1)select person.userid from person

join profession on

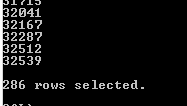
person.userid=profession.userid

where profession.education='Bachelors'

and person.native\_country<>'Cuba'

and person.native\_country like 'U%'

and person.race='Black';



2)

Select person.native\_country from person

join profession on

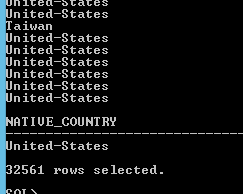
person.userid=profession.userid

where (select count(\*) from profession

where education='Bachelors' group by education) >

(select count(\*) from profession

where education='Doctorate' group by education);



3) select person.userid from person

join profession on

person.userid=profession.userid

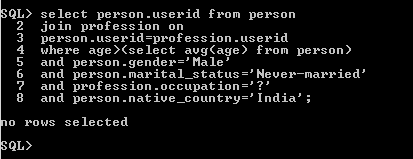
where age>(select avg(age) from person)

and person.gender='Male'

and person.marital\_status='Never-married'

and profession.occupation='?'

and person.native\_country='India';



4) declare

cursor cur1 is

(select person.userid,person.age from person

join profession

on person.userid=profession.userid

where profession.salary='>50K'

and profession.workclass='State-gov'

and person.marital\_status='Separated'

and rownum<=5);

var1 cur1%rowtype;

begin

open cur1;

loop

fetch cur1 into var1;

exit when cur1%notfound;

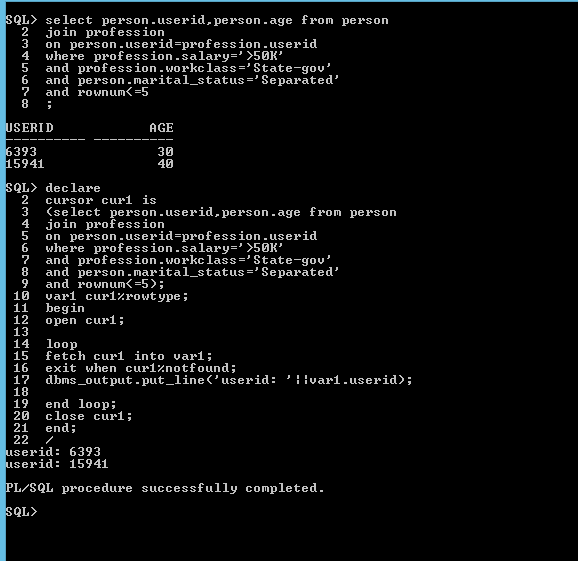
dbms\_output.put\_line('userid: '||var1.userid);

end loop;

close cur1;

end;

/



5)

declare

cursor cur1 is

(select userid from profession where education='Bachelors');

var1 cur1%rowtype;

begin

open cur1;

loop

fetch cur1 into var1;

exit when cur1%notfound;

update profession

set education='MSIT'

where userid=var1.userid;

commit;

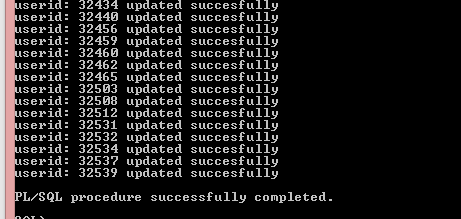
dbms\_output.put\_line('userid: '||var1.userid||' updated successfully’)

end loop;

close cur1;

end;

/



6) create or replace procedure proc1

var1 profession.userid%type;

cursor cur1 is

(select userid from persons where age>30

and education\_num = (select max(education\_num) from profession);

begin

open cur1;

loop

fetch cur1 into userid;

dbms\_output.put\_line('userid: '||var1.userid);

end loop;

end;

/

7) declare

userexc EXCEPTION;

var1 profession.hours\_per\_week%type;

begin

for temp in (select hours\_per\_week from profession) loop

if temp < 0 then

raise userexc;

end if;

end loop;

EXCEPTION

when userexc then

dbms\_output.put\_line('Exceprtion');

end;

/

8)

create or replace trigger var1

before insert on profession

for each row

begin

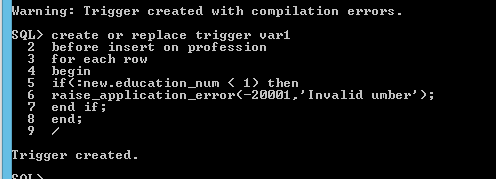
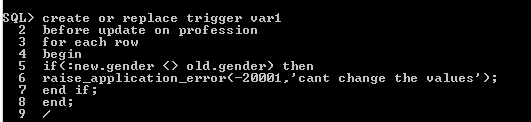
if(:new.education\_num < 1) then

raise\_application\_error(-20001,'Invalid umber');

end if;

end;

/



9) create or replace trigger var1

before update on profession

for each row

begin

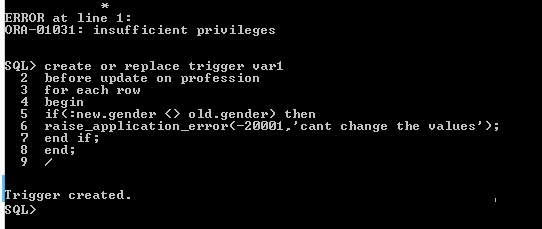
if(:new.gender <> old.gender) then

raise\_application\_error(-20001,'cant change the values');

end if;

end;

/



10)

