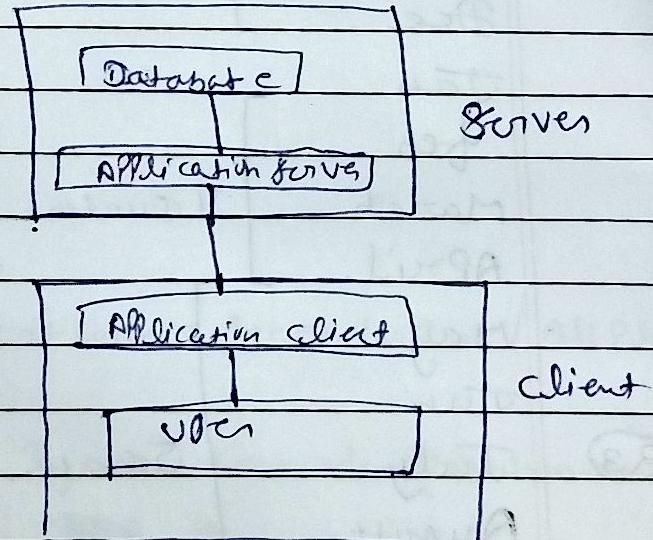


SQL Commands

1 ↓	2 ↓	3 ↓	4 ↓	5 ↓
DDL	DML	DCL	TCL	DQL
[Data Definition]	[Data Manipulation]	[Data Control] Change	[Transaction Control] Change	[Data Query]
Create	* Insert	* Grant		
Alter	* Update	* Revoke	Commit	Select
Drop	* Delete		Rollback	
Truncate	*		SchreibPunkt	Atom.
			Autocommit	

OBNS Architecture: —



~~Scaling Patterns~~ — 7 Patterns to Managing Data Load.

ACID Property

— * stands for —

* Atomicity

* Consistency

* Isolation

* Durability

* Maintain the integrity of data.

* Mainly use in transaction

Atomicity

— (Rollback)

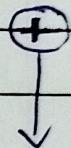
fail + transaction can't be recovered

fail + transaction always restart.

transaction - gg * x
- 100 ↴Consistency

Before transaction start

After transaction complete



$$\frac{T_1}{100} = \frac{T_2}{100}$$

Sum of transactions
(many) should be same.Isolation : — Conversion of Parallel schedule to Serial schedule because serial schedule more consistent.Durability — All changes in database should be permanent.

Commit → Command we use.

SQL foreign key constraint :-

* used to prevent actions that would destroy links b/w tables.

* it is a field in one table that refers to the Primary Key in another table.

Create Table Orders (

orderid int Not Null, Primary Key,

ordernum int Not Null,

PersonId int,

~~Primary Key to~~

foreign key (PersonId) References Persons
(Person_Id)).

Alts :-

→ Alts table Orders

ADD foreign key (PersonId) References
Persons (PersonId).

Drop a foreign key :-

Alts table Orders.

Drop foreign key for PersonOrder;

Provide essential details.

Levels of Abstraction in DB —

Data Abstraction — → goal - store/retrieve.

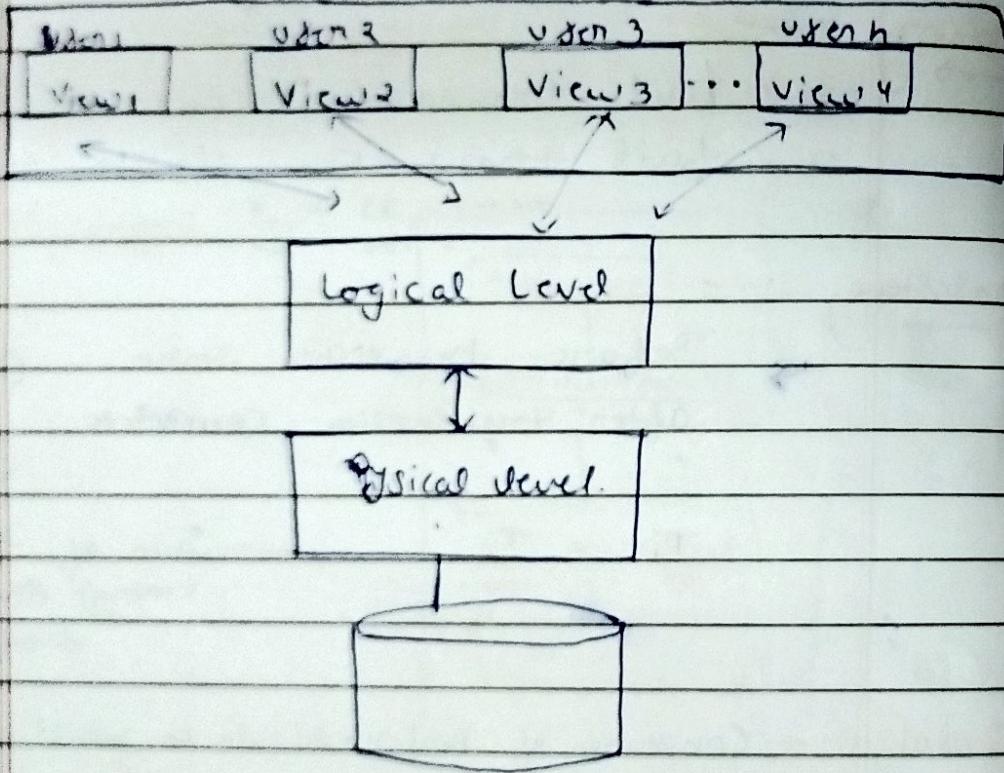
→ efficient data

→ Physical level

→ Hiding the complexity.

→ logical level

→ View level

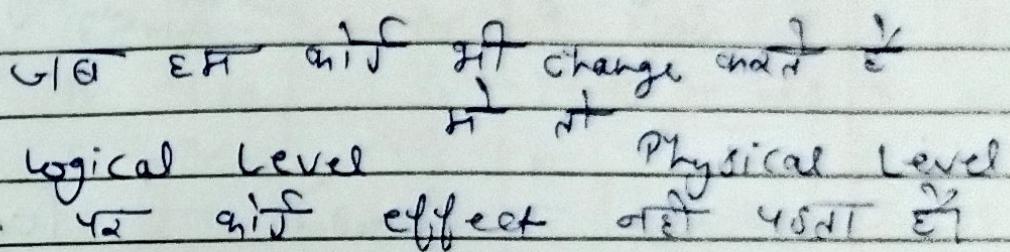


Physical level — → lowest level of abstraction
→ Deals how the data are stored.

* Complex low level data structure.

Logical Level → deals with what
the data are stored.

- * Dealt with relationship among the data.
 - * Simple Data Structure.



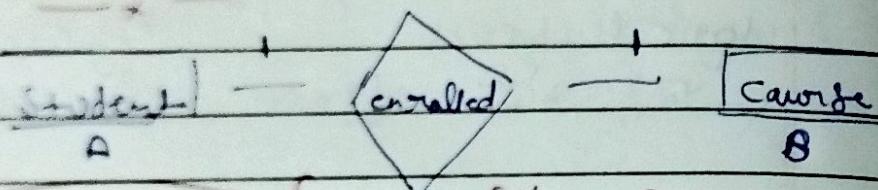
View level :-

- * Deals with Highest level of abstractions.
 - * Integration with System
 - * User / Accts.

Database Relations

- * One to One
- * Many to Many
- * One to Many.
- * Many to One
- * Self References. table had foreign key
not reference its primary key.

One to One



Roll No	Name	City	PT		Cid	Name	Duration
			Roll No	Cid			
1	A	Pune	1	C ₁	C ₁	V	2
2	B	Goa	2	C ₂	C ₂	X	5
3	C	Pune	3	C ₃	C ₃	Y	3
4	D	Mumbai	4	C ₄	C ₄	Z	4

- * is a type of Cardinality.
- * in which one element of A linked with one element of B.

* One to Many :-

Parent

has

Child.

