**《数据库系统与应用》实验报告**

**——实验五**

|  |  |  |
| --- | --- | --- |
| contents：view and index  triggers | lab：10-409 | |
| Instructor: liuli | | |
| Stu\_id:2016329621073 | | Stu\_name:吴永昌 |
| Software Environment ： MYSQL/ | | |
| Purpose：   1. To understand database view and index, to understand triggers on the running database.   notation：  1．Defineview  CREATE VIEW <VIEW\_name> AS(  Select statement  )  2．Define index  CREATE INDEX <INDEXname> ON <table\_name>(  <list of elements>  );  3.Define trigger  CREATE TRIGGER trigger\_name  BEFORE|AFTER INSERT|DELETE|UPDATE ON tbl\_name  [referencing variables]  [FOR EACH ROW]  Action  Example:  CREATE TRIGGER moviedeletedtrigger  after delete on movies  for each row  Delete from starsin  where movietitle=old.title and movieyear=old.year | | |
|  | | |

Experiment contents:

Background description is shown in project1:

Schema for our running example:

Movies(title,year,length,movietype, studioname, producerC)

movieStar(name,address,gender,birthdate)

starsIn(movietitle,movieyear,starname)

Studio(name,address,presC)

Movieexec(name,address,cert,networth)

list your answer in the last page. Handing your answer file with the file name as Experiment5 \_ID.

part1: work on the exiting database “my\_own\_Movie\_database”;Write Create view statement with SQL to define the view based on the tables in the database ;

(1) A view MGMMovies giving the title,year,length,movietype,studioname( (the view involve the movies produced by MGM studio));

(2) A view studioPres giving the name,address,and certificate number of all executives who are studio presidents ;

(3) A view RichExec giving the name,address,cerificate number and net worth of all executives with a net worth of at least $10,000,000

(4) A view executivestar giving the name,address,gender,birthdate,certificate number,and networth of all individuals who are both executives and stars;

(5) A view MGMFemalSTAFF giving the title,year,starname,gender

Part 2:Write the SQL statement to query or modify on view.

(1) Find the names of females who are both stars and executive;

(2) Find the names of those executives who are both studio presidents and worth at least $10,000,000;

(3) Add a new tuple(movies) through the view MGMMovies;

(4) Remove the view MGMFemalSTARFF from databse

Then :copy the picture for each result.

Part 3:delcare indexes on the following attribures or combination of attributes.

(1) Studioname

(2) Movietype

(3) Address of moviestar

Part 4:

define four triggers so that they work as the same functions of the foreign key in the following:

CREATE TABLE starsin (

movieTitle char(100) ,

movieYear int(11) ,

starName char(30) ,

foreign key(movietitle,movieyear) references movies(title,year)

on delete cascade

on update cascade,

foreign key(starname) re

ferences moviestar(name)

on delete set null

on update cascade

);

Part 1:

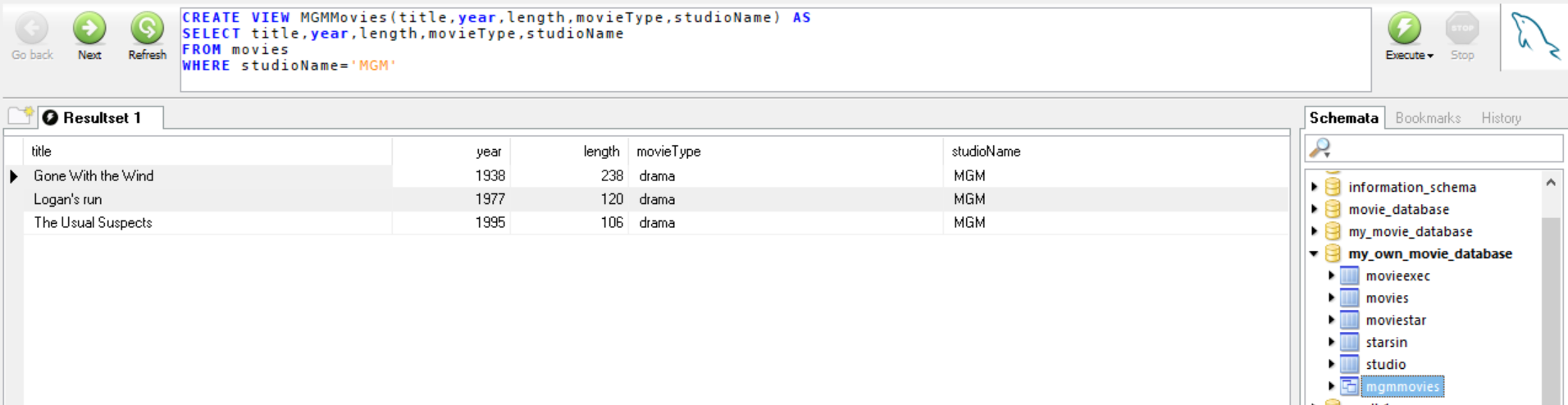
Create table statement including constraint

