**Zhumagali Kanat IT2-2003**

**Lab6. Creating Triggers**

1. Create at least three triggers with the functionality according to the database subject area.
2. Demonstrate the results of triggers functioning.

1.

create or replace trigger movie\_aud\_trigger

after delete or update on movie

for each row

begin

if deleting then

insert into movie\_audit(pr\_movie\_id, pr\_film\_name, pr\_duration, pr\_year\_of\_release, pr\_rating\_imbdb)

values (:old.movie\_id, :old.film\_name, :old.duration, :old.year\_of\_release, :old.rating\_imbdb);

dbms\_output.put\_line('Deleted values saved in movie\_audit table');

elsif updating then

insert into movie\_audit(pr\_movie\_id, pr\_film\_name, pr\_duration, pr\_year\_of\_release, pr\_rating\_imbdb)

values(:old.movie\_id, :old.film\_name, :old.duration, :old.year\_of\_release, :old.rating\_imbdb);

dbms\_output.put\_line('Updated values saved in movie\_audit table');

end if;

end;

select \* from movie

select \* from movie\_audit

delete from movie where movie\_id = 300

insert into movie(movie\_id, film\_name, duration, year\_of\_release, rating\_imbdb)

values(300, 'the thing', '2:22', 1987, '9/10')

create table movie\_audit(

pr\_movie\_id number,

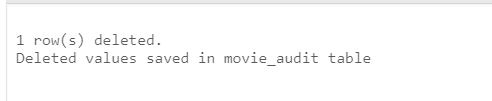
pr\_film\_name varchar2(40),

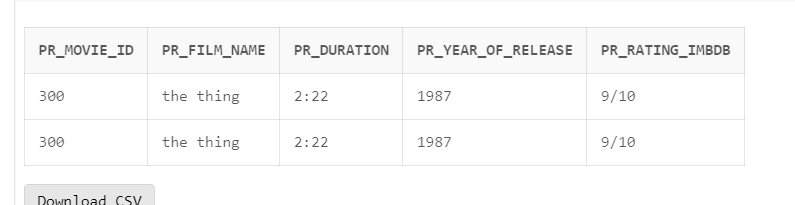
pr\_duration varchar2(20),

pr\_year\_of\_release number,

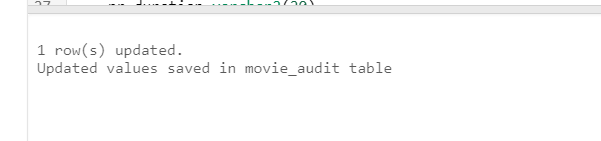
pr\_rating\_imbdb varchar2(10)

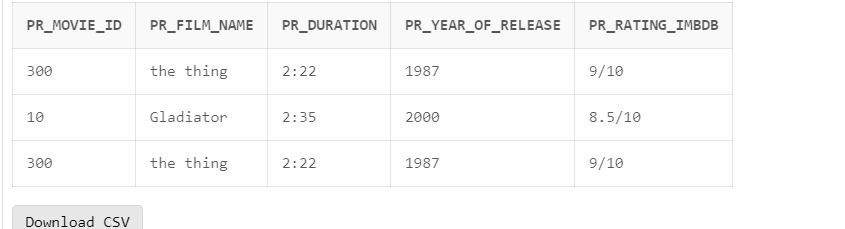
)





update movie set duration = '2:10' where movie\_id = 10





2.

create or replace trigger check\_duration\_rating\_format

before insert or update on movie

for each row

begin

if not regexp\_like(:new.duration, '^\d+:\d+$') is null then

raise\_application\_error(-20001, 'Invalid duration format. Please use the format Hourse:Minute');

end if;

if not regexp\_like(:new.rating\_imbdb, '^[1-9]/10$|^10/10$') then

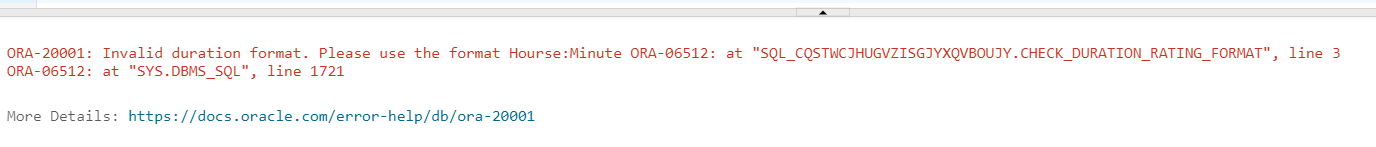
RAISE\_APPLICATION\_ERROR(-20001, 'Invalid rating\_imbdb format. Please use the format X/10, where X is between 1 and 10.');

end if;

end;

