Borshon Sir

Sub:

'C' bay
Cycle-01

Day

24/4/19

Time: Date: / /

Database Systems

Database:

→ Collection of Data

→ Collection of interrelated data

Database Management System

(Database কে Manage করবার জন্য system)

DBMS → to store & retrieve data (in an efficient & convenient way)

Primary Goal

Books:

Database System Concepts (6th edition)

→ Silberschatz

→ Korth

→ Sudarshana

Sub :

Day

Time : Date : / /

Why DB?

* Large Amount of info.
  - → to store in well structured way
* Data Safety (Safety of Informatabn)
  - → From System Crashes
  - → Unauthorized access (Password, A/c info.)

→ (DB আসার আগে এই system এ কাজ করত Data store হতো)

Traditional file-processing System:

- to store data → .dat/.txt files
- to retrieve data → a set of programs.

Sub: Time: Date: / /

Disadvantages:

Data Redundancy & Inconsistency

Reason → files & programs created by several programmers long period

Various copies of same data → in various files

Problems:

→ Higher storage & access cost

→ Data inconsistency

Data Accessing Difficulties:

→ From all data manually (x) খুবই ঝামেলা

→ Need a program to generate the list (x) ঝামেলা

Example: 'Kazla case' (খুনিদের DB থেকে Criminal sort out করা)

'D' Day
Cycle-01

Database:

Database আসার আগের দিন -

File-Processing System:

Disadvantages:

→ Data Redundancy & Inconsistency
→ Difficulty in accessing data
→ Data Isolation

Data → scattered >> in various files in diff. formats (no uniform format)
- → no relationship between any two files
- → Difficult to write programs

Sub :

Day

Time : Date : / /

→ Integrity Problem :

(অখণ্ডতা)

Valid info → in → files / DB

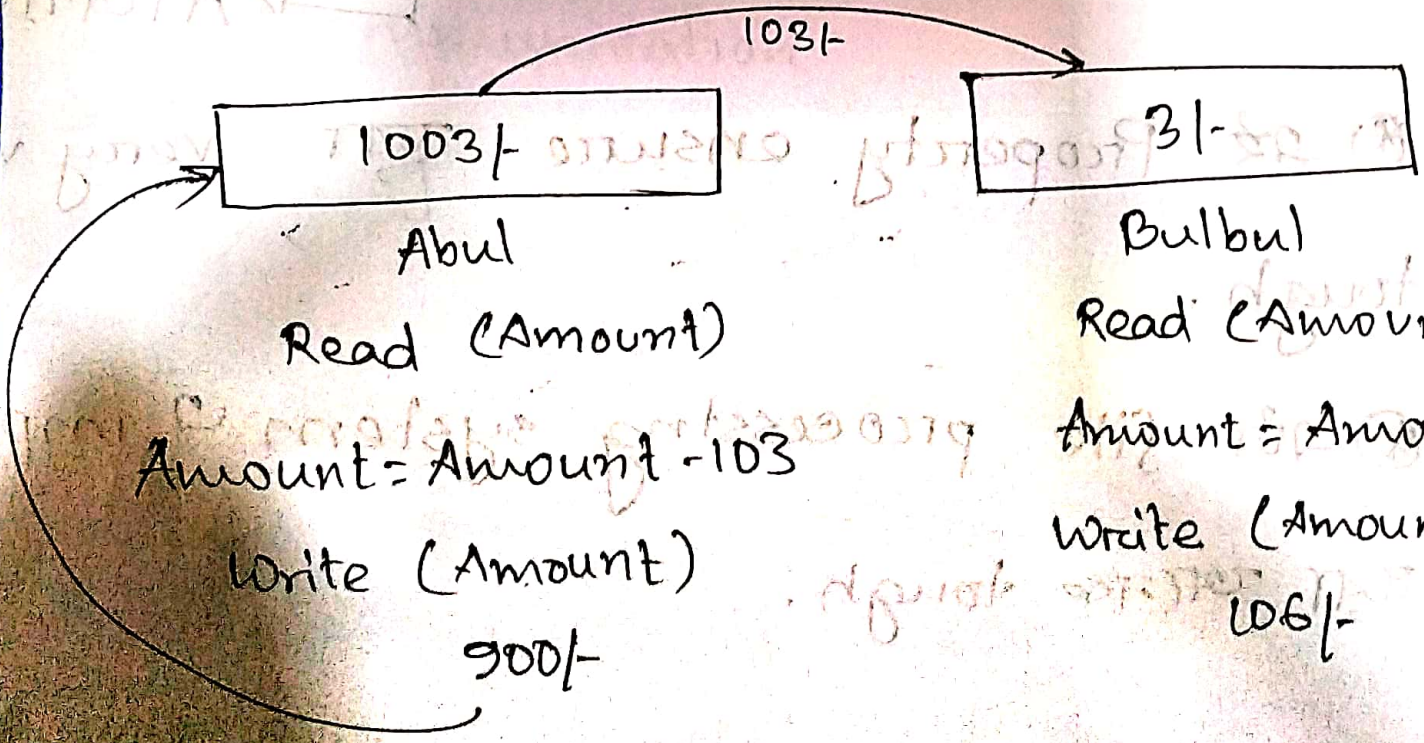
→ Consistency Constraints maintain করতে,

Problem :

Constraints involve data items from multiple files.

→ Atomicity Problem

Fund transfer >> must be atomic



Sub :

Time : Date : / /

⊛ Power off / System Crash হয়ে গেলে সমস্ত -যখানে আবার আগের মত 1003/- হয়ে যাবে, That means, no operation will have impact on DB. This property is called atomicity

Atomicity বলে - 'হইলে সব হবে, না হলে কিছুই হবে না'

Entirely or Not at all

→ ATOMICITY

⊛ এই Property ensure করা very very tough.

⊛ এবং file processing system এ implement করা অনেক tough.

Sub:

Day

Time:

Date: / /

→ Security Problem:

→ not every user need to access all data

security constraints >> Difficult

DBMS:

Applications:

→ Universities

→ Railways / Airlines

→ Telecommunication

→ Banking

Sub: Cycle-01 Time: Date: 29/4/19

Instance of Database: কোনো মুহূর্তে যে Info. এর collection Database এ থাকে তা হল Instance of Database.

→ the stored info. in DB at a particular moment (Frequently change হয়)

Database Schema:

→ Overall design of DB.
(Frequently change হয় না)

Data Model:

→ # Relational Model:

→ DB >> collection of tables

has → unique name

each table

→ a set of columns

columns >> attribute

⇒ Table গুলোকে বলে Relation. Table >> Relation

Entity-Relationship Model: (E-R Model)

Object-Oriented Model

Semi-structured Model

Database Languages:

→ Data Manipulation Language (DML)

→ Data Definition Language (DDL)

DML:

↳ contains the commands used to manipulate data.

