«به نام خدا»

تکلیف پنجم DB1 - مرضیه علیدادی - 9631983

R1a)

```
1  WITH ordered AS(
2      SELECT id, name, dept_name, salary
3      FROM instructor
4      ORDER BY salary)
5  SELECT id, name, dept_name, salary,
6         LAG(salary,1) OVER (PARTITION BY dept_name ORDER BY salary) previous_salary
7  FROM ordered;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	id [PK] character varying (5)	name character varying (20)	dept_name character varying (20)	salary numeric (8,2)	previous_salary numeric
1	14365	Lembr	Accounting	32241.56	[null]
2	57180	Hau	Accounting	43966.29	32241.56
3	79081	Ullman	Accounting	47307.10	43966.29
4	31955	Moreira	Accounting	71351.42	47307.10
5	43779	Romero	Astronomy	79070.08	[null]
6	41930	Tung	Athletics	50482.03	[null]
7	4034	Murata	Athletics	61387.56	50482.03
8	15347	Bawa	Athletics	72140.88	61387.56
9	16807	Yazdi	Athletics	98333.65	72140.88
10	63287	Jaekel	Athletics	103146.87	98333.65
11	80759	Queiroz	Biology	45538.32	[null]
12	81991	Valtchev	Biology	77036.18	45538.32
13	3335	Bourrier	Comp. Sci.	80797.83	[null]
14	34175	Bondi	Comp. Sci.	115469.11	80797.83
15	65931	Pimenta	Cybernetics	79866.95	[null]
16	99052	Dale	Cybernetics	93348.83	79866.95

R1b)

Query History Query Editor

```
1  select id, name, rank() over (order by tot_cred DESC) s_rank
2  from student
3  order by s_rank ASC;
```

Messages Scratch Pad Notifications Explain Data Output

	id [PK] character varying (5)	name character varying (20)	s_rank bigint
1	15328	Chien	1
2	93004	Gibbs	1
3	14581	Vagn	1
4	20803	Mercurio	1
5	75560	Tabor	1
6	71025	Cadis	1
7	61354	Barranco	1
8	72657	Hird	1
9	14214	Yoneda	1
10	33645	Kawakami	1
11	26427	Ende	1
12	82301	Conti	1
13	38476	Rzecz	1
14	56058	Fettes	14
15	99660	OMalley	14
16	10527	Kieras	14
17	23525	DAgostino	14
18	96117	Nisso	14
19	66229	Qvi	14

R2)

جدول Turn_over و اطلاعات ذخیره شده در آن:

Query History Query Editor

```
1  create table Turn_over(
2      dep_id integer,
3      trn_time timestamp,
4      trn_over integer
5  );
```

Messages Scratch Pad Notifications Explain Data Output

```
CREATE TABLE

Query returned successfully in 79 ms
```

✔ Query returned successfully in 79 msec.

```
1  select * from Turn_over;
2
3
```

Messages Scratch Pad Notifications Explain Data Output

	dep_id integer	trn_time timestamp without time zone	trn_over integer
1	1000	2020-12-09 19:47:47.717966	1000
2	1001	2020-12-09 19:48:33.52311	100
3	1002	2020-12-09 19:49:02.19195	-20
4	1000	2020-12-09 19:49:31.907106	-20
5	1002	2020-12-09 19:49:59.264902	1000
6	1001	2020-12-09 19:50:42.798923	80
7	1001	2020-12-09 19:51:02.665298	-500

جدول factdeptrn :

```
1  create table factdeptrn(
2      dep_id integer,
3      trn_time timestamp,
4      trn_over integer,
5      balance integer
6  );
```

Messages | Scratch Pad | Notifications | Explain | Data Output

```
CREATE TABLE

Query returned successfully in 214
```

✔ Query returned successfully in 214 msec.

Query History | Query Editor

```
1  insert into factdeptrn(select *
2                         from (select dep_id, trn_time, trn_over,
3                                      sum(trn_over) over (order by trn_time
4                                                          range between unbounded preceding and current
5                         from Turn_over) as Tmp);
6
```

Messages | Scratch Pad | Notifications | Explain | Data Output

```
INSERT 0 7

Query returned successfully in 65 msec.
```

✔ Query returned successfully in 65 msec.

