

# RDBMS

①

DBMS  $\Rightarrow$  • DBMS interrelated data collection  
zone है।

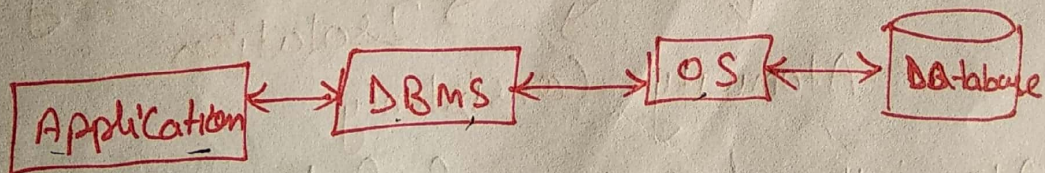
• यह ही word से मिलकर बना है।

Database + Management System = DBMS

- यह inter related information की collection है,  
जिसे use different user different purposes  
के लिए करते हैं।

Rollno.	Name	Branch	Address
1	A	IT	X
2	B	CS	Y
3	C	EC	Z

- database management system Application  
program की operating system के बीच  
interface provide करते हैं।



Application of DBMS  $\Rightarrow$

- 1) College
- 2) Hospital
- 3) Banking
- 4) Reservation

- 5) Telecommunication
- 6) Sales
- 7) Finances

## Advantages of DBMS $\Rightarrow$

1) No Repetition / Less Redundancy

2) Data Independence.

3) more secure.

4) Less Development time.

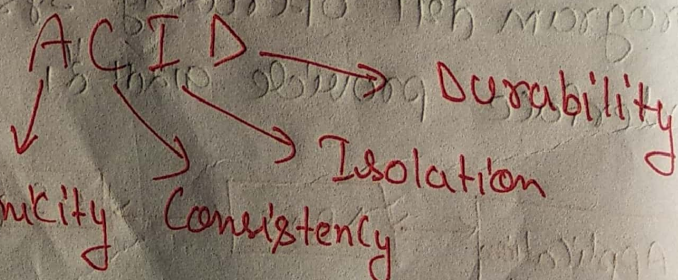
5) Easy Access.

## Disadvantages $\Rightarrow$

1) Cost of software is high.

2) Cost of hardware is high.

3) Complexity of Backup & Recovery.



• Atomicity  $\Rightarrow$  No loss of data.

A = 1000  
read(A);

A = A - 500;  
write(A);

A = 1000

A = 500

B = 1000

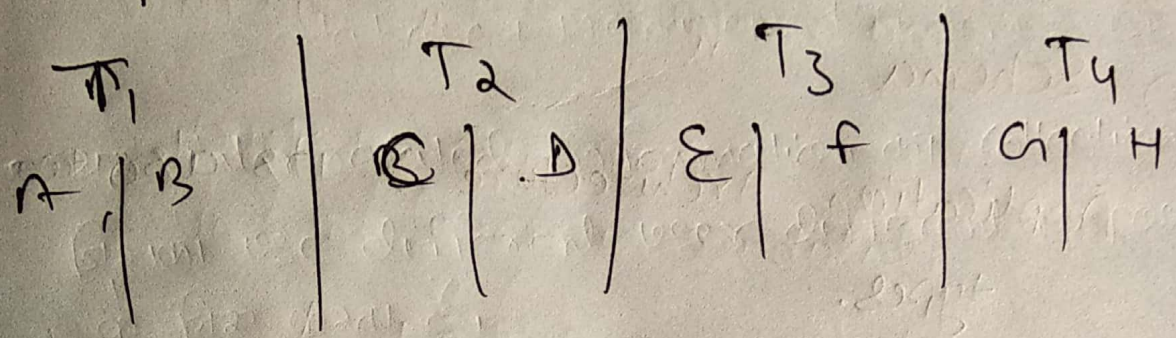
read(B);  
B = B + 500;  
write(B);

B = 1500

C = 2000

Consistency  $\Rightarrow$  initial and last data is equal  
initial sum = final sum.

Isolation  $\Rightarrow$  data is independency perform parallel.



Durability  $\Rightarrow$  Duration of time to complete the transaction.

### Difference between DBMS & file processing

DBMS	file processing
1) Repetition problem Not found	found.
2) Atomicity problem Not found	found
3) ACID problem Not found	found.
4) more secure	less secure.
5) data accessing is easily	difficult.

-11 Database Concept & Architecture ⇒

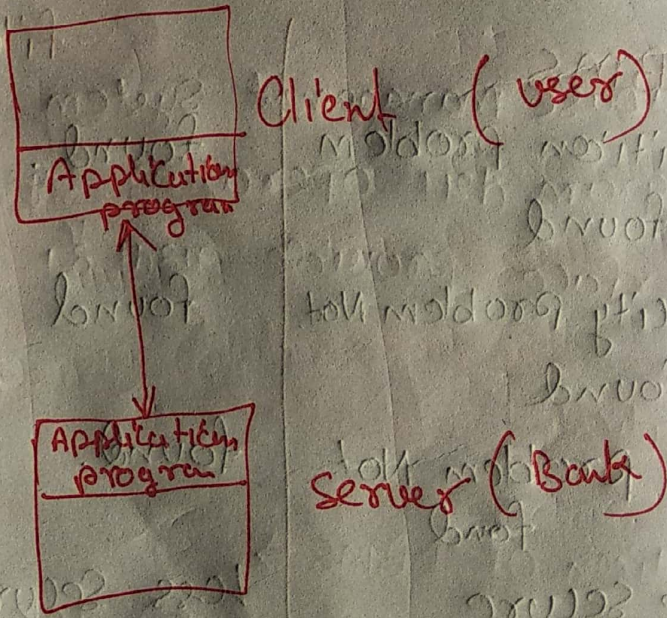
# Architecture of DBMS ⇒ Designing Structure of database are called Architects.

its two types.

