R → Refired.

System table

30/11/2021

* 63 system tables in mysql.
  - Stored in information-schema
  - eg. statistics (stores indexes information), table-constraints, key-column-usage, etc.
  - set of system tables is known as Data dictionary. (also known as database cateloge)

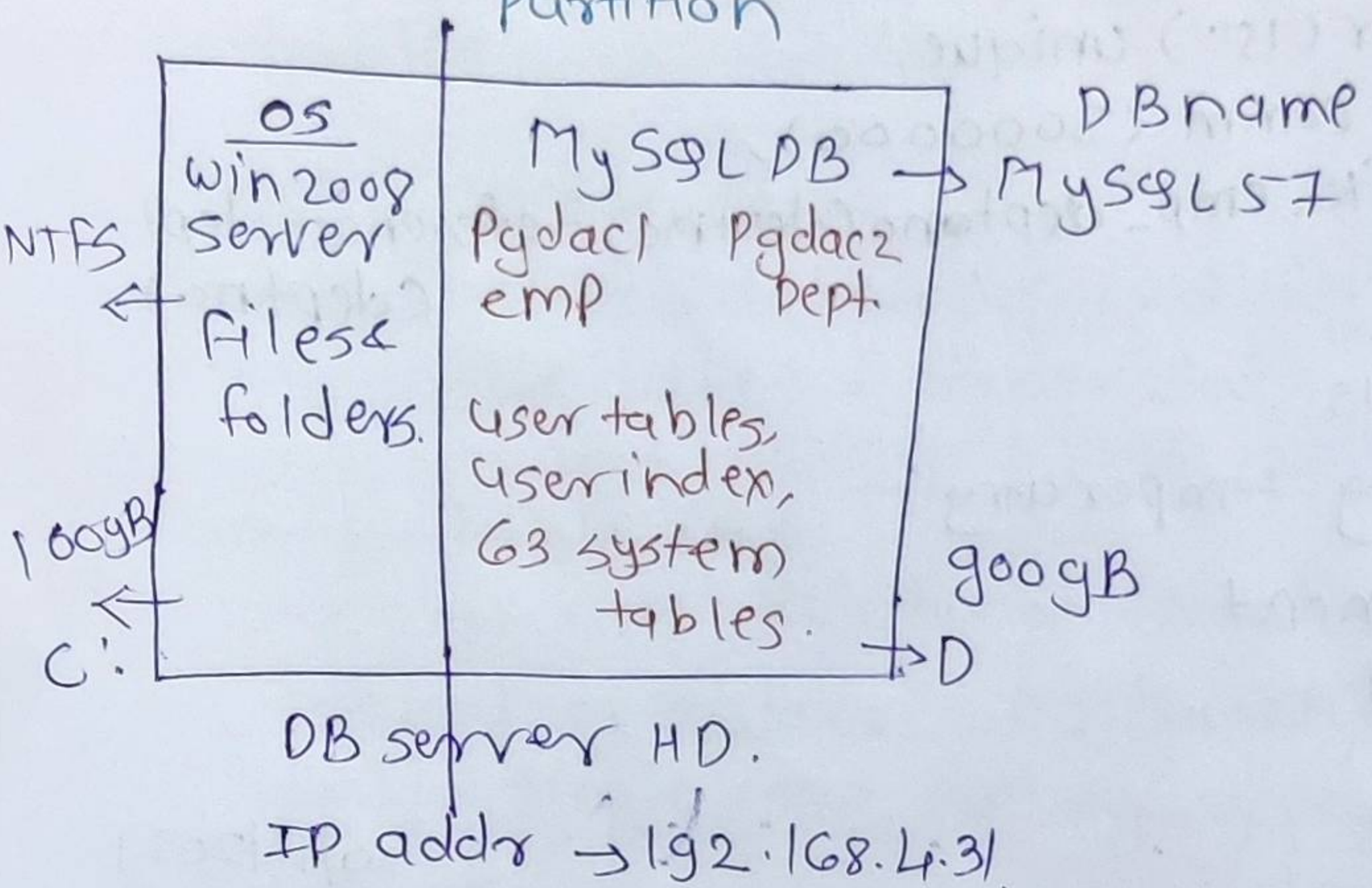
Data is of 2 types:-

① user data.
  - user created
  - eg. tables and indexes.

② system data.
  - system created (MySql created).
  - data that is ~~set~~ stored in the system table

- 63 system tables in MySQL (automatically created when you installed mysql).
- system tables store complete information about the database.

- All system tables are read only; you cannot insert, update and delete. you can only select from the system tables.
- DDL for user is DML for system tables.



Stored objects:
- object that are stored in the database
  e.g. tables, indexes

Views:

EMP

Empno	ENAME	SAL	deptno.
1	A	5000	1
2	B	6000	1
3	C	7000	1
4	D	9000	2
5	E	8000	2

owner → Pgdac1
schema-name → cdac mumbai
user name 2 — scott.

