

DataBase Technologies .

DBT (Database technologies) .

MySQL

Oracle.

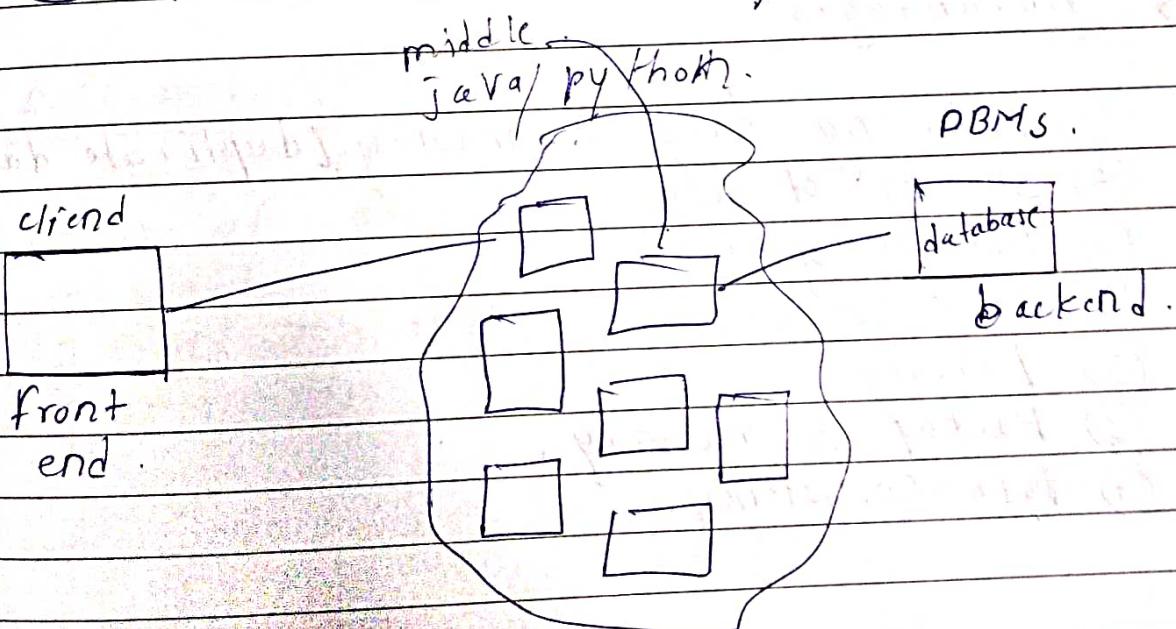
MongoDB.

- * Data → (raw information).
- info → (procured data).

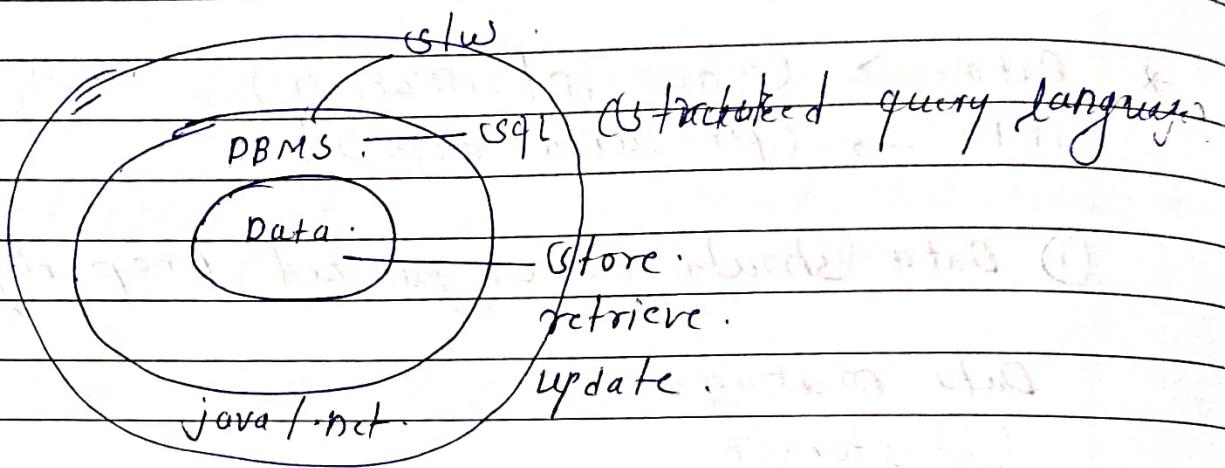
① Data Should be organized properly.

