

Data abstraction :

Physical level : data সংগ্রহ/ store হয় জটা
 logical a : কি ধরনের data ব্যবহার হয়

View a : user

Candidate key : মুক্ত ফিল্ড দিয়ে বেকর্ডগুলোকে separately identify করা সম্ভব.

Primary key : এটি candidate key null হওয়া সম্ভব না,

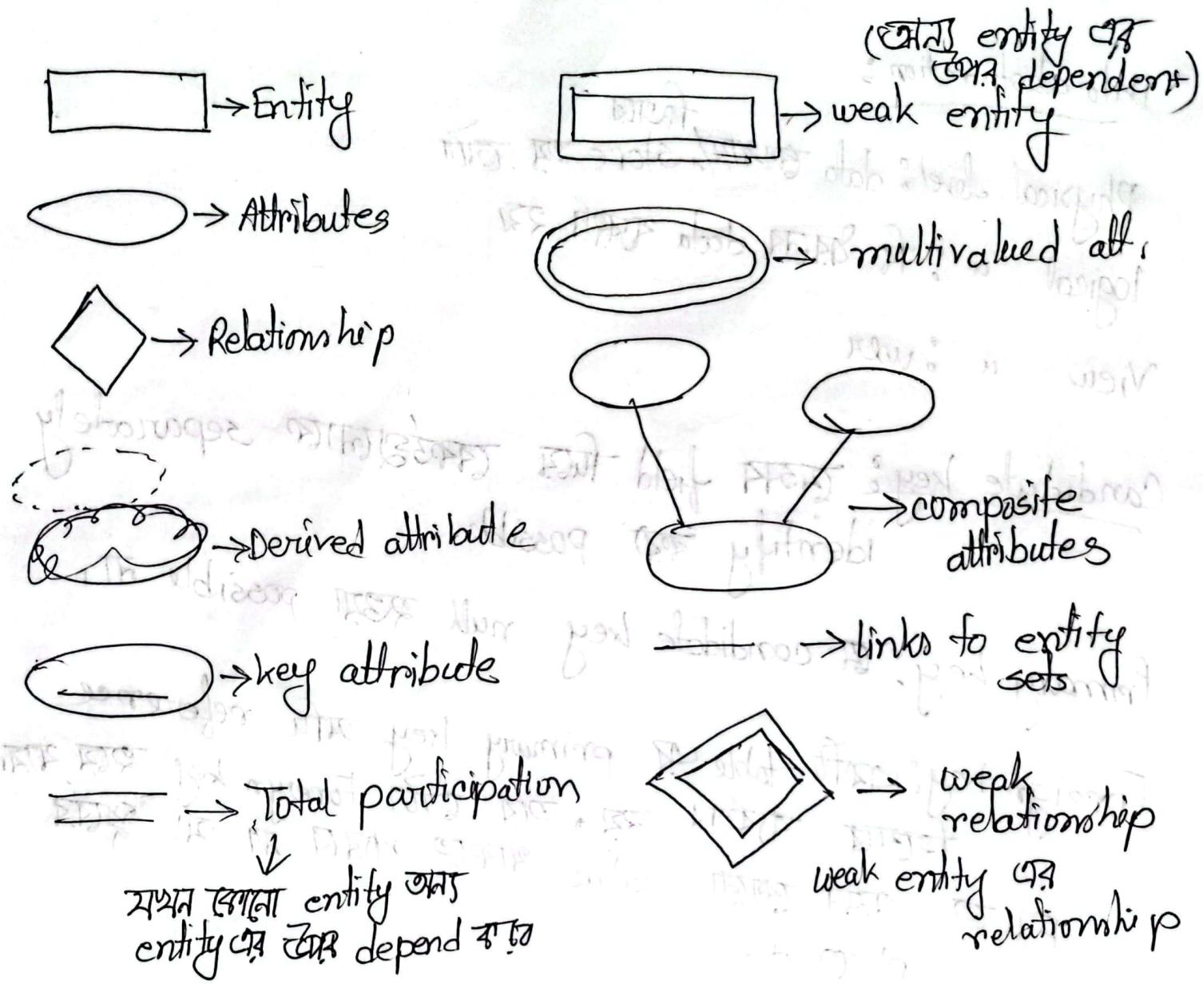
Foreign key : একটি table এর primary key এর reference হিসেবে ব্যবহার হয়, অবে জটা Foreign key হয়ে থাকা FK এর একটি ক্ষণে value থাকতে পারবে না যা primary pk তে নাই

ER → Entity Relationship

Shows the relationship among the entity sets,
 And also helps to visualize the logical structure

