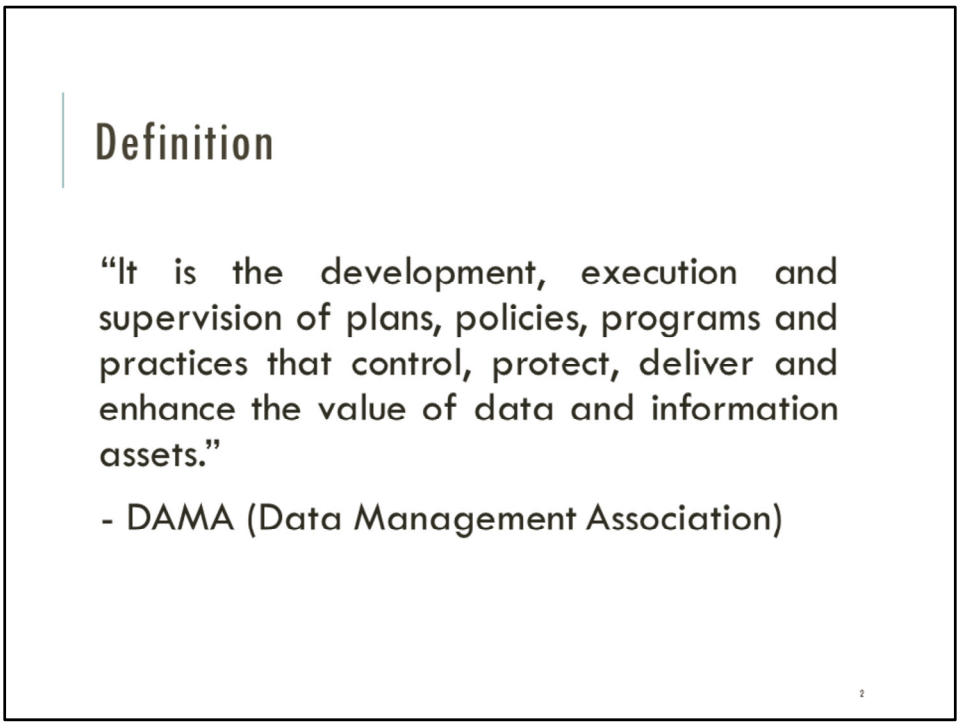
Data management

CARLOS RIVERO
ROCHESTER INSTITUTE OF TECHNOLOGY
DEPT. OF COMPUTER SCIENCE

These slides present the general context of data management and the topics that we are going to study in CSCI 320.

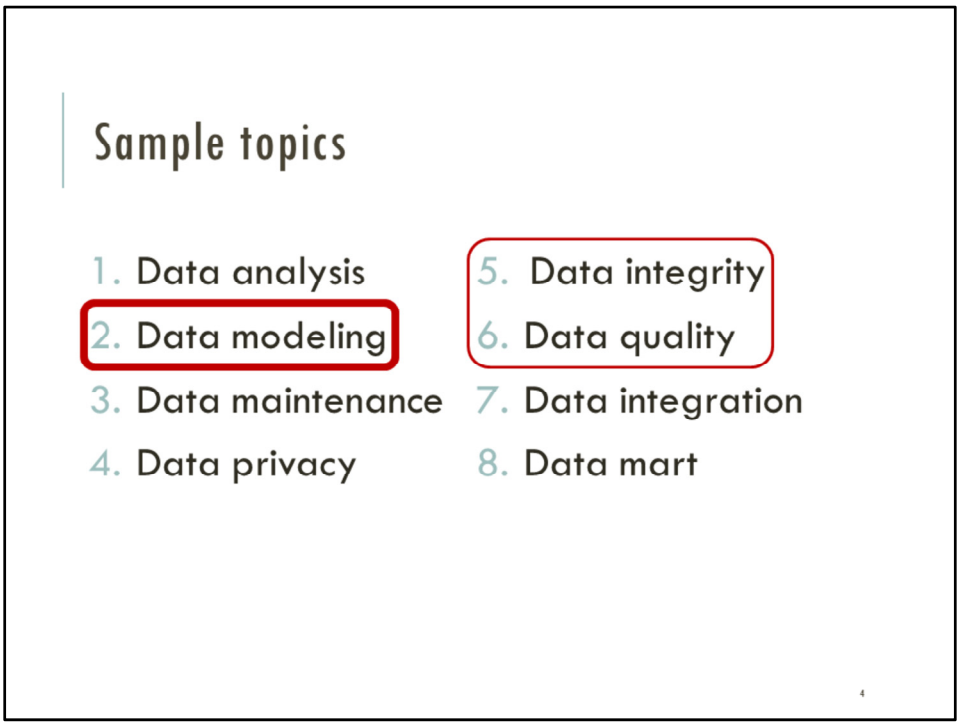


According to DAMA, which is the international Data Management Association, "Data management is the development, execution and supervision of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets."

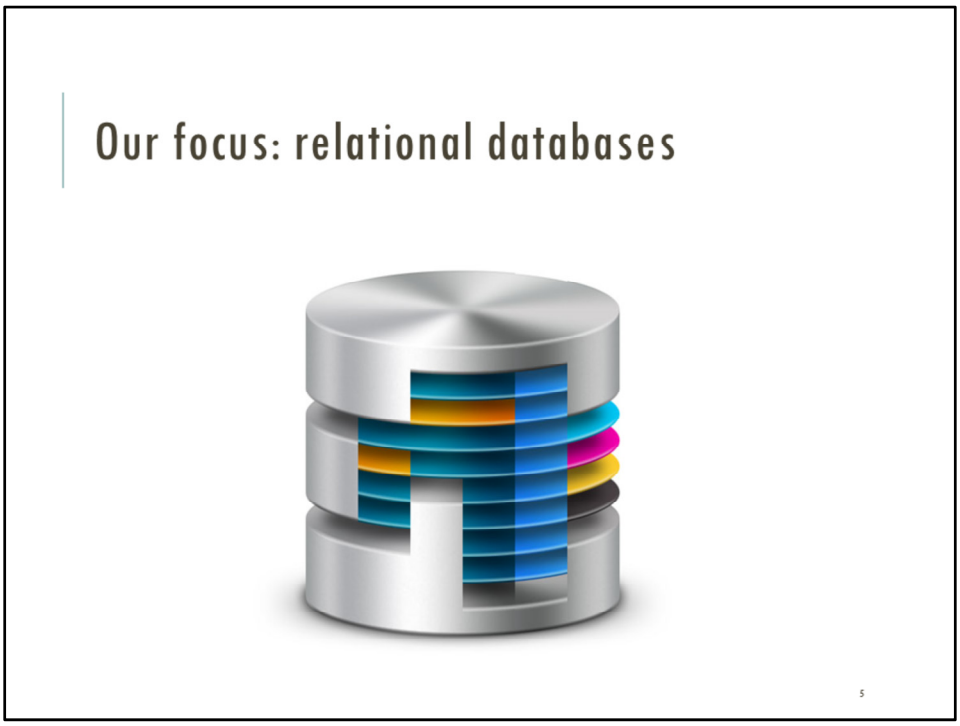
Got it but… what that means?

