Database System

Database (collection of related data)
- Structured
  - IRCTC
  - University
- Unstructured
  - Webpages

DBMS – have collection of operations that help the user to easily perform different operations
- SQL Server (Microsoft)
- Oracle
- My SQL (Oracle)
- DB2 (IBM)

Note
90% of the data in unstructured

File System vs DBMS

Why DBMS?

a) – searching fast
– memory optimization

In file system, data will be first brought to your system let say want to know availability of train, first GB's of data brought to your system (wastage of memory)

But in DBMS with upcoming client server architecture there is requirement of DBMS, person just need to shoot a query and that result say of 1kb will be retrieved.

b) When using file system, we need the location where data is stored (like in C:/Data/Users)
but in DBMS we don't need that, we just send request through web browser / mobile application.

Expt. No. accessing data by multiple users

Date

Page No.

c) Concurrency - In DBMS, there are proper protocols defined for the situation when multiple people are accessing data at the same time
But no such provision in file system, if multiple people access at the same time then data inconsistency takes place

d) Security - In OS, there is no hierachical based security system once u logged in, you can access any data
but in ~~OS~~ DBMS, we have role based access control
(There is specific data for specific user)

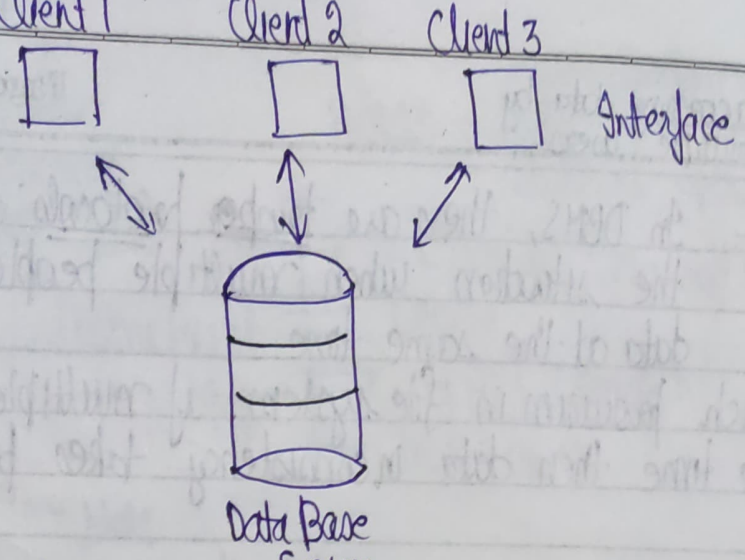
e) Data Redundancy → In DBMS, due to lot of constraints we can maintain the data integrity and as well as reduce data redundancy at the same time

Note -

File System	DB
* one user can access the data	- multiple user can access the data
* data distributed in many files	* data centrally stored and hence easy to share
* gives info of storage and representation of data	* hides internal details of database

Teacher's Signature :

2 tier Layer



- also known as
client - server architecture

- simple
- easy to mantain

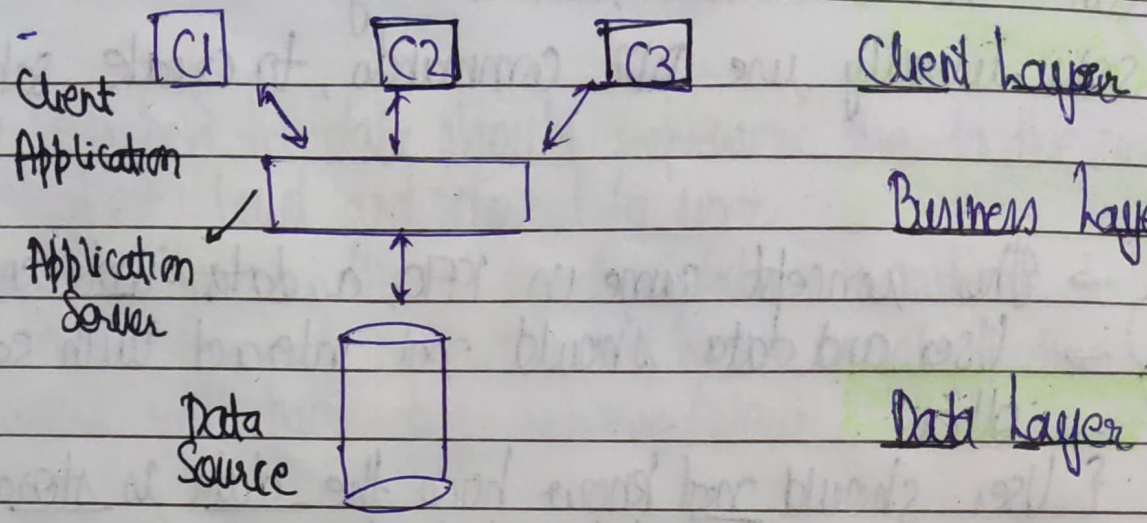
Failure - a) Scalability
- Since now there are millions of users, data can be accessed in any amount and that to at any time (Banking - Net Banking, no need to go to back to access data)
- This can not be done in 2 tier

Security

(b) Security
- Since client is interacting with data layer (Server) directly, he/she can create problem with it either maliciously or by mistake.

