<https://www.youtube.com/watch?v=SVfdmY9H1o0>

CREATE TABLE ORDERS (ORDER\_DATE DATE , ORDER\_ID VARCHAR2(10),PRODUCT\_ID VARCHAR2(10) ,QUANTITY INT , PRICE INT);

INSERT INTO ORDERS VALUES ('02-JUL-2011', 'O10' ,'P2' ,4,10);

01-JUL-11 O1 P1 5 5

01-JUL-11 O2 P2 2 10

01-JUL-11 O3 P3 10 25

01-JUL-11 O4 P1 20 5

02-JUL-11 O5 P3 5 25

02-JUL-11 O6 P4 6 20

02-JUL-11 O7 P1 2 5

02-JUL-11 O8 P5 1 50

02-JUL-11 O9 P6 2 50

02-JUL-11 O10 P2 4 10

1. All PRODUCTS SOLD ON BOTH DAYS AND NUMBER OF TIMES SOLD ?

WITH A AS (

SELECT ORDER\_DATE,PRODUCT\_ID,COUNT(\*) as cnt FROM ORDERS

GROUP BY ORDER\_DATE,PRODUCT\_ID)

SELECT A.PRODUCT\_ID , SUM(A.cnt) FROM A , A B Where A.PRODUCT\_ID=B.PRODUCT\_ID AND A.ORDER\_DATE != B.ORDER\_DATE

GROUP BY A.PRODUCT\_ID;

P1 3

P2 2

P3 2

1. Products Ordered on 2nd not on first

SELECT DISTINCT PRODUCT\_ID FROM ORDERS WHERE ORDER\_DATE = '02-JUL-2011' AND PRODUCT\_ID NOT IN (SELECT PRODUCT\_ID FROM ORDERS WHERE ORDER\_DATE != '02-JUL-2011');

P4

P6

P5

1. Highest sold products in both days

SELECT B.ORDER\_DATE,B.PRODUCT\_ID,B.TOTAL

FROM (

SELECT A.\*,ROW\_NUMBER () OVER(PARTITION BY A.ORDER\_DATE ORDER BY A.total DESC) AS rno FROM

(

SELECT ORDER\_DATE,PRODUCT\_ID,SUM(QUANTITY \* PRICE) AS total

FROM ORDERS GROUP BY ORDER\_DATE,PRODUCT\_ID )A)B

where B.rno = 1;

ORDER\_DATE, PRODUCT\_ID, TOTAL

01-JUL-11 P3 250

02-JUL-11 P3 125

EMPID MGRID

1

2 1

3 1

4 2

5 4

6 5

7 6

8 3

9 3

10 9

WITH TEMP\_X(TEMP\_EMPID,TEMP\_MGRID,HIER) AS

(

SELECT EMPID AS TEMP\_EMPID , MGRID AS TEMP\_MGRID , 0 AS HIER FROM EMP\_HIER WHERE EMPID=5

UNION ALL

SELECT B.EMPID,B.MGRID,A.HIER+1 FROM TEMP\_X A , EMP\_HIER B WHERE A.TEMP\_EMPID = B.MGRID

)

SELECT \* FROM TEMP\_X ORDER BY TEMP\_MGRID;

WITH A AS (SELECT START\_NUM,END\_NUM, ROW\_NUMBER() OVER (PARTITION BY START\_NUM ORDER BY ORDER\_TIME desc) as cnt FROM PH\_NUM ),

B AS (SELECT START\_NUM,END\_NUM, ROW\_NUMBER() OVER (PARTITION BY START\_NUM ORDER BY ORDER\_TIME asc) as cnt FROM PH\_NUM )

SELECT

A.START\_NUM,A.END\_NUM,

CASE WHEN A.END\_NUM=B.END\_NUM THEN 'Y' ELSE 'N' END as F

FROM A,B WHERE A.START\_NUM = B.START\_NUM AND A.cnt = B.cnt AND A.cnt=1;

SELECT ACCOUNT\_ID,AMOUNT,

LAG(AMOUNT,1) OVER(PARTITION BY ACCOUNT\_ID ORDER BY TR\_DATE) as Prev,

TR\_DATE

FROM USAGE\_TEST;

SELECT ACCOUNT\_ID,AMOUNT,

MIN(AMOUNT) OVER(PARTITION BY ACCOUNT\_ID ORDER BY TR\_DATE ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING) AS Prev\_Max,

TR\_DATE

FROM USAGE\_TEST;

WITH A AS (SELECT ACCOUNT\_ID,AMOUNT, TR\_DATE, ROW\_NUMBER() OVER (PARTITION BY USAGE\_TEST.ACCOUNT\_ID ORDER BY USAGE\_TEST.TR\_DATE ASC) as num FROM USAGE\_TEST)

SELECT A.ACCOUNT\_ID,A.AMOUNT ,B.AMOUNT AS Prev,A.TR\_DATE FROM A LEFT JOIN A B On A.ACCOUNT\_ID=B.ACCOUNT\_ID AND A.num-1 = B.NUm ORDER BY A.ACCOUNT\_ID,A.AMOUNT ,A.TR\_DATE ;

CREATE TABLE price\_change (Event\_Id varchar(10), Product\_ID varchar(10), algo varchar(1),price integer , change\_date DATE);

INSERT INTO price\_change VALUES('e2','p3','B',50,'01-APR-2017');

e1 p1 A 10 01-APR-17

e2 p1 A 20 02-APR-17

e3 p1 B 25 03-APR-17

e1 p2 A 20 01-APR-17

e2 p2 B 29 03-APR-17

e3 p2 B 30 02-APR-17

e1 p3 A 30 01-APR-17

e2 p3 B 50 03-APR-17

WITH A AS (

SELECT PRODUCT\_ID,ALGO,PRICE,CHANGE\_DATE,

ROW\_NUMBER() OVER (PARTITION BY PRODUCT\_ID,ALGO ORDER BY CHANGE\_DATE DESC) AS A\_RANK,

ROW\_NUMBER() OVER (PARTITION BY PRODUCT\_ID,ALGO ORDER BY CHANGE\_DATE ) AS B\_RANK

FROM price\_change

)

SELECT PRODUCT\_ID,

MAX(CASE WHEN ALGO = 'A' THEN PRICE END),

MAX(CASE WHEN ALGO = 'B' THEN PRICE END),

MAX(CHANGE\_DATE)

FROM

(

SELECT \* FROM A WHERE A\_RANK = 1 AND ALGO = 'A'

UNION

SELECT \* FROM A WHERE B\_RANK = 1 AND ALGO = 'B'

)B

GROUP BY PRODUCT\_ID

;

Summation – 3 days preceding and 2 days following

CREATE TABLE trans (tid integer,price integer , tran\_date date);

INSERT INTO trans VALUES(10,20,'10-APR-2017');

1 10 01-APR-17

2 20 02-APR-17

3 30 03-APR-17

4 30 03-APR-17

5 50 04-APR-17

6 50 05-APR-17

7 10 07-APR-17

8 20 08-APR-17

9 20 09-APR-17

10 20 10-APR-17

SELECT A.\*,

sum(price) OVER(order BY TRAN\_DATE ROWS BETWEEN 3 PRECEDING AND 2 FOLLOWING) AS CUM\_SUM

FROM TRANS a;