3

The previous definition aims to cover all of the topics that may be included in the context of data management. In our case, we are going to focus on just a couple of these topics.



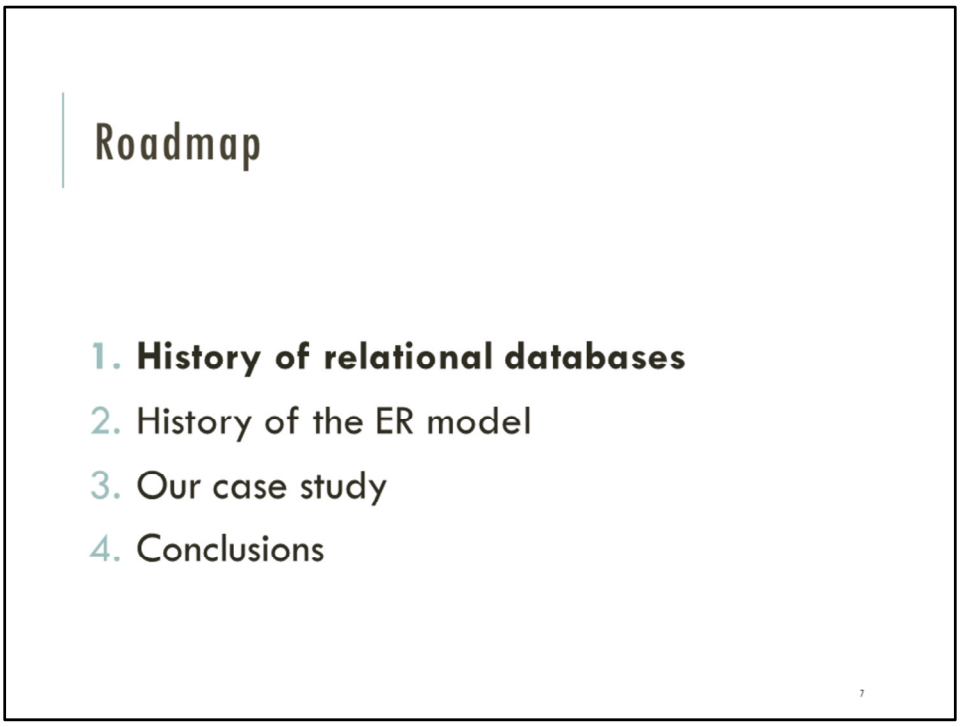
Some examples of the topics that are covered by data management are data analysis, modeling, maintenance, privacy, integrity, quality, integration, or data mart, just to mention a few of them. In our case, we are going to primarily focus on data modeling, and a bit on data integrity and quality.



Within the previous topics, there are also a large amount of subtopics. Since this course is “Principles of Data Management”, we are going to focus on the traditional way of storing data: relational databases.

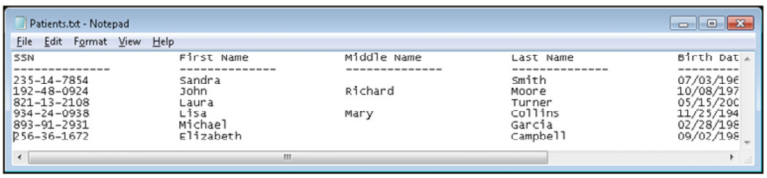
Have you used any database?

Before starting, let me ask you, have you used any relational database? How many of you are completely new to databases?

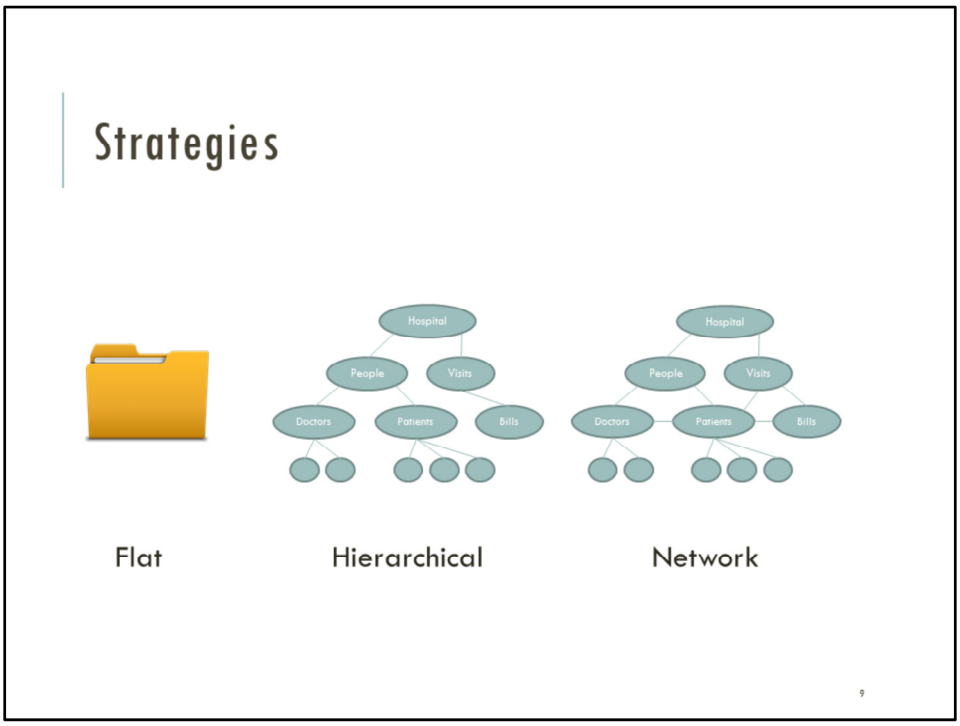


This is the roadmap we are going to follow. We are going to study the history of relational databases and the ER model. Additionally, we are going to introduce a case study that we are going to follow during the whole course and, finally, we will see some conclusions. Starting with the relational databases…

