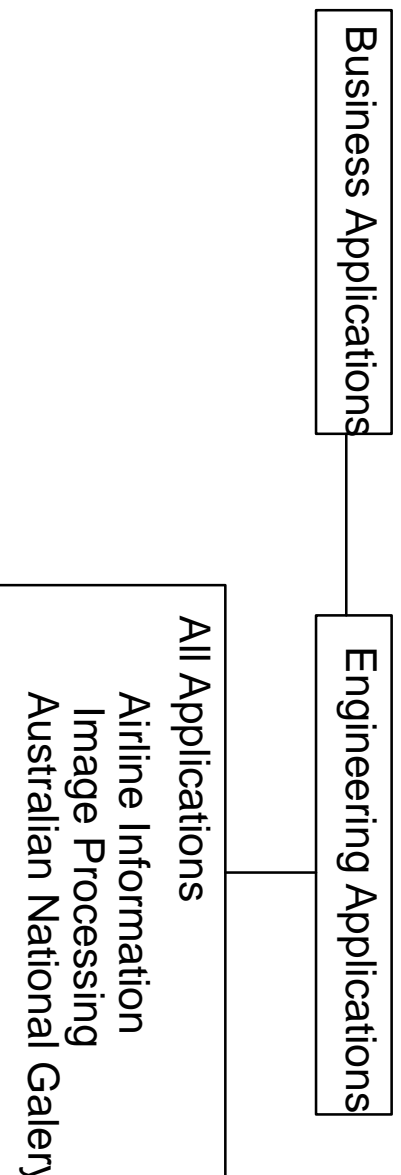


## *New Frontiers for Database Application*

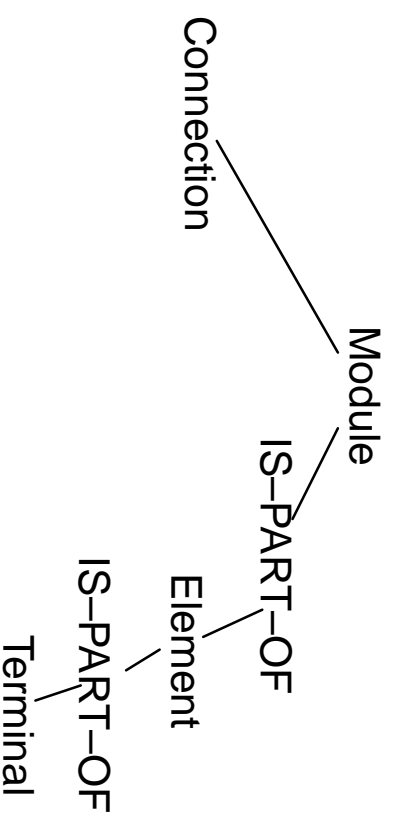
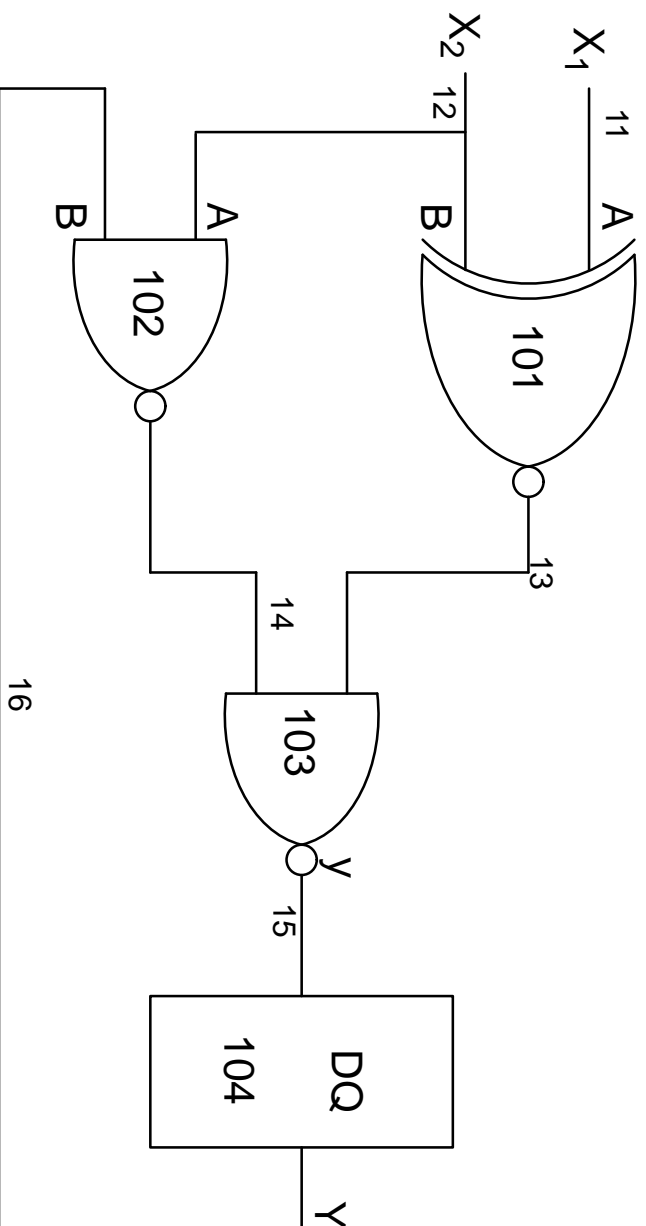
Changes in the focus of applications.



