Lec-7

Relational Model :-

$\Rightarrow$ Tables $\longmapsto$

Customer $\longrightarrow$ Cust-ID
$\longrightarrow$ Name
$\longrightarrow$ address
$\longrightarrow$ Contact

Table $\rightarrow$ Relation $\rightarrow$

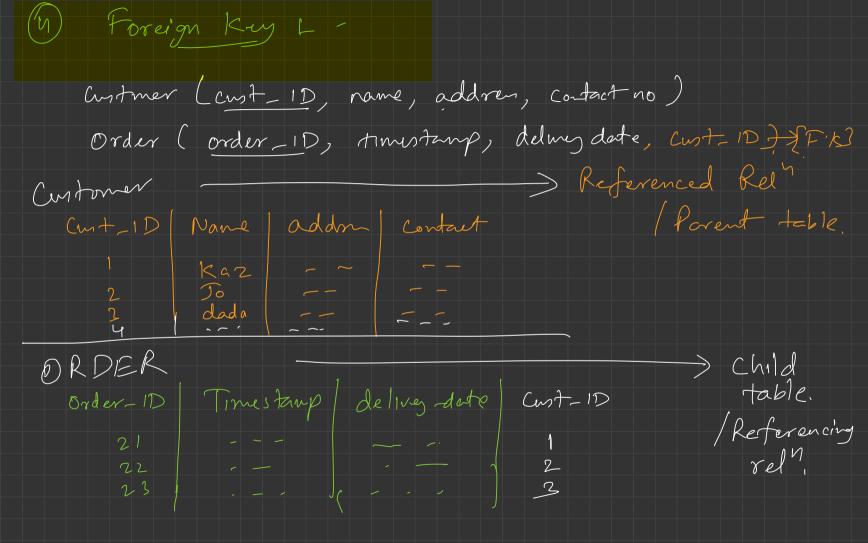|	Cust-ID	Name	addr	Contact
①	1	Lakshay	-- ~	8 8 8 --
②	2	Raj	-- ~	-- --

⇒) Degree of table → No. of attributes.
Cardinality → Total no. of tuple.

⇒) DB design |→

① ER Model → ER diagram.

↕

② Relational Model →

↓

③ RDBMS → Software DBMS.

↓

Software implementation of R.M

MySQL, MS access, Oracle etc.

Online Delivery system

① Customer (<u>Cust-ID</u>, name, address, contact no.)

② ORDER (order-ID, timestamp, delivery-date)

=>   Key =>   Forgein Key

=>   Keys =>   Primary Key   => each data point

uniquely identfy

Cust - ID | Name | Contact | addres | email |

① Super key =>   {Name, contact}, {Name, email}

{Cust-ID} email] , {Cust-ID}   {And}

{Cust-ID, Name, contact, email}

{Raj, 888}
{Raj, 811}

② C.K. $\longrightarrow$ {contact}, {cust_ID}, {cust_ID, contact}
{cust_ID, email}

③ P.K $\Rightarrow$ {cust_ID} $\longrightarrow$ P.K.

④ A.K $\Rightarrow$ {C.K} $-$ P.K. $\Rightarrow$

4) Foreign Key L -

Customer (cust_ID, name, address, contact no)

Order (order_ID, timestamp, delivery date, cust_ID) → { F.K }

Customer ⟶ Referenced Rel^n / Parent table.

Customer

Cust_ID	Name	address	Contact
1	Kaz	-- ~	--
2	Jo	--	--
3	dada	--	---
4			

① ORDER ⟶ Child table. / Referencing rel^n.

Order_ID	Timestamp	delivery date	Cust_ID
21	---	~ -.	1
22	. -	. . -	2
23	. - .	. . .	3

*   Compurite key —

Table of School A

reg.no	name
101	Ram
102	Monika
103	Tata.

Merged. →

table of school B

reg_no	name.
AB101	Tammy
AB102	Nota.

Surr-no	reg_no	name
1	101	Ram
2	102	Monika
3	103	tata
4	AB101	Tommy
5	AB102	Nota.

