PAGE No.
DATE / /

Section 2 → ER Diagram.

Attributes
Mandatory (*) OR optional (o)
volatile OR nonvolatile (Stable)
single OR composite. → can be divided further.

(1) ⇒ How to show database design to clients. ER Model.

(2) ⇒ Entity Relation Model.

Key terms are Entity & Attributes. & (Relationships b/n entities).
↓ ↓
Employee Salary, Name, etc. } Eg:-

Department Entity ← Employee → works in → Department

Attributes ← *EName, *Salary, #Id

Department: #DName, *Location, oNo. of emp

→ UID
* → Mandatory
o → optional.
} ER diagram.

(3) Relationships can be 1:1, one to many, many to many.

} Representations are
|| — exactly one
|o — zero or one
>| — one or more
>o — zero, one OR more

See Checksheet on Net for more & P.T.O.

Eg:- One computer assigned to only one employee.
But one employee manages 3 computers.

Components of relationship
1) No. of relations
2) * (or) (O)
3) Name of rel.

Employ — || — ALLOT_DT — Computer

(1:1).
A relationship can also have an attribute. This will be stored in table where foreign key is stored.

Note:-
* In 1:many relationship foreign attribute always added to many table.

Note 2:- In M:N, completely new table is drawn for relationships.

DATE / /

Section 3 → Normalization.

(1) ⇒ Data Redundancy → same data multiplied across the database.

Eg:-

Emp Name	Numb	Location centre
~	~	Goa
~	~	Goa
~	~	Goa

here location centre is data redundancy. (Taking extra space). even when same value for everybody

(2) ⇒ Data inconsistency → Anything affecting data integrity.

Eg:- 2 tables, having same attribute data but diff attribute.

Eg:- retailoutletaddress — Same data

retailoutletlocation. — Same data.

(3) ⇒ Normalization → Reorganizing database to ensure there are no redundancy (or) data dependency.

⇒ Different stages of normalization are known as 'normal forms'. first we have to understand concept of functional dep.

(4) ⇒ Functional dependency → A functional dep relational can exist b/n 2 (or) more attributes.

3 types → fully functional dependency.
→ Partial dependency.
→ Transitive dependency.

P.T.O.

DATE / /

→ Attribute B is functionally dependent on attribute A if each value of A determines exactly one value of B.

Representation $A \rightarrow B$

A → Determinant

B → Dependent.

→ $A \rightarrow B \rightarrow C$ ∴ transitive dependency.

$A, B \rightarrow C$ ∴ fully $\quad A \rightarrow C$ ∴ partial (B)

(5) Anomaly → Unexpected side effect from trying to insert, update (or) delete a row.

Mainly bez of functionally dep b/n attributes.

(6) Normal forms

→ Diff stages → 1NF — 6NF

After 3NF, BCNF is there befor 4NF

(Boyce - Codd Normal form).

1NF → All attributes of R are atomic in nature

(Removes ambiguity) Should be no multi-valued attribute.

↳ An attribute which can have multiple values at one time.

2NF → 1NF nature present

(Removes data redundancy) No partial dependency (b/n non-key & key attributes).

3NF → Already 2NF.

(data integrity) No transitive dependency which exist b/n key and non-key attributes.

2NF → preferred in applications with extensive data retrival.

3NF → " " " " " data modifications.

conversion from higher to lower form is denormalization.

---

(7) Data Models
- Conceptual Model → identifies table names & relationship b/n them
- Logical Model → also called entity relationship model.
- Physical Model → Logical + index, views, datatypes, etc.

↳ First thing to be created while designing DBS

Logical (ER) Diag ⟶ Physical. (ER implement. into keys, Tables, columns)

(8) Entity
- → Prime (Customer) → Independent
- → Characteristic (Order) → exists bcz of prime
- → Intersection (OrderItem) → bcz of both prime & char.

Instance → Occurance of entity (eg:- JOB (entity) → Cleaner (Instance))

(9) 3 Data model drawing notation
- → Barker notation
- → Bachman notation
- → Information Engineering notation.