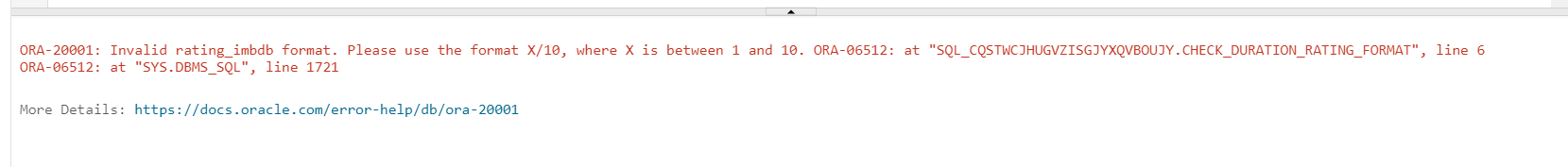
insert into movie(movie\_id, film\_name, duration, year\_of\_release, rating\_imbdb)

values(304, 'the thing', '2 hours 22 minutes', 1987, '10/10');



INSERT INTO movie(movie\_id, film\_name, duration, year\_of\_release, rating\_imbdb)

VALUES(305, 'the thing', '2:22', 1987, '12/10');



3.

alter table movie

add modification\_date TIMESTAMP;

create or replace trigger modification\_trigger

before update on movie

for each row

begin

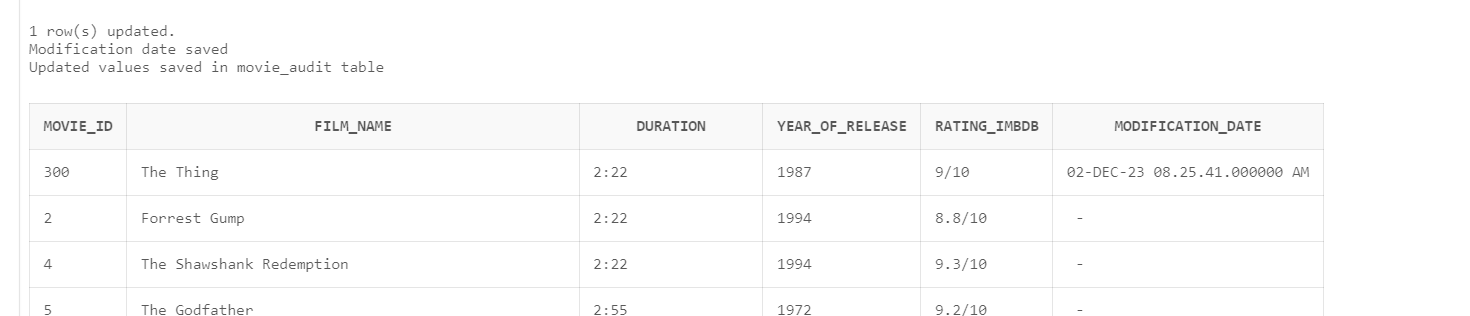
:new.modification\_date := TO\_CHAR(SYSDATE, 'DD-MON-YYYY HH24:MI:SS');

dbms\_output.put\_line('Modification date saved');

end;

update movie set film\_name = 'The Thing' where movie\_id = '300';

select \* from movie



Questions:

1. What are triggers for?
2. What are the types of triggers depending on the execution time?
3. What are the types of triggers depending on triggering events?
4. Describe statement level triggers with examples.
5. Describe row level triggers with examples.
6. What is the sequence of execution of statement and row triggers?
7. Describe **new** and **old** qualifiers, and their structure.
8. What is the section WHEN for?
9. How a trigger is created.
10. How the triggering events can be grouped?
11. When INSTEAD OF triggers are used?
12. How to create a disabled trigger?
13. How to enable or disable a trigger?
14. What is a mutating table?
15. What is a compound trigger?
16. In which sequence the sections in a compound trigger must be defined?
17. How to create a compound trigger?
18. For which database objects can a compound trigger be created?