Relational Database

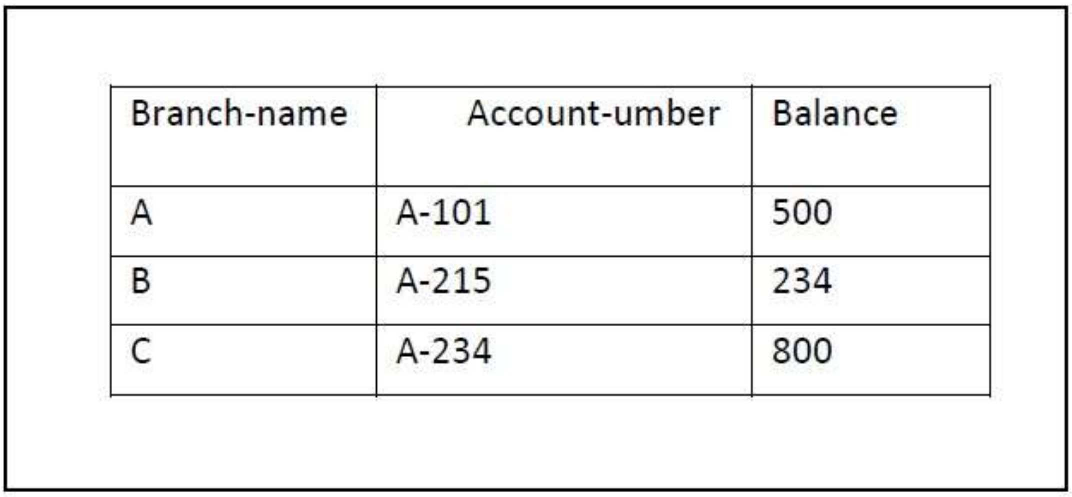
A relational database consists of a collection of tables, each of which assigned a unique name. The table consists of a number of rows . A row in a table represents a relation among a set of values.

A relational database is a collection of data items organized as a set of tables from which data can be accessed in many different ways without having to reorganize the database tables.

The relational database was invented by E. F. Codd at IBM in 1970.
A relational database is a set of tables containing data fitted into predefined categories. Each table (which is sometimes called a *relation*) contains one or more data categories in columns.

Each row contains a unique instance of data for the categories defined by the columns. Consider the account table as shown in figure 3.1. It has three column header (three fields):

- Branch-name
- Account-number
- Balance



Branch-name	Account-umber	Balance
A	A-101	500
B	A-215	234
C	A-234	800

Account Table

Database Instances and Schemas

A **schema** is a description of a particular collection of data, using the given data model, simply means the structure or the form of the database without any data in it.

Schema is of three types:

Physical schema: how the data stored in blocks of storage is described at this level.

Logical schema: programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).

View schema: this generally describes end user interaction with database systems.

Instance : the actual content of the database at a particular point in time.

Keys

As you may already know, databases use tables to organize information. Each table consists of a number of rows, each of which corresponds to a single database record. So, how do databases keep all of these records straight? It’s through the use of keys.

Primary Keys

The first type of key we'll discuss is the primary key. ***Every database table should have one or more columns designated as the primary key***. The value this key holds should be unique for each record in the database. For example, assume we have a table called Employees that contains personnel information for every employee in our firm. We'd need to select an appropriate primary key that would uniquely identify each employee. Your first thought might be to use the employee's name.

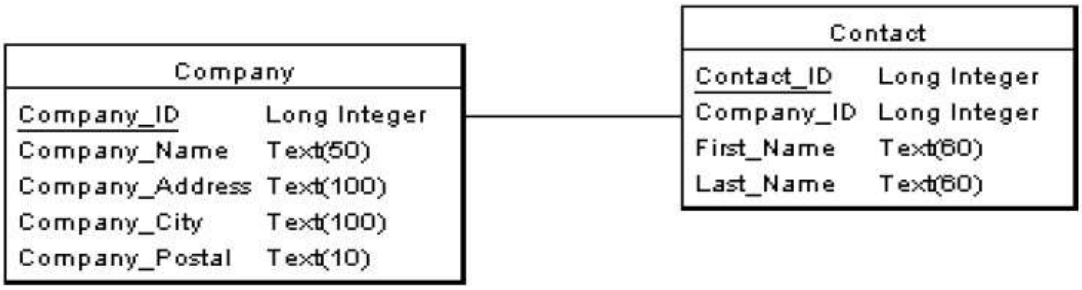
This wouldn't work out very well because it's conceivable that you'd hire two employees with the same name. A better choice might be to use a unique employee ID number that you assign to each employee when they're hired