CRUD

Rollno	Name	Contact.	Age.
1	A	— —	
2	B	——	
3	C	— — .	
N	812	ABC.	

Create

⭐ Referential constraint

Customer :-

Parent reln
Referenced table.

Cust_ID	Name	add	Contact no.
1	Kaz	- - -	— -
2	Jo	- -	- - -
3	Dada	- -	

ORDER :-

↳ *Child*
Reln
Referecing table

Order_ID	Timstamp	delivery date	Cust_ID → F.K.
21	- - -	— —	1
22	- -	— -	2
23	- - -	— —	✗ Null.

① **Insert constrant :-** value can't be inserted in child table if the value is not lying in parent table.

② **Delete constant :-** value can't delete from parent table if the value is lying in child table.
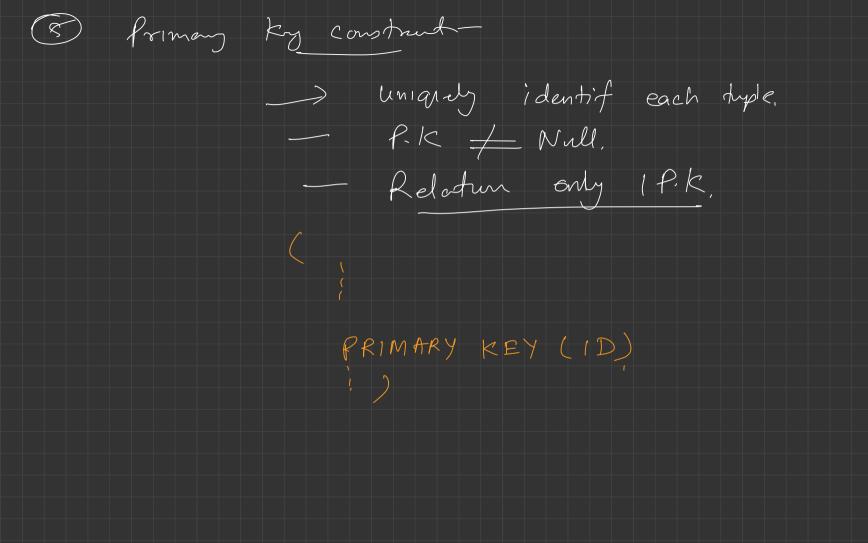
=) **ON Delete Cascade** :—

Can we delete value from parent table if the value is lying
in the child table w/o violating delete constraint?

→ delete value from parent table ⟶ delete corresponding
entry from child table too.

create table Order (order-ID int P.K, ---, ---, cust-ID int
referencing Customer
on delete cascade);

2) Can F.K have Null value?

ON Delete Null :)

delete value from parent table ⟶ put corresponding
F.K value Null.

✳ Key Constraints :-

① **Not Null** → By default a attribute/ column can be Null.

↳ enforce a column to not accept Null.

Create Table Customer (

    ID int Not Null,
    Name varchar(50) Not Null,
    Age int,
);

(2) Unique Constraint →

         ↳ ensure all values in col. are diff.

— Both unique & P.K constraint provide uniqueness.

— you may have many unique constaints per table
         But only one P.K constents per table.

```
create Table Customer (
     ID int Not Null,
     Name varchar(50) Not Null,
     unique (ID)
);
```

③ Default Constant :

→ set default value of col.

```
create table Customer (
            Prime_status   int  DEFAULT 0,
            :
);
```

④ Check :-

— limit value Range (Domain)

```
(
   :
   CHECK (age >= 18)
)
```

(8) Primary Key constraint

$\rightarrow$ uniquely identif each tuple.

— P.K $\neq$ Null.

— Relation only 1 P.K.

(
  :
  :

PRIMARY KEY (ID)

:)

⑥ Foreign Key Constraint

___ Keeps rel^n b/w 2 table.

Create table Order (

PRIMARY KEY (order_ID)

FOREIGN KEY (Cust_ID) Referencing
Customer (Cust_ID)

);