Expt. No. _____ Date _____ Page No. _____

3 tier Architecture



- system little complex since the addition of one more layer
- Client directly don't interact with server but interact with business layer

- Business layer - processes the whole query and ~~also~~ ask the j the data in the most simplified form from the data layer
  - can process lot of request at the same time hence the scalability can be increased

- Since the user don't know the location of the data, hence the security factor is also enhanced

Both the cons of 2-tier architecture are resolved i.e of security and scalability

Teacher's Signature : _____

Schema

→ describe about the structure

→ logical representation of data

→ can be a table or collection of data

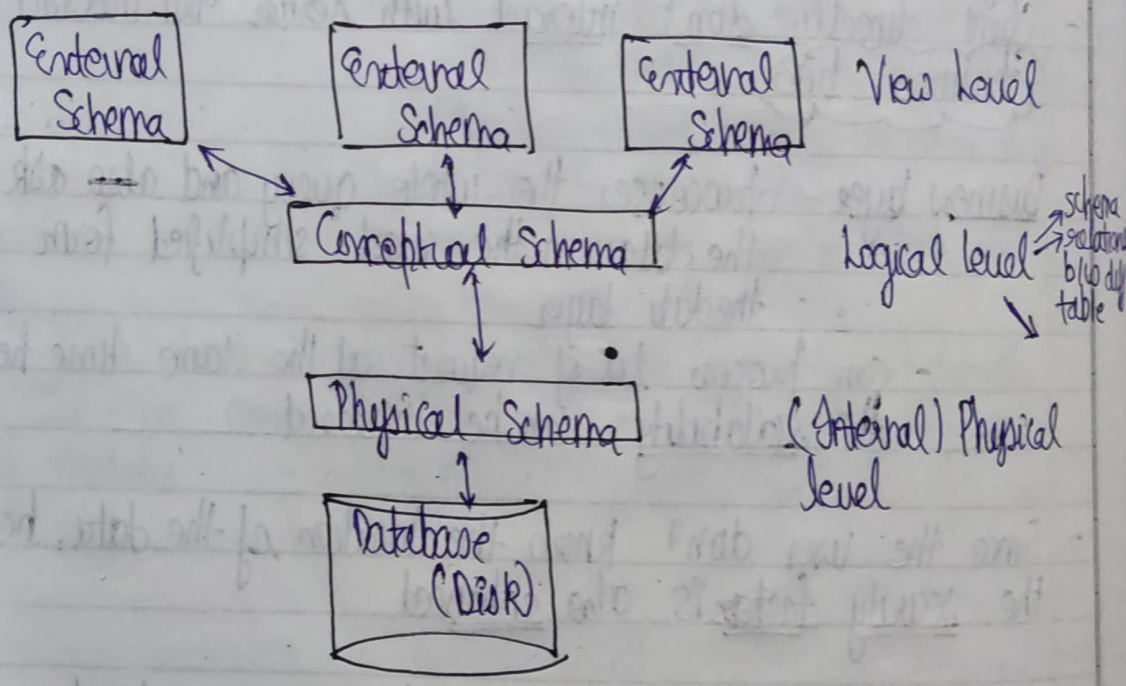
- In SQL, usually use DDL commands, to create schema

Three Schema Architecture

→ This concept came in 1970, as data independence

→ User and data should not interact with each other

{ User should not know how the data is stored, where data is stored } data abstraction



Expt. No.

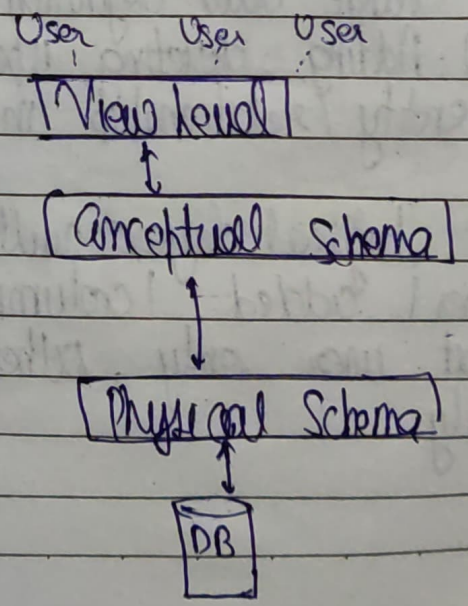
View Level
→ tells how the data should represent the to the user
→ highest level and closest to user
→ only shows the relevant database content to the users in the form of views and hides rest of the data
→ different users will have diff schema/view

eg Student, faculty, Dean - will have different external schema

Conceptual Schema level
- also known as logical level
- schema of table stored here and the relationship b/w diff table are present
- does not care how data stored in database
- DB Architect work here

Physical level
- data is physically/actually present there
- DB administrator works here

Data Independence



} logical data independence

} Physical data independence

Teacher's Signature :

Imp

& VLPS

- We have three level of schema just for the data independence purpose

→ Data Independence → ability to modify schema without affecting program/application to be rewritten

→ change in one level will not affect the data at other levels in the database

→ There are level (2) of data independence

a) Physical Data Independence
- change in physical layer will not affect the the conceptual view
- Table name, relationship b/w tables. primary, foreign key in conceptual view will not change
- Physical level Change includes
  (a) new storage device
  b) new data structure
  c) changing location of database
  d) diff data access method

b) Logical Data Independence
- Change in logical view include
  a) change data definition
  b) Adding, deleting, updating any new attribute, entity / relationship in the database

Note Changing in the logical layer will not affect the view layer. If say user 1 added 1 column, then that will be specific to that user only, others will see the previous table only

Date _____

Expt. No. _____ Page No. _____

- Key - attribute
- Use of key - uniquely identify any two tuples ✓
-

Name	City	Age
Krish	Delhi	11
Shiv	Raj.	12
Dev	Kerala	13

- attributes of table student - (Name, City, Age)

eg ~~Roll No~~, Admission No., Aadhar, License No.
These are some of the attributes that can uniquely identify a person (Set of all such attribute called Candidate key)

- The most apt key for our table from the candidate keys is called primary key

{Candidate - Primary = Alternative}
Candidate: unique
Primary: unique, Not Null

Primary Key
- uniquely identify
- Not Null

Foreign Key - It is attribute / set of attributes that references to primary Key of same table / another table (relation)
attribute → simple
set of attributes → composite

- maintains ⇒ referential integrity

Teacher's Signature : _____

Student

Roll_No	Name	Address
1	Krish	DH
2	Shiv	GOA
3	Deva	Chd.

