Views

Q.1- create view TNS as select a.title,c.name,b.stars from Movie a,Rating b,Reviewer c where b.rID=c.rID and b.mID=a.mID;

select max(a.year) from Movie a,TNS b where a.title = b.title and b.name="Chris Jackson";

Q.2- create view RatingStats as

select distinct(title) as title,count(title) as no\_of\_ratings,avg(stars) as avg\_rating from TNS group by title;

select title from RatingStats where no\_of\_ratings>=3 order by avg\_rating desc limit 1;

Q.3-

CREATE OR REPLACE VIEW Favourites AS SELECT r.mID, r.rID,r.stars

FROM

(( Rating r INNER JOIN Movie m ON r.mID = m.mID) INNER JOIN Reviewer rv ON rv.rID = r.rID)

WHERE

r.stars = (SELECT MAX(Stars) FROM Rating WHERE rID = rv.rID)

GROUP BY

r.mID, r.rID,r.stars;

// Query3

SELECT \* from ((( Favourites f INNER JOIN Movie m on m.mID = f.mID) INNER JOIN Reviewer rv1 on rv1.rID = f.rID) INNER JOIN Reviewer rv2 on rv2.rID = f.rID);

//Query 3 test

SELECT DISTINCT CASE

WHEN re1.name > re2.name THEN re2.name

ELSE re1.name

END AS "Reviewer 1",

CASE

WHEN re1.name < re2.name THEN re2.name

ELSE re1.name

END AS "Reviewer 2",

mo.title AS "Movie"

FROM

Reviewer re1,

Reviewer re2,

Movie mo

WHERE re1.rID IN ( SELECT rID FROM Favourites WHERE mID = mo.mID) AND re2.rID IN ( SELECT rID FROM Favourites WHERE mID = mo.mID) AND re1.rID != re2.rID;

Stored Procedure

Q-1

delimiter $$

Create procedure usp\_get\_employees\_salary\_above(in salary\_above int)

begin

select Fname, Lname from employee where Salary>=salary\_above

order by Fname, Lname ASC ;

end $$

delimiter ;

call usp\_get\_employees\_salary\_above(40000);

Q-2

delimiter $$

create procedure usp\_get\_towns\_starting\_with(in start\_string varchar(20))

begin

select Pname from project where Pname like concat(start\_string, ‘%’)

order by Pname ASC;

end $$

delimiter ;

call usp\_get\_towns\_starting\_with(“Product”);