Once you decide upon a primary key and set it up in the database, ***the database management system (DBMS)*** will enforce the uniqueness of the key. If you try to insert a record into a table with a primary key that duplicates an existing record, the insert will fail.

Foreign Keys

The other type of key that we'll discuss in this course is the foreign key. These keys are used to create relationships between tables. Natural relationships exist between tables in most
database structures The foreign key link is set up by matching columns in one table (the child) to the primary key columns in another table (the parent).

Example 1 :In the example of figure 3.2, there is a link between the Company and Contact tables. The Company table is the parent table in the link. The Contact table is the child: the Company ID field in the Contact table indicates which Company a Contact belongs to.



Company	
Company_ID	Long Integer
Company_Name	Text(50)
Company_Address	Text(100)
Company_City	Text(100)
Company_Postal	Text(10)

Contact	
Contact_ID	Long Integer
Company_ID	Long Integer
First_Name	Text(60)
Last_Name	Text(60)

Example 1 of a relation between two tables

Example 2 : suppose there is a need to design a database to hold the history of each student of a school. The information of each student should have :

- Student name
- Student phone
- Information about the student in each year in the school.

This database should have at least two tables , the first one like the following schema:

St Table-Schema=(st-name, st-phone)

The second table contain the information about the history of the student :

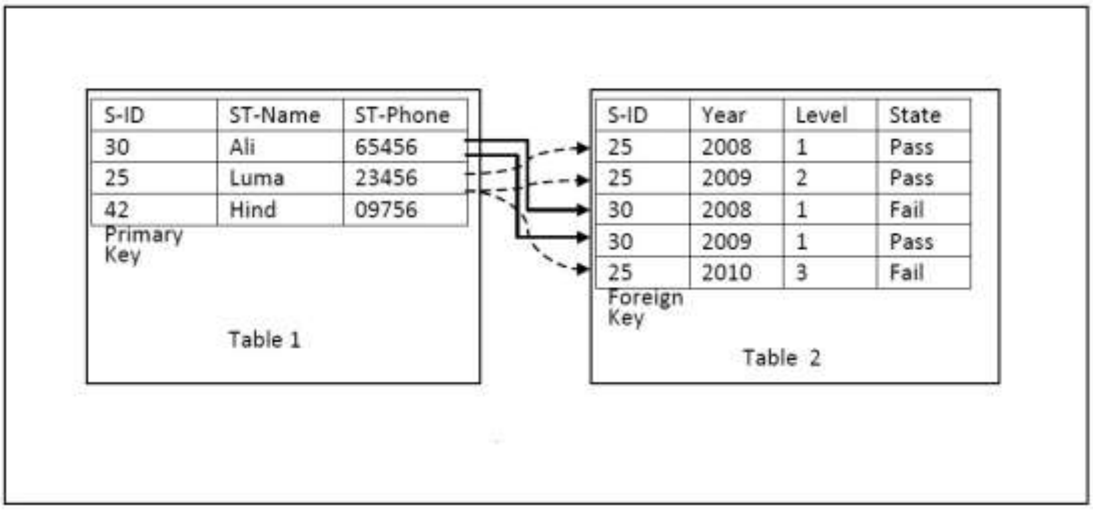
History-schema=(year , level , state)

To link those two tables a primary key in the first table and a foreign key in the second table must be added to the tables as shown in figure below

The new schemas will be :

St Table-Schema=(S-ID , st-name, st-phone)

History-schema =(S-ID , year , level , state)



