NYU Tandon School of Engineering
Computer Science and Engineering
CS6083, Spring 2020

Feb. 28, 2019

Problem Set #1 Sample Solution

Problem 1

(a)

1

```
with
time_consuming as (
      select cid, cname, sum(endtime - starttime) as t
      from Customer left join Watch using(cid)
      group by cid
)
select cid, cname
from time_consuming
where t = (select max(t) from time_consuming);
```

2

```
select cid
from Watch
group by cid
having count(distinct mid) = (select count(*) from Movie);
```

3

```
select distinct mid
from Watch
where inLastWeek(endtime) or inLastWeek(starttime);
```

4

```
select cid
from Watch natural join Customer natural join Movie
where (endtime – starttime) < 10min and mtitle = "The Piano"

-- people watched movie only once and the duration is less than 10min (better solution)
with
people_watched_the_piano_once as (
      select cid, (endtime - starttime) as t
      from Watch natural join Customer natural join Movie
      where mtitle ="The Piano"
      group by cid
      having count(*) = 1
)
select *
from the_piano_watch_history
where t < 10min;
```

5

```sql
select mid, mtitle
from Movie left join Watch using(mid)
where cid is null;
```

6

```sql
with
average_rate as (
     select mid, avg(score) as rate
     from Rating
     group by mid
)
select mid, mtitle
from average_rate natural join Movie
where rate = (select max(rate) from average_rate)
```

7

```sql
with
people_score_five_to_the_piano as (
    select cid
    from Rating natural join Movie
    where score = 5 and mtitle = "The piano"
),
movie_average_rate_except_the_piano as (
    select mid, avg(score) as rate
    from Rating natural join people_score_five_to_the_piano
    where mid != (select Movie.mid from Movie where mtitle = "The piano")
    group by mid
)
select mid
from movie_average_rate_except_the_piano
where rate = (select max(rate) from movie_average_rate_except_the_piano);
```

(b)

1

$TotalTime \leftarrow {}_{cid,cname}\mathcal{G}_{sum(endtime-startime)\ as\ t}(Customer \bowtie Watch)$
$MaxTime \leftarrow \mathcal{G}_{max(t)\ as\ mt} TotalTime$
$\pi_{cid,cname}(\sigma_{t=mt} TotalTime \bowtie MaxTime)$

2

$\pi_{cid}(Watch \div Movie)$

3

$\pi_{mid}(\sigma_{now()-starttime \leq 7*24*3600} Watch)$

4

$\pi_{cid}(\sigma_{mtitle="The\ Piano" \wedge endtime-starttime<10*60}(Watch \bowtie Movie))$

5

$\pi_{mid,mtitle}\big(Movie \bowtie (\sigma_{mid}Movie - \sigma_{mid}Watch)\big)$

6

$AvgRate \leftarrow {}_{mid}\mathcal{G}_{avg(score)\ as\ ar}Rating$
$MaxAvgRate \leftarrow \mathcal{G}_{max(ar)\ as\ mar}AvgRate$
$\pi_{mid,mtitle}\big(\sigma_{ar=mar}(Movie \bowtie AvgRate \times MaxAvgRate)\big)$

7

$PeopleScoreFiveToThePiano \leftarrow \pi_{cid}\big(\sigma_{mtitle="The\ Piano" \wedge score=5}(Rating \bowtie Movie)\big)$
$MovieAverageRateExceptThePiano$
$\leftarrow {}_{mid}\mathcal{G}_{avg(score)\ as\ ar}\big(\sigma_{mtitle \neq "The\ Piano"}(PeopleScoreFiveToThePiano \bowtie Rating \bowtie Movie)\big)$
$MaxAvgRate \leftarrow \mathcal{G}_{max(ar)\ as\ mar}MovieAverageRateExceptThePiano$
$\pi_{mid}\big(\sigma_{ar=mar}(MovieAverageRateExceptThePiano \times MaxAvgRate)\big)$

(c)

1

Cannot be expressed with Relational Calculus as it requires aggregate functions.

2

$\{res | \forall m \in Movie(\exists w \in Watch(res[cid] = w[cid] \wedge m[mid] = w[mid]))\}$

3

$\{res | \exists w \in Watch(res[mid] = w[mid] \wedge now - w[starttime] \leq 7 * 24 * 3600\}$

4

$\{res | \exists w \in Watch(res[cid] = w[mid] \wedge w[endtime] - w[starttime] < 10 * 60 \wedge \exists m \in Movie(w[mid] = m[mid] \wedge m[mtitle] = "The\ Piano"))\}$

5

$\{res | \exists m \in Movie(res[mid] = m[mid] \wedge res[mtitle] = m[mtitle] \wedge \neg \exists w \in Watch(m[mid] = w[mid]))\}$

Problem 2

(a)

Customer(<u>cid</u>, cname, caddress, creditCard)
Product(<u>pid</u>, pname, description, pprice)
Subscribe(<u>cid, pid</u>, frequencyYear, frequencyMonth)
Order(<u>orderId</u>, pid, cid, deliver_time, deliver_price, number, changeDeliver)
Discount(<u>number</u>, discount)

Subscribe pid refer to Product pid
Subscribe cid refer to Customer cid
Order pid refer to Product pid

Order cid refer to Customer cid
Order number refer to Discount number

//changeDeliver is the month number the customer change the deliver time for his subscribed product. For example, the customer want to deliver the product by each 6 month, so we should deliver the product at June and December, but for the December deliver, the customer change the deliver time to November, so now the deliver_time in Order table for this order is November, the changeDeliver is -1

(b)
i.

```
select s.cid, (s.frequencyYear+ year(o.deliver_time)) as NextDeliveryYear, (s.frequencyMonth + month(o.deliver_time)) as NextDeliveryMonth
from Subscribe s natural join Order o
where s.cid = 7184995
order by o.deliver_time descending
limit 1
```

//We get the next deliver time by adding the deliver interval to the latest product deliver time

ii.

```
WITH user_subscribe(cid, name, number) as (
        select c.cid, c.name, count(*) as number
        from Customer c, Subscribe s
        where c.cid= s.cid group by c.cid
)

select uo.cid, up.name
from user_subscribe as uo
where uo.number = (select max(number) from user_subscribe)
```

iii.

```
select cid, cname
from Customer natural Join Product natural Join subscribe natural Join Order
where pname= “Acme Rocket Powered Roller Skates” and month(deliver_time) = 3 and changeDeliver = -1 and year(deliver_time)= 2020
```

iv.

```
select pid, count(*) as totalNumber, sum(deliver_price* discount) as totalMoney
from Order natural join Discount
where year(deliver_time) = 2019 group by pid
```