Example:

⇒ Insertion of data
⇒ Deletion of data
⇒ Retrieval of data
⇒ Modification of data

} এগুলো করতে যে Command গুলো ব্যবহৃত হয় তারা সবাই DML এর অন্তর্ভুক্ত

Sub:

DDL:

used to define database schema

Example / Purpose:

→ to create a table
→ to drop table
→ to create column
→ to drop column

✱ এছাড়াও DCL, TCL আছে, পরে পড়ানো হবে.

DBA: (Database Administrator)

has the → central control of DB

Functions:

→ DB schema কে define করা / overall design কে define করা

→ Data Access Method definition
(কোন Method এ access করবে তা define করা)

→ Authorization of Data Access → which part of DB can be accessed by which user

→ Routine Maintanence of DB

- → Periodically backing up data (info)
  - → Remote Server এ
- → Free Space in the disk (checking)

BS Sir

Sub:

'a' Day cycle-02

Day

Time: 11:40 Date: 16/6/19

CHAPTER-2

Most used DB model is-

Relational Database Model:

DB is nothing but collection of many tables.

DB → collection of Tables = (Relation)

* (यहाँ Tables = Relation so, it's called Relational DB model.

⇒ Why table = Relation? ——— (1)

Let us take an example-

Movie-Info (Database)

Attribute

Movie (Table)

ID	YEARS	Awards	Name	Genre	Imdb Rating
1.	1980	7	Heerak Rajar Deshe	Drama	8.8
2.	2009	5	3 IDIOTS	Drama	8.4
3.	2019	Null	Password	Action	Null

Tuples are here in this table

Table	→ Relation
Row	→ Tuple
Column	→ Attribute

Ans. of ①:

Table এ একটা tuple এর যেসব Attribute এর Different values তাদের মধ্যে সম্পর্ক বোঝায়. Relationship among a set of values 'অনুসৃ'. এই কারণেই Table কে বলা হয় Relation.

Sub:

Attribute value → কোনো value যদি আমাদের জানা না থাকে, তাহলে null/unknown/doesn't exist লিখব।

→ null

→ (unknown/ doesn't exist)

Attribute

└→ a set of permitted value

Domain (Ex: IMDB এর domain হচ্ছে 0-10.0) অর্থাৎ IMDB এর জন্য Permitted.

Domain is Atomic sometimes, when the elements of the domain are considered as indivisible unit.

Let us take an example-

Mobile No.
+880 1670420420

+880 → Country Code

1670420420 → Operator

যদি চাই তাহলেই c.c এবং Operator ভাগ। আর সেক্ষেত্রে সেগুলো Domain ভাঙা

Sub:

পারব না, আর না ভেঙতে পারলে/ ভাঙার Permission থাকলে তাকে Atomic বলতে পারি.

Such as- এখানে Mobile No. টাকে আমরা Member of non-atomic domain বলতেছি.

* To identify each type uniquely.

* Movie এর দুই জায়গায় record থাকলে Redundancy হবে,

Uniquely identify করতে চলে আসবে 'KEY'.

KEY: → Superkey:

- a set of 1/more attributes used to identify tuples (uniquely)

	Superkey	
{ID, Name}	✓	ID, Name দিয়ে প্রত্যেকটা Movie কে uniquely identify করতে পারলে যেটা {ID, Name} হচ্ছে Superkey. আবার {ID} ও Superkey. কিন্তু Name টা superkey না.
{ID}	✓	
{Name}	✗	
{Name, Year}	✓	
{Name, Year, Awards}	✓	

So, যেই AM. এর set দ্বারা uniquely identify করা সম্ভব সেটা হচ্ছে superkey.

But, Superkey তে সমস্যা হচ্ছে একটা single attribute দিয়ে অনেকসময় uniquely identify করা যায়, তখন extra attribute দিয়ে সবসময় সমস্যা.

Sub: Day

Time: Date: / /

Key:

Set of Attributes	Superkey	Candidate Key
{ID, Name}	✓	✗
{ID}	✓	✓
{Name}	✗	✗
{Name, Year}	✓	✓
{Year}	✗	✗
{Name, Year, IMDB}	✓	✗

* Superkey Concept এর আলোচনা হচ্ছে -

It may contain some extra attri

যেমন: {ID, Name} এ Name হল extraneo

attribute.

Candidate Key >> Minimum Superkey's key which uniquely identify করতে পারে.

Such as- {ID}, {Name, Year}

→ একটি table এ একাধিক candidate key possible (multiple candidate key for a single relation)
এটা একটা Problem..

Primary Key: যেকোনো একটা candidate key database designer choose করে নিবে uniquely identification এর জন্য.

→ Chosen by DB designer.

Ex: In the table, there are 3 Superkeys 2 Candidate keys.

Sub: Day Time: Date: / /

✓ Primary key তে যাদেরই রাখা হয় এমন সব Attribute কে যার value গুলো সচরাচর change হয় না, so, Name can be change Thus, {Name, Year} use করতে পারব না. {ID} টা খুব Rarely change হয়, so {ID} হচ্ছে Primary Key.

***: একটা Table এ অবশ্যই একটাই Primary key থাকে. সেখানে একাধিক Attribute থাকতে ও পারে নাও পারে.

* Foreign Key → এক Table এর Primary Key অন্য Table এ রাখতে চাইলে সেটা Foreign key.

→ A relation may contain the primary key of another relation

Relational schema >> Overall design of relation

↓

Director (Director ID, Name, No. of films, Awards)

* Primary Key এর নিচে একটা দাগ দিতে হয় always.

Sub: Day Time: Date: / /

এখন Director ID নামের একটা Column, Movie Table এ add করব.

ID	Name	Year	IMDB	Genre	Awards	Director ID (Foreign Key)
1.	Heerak Rajar Deshe					1
2.	3 Idiots					3
3.	Password					null

Foreign key টা একটা Table এর সাথে অন্য Table এর Dependency তৈরি করে, এখানে, Movie Table এর সাথে Director Table এর,

◊ Relational Instance:

কোনো একটা Relation Particular Moment এ কি অবস্থায় আছে,

Sub :

Time : Date : / /

Chapter-3 (How to prepare DB)

Query Language:

SQL → most popular

* আগে ছিল Sequel
* 1986 তে ANSI & ISO মিলে SQL-86 বানায়

→ Structured Query Language

QL → to be learned in next class.

. . . . .

