SQL Training, Session 4

In our previous sessions, we covered the basic SQL syntax and operators that should be sufficient enough to get you started with writing your own queries. Today's lesson is going to focus on databases. Our lesson will be based off the following presentation
http://user.it.uu.se/~arnea/datastrukturer/database.pdf

Introduction:

A **Database Management System** (DBMS) is defined as a collection of interrelated data. This data can be accessed through a set of programs. DBMS provides an environment that is efficient and straight forward to include.

DBMS systems are active in all aspects of our lives ranging from transactions, universities, sales, manufacturing…etc. We need DBMS to solve the problems that arise as a result of having individual data files that can be redundant, different formats, isolated, and difficult to update manually. There are also numerous issues involved with security.

Databases contain different levels of abstraction. The first level is **physical** and it is where a description of records is stored. The second level is **logical.** This level stores the data in databases and relationships in the data. The third level is **view**, where apps can control what data details are seen for security purposes.

Instances and Schema:

Schema contains information regarding the logical structure of the database such as the three levels of abstraction.

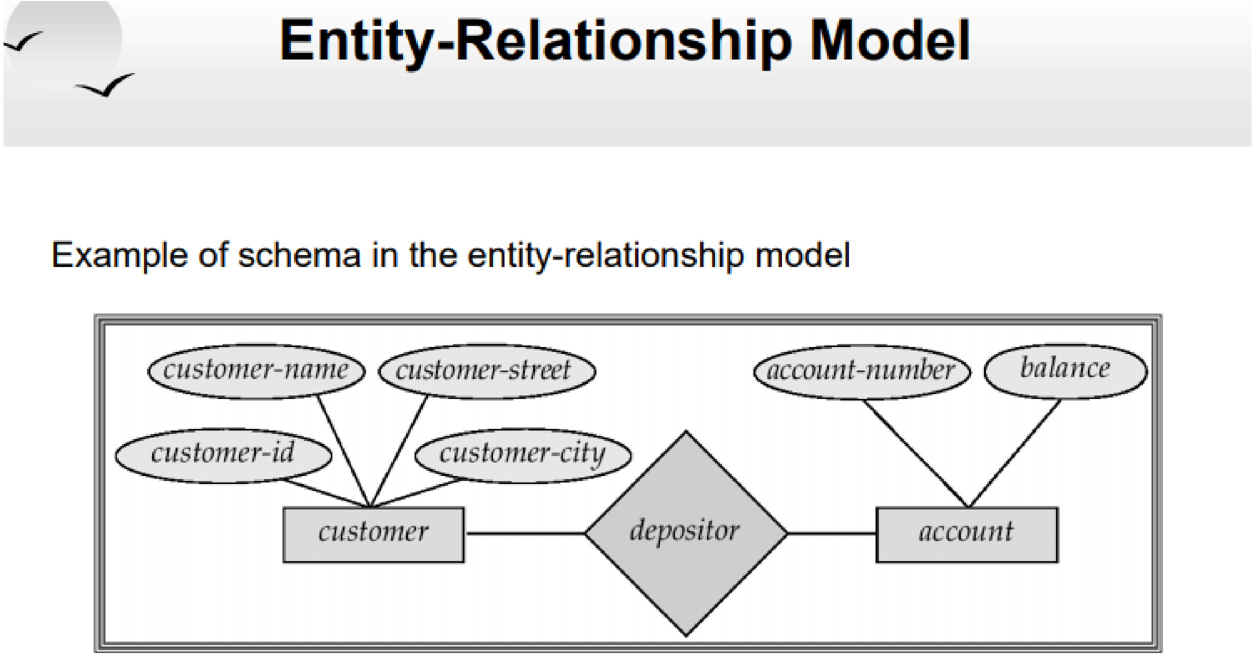
Instance is the actual content of the database at a particular point in time. This is interchangeably known as the value of a variable.

Data Models:

Data models are a collection of tools for describing the following:
-the data it's self
-data relationships
-data semantics