- 1) Application Architecture of DBMS
- 2) Schema

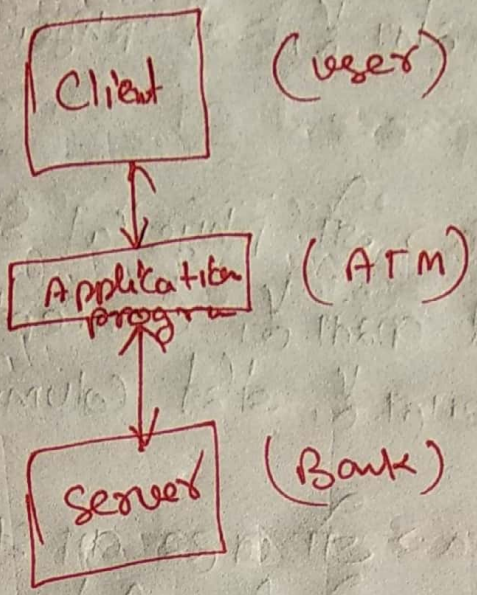
1) Application Architecture ⇒ its also two types.

1) two tier Architecture ⇒ data on client or server side & application program on client side directly communicate with server.



2) Three tier ⇒ data client server & application program work perform on client side & data on server side.

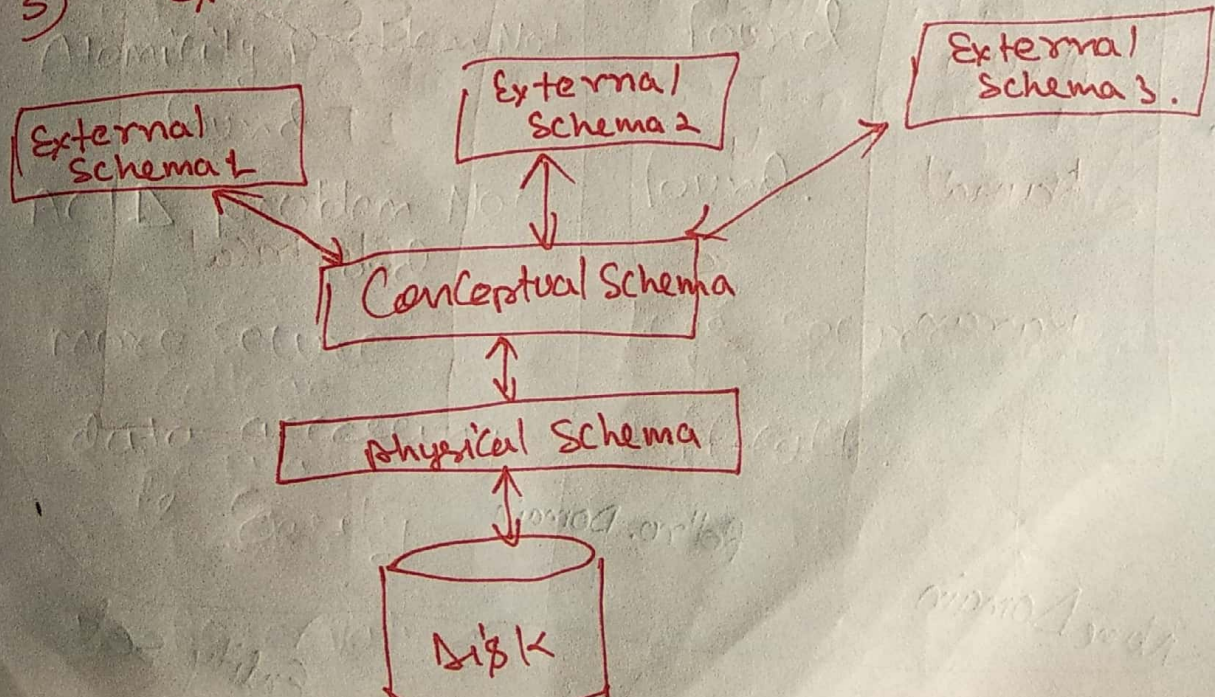
Client or send orat & )



# Schema Architecture ⇒ Overall designing of database are called schema architecture.

as a part of divided & it & )

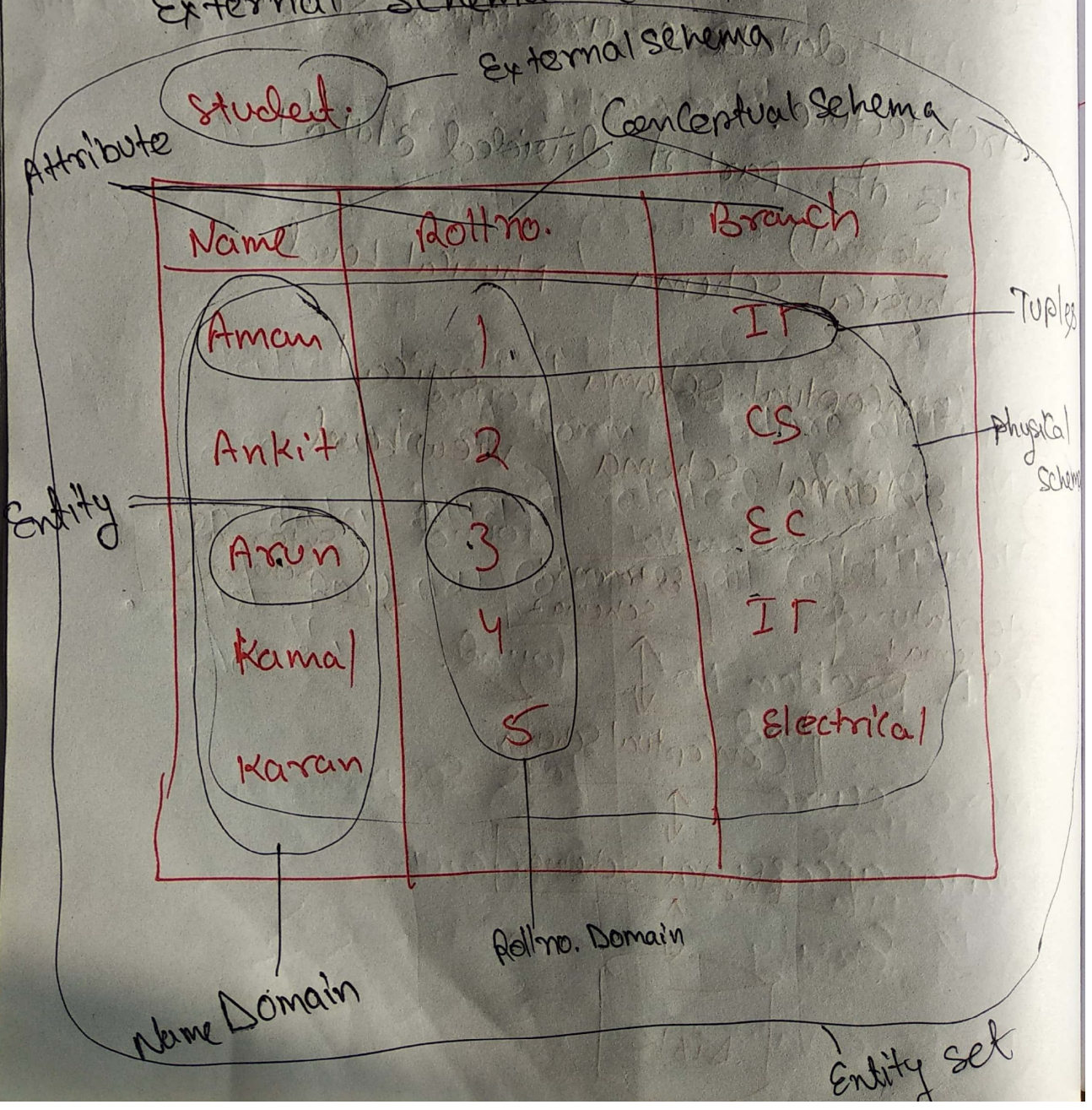
- 1) (Internal) Physical schema / physical level
- 2) Conceptual schema / Logical
- 3) External schema / view level