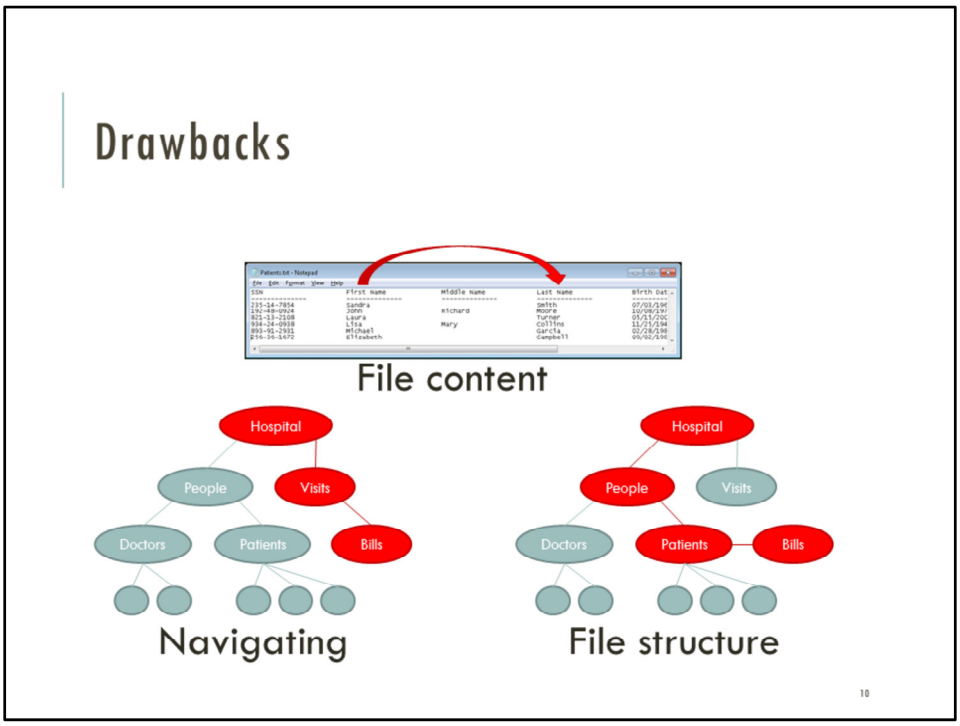
History: files



To contextualize relational databases, let's study a bit of history. During late 1960s, computers started to be quite popular among companies because they were a cost-effective option to manage the data they needed to store and process. At this time, companies used files to manage their data needs, for instance, here you can see an example of a file that contains data of some patients being treated in a hospital.

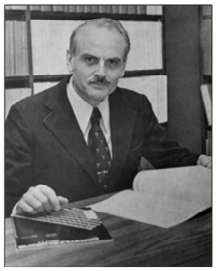


There were different strategies to manage these files. The easiest one was the flat files, in which all of the files were stored at the same level or folder. The hierarchical strategy was designed to store files in a tree structure so it was possible to navigate until finding the right type of files. Similarly, the network strategy was similar to the previous one but using a general graph instead of a tree.



When these systems evolved, it was usually necessary to make changes in the content of a given file, e.g., we wish to switch the first and last name columns. All programs that read that file needed to be changed to take the new situation into account. Another problem was that, to access some data, it was mandatory to navigate through the file system to find that data, e.g., if I want to access a specific bill I need to navigate from different nodes. A final drawback is that, if there is a change in the file structure, we need to change all programs that perform navigations, e.g., bills are accessed from patients.

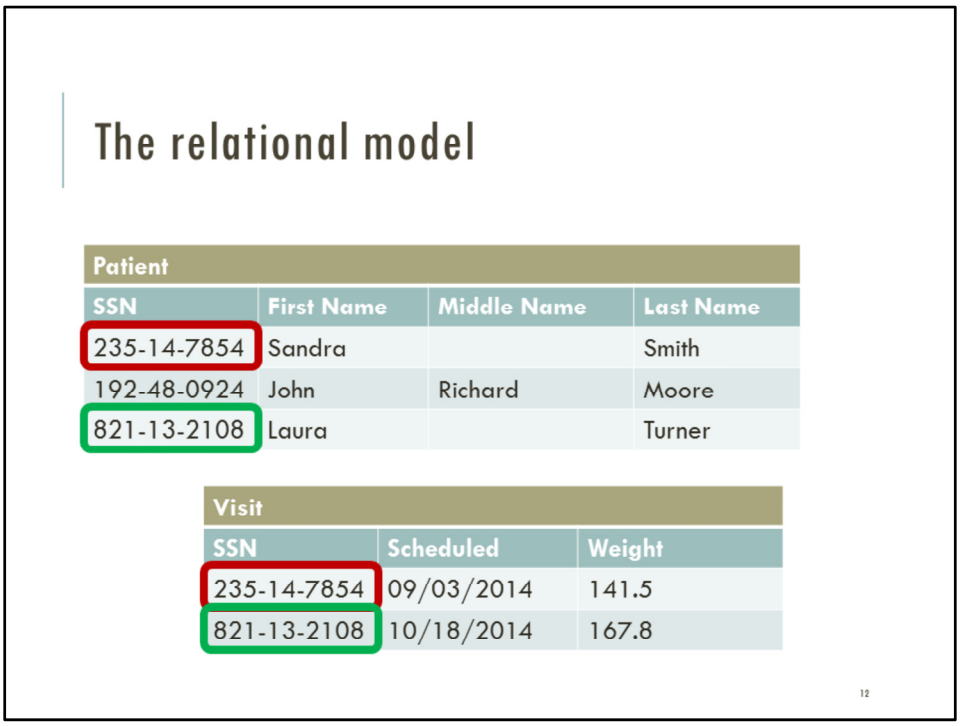
Edgar F. Codd (1923 – 2003)



- E. F. Codd. A Relational Model of Data for Large Shared Data Banks. Commun. ACM 13(6): 377-387 (1970).
- IBM Almaden
- Turing Award, 1981

11

Edgar Codd devised the relational model to solve the problems with data stored in the file system. He proposed the model in 1970 in a paper called: “A Relational Model of Data for Large Shared Data Banks”. He was working at IBM Almaden at that time. He received the Turing Award in 1981.



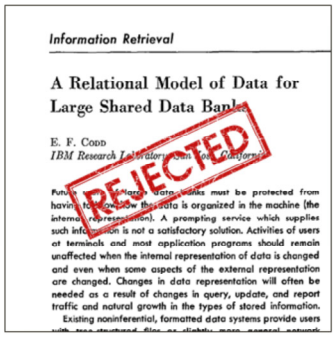
A relational model consists of “tables” called relations, each of which comprises fixed-length tuples. Each relation is used for a different type of entity. We define keys over relations, which allow us to uniquely identify a tuple in a relation. By using keys, we are able to refer to different tuples. For instance, in this example, we define relations Patient and Visit with different attributes like the social security number (SSN) or first, middle and last names. We use the SSN as the key for a patient since it uniquely identifies a person. Using that key, we are able to refer to each patient in the Visit relation, e.g., Sandra Smith attended a visit that was scheduled for 09/03/2014 and her weight was 141.5 pounds.

Such a good idea!

13

The relational model was a revolutionary idea but, do you think it was successful?

At the beginning...



Information Retrieval

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

REJECTED

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users

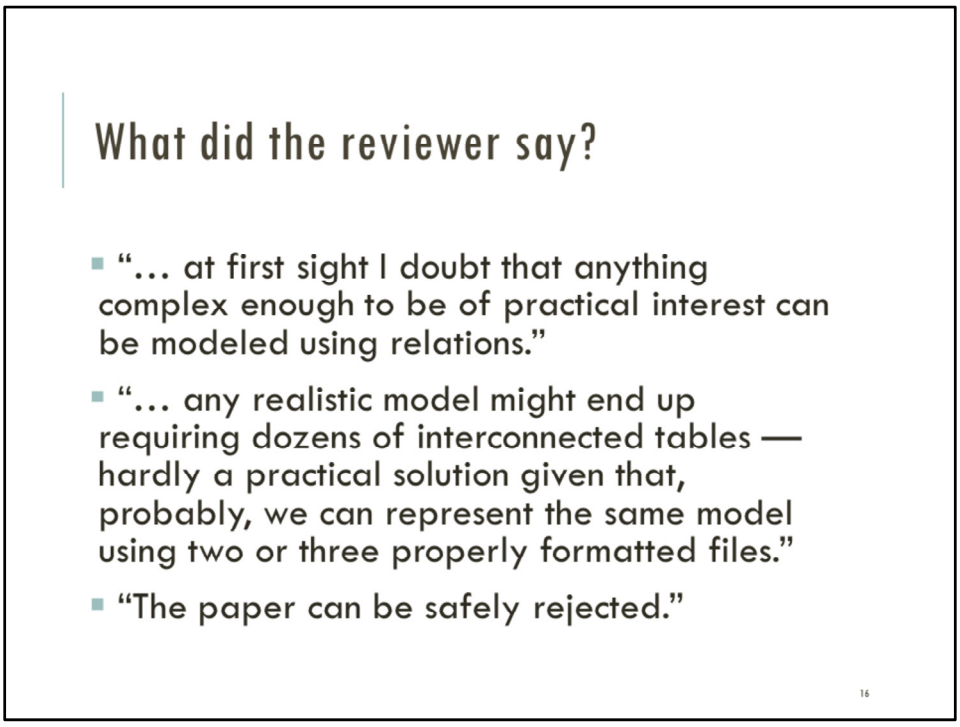
14

His paper was initially rejected!