1. CREATE VIEW MGMMovies(title,year,length,movieType,studioName) AS

SELECT title,year,length,movieType,studioName

FROM movies

WHERE studioName='MGM'

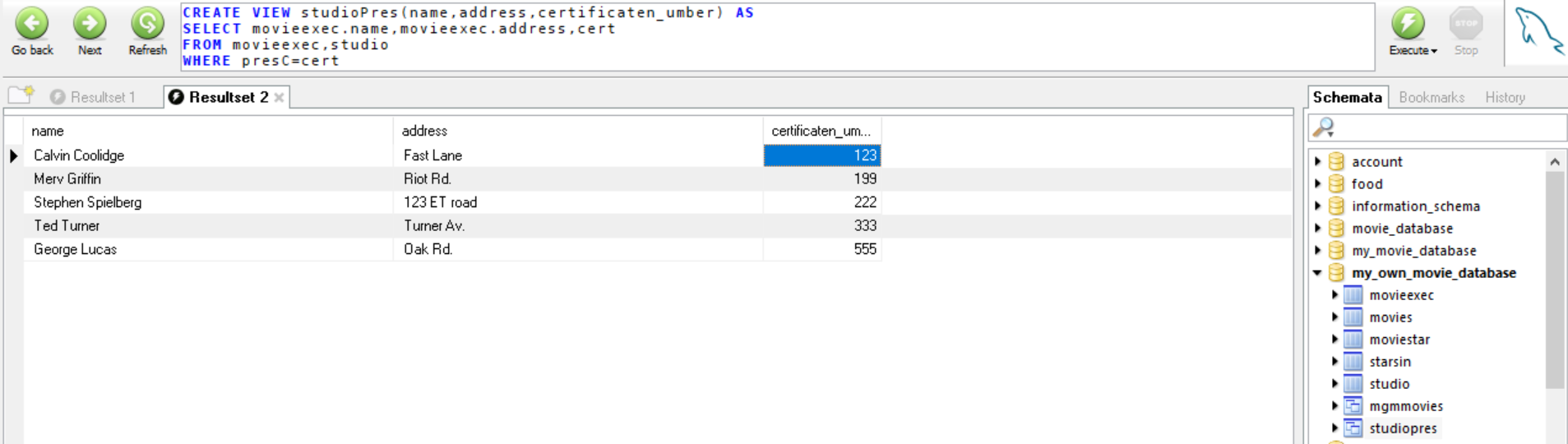


1. CREATE VIEW studioPres(name,address,certificaten\_umber) AS

SELECT movieexec.name,movieexec.address,cert

FROM movieexec,studio

WHERE presC=cert;