PK

Id	Name	City	F.K		F.K		<u>PK</u>
			Id	Cd	Cd	Name	
P ₁	A	Pune	P ₁	C ₁	C ₁	X	15
P ₂	B	Goa	P ₂	C ₂	C ₂	Y	18
P ₃	C	Delhi	P ₃	C ₃	C ₃	Z	20
			P ₃	C ₄	C ₄	P	16
			P ₃	C ₅	C ₅	Q	19
			P ₃	C ₆	C ₆	R	22

Id	Cd	Name	Age
P ₁	C ₁	X	15
P ₁	C ₂	Y	18
P ₃	C ₃	Z	20
P ₃	C ₄	P	16
P ₃	C ₅	Q	19
P ₃	C ₆	R	22

* One row in table A is linked to many rows in table B.
but

The row in table B is linked to one row in table A.

Many to One

multiple

One record in

table B can be

associated with one record in table A.

Many to Many — each record in table
A associated with
multiple record in table B. vice versa.

Student ↔ Course

(1) Student

(P) Course

Student Id	Student Name	Course Id	Course Name
1	Alice	101	Math
2	Bob	102	History
3	Charlie	103	CS

Student-course

(1)

(P)

Student Id	Course Id
1	101
1	102
2	103

Concurrently — execution of multiple instruction at the same time.

Ex → transaction

Concurrency Control — Process of managing simultaneous execution of transaction.

Purpose —

- * Enforce isolation
- * Prevent database consistency
- * Resolve read/write and write/write conflicts

Part of Candidate
↳ Prime Attribute

Not Part of Candidate
Date ↳ Non Prime Attr.

Accomplish
of table
to maintain redundancy.

Normalization → Process of Managing
Redundancy in Database
from a relation.

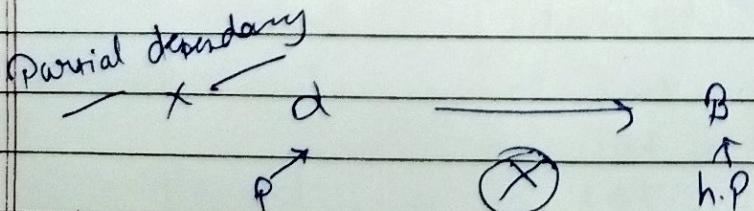
* for the normalization we use functional dependency.

* 1NF : → * every cell should be
Contain atomic value

Roll No	Name	Course
101	Rodi	CN OS
102	Sonia.	DBMS Co.
101	Modi	CN
101	Modi	OS
102	Sonia.	DBMS
102	Sonia	Co

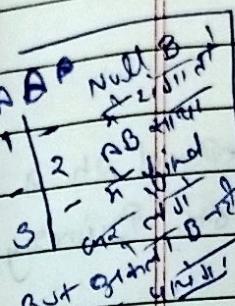
* 2NF : → Knowledge exists by partitioning

* Relation must be in 2NF.
* Relation is in 2NF if it
has no Partial Dependency.



Candidate Key → Combination of multiple columns
 Partial Functionality (P.F.F)
 A non Prime Attribute (C) instead of depending entire Candidate Key is defined on part of candidate key.
 $R_1(A, B, C)$
 $R_2(B, C)$
 * Table should be in 1NF
 * No Partial dependency.

Dependency ≈ functional dependency ≈ all columns dependent on one primary key.


 When there are two / more Primary Key in one table (Candidate Key).
 An attribute defined only on a part of Candidate Key.

Student			Subject		
A	B	C	P	B	Marks
a	1	y	a	1	
b	2	y	b	2	
a	3	z	c	3	
c	3	z	d	3	
d	3	z	e	3	
e	3	z			

Candidate Key.

Stu. ID	Subject	Marks	Teacher
1	Maths	1	Mr. A
2	Science	2	Mr. B
3	English	3	Mr. C

Primary Key → Prime Attribute
 Depends on Primary Key → non Prime Attribute

Page No. _____

Date _____

3NF

- table should be in 2NF.
- No transitive dependency

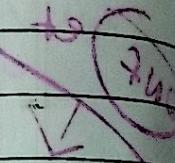
Transitive Dependency :-

When one column depend on a column with which it is not Primary Key.

$$A \rightarrow B, B \rightarrow C$$

$$\boxed{A \rightarrow C}$$

non Prime → non Prime.



[Non Primary Key can take null value - X]

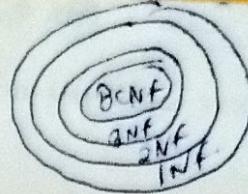
Score :-

Score Id	Stud_Id	Sub_Id	marked	exam_name	total_mark
101	201	1	16	mid	20
102	201	2	69	final.	80
103	202.	1	19	mid	20

Score -

Score Id	Stud_Id	Sub_Id	marked	exam

+



BCNF — (Boyce Code Normal form)

- * Advanced Version of 3NF.
 - * table should be in 3NF.
 - * Left hand side should be Candidate Key / Super key.

If a table is in 2nd Normal form then
It doesn't mean it is in 3rd NF If
Can be / can be not.

Roll No	Name	Voter Id	Age
1	Ravi	K0123	20
2	Varmi	M034	21
3	Ravi	K786	23
4	Ramya	D286	21

(RollNo , VoterId = C.K]

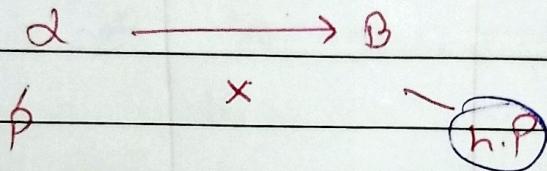
Prime attribute — Part of Candidate Key.
Non Prime attribute — Not a Part of Candidate Key.

Page No.: _____

Date: ____ / ____ / ____

~~2NF~~

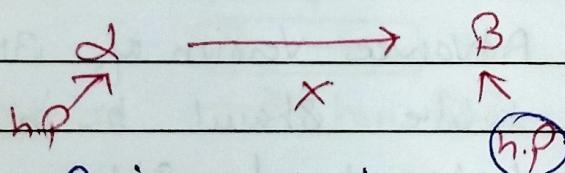
Partial dependency —



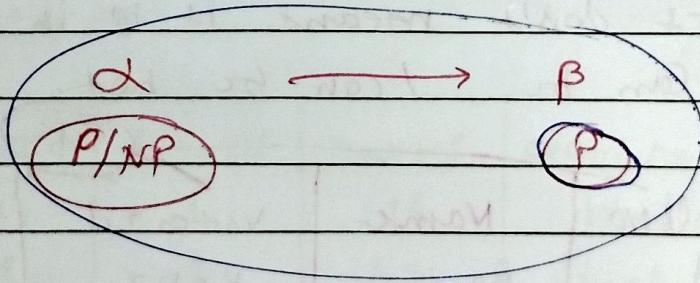
non Prime attribute depend on entire Candidate Key instead of it depend Part of Candidate Key.

~~3NF~~

Transitive Dependency —



When non Prime attribute β depend on non Prime α .



\Rightarrow BCNF Deal
with this kind
of case.

Dehydration trend for g block.

$\text{P}(\text{A}, \text{B})$

f.

Not \rightarrow condition
not

$\text{AB} \rightarrow \text{C}$
 $\text{C} \rightarrow \text{D}$

$\text{②} \rightarrow \text{E}$
Jutting out with

P(A, B)

$\text{AB} \rightarrow \text{C}$

$\text{C} \rightarrow \text{D}$

A	B	C
a	1	x
b	2	3
c	2	2
d	3	2
e	3	1
f	3	1

A	C		C	B
a	n		n	1
b	y		y	2
c	z		z	2
d	w		w	3
e	w		w	3
f	w		w	1

No in 3 conf.

1 st NF	2 nd NF	3 rd NF	BCNF
* No Multivalued Attrib.	> in 1 st NF.	* in 2 nd NF	* in 3 rd NF
* only Single value.	Dependency.	Dependancy.	L.H.S must be CR/SK
	* only full Dependency.	* No Non Prime Should be determine non Prime.	
	(R)	C	

Computer Science.

