19-10-24

Day 11

PL-CURSORS (Most Important)

	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

Row ID : X1,X2,X3,X4,X5

{empno int,ename varchar (15),sal int , deptno int}

Tempp table (Output table)

	TEMPP
FIR	SEC

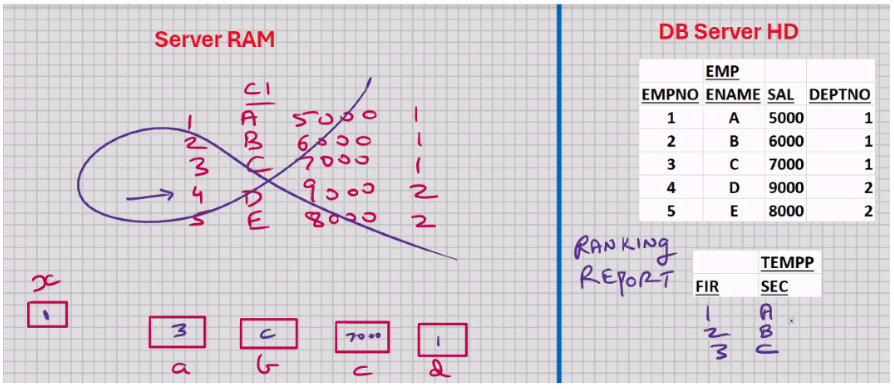
- Present in all RDBMS , Some of the DBMS & some of the front ends softwares
- a cursor is a type of a variable
- Cursor can Store Multiple rows
- Cursor is Similar to 2-D array

```
declare pqr cursor for select * from emp
```

- Use for storing multiple rows
- Use for Proccesing Multiple rows
- Use for handling Multiple rows
- Use for storing the data temporary

```
delimiter //
crate procedure abc()
begin
    declare a int;
    declare b varchar(15);
    declare c int;
    declare d int;
    declare x int default 0;
    declare c2 cursor for select * from emp;  <--- here just
    open c1;  <-- open curson /execute select statement /popu
    while x< 5 do
        fetch c1 into a,b,c,d; <-- fetch next row
        /* processing, e.g. set hra= c*0.4,etc.*/
        insert into tempp values(a,b);
        set x=x+1;
    end while;
    close c1;  <-- free the ram
end;//
delimiter ;
```

- cursor has to be declare after all the variables
- Cursor is based on select statement
- the select statement on which the cursor is based colud be anything , e.g.select col 1, col2....
- WHERE ,ORDERD BY ,GROUP BY clause
- bsed on join , subquery ,set operator,view
- computed columns ,expressions, function,etc
- cursor is a read only variable
- the data that is present inside the cursor , it cannot be manupulated ou will have to fetch 1 row at a time into some intermidiate variables, and do your proccesing with those variables
- you can only fetch sequencially (top to bottom)
- you can only fetch 1 row at a time



- `while x<11` <-- it will give error

```
delimiter //
crate procedure abc()
begin
    declare a int;
    declare b varchar(15);
    declare c int;
    declare d int;
    declare x int default 0;
    declare y int ;
    declare c2 cursor for select * from emp;  <--- here just
    select count(*) int y from emp ; here y is no of rows in
    open c1;  <-- open curson /execute select statement /popu
    while x< 5 do
        fetch c1 into a,b,c,d; <-- fetch next row
        /* processing, e.g. set hra= c*0.4,etc.*/
        insert into tempp values(a,b);
        set x=x+1;
    end while;
    close c1;  <-- free the ram
end;//
delimiter ;
```

- declare a continue handler for not found event
- not found is a cursor attribute ; it returns a boolean true value is the last fetch was unsucssesful and false if thew last fetch was sucssesful

```
delimiter //
crate procedure abc()
begin
    declare a int;
    declare b varchar(15);
    declare c int;
    declare d int;
    declare y int default 0;
    declare c1 cursor for select * from emp;
    declare continue hander for noot found set y=1;
    open c1;
    cursor_c1_loop:loop
        fetch c1 into a,b,c,d
        if  y=1 then
        leave cursor_c1_loop;
        end if;
        insert into tempp values (a,b);
    end loop cursor_c1_loop;
    close c1;
end;//
```

- by using parameters we can make it flexible

uses :

- Locking the rows manually
- storing / proccesing multiple rows

```
Use of Cursors:-
a. used for storing/processing multiple rows
b. USED FOR LOCKING THE ROWS MANUALLY

delimiter //
create procedure abc()
begin
        declare c1 cursor for select * from emp for update;
        open c1;
        close c1;
end; //
delimiter ;

mysql> call abc();
```

- if you want to lock rows manually open and close cursor
- when rollback done then lock automatically get reasles

