



**CHAITANYA**

**WOMEN'S COLLEGE**

DEGREE & PG

AFFILIATED TO ANDHRA UNIVERSITY

# **DATABASE DESIGN**

UNIT

3

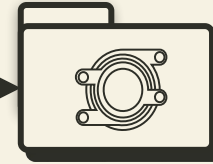
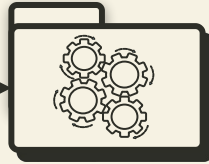
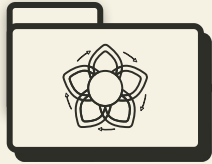
01

# DESIGN PROCESS





# Database life cycle



**Phase 1**

Requirement  
analysis

**Phase  
2**

Logical design

**Phase 3**

Physical  
design

**Phase  
4**

Database  
implementation,  
monitoring,  
and  
modification.



# 1. DATABASE INITIAL STUDY

- Requirement collection & analysis
- Dbms Software Selection

Factors affecting purchasing decisions

- Cost
- DBMS feature and tools
- Underlying model
- Portability
- Hardware requirements



## 2. LOGICAL DESIGN

- Most critical phase of DBLC
- Makes sure the final product meets requirements
  - A. Conceptual data modeling
  - B. View integration
  - C. Transformation of the conceptual data model to SQL tables.
  - D. Normalisation of tables



## a. CONCEPTUAL DESIGN


- Data modelling creates abstract data structure to represent real world items.
- High level abstraction
- Steps:
  - data analysis & requirements
  - er modelling and normalisation
  - data model verification
  - distributed database design



# ER MODEL

1. The major activity of this phase is identifying entities, attributes, and their relationships to construct model using the Entity Relationship Diagram.

a. Entity  $\Rightarrow$  table



Entity

b. Attribute  $\Rightarrow$  column

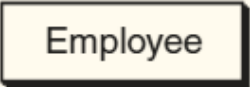
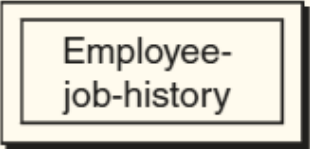



Attribute

c. Relationship  $\Rightarrow$  line



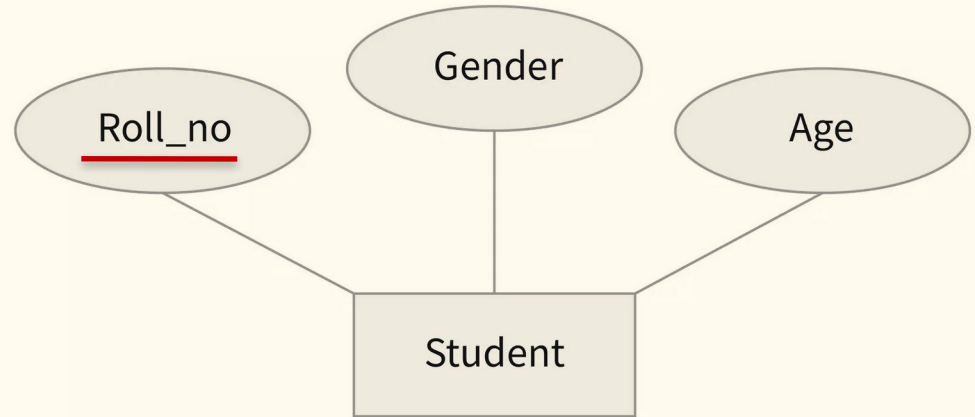
relationship

Concept	Representation & Example
Entity	
Weak entity	
Relationship	



## CLASSES OF ATTRIBUTES

1. Simple attribute
2. Composite attribute
3. Derived attributes
4. Single-valued attribute
5. Multi-valued attribute
6. Key attribute



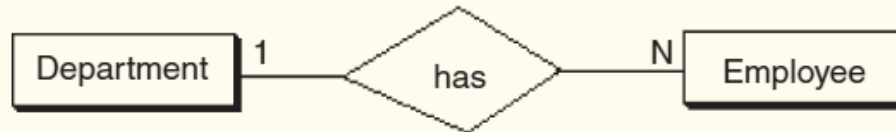
Note: Click on the names of the attributes

**Connectivity**

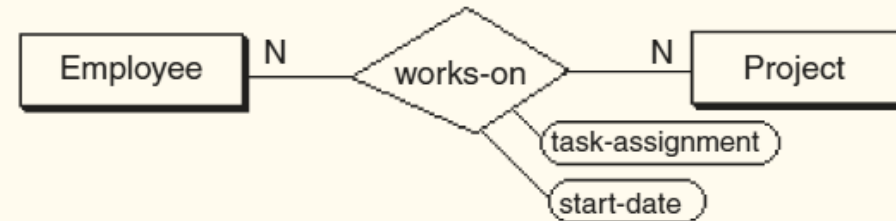
one-to-one

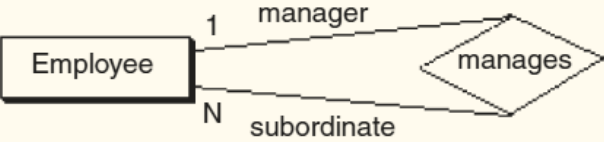
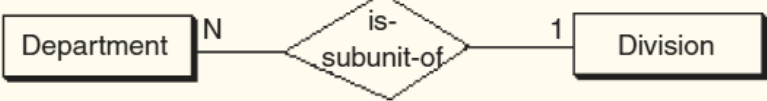
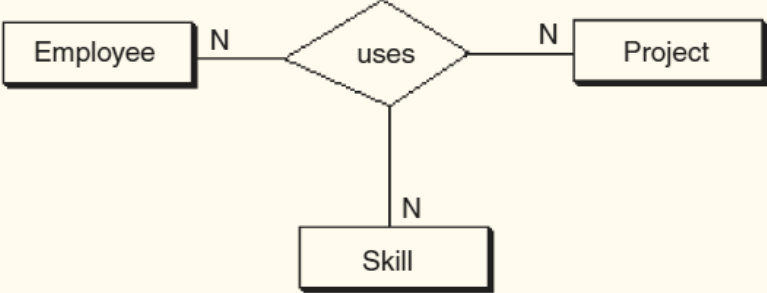