Student ID.	Department ID.	Student Name
1	1	Ajed
3	1	Sam

Chemistry.

Student ID	Department ID.	Student Name
2	2	Marie

Botany.

Student ID	Department ID	Student Name
4	3	Sara

4 th NF	5 th NF
* in BCNF.	* in 4NF.
* NO Multivalued Dependency.	* Loopless Decomposition.
$x \rightarrow y$.	

④ Denormalization → Denormalization is
used on Prevalley
Normalized database to increase
the Performance of database.

Student ID	Department ID	Student Name
1	1	Alex
2	2	Marie
3	1	Sam
4	3	Sara

Keys — it used to Uniquely Identify an
→ tuple (row) / recd. of data
from the tab. tab.

Types of Key — 8 types of
Key.

- ① Super Key. (default Key).
- ② Candidate Key
- ③ Primary Key.
- ④ Foreign Key.
- ⑤ Composite & Compound Key.
- ⑥ Alternate Key.
- ⑦ Surrogate Key.

⑧ Suffix Key —

Any Set of Attributes that Uniquely
Identifies each tuple of a
relation.

Eg. — Student table : —

St Id	Reg. Id	Name	Branch	Email
1	CS 2019-37	John.	CS	John@xyz.com
2	CS 2018-03	Adam	CS	adam.coole@xyz.com
3	IT 2019-01	Adam.	IT	adamhardy@xyz.com
4	ECE - 2019-07	Ellie	ECE	ellie@xyz.com

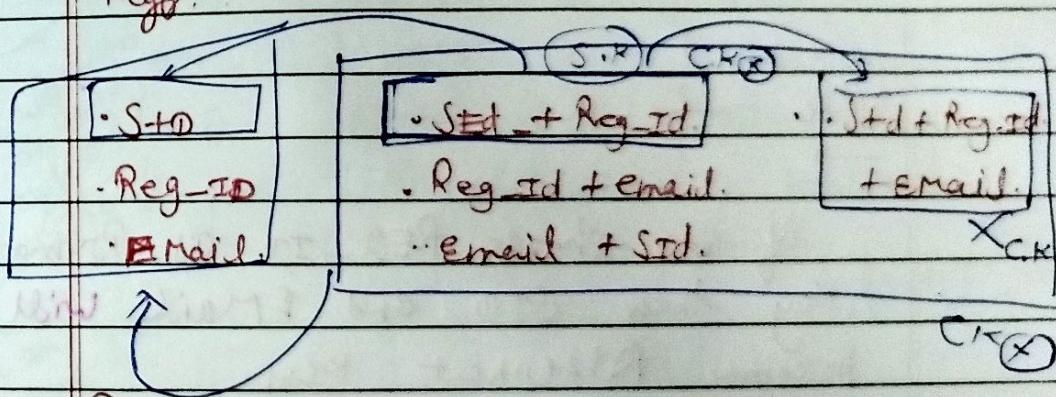
Key: —

- SID
- REG-ID
- Email.
- SID + REG-ID
- REG-ID + Email
- Email + SID
- SID + REG-ID + Email

(*) Candidate Key —

- Sub-set of Super Key.
- A Column / Combination of Columns that Uniquely identify each row in table.

Key: —



(*) Primary Key —

- A Primary Key is a Candidate Key that Uniquely Identify each row of data in table.

- Primary Key Value Can't be Null.

Student Table

SID	REG-ID	NAME	BRANCH	EMAIL
1	CS-2019-37	John	CS	john@gmail.com
2	CS-2018-03	Adam	CS	adam@gmail.com
3	IT-2019-01	Adam.	IT	adambhardwaj@gmail.com
4	ECE-2019-07	Emily	ECE	emily@gmail.com

Candidate Key —

• SID
• REG-ID
• EMAIL

PICK ANY ONE AS
Primary Key.

Note —



if we choose REG-ID as Primary
Key then SID and Email will
become Alternative Key.

Alternative Key ↑
= x =

② Foreign Key —

- ⊗ It is an Attribute in one table which is used to define Relationship with another table.
- ⊗ Help to Maintain Data integrity.

→ e.g. Branch Table —

Branch Code	Branch Name	HOD	
CS	Computer Science	John Rayn	...
ME	Mechanical Engineering	Adam Lee	...
IT		L. Subramanyam	...
ECE	Electronics And.	L. Subrat	...
	Communication	Wind 800	...

if we want to
update an entry in
Branch table
which is defined in
Student table.

Student Table —

SID	REG-ID	NAME	Branch Code	Email
1	CJ 2019-37	John	CS	John@xyz.com
2	CJ 2018-02	Adam	CS	adamcool@xyz.com
3	IT 2019-01	Adam.	IT	adamlord@xyz.com
4	ECE 2019-07	Elly.	ECE	elly@xyz.com
5	EIC 2019-12	Shelly	EIC	Shelly@xyz.com

New entry NOT allowed by DB if no entry in Branch table for "EIC".

(2)

Composite Key

- Any key with more than one attribute is called Composite Key.

Ex :-

→ (Sid, REG-ID), (REG-ID, Email)
 (Sid, ~~Reg-ID~~, Email) all are
 Composite Key.

- All the super key with more than one attribute is also called Composite Key.

(3)

Compound Key :-

- If a Composite Key has at least one attribute which is a foreign key then it is called Compound Key.

Ex :-

In above example if we have a compound key (REG-ID, Branch City) then it will be known as a Compound Key because Branch Attribute is a foreign key.

SURrogate Key

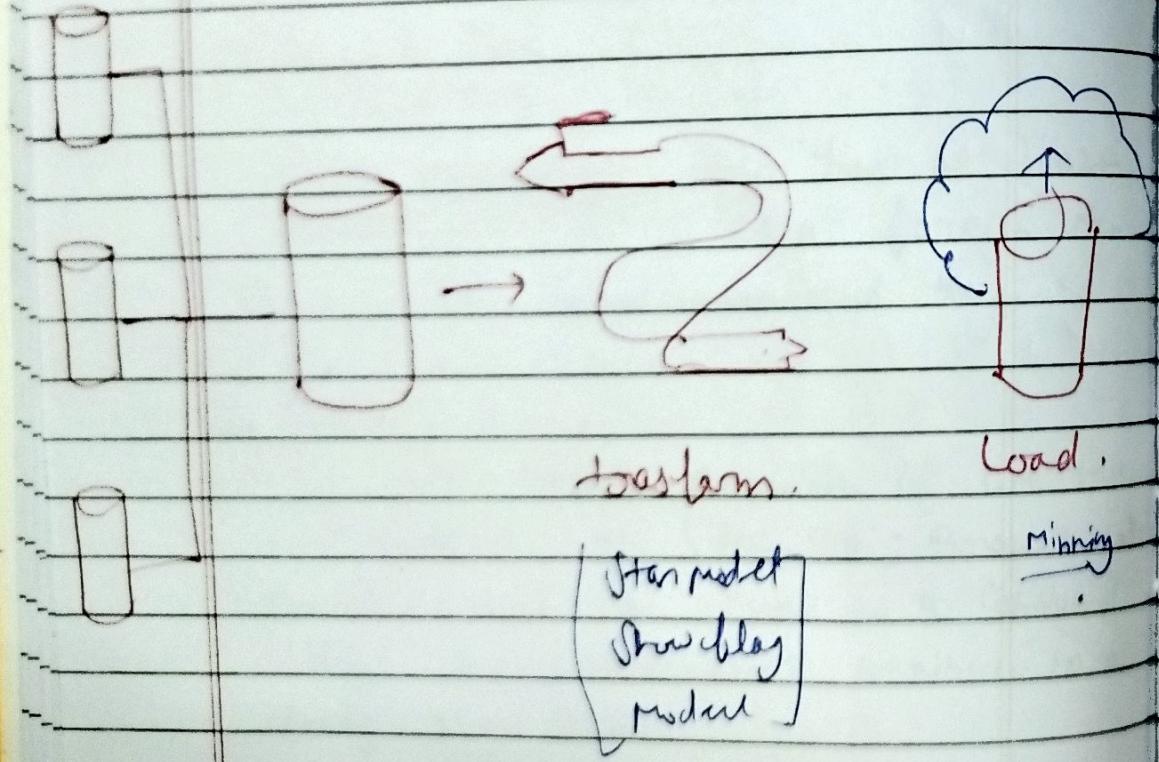
If a relation has no attribute which can be used to identify the data stored in it, then we create attribute for this purpose.