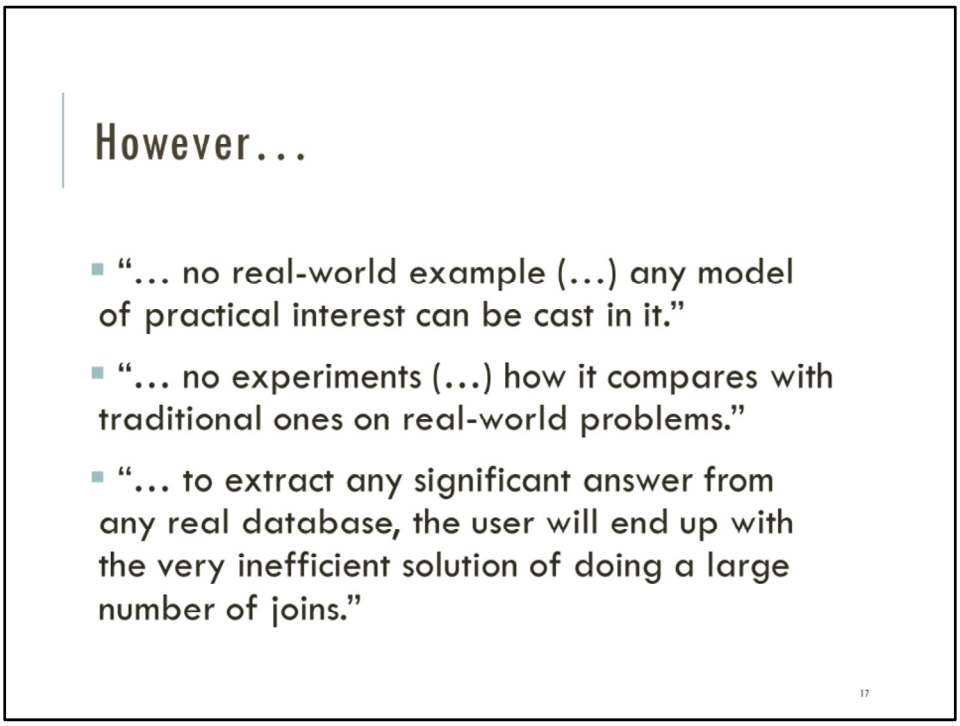
<Digress>

15

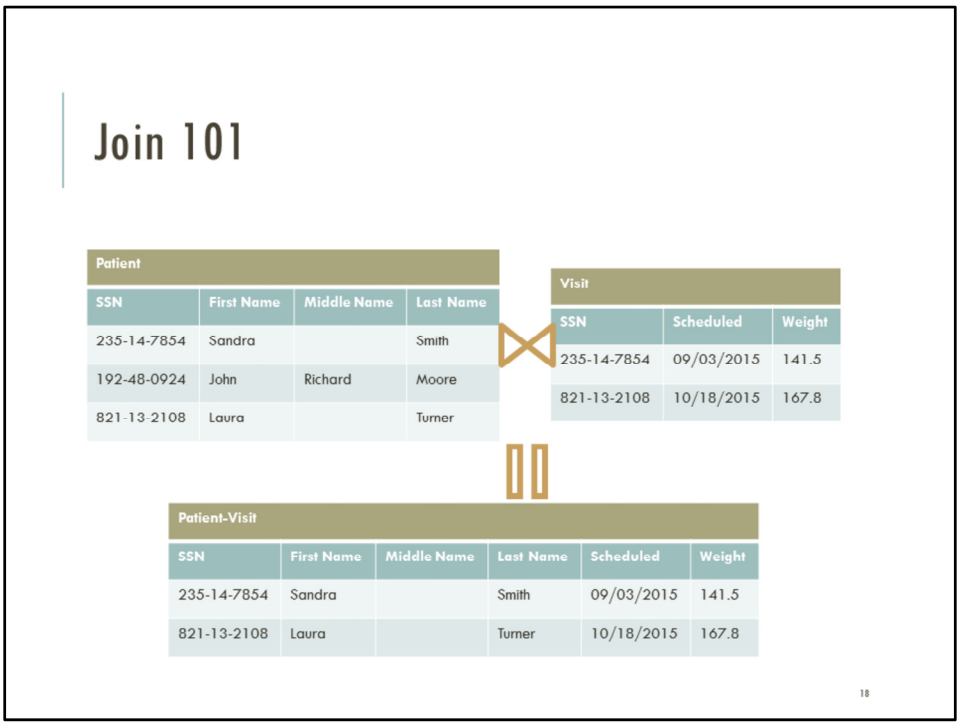
Let’s digress a bit…



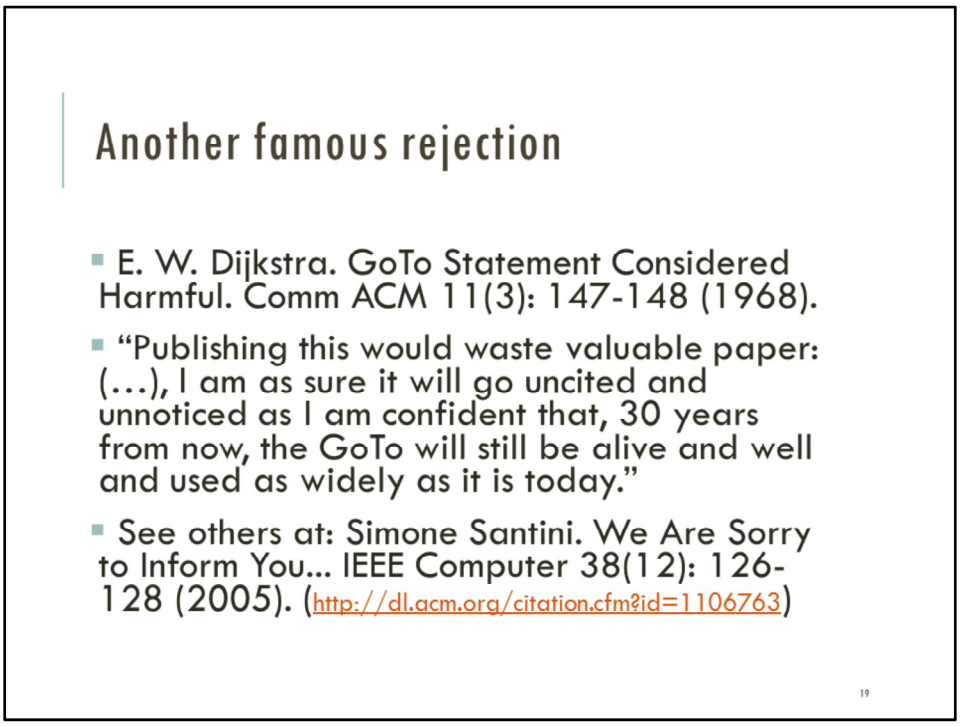
The reviewer said the following about Codd's paper: "… at first sight I doubt that anything complex enough to be of practical interest can be modeled using relations.", "… any realistic model might end up requiring dozens of interconnected tables —hardly a practical solution given that, probably, we can represent the same model using two or three properly formatted files.", and "The paper can be safely rejected." As we will see, these comments were not very proper.