Sub:

Cycle: 02

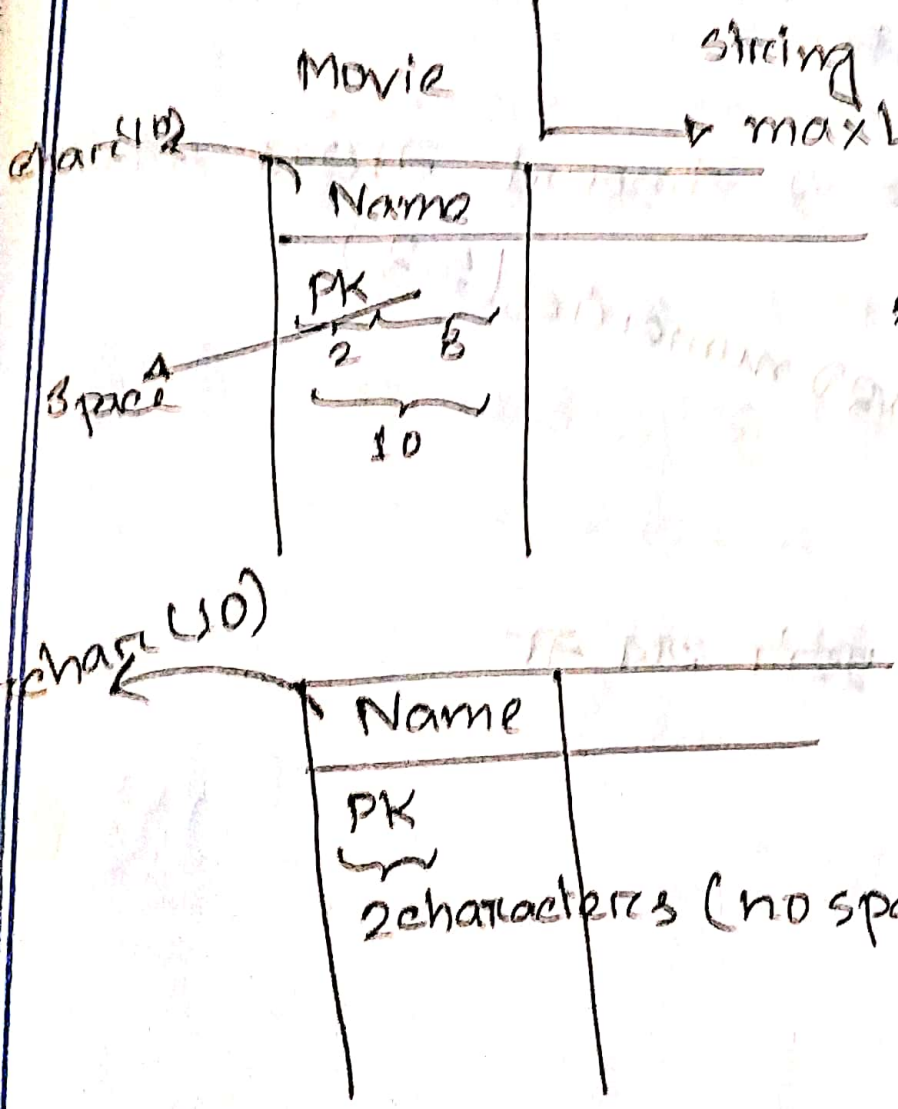
Time: 9:40 Date: 18/6/19

SQL:

Basic Types:

(1) char(n) → Fixed-length character string (length)

(2) varchar(n) → variable-length character string (max length)



* char(n) ↳ exactly n টা character থাকবেই।

* varchar(n) ↳ n হল highest limit of character

* সচরাচর varchar(n) used হয়।

(3) int ⟶ integer number

(4) numeric (p, d) ⟶ Floating point number.

- p ⟶ precision (total no. of significant digits)
- d ⟶ digits' number after decimal point

ধরি কোনো Table এ Amount নামের একটা Field আছে

Amount ⟶ numeric (5, 2)

Amount
375·73 ✓
72·357 } এই দুইটা হবে না
43256·952 }

Parts of SQL:

- DDL (Data Definition Language)
  - ↳ provides commands for → ⊕ defining rel-ional schema ⊕ Adding columns ⊕ Dropping columns/relation
- DML (Data Manipulation Language)
  - ↳ provides commands for →
    - ⊕ inserting new tuple (row)
    - ⊕ Deleting tuples
    - ⊕ Modifying tuples

Table Creation:

```
create table TableName
    (Attribute1 Type,
     Attribute2 Type,
     AttributeN Type)
```

* Director' Table বানাই:

```
create table Director
    (DirectorID int,
     Name varchar(100),
     NumberofFilms int,
     Awards int),
     primary Key (Director ID));
```

Sub:

Day

Time: Date: / /

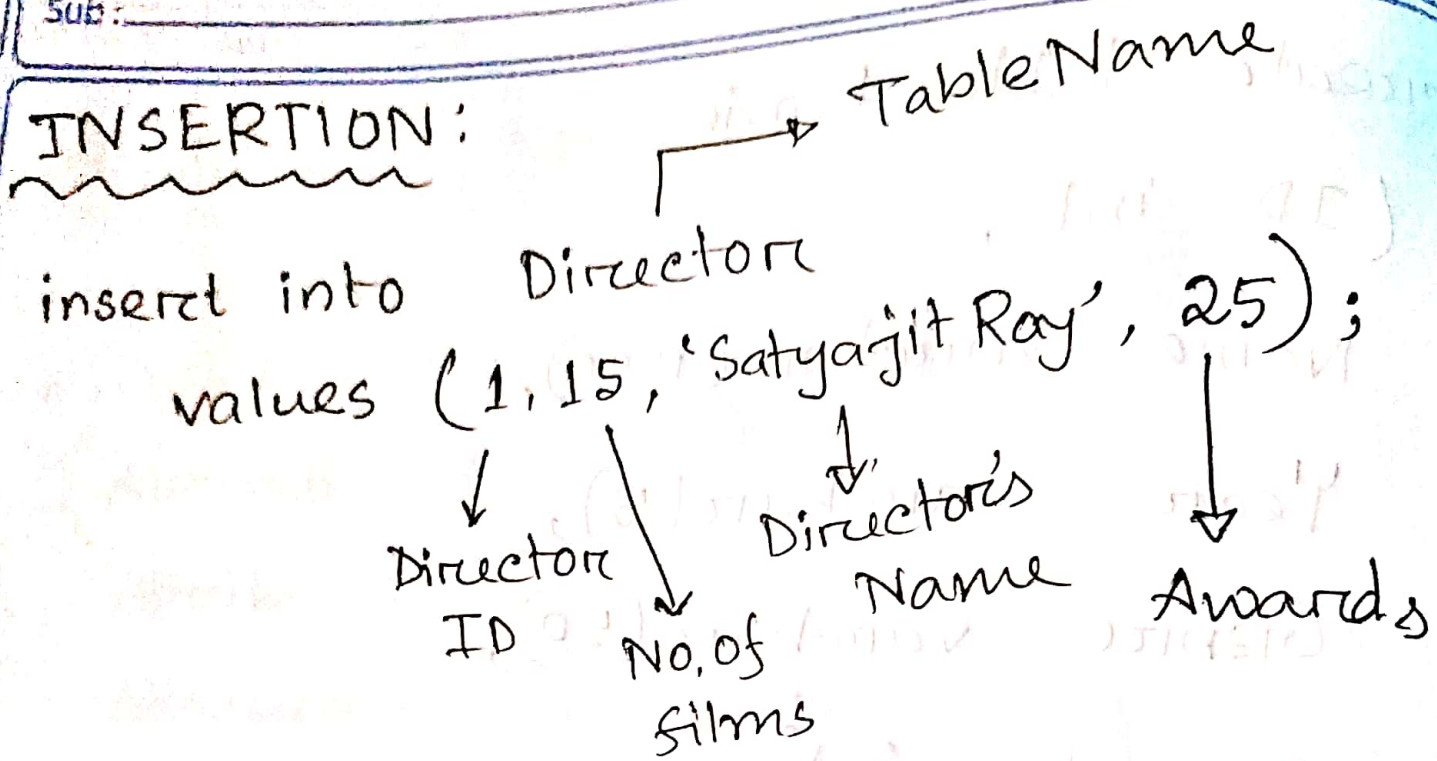
```
create table Movie
(ID int,
Name varchar(1000),
Year varchar(5),
Genre varchar(50),
Awards int,
IMDBrating numeric(3,1), DirectorID int,
primary key (ID),
foreign key (DirectorID) references (Direct
);
```