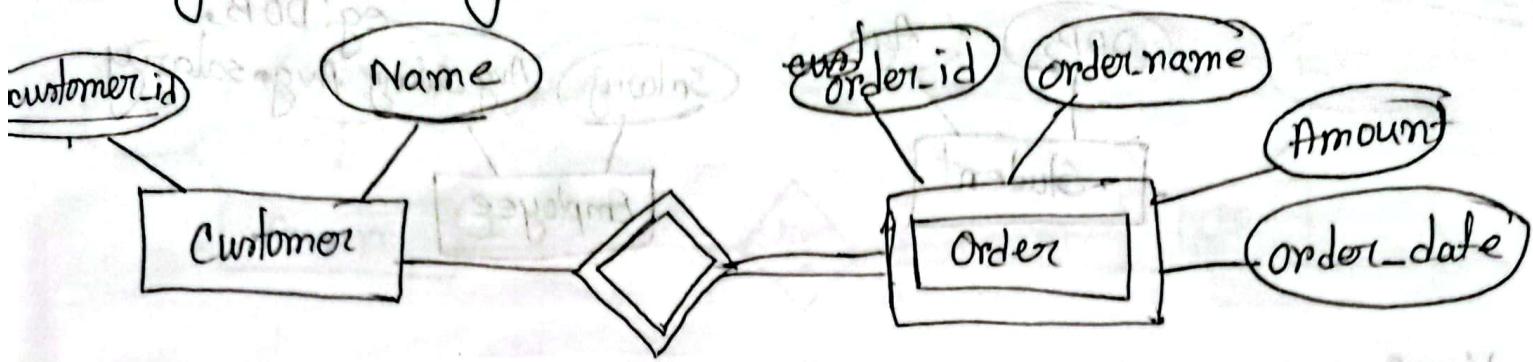
of databases.

Table/entity : rectangle



Strong entity: মুখ্য primary key আছে
 Weak entity: কোর্স নির্ভরশীল, strong entity-এর নির্ভরশীল
 এবং কোর্স নির্ভরশীল

Strong-weak entity relation:



customer_id হলে order-এর foreign key

Types of attributes:

① simple ② Derived ③ Composite ④ Key

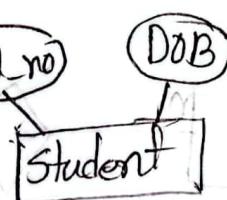
⑤ Single valued ⑥ Stored ⑦ Multivalued

⑧ simple: যাকে অঙ্গীকৃত কোন না

⑨ Composite: যাকে subdivided
করা যাবে।

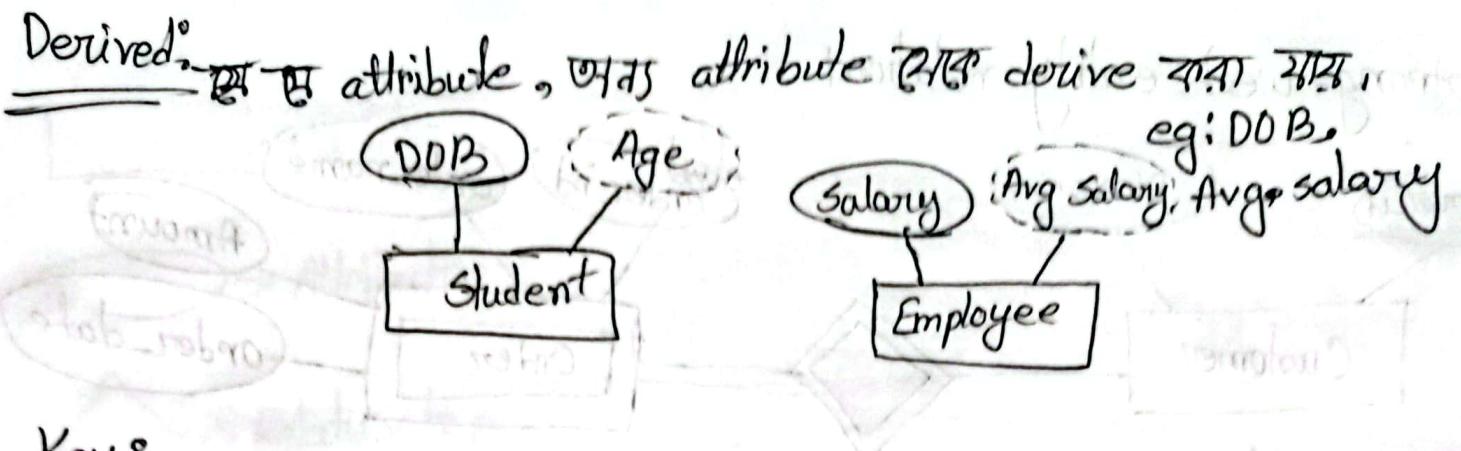
⑩ single valued: মাত্র একটি কাউণ্ট value ২/১১১১১১

eg: Age, Roll, DOB



⑪ Multivalued: যার একটির একটি value থিভে আছে, eg: phone, email





Key: primary key.

Stored: सारी value future में बदलने के change करते हैं, eg: DOB, Birth place.

