

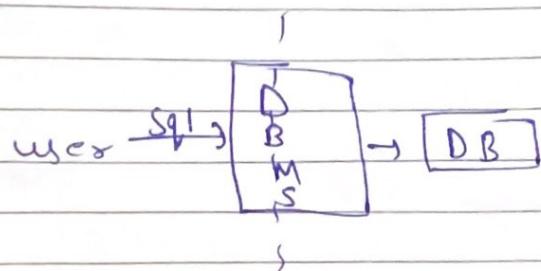
Date

Growth

SQL

Database - collection of data stored in a format which can be easily accessed is called database.

DBMS - Software application used to manage our DB. (Database).



types of DBMS

✓ (RDBMS)

Relational

↓

Data stored in
tables

non relational (NoSQL)

↓
Data is not stored in
tables.

Eg → MongoDB.

Eg → MySQL

oracle

Spiral

To marks
teacher's Siz

Date.....

SQL (Structured Query language)

→ used to interact with Relational Database management system.

→ used to perform CRUD operations

C create

R read

U update

D delete.

→ SEQUEL

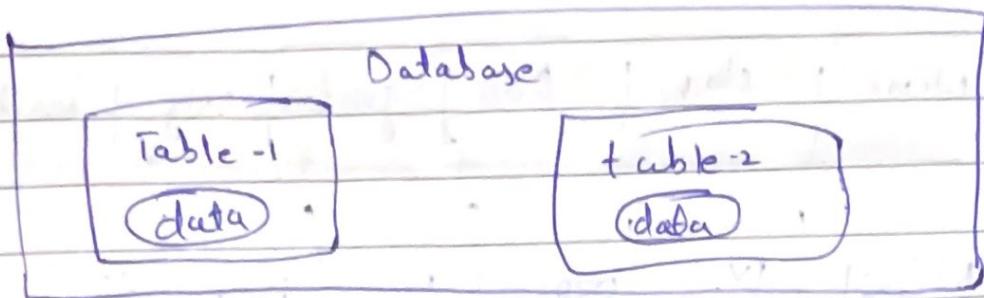
b.

Structured
English
Query
language

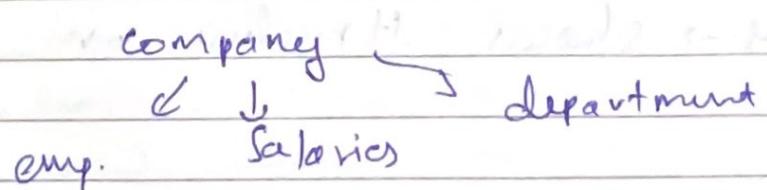
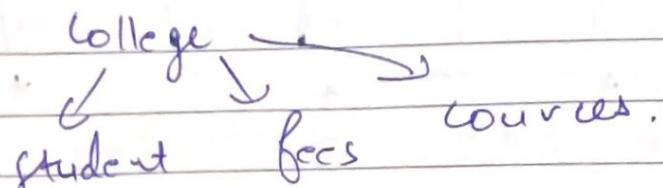
SQL

created by IBM

Database Structure.



interrelated tables database ke ander hoti
hai



vertical direction → col

horizontal direction → rows.

group of column & rows is called tables.

Date.....

Eg. of table.

Student table.

#	Poll no	Name	class	DOB	gender	city	marks
1		Nihil	IX	03/09/03	m	Delhi	93
2		Shikhar	X	-	m	Fatehpur	03
3		Rushayra	XI	-	m	Bihar	09
4		Nimish	XII	-	m	Delhi	57

columns → shows structure or schema.
(design)

values → individual data.

SQL

Spiral

Teacher's Sign

SQL is not a case sensitive language
Date.....

1st command.

1) Create database db-name;

2) DROP DATABASE db-name; # to delete DB.

3) USE college; # will perform all query inside college.

CREATING FIRST TABLE.

use college;

create table table-name (

column-name datatype constraint,

column-name datatype constraint,

column-name datatype constraint

) ;

Create table student (

?d int primary key,

Name varchar(50),

Age int not null

) ;

www query.
Select query.

Date.....

* Insert into student values (1, "nivin", 19);

* Insert into student values (2, "chheu", 21);

* Select * from student; #display
(all) table.

Data types in SQL

1.) char (0-255) Store char of fixed length

2.) Varchar (0-255) Store char of variable length.

