

DBMS Tutorial – 9

Implement the following PL/SQL programs on Covid Pandemic Country-wise Database System

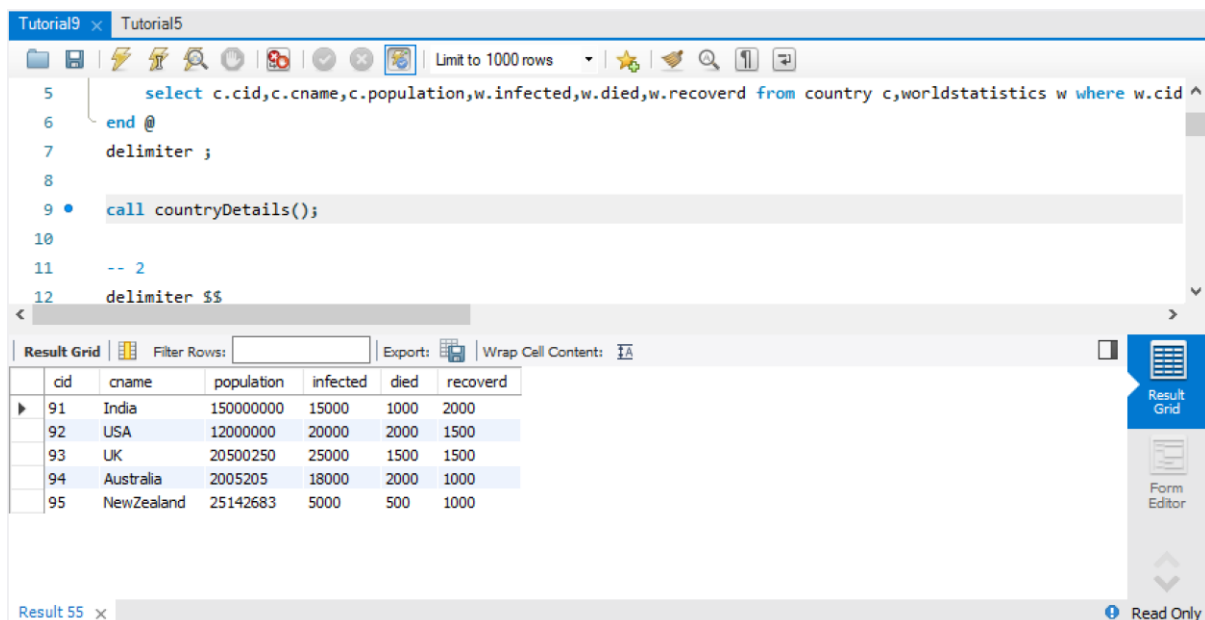
1) Create a PL/SQL program to display the details of countries

```

delimiter @
create procedure countryDetails()
begin
    select c.cid,c.cname,c.population,w.infected,w.died,w.recoverd from country
    c,worldstatistics w where w.cid=c.cid;
end @
delimiter ;

call countryDetails();

```



The screenshot shows a SQL IDE with two tabs: 'Tutorial9' and 'Tutorial5'. The SQL editor in 'Tutorial5' contains the following code:

```

5      select c.cid,c.cname,c.population,w.infected,w.died,w.recoverd from country c,worldstatistics w where w.cid =
6      end @
7      delimiter ;
8
9      call countryDetails();
10
11      -- 2
12      delimiter $$

```

Below the editor, the 'Result Grid' displays the output of the query. It shows a table with 6 columns: cid, cname, population, infected, died, and recoverd. The data is as follows:

cid	cname	population	infected	died	recoverd
91	India	150000000	15000	1000	2000
92	USA	120000000	20000	2000	1500
93	UK	20500250	25000	1500	1500
94	Australia	2005205	18000	2000	1000
95	NewZealand	25142683	5000	500	1000

The IDE also shows a 'Form Editor' button on the right and a 'Read Only' status at the bottom right.