- Views are present in all RDBMS and som DBMS also.
- View is a handalle to table.
- View is hard disk pointer (stores the address of table)
- hard disk pointer is known as a Locator
  - used for indirect access to the table.
  - used for security purposes.

#Create view viewname ----.

- Create view V1. (max 30 char.)
  - view name table name can not be same.
- Create view V1
  as
  select empno, ename from emp; (To create a view)
  - view is a stored query.
- grant select on v1 to scott;
- select* from cdacmumbai.emp ← error.
- select * from cdacmumbai.V1;
  - used to restrict the column access.
  - View does not contain data.
  - only the defination is stored, data is not stored
  - view is a stored query.
- select statement on which is based, it is stored in the DB in the compiled format.
- view is an executable format of select statement.
- hence the execution will be very fast
- hiding source code from end users.

- grant select, insert on V1 to scott;
- insert into cdacmumbai.~~emp~~ V1 values(6,'f');
  - DML operations can be performed on a view
  - DML operations done on a view will affect the base table
  - constraints that are specified on the table will be enforced even when you insert via the view.
  - entire application is built using views

- drop view V1;

- Create view V2 as
  select * from emp where deptno = 1;

mysql
- grant select, insert on V2 to niraj, snehal;

N
- select * from cdacmumbai.V2;

Empno	Enume	sal	dept
1	A	5000	1
2	B	6000	1
3	C	7000	1

- use to restrict row access,

- insert into cdacmumbai.V2 values(6,'f',6000,2);

- create view V2
  as select * from empwhere deptno = 1 with check option
  - view with check option is similar to check constraint
  - with help of this feature you can enforce different checks for different users.

to change the select statement on which the view is based:-
- drop view V1;
- create view V1 as select ...

- create or replace view V1
  as
  select ename, sal from emp;
- desc V1;
- Create or replace viewV1
  as
  select ename, sal*12 annual fromemp;
  - select * from V1;
- view based on computed column, expression, function, etc.
- you can only select from this view.
- DML operations are not allowed (common for all RDBMS)
- in oracle you can perform DML operations on this view by writing insted of trigger.

View based on join.
- you can only select from this view.
- DML operations are not allowed (common for all RDBMS)

to view views.
- show tables; ← will show tables & views but it won't tell which is which).

to see which is a table and which is a view
- show full tables;

to see the select the statement on which the view is based.
- show create view V1;

View based on view.
- used to exceed the limits of SQL.

  e.g.
  1. function within function > 255-level
  2. subqueries > 255 levels.
  3. union > 255 select statement
- to simplify the writing of complex select statements.

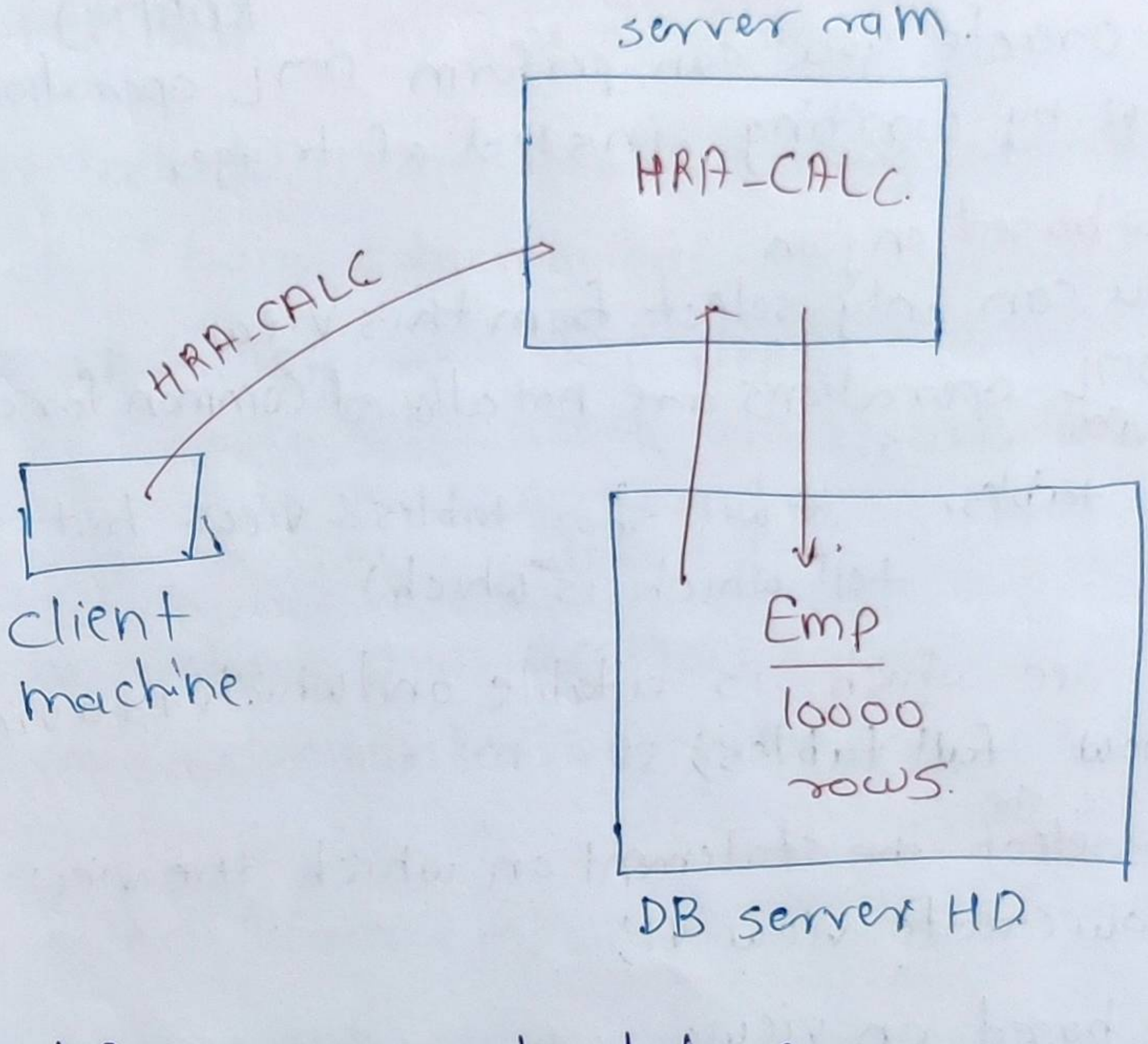
  e.g. join of 20 tables.