3.) BLOB (0 - 65535 bytes)

4.) int

5.) tinyint

6.) bigint

7.) BIT (2) can store x-bit values . x range from 1 to 64.

BIT(1)

↓

0, 1

BIT(2)

↓

00 10

01 11

Spiral

Teacher's Sign

Date.....

8.) float.

9.) Double

10.) Boolean

11.) Date - YYYY-MM-DD

12.) time

signed & unsigned



(-ve) + (+ve)

numeric

int

float

double

tiny int.

tinyint (-128 to 127)

tinyint unsigned (0 to 255)
(range increases).

Types of SQL Commands → 5 diff types.

→ DDL (Data Query language) →

Date.....

Database Related Queries.

- 1) create database db-name;
- 2) create database IF NOT EXISTS db-name;
→ Preferred as it shows warning.
- 3) Drop database db-name;
- 4) Drop database If Exist db-name;
- 5) Show databases; # shows all the database.
- 6) Show tables;

Table related Queries

- 1) create

create table table-name(

column name1 datatype constraint,

column name2 datatype constraint +

)

Spiral

Teacher's Sign

Date.....

2) select & view all columns.

Select * from table_name;
↳ means all.

3) insert into student (rollno, name) values
(rollno, name)

values:

(100, "karan"),
(101, "nisha");

or short version.

↓.

insert into student values (104, "Chyam");

Practise Q1

create a database for your company named
X Y Z

→ create a table inside this DB to store
employee info (id, name, salary)

→ add following info:

1, "adam", 25000
2, "bob", 30000
3, "casey", 40000

→ select and
view all
your table
data.

Spiral

Teacher's Sign.....

Date.....

keys

foreign key.

Primary key
(unique, not null)

- Main key
- column or set of columns that uniquely identify each row (a unique id)

→ there is (Primary key and it not be null)

Eg → id is primary key;

jo column dusre

kisi key ke ander

primary key hota hai

vo kisi aur

column ke andar

aa kar foreign

key kam jata

hai

Eg →

table - 1

(PK)

(FK)

id (PK)	name	city_id	city
101	Nishil	1	Pune
102	Shivaji	2	Mumbai
103	Nikunj	1	Pune

table - 2

(PK)

id (PK)	city name
1	Pune
2	Mumbai
3	Delhi

iski primary key table - 1

mei aa kar foreign key

hoga.

Date.....

foreign key here can have null values as well
as duplicate values.

Constraints

use to specify rules for data in a table

```
create table student (
```

```
    column names datatype constraints );
```

i) NOT NULL

2.) Unique values

3.) primary key → unique & not null.

primary key syntax:

```
create table temp1 (
```

```
    id int primary key, or
```

```
    id int,
```

```
    name varchar(50),
```

```
    age int
```

```
    city varchar(50),
```

```
    primary key (id, name)
```

```
);
```

ii) foreign key

Date.....

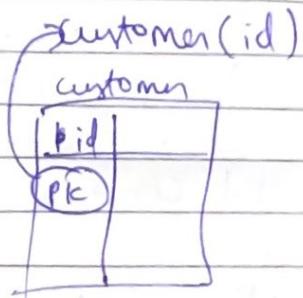
Foreign key - present ~~act~~

create table temp (

cust_id int

foreign key (cust_id) references (

);



Default → to set default values of a column.

salary int default 25000 ;

Create table emp (
id int primary key,
name varchar (50),
salary int default 25000
);

insert into emp (id, name) values (69, "Dilip");

Date.....

- check → limit the values allowed in a column.

```
create table city (
    id int primary key,
    name varchar(50),
    age int,
    constraint age-check check (age>=18
        AND city='Delhi')
);
```

Or.

```
create table city (
    age int check (age>=18)
);
```

Select →

```
select * from student;
```

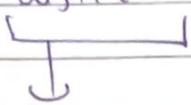
all

```
select name, marks from student;
```

specific

Date.....

Select distinct city from students



Shows unique

where clause → to define some condition.

Select to search we can use various clauses.

Select col1, col2 from table-name
where conditions;

→ Select * from student where marks > 80.

→ Select * from student where city = "mumbai".

Select *

from student

where marks > 80;

or where marks > 80
AND city = "mumbai"

Teacher's Sign.....

Spiral

Date.....

Using Operator in where.