Types of Relationship:

① One to one:



② One to many:



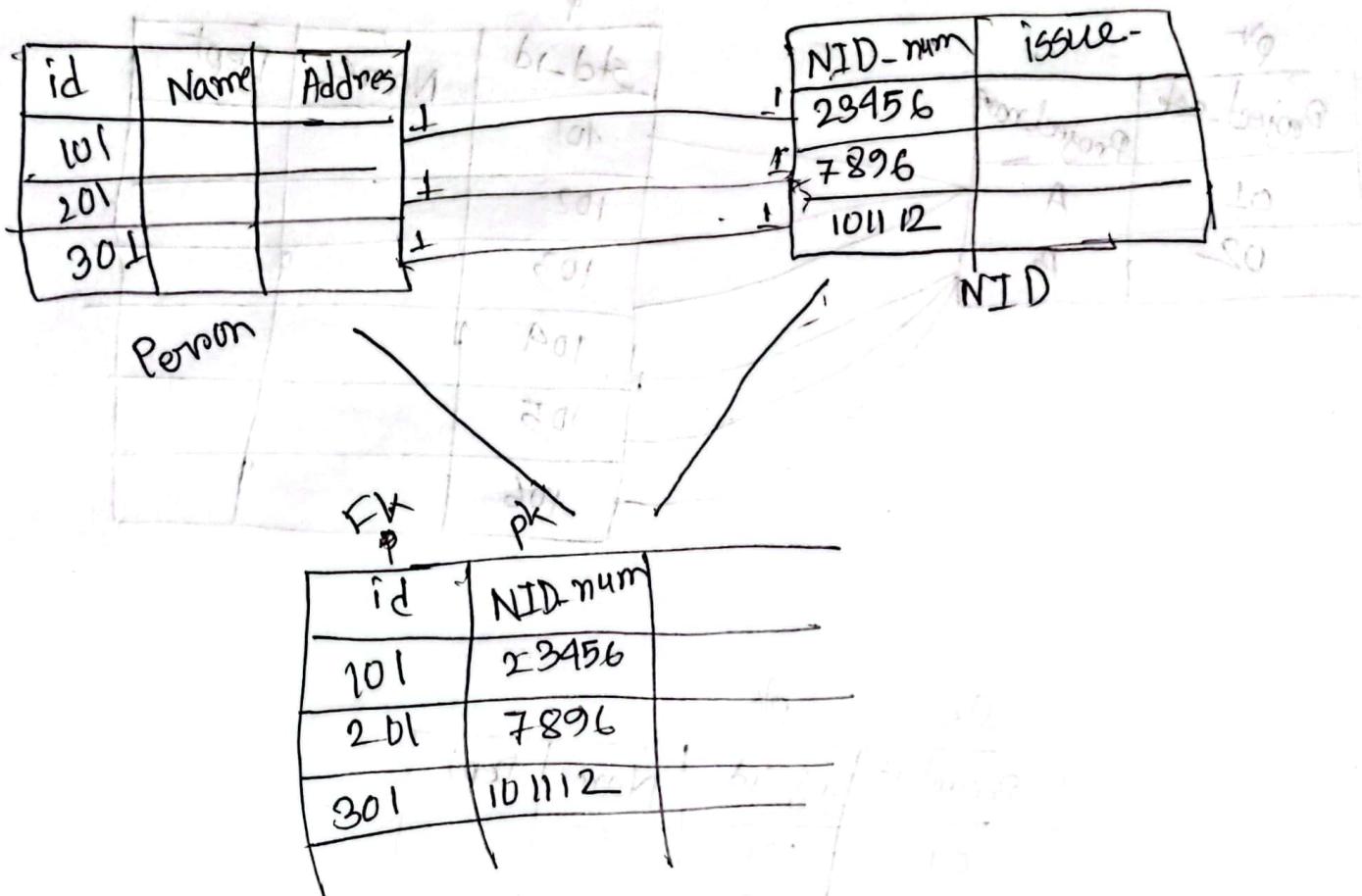
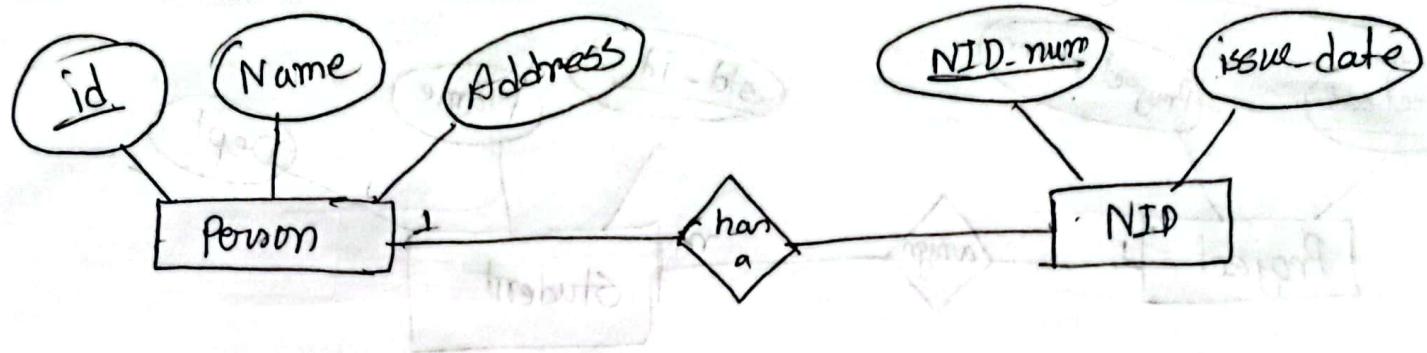
③ many to one:



