

სასწავლო კურსის სახელწოდება:	მონაცემთა ბაზების ადმინისტრირება
ლექტორი:	ალექსანდრე ჩახვაძე
სტუდენტის სახელი და გვარი	NIKITA GUDIEV
ჯგუფის #	7

ქვიზი 4

შუალედური გამოცდის ნიმუში

გამოცდაზე მუშაობთ თქვენთვის განკუთვნილ ამ ფაილში, სათანადო ადგილას უთითებთ საჭირო ინფორმაციებს, სვამთ სკრიპტებსა და სქრინებს. გამოცდაზე მუშაობთ ვირტუალურ გარემოში არსებულ ბაზაზე. გამოცდაზე შეგიძლიათ გამოიყენოთ ფაილი: oca_dba_exam.txt . დასრულებულს ფაილს დაარქვით სახელი და გვარი, დააპედუფეთ და ატვირთეთ საგამოცდო პროგრამის სათანადო ველში.

შეფასება: 15 საკითხი ჯამში 25 ქულა, კერძოდ საკითხები 1,2,4,6,7,8,9,10,11,12 თითო 2 ქულა, ხოლო საკითხები 3,5,13,14,15 თითო 1 ქულა.

1. (2 ქულა) დაუკავშირდით ორაკლ სერვერს როგორც პრივილეგიებული მომხმარებელი system და შექმენით როლი სკრიპტულად თქვენი გვარი_role და მიაწიქეთ შემდეგი სისტემური პრივილეგიები create session, create any table, create any view, create any index, create any sequence, create any synonym, alter any table, alter any index, alter any sequence, drop any table, drop any view, drop any index, drop any sequence, drop any synonym. მიაწიქეთ ობიექტ პრივილეგია მოსელექტება hr სქემის employees ცხრილზე. მიაწიქეთ როლი dba შექმნილ როლს.

სკრიპტი:

create role gudiev_role;

**grant create session, create any table, create any view, create any index, create any sequence,
create any synonym, alter any table, alter any index, alter any sequence, drop any table,
drop any view, drop any index, drop any sequence, drop any synonym
to gudiev_role;**

სქრინი:

<pre>create role gudiev_role; grant create session, create any table, create any view, create any index, create any sequence, create any synonym, alter any table, alter any index, alter any sequence, drop any table, drop any view, drop any index, drop any sequence, drop any synonym to gudiev_role;</pre>	
<div>Script Output x</div> <div>Task completed in 0.51 seconds</div>	
Grant succeeded.	

```
grant select on hr.employees to gudiev_role;
grant dba to gudiev_role;
```

Script Output x

Task completed in 0.137 seconds

Grant succeeded.

2. (2 ქულა) სკრიპტულად შექმენი მომხმარებელი **btu_გვარი** პაროლი btu, დეფაულტ ცხრილსივრცე users ულიმიტოდ, დროებით ცხრილსივრცედ temp, პროფაილი default. გადაამოწმეთ იუზერის არსებობა. მიაწიჭეთ შექმნილი როლი ამ მომხმარებელს.