It add no meaning to data but serves the sole purpose of Identifying the uniqueness in a table.

④ ETL (Extract, Transform, Load) - Data Warehousing

- * It is a process used in data integration
- to extract data from various sources
- transform the data to meet specific business requirements
- load the transformed data into a target system such as data warehouse or a database.

Tool - [SSIS]



Constraints are the business rules which allow us to maintain proper data integrity.

| Set of Rule |

x

Constrains

* Prohibition / Limitations on data in database

- * Inherent Model-based -
- * Schema based -
- * Application based -

Constraints can be specified when the table is created with the Create table statement, or after the table is created with Alter table statement.

Syntax :-

>Create Table table name (

 Column datatype Constraint,

 " 2 " "

 " 3 " "

 - - - - -

) ;

Column Level — apply to be a column.
Table Level — apply to be a whole table.

Commands used in SQL —

- * NOT NULL — Can't have Null value.
- * Unique — All values in column are diff.
- * Primary Key — Combination of not null / unique.
- * foreign key —
- * Check — Used to limit the value range.
- * Default —

~~ANSWER~~

PRACTICE

Time

- * Create Table :-
- * Alter Table :-

Ex - Not Null -

Create Table Persons (

Id int NOT NULL, Unique, Primary
Bigint Name VarChar(255) NOT NULL,
Age int,

J;

Alter Table :-

Alter Table Persons

Alter Column Age int NOT NULL;

~~ALTER~~ modify Column Age int NOT NULL;

Note:- if we want Alter Table to Primary Key,
Column must have been declared to
not contain null value

Drop a Primary Key :-

Alter TABLE Persons.

Drop Primary Key;

~~for foreign key~~ -

- * used as Primary column that
cannot store null or blank.
- * it is a field in one table that refers
to the Primary key in another table.

Create table Orders (

Orderid int not null, Primarykey
Addressid int not null,

Personid int,

~~Personid~~ foreign key

foreign key (Personid) References Persons
(PersonId).

Alters -

→ Alter table Orders

ADD foreign key (Personid) References
Persons (PersonId);

Drop a foreign key -

Alters table Orders.

Drop foreign key for Personarks;

Check Constraints

* Used to limit the value range that can be placed in column.

g - Create table Person

Id int NOT NULL,

lastName varchar(255) NOT NULL,

firstName varchar(255),

Age int

check (Age >= 18)

j:

Check constraint on the "Age" column when the "Persons" table is created

↳

To ensure that age of a Person must be 18 or older.

SQL Check on Alter table

+ - x ---

ALTER TABLE Person

ADD CHECK (Age >= 18);

Add table already created.

Drop a check constraint

ALTER table Persons

Drop constraint ~~check~~ PerAge

② SQL Default Constraint -

Used to set of Default value for column
 Create Table Person
 Id int NOT NULL,
 LastName varchar(255) NOT NULL,
 FirstName varchar(255),
 Age int,
 City varchar(255) DEFAULT "Sandwich";

;

Default on Alter Table -

Alters Table Person;

ALTER City SET DEFAULT "Sandwich";

Drop or Dropout constraint.

Alters Table Person

ALTER City DROP Default;

③

Auto-increment — Unique num to be generated automatically.

CREATE TABLE Person (

PersonId int NOT NULL Auto_Increment,
 LastName varchar(255) NOT NULL,
 Age int

);

an Alter —

Alters Table Person Auto_Increment —

India Potash -

Diff b/w Drop / Truncate / Delete

Delete

DML

(Data manipulation L.)

Drop

DDL

Data Definition Language

Truncate

DDL

Data Definition
Language

Delete from table name

Drop from table

Student

Truncate Student

facts

Sometimes

Rollback

Not Rollback

Not Rollback

Student :-

Student

Student

ID	name
1	A
2	B
3	C

ID	name
1	A
2	B
3	C

ID	name
1	A
2	B
3	C

Structure Audit

Structure / Row

Structure
Audit

DSC student

DSG student

ID	name
1	A
2	B
3	C

ID	name
1	A
2	B
3	C

ID	Student
1	A
2	B
3	C

EN DSN where you
use not in,
Delete from student where
ID = 1,

EN DSN where
use exists in

Join

Combination of rows b/w two

Columns

inner JOIN (A, B).

Inner Join

1	A ₁
2	A ₂
3	A ₃

1	B ₁
2	B ₂
3	B ₃

1	A ₁	B ₁
2	A ₂	B ₂

Syntax :-

Select column-name(s)

from TableA

add E+

Inner JOIN TableB

On TableA.col-name = TableB.col-name

Left

Join — Return all record from the left table, and the matched record for the right table.

1	A ₁	B ₁
2	A ₂	B ₂
3	A ₃	B ₃

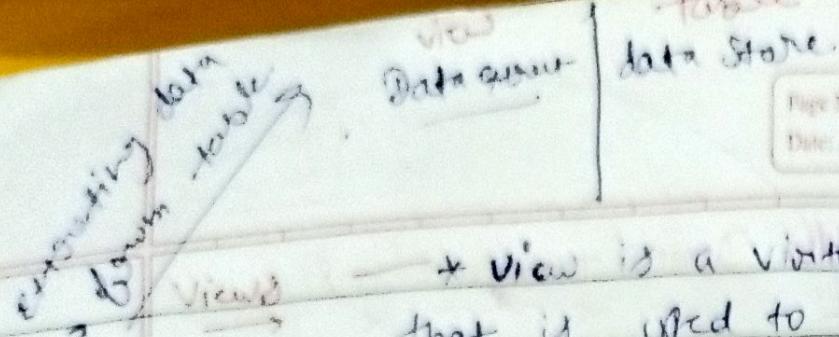
1	A ₁	B ₁
2	A ₂	B ₂
3	A ₃	

Syntax :-

Select column-name(s)
from TableA

Left Join TableB

On TableA.col-name = TableB.col-name



+ View is a virtual table.
that is used to manipulate
rows of the tables.

Use of View —

Table —

Dept	ename	Salary
HR	A	1.2K
Admin	B	10K
Fin	C	15K.
.	:	:

* Read only view
* Updatable view.

Default view —

— x

row view VR:

View will be created at original table + it will be visible.

V_HR → HR

V_Admin → Admin

V_Fin → Fin

Syntax — Create view view-name AS

Select Column1, Column2, ...

from table-name

Where Condition;

①

Create view V_HR AS

Select Dept, ename, Salary from table

Where Dept = "HR";

②

Create view V_Admin AS

Select Dept, ename, Salary from table

Where Dept = "Admin";

①

Join — Combination of rows b/w two
Tables.

Inner Join —

inner_join(x, y).

1	x_1	1	y_1
2	x_2	2	y_2
3	x_3	4	y_3

1	x_1	y_1
2	x_2	y_2

Syntax :-

Select Column name(s)

from TableA

Inner JOIN TableB

On TableA.Columnname = TableB.Columnname

Left Join :- Return all records from the left
table, and the matched record from
the right table.

1	x_1	1	y_1
2	x_2	2	y_2
3	x_3	4	y_3

1	x_1	y_1
2	x_2	y_2
3	x_3	

Syntax :-

Select Column name(s)

from TableA

Left Join TableB

On TableA.Columnname = TableB.Columnname

Right Join — Return all the records from the right table and match record from the left table.