Data management .

- i Storage .
- ii Easy Retrieval .
- iii Updation should be easy (modify).



* what is MySQL → this is RDBMS (Relational database management system).



* query is nothing but a question.

* Advantages

- ① Reducing data redundancy (duplicate data).
- ② Sharing of data.
- ③ Data integrity.
- ④ Data Security.
- ⑤ Privacy.
- ⑥ Backup & recovery.
- ⑦ Data consistency.

Database Design

Types of Databases.

- 1) SQL Database . MySQL (Table).
- 2) NOSQL Database . (MongoDB , Casendra , HBASE)

Data stored in the form of JSON file format (key value pair).

3) Graph database.

e.g. Neo4j, GraphDB.

4) Memory Database . (RAM).

It is used where runtime data info is needed. e.g. satellite launching MemDB.

Data models

3 Types of data models.

way of data storage.

- Data

- Data relationship.

- constraints.

In SQL data is stored using table.

column / attribute / field

Name	Row
Ajay	1
Nijay	2

→ Row / Record

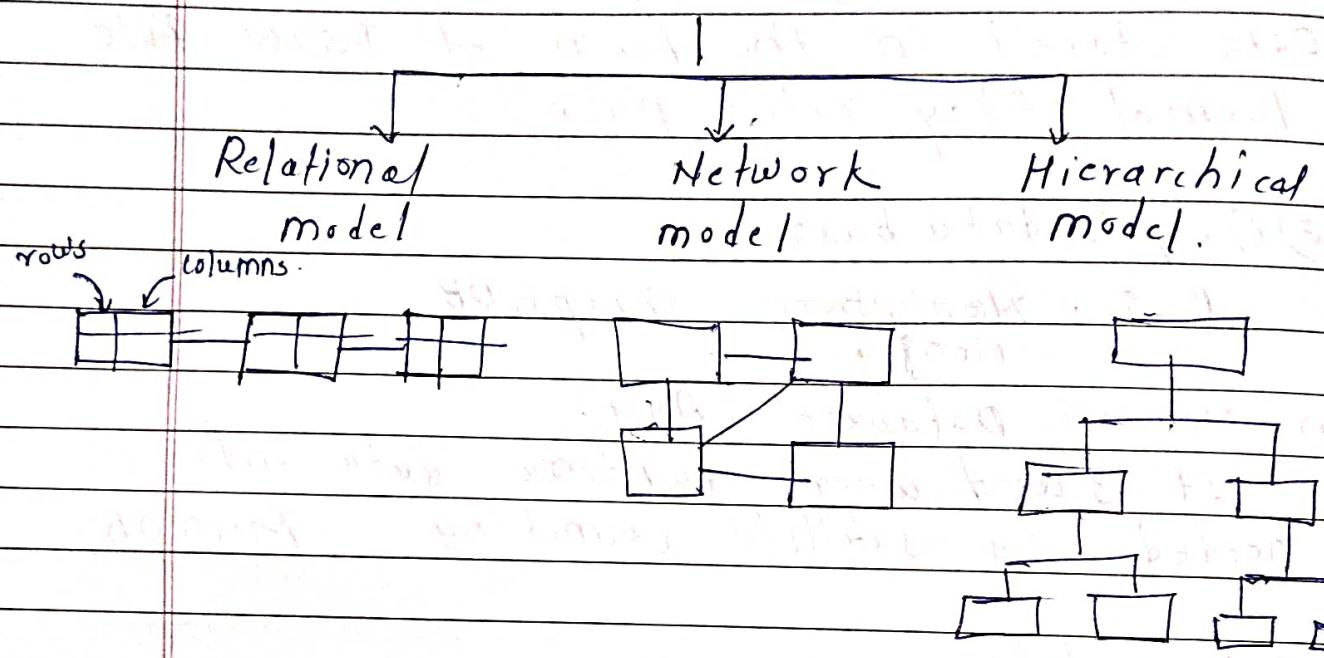
RDBMS

(Relational database management system)

* RDBMS → (structured tables)