CHAPTER-2 → Student Database Schema Diagram

Day
Time :
Date : / /

Sub :

INSERTION:



create Database Movie_Info,

Director (DirectorID int, Name varchar(50), No.of films int, No.of Awards int, Primary Key (DirectorID))

Director

Director ID	Name	No. of films	No. of Awards

* Table drop করার জন্য -

Drop table Movie;
(Table Name)

* Database drop করার জন্য

Drop database Movie_Info;
(DB Name)

'C' Day
Cycle-03

Sub:

Day C

Time: 11:40 Date: 23/6/19

SQL:

↔

Create Table

Insert into Table values

① CHAP-3 . w3school → Resources for Queary

Silvershehatz . . . . .

* একটা Table এর Primary Key একটাই , সেটা একাধিক Column/Row মিলে হতে পারে। But তাদের attribute গুলো unique হতে হবে।

* এমন কোনো Column কে আমরা Primary Key বানাতে পারব না যার attribute unique না।

Primary Key's Attributes

→ Unique

→ Can't be Null

কোনো tuple ডিলিট করার জন্য -

Drop table TableName;

Delete from TableName;

Director

ID	Name	Awards	Films
1	A	5	500
2	B	50	50
3	C	500	5

এই table এর জন্য Delete from TableName
Director
table এর সব entry delete করে দিবে কিন্তু সব tuple নষ্ট হয়ে যাবে but Table থেকে যাবে।

Sub:

Time: Date: / /

⇒ But, Drop table Director চালালে পুরো table টাই চলে যাবে.

DDL	DML
Create Table TableName	Insert into TableName
Drop Table "	Delete from "
Create Database DBName	
Drop Database "	

Movie

↓ এটা পরে add করছি

ID	Name	Genre	IMDB rating	Awards	Box office collection
1.	A	Z	8.3	5	
2.	B	Y	8.9	10	

Sub:

Alter Table TableName ADD columnname Type;

Movie Boxoffice Num
Collection (10,

⊛ হঠাৎ করে add করলাম Column, তাই by default তারা attribute null করবে।

কিছুদিন পর Column টা আর ভাল্লাগতেছে না
Let's drop it -

Alter Table Movie Drop Boxoffice Collect

(এখানে আর Type দেয়া লাগবে না)

Sub: Day Time: Date: / /

4. এছাড়াও

Alter Table Actor ADD ActorID int primary key;

এভাবে করা যায় পরবর্তীতে Update করার মাধ্যমে আগেরগুলো ও ঐ কলামেই বসিয়ে দিচ্ছি, এটাতে যে কোনো সময়েই বানাতে পারতেছি।

5.

Alter Table Actor ADD primary key (.....);

যে যে attribute কে add করব।

→ But এটাতে Problem আছে। By default এর value তো নতুন added attribute গুলো null হবে So, unique তো হচ্ছে না।

তখন যা করতে হবে -

Alter Table Actor ADD ActorID int not

Alter Table Actor ADD primary key (Actor

ActorID	Name	Films	Awards

For foreign key :

Alter table TableName ADD foreign key (Attribute)

→ যেটাতে foreign key বসবে

foreign key এর Att.

Reference Tablename (Att.)

foreign key এর attribute

foreign key যে Table এর Primary Key

D Day

Sub: Cycle-03

Day D

Time 9:40 Date 24/6/19

Movie

(3)

ID	Name	Genre	IMDB Rating	DirectorID	BoxOffice
1	Heerak Rajar Deshe	Drama	8.8	1	100000
2	3 Idiots	Drama	8.4	3	50000000
3	Jibon theke neya	History	9.4	2	50000
4	Pather Pachali	Drama	8.5	1	3000
5	Baishe Srabon	Mystery	8.0	4	1500000
6	Vinci Da	Mystery	7.8	4	50000

Director.

Director ID	Name	No. of films
1	Satyajit Ray	15
2	Zahir Rayhan	5
3	Rajkumar Hirani	5
4	Srijit Mukherjee	20

Sub :

Day
Time : Date : / /

⊛ Find the name, Genre & IMDB rating of all the movies

→ Select Name, Genre, IMDB_rating
from Movie
Tablename

এই Query লেখার পর নতুন একটা Table Show করবে User কে; Table be like.

Name	Genre	IMDB_rating
Heeraak...	- - - -	-

Sub:

Day

Time: Date: / /

⊛ Find the Name of all the movies:

```
Select Name
    from movie
```

So, the pattern be like-

```
Select  list of attributes
    from  list of relations
```

⊛ Find the genre of all movies

Genres
Drama
Drama History
Drama
Drama
Mystery
Mystery

```
Select. Genre
    from Movie
```

⊛ Find the distinct Genre of all Movies

O/P

Genre
Drama
History
Mystery

Query

```
Select distinct Genre
    from Movie
```

⊛ attribute এর কোনো distinct value only একবারই দেখাবে.