```
1  select * from factdeptrn
2
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	dep_id integer	trn_time timestamp without time zone	trn_over integer	balance integer
1	1000	2020-12-09 19:47:47.717966	1000	1000
2	1001	2020-12-09 19:48:33.52311	100	1100
3	1002	2020-12-09 19:49:02.19195	-20	1080
4	1000	2020-12-09 19:49:31.907106	-20	1060
5	1002	2020-12-09 19:49:59.264902	1000	2060
6	1001	2020-12-09 19:50:42.798923	80	2140
7	1001	2020-12-09 19:51:02.665298	-500	1640

R3a)

با فرض اینکه، هم میانگین و هم مجموع، به تفکیک مشتری، از ابتدا تا پس از آن سفارش مد نظر است:

```
1  select distinct payment_id, customer_id, amount, payment_date,
2         avg(amount) over (partition by customer_id order by payment_date range between unbounded preceding and current row),
3         sum(amount) over (partition by customer_id order by payment_date range between unbounded preceding and current row)
4  from payment
5  order by customer_id, payment_date;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	payment_id [PK] integer	customer_id smallint	amount numeric (5,2)	payment_date timestamp without time zone	avg numeric	sum numeric
1	18495	1	5.99	2007-02-14 23:22:38.996577	5.9900000000000000	5.99
2	18496	1	0.99	2007-02-15 16:31:19.996577	3.4900000000000000	6.98
3	18497	1	9.99	2007-02-15 19:37:12.996577	5.6566666666666667	16.97
4	18498	1	4.99	2007-02-16 13:47:23.996577	5.4900000000000000	21.96
5	18499	1	4.99	2007-02-18 07:10:14.996577	5.3900000000000000	26.95
6	18500	1	0.99	2007-02-18 12:02:25.996577	4.6566666666666667	27.94
7	18501	1	3.99	2007-02-21 04:53:11.996577	4.5614285714285714	31.93
8	22680	1	4.99	2007-03-01 07:19:30.996577	4.6150000000000000	36.92
9	22681	1	3.99	2007-03-02 14:05:18.996577	4.5455555555555556	40.91
10	22682	1	0.99	2007-03-02 16:30:04.996577	4.1900000000000000	41.90
11	22683	1	4.99	2007-03-17 11:06:20.996577	4.2627272727272727	46.89
12	22684	1	0.99	2007-03-18 02:25:55.996577	3.9900000000000000	47.88
13	22685	1	0.99	2007-03-19 08:23:42.996577	3.7592307692307692	48.87
14	22686	1	2.99	2007-03-19 12:25:20.996577	3.7042857142857143	51.86
15	22687	1	0.99	2007-03-21 22:02:23.996577	3.5233333333333333	52.85
16	22688	1	1.99	2007-03-21 23:56:23.996577	3.4275000000000000	54.84
17	22689	1	2.99	2007-03-22 18:10:03.996577	3.4017647058823529	57.83

R3b)

```sql
with c_pay_sum as(
    select distinct customer_id,
           sum(amount) over (partition by customer_id
                             order by payment_date
                             range between unbounded preceding and unbounded following) as pay_sum
    from payment),
     c_pay_ntile as(
    select customer_id, pay_sum,
           ntile(4) over (order by pay_sum DESC) as quartile
    from c_pay_sum)
select customer_id, pay_sum
from c_pay_ntile
where quartile = 1;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	customer_id smallint	pay_sum numeric
1	148	211.55
2	526	208.58
3	178	194.61
4	137	191.62
5	144	189.60
6	459	183.63
7	181	167.67
8	410	167.62
9	236	166.61
10	403	162.67
11	522	161.68

R3c)

```
with c_pay(payment_id, customer_id, amount, payment_date) as(
    select distinct payment_id, customer_id, amount, payment_date
    from payment
    order by customer_id, payment_date),
     c_pre_pay(payment_id, customer_id, amount, payment_date,
               pre_payment_id, pre_customer_id, pre_amount, pre_payment_date) as(
    select distinct payment.payment_id, payment.customer_id, payment.amount, payment.payment_date,
           c_pay.payment_id, c_pay.customer_id, c_pay.amount, c_pay.payment_date
    from payment, c_pay
    where c_pay.payment_date <= payment.payment_date
          and c_pay.customer_id = payment.customer_id)
select c_pre_pay.payment_id, c_pre_pay.customer_id, c_pre_pay.amount,
       c_pre_pay.payment_date, sum(c_pre_pay.pre_amount)
from c_pre_pay
group by c_pre_pay.payment_id, c_pre_pay.customer_id, c_pre_pay.amount, c_pre_pay.payment_date
order by c_pre_pay.customer_id, c_pre_pay.payment_date;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	payment_id [PK] integer	customer_id smallint	amount numeric (5,2)	payment_date timestamp without time zone	sum numeric
1	18495	1	5.99	2007-02-14 23:22:38.996577	5.99
2	18496	1	0.99	2007-02-15 16:31:19.996577	6.98
3	18497	1	9.99	2007-02-15 19:37:12.996577	16.97
4	18498	1	4.99	2007-02-16 13:47:23.996577	21.96
5	18499	1	4.99	2007-02-18 07:10:14.996577	26.95
6	18500	1	0.99	2007-02-18 12:02:25.996577	27.94
7	18501	1	3.99	2007-02-21 04:53:11.996577	31.93
8	22680	1	4.99	2007-03-01 07:19:30.996577	36.92
9	22681	1	3.99	2007-03-02 14:05:18.996577	40.91
10	22682	1	0.99	2007-03-02 16:30:04.996577	41.90
11	22683	1	4.99	2007-03-17 11:06:20.996577	46.89
12	22684	1	0.99	2007-03-18 02:25:55.996577	47.88
13	22685	1	0.99	2007-03-19 08:23:42.996577	48.87

R3d)