1) arithmetic operator → +, -, *, /, %

2) comparison operator : =, !=, >, >=, <, <=

3) logical operator → AND, OR, NOT, IN BETWEEN,
ALL, LIKE, ANY.

4) Bitwise operator: & (Bitwise AND), | (Bitwise OR)

① AND operator. to check for both conditions to be true.

Select * from students where marks > 90
AND city = "mumbai";

② OR to check if one condition is true.

③ between → selects a given range.

Select * from students where marks Between
80 AND 90;

④ IN operator

Select * from students where city in ("Delhi",
"Mumbai")

Spiral

Teacher's Sign.....

Date.....

(5)

NOT (to negate the given condition)

Select * from student where city not in

("Delhi", "mumbai");

limit clause -> set an upper limit

on number of rows to be returned.

Select * from student LIMIT 3;

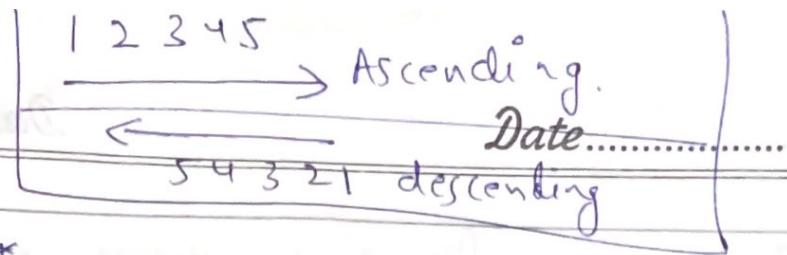
Or.

Select *
From students
where marks > 80
Limit 3;

order by clause

to sort in ascending or descending order.

Select * from student →



Select *
from student
order by city ASC; # order by
DESC

Aggregate functions →

functions perform a calculation on a set of values, and return a single value.

- count()
- max()
- min()
- sum()
- avg()
-

→ Return Only 1 value.

max: Select max(marks)
from student;

avg: select avg(marks)
from student;

Date.....

group by clause → group rows that have the same value into summary rows.

Eg → pure city be under jitesh student hi, marks.

* generally we use group by with some aggregation function.

→ Count number of student in each city

select city, count(name)
from student
group by city;

HAVING clause

where clause → used on group columns.

having clause → used on groups.

Eg → Count no. of student in each city
Where max marks cross 90.

Date.....

General order

select columns
from table name
where condition
group by column
having condition
order by columns ASC, DESC;

where → rows
having → groups.

TABLE OPERATION

1.) UPDATE → to update existing rows.

update table name
set col1 = val1, col2 = val2
where condition;

update student
set grade = "D"
where grade = "A";

Date.....

By default, there is an safe mode in
SQL to turn it off →
Set sql-safe-updates = 0;

update student
Set mgrade = "B"
Where marks between 80 AND 90;

2.) DELETE → to delete existing rows

Delete from table-name
Where condition

Eg → Delete from student
Where marks < 33;

Or.

Delete from student; # remove all
the data from
student.

Spiral

Teacher's Sign.....

Date.....

Revisiting foreign keys.

Dept.

id	name
101	Sci
102	English
103	Hindi

(Parent table)

teacher.

id	name	dept-id
101	Adam	101
102	bob	103
103	casey	102
104	Donald	102

(child table)

Foreign key (dept_id) REFERENCES dept()

Reverse Engineer \rightarrow * R.

\rightarrow EER diagram

\rightarrow (entity relationship
diagrams)

cascading for fk.

\rightarrow Ek gaajah change hua toh dusri place
bhi change ho ja.

ON DELETE CASCADE
ON UPDATE CASCADE.

iske agar such delete ya update hota
hai toh us foreign key mei &
dono table mi ho jata hei.

Date.....

Table related queries.

→ Alter (to change the schema).



column, datatype,
constraints.

add column

1.) alter table payment
add column age int

2.) drop column →

alter table table_name
drop column column_name;

3.) rename table.

alter table table-name
rename to new_table_name

4.) change column (rename)

alter table table_name
change column old_name new_name newname(DT)

Date.....

5) Modify column. (Modify datatype/constraint)

alter table table_name

modify col_name new_datatype new_constraint;

table related Query.

Truncate (to delete table's data)

Truncate table table_name;

DROP

L.

Deletes table

TRUNCATE

L.

Delete table's data.