However, we must also say that some of the comments were accurate. The paper presented no real-world example and no evaluation. Additionally, he did not present any implementation since it was impractical at the moment due to a general lack of sufficient computing power. Finally, the reviewer pointed out one of the main drawbacks of relational databases: for complex queries, we need to perform a (possibly large) number of joins.



We will study joins in this course but, just to let you know, a join is when we join two relations by one or more attributes and get a single relation as a result. In this case, we take relations Patient and Visit and join them using the SSN, resulting in the Patient-Visit relation. Check that John Richard Moore did not attended any visit so he is excluded from the final relation.



Dijkstra published a paper in 1968 called "GoTo Statement Considered Harmful" stating that the GoTo statement should be removed from programming languages and advocating for the structured programming, which became quite popular and it has been the main programming paradigm before the object-oriented programming. Before publishing it, it had some criticism that, according to the reviewer: "Publishing this would waste valuable paper: I am as sure it will go uncited and unnoticed as I am confident that, 30 years from now, the GoTo will still be alive and well and used as widely as it is today." You have studied programming languages, where is the GoTo statement 30 years later? See other famous rejections in that final paper of Simone Santini.

The key message of this digression if that, first, you need to deal with rejection. It is normal to be disappointed with a rejection but you need to overcome the issue and try again and again until you are successful. I think that the previous examples help us with this, even Codd or Dijkstra were rejected!

. . . but do not underestimate reviews!

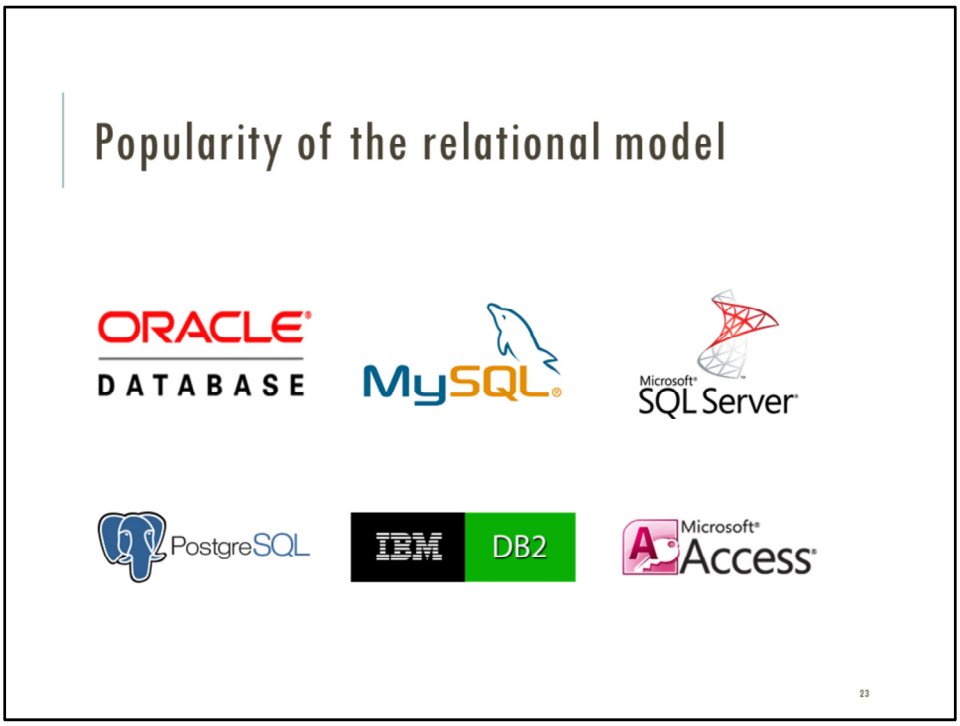
21

The previous is true but it is also true that you should not discard your reviews. You usually need to convince people that your work is worthy (the scientific community, your boss,…), so you should take reviews into account and always use them to improve your work.

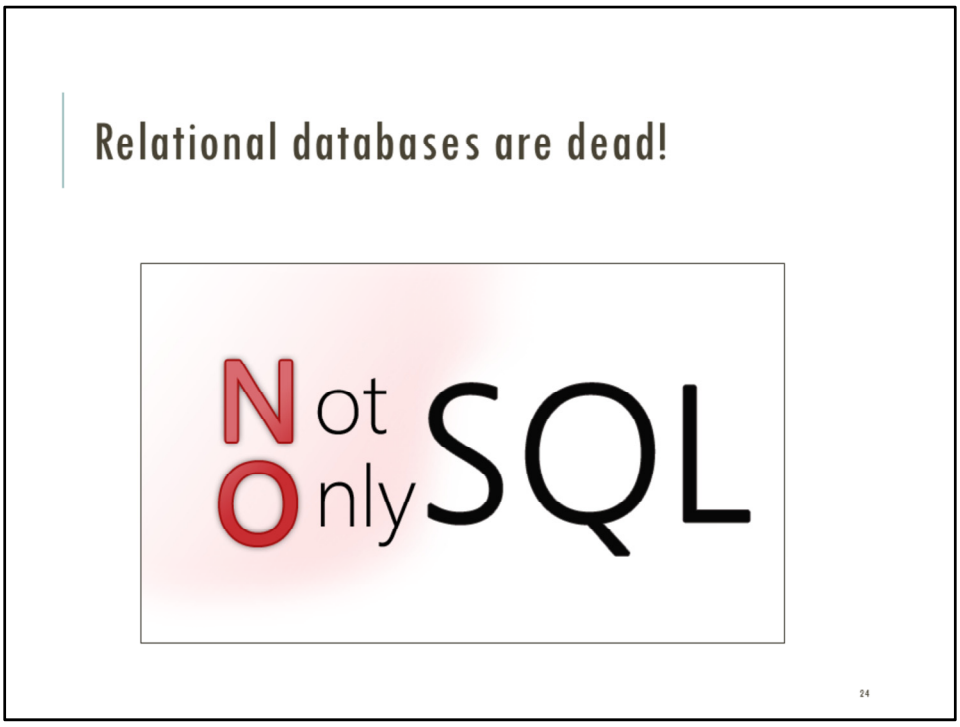
</Digress>

22

End of the digression. Let’s continue.