S-ID	ST-Name	ST-Phone
30	Ali	65456
25	Luma	23456
42	Hind	09756

Primary Key

Table 1

S-ID	Year	Level	State
25	2008	1	Pass
25	2009	2	Pass
30	2008	1	Fail
30	2009	1	Pass
25	2010	3	Fail

Foreign Key

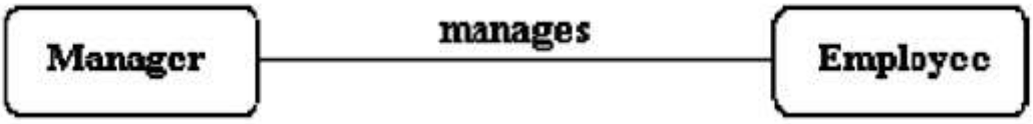
Table 2

Example (2) of a relation between two tables

Degree of a Relationship

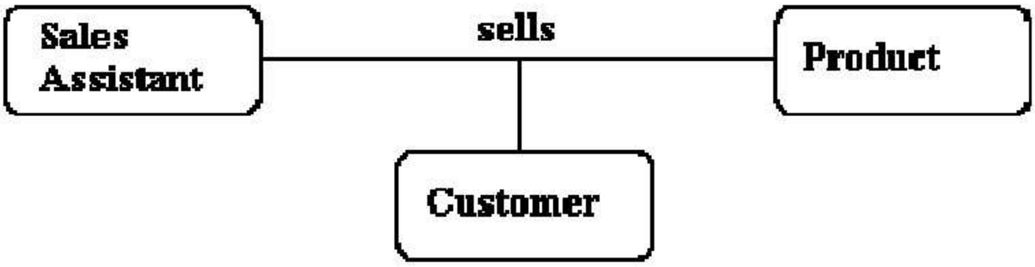
The number of participating entities in a relationship is known as the degree of the relationship.

If there are two entity types involved it is a one-to-many relationship. single-valued or multi models. is a meaningful association between entity types.



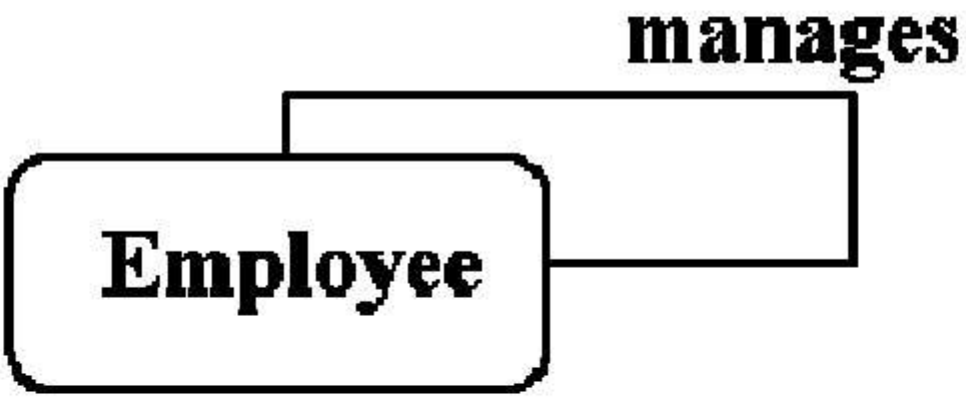
Binary Relationships

If there are three entity types involved it is a ternary relationship type.



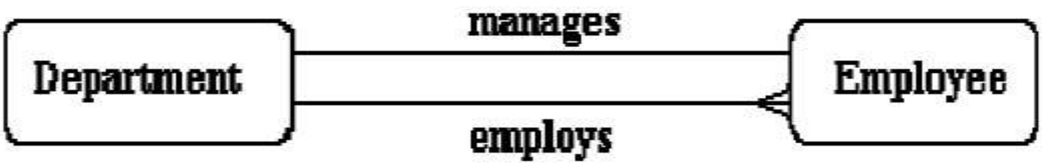
Ternary relationship

- It is possible to have a n – relationship (e. g. quaternary or unary).
- Unary relationships are also known as a recursive relationship.



Recursive relationship

- In the example above we are saying that employees are managed by employees.
- It is also possible to have entities associated through two or more distinct relationships



Multiple relationships