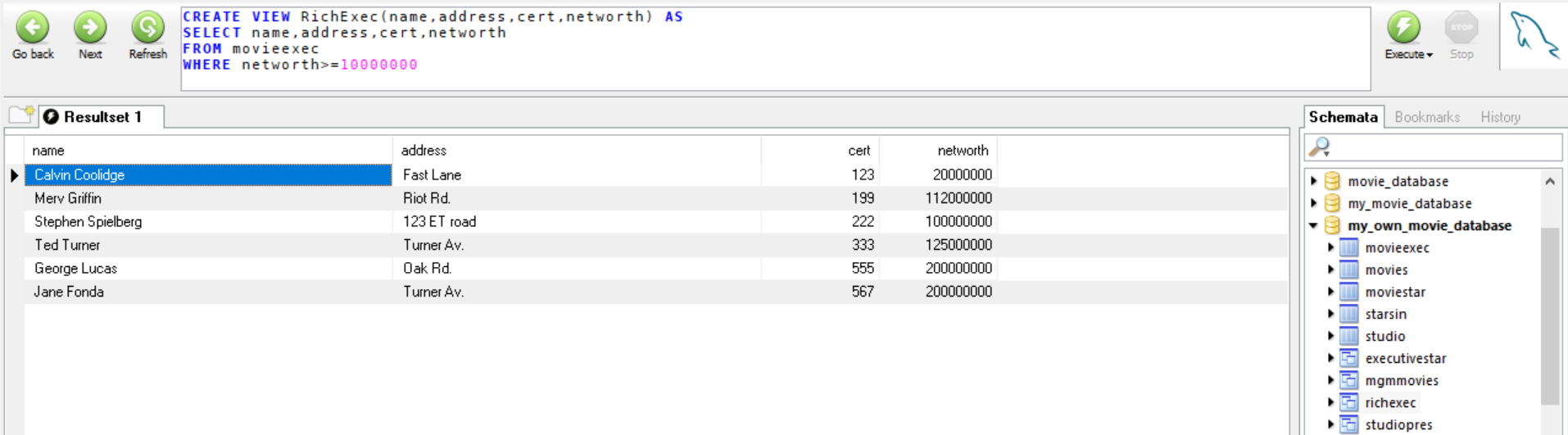


CREATE VIEW RichExec(name,address,cert,networth) AS

SELECT name,address,cert,networth

FROM movieexec

WHERE networth>=10000000

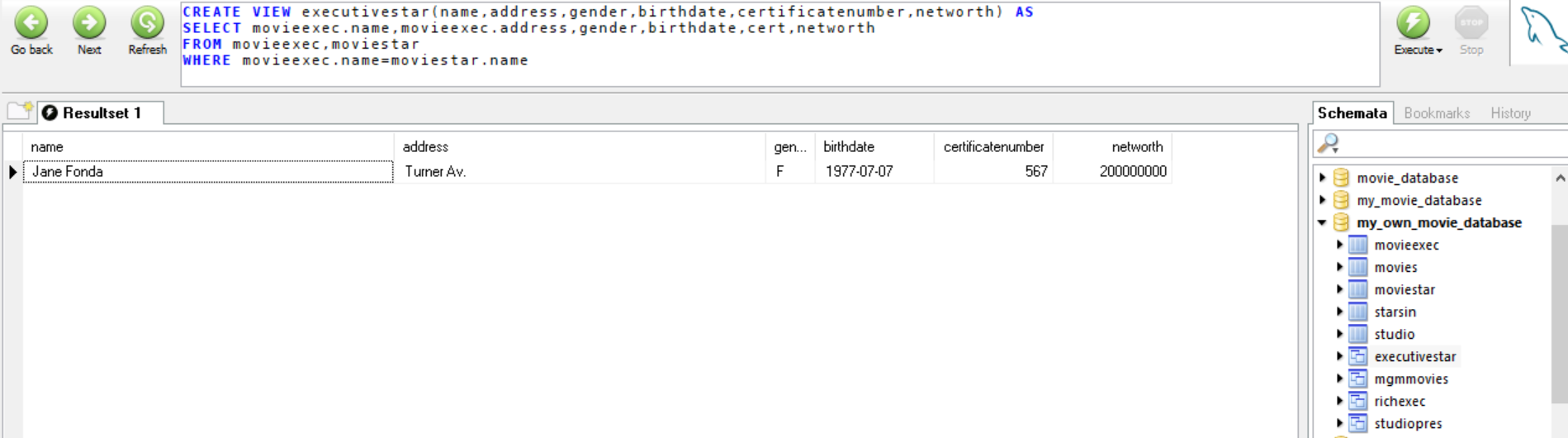


1. CREATE VIEW executivestar(name,address,gender,birthdate,certificatenumber,networth) AS

SELECT movieexec.name,movieexec.address,gender,birthdate,cert,networth

FROM movieexec,moviestar

WHERE movieexec.name=moviestar.name



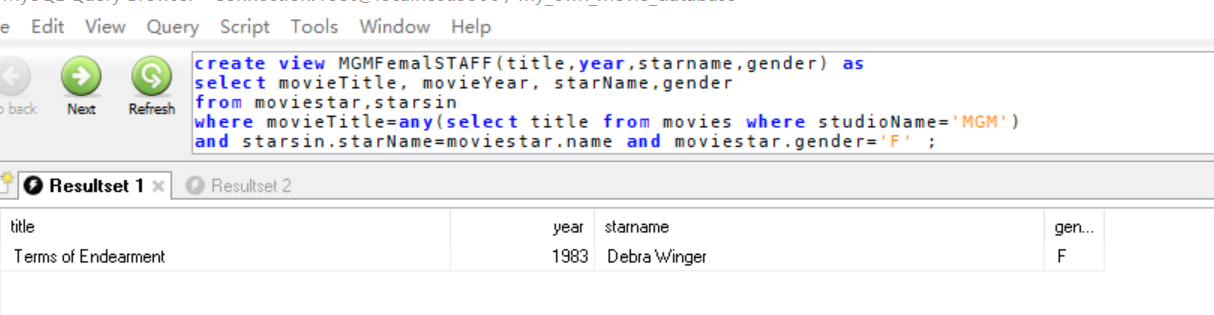
1. CREATE VIEW MGMFemalSTAFF(title,year,starname,gender) AS