Sub: Day Time: Date: / /

- Find all the attributes of all movies

Query

```
Select *  → All attributes
from Movie
```

- Find Name, Genre & Box office Collection (in taka) of all the movies :

```
Select Name, Genre, BoxOffice * 83.33
from Movie
```

→ এটা Multiply করলে Dollar কে টাকায় নেয়ার জন্য

O/P:

Name	Genre	BoxOffice * 83.33

Sub:

Day
Time: Date: / /

⊛ এই যে Table টা তৈরি হলো এর Value গুলো Permanent না, So, এটা একটা Tempory Relation show করছে।

⊛ Find the Name, IMDB Rating & BoxOffice Collection of all the movies of Thriller Genre:

```
Select Name, IMDB Rating, BoxOffice
    from Movie
    where Genre = 'Thriller';
```

O/P:

Name	IMDB Rating	BoxOffice
Baishe Srabon	8.0	1500000
Vinci Da	7.8	50000

Sub :

Day

Time :

Date : / /

⊛ Find Name, IMDB Rating, Box Office Collection of all movies of Thriller Genre & containing IMDB rating greater than 8.5.

Query:

```
Select Name, IMDB_Rating, BoxOffice
    from Movie
    where Genre = 'Thriller'
    and IMDB_Rating > 8.5;
```

⊛ All rows & all columns.

```
Select *
    from TableName;
```

Sub:

* Some selected columns & all rows:

```
Select att1, att2, ....
    from TableName;
```

* All column, selected rows:
  ↳ (এক্ষেত্রে where আসবে)

```
Select *
    from TableName
    where Condition;
```

NOTES:

✓ where এর অনেক কার্ড থাকলে প্রয়োজন অনুযায়ী (and, or, not) → logical connectives use করা যায়.

✓ আর >, <, >=, <=, =, <> use করা যাবে. (<> মানে not equal)

Sub: E' Day Cycle-03 Day E Time 19:40 Date 25/6/19

Select (List of Attributes)
from (List of Relations)
where (condition)

Constraints:

1. Primary Key → (Attribute এর value গুলো not null & unique)
2. Foreign Key (Non-existent director ID)
3. Not null
4. Unique → (কোনো attribute এ একই Value আর দ্বিতীয়বার না আসার জন্য)

```
create TableName
    (Email varchar(20)
        not null unique);
```

Sub :

Day
Time :
Date : / /

Director

DirectorID	Name	Awards	Films
1	Satyajit Ray		
2	Zahir Raihan		
3	Rajkumar Hirani		
4	Srijit Mukhurjee		

Movie

ID	Name	Genre	IMDB Rating	Direct
1	Heerak Rajar Desh			1
2	3 Idiots			3
3	Jibon theke neya			2
4	Baishe Sraban			4
5	Vinci Da			4

Sub: Day Time: Date: / /

⊛ Find the Name, Genre, IMDB Rating & the Name of the Director for all the movies

(এখানে 'Name' নামের ২টা attl. আছে)

Movie table এর নাম

```
Select Movie.name, Genre, IMDB_Rating,
       Director.name
from Movie, Director
Where Movie.DirectorID = Director.Director_ID;
```

(এইটে

lation1, Relation2, ...

Cartesian Product of Relations

Like,

{A, B} × {1, 2}

{(A,1), (B,1), (A,2), (B,2)}

where টা আনার করার আগের instruction টা যেটা 4×5 → 20 tuples create করবে যার অনেকগুলা Valid না, so, validation করতে অর্থাৎ এইখানে proper movie এর সাথে respective director গুলাকে আমরা condition দিচ্ছি

Sub:

Time: Date:

(*) 3 টা Table জোড়া লাগাতে চাই -

select [ ]

$R_1; R_2, R_3$

where [ ] ;

(*) বিভিন্ন Table এ যেই attr. same জিনিস indicate করে তাদের নাম same ই রাখব.

(#) Another example - সব thriller genre এর movie:

Select Movie_Name, Genre, IMDB_Rating,

Director_Name

from Movie, Director

where Movie.DirectorID = Director.Direct

and Genre = 'Thriller';

Sub:

Day

Time: Date: / /

Natural Join: (এটাও জোড়াই দিবে Table কে, বাট আগেরটার চাইতে কিছু কম লেখা লাগবে)

↓

✓ common att. থাকা লাগবে must

✓ join করবে common att. এর ভিত্তিতে

(*) Find Name, Genre, IMDB_rating, Director_Name of all movies:

```
Select Movie_Name, Genre, IMDB_Rating,
Director_Name
from Movie Natural join Director;
```

[Note- Common Attribute টা তে Automati... বুঝে নিবে]

Update Data:

Update Movie
TableName
set Movie_Name = 'Padmavat'
where Movie_ID = 333;

ID দিয়ে update করতে হবে

Because ID টা এখানে Primary Key.

আমাদের ছিল -

Select MovieName, Box-Office-Collection *83.33
from Movie ;

যেখানে -

MovieName	BoxOffice Collection *83.33

But, আমরা চাচ্ছি, BOC-BDT এভাবে লিখব
নতুন নামটাকে বলা হয় Alias

Query can be like -

Select Movie.Name as Movie_Name, BoxOffice Col
*83.33 as BOC_BDT
from Movie;

Sub :

Day

Time : Date : / /

Alias of Relations ⟶ from Movie as M, Director as D (কিন্তু কেন? এর সুবিধা কী?)

Select M.Name, D.Name
from Movie as M, Director as D
where M.DirectorID = D.DirectorID

এই যে এইক্ষেত্রে অনেক কম লেখা লাগছে এটাই সুবিধা।

Sub:

⊛ String Operations:

'Abul'

_ (underscore) → any character (single character)

% (percent) → (0 or more characters or any substring)

⊛ All directors → name starts with a 'S'

```
select * from Director
where Name Like 'S%'
```

This is used for pattern matching

⊛ All movies → name starts with 'V'

```
select * from Movie
where Movie_Name Like 'V%
```

Sub :

Day

Time :

Date : / /

⊛ All directors → name's 2nd character is 'a'

```
Select * from Director
    where Name like '_a%'
```

⊛ All directors → if last character is 'a'

```
Select * from Director
    where Name like '%a'
```

⊛ All directors with surname → Ray

```
Select * from Director
    where Name like '%Ray';
```

⊛ All Movies → যার আগে, পিছে, মাঝখানে যে কোনো জায়গায় 2 [redacted] আছে.

```
Select * from Movie
    where Movie_Name like '%2%'
```

Sub:

Day
Page: Date: / /

(*) 2 or two থাকে।

Select * from Movie
where Movie_Name like '%2%'
or Movie_Name like '%Two%';