1) Physical Schema  $\Rightarrow$  set Actual data database  
 store data & set physical Schema  
 create &

2) Conceptual Schema  $\Rightarrow$  set physical Schema  
 not represent create & Conceptual  
 Schema create & set Column name.

3) External Schema  $\Rightarrow$  set create not structure  
 to principal database set create & not set  
 External Schema create &



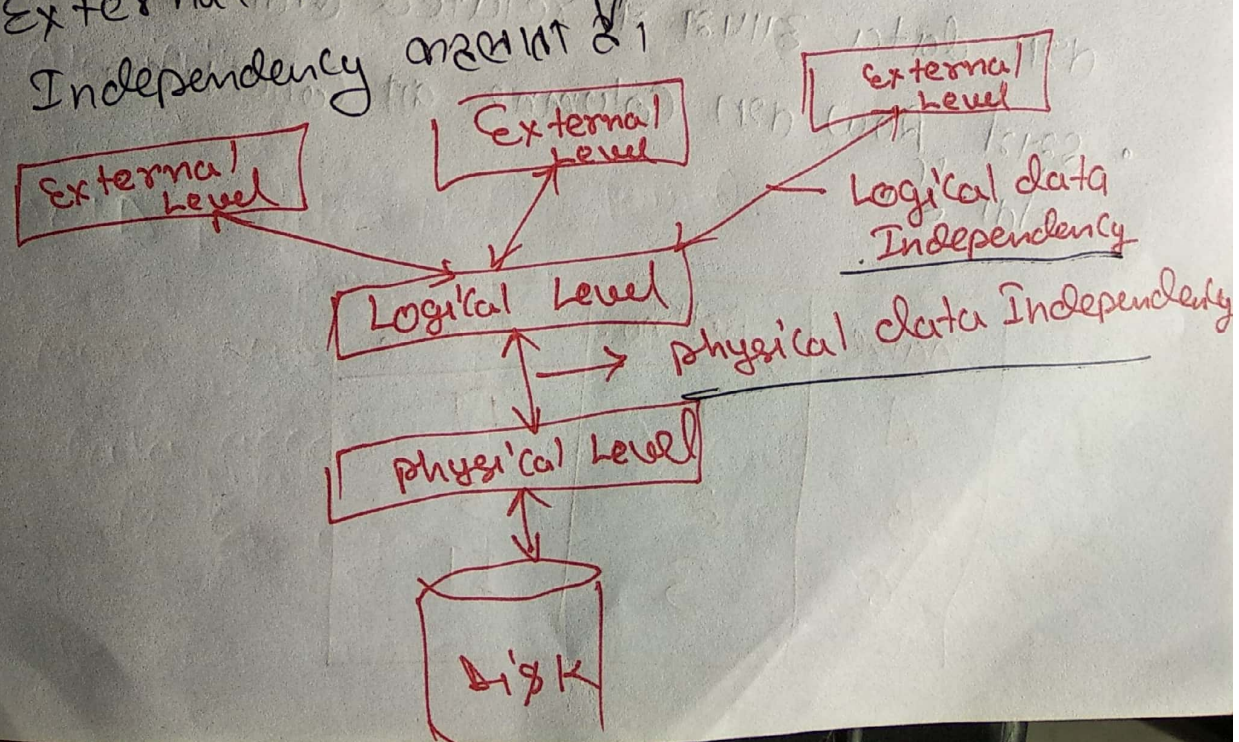
Data Independency ⇒ यदि एक Schema level में कोई changes करते हैं तो असर effect दूसरे Schema level पर नहीं होता data

Independency.  
It's two types.

- 1) physical data Independency
- 2) Logical data Independency.

1) physical data Independency ⇒ यदि physical level में कोई change किया जाता है तो असर effect logical level में नहीं होता, physical data Independency कहलाता है।

2) Logical data Independency ⇒ यदि logical level में कोई change किया जाता है तो असर effect External level में नहीं होता, Logical data Independency कहलाता है।



# Data Models ⇒ Data models means to represent of data.

- Data ko structure Create karke data model create krte hai

Types of Data models.

- 1) Relational data model.
- 2) Object-oriented data model
- 3) Network data model
- 4) Hierarchical data model
- 5) Object relational data model.