SELECT movieTitle, movieYear, starName,gender

FROM moviestar,starsin

WHERE movieTitle=any(select title from movies where studioName='MGM')

and starsin.starName=moviestar.name and moviestar.gender='F' ;



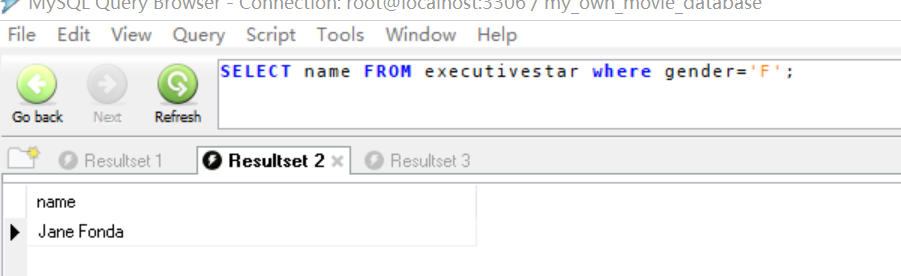
Part 2:

Pictures:

1. SELECT name

FROM executivestar

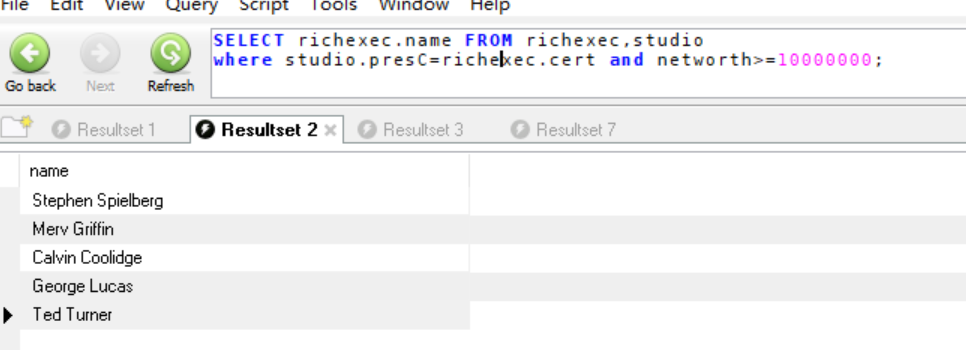
WHERE gender='F';



1. SELECT richexec.name

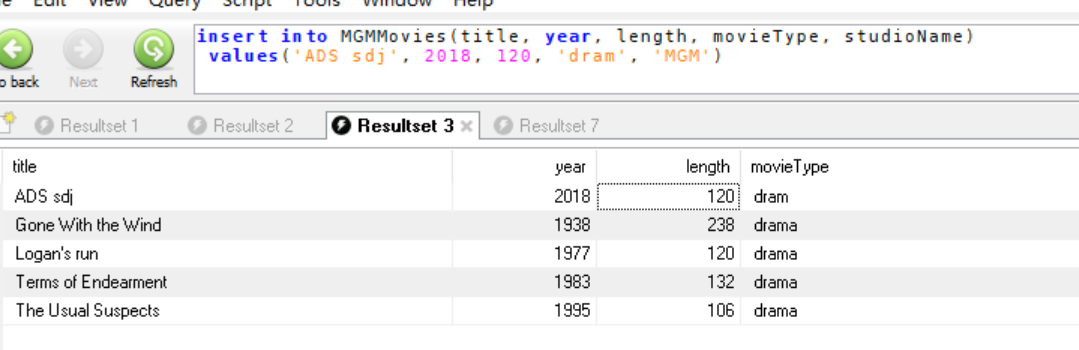
FROM richexec,studio

WHERE studio.presC=richexec.cert and networth>=10000000;

