

Relations:

MOVIE(TITLE,DIR,YEAR,BUDGET,LANGUAGE)  
CAST(TITLE,ACTOR)  
REVIEW(CRITIC,TITLE,SCORE)

Projection:  $\pi$

Sigma:  $\sigma$

Join:  $\bowtie$

Union:  $\cup \cap$

\*\*\*

brackets placed in RA solutions should be interpreted as a subscript  
ex.  $\text{Movie} \bowtie [\text{movie.title}=\text{review.title}] \text{Review}$   
relation join on title of each table

\*\*\*

1.1: Find all the movies with maximum existing scores.

SQL-

```
Select Movie.title, Review.score
From movie
inner join Review
on Movie.title=Review.TITLE
where Review.SCORE in (select s1.score from review s1
where not exists
(select * from Review s2
where s2.score >s1.score));
```

RA -

```
p1 := Movie  $\bowtie$  [movie.title=review.title] Review
p2 :=  $\pi[t1]$  (p1(t1,s1)  $\bowtie$  [s1,s2] p1(t2,s2))
Result := p2 -  $\pi[t]$ p1
```

1.2: Find all the critics that reviewed all the movies directed by Spielberg, possibly except "E.T.".

SQL -

```
select Review.critic as reviewed_Spielberg, review.TITLE
from review
inner join movie
on review.title=movie.title
where movie.dir='Spielberg';
```

RA -

```
p1 := movie ⋈ [movie.title=review.title] review
S_movies :=  $\sigma$ [dir='Spielberg' and title<>'E.T']
Result :=  $\pi$ [critic](S_movies)
```

1.3:

For every director, list the director and the number of movies in French made by this director. If a director has made no such movies, then the director should be listed with zero. The result should be sorted in descending order.

SQL -

```
select dir, Count("language")
from movie
where "language" in
(Select "language" from movie where "language"='French')
group by dir
Order by dir desc;
```

1.4: Find all the actors whose average movie budget was greater than the average budget of Tom Cruise's movies.

SQL-

```
Create view Actors_Avg_budget As
Select "cast".actor, avg(movie.budget) as avgB
from "cast" inner join movie on "cast".title=movie.title
group by "cast".ACTOR;
Create view TomCruise_Avg_budget as
select actor as tom, AVGB as tomB from ACTORS_AVG_BUDGET
where actor='Tom Cruise';
Select actor as greaterBudget from ACTORS_AVG_BUDGET
inner join TOMCRUISE_AVG_BUDGET on
ACTORS_AVG_BUDGET.AVGB>TOMCRUISE_AVG_BUDGET.TOMB;
```

2.1: List all the critics, each with their average movie score.

SQL-

```
select critic, avg(score)
from review
group by CRITIC;
```