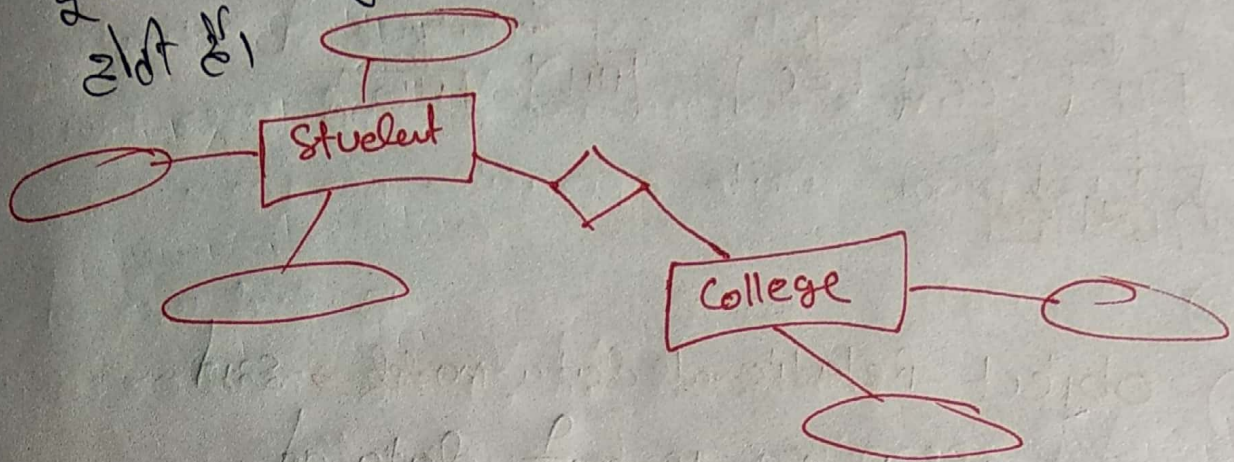
1) Relational data model ⇒ इसमें data को table की form में store होता है, जहाँ data आपस में related होता है।

- इसमें Row जहाँ Columns की Collection होता है।

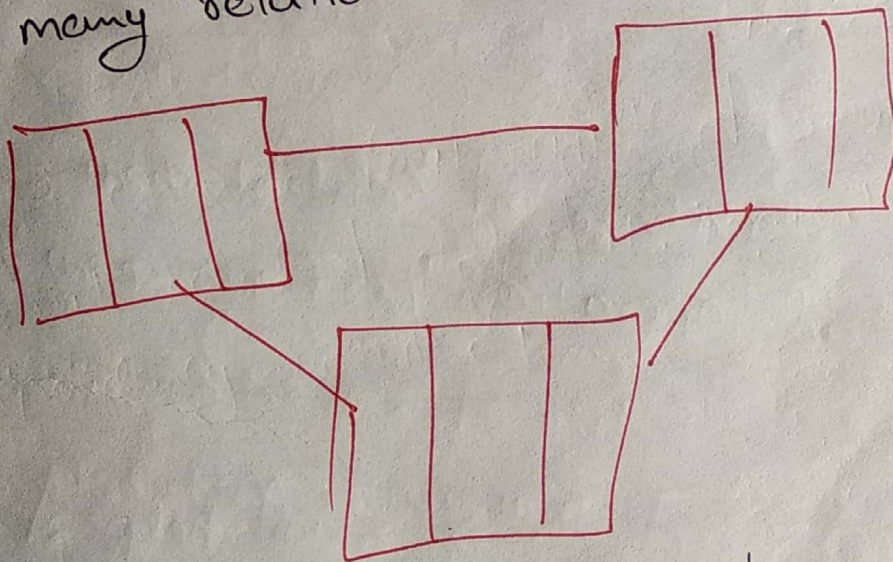
name	Roll no.	Branch
A	1	IT
B	2	CS
C	3	IT



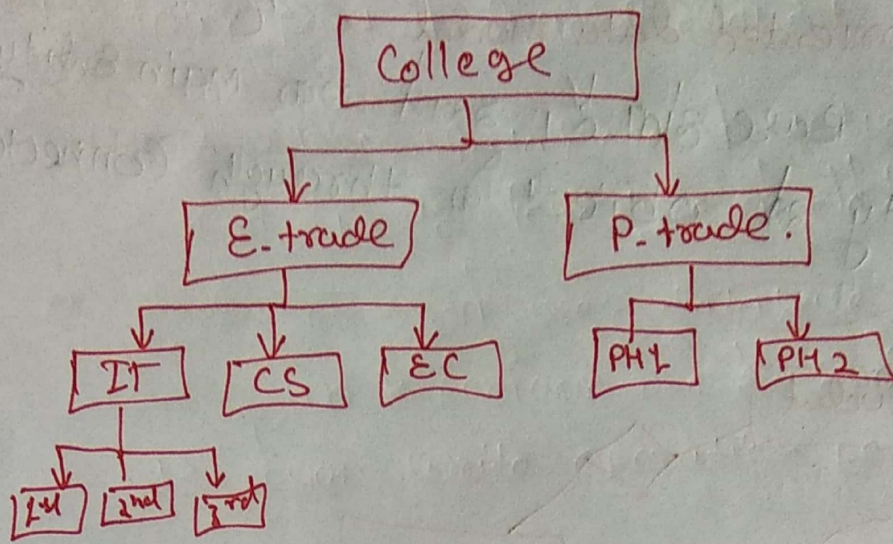
2) Object-oriented data model  $\Rightarrow$  यह E-R model  
 का Base होता है, इसमें एक main Entity  
 दूसरी Entity से object के through Connected  
 होता है।



3) Network data model  $\Rightarrow$  इसमें data many to  
 many relation के through Connected  
 होता है, इसमें बहुत सारी table होता है, जो many  
 to many relation से Connected होता है।



4) Hierarchical data model  $\Rightarrow$  इसमें data एक  
 tree structure से होता है जिसमें  
 एक root data होता है जिससे subchild होता है।



5) Object-Relational data model ⇒ सार

प्रत्येक object create मात्र data अत  
Access or retrieve प्रत्येक सार व

सार ORDBMS सार अत व

Ex ⇒ C++, Java.