④ many to many:



One to one:



One to many:

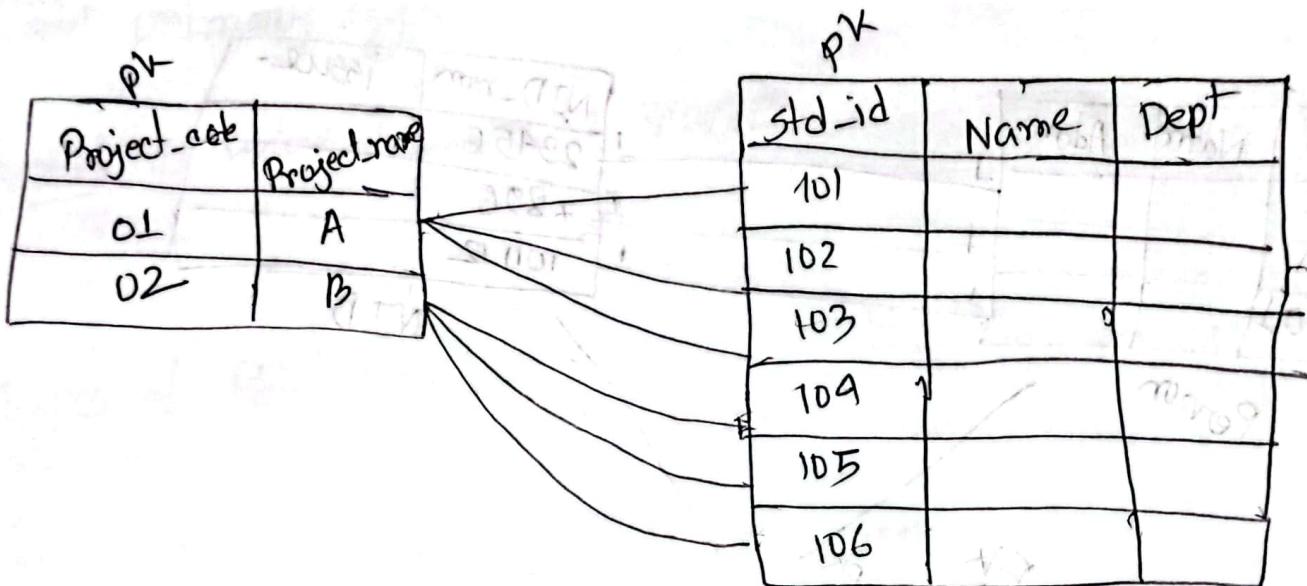
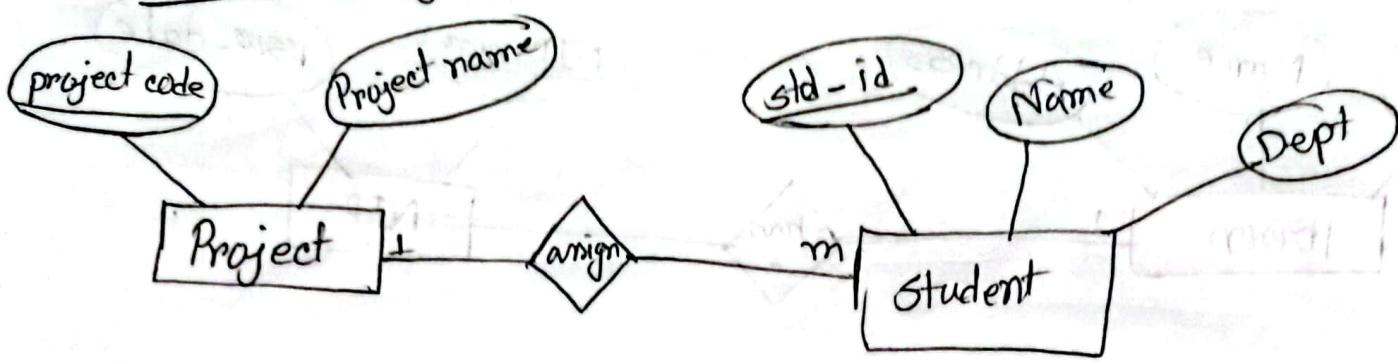