(*) শুরু 2য় character থাকে।

Select * from Movie
where Movie_Name like '_2%';

Movie

ID	Name	IMDb_Rating
1	Pather Pachali	8.3
2	Jibon theke Neya	9.3
3	Andhadhun	8.3
4	Vikram Vedha	8.3

Sub :

Day

Time : Date : / /

Order by :

(*) Movie Table থেকে সকল Name, IMDB_Rating নামের alphabetical order এ তথ্য নিয়ে করে,

```
Select Name, IMDb_Rating
    from Movie
Order by Name
```

Name → Attribute Name

Output :

Name	IMDb_Rating
Andhadhun	8.3
Jibonthuke Neya	9.3
Pather Pachali	8.3
Vikram Vedha	8.3

Day
Time: Date: / /
Sub:

যদি descending order এ দেখতে চাই,

```
Select Name, IMDb_Rating
    from Movie
Order by Name desc
```

এছাড়া asc লিখলে আগে যেমন হত করবে by default ascending করে,

* যদি IMDB_Rating এ desc এ দেখতে চাই -

```
Select Name, IMDb_Rating
    from Movie
Order by IMDb_Rating desc
```

এমন 9.3 এর চাইতে ছোট এরমত then 8.3 এর মুল্য table এ যেখানে যে Order এ এসে ঐখানে আসবে.

Sub: Day Time: Date: / /

এখন আমরা চাই 8.3 এর নিচে সবাইকে ascending করে. তাহলে —

Select Name, IMDb_Rating
from Movie
Order by IMDb_Rating desc,
Name asc.

* All directors → যাদের দ্বিতীয় অক্ষরে 'a' নাই.

Select *
from Director
where
Name not like '_a%'

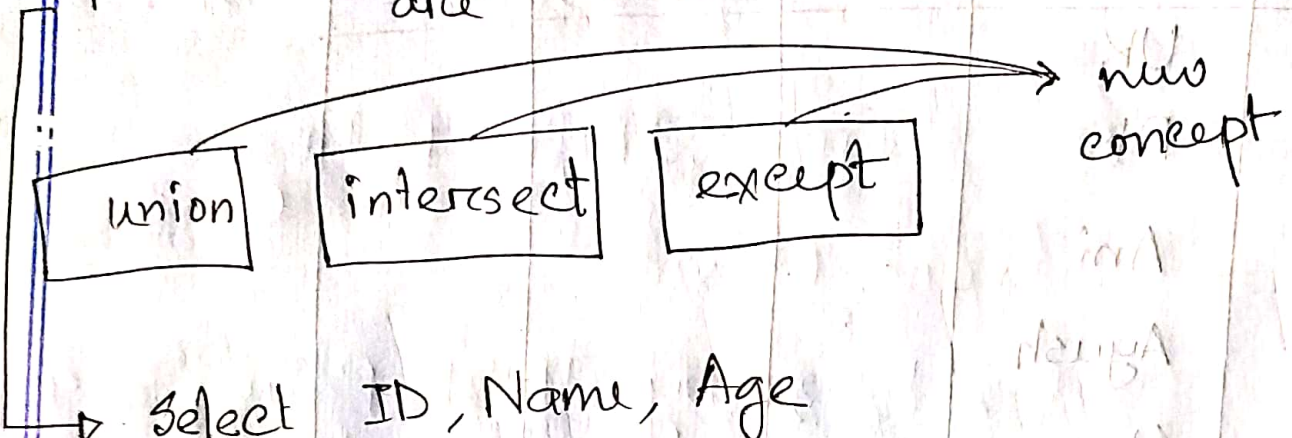
* All movies → rating between 8.5 to 9.5.

Select * from movie
where IMDb Rating >= 8.5 and
IMDb Rating <= 9.5

Sub :

Time : Date : / /

⊛ Find the ID, Name & age of all the persons $\xrightarrow{\text{who are}}$ either Nayok or Gayok



→ Select ID, Name, Age
from Nayok
Union
Select ID, Name, Age
from Gayok

⊛ Union করার জন্য দুইটার All. no. same থাকতে হবে।

⊛ যদি নামের সাথে নাম Mismatch হয় তাহলে প্রথম নামটা আসবে Like- N-Name, G-Name থাকলে N-Name আসবে।

Sub: Day Time: Date: / /

* সাধারণত Same কলাম ফেলতে চাইলে use করব।

* Common tuple ফেলতে চাইলে Intersection

* except হচ্ছে set এর A - B এর মত, A তে আছে B তে নাই type.

Null হতে সাবধান:

B০৫ * 83.33 (Arithmetic Operation)

→ But, কোনো tuple এ তার value null আছে যদি, Then, Result Null ই হবে।

→ where IMDb_Rating > 5 যদি IMDb_Rating কোনোও Null হয় then Result 3 হবে Null

Sub :

* where [ ] and [ ]
null True
দুইটা সত্য হলে Null

But,
where [ ] and [ ]
null False
দুইটা সত্য হলে False

* where [ ] or [ ]
null True
True

* where [ ] or [ ]
null False
null

Sub:

Day
Time: Date: / /

Student

ID	CT1	CT2	CT3

select ID, (CT1+CT2+CT3)/3 CT_A from Student;

Avg. marks of CT1 of all the students:

⇒ এটা করার জন্য আমাদের Tuple এর কোন দরকার নেই, এটার জন্য কিছু fune. দরকার,

Aggregate function:

takes a set of value

generates a single value as result

avg() ⟶ this is for average

count() ⟶ this is for counting the number user wants.

Sub:

Day
Time :
Date : / /

* Find the average Box Office Collection of all the movies (in the DB)

Query → Select avg (Box Office Collection)
from Movie;

This shows —

avg (Box Office Collection)
1959.05

* Thriller Genre Movie ka avg. —

select avg (Box Office Collection)
from Movie
where Genre = 'Thriller';

Sub :

Day

Time : Date : / /

⊛ Find the total number of movies in DB.

```
Select count (ID)
    from Movie
```

অথবা,

```
Select count (*)
    from Movie
```

⊛ Find the number of Movie categories :

```
Select count (distinct Genre)
    from Movie
```

E Day
Cycle-04

Sub:

Day E
Time: 9:40 Date: 27/7/19

⊞ Aggregate function:

⇒ Set of values নিয়ে কাজ করবে যেখানে কেবলমাত্র একটি Value নিতে হবে।

Such as- Avg(), Count(), Sum(), Max(), Min()

⊛ Average Box Office Collection of each movie category.

```
Select Genre, AVG(BOC)
    from Movie
    group by Genre
```

→ group এ রেখে করে নিবি

Genre	AVG(BOC)

Sub :

Day

Time : Date : / /

⊛ Find the movie categories with average BoxOffice Collection more than $1000000

```
Select Genre, AVG(BOC)
        from Movie
        group by Genre
having AVG(BOC) > 1000000
```

where : conditions for each tuple

having : conditions for each group

Order of working of DB server compiler :

1) → from (Relation)

2) → where (Tuple)

3) → group by (Grouping)

4) → having

5) → Select

✱

Select * from Movie
where IMDB_Rating >= 8.5
and IMDB_Rating <= 9.5;

অথবা

✱ between :

Select * from Movie
where IMDB_Rating between 8.5 and 9.5;

আর, between এর আগে not দিলে 8.5 থেকে 9.5 (সহ) পর্যন্ত সব Value বাদ দিয়ে বাকিগুলো নিবে,

Time: Date: / /

④ Find all the Movies of Action, Thriller, Sci-fi Genre.