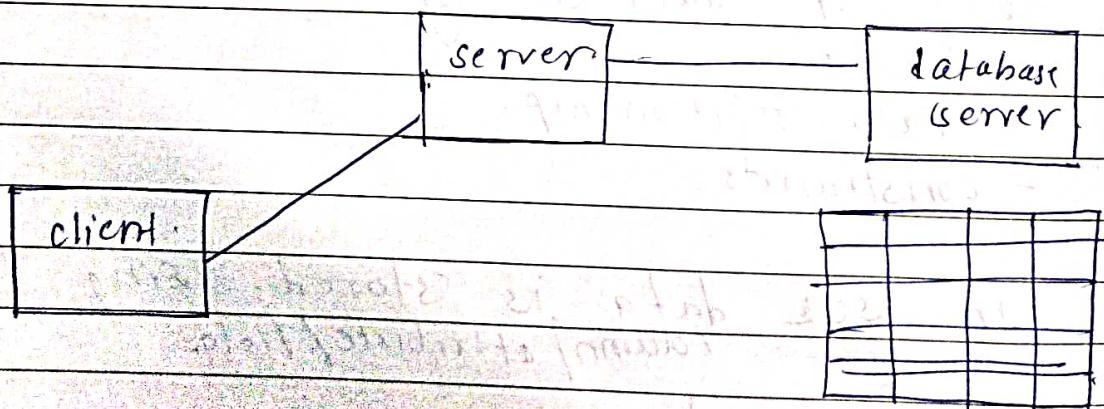
Sometimes data stored in one table is not sufficient so we need to create more tables & relation b/w the tables is made i.e. connection is established.

Data models.



* ACID properties

RDBMS .



RDBMS follows ACID properties.

- A: Atomicity
- C: Consistency
- I: Isolation
- D: Durability

~~VIMP~~
~~IQ~~

atomicity states that either one complete transaction or nothing i.e. all steps must be executed or nothing will be executed.

consistency states that data should be in the correct state before after transaction.

Isolation means unless user commits, others will not see the effect.

Durability means changes of successful transaction should be reflected even if system fails.

All SQL Database follows ACID properties except Cassandra ALP property.

* Key

Schema is logical info about data which you are going to put into table.

key - attribute in the table which uniquely identifies the row.

e.g.

id	name
1	Komal
2	Amruta
3	Ankita
4	Urmila
5	Amruta

* composite key -

Sometimes tables become so complex that one key will not fetch a row then com by combining ~~two~~ attributes we get the composite key.

* Primary key

primary key is a key in the table which is unique & it is not null. Which defines row uniquely.

it changes according to system requirement.

e.g. empno

name

mobile

Aadhar

Passport

(S)

empno
key

→ Aadhar

Passport

Primary

key.

alternate
key.

In above case empno can be possible primary key.

* Candidate key - (Set of all possible primary key.

e.g (empno, Aadhar, mobile, Passport)

* Composite key = Primary key + Alternate key.

→ Datatype

① varchar (default value of varchar - 255)
 └ store string value.

Name Varchar(15).

Name is variable length.

* command to create table.

create table table-name ()

(
 column1 datatype,
 column2 datatype,
 column3 datatype
) ;

create table student ()

(
 roll no int,
 name varchar(15),
 grade varchar(1)
) ;

rollno	name	grade

* command to add records-

insert into table-name.

values (value1, value2, value3);

or

insert into tablename (col1, col2...)
values (value1, value2).

e.g.

insert into student.

values (1, 'Raj', 'A');

rollno	name	grade
1	Raj	A
2	Sarika	B
3	Rohit	NULL

insert into student

values (2, 'Sarika', 'B')

insert into student (rollno, name)

values (3, 'Rohit');

insert into student (name, rollno).

values ('Rohit', 3);

In database whenever there is no value
System assign it as 'Null'.

* Select clause →

* command to display data from table.

Select * from student; table-name;

Select * ← all columns
from table-name;

Select name

from table-name;

→ only name displayed.

Select name, grade .

from table-name .

→ name & grade is displayed.

* command to create the database .

→ create database .

> create database knowitdb ;

> use knowitdb ;

You can create multiple database in MySQL. In oracle, it is not possible.

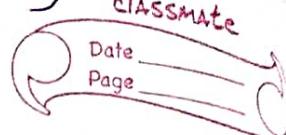
→ createdb .

1) & create database knowitdb

2) > use knowitdb .

3) crca

whatever commands we are giving
in MySQL is called Query.



- * If you want to see how many databases are there
 - `Show databases;`
 - `Show tables;` ← (shows tables)
 - `describe student;` ← (info in table)
- * If you want to access through diff databases.
`database name. table name;`
- * Arithmetic Expressions in Select Statement

Operators + , - , * , / and %

`Select eno, ename, sal + 1500 from emp;`

eno	ename	sal
1	Raj	30000
2	yash	40000
3	Greta	45000

after addition.

eno	ename	sal
1	Raj	31500
2	yash	41500
3	Greta	461500

→ only display.

null is not zero.
is not space.

classmate

Date _____
Page _____

- Null → unavailable.
- unassigned.
- unknown.
- inapplicable.