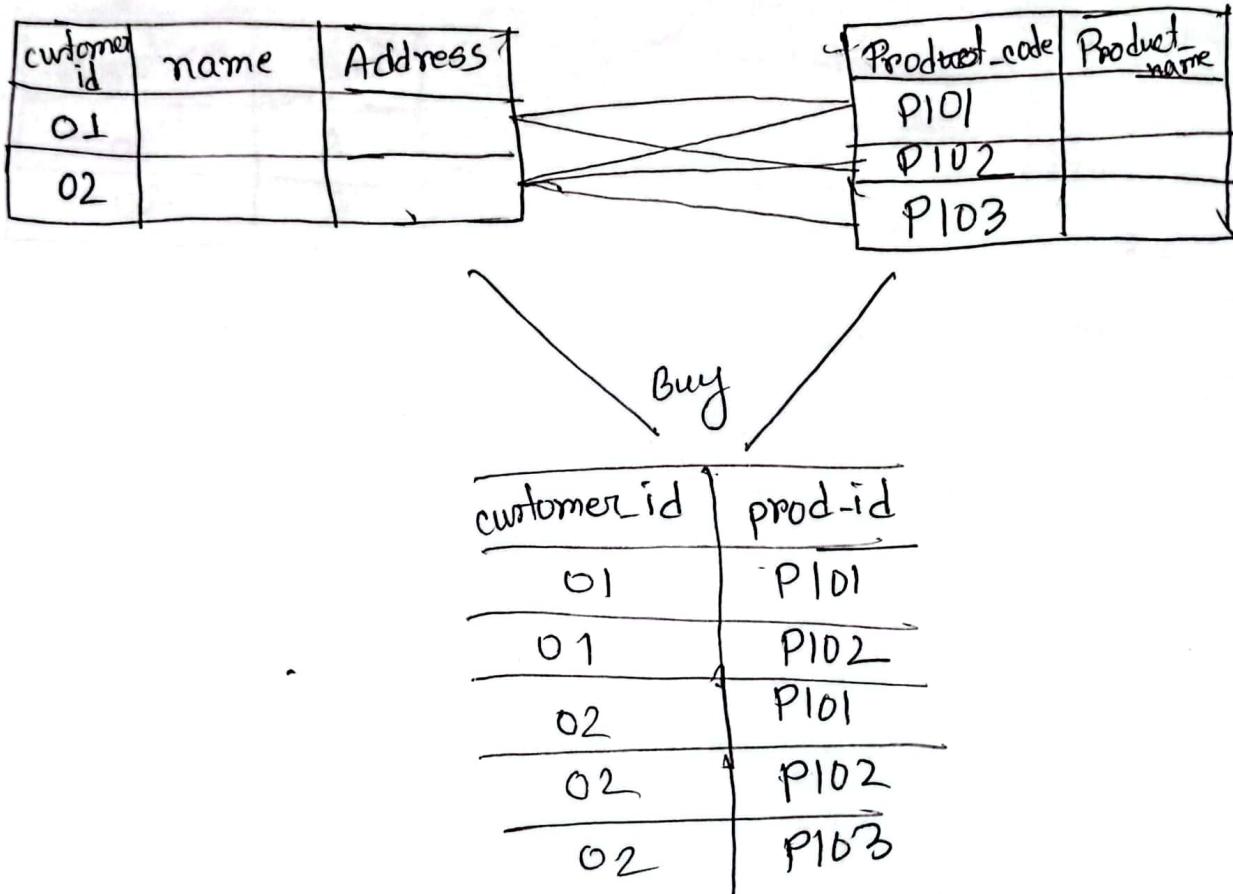
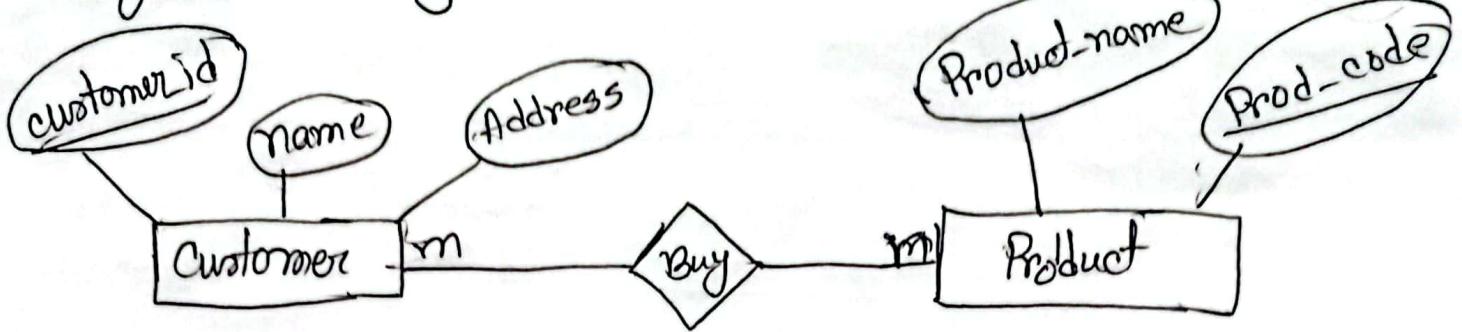
Diagram showing the normalized form of the relationship table.

Relationship Table: Primary Key (PK) is Project_id and std-id. Secondary Key (SK) is Project_id.

