Database management system – software system allowing users to create/maintain a database

DB – Is a model of structures of reality

Physical data independence allows the DB admin to change the internal schema without changing the conceptual schema

Logical data independence allow the DB admin to change the conceptual schema without affecting external schema/application programs.

Database admin	
Can change internal schema w/out changing conceptual schema	

Database approach
Different users can see different views Info describing the DV us stored in a catalog Can be shared among many viewers

Implementation data model – relational data model (not entity relationship model)

A Database Schema…True	A Database Schema…False
Is the structure against which DB queries are written to access the DB state	DB state is relatively stable over time while the DB schema changes over time
Describes the structure of the data	
Relatively stable over time while the DB state changes over time	

Three-schema DB architecture: external, conceptual, and Internal schemata

A surrogate

1) Is a system generated artificial identifier/ unique internal identifier
2) Represents an entity of the real world inside the DB
3) Is immutable but the app programs
4) Value cannot be changed by the user

DB Integrity Constraints

1) The name of a customer must be less that 30 alpha chars in length
2) The name of the day must be one of Monday, Tuesday, Wednesday, Thursday…
3) Two different entities in the real world must have different identifiers in the DB

Concepts of integrity and consistency

1) Integrity guarantees DB reflects the world it models
2) Consistency guarantees the DB is free of internal conflicts

Poor consistency in a DB

1) Storing street, city, state and zip as well as an address string
2) Storing both DOB and Age to save computing time for age

EER - Entity types, relationship types, and properties

True	False
Relationship type names need only be unique if there are more than one relationship between a same pair	All relationship type names must be unique
All entity names must be unique	All property names must be unique

NULL values

1) Inapplicable values
2) Unknown values

Key attribute/partial identifer = oval with underlined name

Composite attribute = many other attributes. Oval with many other ovals

Multi value attribute = double oval

Derived attribute = dotted oval

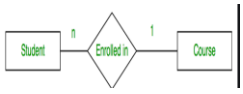
Double line = total participation / single = partial participation

1 to 1 – cardinality is 1:1

a male can marry to one female and a female can marry to one male

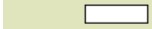
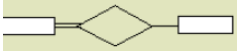
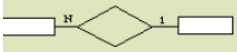
Many to 1 – cardinality is many:1

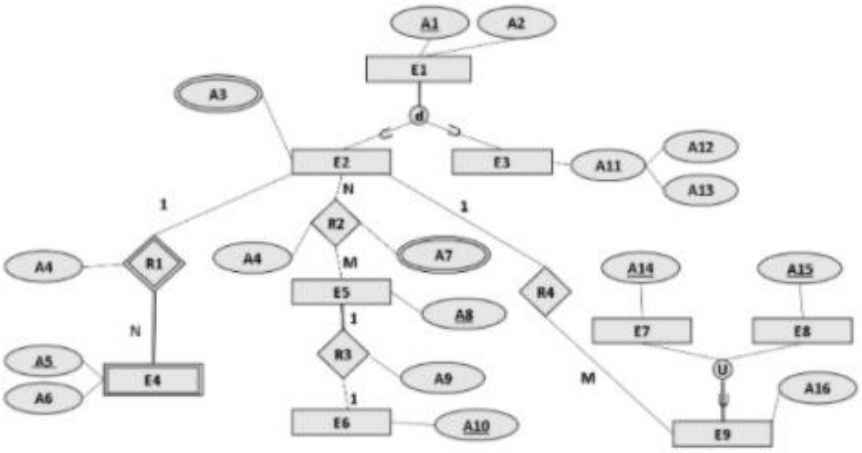
a student can take only one course but one course can be taken by many students



Many to many – cardinality is many: many

a student can take more than one course and one course can be taken by many students

Notation	Meaning
	Entity type
	Attribute
	Key attribute
	Derived attribute
	Multivalued attribute
	Composite attribute
	Relationship type
	Total participation
	Many-to-one relationship



Entity Type E4:

Instances of E4 are…

1) Identified by pairs o property values from A1,A5
2) Identification dependent on instances of E2
3) Existence dependent on E2

It is a weak entity type

If there are 20 instances of E4, any number equal or greater of E2 are possible

If there are 20 instances of E2, 0 instances of E4 are possible

E1 and E3 identifier key = A1

E4 identifier/key = (A1,A5)

All instances of entity type E9 are identified by values of A14 or A15

Union Entity type E9:

True	False
E9 is a subset of the union of E7 and E8	Some instances of E9 are instances of both E7 and E8
An instance of E9 is also and instance of E7 or E8 but not both	The union of E7 and E8 Is equal to E9
	The union of E7 and E8 Is a subset to E9

Multi valued properties: A3, A7 (double oval)

Composite properties: A11

Super-subtype relationship type:

1) Every instance of E2 must be related to an instance of E1
2) Every instance of E3 must be related to an instance of E1
3) Every instance of E1 must be related to an instance of E2 or E3, but not both

R1:

1) Every instance of E4 must be related precisely to on instance of E2
2) An instance of E2 may be related to multiple instances of E4

R3:

1) Every value of A8, there must be 1 related value of A10 through R3
2) Every value of A8, must be 1 related value (NULL okay) of A9 through R3

R4:

1) Each instance of E2, there is 0,1, or many instances of E9 via R4
2) Each instance of E9, there is at most 1 instance of E2 via R4

Partial identifiers: A5

****

Plane-type	Model	
Employee	SSN	
Service	Reg #, Date/Workcode	
Owner	SSN or Name	

FLIES	Model and SSN	
UNION	Each owner is either a CORPORATION or a PERSON	
Date/workcode	Composite attribute	
Pdate (PurchaseDate	None of the answers	

10 instances of plane-type and 25 instances of pilot

1) Max # of instances of FLIES = 250
2) Min # of instances of flies = 0

10 instances of service

1) Max # of airplanes possible serviced = 10
2) Min # = 1

# of nicknames can Person have = 0,1,..many

# of Airplanes one person can own = 0,1…many

# of HANGARs can an AIRPLANE be STORED-IN = 0 or 1

True	False
A PILOT can OWN an AIRPLANE	a CORPORATION and a PERSON OWN an AIRPLANE together
The PILOT who FLIES a PLANE-TYPE can be the same PERSON as the EMPLOYEE who MAINTAINs AIRPLANES of that PLANE-TYPE.	If all SERVICE records for a particular AIRPLANE are deleted from the database, then that AIRPLANE must be deleted too
Given a particular SERVICE instance, do we know which HANGAR the AIRPLANE being SERVICED is STORED-IN	
All SERVICE records for that airplane are deleted If an airplane is deleted	

Change the EER Diagram to prevent a pilot from also being an employee = Define PILOT and EMPLOYEE as a disjoint specialization of PERSON