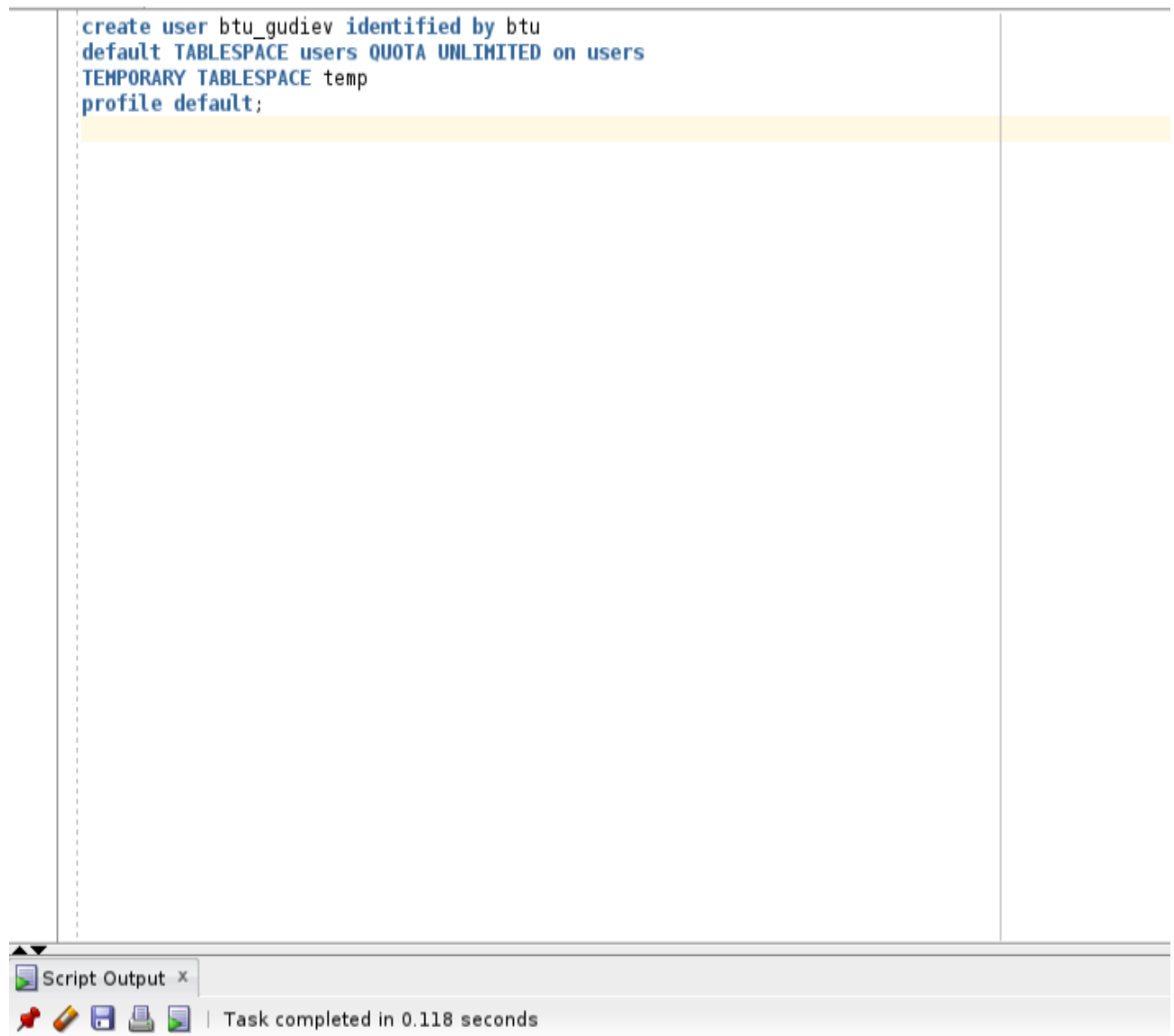
სკრიპტი:

```
create user btu_gudiev identified by btu
default TABLESPACE users QUOTA UNLIMITED on users
TEMPORARY TABLESPACE temp
profile default;
select * from dba_users where username ='BTU_GUDIEV';
```

```
grant gudiev_role to btu_gudiev;
```

სკრინი:

```
create user btu_gudiev identified by btu
default TABLESPACE users QUOTA UNLIMITED on users
TEMPORARY TABLESPACE temp
profile default;
```



User BTU_GUDIEV created.

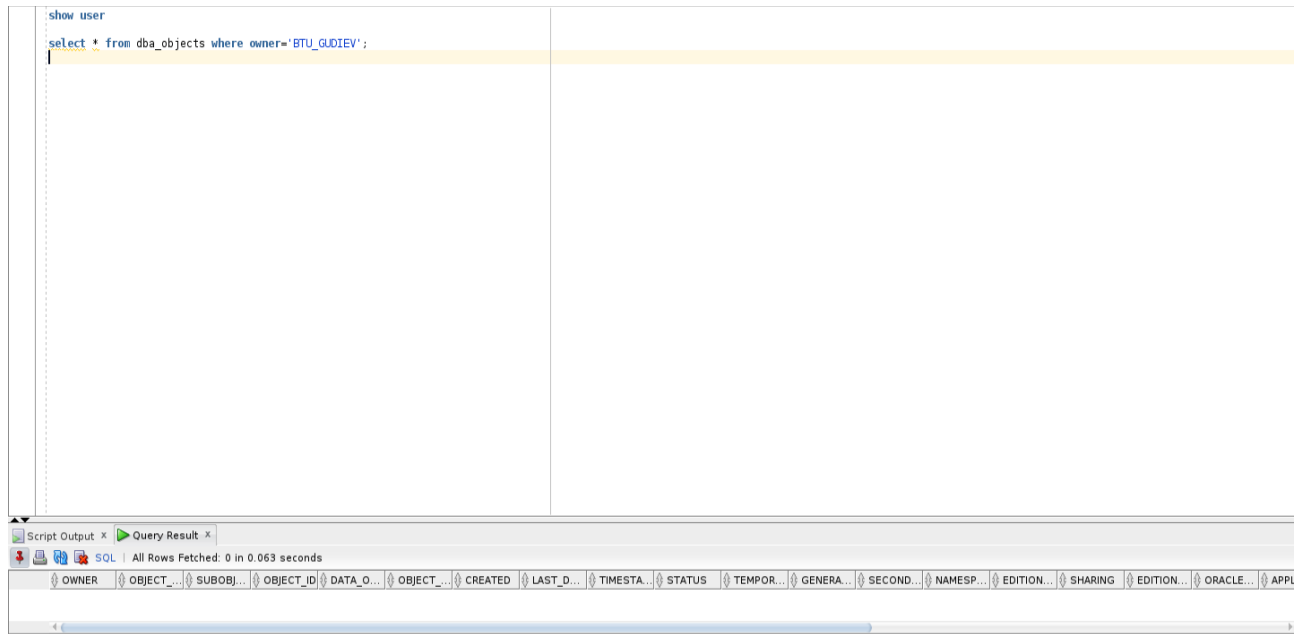
3. (1 ქულა) მიუერთდით ბაზას როგორც მომხმარებელი **btu_გვარი**. გადაამოწმეთ ვინაობა. გადაამოწმეთ თუ ფლობს რაიმე ობიექტს.

