

18G

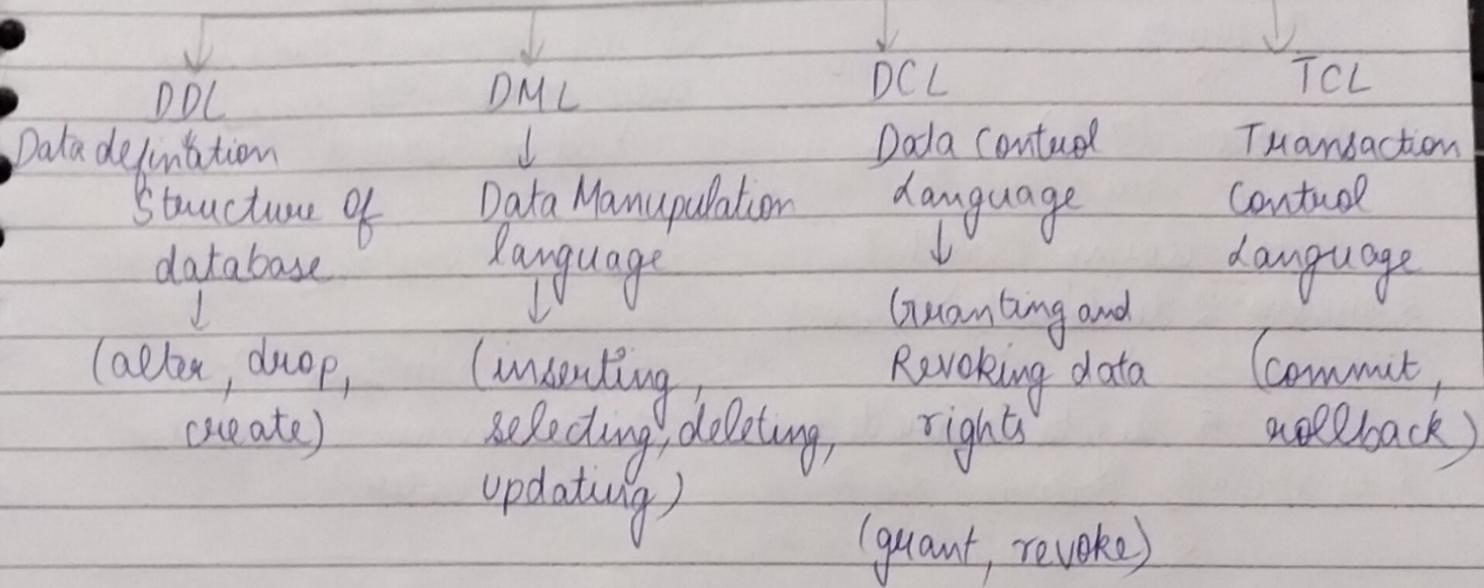
DBMS LAB

PAGE NO.	100
DATE :	/ /

SQL - Structured Query language

Q) SQL is procedural or Non Procedural?
Ans) Non Procedural

- PL/SQL - Procedural (what I need, how to fetch, a complete procedure is specified to fetch particular data)



Q) Alter vs Update → Changing values of cell / record / column

↓
Changing the structure of database eg - adding new column

field = column

PAGE NO.
DATE: / /

SQL DATATYPES

1. number

- integer datatype

e.g) number (5)

A particular field
will have max
5 digits

e.g) number (7,2)

↓ (equivalent to float)

Total column length=7

2 means 2 digits after decimal

$7 - 2 = 5$ digits before decimal

2 is already included in 7

2. char

- name char (6);

6 characters are reserved
for name column

| P | E | A | R | C |

Max u can enter is 6

• we are entering 5 characters
only so rest of space
will be wasted

varchar (variable character)

- name varchar (6);

• maximum 6 characters u
can enter

3. date

| P | E | A | R | L |

"dd/mm/yyyy"

• It will only allocate us
5 characters

• Rest of space will not get
wasted

ONLINE COMPILER

PAGE NO.

DATE: / /

create table student26 (rollno number, name varchar(10),
dob date, fee number (7,2));

// Table is created

insert into student26 values (101, 'Ajay', '11/Dec/1998', 1234.67);

(In month 3

characters can come)

insert into student26 (rollno, name, dob) values (102, 'Ahan', '12/Mar/1999');
when we don't want to insert fees

select * from student26;
↓ All values

desc student26; Structure of table will be shown

rename student26 to student; Table name is changed

select * from student where roll-no = '101';
Particular value

select roll-no from student;

Only one column from student table will be displayed

alter table student add (address varchar(20)); //Add extra column

// Dealing with values update.

update student set address = 'kota' where roll-no = 101;

// Creating other table

Create table student123 (roll-no number, books_issued number);

insert into student123 values (103, 3);

// Union rollno. It will combine all values of rollno of both
select rollno from student union all select roll no
from student 123;

// Delete entire data from table.
delete from student;

// Delete specific row
delete from student where roll no = 102;

DROP Vs TRUNCATE

Deletes table
content as
well as
table from
memory

Deletes table contents as well as
table from memory
(Entire structure gets deleted)

drop table student;
truncate table student 123;

// Union roll no. It will combine all values of roll no of both tables
 select roll no from student union all select roll no from student 123;

// Delete entire data from table
 delete from student;

// Delete specific row
 delete from student where roll no = 102;

DROP Vs TRUNCATE

Deletes table content as well as table from memory

Deletes table contents as well as table from memory
 (entire structure gets deleted)

drop table student;
 truncate table student 123;

create table stud (roll no number, name varchar(10),
 age number, fee number);

insert into stud (101, 'Ajay', 16, 1000);

insert into stud (102, 'Ajay', 16, 12000);

insert " " (103, 'Ajit', 17, 15000);

" (104, Amity, 18, 19000);

select * from stud where name = 'Ajit';

PAGE NO.	1
DATE:	1/1

only one icon

student

like clause - used with string type data or char type data

select * from stud where name like 'AJ%';

First & char of name are a and %

(means presence of o or any no of character)

% is a wildcard character which is used with like clause and represents exactly one character

so 2 entries

select * from stud where name like '%i%';

name with third char is i

check kaise ye hai
kyo table me

select * from stud where name in ('Ajay', 'Ajit');

clause

Salary b/w 15000 and

select roll-no, name from stud where fee between
15000 and 19000;

Upper limit and lower limit is included

// Order by Used to sort data

ascending order by

select * from stud order by roll-no default

select * from stud order by roll-no desc

decending order
not describe

select * from stud order by

name desc;

Done on
steung

PAGE NO.

DATE: / /

// And clause Age 116 and fee = 10000

select * from stud where age = 16 and fee = 10000;
both should be true

// OR clause Any condition true

select * from stud where age = 16 or fee = 1000;

// list roll_no, name, age , fe and 15% hike in fee

seled roll-no, name, age, fee, fee + fee * 0.15 from
stud,
keyword

select roll_no, age, name, fee, fee + 0.15 as updated fee
from stud;

Dept

DID	ENO	Salary
10	101	10000
10	102	20000
10	103	30000
20	104	50000
20	105	40000

group by

Dept 10 ki total

Dept 20 ki - total

select DID, sum(salary) from dept group by DID

count = no. of rows

requires aggregate function like min, max, count

create table dept(did number, eid number, sal number);
insert into dept (10, 101, 10000);

PAGE NO.	icon-
DATE:	/ /

insert into dept (10, 102, 20000);
insert into dept (10, 103, 30000);
" " " (20, 104, 50000);
" " " (20, 101, 60000);

select did, sum(sal) from dept group by did;

select did, max(sal) " " " " ;
" " min(sal) " " " " ;

select sum(sal) from dept;

create table dept (did number, eid number, sal number);
insert into dept (10, 101, 10000);
insert into dept (10, 102, 20000);
insert into dept (10, 103, 30000);
" " " (20, 104, 50000);
" " " (20, 105, 60000);

PAGE NO.	1
DATE:	/ /

select did, sum(sal) from dept group by did;

select did, max(sal) " " " "

" " min(sal) " " " "

select sum(sal) from dept;

Commenting

② PL/SQL Procedure

DECLARE

Declare all variable

BEGIN

dbms_output.put_line('HELLO');

Output will print

END;

You concatenation

dbms_output.put_line('HELLO' || 'Welcome to Class');

You concatenation

DECLARE Assignment

a number := 7;

b number := 8;

total := 0;

BEGIN

total := a + b;

dbms_output.put_line('Sum of 2 nos is ' || total);

```

BEGIN
if (mod(var1,2) == 0) then
    dbms_output.put-line('even');
else
    dbms_output.put-line('odd');
endif;
END;

```

= comparison
sign

:= Assignment

| DECLARE

~~BEGIN~~

num1 number;

BEGIN num1 := 9;

if (num1 = 0) then

dbms_output.put-line("number is 0");

elsif (num1 > 0) then

dbms_output.put-line("number is +ve");

else

dbms_output.put-line("num is -ve");

endif

END;

DECLARE

ij number

BEGIN

ij := 1

while (ij <= 10) loop

dbms_output.put-line(ij);

ij := ij + 1;

end loop;

END

Output

1

2

3

4

5

6

7

8

9

DECLARE

ij number;

BEGIN

ij := 1

for ij in 1...10 loop

dbms-output.put-line(ij);

end loop;

END;

PAG
DAT