1	x1	1	y1
2	x2	2	y2
3	x3	3	y3
4	x4	4	y4

1	x1	1	y1
2	x2	2	y2
3	x3	3	y3

Left Join —

Select Column-name

from TABLEA

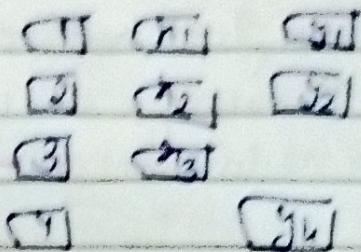
Right JOIN TABLEB.

Or TABLE A . Col-name = TABLE B . Col-name

Full Join —

Select all the records

1	x1	1	y1
2	x2	2	y2
3	x3	3	y3



Database Transaction

It is set of operations used to form a logical unit of work.

Transaction generally represent change in database.

Operation → Write / Read.

~~control transaction~~
~~of database~~ Transaction Control — maintains
 database

① Auto Commit.

② Commit.

③ Rollback.

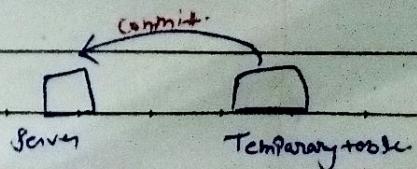
④ (Savepoint)

Any operation perform on a table using DML.

⑤ Insert ⑥ Update ⑦ delete.

~~changes static in server.~~
 COMMIT Command : — used to save the
 changes made to the
 table permanently.

Syntax : → COMMIT;



1st Step Done

Rollback

→ get back to the Previous

Permanent Status of the table

Similar to undo Sqlct → ROLLBACK.

SavePoint

→ use along with rollback command

SAVEPOINT SavePoint-name;

SavePoint is a Point in the transaction where transaction can be rolled back ~~with~~ without closing transaction.

SavePoint Rollback at ~~2nd~~ ~~3rd~~ step entry

SavePoint SavePoint-name;

roll back to SavePoint-name;

(5)

(5)

SavePoint 1 → 5

SavePoint 2 → 10

begin

commit

Backup / Restore

acknow.
TC

Holbein
4.7.03

to inner query } }

Subquery — when we write
query under query then
called Subquery.

exists —

~~Select column name~~

~~from table names~~

~~Where column name operator.~~

~~(Select column name from table)~~

Select column name

from table

where ~~comparison operator.~~

~~(Select column name from table)~~

Exists → return true.

NOT EXISTS → false

Sharding / Partitioning the method of distribution of data.
→ to maintain the traffic loads

Page No. _____
Date _____ / _____ / _____

Database Partitioning / Sharding : —

Sharding — Method of distributing data across multiple Machine.

Partitioning — Splitting a subset of data within the same instance

Where Single table divide into smaller segments.

Note Sharding / Partitioning are techniques to divide and scale large databases.

Sharding — distribute data across multiple servers. While Partitioning splits table within one server.

- UFC : — → improve Performance
* Manageability
* availability of large database

Scaling : —

Recovery in time 42 Check Point times
at same cost it help and 8.

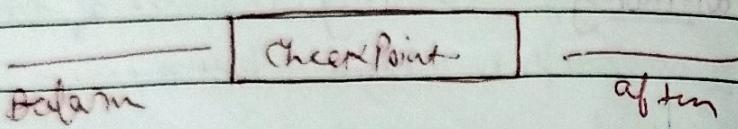
Page No. _____

Date _____

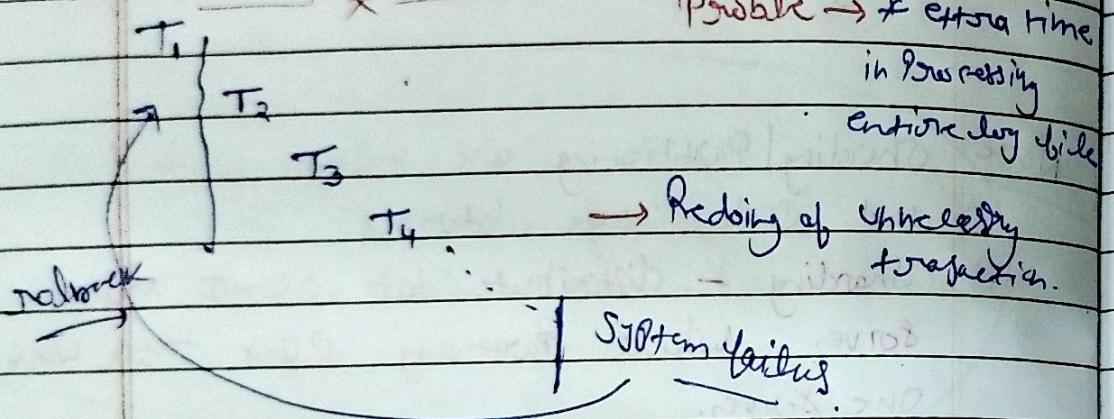
⑤

Check Points

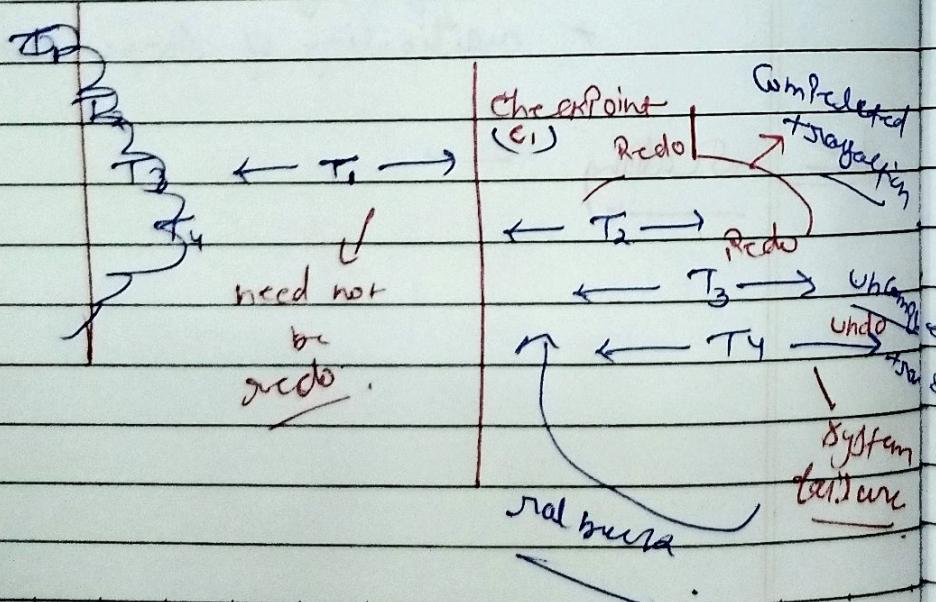
All the Previous logs
are remove from the system
and stored permanently in storage disk.



Without Checkpoints transaction



With Checkpoints transaction



- GET an event trigger & it triggers automatically execute \rightarrow GET trigger.

Page No. _____
Date: ___/___/___

Triggers in SQL

Triggers are Stored Program, which are automatically executed / fired when some event occurs.

Event Can be any of the following —

Database manipulation (DML) — Delete / Insert / Update.
(DDL) — Create / Alter / Drop.

Database operation — Logon, Logout, Start up,
Shut down.

Trigger can be define on the tables,
view, schema, database.

Benefits

- Child table \rightarrow can update
 \rightarrow at Parent table \rightarrow can update
- * Security.
- * Referential integrity.
- * Auditing.
- * event login and storing information
on table audit.
- * Synchronous replication of table.
- * Preventive invalid transaction.

ted
actions.