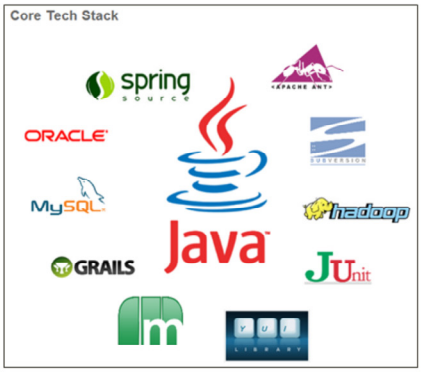
Resuming the previous slides, the relational model was rejected at the beginning but it became quite popular. A solid proof of this is the large number of commercial relational databases that exist nowadays, for instance, Oracle, MySQL, MS SQL Server, PostgreSQL, IBM DB2 or MS Access, just to mention a few examples. You can check for more relational databases online.



A coupe of years ago, some of the new big players like Facebook or Google stated that relational databases do not fit their needs due to the large amount of joins that they needed to perform. Do you remember the reviewer that criticized Codd's paper? NoSQL databases aim to avoid the relational model due to this problem. Not only that, they also claim that relational databases focus on structure data and there is a need for storing semi-structured data like documents. We will introduce NoSQL databases at the end of this course.

The truth about relational databases being dead is that, in my opinion, they are not dead at all. Despite of the large amount of new NoSQL tools and frameworks, relational databases have more than 30 years of experience and commercial products are quite stable and efficient. I agree that there is a need for new technologies, it is not possible to solve all problems with relational databases, but it is also not possible to completely discard these databases. I think that, nowadays, there are many commercial interests behind new technologies and it does not make sense for new players like Google or Facebook to design a new relational database to compete with Oracle or MySQL. This may be the reason behind some campaigns against relational databases.

The example of LinkedIn



Voldemort

https://engineering.linkedin.com/technology

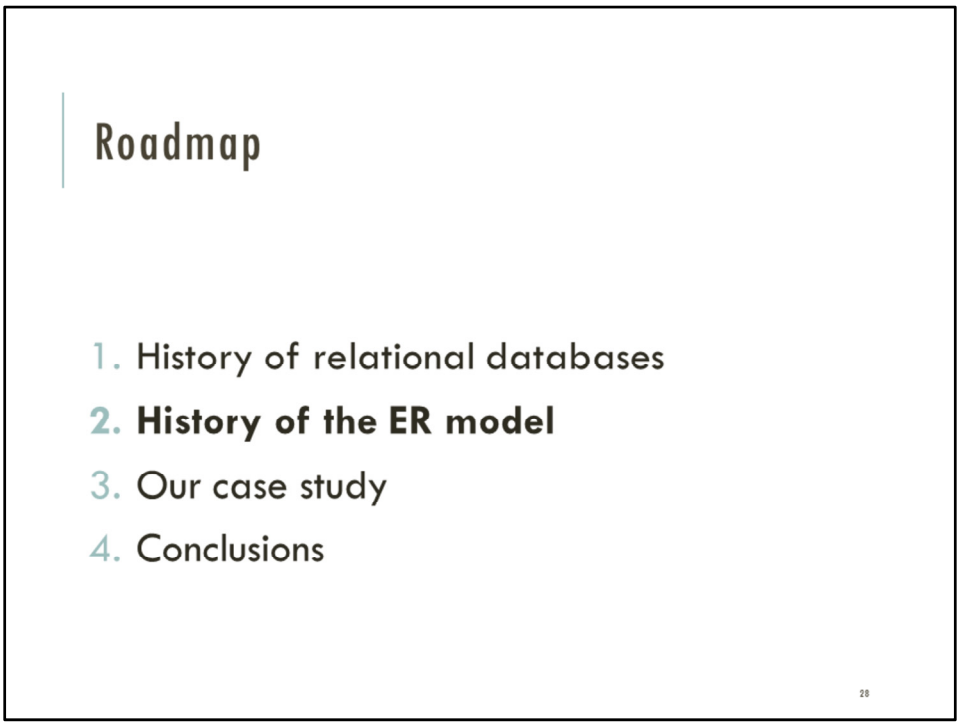
26

Companies today are using and integrating different technologies to support their services. In case of LinkedIn, they use two relational databases like MySQL and Oracle in conjunction with other technologies like Spring or Hadoop. You can also check that engineers at LinkedIn are devising their own NoSQL database that is called Voldemort.

How it works?

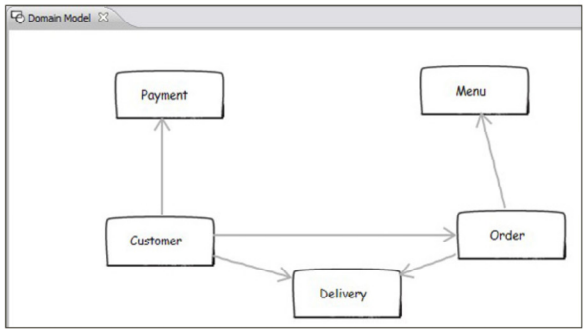
27

Do you know how your favorite email app, social network or online shop works? I encourage you to try to find additional examples on the technologies other companies use. Since companies usually try to hide how they work internally, this may not be an easy task.



Now, let's focus on the history of the ER model…

History: informal models



29

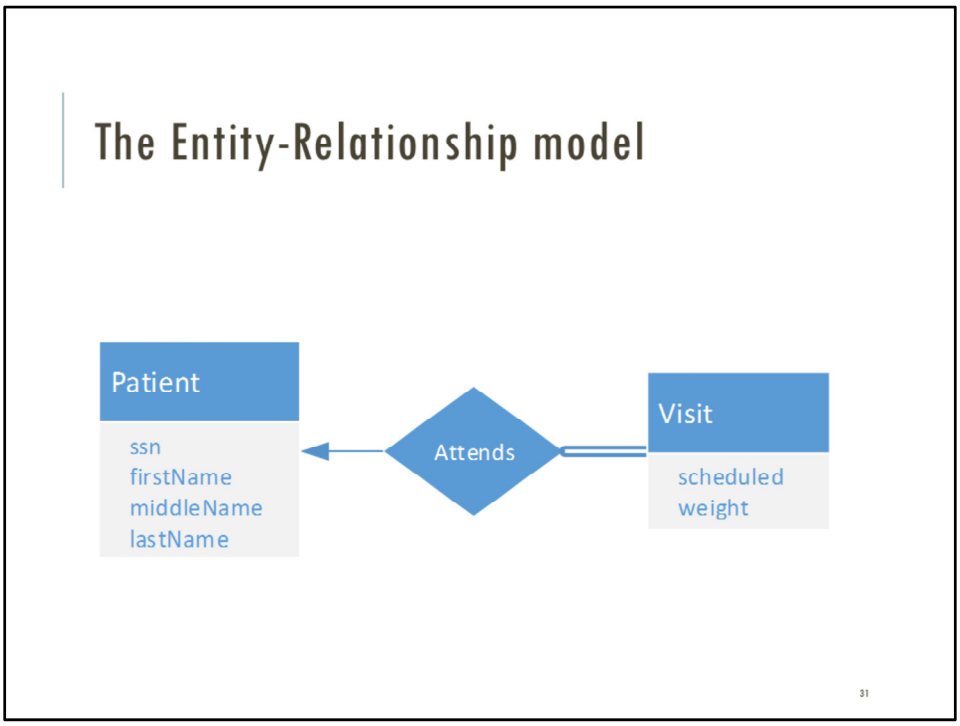
After the relational model was devised, there was a need for representing and designing how data were going to be stored. Usually, this representation was informally depicted and there was not clear its use. For instance, in this case, we have an informal representation of placing orders for customers but, what does it mean? What are the boxes and the arrows?