Types of parameters

1. In Parameters (by default)
    - read only

```
crete procedure abc(in y int)
begin
    insert into tempp values(y,'inside abc')
    end;//
    delimiter;
    ---------------------
    delimeter //
    create procedure pqr()
    begin
        declare x int default ;
        call abc(5);
        call abc(x);
        call abc(2*x+5);
    end;//
    delimiter;
    ---------------------
    call pqr();
```

---

2. Out Parameters

```
delimiter//
create procedure abc(out y int )
begin
    set y=100;
end;//
delimiter ;
-------------------------
delimiter //
create procedure pqr()
begin
    declare x int default 10;
    insert into tempp values(x,'before abc');
    call abc(x);
    insert into tempp values(x,'after abc');
end;//
delimiter ;
-------------------------
call pqr();
```

- Write only
- we can pass variables only not expression , constants
- call by referance
-

---

3. INOUT parameters

- Read and Write

```
DBT.txt        DBT2.txt
File   Edit   View
*        procedure can return a value indirectly if you call by reference

delimiter //
create procedure abc(inout y int)
begin
        set y = y*y*y;
end; //
delimiter ;
--------------------------------------------
delimiter //
create procedure pqr()
begin
        declare x int default 10;
        insert into tempp values(x, 'before abc');
        call abc(x);
        insert into tempp values(x, 'after abc');
end; //
delimiter ;
----------------------------------------
call pqr();
```

- call by referance
- read write and returen value
- used in local networks
- most powerful and best functionality

Stored Functions

Stored Objects

- Object that are strored in the database
- e.g. CRETAE,index,views ,stored procedure
- anythenf that you do with create command is a stored object

STORED FUNCTION

- Routine that returns a value directly and compulsorily
- Stored functions are global functions
- store din the database
- can be called in MySQL command line client , My sql Workbench ,java
- stored in the databsa in compiled formate
- hance execution is fast
- hiding the source code from end user
- Within the function all Mysql-pl statements are alloewed
- stored procedure can call stored function
- stored function can call stored procedure
- function can call itself
- function can call another function
- to make it flexible we can pass parameter to function

- overloading of stored function is not allowed coz it is stored object ;you can not create 2 or more functions with the same name even if
- the numbers of parameters passed is differemnt or the DATATYPES of parameters passed is different
- in parameter only

Stored Functions are of two types

1. Deterministic
2. Not Deterministic

- for the same input parameters if the stored function returnds the same result , it is considered deterministic and other wise the stored function is not deterministic
- you have to decide wether a stored function is dertministic or not
- f you declare it incorrectly the stored function may produce an unexpected resul or the avalilable optimization is not used which degraddes the performance `mysql>call abc()`
- inlike a stored procedure or function can not be call itself becoz a function returns a value and that value has to be stored some where and therefore the function has to be equeted with a variable , or it has to be a part of some expression

How to creta a function :

```
delimiter //
create function abc()
returns int
deterministic
begin
    return 10;
end;//
delimiter;
```

"Function created"

- stored it in database in compailed formate

```
delimiter //
 create procedure pqr()
 begin
    declare x int ;
    set x= abc();
    insert into tempp values(x,'after abc');
end;//
delimiter;
```

drop function : `drop function abc()`

```
delimiter //
create function abc(y int)
returns int
deterministic
begin
    return y*y;
end;//
delimiter ;
----------------------------
delimiter //
creta procedure pqr()
begin
     declare x int;
     set x=abc(10);
     insert into tempp values(x,'after int');
end;//
delimiter ;-----------------------------
call pqr;
```

[STORED FUNCTIONS CAN BE CALLED IN SELECT STATEMENT]

[STORED FUNCTIONS CAN BE CALLED IN DML COMMANS ALSO]

```
delimiter //
create function abc (y int)
returns boolean
deterministic
begin
    if y> 5000 then
        retrun TURE ;
    else
        return FALSE;
    end if;
end;//
delimiter ;
---------------------------
deelimiter //
create procedure pqr()
begin
    declare x int;
    select sal into x from emp where ename='KING';
    if abc(x) then
        insert into tempp values(x,'>5000');
    else
        insert into tempp values(x,'<=5000');
    end if;
end;//
delimiter;
--------------
call pqr();
```

- function is normaly used as validation routine
- a function norammlly return a boolean TRUE or FALSE value , accordingly some future proccesing
- if function returns a boolean value , then you can directly use the functions name as a condition for if statement

to see all functions:`show function status ;`

to see database functions:`show function status where db='cdacmumbai';` to share functions with others

```
grant execute on functions cdacmumbai.abc to scott@localhost;
```

```
show function status where db = 'cdacmumbai';

show function status where name like 'a%';

To view the source code of stored function:-

show create function abc;

To share the function with other users:-

grant execute on function cdacmumbai.abc to scott@localhost;

scott_mysql> select cdacmumbai.abc() from dual;

revoke execute on function cdacmumbai.abc from scott@localhost;
```

EXAM TIP POINT

1. Create table
2. Insert
3. SELECT 5-10
4. stored procedure
5. stored function