

A decorative graphic on the left side of the slide, consisting of a network of white lines and circles on a blue gradient background, resembling a circuit board or a neural network.

WEEK 2

BEFORE THE RELATIONAL MODEL– SOME HISTORY

CS3319

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
 - List 2 of the models that were used before the relational model to build databases

THE PROBLEM:

- Now that we have figured out how to represent our data as a simple paper MODEL (the ER Diagram)..... How do we translate that ER model to a REAL working database on a computer?

A LITTLE HISTORY...

- The Hierarchical Model
- The Network Model

HIERARCHICAL DATA MODEL

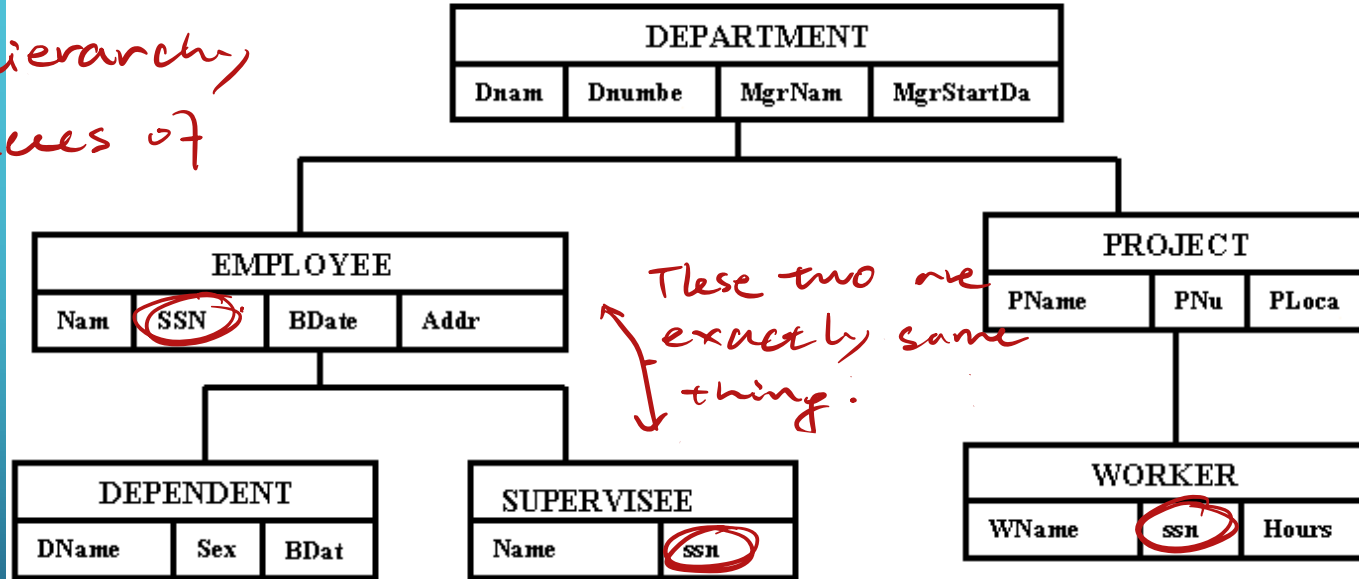
- Most popular DBMS using this model is IBM's IMS (STILL BEING USED!)
- Based on 2 main concepts: records and parent-child relationships. Records are grouped into record types
- A parent-child relationship type is a 1:M relationship between 2 record types
*i.e. Department - Employees
parent - children*
- An occurrence (instance) of a PCR is 1 parent record type and a number of records from the child type

Example:

tree-like structure.

But there's a problem that we could not get a clear hierarchy between the pieces of data.

Hierarchical Schema for part of COMPANY Database



Question: Can you see where there will be data redundancy (a problem)?

WHAT IS THE PROBLEM...