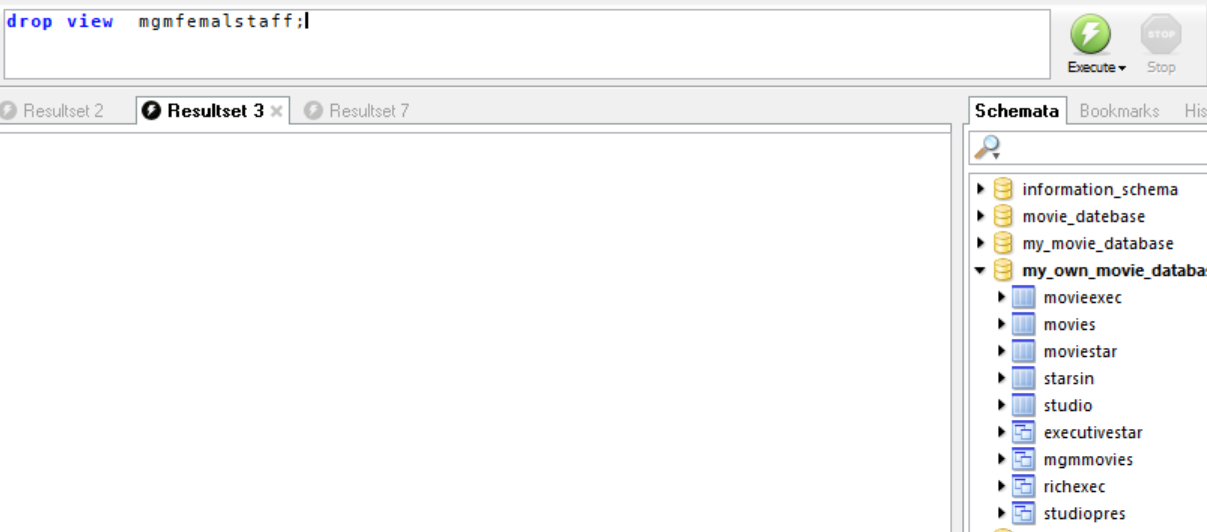


1. INSERT INTO MGMMovies(title, year, length, movieType, studioName)

values('ADS sdj', 2018, 120, 'dram', 'MGM')



1. DROP VIEW mgmfemalstaff;



Part 3:

1. CREATE INDEX studioname\_index ON movies(studioName);
2. CREATE INDEX movietype\_index ON movies(movieType);
3. CREATE INDEX address\_index ON moviestar(address);

Part 4:

foreign key(movietitle,movieyear) references movies(title,year)

on delete cascade:

1. CREATE trigger delete\_movie after delete on movies for

each row delete from starsin

WHERE movieTitle=OLD.title and movieYear=OLD.year; on update cascade:

1. CREATE trigger update\_movie after update on movies for each row update

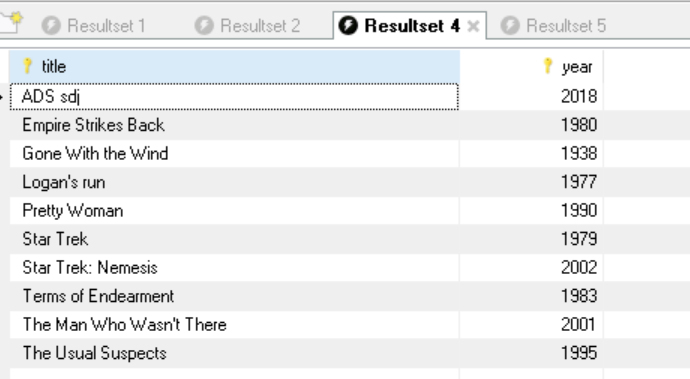
starsin set movieTitle = new.title and movieYear=new.year

WHERE movieTitle=OLD.title and movieYear=OLD.year;

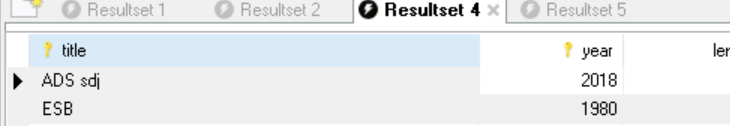
测试删除：

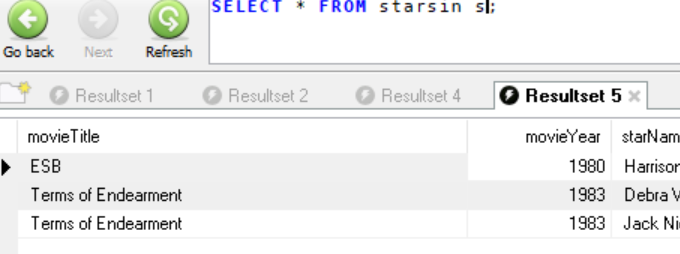
DELETE FROM movies

WHERE title='Star Wars' and year='1977'





测试更新：  




foreign key(starname) references moviestar(name)

on delete set null:

CREATE trigger de\_moviesetnull after delete on moviestar for each row update

starsin set starName = null

WHERE starName=OLD.name;  
on update cascade:  
 CREATE trigger update\_moviename after update on moviestar for each row update

starsin set starName = new.name

WHERE starName=OLD.name;