Data:

Project_id	std-id	Name	Dept
01	101		
01	102		
01	103		
02	104		
02	105		
02	106		

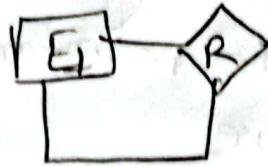
Many to many:



Degree of relationship:

एकी relation वा कठशळी entity असते

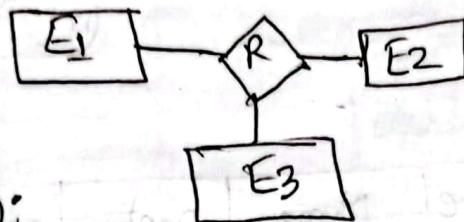
(Unary (1 degree)):



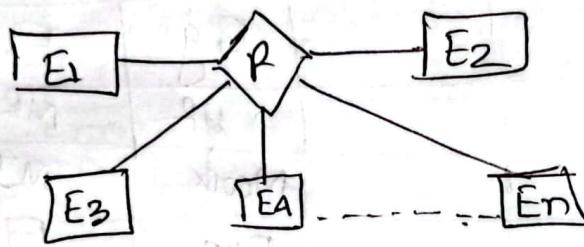
Binary (2 degree):



Ternary (3 degree):



N ary (n degree):



Tuple: row असूजी

Cardinality:

Relational schema:

Table = name(Attribute₁, attr₂, ..., attr_n)

Normalization:

- It is the process of organizing the data in database.
- It is used to remove the redundancy

① Row wise

② Column wise (remove ~~जुड़ी जूड़ी~~, reduce बाकी जूड़ी)

① Insert Anomaly:

② Update anomaly

③ Delete anomaly

Emp-id	Emp-name	Phone	Dept	Dept-manager
101	A		Eng	FH
102	B		HR	MR
103	C		Mark	MJ
104	D		Eng	Flt
105	E		Mark	MJ
101	A		Eng	FH

row
wise
delete

PK ↓ FK ↓

empid	name	phone	dept-id
101			1
102			2
103			3
104			1
105			3

dept-id	dept	dept. manag
1	Eng	
2	HR	
3	Mark	

1NF (First Normal Form)

- ① In a column, multi values are not allowed
- ② At. domains same
- ③ every col unique name
- ④ Doesn't matter the order.

s-id	Name	course
101	A	OS, DBMS
102	B	OS
103	C	DBMS
104	D	Algo DBMS

composite primary key

s-id	Name	course
101	A	OS
101	A	DBMS
102	B	OS
103	C	DBMS
104	D	Algo
104	D	DBMS

s-id	Name
101	A
102	B
103	C
104	D

s-id	Name	Course1	Course2
101	A	OS	DBMS
102	B	OS	
103	C	DBMS	
104	D	Algo	DBMS

s-id	Course
101	OS
101	DBMS
102	OS
103	DBMS
104	Algo
104	DBMS

2nd Normal Form (2NF)

- ① Table should be in INF
- ② Partial Dependency not allowed
Need to remove partial dependency
- ③ Partial Dependency not a composite primary key

composite primary key

partial dependency

S-id	Name	Address	course-id	course-name	Credit	Grade
101	A	DH	1	C	3	AT
101	A	DH	2	Java	3	A
102	B	GN	1	C	3	B
102	B	GN	2	Java	3	A

partial dependency: composite pk-এর জন্যে যে column দ্বারা
যাতায়ারি যা pk-এর অন্যর ডিপেন্ডেন্স না, আবাব ক্ষণাত্মক
pk-এর relation

এটা p_k এর টিপ্পিতে এটা table

c-id	c-name	Address	credit
1	C	DOPP B	
2	Java	GOD B	

S-i d	Name	Address
101	A	DH
102	B	GN

PK

→ fully dependent

s-id	c-id	Grade
101	1	A+
101	2	A
102	1	B
102	2	A

steps:

- ① composite pk বের করো
- ② ~~প্রতি~~ ~~প্রতি~~ pk এর স্তর
fully dependent attribute
বিষয়ে টাইল আলাদা রাখো
- ③ এই column দুই pk এর
স্পষ্ট অন্তর্ভুক্ত dependent
আলাদা table

Third Normal Form (3NF)

① Table must be in 2NF

② Transitive dependency not allowed

Course-name	Teacher-id	Teacher-name	credit
C	1	A	3
OS	2	B	3
DBMS	1	A	3
C-LAB	2	B	2

Transitive dependency: monkey কলা মেরে ক্লাসের ক্লাস একে অন্যের পুরো নির্ভর নির্ভর নির্ভর

Pk
↓

D1

C-name	T-id	credit
C	1	3
OS	2	3
DBMS	1	3
C-lab	2	2

T-id	T-Name
1	A
2	B

From 2NF example partial dep.

C-id	C-name	credit
1	C	3
2	Java	2

⇒ 3NF

C-id	C-name
1	C
2	Java

C-id	credit
1	3
2	2

SQL statement

↓
DDL

1. Create
2. Alter
3. Drop
4. Truncate
5. Rename

↓
DML

1. Insert
2. Update
3. Delete

↓
DCL

1. Grant
2. Revoke

TCL

1. Commit
2. Roll back
3. Save point

DQL

1. Select

DDL → Data Definition Language

DML → Data Manipulation Language

DCL → Data Control Language

TCL → Transaction Control Language

DQL → Data Query Language

Database create:

CREATE DATA BASE Database-name;

TABLE create:

CREATE TABLE table-name

col1 datatype,

col2 datatype;

datatype int, varchar,
datetime,

),

Index:

CREATE INDEX index-name

ON table-name (col1, col2, ...)

