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**

**SQL :**

**1.**

SELECT CUST\_ID

FROM (SELECT CUST\_ID

FROM CUST\_STATUS

GROUP BY CUST\_ID,

STATUS)

GROUP BY CUST\_ID

HAVING COUNT(\*) =1;

**2.**

SELECT CUST\_ID

FROM (SELECT CUST\_ID,

COUNT(\*) CNT

FROM (SELECT CUST\_ID,

STATUS

FROM CUST\_STATUS

GROUP BY CUST\_ID,

STATUS )

GROUP BY CUST\_ID

ORDER BY CUST\_ID)

WHERE CNT =1;

**3.**

SELECT CUST\_ID

FROM (SELECT CUST\_ID,

ROW\_NUMBER() OVER(PARTITION BY CUST\_ID

ORDER BY STATUS) ID\_ST\_NUM

FROM CUST\_STATUS

GROUP BY CUST\_ID,

STATUS) A

GROUP BY CUST\_ID

HAVING MAX(A.ID\_ST\_NUM) = 1;

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

**SQL :**

**1.**

SELECT CUST\_ID, CUST\_ID\_SEQ, STATUS

FROM (SELECT CUST\_ID,

CUST\_ID\_SEQ,

STATUS,

COUNT(\*) OVER(PARTITION BY CUST\_ID) ID\_CNT,

COUNT(\*) OVER(PARTITION BY CUST\_ID, STATUS) ID\_STA

FROM CUST\_STATUS ) A

WHERE A.ID\_CNT = A.ID\_STA;

**2.**

SELECT CUST\_ID,

CUST\_ID\_SEQ,

STATUS

FROM CUST\_STATUS

WHERE CUST\_ID NOT IN(SELECT CUST\_ID

FROM (SELECT CUST\_ID,

STATUS,

ROW\_NUMBER() OVER(PARTITION BY CUST\_ID

ORDER BY STATUS) NUM

FROM CUST\_STATUS

GROUP BY CUST\_ID,

STATUS )

WHERE NUM !=1)

GROUP BY CUST\_ID ,

CUST\_ID\_SEQ,

STATUS

ORDER BY CUST\_ID;

**3.**

SELECT \*

FROM CUST\_STATUS

WHERE CUST\_ID NOT IN (SELECT CUST\_ID

FROM (SELECT CUST\_ID,

LEAD(STATUS, 1, STATUS) OVER(PARTITION BY CUST\_ID

ORDER BY CUST\_ID) LEAD\_STATUS ,

STATUS

FROM CUST\_STATUS

GROUP BY CUST\_ID,

STATUS,

CUST\_ID\_SEQ)

WHERE STATUS != LEAD\_STATUS)

ORDER BY CUST\_ID,

CUST\_ID\_SEQ;

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

TABLE명 : REPAY\_TEST

**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 |

**SQL : ( :1 = 2019/01/01 DATE :2 = 2019/12/31 DATE )**

SELECT DISTINCT TOT\_DATE,

DETR\_NM,

RBNO,

LOAN\_BAL\_AMT

FROM (SELECT TOT\_DATE,

DETR\_NM,

RBNO,

CASE

WHEN TOT\_DATE BETWEEN REPAY\_DATE AND ED THEN LOAN\_BAL\_AMT

WHEN TOT\_DATE < ST\_REPAY\_DATE THEN 0

ELSE NULL

END LOAN\_BAL\_AMT

FROM (SELECT TO\_CHAR( :1 - 1 + LEVEL, 'YYYYMMDD' ) AS TOT\_DATE

FROM DUAL CONNECT BY LEVEL <= (:2 - :1) + 1) B,

(SELECT REPAY\_DATE,

DECODE(LEAD(REPAY\_DATE) OVER(

ORDER BY REPAY\_DATE), NULL, :2, LEAD(REPAY\_DATE) OVER(

ORDER BY REPAY\_DATE)-1 ) ED,

RBNO,

LOAN\_BAL\_AMT,

DETR\_NM,

:1 SD,

CASE

WHEN ROW\_NUMBER() OVER(

ORDER BY REPAY\_DATE)= 1 THEN REPAY\_DATE

ELSE NULL

END ST\_REPAY\_DATE

FROM REPAY\_TEST CONNECT BY LEVEL <2

ORDER BY REPAY\_DATE ) A

WHERE REPAY\_DATE BETWEEN :1 AND :2

ORDER BY REPAY\_DATE )

WHERE LOAN\_BAL\_AMT IS NOT NULL

ORDER BY TOT\_DATE ;