Source of Trigger

CREATE [OR REPLACE] ~~TRIGGER~~ trigger_name
[BEFORE | AFTER]
[INSERT | UPDATE | DELETE]
ON table_name
[FOR EACH ROW]

DECLARE :-

Declaration - Statements

BEGIN

Executable - Statements

END;

↓ (try again)

Page No. _____
Date _____

Example of triggers —

- Create trigger salary_difference
- before insert or delete or update on emp
- for each row
- declare
- Salary_difference number;
- begin
- Salary_difference := new.salary - old.salary;
- dbms_output.put_line ('Old salary : ' || old.salary);
- dbms_output.put_line ('New salary : ' || new.salary);
- dbms_output.put_line ('Salary difference : ' ||
 Salary_difference);
- end;

(Implicitly) → Explicitly

Triggers / Stored Procedures :-

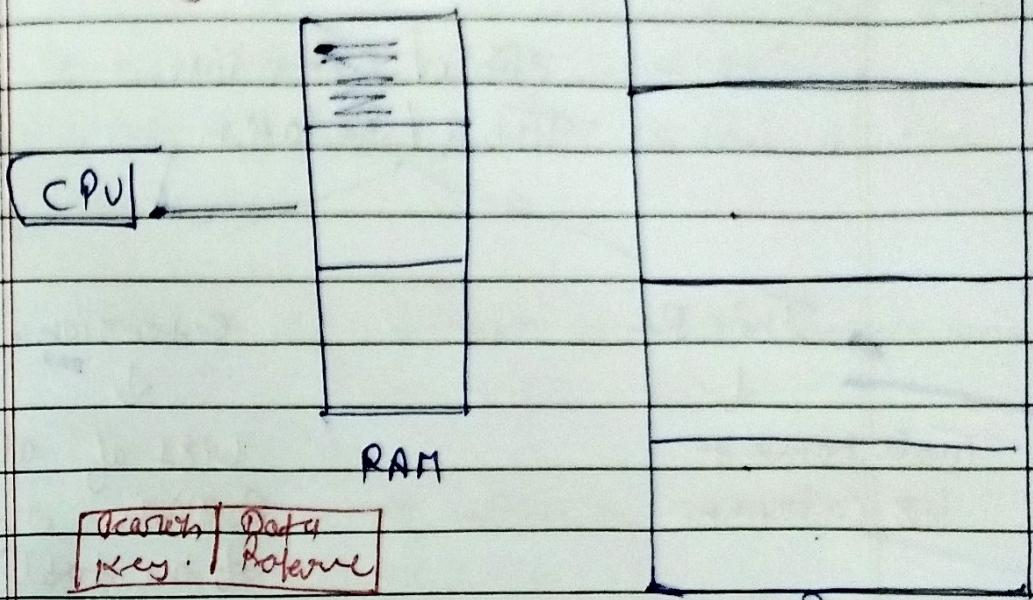
- ⑤ Stored Procedure are explicitly invoked by the program while triggers are implicitly invoked by database.
- ⑥ Stored Procedure execute independently of any table / view, while trigger associate with specific table or view.

March 10
Cost at Retailer 012.47

Page No. _____
Date / /

Indexing

Optimize the
Search time



Search time can be reduced by Indexing.
Indexing is a technique used to speed up the search process. It involves creating an index file that maps search keys to their corresponding data locations. This allows the database system to quickly locate the data without having to search through every record in the main file. Indexing can significantly reduce search times, especially for large databases.

Index is a data structure technique that allows you to retrieve data from database.

Types of indexing :-

Orderd

Clustering

Primary

Secondary

Dense

Sparse

Table
Schema

Page No. _____

Date _____

Intension | Extension

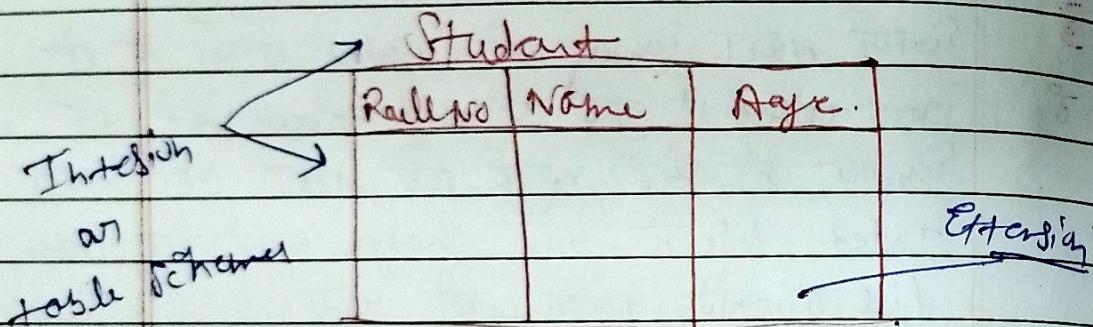
Table extension.
Table / Relation

Intension

extension.

Table name +
its column.

List of all
possible rows
of the table.



Oracle with memory area ~~QUTT~~ & JET
area out Context area ~~QUTT~~ & JET
Context area and Pointer ~~QUTT~~ & JET
and ~~QUTT~~ Cursor

Cursor — ~~cursor~~

Create a memory area,
Known as Context area, for processing an
SQL Statement, which contains all the
information

Cursor is a pointer to this Context area.

Types —

* implicit.

* explicit.

Implicit Cursors — Automatically Created.
— by Oracle itself
Explicit Cursor \rightarrow if $c1$

④ Programmes can't control implicit cursor and information it is.

⑤ Whatever DML Statement (INSERT, UPDATE,
DELETE) implicit cursor associate
with this statement.

Explicit cursor —

Programmer-defined cursor class giving more control over the output areas.

Select statement which returns more than one row.

Cursor Cursor-name Is Select Statement

Working with explicit cursor —

- ④ Declaring the cursor class initializing the memory.
- ⑤ Opening the cursor for allocating memory.
- ⑥ fetch the cursor for retrieve the data.
- ⑦ Close the cursor class deallocate the allocated memory.

912976
Date / /

Declaring the Cursor -

CURSOR C_Customers IS SELECT id, name,
address from Customers;

Opening the cursor -

OPEN C_Customers;

fetching the Cursor

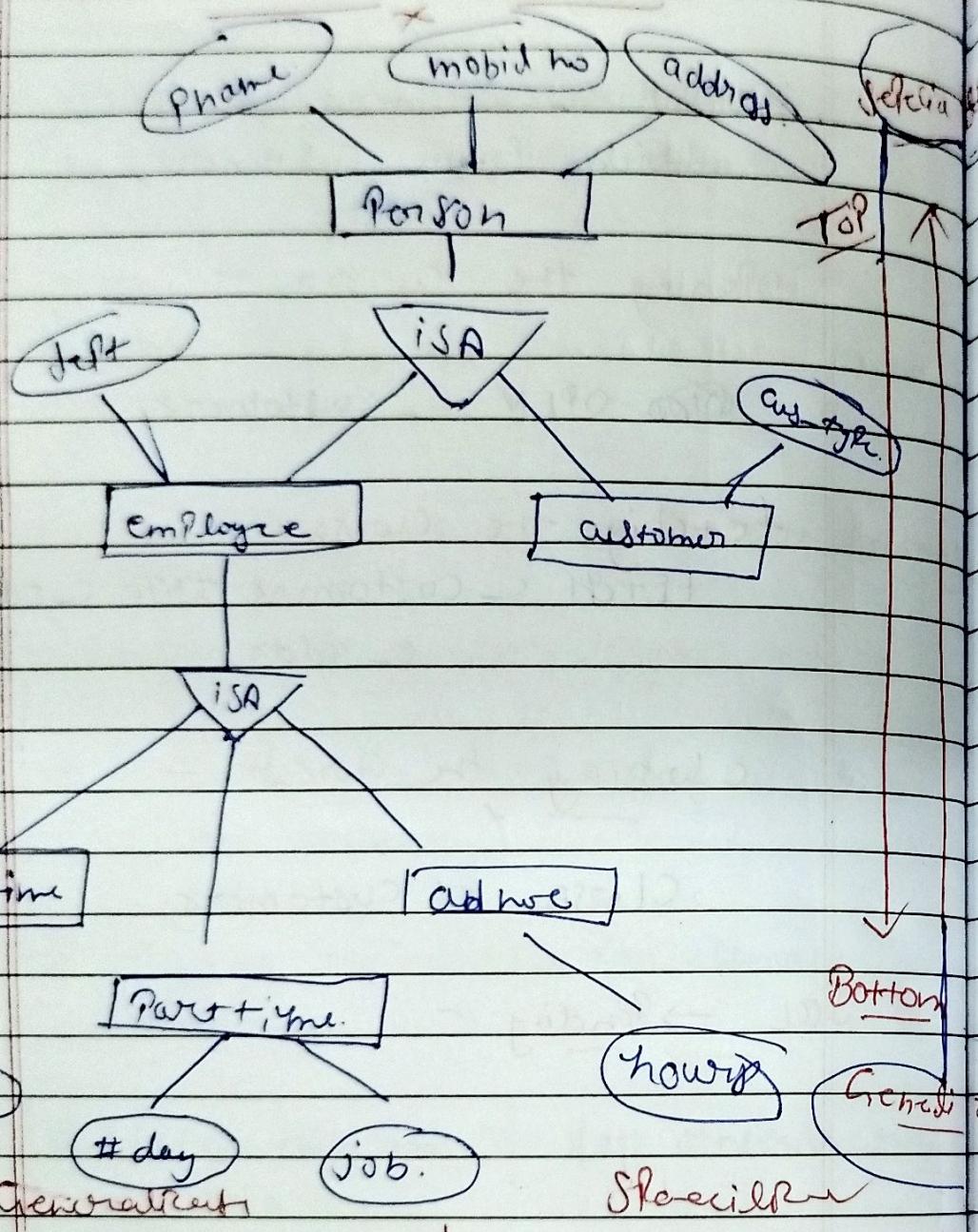
FETCH C_Customers INTO C_id, C_name,
C_addr;