- * Any column in arithmetic expression is null then the result is null.
- * If you divide by null value will be null.

- * column heading To increase readability of table we are giving column heading!

Select eno as enumber
from emp;

((Select is only used for display it is not going to make changes in table)).

Select eno as number, ename as name,
sal as salary from emp;

- * If you have space in column heading then you need to give double quote.

Select eno, ename, sal * 12 as "Annual
Salary" from emp;

* Update clause

modifies current value in the table.

update table-name

(set col 1 = Value1, ,

col 2 = Value2, ,

col 3 = Value3, ,

;

;

;

[where condition] ;

Optional

Q. write a query to change sal. of Raj
to 40000

update emp or update emp.

(set sal = 40000) or set sal = 40000

where eno = 1 ;

where ename = 'Raj'

update emp .

(set sal = (40000 + 9000), commission = 2000,

where ename = 'Raj')

* Delete clause

Delete row from the table.

`delete from table-name;
[where condition];`

e.g. `delete from emp;` deletes specific
[where eno= 4]; row

`delete from emp;` deletes all rows.
(blank table will remain)

* SQ. what will happen if i will not give any
where condition then what will happen?

→ all rows will be deleted & blank table
remains.

* Truncate table

Delete all existing rows.
(same as `delete from emp`)

`truncate table - name;`
[where condition] is not there in
truncate command.

Ex. while giving column name
datatype is compulsory.

classmate

Date _____

Page _____

* To change the structure of table.

Alter table clause (structure of table)

add, delete, modify columns in existing table.

Alter command will work on structure of table.

1) add

Add new table column.

Alter table table-name

add column-name datatype.

e.g. Add new table column.

alter table emp add column.

add column commission int;

add column commission int;

2) delete.

alter table table-name

drop column-name; column-name;

e.g.

alter table emp

drop commission; column commission;

3) Modify (To change datatype of column)

alter table table-name.

modify column column-name datatype;

e.g.

alter table emp

modify column eno varchar(6);

* Delete entire table. (whole structure is deleted)

drop table table-name;

e.g.

drop table emp; ← whole table is deleted.

truncate

drop

(structure is preserved) → structure is deleted.

drop database database-name;

classmate
Date _____
Page _____

* constraints in the database (Rules)

- ① when row is inserted, updated or deleted.
- ② prevent deletion of dependent tables.

constraints

- ① Not null
- ② unique
- ③ check
- ④ default
- ⑤ primary key → table level constraint.
- ⑥ foreign key.

③ check

create table student

- ① Not null →
- ② create table student

(roll int unique not null,
name varchar(15),
age int check (age > = 18) ~~not null~~!) ;

roll int not null,

name varchar(15),

marks int

- e.g. insert into student
- values ('kavita', 98);

* kavita will not be inserted as we have given
constraint roll no. should not be null.

④ default

create table student

(roll int,
name varchar(15),
age int check (age > = 18),
marks int default 35

② Unique (prevent addition of duplicate values)

create table student

(roll int unique not null,

name varchar(15),

marks int

) ;

e.g. insert into student

values (1 , suraj) , 17);

↑
above data is not inserted.

⑤

Primary key

```

create table student
(
    id int not null unique,
    name varchar(12),
    primary key(id)
);

```

Every table will have only one primary key.

⑥

foreign key

If it is a key which is primary key in another table.

eno	cname	dept no.	key	dept no.	dept name
1	Ashok	10		10	Sales
2	Komal	20		20	Tech
3	Raj	40		30	Mkt.
4	Guru	30			
5	Yash	20			

* Select emp.eno, emp.name, emp.dno, dept.deptno, dept.dname from emp, dept
where emp.dno = dept.deptno;

Data is in scattered form which is stored in different tables. To connect the data from one table to another foreign key comes into picture.

- Between, like operators
- logical operators
- null

* Using where clause

* logical & or operators

Select * from table name
where condition;

eno	ename	sal	com.	dept	dname
1	Komal	2000	null	10	Tech.
2	Gnehq	3000	0.0	20	Sales.
3	Pratiksh	4000	200	30	HR
4	Barikq.	4500	100	20	Manager.

* In operator

Select ename
from emp
where dno = 20 or dno = 30 or dno = 10.

Between --- and operator
Select *
from emp
where sal between 3000 and 4500
>= & < = but it executes faster than.