სკრიპტი:

Show user

Selevt * from dba_objects where owner ='BTU_GUDIEV'

სქრინი:

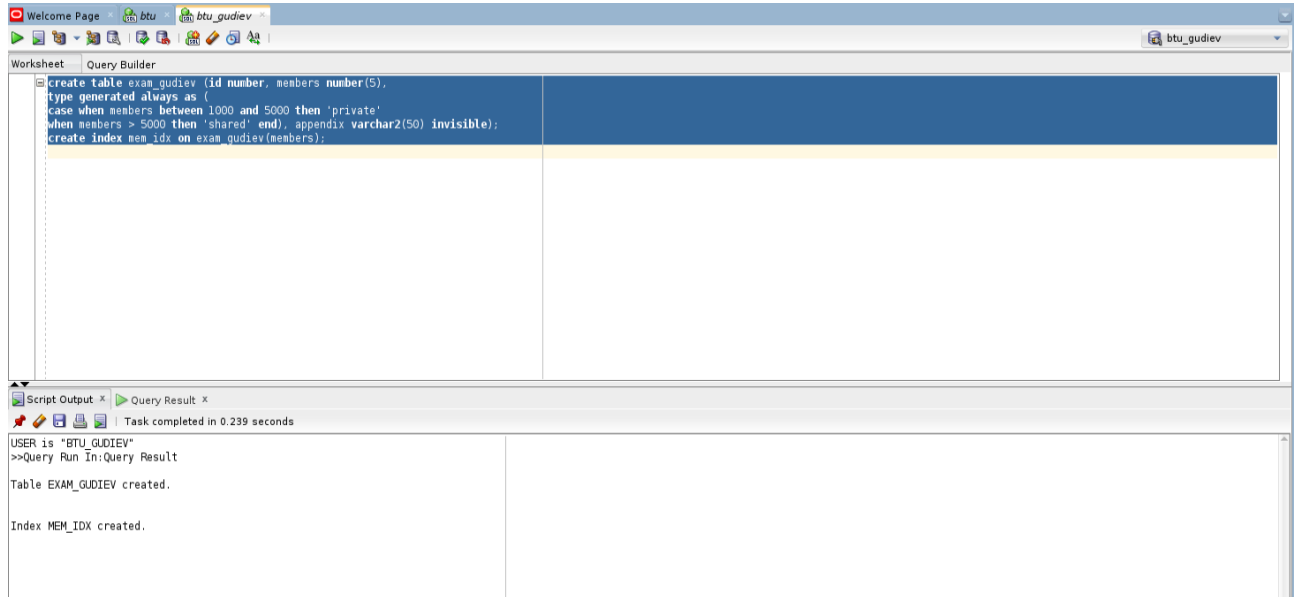


4. (2 ქულა) შექმენი ცხრილი **exam_გვარი** რომელსაც ექნება 4 სვეტი: პირველი id რიცხვითი, members რიცხვითი (5) და მე-3 ვირტუალური სვეტი type, რომელიც შეყვანილი წევრების მიხედვით დააგენერირებს სვეტის მნიშვნელობას თუ შეტანილი იქნება members-ში 1000-დან 5000-ის ჩათვლით private, თუ 5001-ზე მეტი share, მე-4 სვეტი appendix სიმბოლური მაქსიმუმ 50 სიმბოლო უჩინარი. შექმენი mem_idx ინდექსი members სვეტზე.

სკრიპტი:

```
create table exam_gudiev (id number, members number(5),  
type generated always as (  
case when members between 1000 and 5000 then 'private'  
when members > 5000 then 'shared' end), appendix varchar2(50) invisible);  
create index mem_idx on exam_gudiev(members);
```

სკრინი:

