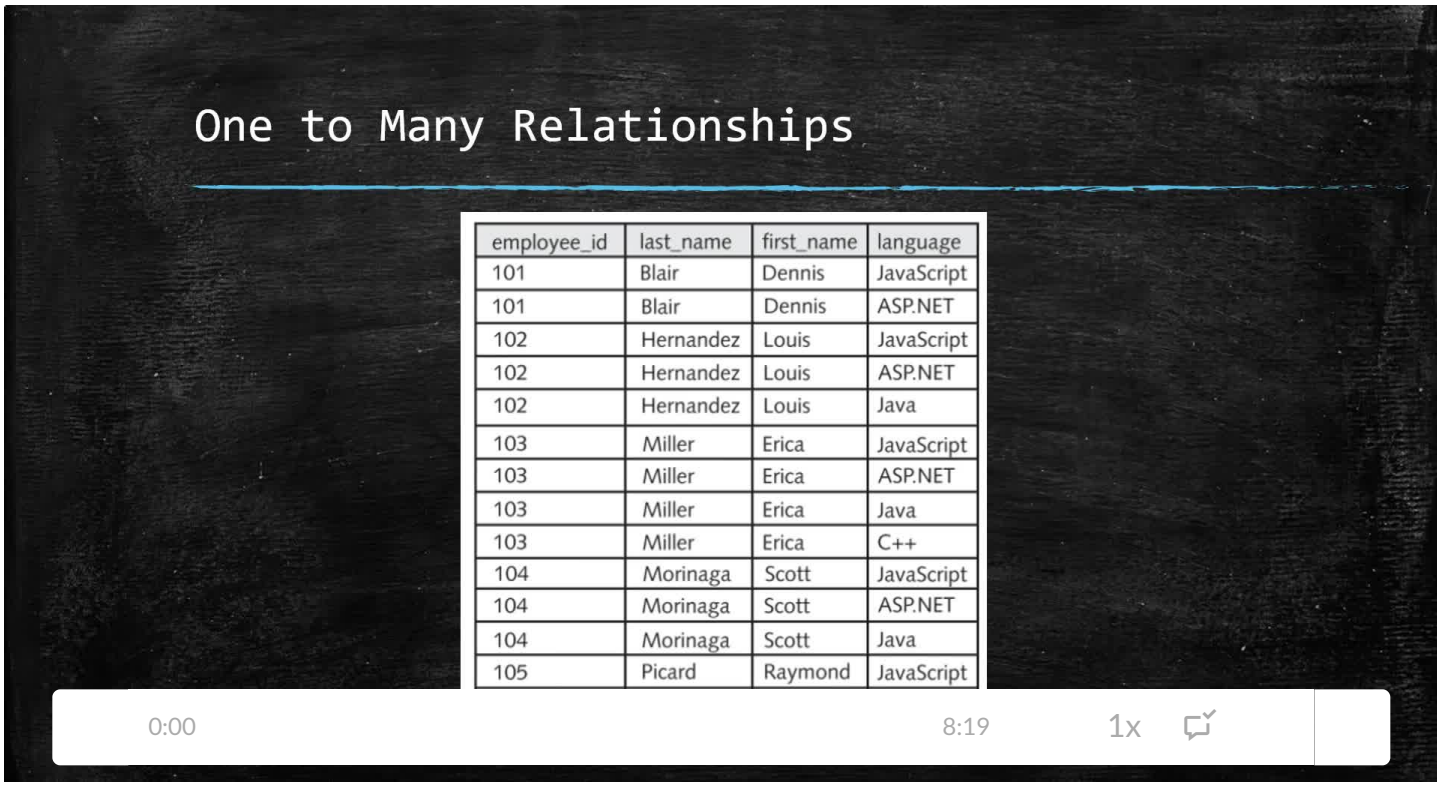
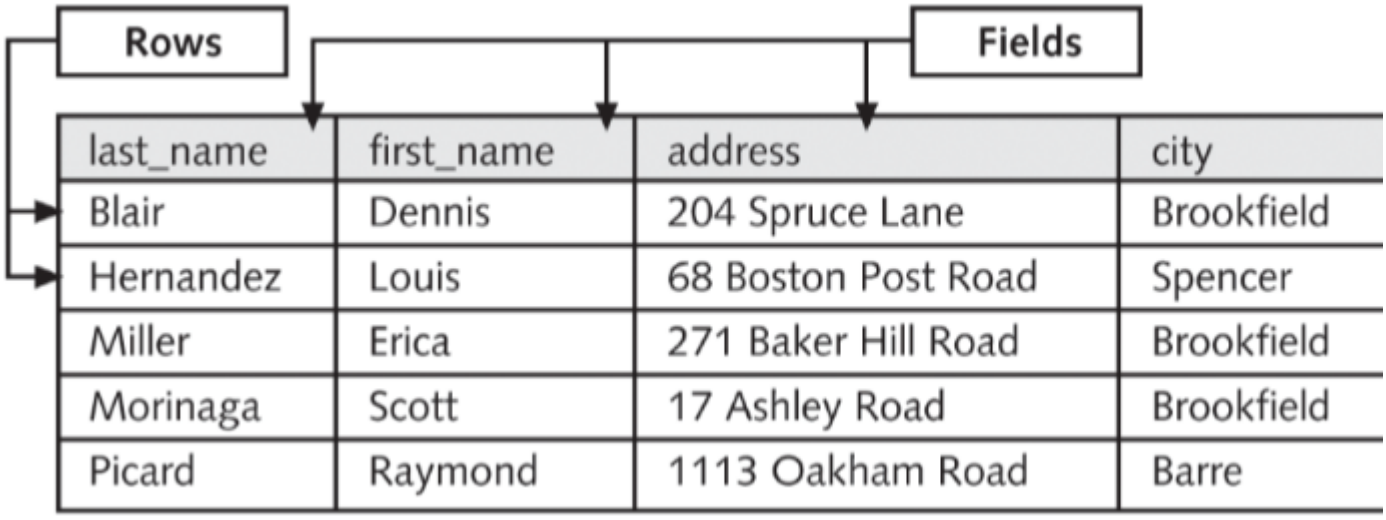
Examples: Relationship Types