closing the cursor -

CLOSE C_Customers;

SQL → Pending :-

Specialization / Generalization



- ① Size of Schema get order increase
- ② Afforded great of entity

increase
single entity.

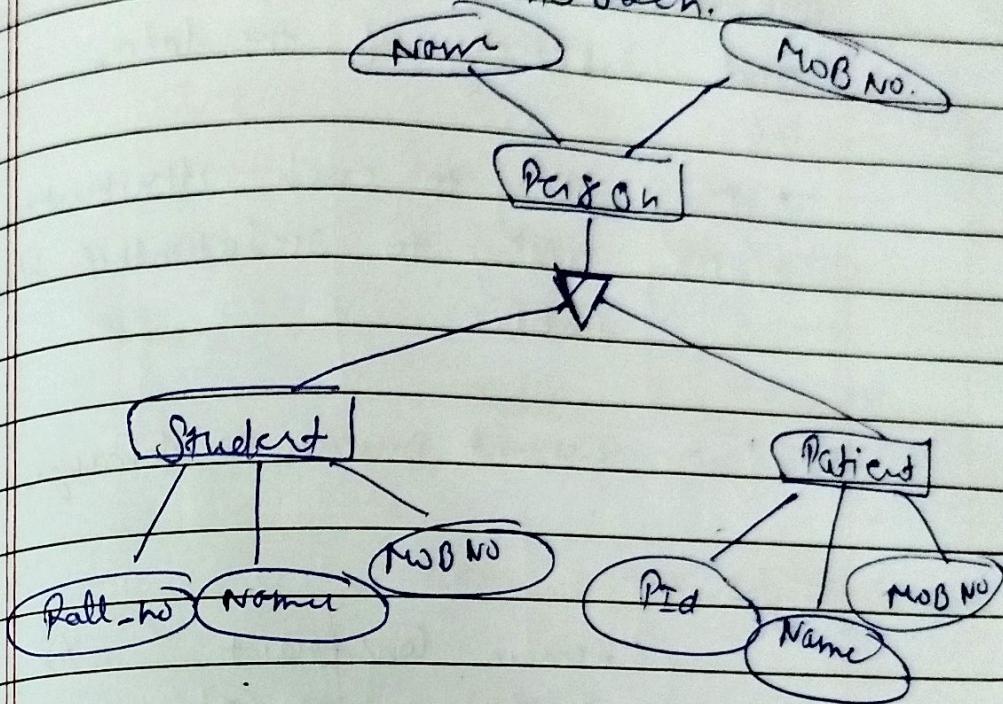
entity. Specialization involves combining them into a single entity broken down into multiple entity.

Page No.	1
Date	/ /

Generalization

→ Size of schema gets reduced

* Bottom up approach.

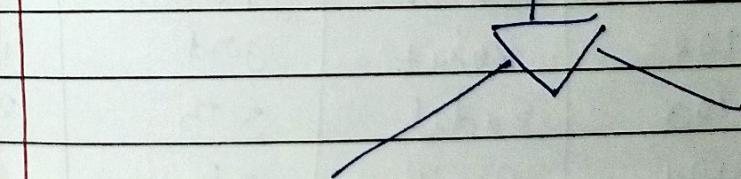


Specialization → Opposite of generalization

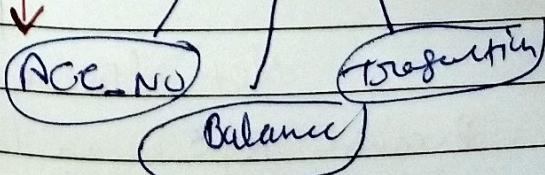
Particular entity divide into sub entity.

Size

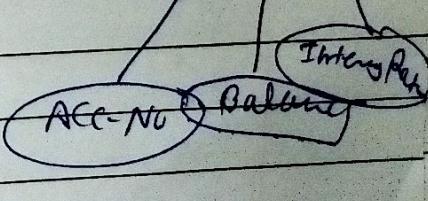
Specialization



Current-Acc



SAVING-Acc



accuracy

Page No. _____
Date. _____

* Integrity Rules / Constraints

Are set of rules that insure accuracy and consistency of the data.

- May apply to each attribute.
- May apply to relationship b/w two tables

Integrity Constraint Prevents to damage database

Type —

- + Domain Constraint A
- + Entity Integrity Constraint A
- + Referential integrity .. R-R
- + Key Constraint. A

* Domain Constraint — Define the Domain.
+ on valid set of values for an attribute.

Student Id	NAME	SEMESTER	AGE
101	Manish.	1st	18
102	Rohit	3rd	19
103	Badal	5th	20
104	Amit	7th.	A

Not allowed

Because Age is an integer
Values.

* Entity Integrity Constraint :-

- * Primary Key Value can't be null.

EMP_ID	EMP_NAME	SALARY
111	Rohan	20000
112	Rohan	30000
113	Sohan	35000
114	Logan	20000
	Amesh	50000

↓
Not allowed as Primary Key Can't contain Null Value.

X X X X X X

* Referential Integrity Constraint :-

- * Specified b/w two tables

- * foreign key references the Primary key of a table

The Rules : — * Null allowed in foreign key but not in Primary Key.

* Can't delete record from a Primary table.

if matching records exist in related tables.

Can't change Primary Key value in Primary table if that record has selected record.

Can't insert value in foreign key field of the related table that doesn't exist in the Primary Key of the Primary table.

EMP (Table 1) Related / Reference Table foreign key

Emp-ID	Emp-Name	Age	Dep-No
111	Mohan	21	1
112	Aohan	33	2
113	Sohan	27	3
114	Logan	25	5

Primary Key

DEP (Table 2): Primary / Referenced table.

Dep-No	Location
1	Mumbai
2	Delhi
3	Noida

Not allowed as Dep-No 5 is not defined as a Primary Key of table 2 and in table 1, Dep-No is a foreign key defined.

Key Constraint

* Value of Primary Key
must be unique.

* The value of Primary Key must not be Null.

Student_Id	Name	Semester	Age
101	A-	-1	- 12
102	B	-2	- 13
103	C	-3	- 20
102	D	- 4	- 11

↓
Not allow because all row must be unique

Sunday Period at 09:00 at date at
2021

Page No.