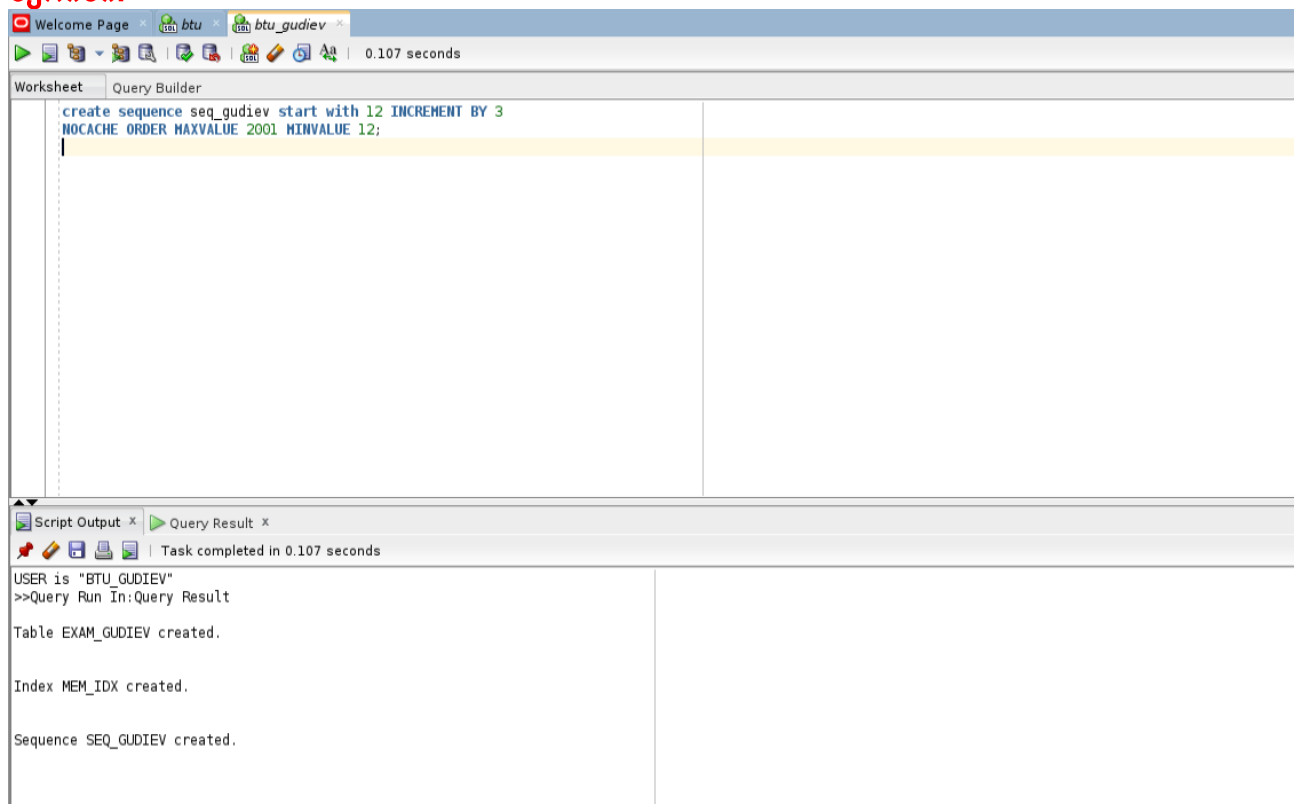


5. (1 ქულა) შექმენით მიმდევრობა **seq_გვარი** სახელით რომლის სტარტის წერტილია 12, ბიჯი 3, არ არის ციკლური, არის მოწესრიგებული, არ იყოს ქეშირებული, მაქსიმალური მნიშვნელობა 2001 და მინიმალური მნიშვნელობა 12.

სკრიპტი:

**create sequence seq_gudiev start with 12 INCREMENT BY 3
NOCACHE ORDER MAXVALUE 2001 MINVALUE 12;**

სკრინი:



6. (2 ქულა) შექმენი ტრიგერი exam_tr რომელიც ავტომატურ ზრდად id სვეტის მნიშვნელობებს დააგენერირებს ახალი ჩანაწერის შეტანისას exam_გვარი ცხრილში.

სკრიპტი:

```
create or replace trigger exam_tr
before insert on exam_gudiev
for each row
begin
select seq_gudiev.nextval into :new.id from dual;
```

end;

სკრინი:

```
create or replace trigger exam_tr
before insert on exam_gudiev
for each row
begin
select seq_gudiev.nextval into :new.id from dual;
end;
```

Script Output x Query Result x

Task completed in 0.319 seconds

USER is "BTU_GUDIEV"

>>Query Run In:Query Result

Table EXAM_GUDIEV created.

Index MEM_IDX created.

Sequence SEQ_GUDIEV created.