-data constraints

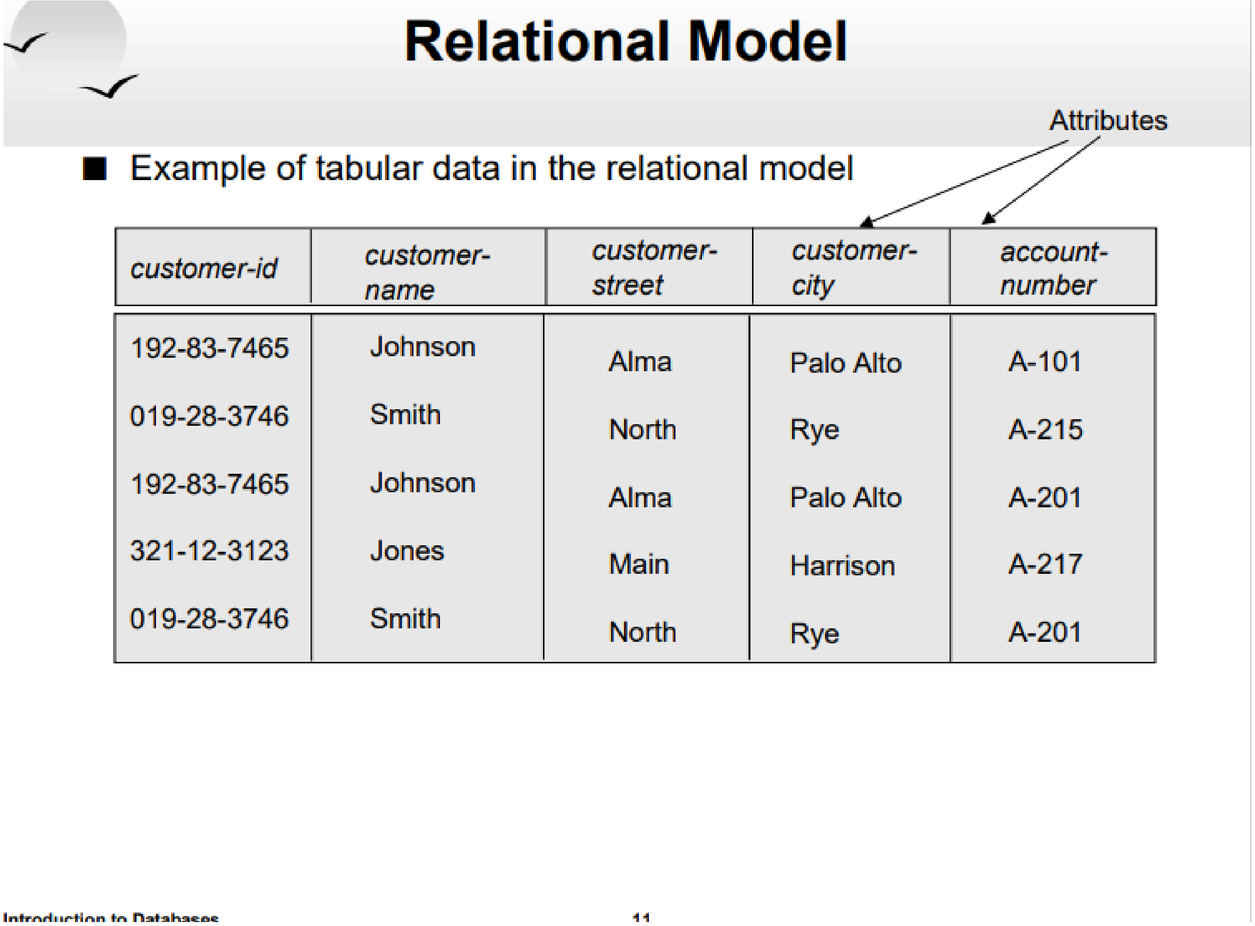
Data models can be entity-relationship, relational, object-oriented, semi-structure, network, and hierarchical. For our purposes, we will be focusing on the entity-relationship and relational model.



Entity-Relationship Model

Example of schema in the entity-relationship model

The schema is the design of the database. Notice that many of the outer tables can be traced to the center. In this case, this schema represents a database for some bank. Each shape is a relational table.



