Relational Databases

* Relational databases are the most common way to persistently store data in web applications.
* A relational database is used to store a collection of relations. This involves storing “records” in tables.
* Each row in a table (or entity) corresponds to one record, and the columns correspond to fields (or attributes) of the record.

Example:

|  |  |
| --- | --- |
| people | |
| id | first\_name | last\_name | address | phone |
| 1 | Mainul | Hasan | 385/E,Mohammdpur | 01926008559 |
| 2 | Mainul | Hasan | 385/E,Mohammadpur | 01817781353 |

* Id is used to form relationships to other tables. It’s referred to as the primary key of the table.
* For example, Mainul Hasan appeared in the table twice. Why? Because he has two phones. More specifically, there is a one-to-many relationship between people and phones – one person can have many phones.
* We can normalize the database by creating two tables, one for people and a separate table for phones:
* Each record in the phone table will hold the id of a person.
* Database normalization is important for removing data redundancy or dependency
* Divide a large table into many smaller tables and then forming relationships between those tables.
* In our case in the phone table, this person\_id is referred to as a foreign key.
* Given the id of a person, we can now search the phone table for all of the phones that belong to a person.

Example – A one-to-many relationship between tables:

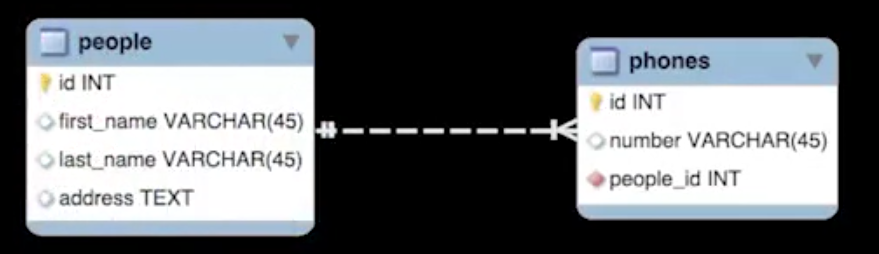
|  |  |
| --- | --- |
| people | |
| id | first\_name | last\_name | address |
| 1 | Mainul | Hasan | 385/E,Mohammdpur |
| 2 | Mainul | Hasan | 385/E,Mohammadpur |

|  |  |
| --- | --- |
| phones | |
| id | person\_id | | number |
| 1 | Mainul | | 01926008559 |
| 2 | Mainul | | 01817781353 |

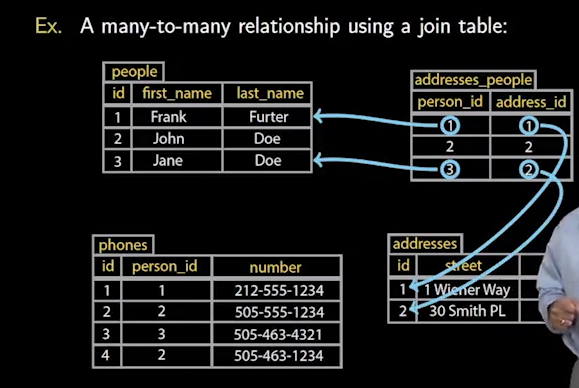
* Foreign key points to primary key of parent table.

Schema and Entity-Relationship Models

* The structure/organization of the tables in a database is referred to as a schema.
* An entity-relationship model is a common way of abstractly capturing a database schema.



* A phone belongs to a person
* A person has many phones
* Notice that we could further normalize the database by creating an address table.
* However, in this case, the one-to-many relationship is in the other direction, i.e., we have one address for many people in the table.
* We can imagine a situation where one person also has many addresses, e.g., one for work, one for home, etc.
* Thus, we really need to create a many-to-many relationship between people and addresses.
* This is done by creating a join table – it’s called this because it “joins” the people and addresses tables.
* The join table in the following example is called addresses\_people. Notice that it only stores foreign keys, and has no primary keys.



* Entity-Relationship Schema