Trigger EXAM_TR compiled

7. (2 ქულა) შეიტანეთ test_გვარი ცხრილში შემდეგი ჩანაწერები members და appendix სვეტებში

2450, 'Tbilisi, btu is the best'

11000, 'GTU cool exam'

3150, 'GAU this day'

5580, 'SEU perfect thing'

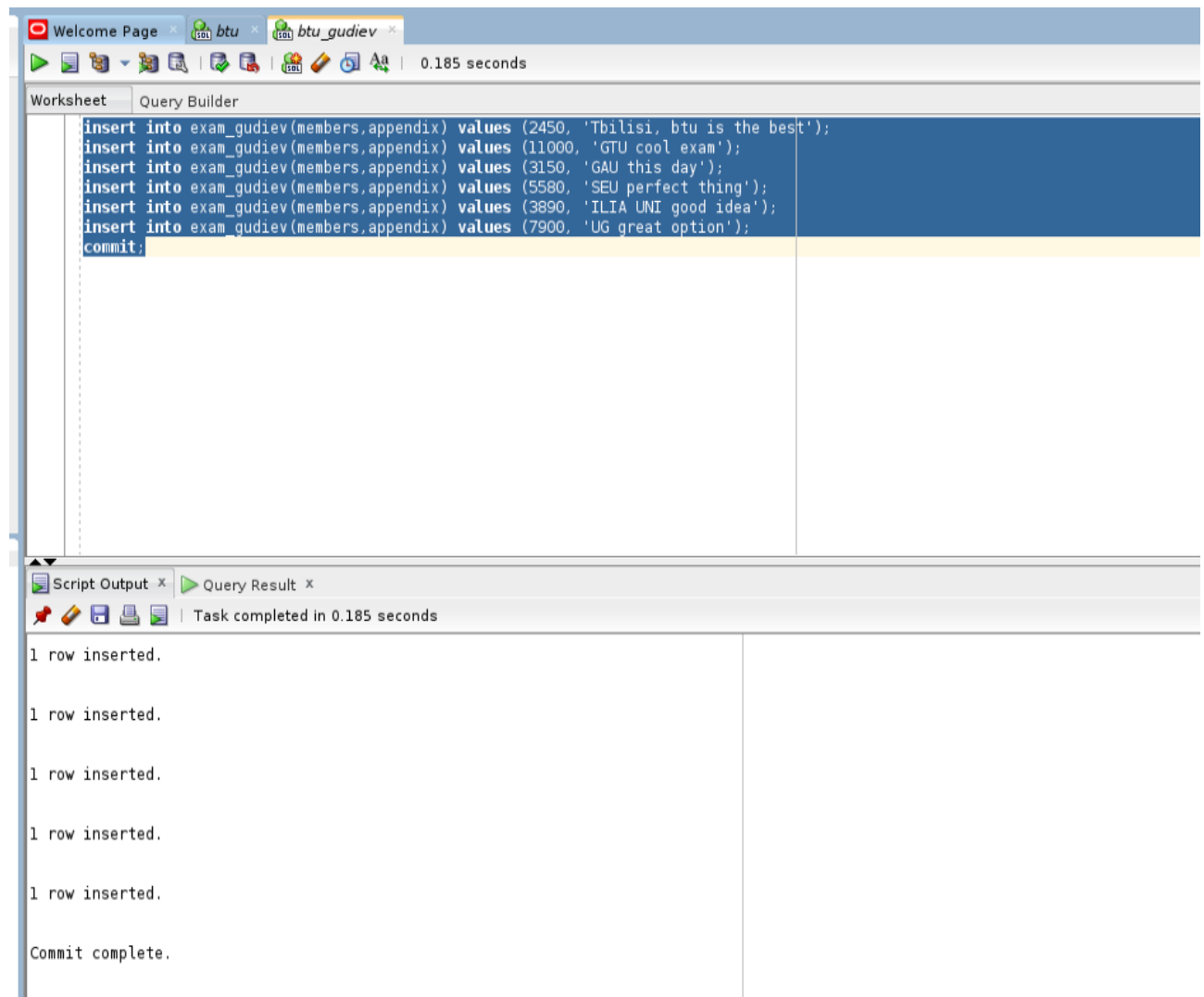
3890, 'ILIA UNI good idea'

7900, 'UG great option'

სკრიპტი:

```
insert into exam_gudiev(members,appendix) values (2450, 'Tbilisi, btu is the best');
insert into exam_gudiev(members,appendix) values (11000, 'GTU cool exam');
insert into exam_gudiev(members,appendix) values (3150, 'GAU this day');
insert into exam_gudiev(members,appendix) values (5580, 'SEU perfect thing');
insert into exam_gudiev(members,appendix) values (3890, 'ILIA UNI good idea');
insert into exam_gudiev(members,appendix) values (7900, 'UG great option');
commit;
```

სკრინი:



8. (2 ქულა) შექმენით ქვემოთხოვნიტ ცხრილი emp სქემა hr-ის employees-ზე დაყრდნობით, საიდანაც აიღება მხოლოდ ის ჩანაწერები სადაც გვარის სიგრძე არის 7-სიმბოლოზე ნაკლები. შექმენით ქვემოთხოვნიტ ცხრილი emp2 ცხრილ emp-ზე დაყრდნობით მხოლოდ კონსტრუქცია და არა ჩანაწერებიც.