Date

Data Replication

- * Maintaining defined set of data in ~~marked~~ our location

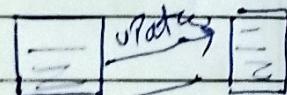
*

Different Replica / Backup : —

— → — X —

Replica — Update again / again.

Backup — One complete copy.



DB → MWh

Replica update at 8 → Tues.

8th ~~now~~ Backup week of 20th Oct at 8:22

in day end up at 8 at 19
update at 8,

→ Fri

* Asynchronous Replication

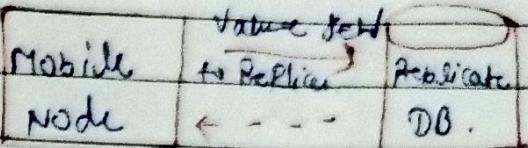
* Synchronous 11.

A Synchronous Replication

'MN' Sends Command to replicate data on generate database and doesn't wait for ACK.

Value sent to
Replica DB

Main DB



fast, wait at OJ Level.

succesful. ↗ Recd ↓ NO wait for ACK..

UPDATER.

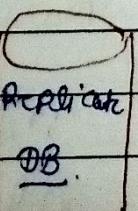
DISADV: High risk for data loss.

Synchronous Replication

MN Sends update request and waits for an ACK from database being used to replicate.

Main DB.

Value sent to
Replica DB



Adv - No data loss.

Long waiting time.

MN

Value sent to
Replica DB

Replicate DB

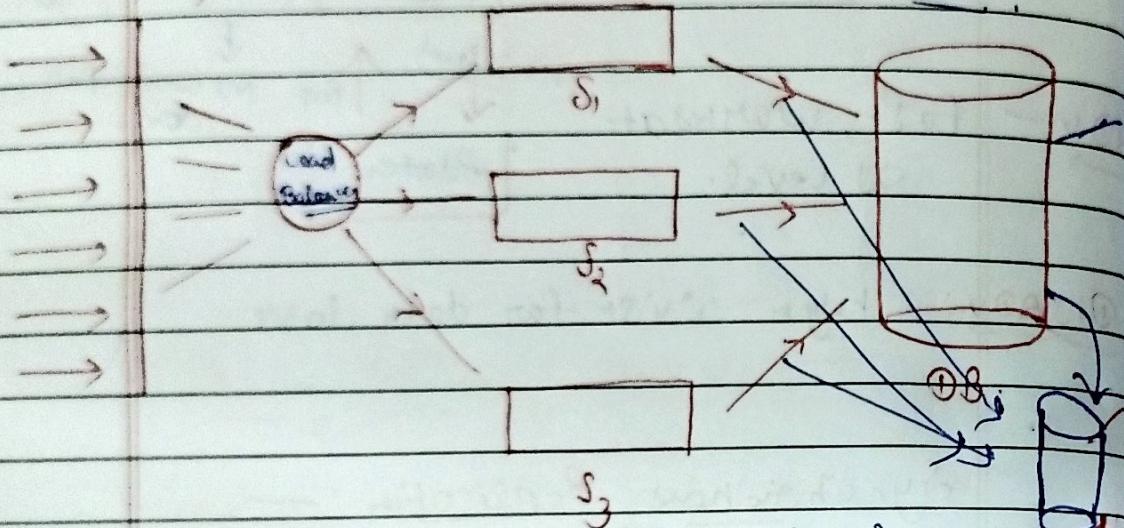
Wait for
an ACK

UPDATER.

Advantage of Replication

- ① Data Availability
- ② faster response time

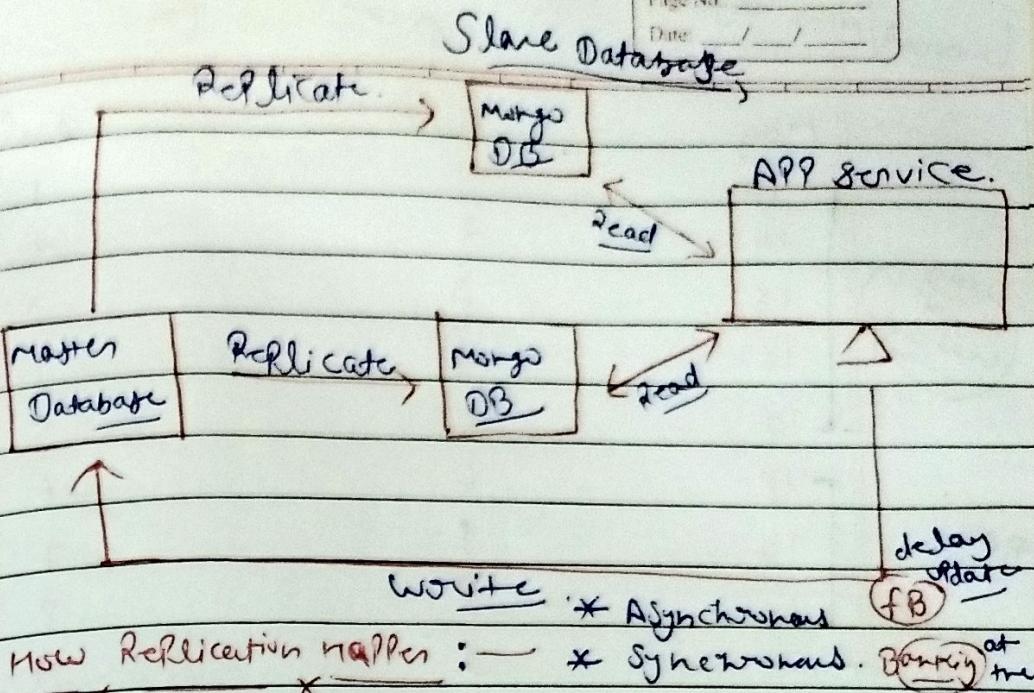
Master Slave Replication of DB — Single point of failure.



• यदि एक वर्तमान में दो डेटाबैस हैं जिनमें एक वर्तमान में दो डेटाबैस हैं जिनमें एक मास्टर है और अन्यत्री स्लेट है तो यह एक सिंगल पॉइंट ऑफ फैल्युर है।

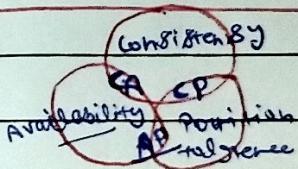
Master node (original db, latest DB, Primary DB)
 ↳ write operation.

Slave (Replica) — Read operation.



Advantage — **Backup.**

*** System fail et 42 read operation off Slave et ghot si ji load ari diji.**

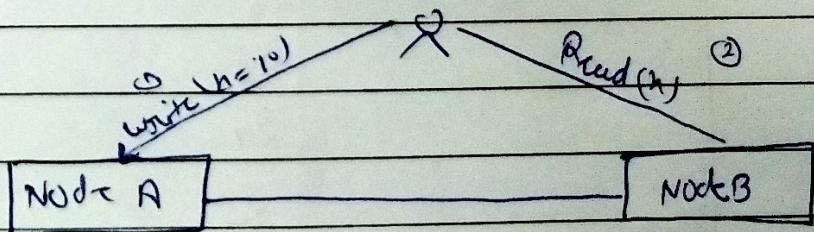


- * Consistency.**
- * Availability.**

*** CAP theorem**

— *** Partition Tolerance.**

3HTZ A node et value 10 et 8 et 29 et 10 et 11 et



3HTZ 2 operation et 41 seji,
Consistency et (Both request should proceed).
Availability et — if we fail one request.

(Job Apply)

Aug

Sep

Oct.

CV writing
PP interview
Aug 28

DSA → Kunal + List + Value

Python - Knowledge center
↳ List code
+ List :-

① Django List

= [Python] List - ①
 List - ②

Project → + Portfolio +
 resume + coverle
(English) then apply

DBMS / SQL / MongoDB

JS → List

[React - | Angular (view)] -

[NodeJS, Express - | NextJS.] -

Python (Django)

Architecture

Java (JPA)

(JPA) + Leetcode.

PHP Laravel