Where condition

the column ব্যবহার

য়ে

Alter %

Rename Table: ALTER TABLE old-table-name
TO new-table-name;

Add column: ALTER TABLE table-name
ADD column-name datatype;

Modify column datatype: ALTER TABLE table-name
MODIFY COLUMN col-name datatype;

Rename col name: ALTER TABLE table-name
RENAME COLUMN old-col TO new-col;

Drop col: ALTER TABLE table-name
DROP COLUMN col-name;

Drop Table: DROP TABLE table-name;

Truncate: ~~set~~ column ~~to~~ ~~from~~ data remove.
TRUNCATE TABLE table-name;

Insert %

Insert into table-name
VALUES (val1, val2, ...),
(val1, val2, ...);

update: (to modify the existing records in a table)

UPDATE table-name
SET col1=val1, col2=val2
WHERE filter condition;

Update student for id=3
where dept is "NULL" would
be replaced by dept = "CSE"

UPDATE Student
SET dept = 'CSE'
WHERE id = Null;

UPDATE table-name
SET col1=val1, col2=val2;

अपडॉट टेबल का डाटा
modification

Update all the students अपडॉट
dept as CSE

UPDATE Student
SET dept = 'CSE';

DELETE:

→ To delete existing records in a table

DELETE FROM table-name
WHERE filter condition;

specific record delete
दोषी

DELETE FROM table-name;

दोषी records delete 262
2160.

SELECT: to select data from database

SELECT col1, col2
From table-name;

only col1,
col2 will be
shown

SELECT * → All col will
be shown
FROM table-name;

SELECT DISTINCT col1, col2,
FROM table-name;

col 1 का unique
value select
show

Operators

① Arithmetic: +, -, *, /, % *not equal*

② Comparison: =, <, >, <=, >=, <>, !

③ Logical: ANY, ALL, AND, OR, NOT, BETWEEN, EXISTS, IN
LIKE, IS NULL

WHERE: Record fetch করার condition ব্যবহাৰ
আপনা,

SELECT col1, col2, ...

FROM table-name

WHERE filter-condition;

AND: AND দিয়ে দ্বয়া condition এর অন্তর্মালা data fetch
করতে প্রতিটি condition অৱ্যাপ্ত নাগাদ

SELECT col1, col2, ...

FROM table-name

WHERE cond1 AND cond2 ...;

OR: এখন এটি condition true হলেই হয়

SELECT col1, col2, ...

FROM table-name

WHERE cond1 OR cond2 ...;

NOT : condition নির্বাচন হলে data fetch করবে।

```
SELECT col1, col2...
FROM table-name
WHERE NOT cond;
```

LIKE & NOT LIKE:

LIKE: where এ ব্যবহার হয়, জো স্পেসিফিক প্যাটেরি এবং
ডিটিতে data fetch করতে।

- % represents zero, one or multiple character
- - (underscore) represents one single character

```
SELECT col1, col2...
FROM table-name
WHERE col-name LIKE pattern;
```

pattern rules:

- ① ' ' এ ছেতের
- ② '%' → আনে এবং পর আরো অন্যক্ষণ character থাকতে পারে
- ③ '-' আনে একটি character

WHERE name '%F%' → যাদের নামের প্রথম letter F

WHERE name '-a%' → u u এবং u a

WHERE phone '017%10' → u sim gp এবং

Aggregate functions

- For performing calculation on multiple values and return a single value.
- ignores NULL values except COUNT function
- GROUP By & Having clause এর সাথে SELECT এর জন্য

5 Aggregate functions:

① MIN ② MAX ③ AVG ④ SUM

id	name	salary
1	Faizan	10K
2	Hossan	40K
3	Abir	20K
4	Hasan	30K
5	Nurul	NULL

⑤ COUNT

কয়েক্ট রেও
অটোমেটিভ কাউন্ট

*SELECT MIN(salary)
FROM employee; $\Rightarrow 10K$

*SELECT MAX(salary)
FROM employee;

*SELECT AVG(salary)
FROM employee; $\Rightarrow 25000$

*SELECT SUM(salary)
FROM employee; $\Rightarrow 100000$

*Row count:

SELECT count(*)
FROM employee;
 \downarrow
5

*কয়েক্ট not null value এর কলাগু:

SELECT count(salary)
FROM employee;
 \downarrow
4

Subquery / Inner query / Nested query

Subquery (=) করা করে সবুজ inner query একান্ত
সাথ output দেয়

Outer query এর where ও ম্যে column থাকবে, inner query
select ও একান্ত column এতে থারে,

employee1	
id	name
1	A
2	B
3	C
4	D
5	E

employee2	
id	salary
1	20K
2	30K
3	50K
4	20K
5	15K

* Find the name of the employee whose salary is 20K

SELECT name
FROM employee

WHERE ID (=) (select id

FROM employee2

WHERE salary = 20.00.0)