- The first generation: CODASYL, network data model; IMS, Hierarchical Data Model.
- The second generation: The Relational Data Model by Edgar F. Codd, 1970
- The third generation (Object–Oriented Data Model ?)
- Object Relational Database Systems

*Application Areas: Computer–Aided VLSI Design  
(refer to the readings)*

- Vast amounts of data encountered.
- A large number of iterations.
- A large number of primitive parts.
- Versioned Objects: A versioned object is an occurrence of a single object type.
- Instantiation: a copy of a prespecified cell. (sharing)



## Module

M-id	M-Name
7	Module-7

## Element

M-id	E-id	E-Name
7	101	2 inp. XNOR
7	102	2 inp. NAND
7	103	2 inp. NAND
7	104	Latch

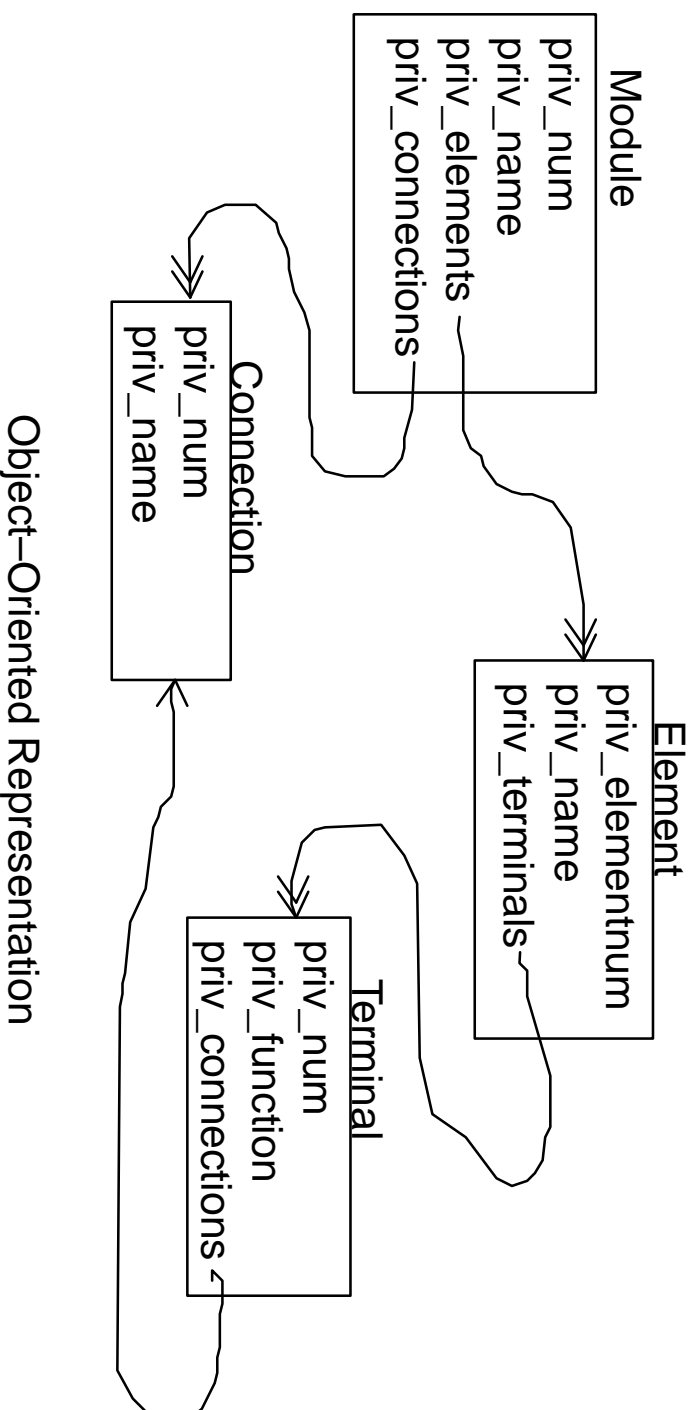
## Connection

<i>M-id</i>	<i>C-id</i>	<i>C-name</i>
7	11	X <sub>1</sub>
7	12	X <sub>2</sub>
7	13	
7	14	
7	15	Y
7	16	Y

*Terminal*

<i>T-id</i>	<i>C-id</i>	<i>T-name</i>	<i>Input/Output</i>
101	11	A	I
101	12	B	I
101	13	XNOR (A,B)	O
102	12	A	I
102	16	B	I
102	14	NAND (A,B)	O
103	13	A	I
103	14	B	I
103	15	NAND (A,B)	O
104	15	D	I
104	16	Q	O

- Problems
  - Join operation (performance)
  - Programming language vs Data Manipulation language Impedance—dismatch
  - Difficulties to update (exchange 101 for 103)



*New Applications non-traditional applications,  
engineering applications*