এখানে, `=, Like, IN` এর ব্যাপার ম্যাপার বুঝতে হবে।

```
Select *
    from Movie
    where Genre IN
        ('Action', 'Thriller', 'Sci-fi');
```

এটা '=' চিহ্ন দিয়েও লিখতে পারতাম। কি অনেক set of Attributes এর values (বার বারতে হলে 'IN' use করাটাই Better. E এর চেয়ে ছোট করে হলেই।

Sub :

Day
Time :
Date : / /

⊛ Find all the Movies of → Name, IMDB, BoC

Satyajit Ray, Morshedul Islam, Chashi Nazrul Islam

```
select Movie_Name, IMDB_Rating, BOC
    from Movie
where Director_ID
    IN
(select Director ID
    from Director
where Name IN
('satyajit Ray', 'Morshedul Islam
    'Chashi Nazrul Islam'));
```

Sub:

Day

Time: Date: 7/7

Nested Subqueries:

```
Select Movie_Name, Genre
    from Movie
where IMDB_Rating = (Select Max(IMDB_rating)
                     from Movie);
```

Movie

ID	Movie_Name	Genre	IMDB	Director_ID
1.	Pather Pochali	Drama	8.3	1
2	Hoorak Rajar Deshe	Drama	8.8	1
3	Dipu Number 2	Drama	8.0	2
4	It	Horror	7.6	null
5	Bhooter Bhabishyat	fiction	8.2	null

Sub :

Time : Date : / /

Director

Director_ID	Director Name	Films	Awards
1	Satyajit Ray	45	55
2	Morshedul Islam	5	4
3	Anjan Datta	15	9
4	Shubhash Datta	6	2

(*) Find the Name & IMDB_Rating of the movies with corresponding Director Name

Select Movie_name, IMDB, Director_Name from Movie natural join Director using (Director_ID);

Sub :

Day

Time : Date : / /

⊛ Find the name & IMDB_rating of all the movies with corresponding Director name. (If director Name & other info. are unknown show null)

*** Outer Join:

→ Left outer join

→ Right outer join

→ Full outer join

→ Left Outer join : (Preserves all the tuples of left relation)

Select *

from Movie left outer join Director using (Director_ID);

এখানে Movie Table কে Left এ রাখায় তার সব Tuple কে Preserve করে

Sub :

এতে এর সকল Director-ID show করে, যেগুলো মিলেনি, না আসে Null দেখাবে, এক্ষেত্রে Director-ID, films, awards এ Null দেখাবে last 2টা movie এর

natural join এর জায়গায় join লিখলে

Director-ID দুইবার না দেখিয়ে একবার দেখাবে, অর্থাৎ, এক্ষেত্রে 8 টা Attribute দেখাবে, Natural join এর ক্ষেত্রে 7টা দেখাবে;

Chap-4 → শুরু হবে Join, Outer Join দিয়ে

Assignment → আগামিকাল

Class, Lab related মতামত

Sub: cycle-05

Day D

Time: 9:40 Date: 8/7/19

join:

⟷ natural join

Table: সাতদিনেরটাই follow করছি।

```
select *
    from Movie join Director using (Director_ID)
```

→ This displays-

ID	Name	Genre	IMDB_rating	Director_ID	Director Nam	Films	Awa
1	Pather Pachali			1			
2	Hkon thake neya			2			

*** null এর tuple গুলো আসবে না।
Cause তারা Match করার মত নাই 2 Table এ।

Sub:

Day
Time: Date: / /

```
select *
from Movie left outer join Director
    using (Director_ID)
```

এক্ষেত্রে,

1				1			
2				2			
3		Horror	7.2	null	null	null	null
4		Fiction	7.9	null	null	null	null

অর্থাৎ, এক্ষেত্রে যতটুকু আগে আসছে ততটুকু আসবে, এবং বাকি গুলোর null দ্বারা Replaced হবে।

Sub: Day
Time: Date: / /

(*) আর Right Outer join করলে-

Director. ID তে 3, 1, Add হচ্ছে correspondingly আর Right এ যে value গুলো নিচে.

কিন্তু left এ Movie table এ তাদের কোনো attribute নাই। so, null দ্বারা replaced হচ্ছে.

H.W

(*) Full outer join (Combination of right outer join & left outer join)

H.W → এগুলো Must বাসায় করে যেতে হবে তাহলে ক্লাস করে বোঝা যাবে।

*** Outer join এর উল্টো হচ্ছে inner join. আর এই inner join আর natural join এক একই জিনিস।

Day
Time: Date: / /

Sub:

(*) Find Movie_Name, Genre, IMDB-rating, Director Name, No. of films of the director for all the movies of Thriller, Drama & Horror Genre.

↳ জেনারে virtual relation আকারে রাখা হয়

Create view Tview as
(select M.Name, Genre, IMDb, D.Name,
D.films
from Movie join Director
using (D.ID));

Sub:

Day

Time: Date: / /

Virtual relation:

View → Purpose: security

View:

Create view ViewName as

(query);

Create view Nextview as (select Genre, count(*), AVG(IMDb) from Tview Group by Genre);

প্রতি Genre তে কয়টা tuple আছে

'D' Day
Cycle-06

Sub:

Day D
Time: 9:40 Date: 15/7/19

✱ View:

Virtual Relation

```
Create View IMPmovieInfo as
Select Movie.Name, Genru, IMDB_Rating
    from Movie
```

↓

এটার ক্ষেত্রে বেশ কিছু প্রশ্ন আছে।

✱ DB ত View তৈরি করার সময় ঐ নামে Table তৈরি হয় না। তাই এটা Virtual. That means, relation টা DB ত থাকে না.