one-to-many







many-to-many

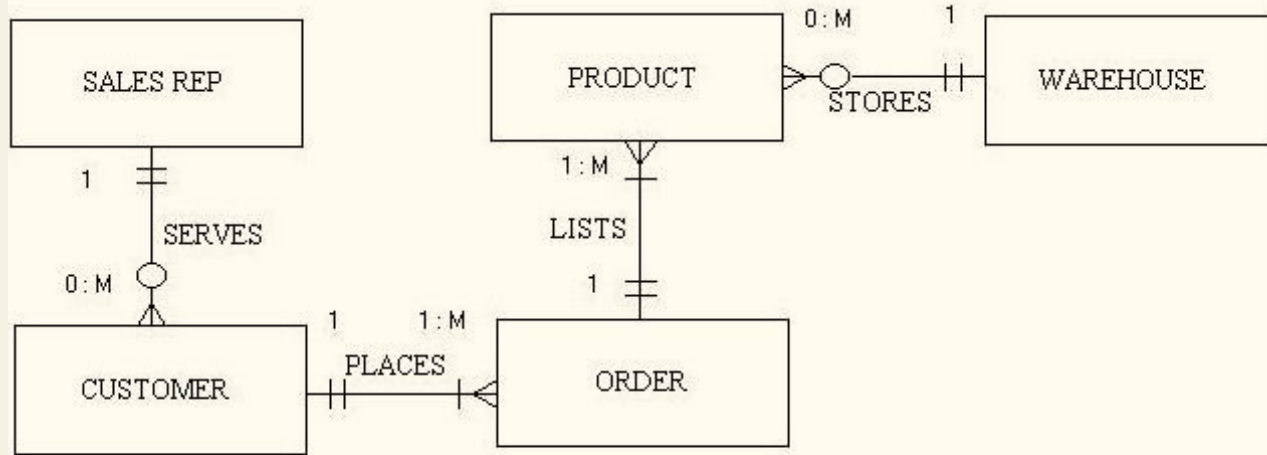


Concept	Representation & Example
Degree recursive binary	
binary	
ternary	

# CARDINALITY

Symbol	Meaning
	Mandatory—One
	Mandatory—Many
	Optional—One
	Optional—Many

## Example:



**Figure 1. Entity-Relationship Diagram**

- \* 1 INSTANCE OF A SALES REP SERVES 1 TO MANY CUSTOMERS
- \* 1 INSTANCE OF A CUSTOMER PLACES 1 TO MANY ORDERS
- \* 1 INSTANCE OF AN ORDER LISTS 1 TO MANY PRODUCTS
- \* 1 INSTANCE OF A WAREHOUSE STORES 0 TO MANY PRODUCTS



## b. VIEW INTEGRATION

- To eliminate redundancy and inconsistency
- Views must be “rationalized” and consolidated into a single global view.
- Uses ER semantic tools such as identification of synonyms, aggregation, and generalization

A stylized window with a title bar and a list of bullet points. The window has a white background and a black border. The title bar contains three small circles (yellow, white, white) on the left and the text "c. CONCEPTUAL DATA MODEL TO SQL TABLES" in black. The main content area contains two bullet points: "- Data modeling constructs" and "- Redundant tables are eliminated". There are several rectangular tabs or folders of various sizes and positions around the window, including one on the top right, one on the left side, one on the bottom left, and one on the bottom right.

## c. CONCEPTUAL DATA MODEL TO SQL TABLES

- Data modeling constructs
- Redundant tables are eliminated

A stylized window with a white background and a black border. The title bar at the top contains three circles (one yellow, two white) on the left and the text "d. NORMALIZATION" in bold black font. Below the title bar, there is a list of two bullet points: "- Organisation of data" and "- Reduces data redundancy". The window has several overlapping rectangular shapes on its left and right sides, suggesting it is part of a desktop environment.

## d. NORMALIZATION

- Organisation of data
- Reduces data redundancy





## 3. PHYSICAL DESIGN

- Selection of data storage and access characteristics
- Becomes complex for distributed systems
- Designer favors the software that hides physical details



## **4. DATABASE IMPLEMENTATION, MONITORING, AND MODIFICATION.**

**4. a. IMPLEMENTATION & LOADING**

**4. b. TESTING & EVALUTION**

**4. c. OPERATION**

**4. d. MAINTANCE & EVALUTION**



# QUERY LANGUAGES

- Primarily created for creating, accessing and modifying data in and out from a dbms.
- Types:
  1. procedural language
  2. non- procedural language



# EMBEDDED SQL

- Embedded sql applications connect to databases and execute embedded sql statements.
- Advantages:
  - lets access to db from anywhere,
  - allows integrating authentication service for large scale apps
  - provides extra security to database transactions
  - avoids logical errors
  - easy to integrate frontend and backend of our applications



# DATA INDEPENDENCE

- Ability to modify a scheme such that it doesn't affect the schema in the next higher level

- Types:

1. physical data independence

change the capacity to change the internal schema without having to change the conceptual schema.

2. logical data independence

change ability to change the conceptual schema without having to change the external schema.