→ functional dependency

→ schema

→ axioms of functional dependency

→ DB keys

→ Normalisation

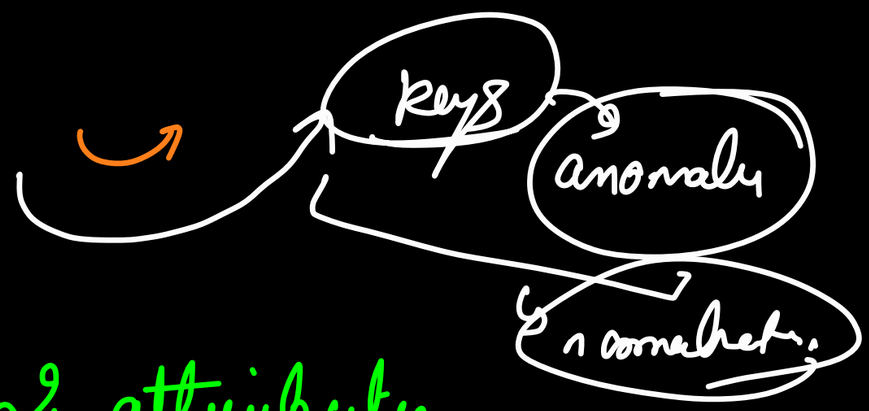
Database Schema → blueprint

blueprint of the actual db, that we will create. How we will store data, structure of table etc is defined in it.

cols → attributes

rows → tuples

Functional Dependency



It defines relationship between 2 attributes

$X \longrightarrow Y$ → Y is functionally dependent on X

X → determinant

Y → dependent

Y depends on X, means that, for every valid value of X we can uniquely identify Y.

e_id → e_name ✓

e_id → e_salary ✓

(e_name → e_salary) ✗

employee table

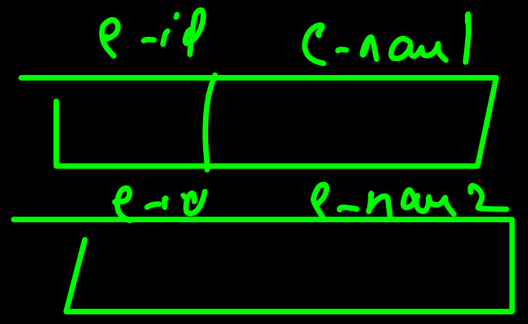
e_id	e_name	e_salary
1	abc	1000000
2	def	50000000
3	abc	8000000

e_id	e_nam	e_salary

verify the f.d ??

```
CREATE ASSERTION emp-name
 CHECK (NOT EXIST ( SELECT *
 FROM Emp As E1, Emp As E2
 WHERE E1.e-id = E2.e-id AND
   E1.e-name <> E2.e-name ))
```

SQL Standards



IBM DB2

CONSTRAINT emp-name CHECK (e-name) DETERMINED BY e-id;

Axioms / Rules

Reflexivity

↳ address
↳ state
↳ H-NO

state ⊆ address → address ⟶ state
hno ⊆ address → addrs ⟶ hno

Augmentation → Partial dependency

if $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any attribute Z

then there is Partial dependency

→ Foreign key

c_id	c_name	c_age	P_id	c_addrm
1	Abc	21	11	a-1
2	dy	21	12	a_2
3	gui	21	13	a-3
2	dy	21	14	a-2
3	ghi	21	12	a-3

c_id ⟶ c_name

(c_id, p_id) → c_name

} RHS should be completely dependent on LHS, not even part of it

Transitivity

if $X \rightarrow Y$ and $Y \rightarrow Z$ then

$X \rightarrow Z$

→ AI comm.

e-id	e_name	e_zip	e_state	e-phn
1	abc	12345	S_1'	⟶
2	dey	12345	S_1'	⟶

e_id → e_zip

e_zip → e_State

⟹ e_id ⇒ e_State

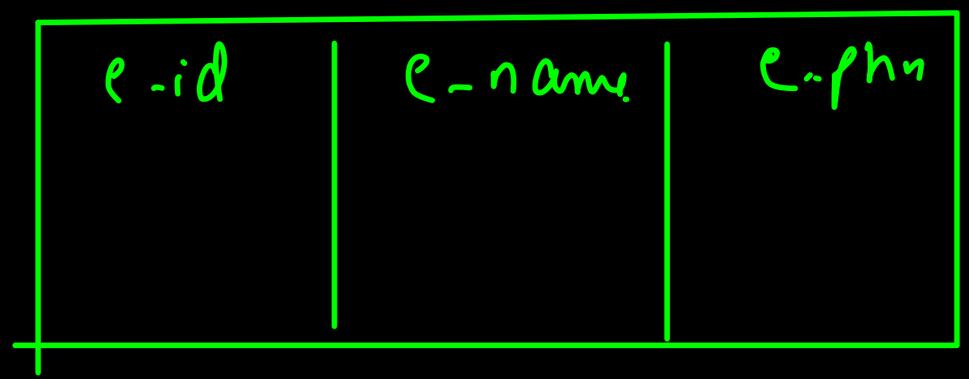
DB Keys

Keys are set of attributes that helps us to uniquely identify a record in diff situation.