{Referenced Table}

⇒ also called base table

⇒ Primary key is present in this table

Course

Course_ID	Course Name	Roll_No
C1	DBMS	1
C2	Network	2

{Referencing Table}

⇒ foreign key in this table

Foreign Key

- (Taking reference from the primary key of Student Table
- need to be very careful while adding data points since it is referenced with primary key

Command Create table course
{ Course_ID varchar(10),
Course_Name varchar(20),
Roll_No int references Student (Roll_No).

When add foreign Key after creation of Table

Alter table course
add constraint
foreign key (Roll.No)
references Student (Roll.No);

Note → There can be more than 1 foreign key but only 1 primary key

→ It is not necessary that the column name should be same of primary and foreign key

→ foreign key and primary key can be in same table

Expt. No. _____ Date _____

Page No. _____

Referential Integrity -

Referenced Table	Referencing Table
1) Insertion ↳ No violation	1) Insertion - May cause violation
2) Deletion ↳ May cause violation	

- If say 1st row deleted then that means, he is no more in college but referring table will say that he studying course 1. this create integrity loss

If row added with C_3, C++ and roll_no = 7, since no student with 7, hence problem will be created.

2) Deletion
↳ will not cause any violation

3. Updation
↳ may cause violation

Solution of above problem

a) On delete cascade - means when row 1 (Roll_No = 1) ~~tel~~ gets deleted then all the rows of roll no = 1 from all the referencing will get deleted.

b) On delete set null - means when row 1 (Roll No = 1) gets deleted then all rows with Roll_No = 1 in referencing table will get NULL value

(But foreign key can be Primary key also, so do keep a check)

Teacher's Signature : _____

c) On delete no action

This will throw an error since the referencing table can not able to find value in referenced table

(These three solution can be applied when applying foreign key)

3) Updation May cause validation problem

Solution

a) On update cascade
b) On update set null
c) On update no action

- All super key can not be candidate key but all candidate key is a super key

- Power Set - power set of A defined as the set of all subsets of set A including set itself and the null / empty set

- combination of all sets including itself and including null set

- If given set has n element ⇒ 2^n elements in power set

eg $\{a, b\}$ ⇒ $n \Rightarrow 2$

$P(A) = \{\{\}, \{a, b\}, \{a\}, \{b\}\}$

{ arrangment matter }

superset of any candidate key
atleast one candidate key

Super key

- It is a combination of all possible attributes which can uniquely identify two tuples in a table

- this should have atleast one candidate key

eg

Roll No	Name	Age

Candidate Key = Roll No

Super Key

Roll No, name
Roll No, Name, Age
Roll No, Age

Imp
Not Super key - Name, Age
{ because not contain candidate key

is super key

Expt. No. _____ Date _____

Page No. _____

* Super set of any candidate key is a super key

Q R(A_1, A_2 ... A_n) then how many super keys possible

(a) If A_1 is a candidate key ⇒ A_1 will remain in the set, arrangment / order does not matter

hence Power set with "n-1" elements → 2^{n-1}

(b) If A_1, A_2 is a candidate key → $A_1 \Rightarrow 2^{n-1}$

$A_2 \Rightarrow 2^{n-1}$

But when A_1 and A_2 together, then repeation takes place means

$\underbrace{A_1 A_2}\ \underbrace{____}_{n-2 \text{ left}} \rightarrow 2^{n-2}$ Ans

Hence Ans → $2^{n-1} + 2^{n-1} - 2^{n-2}$

(c) $\underbrace{A_1 A_2}_{CK_1}, \underbrace{A_3 A_4}_{CK_2}$ $2^{n-2} + 2^{n-2} - 2^{n-4}$

CK_1 → composite candidate key

CK_2 → composite candidate key

Teacher's Signature : _____

Entity Relationship Model (ER Model)

↳ conceptual design of database

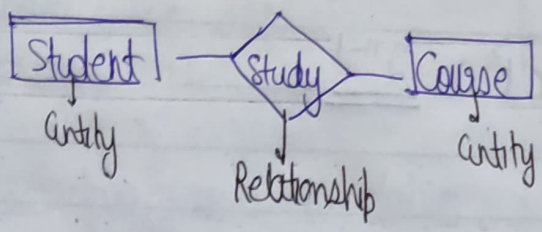
Entity ⇒ Student (Roll No, Name, Age)

Student ↑ entity; Roll No, Name, Age ↓ Attributes

- object with physical significance/existence

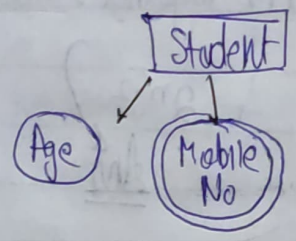
Note: □ - entity; ◇ - relationship; ⬭ - attributes

eg



Types of Attributes in ER Model

(a) Single Vs Multi Valued Attributes

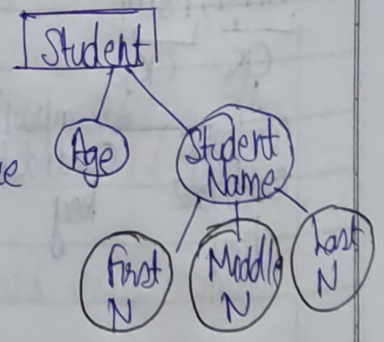


* Single Valued - have only one value
  - represent by ○
* Multi Valued - have more than one value
  - represent by ◎

