**SQL DATABASE QUESTIONS AND ANSWERS**

**1.**Which actors have the first name ‘Scarlett’

**Query:** **select \* from actor**

**where first\_name='Scarlett'**

**2.**Which actors have the last name ‘Johansson’

**Query: select \* from actor**

**where last\_name='Johansson'**

**3.**How many distinct actors last names are there?

**Query: select distinct last\_name**

**from actor**

**4.**Which last names are not repeated?

**Query: select last\_name from actor**

**group by last\_name**

**having count(last\_name) = 1**

**5**.Which last names appear more than once?

**Query: select last\_name from actor**

**group by last\_name having count(last\_name >1)**

**6**.Which actor has appeared in the most films?

**Query: select count(film\_actor.actor\_id),**

**actor.first\_name, actor.last\_name from actor inner join film\_actor**

**on actor.actor\_id = film\_actor.actor\_id**

**group by film\_actor.actor\_id ;**

**7**.Is ‘Academy Dinosaur’ available for rent from Store 1?

**Query:** **select inventory.inventory\_id from inventory join store using (store\_id) join film using (film\_id)**

**join rental using (inventory\_id) where film.title = 'Academy Dinosaur' and store.store\_id = 1**

**and not exists (select \* from rental where rental.inventory\_id = inventory.inventory\_id and rental.return\_date is null);**

**8.**Insert a record to represent Mary Smith renting ‘Academy Dinosaur’ from Mike Hillyer at Store 1 today .

**Query: insert into rental (rental\_date, inventory\_id, customer\_id, staff\_id)values (now(), 1, 1, 1);**

**9.**When is ‘Academy Dinosaur’ due?

**Query: Answer: select rental\_date,rental\_date + interval(select rental\_duration from film where film\_id = 1) day**

**as due\_date from rental where rental\_id = (select rental\_id from rental order by rental\_id desc limit 1);**

**10.**What is that average running time of all the films in the sakila DB?

**Query:** **select avg(length) as 'Average time of film' from film**

**11.**What is the average running time of films by category?

**Query: select name,avg(length)**

**from film join film\_category using (film\_id) join category using (category\_id) group by name**

**order by avg(length) desc;**

**12.**Why does this query return the empty set?

select \* from film natural join inventory;

**Answer:**