Peter P. Chen (1947 –)

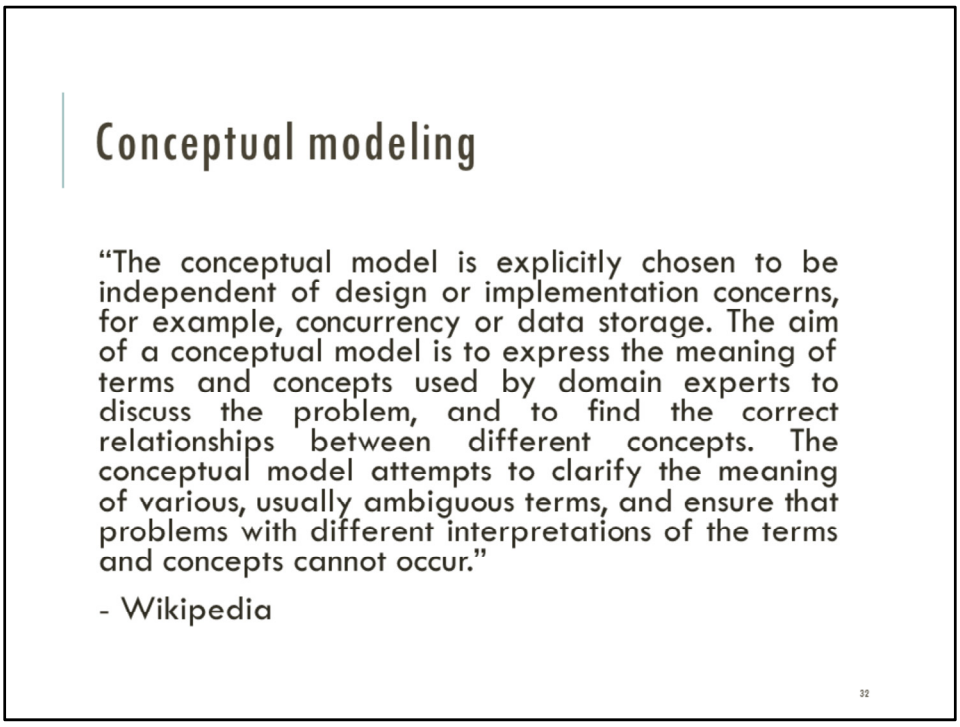
- Peter P. Chen. The Entity-Relationship Model - Toward a Unified View of Data. ACM Trans. Database Syst. 1(1): 9-36 (1976).
- MIT, Sloan School of Management

30

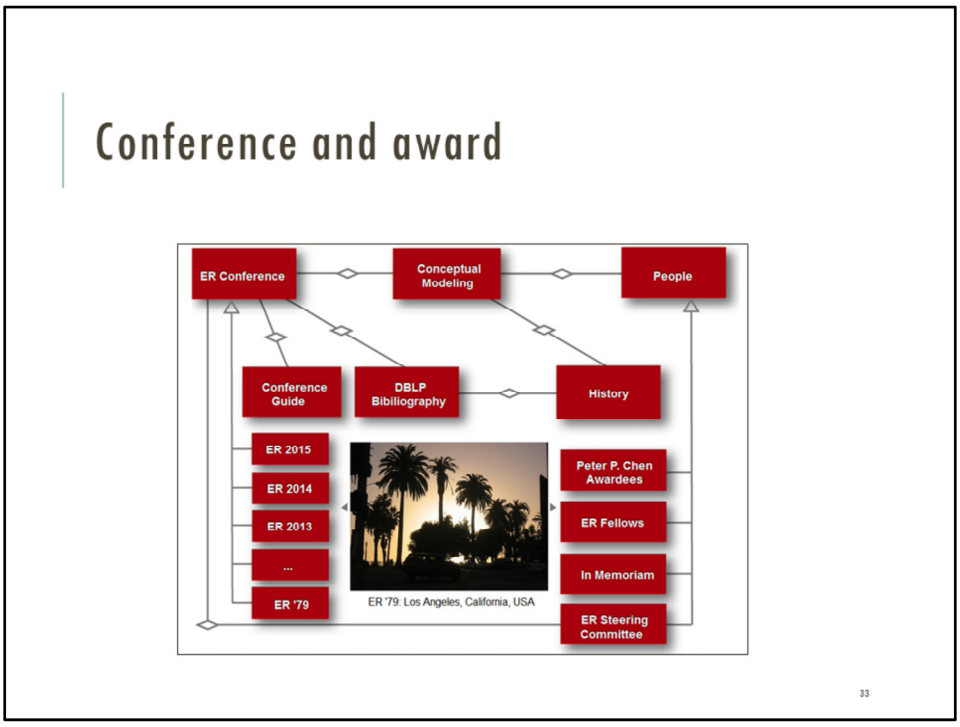
To overcome this problem, Peter Chen devised the Entity-Relationship (ER) model that was published in a paper called: “The Entity-Relationship Model - Toward a Unified View of Data”. He published this paper when he was a professor at the MIT Sloan School of Management.



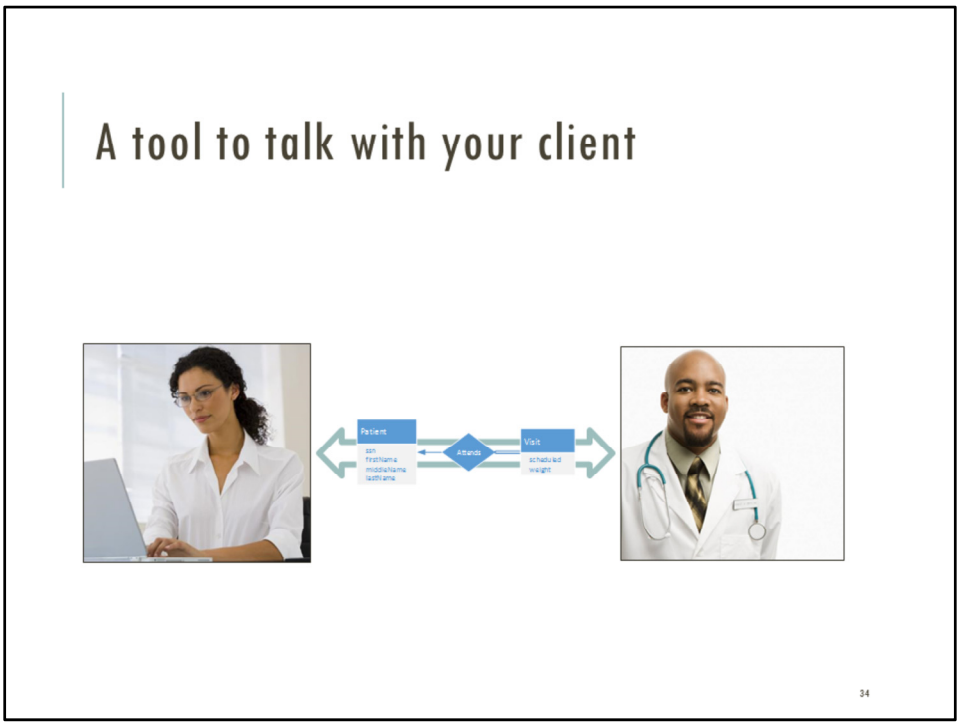
The ER model describes the data or information aspects of a domain in an abstract way, which can be ultimately implemented in a database like a relational database. It comprises entities and relationships among them. For instance, in the sample diagram of the figure, we have two entities (Patient and Visit) and one relationship (Attends). Additionally, we have different attributes for the entities like ssn or scheduled.