სკრიპტი:

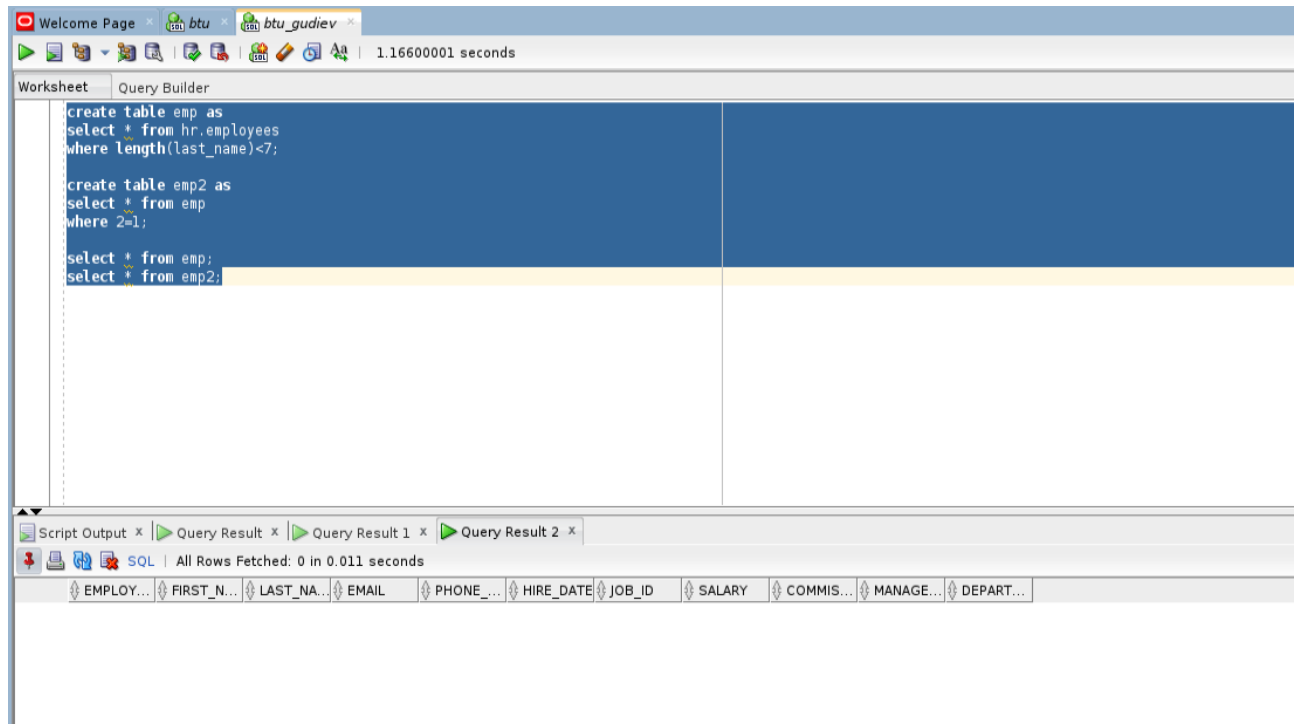
```
create table emp as
select * from hr.employees
where length(last_name)<7;
```



```
create table emp2 as
select * from emp
where 2=1;
```

```
select * from emp;
select * from emp2;
```

სკრინი:

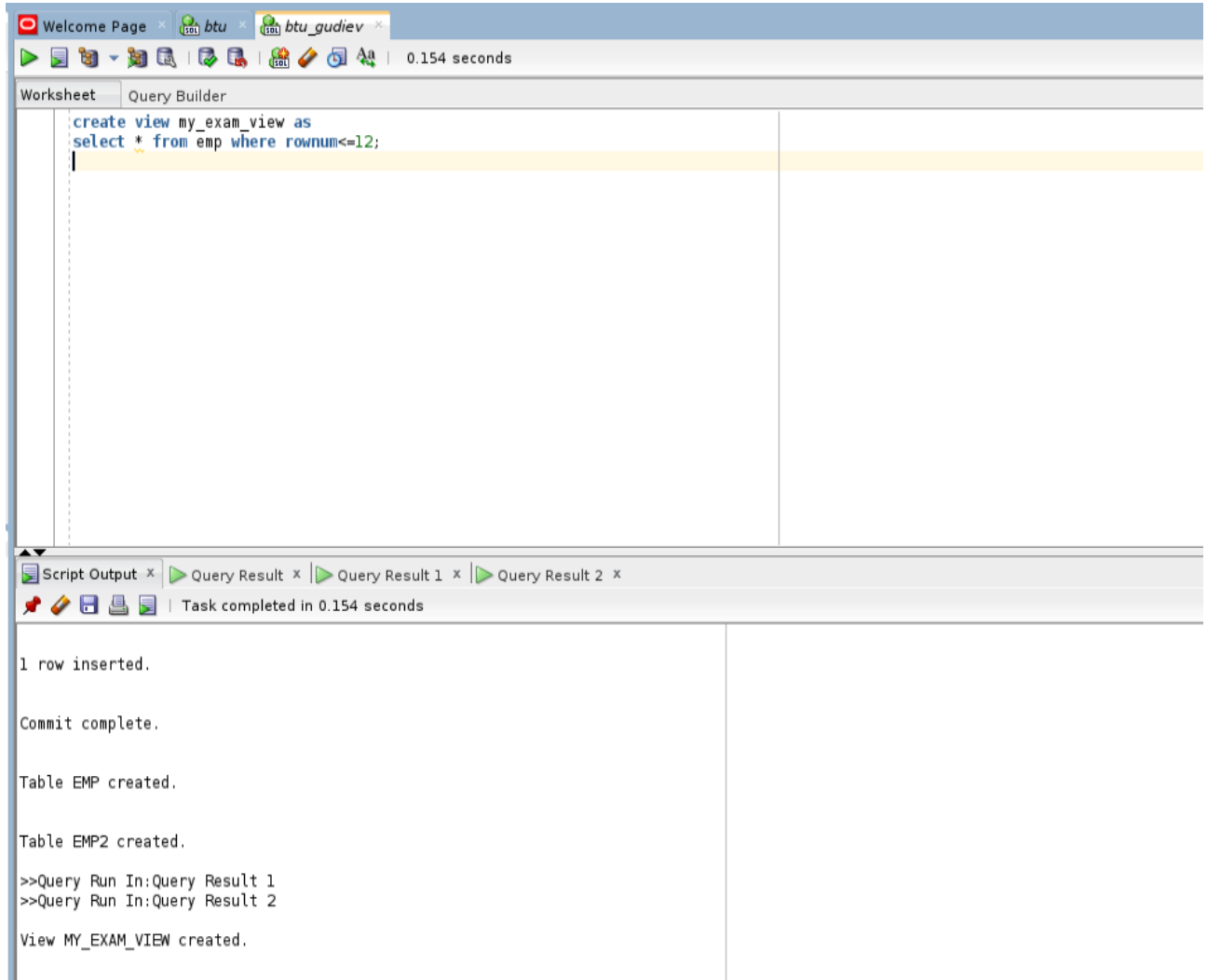


9. (2 ქულა) შექმენი **my_exam_view** ხედი emp ცხრილის პირველი 12 ჩანაწერით.

სკრიპტი:

```
create view my_exam_view as
select * from emp where rownum<=12;
```

სკრინი:



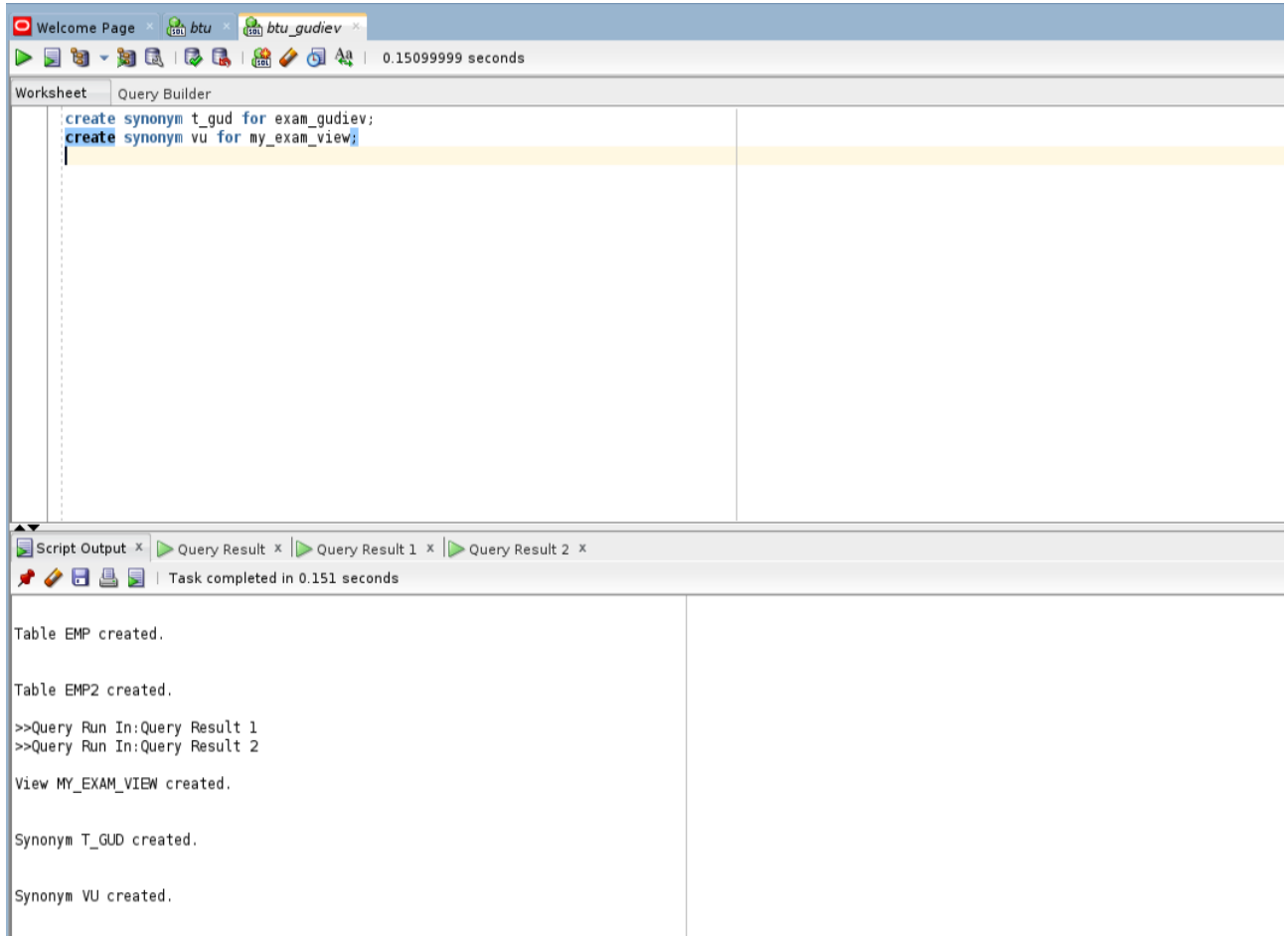
10. (2 ქულა) შექმენი სინონიმები **test_გვარი** ცხრილზე t_გვარის სამი ასო (t_ქართულად გვარის პირველი 3 ასო ლათინური შრიფტით) სახელით და **my_exam_view** ხედზე vu სახელით. გადაამაწმეთ თუ რა ობიექტებს ფლობს მომხმარებელი.