(b) Simple Vs Composite attribute

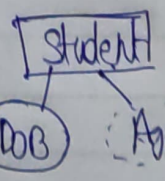
* Simple attribute - can not divide further
  - eg

* Composite attribute - composed of more than one value
  - student name < First, Middle, Last



(c) Stored Vs Derived

* Stored - can not be derived eg DOB
* Derived :→ represent using dotted lines
  → value of this attribute derived from other attribute



Expt. No. _____ Date _____

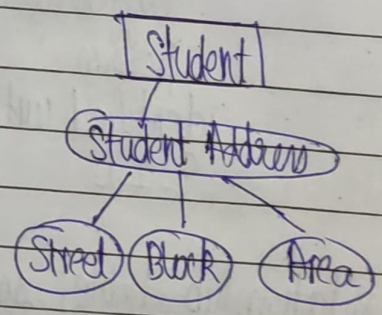
Page No. _____

4) Key Vs Non-key Attribute → Key → which is uniquely identify
→ contain always unique value (no repetition)
→ eg Reg. No
→ represent with underline

Non Key → not necessary to be unique
eg Student Name, DOB

5) Required Vs Optional Attributes → not a part of ER Model
→ Required - always required
optional - not mandatory
→

6) Complex → composite + multi valued



Types of attributes
1) single / multi valued attributes
2) Simple / Composite
3) Stored / derived
4) Key / Non-key

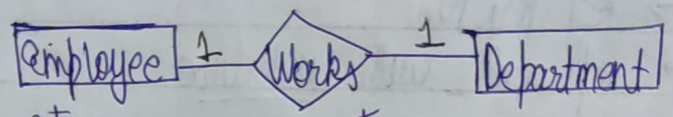
Teacher's Signature : _____

Degree of Relationship (Cardinality)

Types of Relationship – how the entities are connected with each other

Types - 1-1
1-M
M-1
M-N (Many to Many) to avoid confusion written by (M-N)

One to One Relationship →



Primary Key

act as foreign key

act as foreign key

Employee

EId	Ename	Age
E_1	A	20
E_2	B	25
E_3	C	28
E_4	A	24
E_5	B	25

Work

E_Id	D_Id
E_1	D_1
E_3	D_2
E_2	D_3

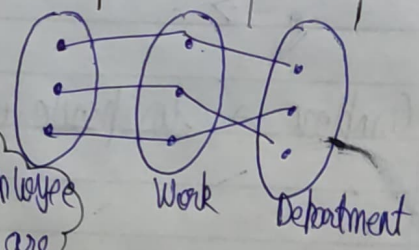
Department

D_Id	D_name	Loc.
D_1	IT	Bang.
D_2	Prod.	Delhi
D_3	H.R.	Delhi

different person identified with E_Id

{Relationship Table}

it can have many attributes but the mandatory one's are E_ID and D_Id



* In relationship table, since both have unique data, E_Id or D_Id can be the primary key (E_Id chosen as Primary Key)

* Can we reduce the ~~three~~ two tables? Yes and we get 2 tables

E_ID	E_name	age	D_Id
E_1	A	20	D_1
E_2	B	25	D_3
E_3	C	28	D_2
E_4	A	24	–
E_5	B	25	–

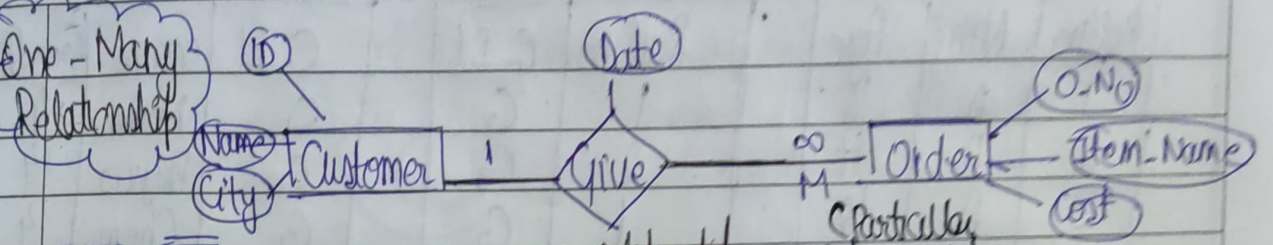
=>

{Foreign key need not be unique}

Date .....

Expt. No. ..... Page No. .....

One - Many Relationship



|| ER-Model || → into relational

Customer

ID	Name	City
C_1	A	Tal
C_2	B	Delhi
C_3	C	Mum
C_4	A	Mum

Give (PK) (FK)

ID	O_no	Date
C_1	O_1	
C_1	O_2	
C_2	O_3	
C_2	O_4	

Order

O_no	Item_name	Cost
O_1	Bucket	1000
O_2	Shoes	2000
O_3	Shirt	1500
O_4	Jeans	2000

Primary key {many side की primary key} Imp

- Relationship Table - it can also have attribute called descriptive attribute (In our eg it is Date, optional)
  - mandatory attributes are the primary key of two tables

- Yes we can reduce the 3 table into 2

ID	O_no	Item_name	Cost
C_1	O_1	Bucket	1000
C_1	O_2	Shoes	2000
C_2	O_3	Shirt	1500
C_2	O_4	Jeans	2000

- Similarly we have many to one relationship

Teacher's Signature : .....

Many to Many Relationship → Designing

{Implementing design with the help of SQL}

Student —M— Study —N— Course

(M-N)

P.K.

Rollno	Name	Age
1	A	16
2	B	17
3	A	16
4	C	17
5	D	15

(Base Table)

F.K. f.k.