2) Create a Cursor to display corona virus affected countries with no. of cases and death toll

```

delimiter $$
create procedure CasesDetails()
begin
    declare ctype varchar(20);
    declare cases int;
    declare death int;
    declare c_finished int default 0;
    declare c1 cursor for select cname,infected,died from worldstatistics;
    declare continue handler for not found set c_finished=1;
open c1;
    c_details:loop
        fetch c1 into ctype,cases,death;
        if c_finished=1 then

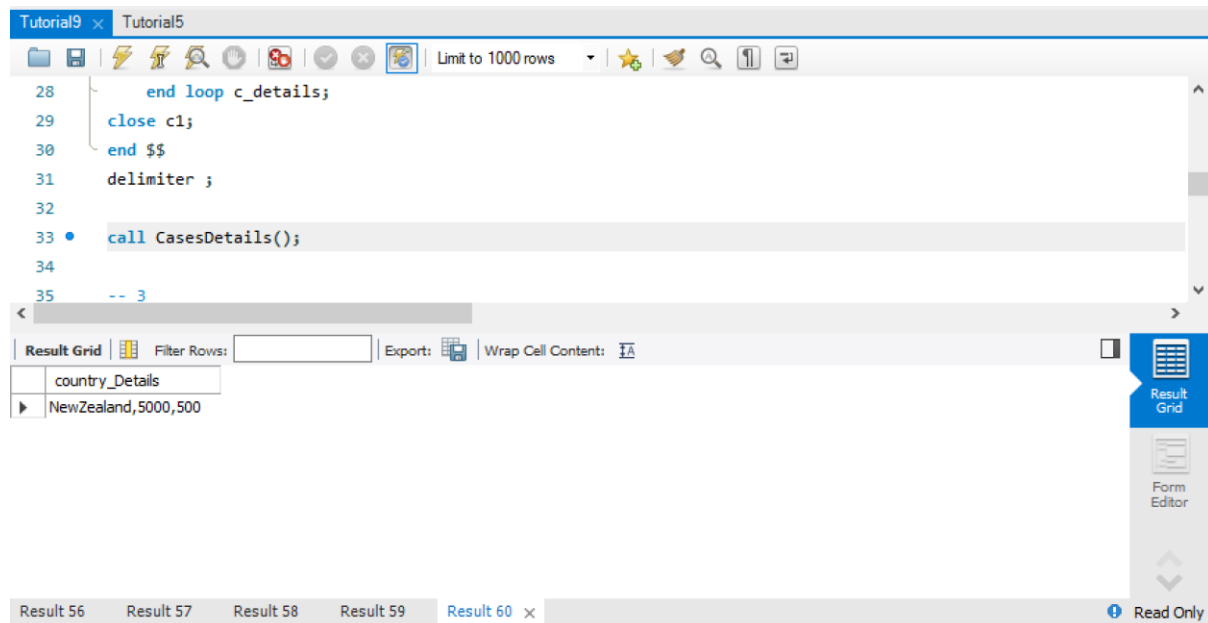
```

```

        leave c_details;
    end if;
    select concat(ctyname,',',cases,',',death) as country_Details;
    end loop c_details;
close c1;
end $$
delimiter ;

call CasesDetails();

```



3) Create a trigger to insert patient details into CASES table whenever the person status in PERSON table changes from Negative to Positive.

```

create table cases(ssn int,cid int,name varchar(20),result varchar(20),date varchar(20),status
varchar(20),changed_At TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP);

```

delimiter @@

create trigger addPatientLog after update on person

for each row

begin

if OLD.result='Negative' and NEW.result='Positive' then

insert into cases

values(OLD.ssn,OLD.cid,OLD.name,NEW.result,OLD.date,OLD.status,current_timestamp);

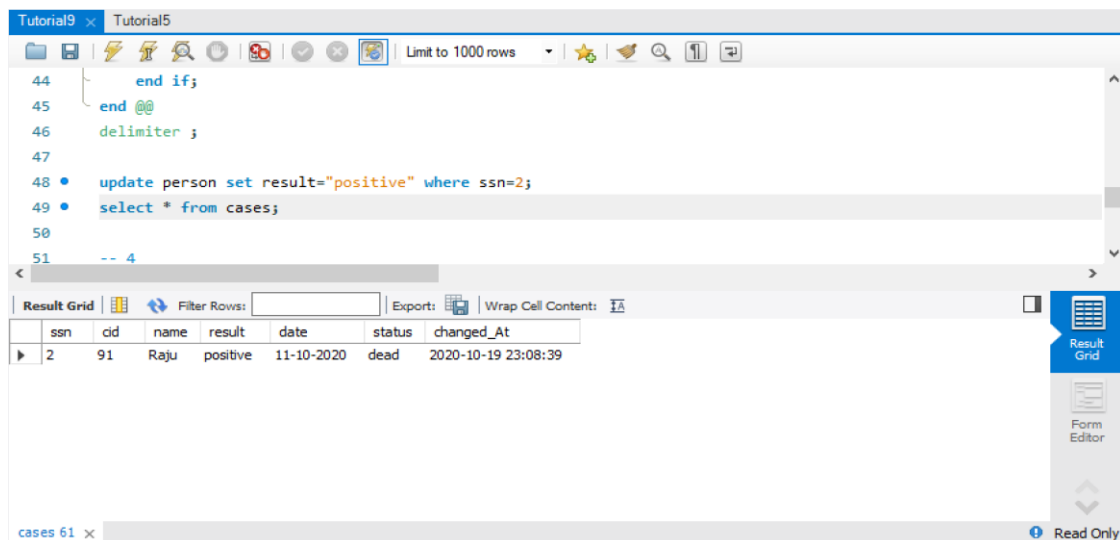
end if;

end @@

delimiter ;

update person set result="positive" where ssn=2;

select * from cases;



4) Create a trigger to store the person details whose status is dead.

```
create table personlog(ssn int,cid int,name varchar(20),result varchar(20),date
varchar(20),status varchar(20),changedAt TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP);
```

```
delimiter $
```

```
create trigger addPersonLog after insert on person
```

```
for each row
```

```
begin
```

```
    if NEW.status = 'dead' then
```

```
        insert into personlog
```

```
values(NEW.ssn,NEW.cid,NEW.name,NEW.result,NEW.date,NEW.status,current_timestam
p());
```

```
    end if;
```

```
end $
```

```
delimiter ;
```

```
insert into person values(13,95,'akhil','positive','12-10-2020','dead');
```

```
select * from personlog;
```

