1. 아래와 같이 Table 생성,Data를 입력한 후 질문에 대한 SQL을 최소 각각 3개씩 작성하시요!!!!

drop table cust\_status;  
create table cust\_status  
 (cust\_id      char(1)      not null,  
  cust\_id\_seq  number       not null,  
  status       varchar2(10) not null)  
 ;

insert into cust\_status values ('A',1,'정상');  
insert into cust\_status values ('A',2,'위험');  
insert into cust\_status values ('B',1,'정상');  
insert into cust\_status values ('B',2,'정상');  
insert into cust\_status values ('C',1,'위험');  
insert into cust\_status values ('C',2,'위험');  
insert into cust\_status values ('D',1,'위험');  
insert into cust\_status values ('D',2,'위험');  
insert into cust\_status values ('D',3,'정상');  
insert into cust\_status values ('E',1,'정상');  
commit;

Q1>  cust\_id 별로 status 값이  한 종류만 가진  cust\_id 만 출력

select a.cust\_id

from CUST\_STATUS a,(select cust\_id,

count(\*)aa

from CUST\_STATUS

group by cust\_id,

status

order by cust\_id)b

where a.CUST\_ID=b.cust\_id

group by a.CUST\_ID

having count(\*)= max(cust\_id\_seq)

order by cust\_id;

Q2> ﻿ cust\_id 별로 status 값이  한 종류만 가진  Row 전체를 출력

select a.cust\_id,

a.CUST\_id\_seq,

a.status

from CUST\_STATUS a,

(select a.cust\_id

from CUST\_STATUS a,

(select cust\_id,

count(\*)aa

from CUST\_STATUS

group by cust\_id,

status

order by cust\_id)b

where a.CUST\_ID=b.cust\_id

group by a.CUST\_ID

having count(\*)= max(cust\_id\_seq))c

where a.cust\_id = c.cust\_id;

2. 아래와 같은 Table을 생성 ,Data를 입력한 후 20190101~20191231까지 총 365건 이 조회되도록 일자별집계를 구하시요.

TABLE명 : REPAY\_TEST

create table repay\_test (repay\_date date,detr\_nm varchar(100),rbno char(20),loan\_bal\_amt number(20));

drop table repay\_test;

insert into repay\_test(repay\_date,detr\_nm,rbno,loan\_bal\_amt) values('20190103','홍길동','1234567-1234567',1500000);

insert into repay\_test(repay\_date,detr\_nm,rbno,loan\_bal\_amt) values('20190906','홍길동','1234567-1234567',1000000);

insert into repay\_test(repay\_date,detr\_nm,rbno,loan\_bal\_amt) values('20190909','홍길동','1234567-1234567',500000);

**1) 상환 후 잔액 거래 내역**

|  |  |  |  |
| --- | --- | --- | --- |
| **REPAY\_DATE** | **DETR\_NM** | **RBNO** | **LOAN\_BAL\_AMT** |
| 20190103 | 홍길동 | 1234567-1234567 | 1500000 |
| 20190906 | 홍길동 | 1234567-1234567 | 1000000 |
| 20190909 | 홍길동 | 1234567-1234567 | 500000 |

**2) 상환 후 잔액 일별 집계**

|  |  |  |  |
| --- | --- | --- | --- |
| **TOT\_DATE** | **DETR\_NM** | **RBNO** | **LOAN\_BAL\_AMT** |
| 20190101 | 홍길동 | 1234567-1234567 | 0 |
| 20190102 | 홍길동 | 1234567-1234567 | 0 |
| 20190103 | 홍길동 | 1234567-1234567 | 1500000 |
| 20190104 | 홍길동 | 1234567-1234567 | 1500000 |
| 20190105 | 홍길동 | 1234567-1234567 | 1500000 |
| 20190106 | 홍길동 | 1234567-1234567 | 1500000 |
| 상동 | 상동 | 상동 | 상동 |
| 20190905 | 홍길동 | 1234567-1234567 | 1500000 |
| 20190906 | 홍길동 | 1234567-1234567 | 1000000 |
| 20190907 | 홍길동 | 1234567-1234567 | 1000000 |
| 20190908 | 홍길동 | 1234567-1234567 | 1000000 |
| 20190909 | 홍길동 | 1234567-1234567 | 500000 |
| 20190910 | 홍길동 | 1234567-1234567 | 500000 |
| 20190911 | 홍길동 | 1234567-1234567 | 500000 |
| 상동 | 상동 | 상동 | 상동 |

select distinct tot\_date,

a.detr\_nm,

a.rbno,

case

when tot\_date < first\_value(REPAY\_date)over() then 0

when tot\_Date >= REPAY\_date then loan\_bal\_amt

end loan\_bal\_amt

from (select repay\_date,

loan\_bal\_amt,

detr\_nm,

rbno,

lead(REPAY\_date)over(

order by REPAY\_date)r\_date,

first\_value(REPAY\_date)over()f\_date,

last\_value(REPAY\_date)over()l\_date,

last\_value(loan\_bal\_amt)over()l\_amt

from repay\_test)a,

(select to\_char(to\_date(:1-1)+level, 'yyyymmdd')tot\_Date

from dual connect by level<=:2)

where tot\_date between REPAY\_date and r\_Date-1

or tot\_Date<f\_Date or tot\_Date > l\_date and loan\_bal\_amt = l\_amt

order by tot\_date;