- ↳ Super
- ↳ Composite
- ↳ Candidate
- ↳ Primary
- ↳ Foreign.
- ↳ alternate

Super

↳ a set of attributes within a table that can uniquely identify a record



{ e_id
(e_id, e_name)
e_phn
(e_phn, e_name)
⋮
}

Candidate

↳ minimum set of attributes that can uniquely identify a record.

e_id ~
e_phn ~ } these are 2 diff candidate keys for the table.

Composite → A key that consist of 2 or more than 2 attributes, that together uniquely identify a record. The attributes that form composite key are not any key independently.

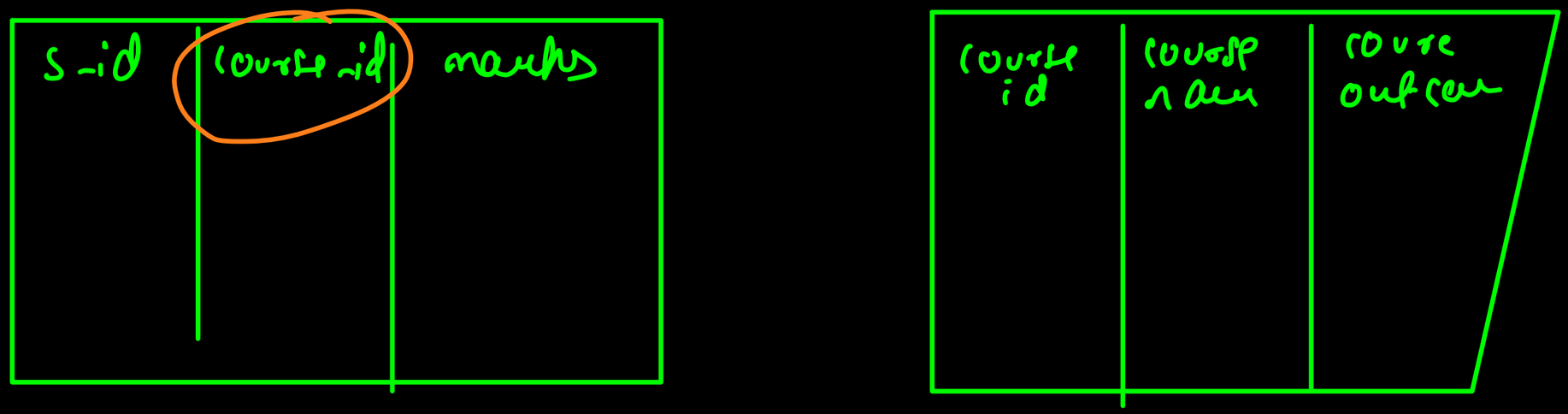
S_id	Course_id	marks
1	1	80
1	2	70
2	1	50
2	3	30

(S_id, Course_id)

Primary key → There can be more than one candidate key We can choose any one non-null candidate key to become primary key.

Alternate key ⇒ all candidate keys apart from primary key are alternate keys

Foreign key -> it is an attribute which is primary key in some other table.



Q1 →

$R = \{A, B, C, D, E\}$

$A \to B$

$A \to E$

$C \to B$

$C \to E$

$B \to D$

Calc → Candidate key

Q2 →

$R = \{A, B, C, D\}$

$AB \to C$

$BC \to D$

$CD \to A$

$A^+ \to \{A\}$

$AB^+ \to \{C, D\}$

$BC^+ \to \{E, D, A\}$

f0 → closure → f^* → it contains all the rules implied by a fd.

$f \rightarrow \{ A \rightarrow B, \quad C \rightarrow D, \quad B \rightarrow C \}$

$f^* \rightarrow \{ A \rightarrow A, A \rightarrow B, B \rightarrow C, A \rightarrow C, C \rightarrow D, A \rightarrow D, B \rightarrow D$
$\quad \ldots\ldots \}$

↳ attribut closure :→ this defen all the attributes that can be determines usey an attributs.

$A^* \rightarrow B\ C\ D$

$B^* \rightarrow C\ D$

↳ $R \rightarrow \{A\ B\ C\ D\ E\}$

$A \rightarrow B$

$A \rightarrow E$

$C \rightarrow B$

$C \rightarrow E$

$B \rightarrow D$

Candidate key

$A^+ \rightarrow \{A\ B\ E\ D\}$

$B^+ \rightarrow \{B\ D\}$

$C^+ \rightarrow \{C\ B\ E, D\}$

$D^+ \rightarrow \{D\}$

$E^+ \rightarrow \{E\}$