(Select * from emp where sal > 2000 and comm. = 100.)

Select *
from emp
where sal > 2000 or comm. = 100.

Select ename
from emp
where dno in (20, 30, 10);

Q * display name & salary of employee
working in tech department.

Select *
from emp
where sal >= 3500 and sal <= 4500

Where dname = 'Tech';

Select database() from student
display working database.

tech or hr dept.

Display all rows, all employees who earn
do not get any commission.

Select ename, sal
from emp
where dname = 'Tech' or dname = 'HR'
or
~~where~~ where dname in ('Tech', 'HR');

- * Like operator
 - single character.
 - set of characters.

~~'R'~~ → Name starts with R character
(%, %) → Name ending with R character

Order by clause

Sorting the data.

Select *
from emp
order by sal;

'%' → one underscore indicates
one character.

entire table will be arranged acc-to
Sal (descending to default ascending)

Select *

from emp
order by sal desc;

Select ename
from emp
where ename like '%R%';

* copy one table to another table.

create table emp10
as

select * from emp;

copies employee table to emp10;

* create table emp11
as

select ename, sal, deptname from emp;

above command will copy required fields from employee table.

* create table emp12
as

select * from emp
where 1=2;

above command will copy only the structure because where 1=2

is condition becomes false & it will not copy any row.

condition can be anything 1=2, 2=3
3=4, 4=5 etc.
but 2=2 not works as this condition becomes true.

* types of SQL statements.

i) DDL

ii) DML

iii) TCS.

iv) Inbuilt functions in MySQL.

v) Single line functions.

vi) Aggregate functions.

vii) String functions.

viii) Length.

Select length(ename) from emp;

vii) lower

Select lower(ename) from emp;

viii) upper

Select upper(ename) from emp;

ix) ASCII

Select ASCII(ename) from emp;
it gives ascii value for the first character in string.

x) concat(str1,str2, --)

Select ename, job, concat(ename, ',', job) from emp;

Select lower (ename), know it @. com from emp;

or my substr(*str, m, n*).
start and posn.

Select substr(ename, 2, 3) from emp;

with concat('...', ...), we can do this.

Select eno "employee no", ename name, job designation, say "Net Salary", from emp;

Order by sequence to follow,

Select eno "emp. no", sal + ifnull(comm, 0)

from emp;

String functions.

* Date clause.

* Nested query → .

Display name of all employees working in dno 10, the department where smith works.

Select ename
from emp
where dno = 10;

Select dno
from emp
where ename = 'smith';

* ifnull(comm, 0). If commission is null then it will replace that null value by 0.

Select enam, sal + comm as "net salary"
from emp;

Select * from emp inner query

where dno = (Select dno from emp where ename = 'Smith'));

- o. Display all employees working in Research dept.

Select * from emp

where dno

= (Select dno

from dept

where dname = 'Research');

- o. Display all employee who earn salary more than Smith & Allen. Smith or Allen.

Select *
from emp

where salary > (Select sal

from emp

where ename in ('Smith', 'Allen'))

Select *
from emp

where sal > all (Select sal

from emp

where ename in ('Smith', 'Allen'))

- o. Display all employees who works in Smith's dept or Allens dept.

* Use of insert into (nested query).

```
insert into emp10  
(Select eno,ename,deptno,sal  
from emp  
where deptno=10  
);
```

* Group functions (Aggregate functions).

(Select max(sal), min(sal)
from emp;

* 3) sum()

finds sum of all the values.
(Select sum(sal).
from emp;

4) avg().

finds avg of all the values.
(Select avg(sal)

from emp;

5) count(*)

count (*)
finds total number of rows in table.

6) min()

(Select count(*)
from emp;

7) max()

finds max. value in given set.
(Select max(sal)

from emp;

8) variance.

1) max()
finds max. value in given set.

(Select count(*)
from emp
where sal > 1500;

2) min()
finds min. value in given set.

(Select min(sal)
from emp;

g) count (ename)

(Select count (ename)
from emp;

(Select count (lomm)
from emp;

This query is going to ignore null values.

(Select count (distinct deptno)
from emp;

Having clause (condition on group) by group

(Select ename, dno, sal
from emp

Group by deptno;
having count (*) > 2;

having clause is used when you want
to give condition on group

* where used on columns
having used on groups

* (Select job, count (*)
from emp
group by job;

* (Select deptno, sum (sal), count (*), avg
(sal) min (sal), max (sal)
from emp
group by deptno;

* all the functions sum, count etc will
work on groups created by deptno.

(Select deptno, job
from emp