Roll-no	C-Id
1	C_1
2	C_2
1	C_2
2	C_1
3	C_3

(Referencing table)

P.K.

C-Id	name	Credit
C_1	Maths	4
C_2	Physics	4
C_3	Chem	4
C_4	Hindi	4

Many to Many — — — —

Many to Many = One to Many + One to Many

from both sides

Note
:- ER diagram just a design we need to take help of language (SQL) to implement that design — —

* Primary key in Referencing Table - Formation of composite key (Both Roll-No and C-Id will be taken)

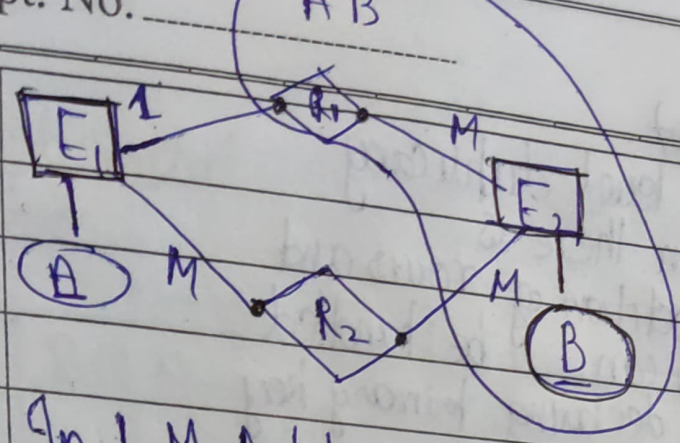
Note - We can not combine the tables, since the primary key in relationship table is composite we can not club any two tables
(Hence no reduction of table)

		In relationship table primary key	Can the no of tables be reduced
(a)	1-1	any one of the F.K.	Yes
(b)	1-M	F.K associated with many side	Yes
(c)	M-N	Composite key	No

Date

Page No.

Q



What is max. no. of tables required to represent ER model into Relational Model?

Sol In 1-M Relation, primary key of many side hence R_1 and E_2 will be combined

In M-N relation, composite key in relationship model hence no combination

In total 3 tables $T_1 = E_1$

$T_2 = R_1 E_2$

$T_3 = R_2$

Teacher's Signature :

Normalization - technique to remove / reduce redundancy from a table

(a) Row level (b) Column level

(a)

SID	S_Name	Age
1	Ram	20
2	Varun	25
1	Ram	20

redundant data

When we put SID as primary key (Not Null + Unique) → It can handle row level duplicacy

~~(b)~~ Row level duplicacy when there is repetition of rows and this can be handled by declaring primary key

(b) Column level

	SID	S_name	CId	C_name	F_Id	F_name	Salary
1	1	Ram	C_1	DBMS	f_1	John	30000
2	2	Ravi	C_2	Java	f_2	Bob	40000
3	3	Nitin	C_1	DBMS	f_1	John	3000
	4	Amit	C_1	DBMS	f_1	John	3000
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	5	⋮	C_{10}	MBBS	⋮	⋮	⋮

column redundancy

Insertion Anamoly → If new course introduced then what will we add in S_ID since the student have not enrolled now, only new course is introduced

problem in special circumstances

⇒

Deletion Anamoly - If need to delete student with Roll_No 2 but with that, course details will also be deleted.

Updation Anamoly - If need to change the salary of F_1 to 40,000 then multiple times it will change.

E.F. Codd - father of DB

Date

Expt. No.

Page No.

First Normal Form → Table should not contain any multi valued attribute

Not in 1st N.F.

Roll No	Name	Course
1	Sai	C/C++
2	Harsh	Java
3	Omkar	DBMS/C

Solution 1

Roll No	Name	Course
1	Sai	C
1	Sai	C++
2	Harsh	Java
3	Omkar	DBMS
3	Omkar	C

⇒ Primary Key - Roll_No + Course
{Composite Primary Key}

Solution 2

Roll No	Name	Course 1	Course 2
1	Sai	C	C++
2	Harsh	Java	NULL
3	Omkar	DBMS	C

⇒ Roll No - Primary Key

Solution 3

Roll_No	Name
1	Sai
2	Harsh
3	Omkar

Base table
(Primary Key exist)
Referenced table

foreign Key

Roll_No	Course
1	C
1	C++
2	Java
3	DBMS
3	C

Referencing Table
Primary Key - Roll No + Course

Teacher's Signature :

Closure Method → helps to find the candidate key in the table

Imp

eg R(A B C D) → These are the 4 attributes of the relation

relation

F.D { A→B, B→C, C→D }

⇓

(Functional Dependency)

A^+ = Closure of A ⇒ BCDA {A is determining B}

Transitive

Reflexive

What A can determine

* A→B {A can determine B}
B→C {B can determine C}
A→C {A can " C}
} Transitive Property

* $A^+ \to A$ } reflexive

A say Roll-No can determine itself

Similarly
A→C
C→D
$\boxed{A \to D}$

Candidate Key can determine all the attributes that are present in the table

A^+ = BCDA
↓
A can determine the property of all the attributes since R(ABCD), relation also has only 4 attributes. ⇒ Hence A is a candidate key

⇓

It can find the property of other attributes

* B^+ = BCD
→ B can not determine A
Hence B can not be candidate key

* C^+ = CD → C can not be candidate key

* D^+ = D ⇒ D can not be candidate key

attribute which help in making candidate key ← Prime Attribute = {A}

Non Prime Attribute = {B

Date

Expt. No. Page No.

