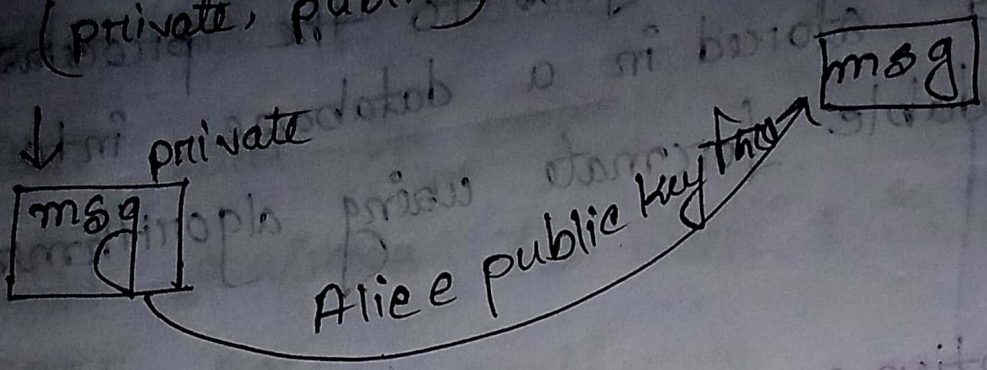

Segment-2

Functional Dependency & Normalization:

Functional Dependency:

- It is a relationship status between attributes
- It indicate through (⟶) arrow sy
- determinant → dependent

Ex: $x \rightarrow y$ (y depended on x)

x	y = 2x
1	2
2	4

⇒ x → y

id	Name	age
1	x	11
2	y	12
3	x	13
4	z	13

ID → Name ✓

Name → id X

ID → Name ✓

ID → age ✓

Functional Dependency: Benefits/uses of functional dependency:

→ Prevent Data Redundancy
→ Maintain quality and integrity of data.
→ Reduce the risk error.
→ Define meaning and constraints of database.
→ identify poor design.
→ Gain productivity and save costs.

Types of functional Dependency:

1) Trivial functional Dependency
2) Non Trivial functional Dependency
3) Multivalued functional Dependency
4) Partial functional Dependency
5) Transitive functional Dependency

Trivial FD:

A trivial dependency refers to a type of functional dependency refers to a functional dependency between two attributes (columns) in a relation (table) that is somewhat self-evident or obvious.

(dependency এরর determinant এর subset হয়)

Id	Name	age
1	X	12
2	Y	13
3	X	13

Id, Name → Name

↓

Composite key এর মতো অনেক

Non-Trivial FD: A non-trivial functional dependency is a concept related to the relationships between attributes in a database table.

(dependency তথ্য determinant এর subset হয় না)

ID → Name

Multivalued FD:

car_model	name	color
C001	Toyota	Blue
C002	Toyota	Green
C001	Mari	Red

car_model →→ name

car_model →→ color

A multivalued functional dependency is a concept in database theory that extends the idea of functional dependencies.

Partial dependency: It is a concept in database and normalization that occurs when an attribute in a relational database depends on only part of a multivalued candidate key (partial key) rather than the

entire candidate key.

(student_id, zid_code) → student_nam

Transitive dependency: Transitive dependency is a concept in database normalization that occurs when an attribute in a relational database depends on another attribute through a chain of dependencies. (যখন দুইটি key এর মধ্যে dependency থাকবে।)

Author_id	Author_id	Book	Author_nationality

Book → A_n → A→B

A_n → A_nationality → B→C

Book → Author_nationality → A→C

Closure of Functional Dependency:

The complete set of all possible attributes that can be functionality derived from given functional using Armstrongs rules.

Armstrongs rules: It provides a simple technique from reasoning functional dependency.
(dependency বের করার জন্য যে rules use করলো তাদেরকে বলে Armstrongs rules.

Rules:

Primary

① Reflexivility: If α is a set of attributes and $\beta \subseteq \alpha$ then $\alpha \to \beta$ holds.

② Augementation: If $\alpha \to \beta$ holds and γ is a set of attributes then $\gamma\alpha \to \gamma\beta$ holds.

③ Transitivity: If $\alpha \to \beta$ holds and $\beta \to \gamma$ holds, then $\alpha \to \gamma$ holds.

Secondary

④ Union: If $\alpha \to \beta$ holds and $\alpha \to \gamma$ holds, then $\alpha \to \beta\gamma$ holds.

⑤ Decomposition: If $\alpha \to \beta\gamma$ holds then $\alpha \to \beta$ holds and $\alpha \to \gamma$ holds.

⑥ Pseudo transitivity: If $\alpha \to \beta$ holds and $\gamma\beta \to \delta$ holds then $\alpha\gamma \to \delta$ holds.

closure of Functional Dependency
- set of attribute
- derive from functional Dependency

R(A, B, C, D)

Functional Dependency = {A → B, B → C, C → D}

closure of A, A^+ = {BCDA} → candidate key

closure of B, B^+ = {BCD}

closure of C, C^+ = {CD}

closure of D, D^+ = {D}

↓
যে closure এ সব attribute থাকবে

Most Imp:

Normalization:

→ Normalization is the process of organizing the data in database.

→ use to reduce data redundancy.

→ divides larger table into smaller table & links them using relationship.

→ A form of decomposition

→ Reduces anomalies (insert, update, delete)

→ Stage to do the normalization is called Normal form (NF)

Types of NF
- 1NF
- 2NF
- 3NF
- 4NF / BCNF (Boyce code normal form)
- 5NF

1NF: A relation schema is said to be 1NF if the values of each row is atomic.

2NF: A relation schema is said to be 2NF if it is a 1NF and have no partial dependency.

3NF: A relation schema is said to be 3NF if it is a 2NF and have no transitive dependency.

4NF: A relation schema is said to be 4NF if it is a 3NF and have no multivalued dependency.

5NF: A relation schema is said to be 5NF if it is in 4NF or BCNF and aggregation of decomposed table is loss less.

1NF:

Sid	Sname	cid	mobile
501	X	C01	017..

2NF:

Sid	cid

Sid	Sname	mobile

3NF:

Sid	Snam

Sid	mobile