სკრიპტი:

create synonym t_gud for exam_gudiev;

create synonym vu for my_exam_view;

სკრინი:



11. (2 ქულა) a) შეიტანეთ ახალი ჩანაწერი ცხრილში **test_გვარი** 3000, 'my gtu forever'; b) დაამოდიფიცირეთ ჩანაწერი რომლის აიდიცა 21 appendix სვეტში არსებულს მიუკონკატენირეთ აიდი და members სვეტის მნიშვნელობა; c) წაშალეთ ის სტრიქონები რომლებშიც appendix სვეტის სიგრძე არის 3-ის ჯერადი; d) დაამატეთ სვეტი leader რიცხვითი; e) სვეტი leader გაზადეთ სიმბოლური მაქსიმალური 60 სიმბოლო; f) გადაარქვით სახელი leader სვეტს boss სახელით; g) ამოშალეთ სვეტი boss; h) გაზადეთ გამოუყენებელი appendix; i) გადაარქვით ცხრილს **test_გვარი** სახელი დაარქვით გვარის 5 ასო_tab; j) გაზადეთ ცხრილი მხოლოდ ამოკითხვადი.

სკრიპტი:

A) insert into exam_gudiev(members, appendix) values (3000, 'my gtu forever');

commit;

**b) update exam_gudiev set appendix=id||members
where id = 21;**

commit;

c)

სკრინი:

12. (2 ქულა) შექმენით ახალი პროფილი **exam_prof_გვარი** რომელსაც ექნება შემდეგი პარამეტრები

COMPOSITE_LIMIT	default
SESSIONS_PER_USER	70
CPU_PER_SESSION	unlimited
CPU_PER_CALL	unlimited
LOGICAL_READS_PER_SESSION	700000
LOGICAL_READS_PER_CALL	60000
IDLE_TIME	12
CONNECT_TIME	900
PRIVATE_SGA	default
FAILED_LOGIN_ATTEMPTS	30
PASSWORD_LIFE_TIME	300
PASSWORD_REUSE_TIME	240
PASSWORD_REUSE_MAX	18
PASSWORD_VERIFY_FUNCTION	null
PASSWORD_LOCK_TIME	10
PASSWORD_GRACE_TIME	30

სკრიპტი:

სქრინი:

13. (1 ქულა) დაამოდიფიცირეთ **btu_გვარი** იუზერის პარამეტრები პაროლი გახადეთ **btu23** და პროფილი **exam_prof_გვარი**.

სკრიპტი:

სქრინი:

14. (1 ქულა) გააქტიურე აუდიტი **btu_გვარი** იუზერის განხორციელებულ ნებისმიერ ბრძანებაზე. ნახე აუდიტის ლოგი. აუდიტი გააუქმე.

სკრიპტი:

სქრინი:

15. (1 ქულა) წაშალეთ იუზერი **btu_გვარი**. წაშალეთ როლი **გვარი_role**. წაშალეთ პროფილი **exam_prof_გვარი**. გადაამოწმეთ არსებობს თუ არა ეს იუზერი.

სკრიპტი:

სქრინი: