

DBMS Practice Lab Exercise

Roll No CB.EN.U4CSE18207

```
create table BusDriver(  
  driverid varchar2(4) ,  
  Drivename varchar2(30),  
  dob date,  
  age numeric,  
  doj date,  
  service number(6,2),  
  salary number(6,2),  
  constraint BusDriver_pk primary key (driverid)  
)
```

1) DECLARE

```
d_driverid BusDriver.driverid%type;  
d_Drivename BusDriver.Drivename%type;  
d_service BusDriver.service%type;  
d_doj BusDriver.doj%type;  
d_dob BusDriver.dob%type;  
d_age BusDriver.age%type;  
CURSOR c_busdriver is
```

```
SELECT driverid, Drivename,doj,dob,age,service FROM BusDriver;
```

BEGIN

OPEN c_busdriver;

LOOP

FETCH c_busdriver INTO d_driverid,
d_Drivername,d_doj,d_dob,d_service,d_age;

EXIT WHEN c_busdriver%notfound;

insert into BusDriver (driverid, Drivername, dob,doj) values (3,'sd','19-aug-
1992','20-dec-2002');

update BusDriver set age= 20 where driverid=3;

update BusDriver set service= 27 where driverid=3;

END LOOP;

CLOSE c_busdriver;

commit;

END;

select * from BusDriver;

DRIVERID	DRIVERNAME	DOB	AGE	DOJ	SERVICE	SALARY
3	sd	19-AUG-92	20	20-DEC-02	27	-

2)

```
CREATE OR REPLACE TRIGGER salary_change_monitoring
  BEFORE UPDATE OF salary ON BusDriver
  FOR EACH ROW
BEGIN
  IF (:new.salary < 1000
      OR :new.salary > 10000)
  THEN
    raise_application_error(-20002
      , 'Salary cannot be less than 1000 or greater than 10000.');
```

END IF;

END;

Update busdriver set salary = 2
where driverid = 1;

```
ORA-20002: Salary cannot be less than 1000 or greater than 10000. ORA-06512: at
"SQL_TGRINVPSVPRGLIZVFXHKHXUEV.SALARY_CHANGE_MONITORING", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

3) create or replace trigger busDriverChange
before update or delete on busDriver
begin

```
RAISE_APPLICATION_ERROR (-20002, 'You are not allowed to update or delete');
```

```
end;
```

```
update busdriver set salary=9000 where driverid=3;
```

```
Trigger created.
```

4) create table Dept

(

dno varchar2(5) not null,

depoName varchar2(10),

depoLocation varchar2(10),

constraint Dept_pk primary key(dno)

);

```
insert into dept values('D1','mm depo1','mumbai');
```

```
insert into dept values('D2','tt depo2','delhi');
```

create table Cleaner

(

cno varchar2(5) not null,

cname varchar2(20),

csalary number(6,2),

dno varchar2(5),

```

status varchar2(20),
constraint cleaner_pk primary key(cno),
constraint dept_fk foreign key(dno) references Dept(dno)
);

insert into Cleaner values ('C1','King',5000,'D1','Y');
insert into Cleaner values ('C2','Ram',4300,'D2','N');

```

DNO	DEPNAME	DEPOLOCATION
D1	mm depo1	mumbai
D2	tt depo2	delhi

CNO	CNAME	CSALARY	DNO	STATUS
C1	King	5000	D1	Y
C2	Ram	4300	D2	N

5) Create or replace function get_cleaners_location(cleanerNum in Cleaner.cno%type)

```

return dept.depoLocation %type as
dLocation dept.depoLocation %type;

begin
select depoLocation

```

```
into dLocation from Cleaner c, Dept d
where cNo= cleanerNum
and d.dNo=c.dNo;
return (dLocation);
end;
```

```
select cName, get_cleaners_location(cno) "Address"
from cleaner
where cno='C1';
```

CNAME	Address
King	mumbai