# Overall database structure =>

Users

End-user  
Application Interface

Application programmer  
Application Program

Sophisticated user  
Query

DBA  
Database Schema

DML pre-Compiler

DDL Compiler

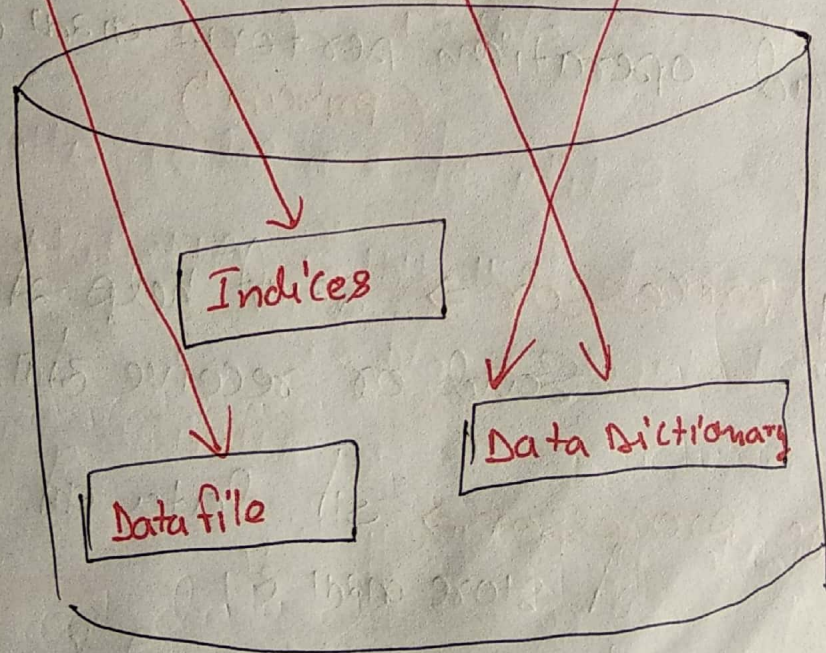
Query processor

(2) Query processor

Database manager

file manager

(3) Storage Processor.



(4)

(6)

यह four part में divided होता है।

Imp  
1) **Users** ⇒ user database में actual work perform करता है, यह four type के होते हैं।

a) **End-user** ⇒ सिर्फ user database में actual work perform करता है End-user करता है, यह data write operation perform करता है, (students)

b) **Application programmer** ⇒ यह End-user को Command देता है, यह read or write operation perform करता है, (Teacher)

c) **Sophisticated user** ⇒ यह expert होता है और Application programmer को Command देता है, यह read operation perform करता है, (Hods)

d) **DBA (Data Base Administrator)** ⇒ यह पूरा database को control करता है यह read operation perform करता है, (Principal)

2) **Query processor** ⇒ सॉफ्टवेयर की help से data send or receive होता है,

3) **Storage processor** ⇒ सिर्फ data को database में store करता है।

4) **Storage** ⇒ सिर्फ data को store करता है।

DBMS Languages  $\Rightarrow$  Database of four languages होते हैं।

(7)

- 1) DDL (Data Definition Language)
- 2) DML (Data Manipulation Language)
- 3) DCL (Data Control Language)
- 4) DQL (Data Query Language)

1) DDL  $\Rightarrow$  इसकी सहायता से database की structure में change किया जाता है But data में कोई भी changes नहीं किया जाता है,  
Ex  $\Rightarrow$  table name change  
या table को remove करना है,  
इसमें DBA perform करना।

query  $\Rightarrow$  Create, Alter, Drop.

2) DML  $\Rightarrow$  इसकी सहायता से data में change होता है But database में कोई changes नहीं होता

query  $\Rightarrow$  Insert, update, delete etc.

3) DCL  $\Rightarrow$  इसमें user, data को security access करने में काम किया जाता है।

query  $\Rightarrow$  Rollback, privacy, grant, revoke

DQL  $\Rightarrow$  इसमें data को database से Access किया जाता है।

query  $\Rightarrow$  Select.

## Data modeling using the E-R Model

E-R model  $\Rightarrow$  यह एक high level data model है, जिसमें Entity-Relationship पर work होता है।

Entity  $\Rightarrow$  कोई भी उसी चीज को हम information देती है Entity कहलाती है।

Entity set  $\Rightarrow$  यह same type की Entity का Collection होता है।

Attribute  $\Rightarrow$  Entity का एक set of attribute से represent किया जाता है। Attribute different types का होता है।

1) Key Attribute  $\Rightarrow$  यह वह Attribute होता है जो table को uniquely identify करता है।  
जैसे - Roll no. ?

2) multivalued Attribute  $\Rightarrow$  यह Attribute बिना value तक से सारा होता है।

8

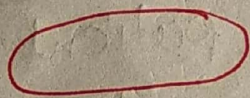
Ex  $\Rightarrow$  phone no.

3) Composite Attribute  $\Rightarrow$  यह attribute को subpart में divide हो जाता है।

Ex  $\Rightarrow$  Name  $\begin{cases} \text{fname} \\ \text{lname} \end{cases}$

4) derived attribute  $\Rightarrow$  यह attribute को दूसरे Attribute से derive कराया जाता है।  
Ex - ~~Age~~ Date of Birth से age findout होता है।

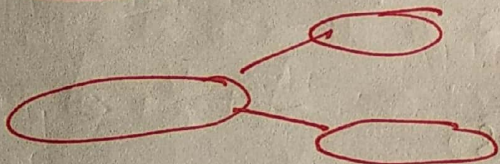
### E-R diagram Notation $\Rightarrow$



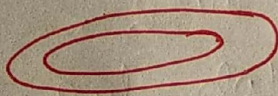
Attribute



Key Attribute



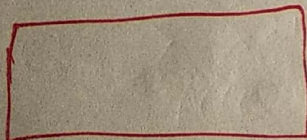
Composite Attribute



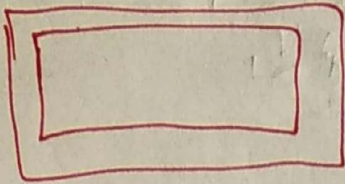
multivalued Attribute



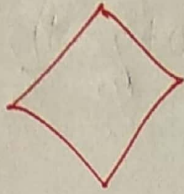
derived Attribute



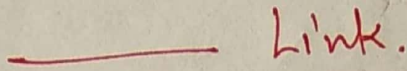
Strong Entity Set



Weak Entity Set

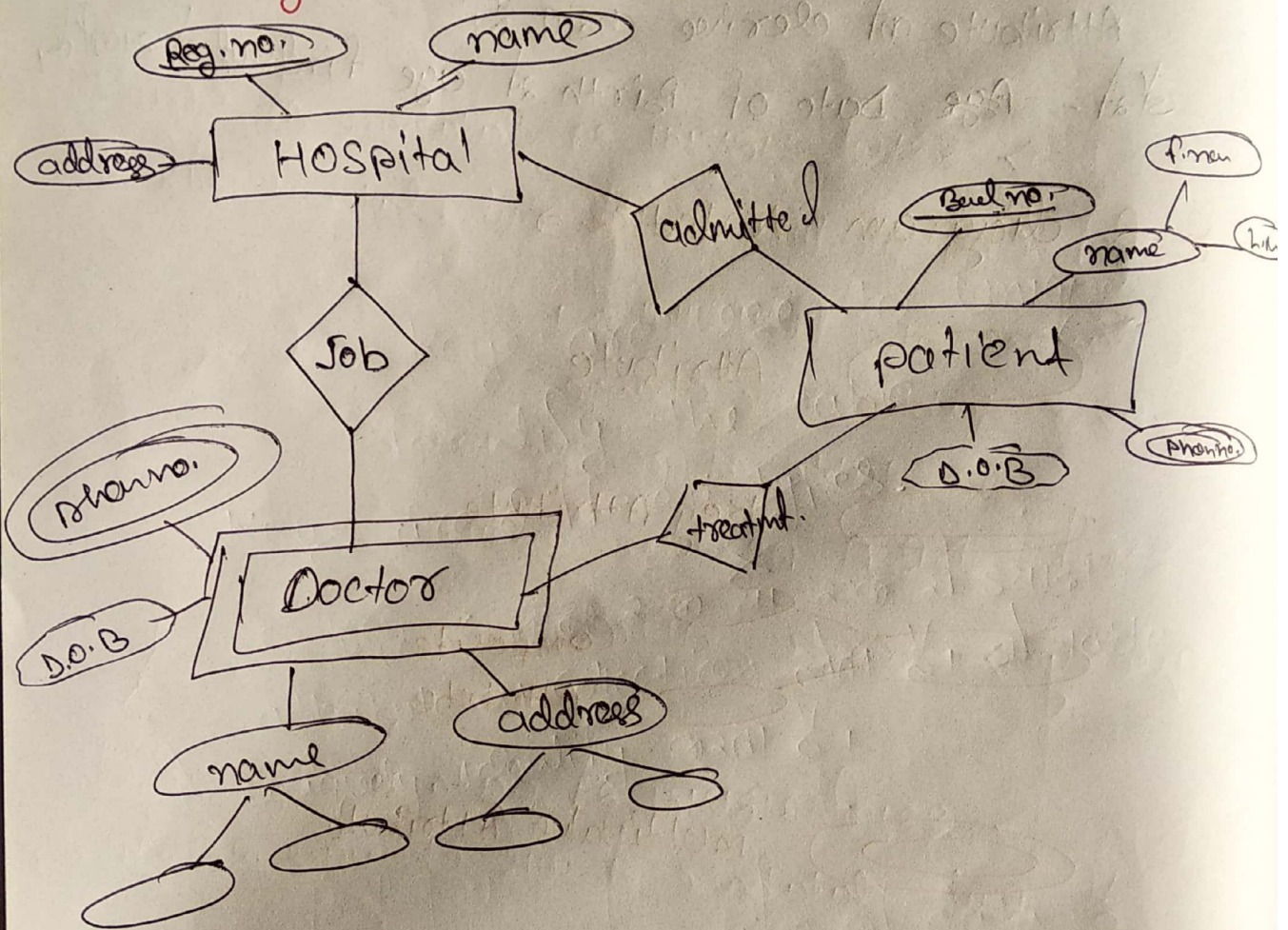


relationship



Link.

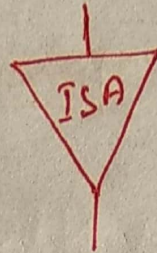
Ex ⇒ design a E-R diagram of Hospital.



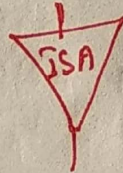


# Generalization & Specialization ⇒

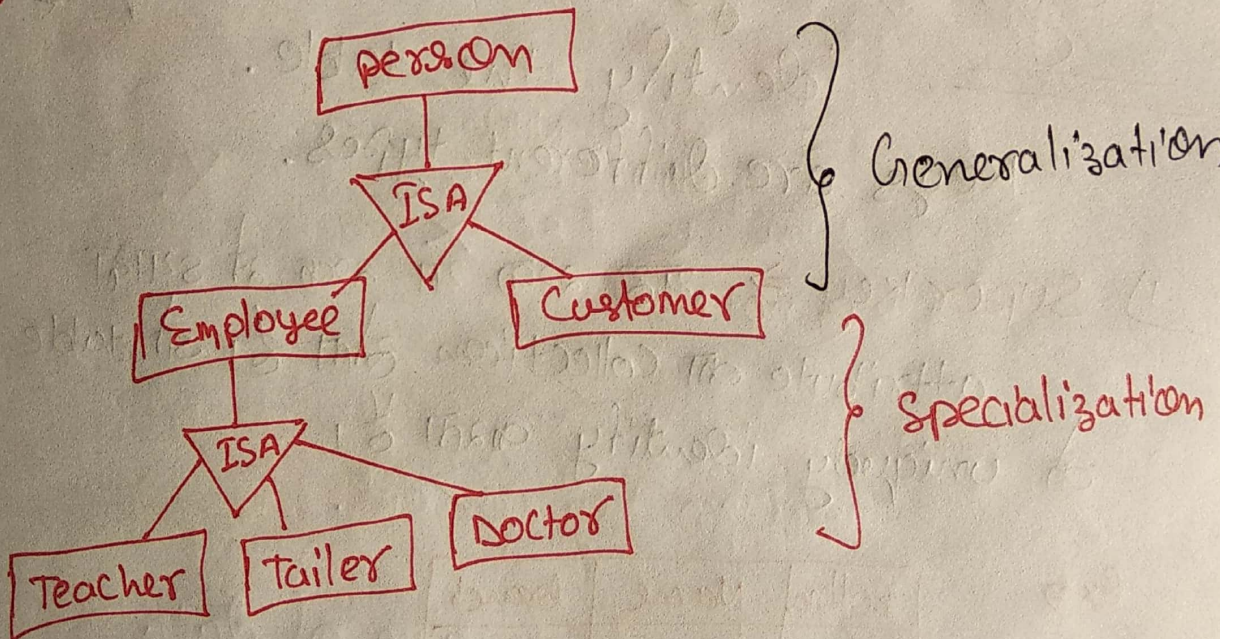
Generalization कािती भी process of General field होती है, जो इसे General value में divide करती है इसे ISA में denote किया जाता है।



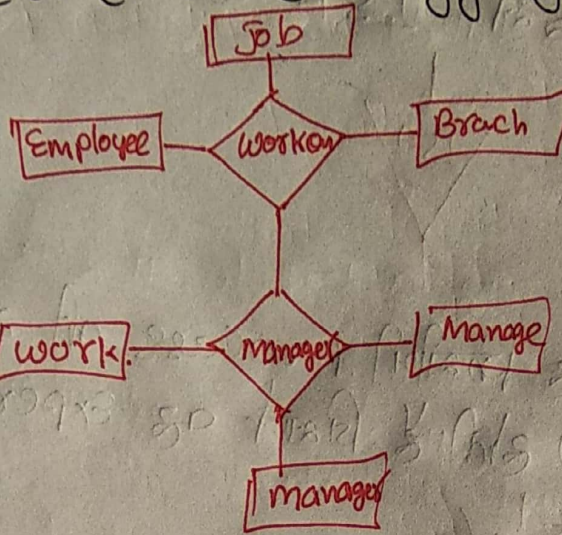
Specialization ⇒ यह किसी भी process की एक special field होती है जिसमें वह expert होती है



Ex ⇒



Aggregation  $\Rightarrow$  Relationship among relationship. या एक relation अरु एक relation अरु एक relation direct connected होला अरु अतः aggregation अरु अतः।



Keys  $\Rightarrow$  Keys are use to uniquely identify the table.

Keys are different types.

1) Super key  $\Rightarrow$  यह एक या एक अरु सारा attribute का collection होला अरु अतः अतः table कः uniquely identify अरु अतः।

Ex  $\Rightarrow$

Rollno.	Name	Branch
1	A	IT
2	B	CS
3	C	EC

Ex  $\Rightarrow S_1 = \text{Rollno.}$

$S_2 = \text{Rollno, name}$

$S_3 = \text{Rollno, Branch}$

$S_4 = \text{Rollno, Name, Branch}$

2) Candidate key  $\Rightarrow$  यह minimum no. of attributes Super key होता है जो पूरी table को uniquely identify करता है, (10)

Ex  $\Rightarrow$   $C_1 = \text{Rollno, Name}$

$C_2 \Rightarrow \text{Rollno.}$

$C_3 \Rightarrow \text{Rollno, Branch}$

3) Primary key  $\Rightarrow$  यह minimum no. of Candidate key होता है, जो पूरी table को uniquely identify करता है। यह एक table में एक ही होता है।

Ex  $\Rightarrow$   $P_1 = \text{Rollno.}$

4) foreign key  $\Rightarrow$  इसका use दो table को combined करने के लिए किया जाता है, इसमें दोनो table में पहले एक Attribute Common होता चाहिए, तथा दूसरी table में इस primary key होता जाकरी है। यदि ऐसा कोई Attribute मिलता है तो वह foreign key बन जाता है।

PK	Rollno.	Name	ID	foreign key	P.K	ID	Branch	address
	1	A	4			4	IT	X
	2	B	6			7	CS	Y
	3	C	8			9	EC	Z

foreign key = ID

# Integrity Constraints  $\Rightarrow$  data in database  
 store data in  $\Rightarrow$  Rules set  $\Rightarrow$   
 बिना Integrity Constraints  $\Rightarrow$   
 ये तीन type होते हैं।

- 1) Entity Integrity Constraints
- 2) Referential Integrity Constraints
- 3) Domain Constraints

1) Entity Integrity Constraints  $\Rightarrow$  यदि कोई  
 Attribute in primary key define  
 किया है वह  $\Rightarrow$  Null नहीं होना।

P.K

Rollno.	name	Branch
1	A	IT ✓
2	NULL	NULL ✓
3	NULL	CS ✓
NULL	B	EC ✗

P.K is not Null

2) Referential Integrity Constraints  $\Rightarrow$   
 foreign key concept.

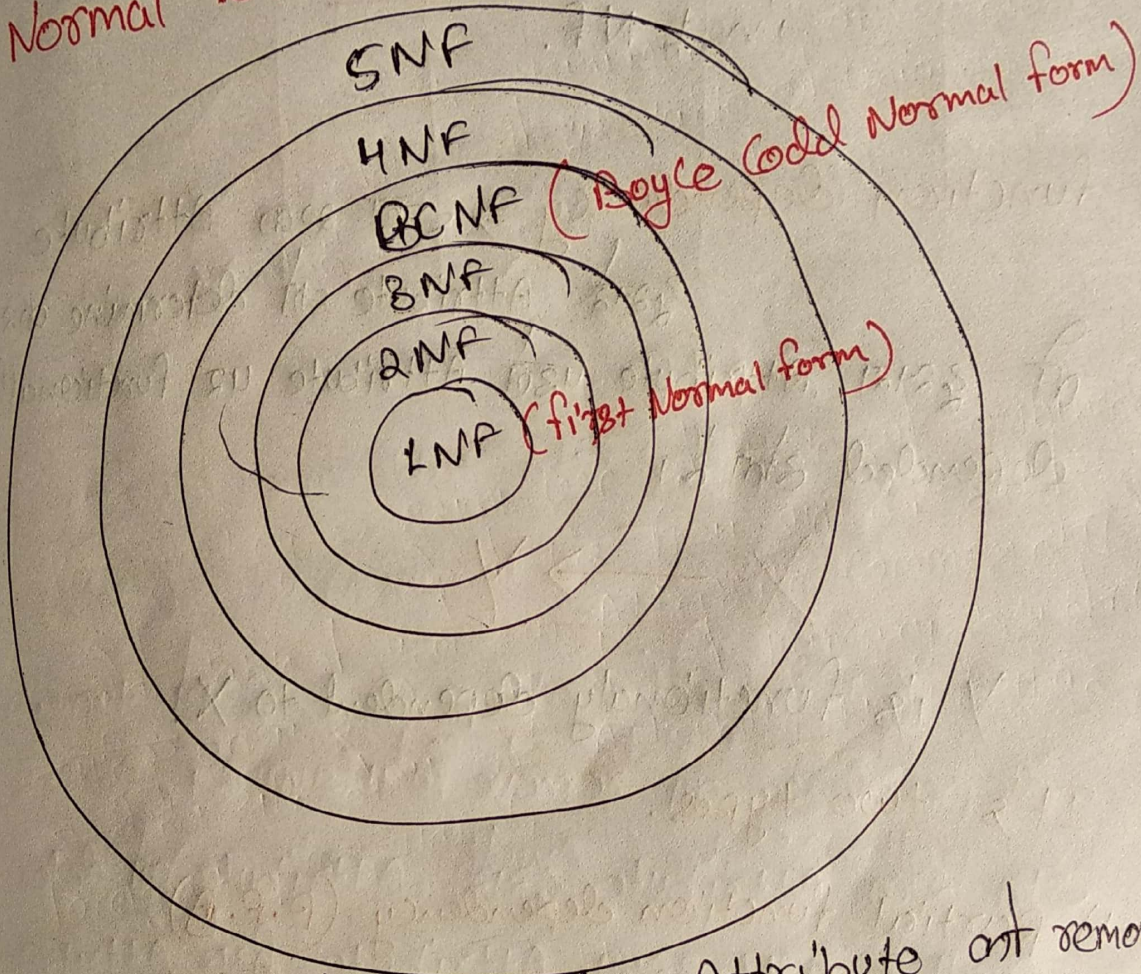
3) Domain Constraints  $\Rightarrow$  value of data  
 should be atomic. Column में  
 data का value होना चाहिए।