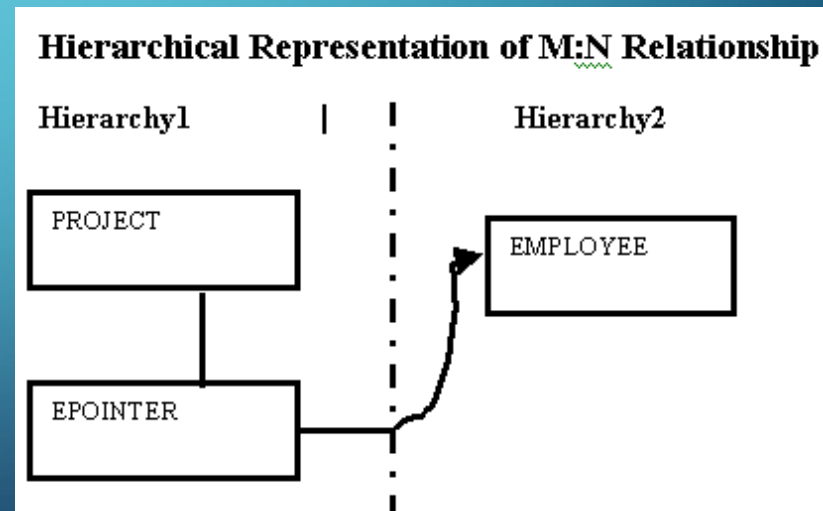
- with the Hierarchical Model?

Rules for Mapping ER Diagram to Hierarchical Model:

- In Hierarchical Models only 1:N representations can be modeled so for M:N relationships represent them as if they were 1:N relationships, and use pointers to save on duplication of data

Has problems modeling the following situations:

- M:N relationships
- The case where a record type participates as child in more than 1 PCR type
- N-ary relationships with more than 2 participating types



NETWORK DATA MODEL

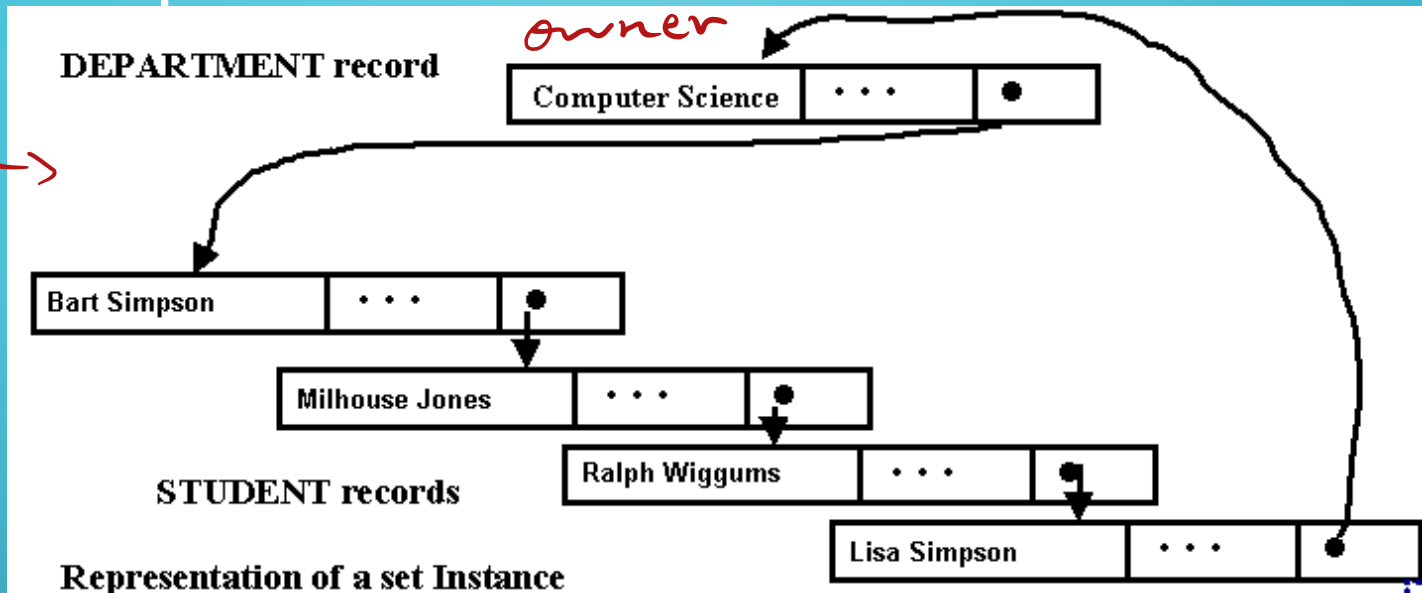
- Designed and created by the CODASYL (Conference on Data System Languages) committee in 1971 *separate data and code.*
- Often called the CODASYL Model (popular DBMS using this model is IDMS)
- Two basic data structures: Records and Sets
- Data stored as Records, Records as classified into Record Types (Record Types have a name and format for each data item)