PL- Programming language:

- product of MySQL.
- works only with MySQL RDBMS
- MySQL - programming language.
- used for database programming.
  e.g HRA_CALC, Attendance_calc, Tax-cal, etc.

MS SQL server → TSQL → Transact SQL
Oracle → PL/SQL → Procedural language programming

- used for processing.



- used for server-side data processing.
- MySQL-PL program can be called for MySQL command line client, MySQL workbench, oracle forms, oracle reports, menus, graphics, oracle Apex (oracle application express), etc, java, Ms. net, etc.
- can be called through any frontend S/W

- 4 GL (supports few oops features) (4 generation language

```
BEGIN
    Insert into dept values(1,'a','B');
END;
```

MySQL-PL Block.

- Begin (main block) (Parent) (outer)
  ≡ (X)
  Begin (sub block) (child) (inner).
  ≡ (Y)
  End;
  ≡
  End;

- you can have block within a block.
- Block level language. (feature of OOPS)
- Benefits of block level language:-
  a. Modularity.
  b. Control the scope of variables
  Encapsulation (form of Data hiding).
  c. Exceptions (efficient error Management)
- Screen input and screen output is not allowed (eg. scanf, printf, SOP, ... , not allowed)
- MySQL-PL is used only for processing.
- Can use select statment inside the block but its no recommended.
- SQL commands that allowed inside the block:- DDL, DML, DQL, DTL/TCL.
- DCL commands are not allowed inside the block.

- Create table tempp (To ~~creat~~ store the output of My-SQL PL program):

```
(
  fir int,
  sec char(15)
);
```

- My SQL-PL program is written in the form of stored Procedures

Stored Objects:

- objects that are stored in the database.
- Create ---- tables, indexes, views.

Stored procedures:-

- Routine (set of commands) that has to be called explicitly.
  - call abc();
- global procedures;
  - stored in database
  - Can be called through MySQL command line client, MySQL workbench, java, Ms.net, etc.
- can be called through any front-end software
- stored in the database in compiled formate.
- hiding source code from end user.
- execcution is in server ram
- Procedure can have local variables
- within the procedure you can have any processing all MySQL-PL statements allowed
  e.g. If statement, loops, cursors, etc.
- One procedure can call another procedure.
- Procedure can call itself (known as Recursion).
- you can pass parameters to a procedure.
- Overloading of procedure is not allowed because it's stored object.

- Create procedure ()
  begin
    insert into tempp values(1,'Hello');
  end;
  ⇒ Read, compile, plan, store it in the DB in the compiled format
  - Procedure created

- delimiter :- ; default delimiter
  cindicates end of command)
  ← we can change to any character. *, @, e

- delimiter //
  create procedure abc() (created permently
  begin
    insert into tempp values(1,'Hello'); / commit;
  end; //
  delimiter; (optional)
  - call abc();
  - select * from tempp;

  output: FIR SEC.
  1 Hello.

- drop procedure abc;

delimiter //
  create procedure abc()
  begin
    declaare x int; }/ decleare x int default 10;
    set x = 10;
    insert into tempp values(x,'Hello');
  end; //
  delimiter;
  call abc();
  select * from tempp;

server Ram
int x
null

- In MySQL - PL
  when you decleare a variable, if you don't initialize it, then it will store a null value.
- you can declare variable & initialize it simultaneasly.

```
delimiter//
create procedure abc()
begin
    declare x char(15) defaut 'CDAC Mumbai';
    insert into tempp values(1,x);
end; //
delimiter;
call abc();
select *. from tempp;
```

HRA = 40% Sal.

```
delimiter //
(reate pro cedure   abc()
 begin
     decleare  xchar(15 )default  'KING';
     decleare  y  float  default  3000;
     declare   z float  default  0.4;
     declare   hra float;
     set hra = y*z;
     insert into tempp value s(y,x);
     insert into temp values(hra, 'HRA');
end; //
delimiter;
 call abc();
 select* from tempp;
```

```
• delimiter //
  create procedure abc (x char(15), y float, z float)
  begin
    declare hra float;
    set hra = y*z;
    insert into tempp values (y, x);
    insert into tempp values (hra, 'HRA');
  end; //
  delimiter;
  call abc ('KING', 3000, 0.4);
  call abc ('SCOTT', 2500, 0.15);
```

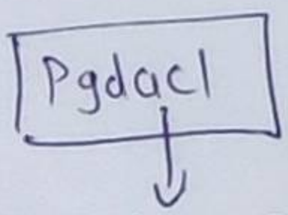
```
• -- single line comment.
  /* multiline
     comment */
```

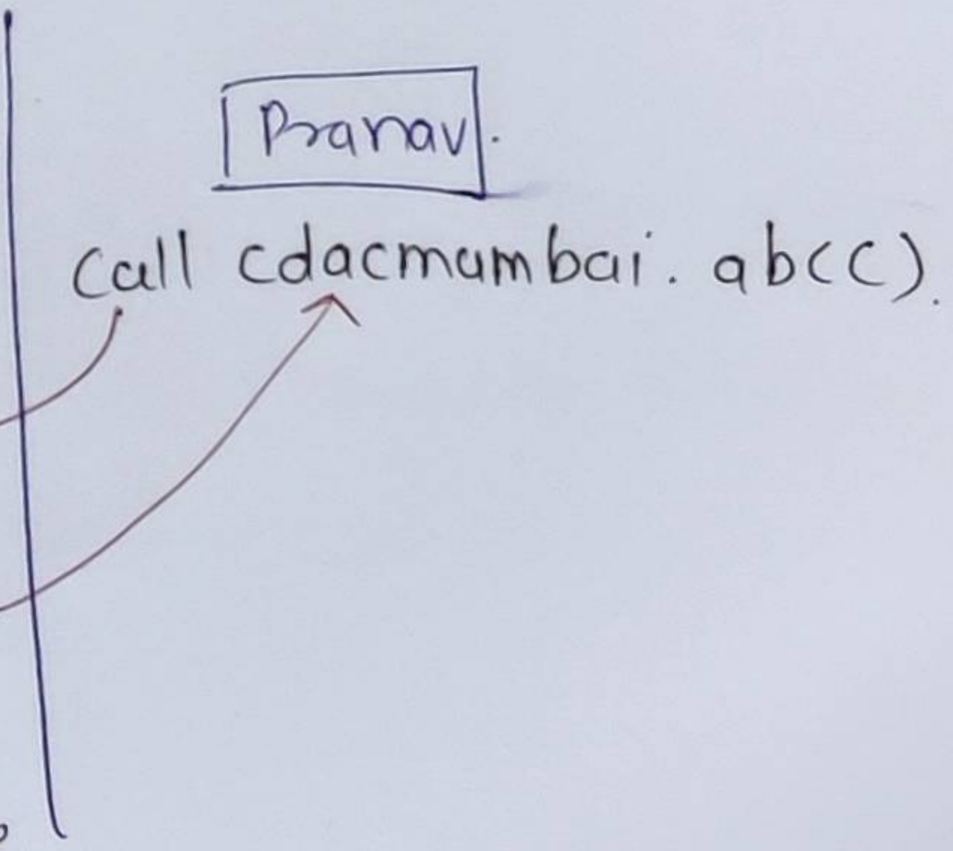
• To see which all procedures are available:-

- show procedure status;
  show all procedures in all databases.
- show procedure status
  where db = 'cdacmumbai';

---

user name → pgdacl
schemaname → cdacmumbai;





grant execute on procedure
abc to pranav;

revoke execute on procedure
abc for pranav;