پاسخنامه تکلیف پنجم درس پایگاه داده ها ۱

ترم پاییز ۱۳۹۹

1.a)

```
select id,name,dept_name,salary, lag(salary,1) over(partition by dept_name
order by salary) first_smaller_salary
from instructor;
```

1.b)

```
select id,name,tot_cred,dense_rank() over(order by tot_cred desc) std_rank
from student;
```

---

2)

```
create table Turn_Over(
                    Dep_id integer,
                    Trn_Time timestamp,
                    Trn_over integer);
insert into Turn_Over
values (1022,'2018-06-15 14:00'::timestamp, 100),
            (1022,'2018-06-15 14:28'::timestamp, -50),
            (1022,'2018-06-16 14:58'::timestamp, 25),
            (1067,'2019-07-18 23:32'::timestamp, 300);

select Dep_id,Trn_Time,Trn_over,sum(Trn_Over) over
(partition by dep_id order by trn_time rows between unbounded preceding
and current row) balance
from Turn_Over
```

3.a)

```
select customer_id,payment_id, payment_date,amount,
            sum(amount) over(partition by customer_id order by
payment_date rows between unbounded preceding and current row),
            avg(amount) over(partition by customer_id)
from payment
```

3.b)

```
with T as
(select customer_id, sum(amount) sum_of_payment
from payment
group by customer_id),
T1 as
(select customer_id,
      ntile(4) over(order by sum_of_payment) tile
from T)
select customer_id,first_name, last_name,tile
from T1 inner join customer using (customer_id)
where tile = 1;
```

3.c)

```
select p1.customer_id,p1.payment_date::date,sum(p2.amount)
from payment p1 inner join payment p2 on
p1.customer_id = p2.customer_id and p1.payment_date::date >=
p2.payment_date::date
and p1.payment_id >= p2.payment_id
group by p1.customer_id,p1.payment_date::date,p1.payment_id
order by customer_id,payment_date
```

3.d)

```
select country, city,
```

```
        count(distinct customer_id) cust_count,
        count(distinct rental_id) order_count
from rental join customer using(customer_id)
        join address using(address_id)
        join city using(city_id)
        join country using(country_id)
group by (country_id, city_id);
```

3.e)

```
select rental_rate, category_id,
        count(distinct film_id) film_count
from film join film_category using(film_id)
group by rollup(rental_rate,category_id);
```

3.f)

```
select payment_date::date, city_id, sum(amount)
from payment join rental using(rental_id, customer_id)
        join customer using(customer_id)
        join address using(address_id)
        join city using(city_id)
group by cube(payment_date::date, city_id);
```