- Set type is a 1:N relationship between 2 record types. Each set type consists of 3 elements
 - A name for the set
 - An owner record type
 - A member record type
- A record type can participate as an owner or member in any number of set types
- Uses circular linked lists

*Each record is a thing. i.e. a student.
a group of record form a set.*

Example:

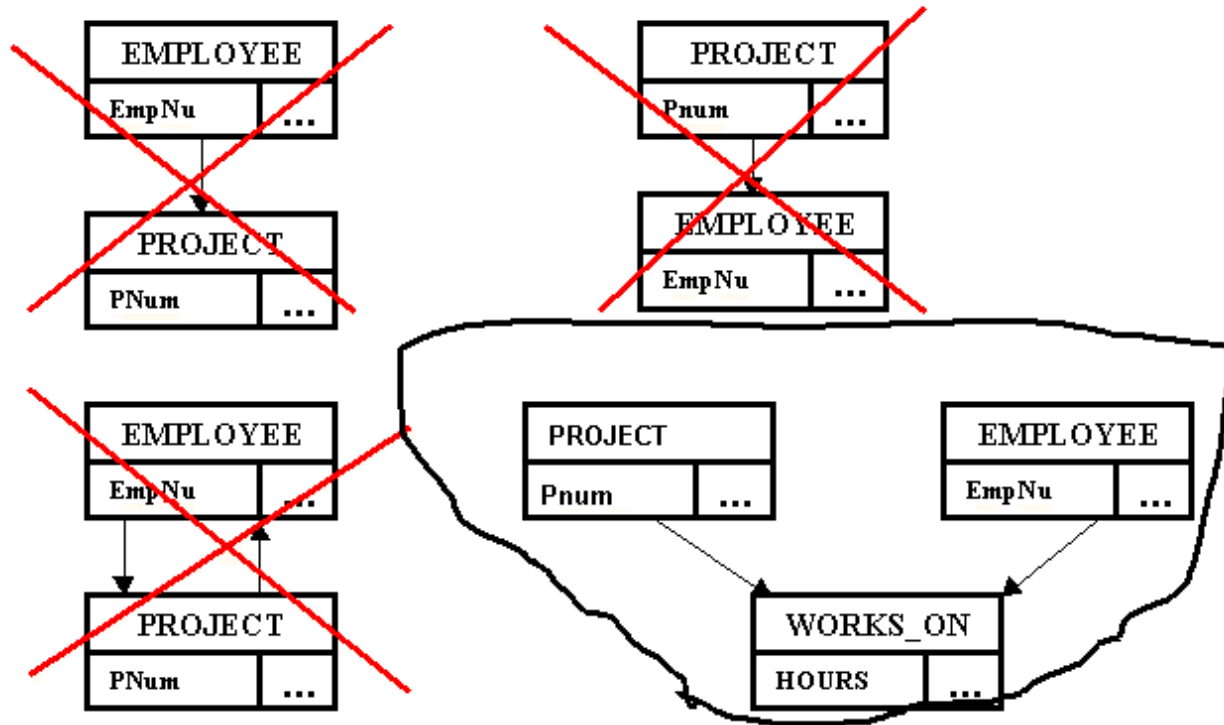
*Circular
Linked
List.*



This model allows us to do things such as:

- Given an owner record, find all member records of the set occurrence
- Given an owner record, find the first or nth, or last record of that set occurrence
- Given a member record, find the next or previous member record
- Given a member record, find the owner record

Representing M:N Relationships using sets



• Rules for Mapping an ER diagram to Network Model

- Step 1: Regular Entities --> Create a record type
- Step 2: Weak Entities --> Create a record type
- Step 3: One to One and One to Many --> Create a set type relating record type S1 to S2. For one-to-one arbitrarily choose one of S1 or S2 as owner, for one-to-many choose make the owner be the One side of the One-to-Many relationship
- Step 4: Many to Many --> Create a linking type X and make it the member record type in the 2 set types.

RELATIONAL DATA MODEL

- Introduced by Codd in 1970
- Most DBMS are based on this model
- Represents the database as a collection of relations.
- Each relation is a table: a table is a set of rows, where each row in the table represents a collection of related data values

*with large volume data, some companies
use no SQL.*