CAD/CAM. VLSI design, Geographic Information System, Image Processing, ...

- Hierarchal data structures
- Complex data structures
- No such general-purpose data structure available
- Different applications use different data structures
- Trial and error design process
- Dynamic changes
- Cooperative design process among designers
- Large number of types
- Small number of instances

## *The Relational Database Approach*

The Advantages of Relational Database Systems

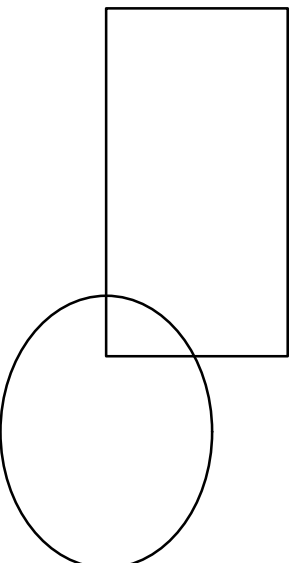
- Simplicity/Flexibility
- Set Theory
- High Data Independence
- Non-procedural Access, SQL

RDBMS's are good for traditional (business) applications such as Bank Management, Inventory Management.

## *Problems with Relational DBMS*

- Need to decompose application objects over relations.
- No one-to-one correspondence between application objects and database objects.
- Fixed built-in types. No set-valued attributes
- Homogeneous collection types
- Identifier Attributes: Uniqueness, Referential Integrity, Operations
- No data abstraction such as aggregation and generalization.
- Modelling actions
- Impedance-dismatch
- Application-specific access methods
- Version-control

## Computationally complete DML



Circle

Center	Radius

Rectangle

X1	Y1	X2	Y2

Select \*  
from Circle, Rectangle  
where Radius < 1.5  
and Overlap(Circle, Rectangle)  
???