DB system → stores → definition of view
                → SQL expression

✱ doesn't store the result of a

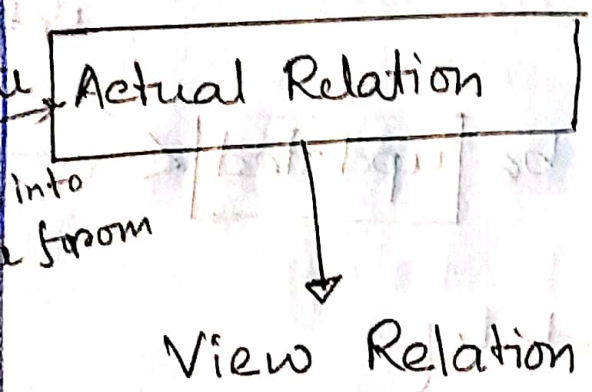
Sub:

Day

Time: Date: / /

```
Select Genre, count(*), AVG(IMDB_Rating)
from IMPmovieInfo
group by Genre;
```

Movie



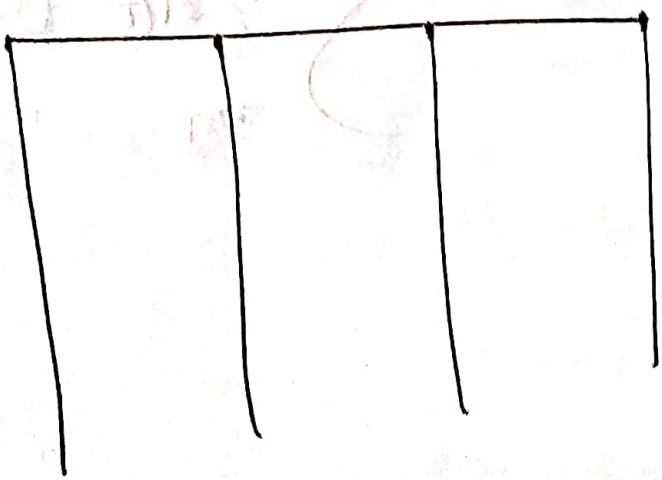
Why View?

⇒ Storage

⇒ Data Redundancy

⇒ Update (For this, SQL is stored)

IMPmovie Info.



Sub :

Day
Time : Date : / /

Materialized View: (Max. Implementation এর কোনো standard না)

→ allows the resultant relation to be stored in DB.

View Maintenance:

any change in actual relation >> insertion/deletion/update

→ View info. must be **updated**

→ actual relation update at the time of
2) View is accessed
3) Periodically

যেভাবে এই 3 ভাবে Update হয়।

Sub: Day Time: Date: / /

Update in a View:

সমস্যা - ১:

```
Insert into IMP movieInfo
    values ('Noukadubi', 'Drama', 6.8);
```

Actual Relation

Attributes >> null এ সমস্যা। কিন্তু Primary Key এ হবে না।

→ যেসব att. নাই - গুলো কি হবে?

⇒ সবগুলো null হবে।

⇒ Primary Key তে View তৈরি করে এবং declare করে।

সমস্যা - ২: Create View MovieDirectorInfo as

```
(Select Movie.Name, MovieId, IMDB_Rating
    Director_Name, Director_NooffFilms
from Movie join Director
    using (DirectorID))
```

Sub:

* এক্ষেত্রে DirectorID যে table এ foreign key হিসাবে সে table এ null হতে পারে।

উদাহরণ-৩:

Create View GenreInfo as
(select Genre, count(*),
AVG1(IMDB_Rating)
from IMPmovie Info
group by Genre);

* এক্ষেত্রে View তে Update হলেও Main Table এ Update হবে কি?
না, কারণ already group by Genre হয়ে গেছে, So, কোন কি?

'E' Day
Cycle-06

Sub:

Day E

Time: 9:40 Date: 16/7/19

View :

Update: যেখানে এ Virtual Relation থাকে সেটাকে Update করা না, আসল Table কে Update করা, But, View কেও Updat করা যায় with some condtⁿs.

View —is→ updatable

(1) If from clause has only 1 relation.

ধরি,
আমাদের DB তে Table গুলো -

Movie (ID, Name, IMDB, BOC, Genre, DirectorID)

Director (DirectorID, D_Name, Films, Awards)

View → MD

Insert into MD
values (100, 'Noukadubi', 68, 'Rituporno Ghosh'),

#Null থাকলে সবগুলোতে Null বসানো যাবে না।

Sub:

Time: Date: / /

এখানে Query-

create view GenreInfo as
select Genre, AVG(IMDB) from Movie
Group by Genre

এখন,
Insert into Genre
values ('Action', 7.2)

↳ এর বদলে না Cause Main table ↳ Avg(Imdb) বলে কিছু নাই.

(2) If select clause contains only attribute names of a relation (Not aggregate function, expression, distinct)
(3) No Group by clause.

Sub: Day Time: Date: / /

(4) Non-listed attribute values should be null

(Exception: যদি কোনো attribute আগে থেকে বলা থাকে not null, সেগুলোর ক্ষেত্রে ভ্যালু করতে ভোলে গেলে সমস্যা হবে

(2) অথবা, যদি কোনো attribute Primary key এর অংশ হয়)

Integrity Constraints: (Data)

→ Name can't be null (not null)

→ 2 different MovieID can't be same. (primary key)

→ DirectorID in movie table must match with the DirectorID in Director Table. (Foreign Key)

→ Unique (Like - {Course_Title, Dept_name})

Day
Time : Date : / /

Sub :

Check :

⇒ Domain এ restrict করতে use হয়।

Create Table Movie

(ID int, Genre Varchar(10), ...

check (Genre in ('Drama', 'Action', ...)

Genre

Genre
Drama
Action
Thriller
Horror
History

→ Genre Domain Just মনে restrict করে।

'C' Day

Cycle-07

Sub :

Day C

Time : 11:40 Date : 28/7/19

Integrity Constraints :

- Primary Key is an Integrity Constraint (Duplicate entry ক্ষে বন্ধ করে)
- not null (null হতে দেয় না)
- unique (value গুলোকে Unique রাখে)
- check (≠Domain এর Value fixed করে দে
- Foreign Key ≠ Domain কে Restrict করতেছে
- Default ≠ Particular set of values রাখবে)

```
Create Table Movie
(ID int primary key,
 IMDB_Rating Decimal (3,1) check
                (IMDB_Rating>0));
```

Foreign Key >> Referential Integrity Constraints

Referenced Relation → Director

Referencing Relation → Movie

Foreign Key এর কারণে Table গুলোর
মধ্যে Dependency তৈরী হবে। Relation
or join ও তৈরি করতে পারে।

Data Types:

) Date >> 'YYYY-MM-DD'

) Time >> 'HH-MM-SS'

) TimeStamp >> 'YYYY-MM-DD HH-MM-S
Date Time