→ $(AB)^+ = ABCD$ {Due to Transitive property AB can determine C}
$A \to B$
$B \to C$

But AB can not be a candidate key.
Since A is the candidate key and we know candidate key is always minimal}
Hence AB is a super key. {AB} ⇒ super key
↑ adding B extra
↓ Candidate Key

Q) R(ABCD)
FD = {A→B, B→C, C→D, D→A}

$A^+ = ABCD$ A is candidate key
↑↑ Transitive Approach

$B^+ = BCDA$ B is a candidate key # Candidate Key {A, B, C, D}

$C^+ = CDAB$ C is a candidate key

$D^+ = DABC$ D is a candidate key

What will be the prime attribute? → used in formation of candidate key

Ans {A, B, C, D} → Primitive Prime
{ϕ} → Non Primitive Prime

{ attribute that helps in making the candidate key }

Teacher's Signature :

Q R(ABCDE)

Imp FD = {A→B, BC→D, E→C, D→A}

Sol

BCDA — since E is missing from this, hence E will also always be there in LHS (means while making/finding Candidate Key)

present at RHS of F.D.

* E^+ = EC

E can not be alone a candidate key

AE = AEBCD ✓

BE = BECDA ✓

CE = CE ✗

DE = DECAB ✓

Functional Dependency — dependency b/w diff attributes — dependency describes relationship b/w the attributes

X → Y (X determines Y) or (Y is determined by X)

X → Determinant

Y → Dependent Attribute

S_Id → S_name

Id	Name
1	Varun
2	Dev

Valid (2 person with 2 ids)

Id	Name
1	Varun
1	Varun

Valid (Data Redundancy)

Id	Name
1	Varun
2	Varun

Valid (2 person with same name)

Id	Name
1	Varun
1	Dev

Invalid (Same Id, two names not possible)

Expt. No. _____ Page No. _____

Functional Dependency

Trivial

→ If $X \to Y$, then Y is subset of X

eg S_Id → S_Id

→ They are always valid

→ LHS of FD ∩ RHS of F.D ≠ ϕ

Non-Trivial F.D.

→ $X \cap Y = \phi$ ⇒ need to check which is valid

eg S_Id → S_name

S_Id → S_semester

S_Id → Phone no

Properties of F.D.

(1) Reflexivity : If Y is set of X then $X \to Y$

: Trivial वाला case Valid

(2) Augmentation : If $X \to Y$ then $XZ \to YZ$

eg

S_Id → S_name

S_Id, Phone No → S_name, Phone No ✓ Valid

(3) Transitive : If $X \to Y$, $Y \to Z$ then $X \to Z$

Imp

{S_Id → Name, Name → City} ⇒ S_Id → City

(4) Union : If $X \to Y$ and $X \to Z$ then $X \to YZ$ Valid

(5) Decomposition : If $X \to YZ$ then $X \to Y$ and $X \to Z$ {Valid}

(6) If $XY \to Z$ then $X \to Z$, $Y \to Z$ {Not Valid}

Teacher's Signature : _____

Pseudo transitivity → If $X \to Y$ and $WY \to Z$ then $WX \to Z$

Composition → If $X \to Y$ and $Z \to W$ then $XZ \to YW$

2nd Normal form

→ Table / Relation must be in 1st Normal Form

→ All the non-Prime attributes (attributes not participating in making candidate key) of table should be fully functional dependent on Candidate key {Non prime attribute should be dependent variable and on the Candidate key}

eg

Customer Id	Store ID	Location
1	1	Delhi
1	3	Mumbai
2	1	Delhi
3	2	Banglore
4	3	Mumbai

- Candidate Key ⇒ Customer ID + Store ID (composite)
- Prime Attribute ⇒ Customer Id, Store Id
- Non Prime Attribute ⇒ Location

Store Id → Location {But not customer_Id}

(Location: Non Prime attribute)

Solution to change in 2nd NF

Store ID	Location
1	Delhi
2	Banglore
3	Mumbai

Candidate key; Location: {Non Prime Attribute}

Customer_ID	Store_ID
1	1
2	1
3	2
4	3
1	3

⇒ both are candidate key

Store_ID → Location

Non Prime attribute (Location) determined by Store_Id ⇒ hence non prime attribute

2nd point of 2nd NF

→ There should be no partial dependency in the relation

Rewrite as

eg (AB) – Candidate key

A → C

But A is used to determine the non prime attribute

Hence this is called partial dependency

Date

Expt. No. Page No.

Q R(ABCDEF)
FD (C→F, E→A, EC→D, A→B)

① All the candidate keys are determined
② Then all prime attributes determined

CK: EC = FADB
⇓ On LHS there must be E and C
तो आएगा ही

→ EC^+ = ECFADB → All the attributes of table are determined by EC hence it is candidate key.

Candidate key → {EC}

Since E is not present on RHS in functional dependency Hence cannot be replaced

Similarly for 'C'

Hence EC only candidate key

2. Prime attributes ⇒ E, C
Non Prime Attributes ⇒ A, B, D, F

Candidate key can determine the attribute, but part of candidate key should not determine the non prime attribute {condition of partial dependency}
eg C→F (Partial dependency)
Hence this table not in 2Normal Formal

E→A
~~EC→D~~

{Partial dependency} ⇒ LHS should be proper subset of C.K. and RHS should be a non prime attribute

Teacher's Signature :

3rd Normal form → table / relation must be in 2N form
→ there should be no transitive dependency in table

~~Non prime~~

Lecture 25

{ Non Prime Attribute should be either dependent on Candidate key or prime attribute but never ever from the non prime attribute }

Imp

Candidate Key → Non Prime Attribute ✓

Prime Attribute → Non Prime Attribute ✓

Non Prime Attribute → Non Prime Attribute ✗ (Not possible)

eg

Roll-no	State	City
1	Punjab	Mohali
2	Haryana	Ambala
3	Punjab	Mohali
4	Haryana	Ambala

Given

CK = {Roll_No}

F.D = { Roll_No → State
State → City }

Sol

~~Primtive~~ Prime ⇒ {Roll_No}

Non ~~Primtive~~ Prime ⇒ {State, City}

all the non prime attribute should be dependent on candidate key

{ NPA → NPA
State → City } Not valid

Q R(ABCD)

FD: AB → C ; C → D

CK: AB

PA: A, B

NPA: C, D

Sol AB → C

Valid because candidate key determining Non Prime Attribute

C → D

Not valid because non ~~valid~~ Prime determining Non Prime

DB DBAC

Date

Page No.