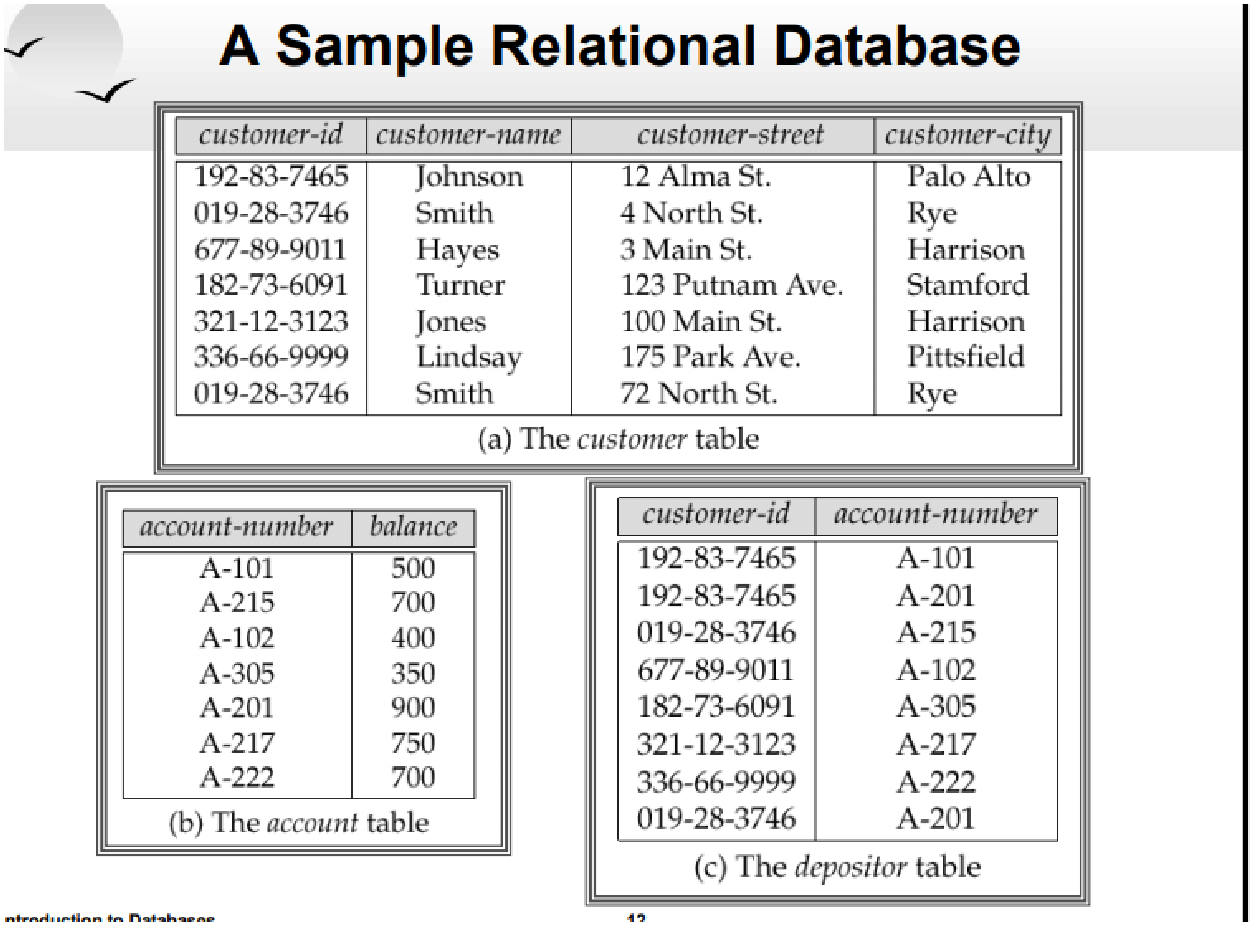
Relational Model

- Example of tabular data in the relational model

Attributes

customer-id	customer-name	customer-street	customer-city	account-number
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Each table in the schema is tabular, meaning the data is divided into rows and columns where rows represent a customer/account and columns represent the features of that customer and account.



A Sample Relational Database

customer-id	customer-name	customer-street	customer-city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	account-number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

This demonstrates the actual content of some of the tables inside of this schema. What can you tell about these tables? The customer table contains the personal information of customers. Each row is a customer. The customer ID is the primary key since each value is unique to a customer and non-empty.

The account table contains the balance and account where each row is a different account. While this table does not show this, accounts are a **one to many** relationship with customers. A customer can have multiple accounts.

The depositor table contains the customer ID and the account number. Each row represents a customer-id to account pair. However as we mentioned, a customer could potentially have multiple accounts.

Running reports on this schema can be done with **SQL**. Depending on the information requested, you will need to know what tables to join on what key. For example, if we wanted to grab the balance for each person, we would need to join customer table on depositor table to get the account number, then join the result from the previous join to the account table on account number to get balance.

Key items:
From this intro to database concepts, you should understand the following:

-The main data models
-One-to-many relationships
-one-to-one relationships
-The main Data models
-Schemas

Homework:

Your homework will test your database intuition. I want you to build a basic schema in the form of simple csv files. Each csv file is a "table". You'll need to think through how to consolidate one to one relationships and one to many relationships. Refer to the sample database we created.

An organization grants key-card access to rooms based on groups that key-card holders belong to. You may assume that users belong to only one group. Your job is to design the database that supports the key-card system.

There are six users and four groups. Modesto and Ayine are in group "I.T." Christopher and Cheong Woo are in a group "Sales." There are four rooms: "101","102"," Auditorium A", and "Auditorium B." Saulat is in group "Administration." Group "Operations" currently doesn't have any users assigned. I.T should be able to access rooms 101 and 102. Sales should be able to access rooms 102 and Auditorium A. Administration does not have access to any rooms. Heidy is a new employee, who has not been assigned to any group.