Name	Rollno.	Branch
A	2	IT, CS ✗
B	3	EC ✓

Normalization

Normalization एक Technique है जिसकी सहायता से database table को manage किया जाता है।

Normal forms ⇒



1) 1NF ⇒ इसमें multi value Attribute को remove किया जाता है, table में ~~सब~~ सभी data Atomic form में होने चाहिए।

Name	Rollno.	address
A	1	Dehradun, N. Nagar
B	2	Rishikesh, Dehradun
C	3	Rishikesh.

NOT in 1NF.

o Convert it 1NF by us.

Name	Rollno.	address
A	1	Dehradun
A	1	N. Nagar
B	2	Rishikesh
B	2	Dehradun
C	3	Rishikesh

in 1NF.

= function dependency  $\Rightarrow$   $\int$  Attribute  
 $\int$  Attribute can determine char of  
 Attribute use Attribute is functionally  
 depended  $\int$  Attribute

$X \rightarrow Y$

Y is functionally dependent to X.

It's two types.

1) partial function dependency (P.F.D)

2) fully function dependency (F.F.D)

1) partial function dependency  $\Rightarrow$   $\int$  Attribute  
 $\int$  Attribute can determine char of

$X \rightarrow Y$

2) fully function dependency (F.F.D) ⇒ इसमें  
 कि Attribute किसी भी data Attribute को  
 determine करत है But individual न determine  
 करत है।

(12)

$A B \rightarrow C$

$A \not\rightarrow C$

$B \not\rightarrow C$

Student.

Name	Rollno.	Branch	address
A	1	IT	X
B	2	CS	X
C	3	EC	Y
A	4	IT	Y

P.F.D Name  $\rightarrow$  Branch

F.F.D

Name Rollno.  $\rightarrow$  address

Name  $\not\rightarrow$  address

Rollno.  $\not\rightarrow$  address

# 2NF (Second Normal form)  $\Rightarrow$  सारत fully functional dependency नएत partial function dependency अत सारत-सारत अत सारत  $\&$ ,

Student 1

Name	Branch
A	IT
B	CS
C	EC

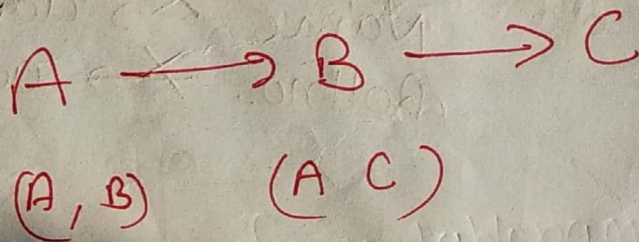
Student 2

Name	Rollno.	address
A	IT	X
B	CS	X
C	EC	<del>X</del>
A	IT	Y

2NF

← Back table Student

# 3NF (Third Normal form)  $\Rightarrow$  सारत transitive dependency अत remove अत सारत  $\&$ , सारत अत Attribute सारत Attribute अत determine अत  $\&$  नएत सारत Attribute अत Attribute अत determine अत  $\&$ , अत सारत transitive dependency अत  $\&$ .





Rollno.	Name	Branch
1	A	IT
2	B	CS
3	C	EC
4	D	EEE

NOT in 3NF  
 Rollno. → Name → Branch.

it divide into two table.

Student 1

Rollno.	Branch
1	IT
2	CS
3	EC
4	EEC

Student 2

Rollno.	Name
1	A
2	B
3	C
4	D

it's in 3NF

# BCNF ⇒ 2<sup>nd</sup> 3NF are Extended form & 1  
 Same 3NF



# SQL Unit 6

## (Structure Query Language)

1) Create ⇒ Create table tablename (Columnname1 datatype, Columnname2 datatype --- Columnnamen datatype),

Ex ⇒ Create table student (name varchar(20), Branch varchar(20), Rollno. int(10)),

2) Insert ⇒ Insert into tablename (Columnname1, Columnname2 --- Columnnamen) values (value1, value2 --- valuen),

Ex ⇒ Insert into student (name, Branch, Rollno.) value ('Ayush', 'IT', '02'),

3) Select ⇒

- 1) whole table selection
- 2) particular row selection
- 3) particular column selection

1) whole table selection ⇒  
Select \* from tablename;

Ex ⇒ Select \* from student,

(14)

2) Particular row selection ⇒

Select \* from tablename where condition,

Ex ⇒ To find Rollno. 2 student data.

Select \* from student where Rollno. = 2;

3) Particular Column selection ⇒

Select Columnname1, Columnname2 ... from tablename,

Ex ⇒

Select name from student,

4) Update ⇒

Update tablename Set Columnname = "value"  
where condition

Ex ⇒ Update student Set Name = 'Aman'  
where Rollno. = 02,