SQL QUERIES

REQUIREMENT:

To calculate weekly working hours for a user in a GREYTHR

SOLUTION:

**WITH**

S1 **AS** (**SELECT** EMPLID, **DATE**, **EXTRACT** ('WEEK' **FROM** **DATE**) **AS** WEEK **FROM** hr\_data\_coe.emp\_checkin **where** status='Present' ),

S2 **AS** (**SELECT** EMPLID, EMPNAME, **EXTRACT** ('WEEK' **FROM** **DATE**) **AS** WEEK, **sum**(total\_duration) WORKING\_HOURS

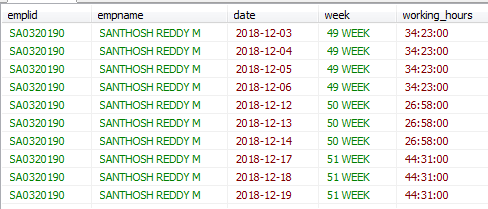
**FROM** hr\_data\_coe.emp\_checkin **where** status='Present'

**GROUP** **BY** EMPLID, EMPNAME, WEEK )

**SELECT** S2.EMPLID, EMPNAME, S1.**DATE**, CONCAT (S2.WEEK,' WEEK') **AS** WEEK,

S2.WORKING\_HOURS **FROM** S1,S2

**WHERE** S1.WEEK=S2.WEEK **AND** S1.EMPLID=S2.EMPLID **AND** S1.EMPLID='SA0320190' **ORDER** **BY** 1, 3;



To calculate weekly working hours for a user in a GREYTHR using date filter

**WITH**

S1 **AS** (**SELECT** EMPLID, **DATE**, **EXTRACT** ('WEEK' **FROM** **DATE**) **AS** WEEK **FROM** hr\_data\_coe.emp\_checkin **where** status='Present' ),

S2 **AS** (**SELECT** EMPLID, EMPNAME, **EXTRACT** ('WEEK' **FROM** **DATE**) **AS** WEEK, **sum**(total\_duration) WORKING\_HOURS

**FROM** hr\_data\_coe.emp\_checkin **where** status='Present'

**GROUP** **BY** EMPLID, EMPNAME, WEEK )

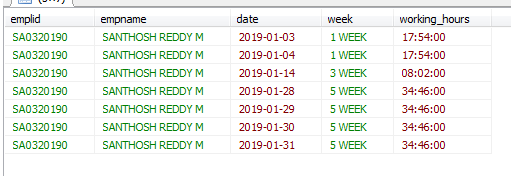
**SELECT** S2.EMPLID, EMPNAME, S1.**DATE**, CONCAT (S2.WEEK,' WEEK') **AS** WEEK,S2.WORKING\_HOURS **FROM** S1,S2

**WHERE** S1.WEEK=S2.WEEK **AND** S1.EMPLID=S2.EMPLID **and** s1.**date** **between** **to\_date** ('30-12-2018','DD-MM-YYYY')

**and** **to\_date** ('02-02-2019','DD-MM-YYYY') **and** **to\_char** (s2.WORKING\_HOURS,'HH24') < '42'

**and** s1.emplid = 'SA0320190'

**ORDER** **BY** 1,3;



Requirement2:

CASE1:

To find how many items are ordered based on hours, if there are missing hours then we have to show zero for those missing hours. Here we have to show orders hours data in HH24 format.

SOLUTION:

**SELECT** S2.HOURS,nvl(S1.CNT,0) cnt **FROM**

(**select**

**TO\_CHAR**(**to\_timestamp**(ordered\_date\_time,'DD-MM-YYYY HH:MI PM'),'HH24') hh,

**count**(\*) CNT

**from** tableau2

**group** **by** **TO\_CHAR**(**to\_timestamp**(ordered\_date\_time,'DD-MM-YYYY HH:MI PM'),'HH24')) S1,

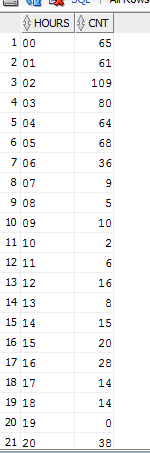
(**select** **case** **when** **length**(l)=1 **then** concat(0,l)

**else** l

**end** hours

**from**(**select** **to\_char**(**level**-1) l **from** dual **connect** **by** **level**<=24)) s2

**where** S1.hh(+)=s2.hours **order** **by** 1;



In above output for 19th HOUR there is no data so I have to show zero(0) for that remaining hours.