Expt. No.

Eg R (ABCD) FD: AB→CD, D→A

Sol Candidate key - {AB, DB}
Prime = {A, B, D}
Non Prime = {C}

Note For all functional Dependency
LHS of all FD is Candidate key / Super Key OR RHS is primary attribute
↓
{any one condition will work}

Note एक कोई Non prime → Non prime attribute attribute को न determine करे

BCNF (Boyce Codd Normal Form)
↳ Table should be in 3rd NF

Teacher's Signature :

ecture-36

Joins → valid only if the table have any attribute common

⇒ Cross product + select statement (condition statement) = Join

Natural Join } →

E_No	E_name	Address
1	Ram	Delhi
2	Varun	Chd
3	Ravi	Chd
4	Amrit	Delhi

Dep_no	Name	E_no
D_1	HR	1
D_2	IT	2
D_3	MRKT	4

Q find the E_name who is working in a department?

Natural Join ⇒ whenever need to equate an attribute in both the table and name should be same in both tables

⇒ eg In side question we need to equate E_no

Sol

* Select Ename from Emp, Dept where Emp.E_no = Dept.E_no

→ this depicts the cross product (here no of rows) ⇒ 12

* Select E_name from Emp Natural Join Dept

{ The attribute name should be same in both the tables which to equate }

E_no	E_name	Add	Dep_no	Name	E_no
1	Ram	Delhi	D_1	HR	1
1	Ram	Delhi	D_2	IT	2
1	Ram	Delhi	D_3	MRKT	4
2	Varun	Chd	D_1	HR	1
2	Varun	Chd	D_2	IT	2
2	Varun	Chd	D_3	MRKT	4
3	Ravi	Chd	D_1	HR	1
3	Ravi	Chd	D_2	~~MRKT~~ IT	2
3	Ravi	Chd	D_3	MRKT	4
4	Amrit	Delhi	D_1	HR	1
4	Amrit	Delhi	D_2	IT	2
4	Amrit	Delhi	D_3	MRKT	4

This is the cross product

Date

Expt. No. Page No.

Self-Join ⇒

S_Id	C_Id	Since
S_1	C_1	2016
S_2	C_2	2017
S_1	C_2	2017

Study

Two same tables are joined

SID	

Student

C_Id	

Course

Find student_Id who is enrolled in atleast 2 courses

- Join ⇒ Cross Product + Condition } (two the same tables)
- Two copies of same table will be made as T_1 and T_2
- Same table is joined

* Select T_1.S_Id from Study as T_1, Study as T_2
where T_1.S_Id = T_2.Sid
and T_1.C_Id <> T_2.C_Id

course should be different

Ans S_1 will be printed

S_Id	C_Id	STd	C_Id	
S_1	C_1	S_1	C_1	✗
S_1	C_1	S_2	C_2	✗
S_1	C_1	S_1	C_2	✓
S_2	C_2	S_1	C_1	✗
S_2	C_2	S_2	C_2	✗
S_2	C_2	S_1	C_2	✗
S_1	C_2	S_1	C_1	✓
S_1	C_2	S_2	C_2	✗
S_1	C_2	S_1	C_2	✗

Teacher's Signature :

Equi Join → find the emp-name who worked in a dept having location same as their address

E-No	E-name	Address
1	Ram	Delhi
2	Ravi Kumar	Chd
3	Ravi	Chd
4	Amrut	Delhi

Dept-No	Location	E-no
D_1	Delhi	1
D_2	Pune	2
D_3	Patna	4

→ Cross product

Select E-name from Emp, Dept
where Emp. E-no = Dept. E-no
and Emp. Address = Dept. Location ;

Equi Join → Just equal to sign b/w any operator

Left Outer Join → Natural Join + left table वाला सारा data { but not in right table }

(वो दोनों table में common हैं who equate हुआ)

⇒ gives the matching rows and row which are in the left table but not in right table

left outer join

natural join

Emp

Emp No	E Name	Dept no
E_1	Varun	D_1
E_2	Amrut	D_2
E_3	Ravi	D_1
E_4	Nitin	—

Dept

Dept no	D-name	Loc
D_1	IT	Delhi
D_2	HR	Hyd
D_3	Finance	Pune

all the data of left outer join

Select emp_no, e_name, d_name, Loc
from emp left outer join dept
on (emp. dept_no = dept. dept_no)

{ Outer Join Join Join }

Date

Expt. No. Page No.

Result

Emp_No	E_name	d_name	Loc
E_1	Varun	IT	Delhi
E_2	Amrit	HR	Hyd
E_3	Ravi	IT	Delhi
E_4	Nitin	—	—

Right Outer Join - it gives the outer matching rows and rows which are in right table but not in left table

Natural Join +

Right outer Join

Select emp_no, e_name, d_name, Loc
from emp right outer join dept
on (emp.dept_no = dept.dept_no)
Condition of Natural Join

Emp

Emp_no	E_name	Dept_no
E_1	Varun	D_1
E_2	Amrit	D_2
E_3	Ravi	D_3

Dept

Dept_no	D_name	Loc
D_1	IT	Delhi
D_2	HR	Hy
D_3	Finance	Pune
D_4	Testing	Noida

Result

emp_no	e_name	d_name	Loc
E_1	Varun	IT	Delhi
E_2	Amrit	HR	Hy
E_3	Ravi	Finance	Pune
—	—	Testing	Noida

} Natural Join ki answer

} row in right table not in left table

Teacher's Signature :

Relational Algebra (1970) → basic of SQL is relational algebra

→ Procedural Query Language → theoritical model of combination of different operations

OR

→ formal Query Language

→ Need to tell
what to do ?
+
How to do ?

Operations ors.

- Basic operator
  - → Projection (π)
  - → Selection (σ)
  - → Cross Product ($\times$)
  - → Union ($\cup$)
  - → Rename (ρ)
  - → Set Difference ($-$)
- Derived operator
  - → Joins ($\bowtie$)
  - → Intersect ($\cap$)
    - $(X \cap Y) = X - (X - Y)$
  - → Divisions ($/, \div$)

Projection (π) → work always on columns

Roll_No	Name	Age
1	A	20
2	B	21
3	A	19

→ used to retrieve the data
→ used to give unique output

Query : $\pi_{Roll\ no}$ (Student) → help to retrieve roll_no from table student

: $\pi_{Roll\ No, Name}$ (Student) → help to retrieve roll_no, Name from table student

} Just the mathematical expression

Expt. No. _____ Date _____

Page No. _____

* $\Pi_{Roll\ No}$ (Student)

Roll-No
1
2
3

* Π_{Name} (Student)

Name
A
B

} unique element will only be printed.

Selection (σ) ⇒ Query ⇒ retrieve the name of student whose roll no = "2";

name ⇓ projection operation

{ works always on rows / tuples }

$\Pi_{Name}(\sigma_{Roll\ No = '2'}(Student))$

generally projection comes in last

Teacher's Signature : _____

Cross/Cartesian Product

A	B	C
1	2	3
2	1	4

C	D	E
3	4	5
2	1	2

- No. of columns ⇒ 3+3 = 6
- No. of rows ⇒ 2×2 = 4
- atleast one column should be common for joining

A	B	C	C	D	E
1	2	3	3	4	5
1	2	3	2	1	2
2	1	4	3	4	5
2	1	4	2	1	2

Set Difference (−)

- (A−B) = A but not B
  ⇒ $A \cap B'$
- not commutative $A-B \neq B-A$

$S_1 = \{1,2,3\}$
$S_2 = \{3,4\}$
$S_1 - S_2 = \{1,2\}$
$S_2 - S_1 = \{4\}$

- No. of columns must be same in number
- Domain of the every column must be same

Roll No	Name
1	A
2	B
3	C

Student

Emp No	Name
4	E
1	A

Employee

* Name of person who is student but not employee [π_{name}(Student) − π_{name}(Employee)]

Expt. No. _____ Date _____

Page No. _____

Union

$S_1 = \{1,2,3\}$

$S_2 = \{3,4,5\}$

$S_1 \cup S_2 = \{1,2,3,4,5\}$

→ no of columns / attributes must be same

→ domain of every column must be same

Refer to previous table

Student ∪ employee

Roll No	Name
1	A
2	B
3	C
7	E

Attributes of the first table

Note when the Name and emp_No attributes are interchanged in the employee table and then we do the union, then it will cause error since it will not able to do mapping properly

Division Method (÷)

- derived operator

→ Query - Retrieve S_ID of the student who enrolled in all the course

S_ID	C_ID
S_1	C_1
S_2	C_1
S_1	C_2
S_3	C_2

enrolled

C_ID
C_1
C_2

Enrolled Course

* this operator used when "all" key word is used

* E(S_ID, C_ID) / C(C_Id)

Denotion of division method

* A(X, Y) / B(Y) - it results X values for that there should be tuple <X, Y> for every Y value of relation B

E(S_ID, C_ID) / C(C_ID)

enrolled table — course table → this is the helping table

this is the table for which the output is retrieved

⇒ Division operator will give those values of S_ID, who have enrolled in all the courses that is present in the course table

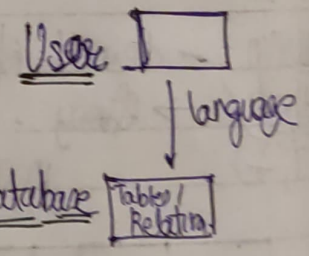
$$\Pi_{sid}(E) - \left[\Pi_{sid}\left[\left(\left(\Pi_{sid}(E)\right) \times \left(\Pi_{cid}(C)\right)\right) - (E)\right]\right]$$

expansion of E(S_ID, C_ID) / C(C_ID)

SEQUEL (simple english QUEL)

SQL (Structured Query Language)

- → domain specific language (only for relational database)
- → declarative language (need to tell what to do but not how to do)
- → DDL, DML, TCL, DCL
- → Keys and constraints
- → Operators (Like, In, not in, exist, conditional)
- → Clauses (order by, group by, from, having, distinct)
- → Aggregate fun^n
- → Joins and Nested Queries



Note Advanced Version of SQL (so R SQL

Procedural language SQL

→ need to tell what and how to do

→ can create trigger fun^n and cursor procedures

Date

Expt. No. Page No.

Tuple Calculus

Relational Calculus

Tuple relational Calculus (Row specific)

Domain Relational Calculus (Column Specific)

→ non procedural query language {need to tell what to do}

→ In tuple calculus, query is expressed as

$\{t \mid P(t)\}$

↓ predicate and these are used to fetch t

Thus it generates set of all tuples t, such that predicate P(t) is true for t.

Operations →

Teacher's Signature :