Examples for One to One Relationship



Let us consider a simple Employee Table:



Let us now summarize key aspects of relational databases:

- Relational databases consist of one or more related tables
- A primary table is the main table in a relationship that is referenced by another table
- A related table (or “child table”) references a primary table in a relational database
- A primary key is a field that contains a unique identifier for each record in a primary table
- A primary key is a type of index, which identifies records in a database to make retrievals and sorting
  faster
- A foreign key is a field in a related table that refers to the primary key in a primary table
- Primary and foreign keys link records across multiple tables in a relational database

Now on to a summary of relationship types. First, let us focus on **<u>One to One Relationships</u>**:

- A one-to-one relationship exists between two tables when a related table contains exactly one record for each record in the primary table
- Create one-to-one relationships to break information into multiple, logical sets
- Information in the tables in a one-to-one relationship can be placed within a single table
- Make the information in one of the tables confidential and accessible only by certain individuals
- In the below tables sampling, one employee in Employees table has exactly one matching row in Payroll table.

employee_id	last_name	first_name	address	city	sta
101	Blair	Dennis	204 Spruce Lane	Brookfield	MA
102	Hernandez	Louis	68 Boston Post Road	Spencer	MA
103	Miller	Erica	271 Baker Hill Road	Brookfield	MA
104	Morinaga	Scott	17 Ashley Road	Brookfield	MA
105	Picard	Raymond	1113 Oakham Road	Barre	MA

employee_id	start_date	pay_rate	health_coverage	year_vested	401
101	2002	$21.25	none	na	no
102	1999	$28.00	Family Plan	2001	yes
103	1997	$24.50	Individual	na	yes
104	1994	$36.00	Family Plan	1996	yes
105	1995	$31.00	Individual	1997	yes