```
1  select distinct country_id, city_id,
2         count(distinct customer_id) customer_num, count(distinct rental_id) rental_num
3  from rental inner join customer using(customer_id)
4       inner join address using(address_id)
5       inner join city using(city_id)
6       inner join country using(country_id)
7  group by cube(city_id, country_id)
8  order by country_id, city_id;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	country_id integer	city_id integer	customer_num bigint	rental_num bigint
1	1	251	1	18
2	1	[null]	1	18
3	2	59	1	28
4	2	63	1	25
5	2	483	1	37
6	2	[null]	3	90
7	3	516	1	20
8	3	[null]	1	20
9	4	67	1	25
10	4	360	1	27
11	4	[null]	2	52
12	5	493	1	35
13	5	[null]	1	35
14	6	20	1	25
15	6	43	1	30
16	6	45	1	28

R3e)

```
1  select rental_rate, category_id, count(film_id) as film_number
2  from film_category inner join film using(film_id)
3  group by rollup(rental_rate, category_id)
4  order by rental_rate, category_id;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	rental_rate numeric (4,2)	category_id smallint	film_number bigint
1	0.99	1	28
2	0.99	2	23
3	0.99	3	21
4	0.99	4	22
5	0.99	5	16
6	0.99	6	29
7	0.99	7	23
8	0.99	8	26
9	0.99	9	21
10	0.99	10	19
11	0.99	11	20
12	0.99	12	20
13	0.99	13	20
14	0.99	14	15
15	0.99	15	22
16	0.99	16	16
17	0.99	[null]	341
18	2.99	1	19
19	2.99	2	26

R3f)

```
1  select city, payment_date,sum(amount) tot_amount
2  from payment inner join customer using(customer_id)
3       inner join address using(address_id)
4            inner join city using(city_id)
5  group by cube(city, payment_date)
6  order by city, payment_date;
```

Messages | Scratch Pad | Notifications | Explain | Data Output

	city character varying (50)	payment_date timestamp without time zone	tot_amount numeric
1	A Corua (La Corua)	2007-02-15 00:06:57.996577	4.99
2	A Corua (La Corua)	2007-02-18 02:22:57.996577	0.99
3	A Corua (La Corua)	2007-02-19 22:15:50.996577	2.99
4	A Corua (La Corua)	2007-02-20 22:30:54.996577	4.99
5	A Corua (La Corua)	2007-03-01 12:40:55.996577	0.99
6	A Corua (La Corua)	2007-03-01 14:06:24.996577	0.99
7	A Corua (La Corua)	2007-03-02 05:36:33.996577	0.99
8	A Corua (La Corua)	2007-03-17 06:53:01.996577	3.99
9	A Corua (La Corua)	2007-03-18 00:40:59.996577	2.99
10	A Corua (La Corua)	2007-03-18 12:11:11.996577	0.99
11	A Corua (La Corua)	2007-03-19 09:08:20.996577	5.99
12	A Corua (La Corua)	2007-03-21 23:47:03.996577	2.99
13	A Corua (La Corua)	2007-03-21 23:49:40.996577	5.99
14	A Corua (La Corua)	2007-03-22 05:04:16.996577	6.99
15	A Corua (La Corua)	2007-03-23 15:28:38.996577	5.99
16	A Corua (La Corua)	2007-04-06 22:15:18.996577	1.99
17	A Corua (La Corua)	2007-04-09 14:27:04.996577	0.99