The work of Peter Chen is considered the pioneer in the field of conceptual modeling. According to Wikipedia: "The conceptual model is explicitly chosen to be independent of design or implementation concerns, for example, concurrency or data storage. The aim of a conceptual model is to express the meaning of terms and concepts used by domain experts to discuss the problem, and to find the correct relationships between different concepts. The conceptual model attempts to clarify the meaning of various, usually ambiguous terms, and ensure that problems with different interpretations of the terms and concepts cannot occur."



Since 1979, the International Conference on Conceptual Modeling gathers researchers from all over the world to discuss issues with conceptual models. There is also a Peter Chen award to honor excellent researchers/educators for outstanding contributions to the field of conceptual modeling.



A crucial property of conceptual models is that they are a perfect tool to talk with your client. Since we aim to clarify the concepts that domain experts use and the relationships among those concepts, we can easily use them to talk with the client and check if we are working in the right direction.

There have been several approaches to devise conceptual models that have their roots in the ER model. One of the most successful examples is UML, Unified Modeling Language, which is object oriented and also allows to specify concepts and their relationships among them. We will also study UML in this course.

Roadmap

1. History of relational databases
2. History of the ER model
3. **Our case study**
4. Conclusions

36

Let’s see our case study…

Hospital management

37

We are going to focus on hospital management that deals with the general management of a hospital. This case study is complex in practice (in the real-world), so we will simplify the problem a bit to deal with it. We will use our case study to discuss all the contents of this course.

People

- Doctor and patient
- Info: SSN, name, phone, address, birth date, gender, email, occupation
- Primary doctor
- Supervisor

38

We are going to store data of the doctors that work in the hospital and patients. Information that need to be accessed is the SSN, name, phone, address, birth date, gender, email and occupation. A doctor is also assigned as the primary doctor to each patient. Some doctors may supervise other doctors.

Visits

- Doctor sees patient
- Bill, diagnostic, prescription
- Info: when, height, weight, blood pressure, temperature, additional notes

39

We call a visit when a doctor sees a patient. A visit always generates a bill and may yield to a diagnostic and/or involve a prescription. The information that needs to be stored is the date of the visit, height, weight, blood pressure and temperature of the patient, and any additional notes that the doctor wants to add.

Bill, diagnosis, prescription

- Bill: billing date, due date, amount and payments
- Diagnosis: name and category
- Prescription: drug, amount, refill

40

Every visit generates a bill and we are interested in storing the billing and due dates, the amount and information about the payments, taking into account that a bill can be paid in more than one payment and one payment can be used to pay more than one bill. We wish to store the date of the payment, the amount and the method. A diagnosis can be categorized in some categories like teeth, infection or lung, and each diagnosis has a name like abscess or sinusitis. For prescriptions, we are interested in the name of the drug prescribed, its amount and whether or not it is a refill.

Insurance company

- A patient is insured by a company with a policy number
- Info: name, phone, fax, address

41

Finally, we are going to store the insurance companies that serve patients. Each patient has a policy number and we also wish to store the name of the company, phone, fax and address.

Roadmap

1. History of relational databases
2. History of the ER model
3. Our case study
4. **Conclusions**

42

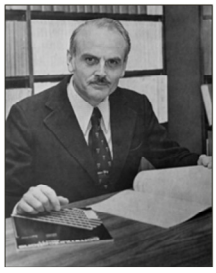
To conclude with these slides…

Relational databases

43

We have seen that the field of data management is extremely wide and we are going to mainly focus on data modeling using relational databases.

History of relational databases



44

We have studied the history of relational databases, which were devised by Edgar Codd and it is based on relations and keys. We have also studied how his paper was rejected at the beginning but, at the end, was very successful and revolutionary. We have also introduced the so-called NoSQL databases that we will study later in this course.

History of the ER model

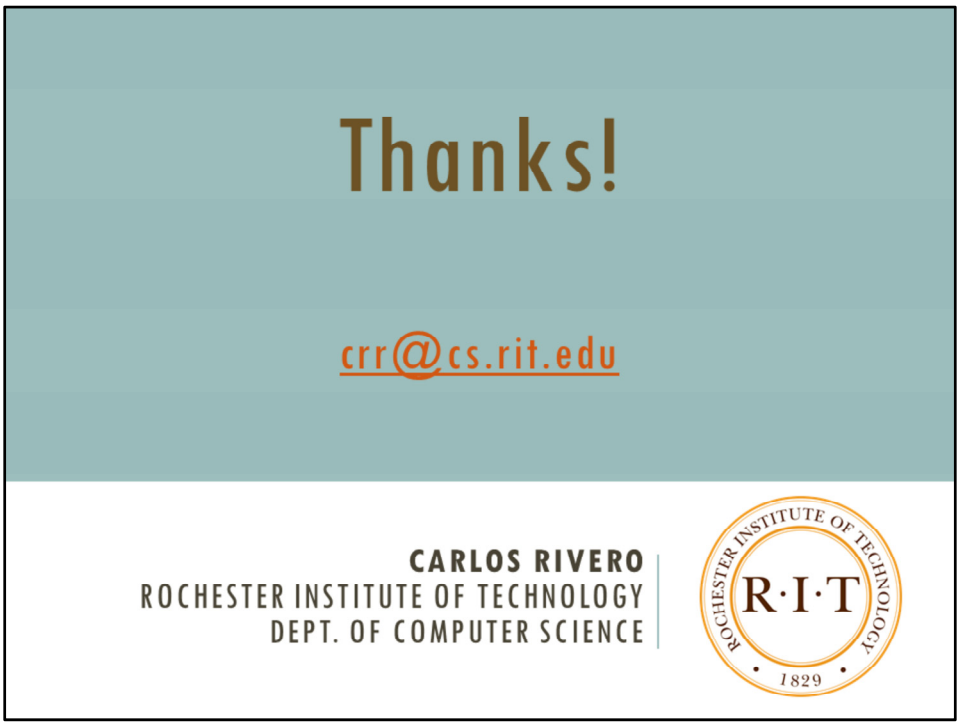
45

We have also studied the history of the ER model, which was devised by Peter Chen, and how this was the first step towards the new field of conceptual modeling, which is useful not only to represent how we are going to store some data, but also to talk with your clients. Several models were devised based on the ER model and we are going to study one of them: UML.

Case study

46

Finally, we have seen the requirements for the case study we are going to use during this course, which is based on hospital management.



Thank you very much.