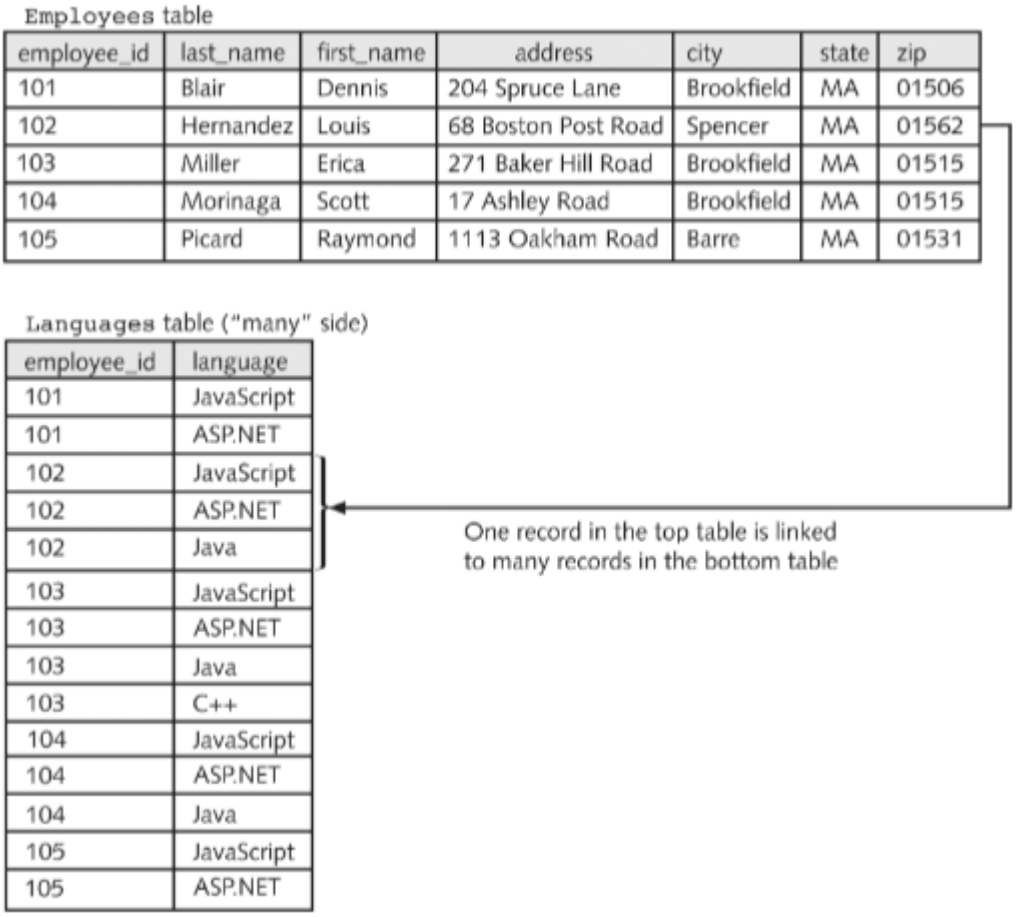
Next, let us check **One to Many Relationships**:

- A one-to-many relationship exists in a relational database when one record in a primary table has many related records in a related table
- Breaking tables into multiple related tables to reduce redundant and duplicate information is called normalization
- Provides a more efficient and less redundant method of storing this information in a database
- In the below table sampling, one employee has proficiency on multiple languages.

employee_id	last_name	first_name	language
101	Blair	Dennis	JavaScript
101	Blair	Dennis	ASP.NET
102	Hernandez	Louis	JavaScript
102	Hernandez	Louis	ASP.NET
102	Hernandez	Louis	Java
103	Miller	Erica	JavaScript
103	Miller	Erica	ASP.NET
103	Miller	Erica	Java
103	Miller	Erica	C++
104	Morinaga	Scott	JavaScript
104	Morinaga	Scott	ASP.NET
104	Morinaga	Scott	Java
105	Picard	Raymond	JavaScript
105	Picard	Raymond	ASP.NET

Now, let us check **Many to Many Relationships**:

- A many-to-many relationship exists in a relational database when many records in one table are related to many records in another table
- A junction table creates a one-to-many relationship for each of the two tables in a many-to-many relationship
- A junction table contains foreign keys from the two tables
- In the below table sampling, we need to figure out how many years of experience each employee has using each programming language.



- We need to create "Experience" junction table, and have primary keys from Employees and Language tables store in the junction table, along with number of years of experience as a third column. Here is the pictorial representation of this manipulation:

`Employees` table

employee_id	last_name	first_name	address	city	state	zip
101	Blair	Dennis	204 Spruce Lane	Brookfield	MA	01506
102	Hernandez	Louis	68 Boston Post Road	Spencer	MA	01562
103	Miller	Erica	271 Baker Hill Road	Brookfield	MA	01515
104	Morinaga	Scott	17 Ashley Road	Brookfield	MA	01515
105	Picard	Raymond	1113 Oakham Road	Barre	MA	01531

One record in the `Employees` table is linked to many records in the `Experience` junction table

`Languages` table

language_id	language
10	JavaScript
11	ASP.NET
12	Java
13	C++

One record in the `Languages` table is linked to many records in the `Experience` junction table

`Experience` junction table

employee_id	language_id	years
101	10	5
101	11	4
102	10	3
102	11	2
102	12	3
103	10	2
103	11	3
103	12	6
103	13	3
104	10	7
104	11	5
104	12	8
105	10	4
105	11	2