* 5th employee এর চেয়ে জ্যেষ্ঠ salary থাপ্পের তাপ্তির
নাম

SELECT name

FROM employee

WHERE salary > (SELECT salary

FROM employee2

WHERE id = 5)

IN & NOT IN

to specify multiple values in a where clause

SELECT col-name
FROM table-name

WHERE col-name IN (val1 val2 ...);

WHERE col-name IN (inner-query);

* ~~SELECT salary 15K, 20K~~ ~~SELECT name, id~~

SELECT id, name

FROM employee1

WHERE id IN (SELECT id,

NOT ~~SELECT 15K, 20K~~ ~~salary FROM employee2~~ WHERE salary ~~NOT~~ IN (20K, 15K)),

* NOT IN is opposite of IN

id	name
1	A
4	P
5	E

Exists & not exists

→ used to test for the existence of any record in a subquery

→ EXISTS returns true → if subquery returns record

Syntax:

SELECT col-name
FROM table-name
WHERE Exists (SELECT col-name
FROM table-name
WHERE cond)

* यदि salary 15K, 20K तादेश name, id (using exists)

SELECT id, name
FROM employee

WHERE NOT EXISTS (

↑
NOT परिलक्षित SELECT ID

salary 15K, 20K तादेश FROM employee2

हैरी रुप

WHERE salary IN (20K, 15)

⇒ and employee1.id = employee2.id);

BETWEEN and NOT BETWEEN

→ Select values within a given range

→ values can be number, texts or dates

SELECT col.name

FROM table-name

WHERE colname BETWEEN val1 AND val2;

id	name	salary	J-date
1	A	20K	2022-01-1
2	B	30K	2022-01-2
3	C	50K	2022-01-10
4	D	20K	2022-02-10
5	E	15K	2022-01-15

* 1-3 id এর employees:

SELECT *

FROM employee

WHERE id BETWEEN 1 AND 3;

id	name	salary	j-date
1	A	20K	2022-01-1
2	B	30K	2022-01-2
3	C	50K	2022-01-10

* J-date মাঝে January এর 10-25:

SELECT *

FROM employee

WHERE j-date BETWEEN '2022-01-10' AND '2022-01-25'

id	name	salary	j-date
3	C	50K	22-01-10
5	E	15K	22-01-15

* NOT BETWEEN ফিলে কোনো range বাইডে যাবিদের
data show করবে।

GROUP BY

TIMES

→ group rows that have the same values.

→ often used with aggregate functions

SELECT colname
FROM table-name
WHERE condition
GROUP BY colname
ORDER BY col-name)

dept	name	salary
HR	A	20K
HR	B	10K
Eng	C	50K
Business	D	30K
Eng	E	50K
Business	F	20K

SELECT dept, COUNT(*)

FROM employee

GROUP BY dept;

Order BY COUNT(*) DESC;

dept	count
HR	2
Eng	2
Bus	2

Count output
Sum(salary) ফিলে প্রতি dept

ORDER BY

→ to sort the result set in ascending or descending

SELECT col1, col2

FROM table-name

ORDER BY col1, col2 ASC / DESC;

* salary এর descending sort,

SELECT *

FROM employee

ORDER BY salary DESC;

LIMIT

→ to specify the number of records to return

SELECT col-name

FROM table-name

WHERE condition

LIMIT number; → মাত্র নোট রেখে কোর্টে

অথবা,

LIMIT number OFFSET number; → এই যেকোনো
অ কস্টম
row বান আবে।

id	name	salary
1	A	40K
2	B	20K
3	C	5K
4	D	15K
5	E	30K
6	F	50K
7	G	70K

যাদের salary 30K এর টেম্প এবের মধ্যে সেক্ষেত্রে ২ অনেকে হলো

SELECT

SELECT *

FROM employee

WHERE salary > 30K

ORDER BY salary DESC

Limit 2;

id	name	salary
7	G	70K
6	F	50K

HAVING

→ WHERE এ aggregate function ব্যবহার করা যায় না, অর্থাৎ
having লাগে

SELECT col_name
FROM table-name
WHERE filter_condition
GROUP BY col_name
HAVING condition
ORDER BY col_name;

যদি department এ অন্তর্ভুক্ত ২ জনের মধ্যে
10K এর ক্ষেত্রে salary পাস, তাদের মধ্যে

emp_id	department	salary
101	Eng	50K
201	HR	10K
301	Business	30K
102	Eng	20K
103	Eng	60K
104	Eng	20K
302	Business	15K
303	Business	5K
202	HR	20K

SELECT department, COUNT(*)

FROM employee

WHERE salary > 10K → যাদের salary 10K এর টেম্প তাদের নিম্নাঞ্চ

GROUP BY department → 10K এর টেম্প যাবা পাস, তাদের department wise
list করনামা,

HAVING COUNT(*) > 2 → department list এ মেখানে emp
কর্তৃপক্ষ করে এর মধ্যে তাদের নিম্নাঞ্চ
ORDER BY COUNT(*) ; → emp কর্তৃপক্ষ অনুযায়ী dept করে ascending