CASE2:

To find how many items are ordered based on hours, if there are missing hours then we have to show zero for those missing hours. Here we have to show orders hours data in HH12 format.

SOLUTION:

**SELECT** S2.HOURS,nvl(S1.CNT,0) cnt **FROM**

(**select**

**TO\_CHAR**(**to\_timestamp**(ordered\_date\_time,'DD-MM-YYYY HH:MI PM'),'HH12 PM') hh,

**count**(\*) CNT

**from** tableau2

**group** **by** **TO\_CHAR**(**to\_timestamp**(ordered\_date\_time,'DD-MM-YYYY HH:MI PM'),'HH12 PM')**order** **by** 1) S1,

(**select** **case** **when** l **between** 1 **and** 9 **then** '0'||concat(l,' AM')

**when** l **between** 10 **and** 12 **then** concat(l,' AM')

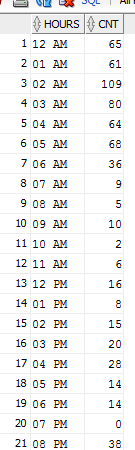
**when** l **between** 13 **and** 21 **then** '0'||concat(l-12,' PM')

**when** l **between** 22 **and** 24 **then** concat(l-12,' PM')

**end** hours

**from**(**select** **to\_char**(**level**) l **from** dual **connect** **by** **level**<=24) **order** **by** 1) s2

**where** S1.hh(+)=s2.hours **order** **by** **to\_date**(hours,'HH12 PM');



In above output for 7PM there is no data so I have to show zero(0) for that remaining hours.

Requirement3:

In this case we have numbers from 1 to n numbers. I have to show even numbers in one column and odd numbers in column.

SOLUTION:

s1 **as** (**select**

**case** **when** **mod**(l,2)<>0 **then** l

**end** odd

**from** s

**order** **by** odd),

s2 **as** (**select**

rownum r,odd

**from** s1

**where** odd **is** **not** **null**),

s3 **as** (**select**

**case** **when** **mod**(l,2)=0 **then** l

**end** even

**from** s

**order** **by** even),

s4 **as** (**select**

rownum r,even

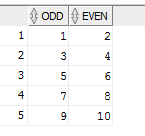
**from** s3

**where** even **is** **not** **null**)

**select** s2.odd,s4.even

**from** s2,s4

**where** s2.r=s4.r;



Requirement4:

In this case we have to query for fetching min value from col1 and max value from col2 from different data sets. I have done this in oracle database

SOLUTION:

with

s as

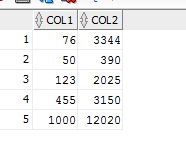
(select rownum-l r,col1,col2,l from (select col1,col2,level l from sam s

start with col1 not in (select col2 from sam)

connect by prior col2=col1))

select min(col1) col1,max(col2) col2 from s group by r

OUTPUT:



Requirement4:

In this case we have to query for fetching multiple table\_names with their column\_names, data\_types, is\_nullable and default column from pg\_columns table in postgresql. In this I have to concat character\_maximum\_length with data\_type using case statement.

SOLUTION:

**SELECT** **TABLE\_NAME**,**COLUMN\_NAME**,

concat(data\_type,

**case** **when** character\_maximum\_length **IS** **NOT** **NULL** **THEN**

concat('(',character\_maximum\_length::**TEXT**||')') **END** ) data\_type,

**case** **when** is\_nullable='NO' **THEN** 'NOT NULL'

**ELSE** 'NULL' **END** IS\_NULLABLE,

COLUMN\_DEFAULT

**FROM** information\_schema.columns

**WHERE**

**table\_name** **IN** ('xt\_user\_profile','xt\_company\_profile');

Output:

