260676

Template registration:

URL: textlocal.in
U/A: admin@maintwiz.com
PWD: Mw#textl0cal$23

URL: https://trueconnect.jio.com/#/
User ID : npmurali@maintwiz.com
New Password: Maintwiz!24
Entity Number : 1201159194917641600

Eg: PO %%|var1^{"inputtype" : "text", "maxlength" : "30"}%% has been created for Equipment %%|var2^{"inputtype" : "text", "maxlength" : "60"}%% at %%|var3^{"inputtype" : "text", "maxlength" : "60"}%%. This PO is updated from AX - MaintWiz

Manula Credentials:

URL: https://admin.manula.com/
User Name: mwsupport
Password: Support@123

DB Change:

update usersaccounts **set** uract_password = **'TWFpbnR3aXpAMjQ='**, uract_email =**'abc@gm.com'**, uract_telephone1 =**'0'**;
delete from escalation_usrmgt ;
update scheduledmaintenancemaster **set** smm_activestatus =**'f'**;
update workbenchmaster **set** wrbnm_active_status =**'f'**;
update calibrationprocessmaster **set** calpm_activestatus =**'f'**;
update coworkbenchmaster **set** cowrbnm_isactive =**'f'**;
update serviceorderschedule **set** sos_activestatus =**'f'**;
update ins_serviceorderschedule **set** insos_activestatus =**'f'**;
update meterreadingschedule **set** mrsd_activestatus =**'f'**,mrsd_select_sch_meterids=**''**;
update conditionalmonitorschedule **set** cbms_activestatus =**'f'**, cbms_select_sch_meterids =**''**;

Send grids Credential:

URL: https://app.sendgrid.com/login
User Name : maintwiz_mail
Password : Mw#sendgr1d$2023

Jio SMS template:

URL: https://trueconnect.jio.com/#/

User ID : npmurali@maintwiz.com
Password : Maintwiz!!25
Entity Number : 1201159194917641600

Eg: PO {#var#} has been created for Equipment {#var#}{#var#} at {#var#}{#var#} . This PO is updated from AX - MaintWiz

NAS login to store files

Nas URL - http://192.168.2.250:5000/
Nas Username - maintwizkt
Nas password - mTecH^2!4a8
Nas Folder - maintwizKT

Soap UI link:

SOAPRunner change to WSDLGen

###Asianpaints --- **Reservation**
UAT_ASP_ReservationURL = https://apmiidevapp.asianpaints.com:443/XMII/WSDLGen/AP_MAINTWIZ/Transactions/Tnx_Maintwiz_Reservation_Creation?wsdl

UAT_ASP_Res_username = ap_sapuser
UAT_ASP_Res_password = ap123
1104,KAS-WO014243,M77953,LL-K054,1,13-OHSF,L01004,NOS,ENG2,MI,P Mahendiran,No,temp2

Plant code, WOid,Part code,Equipment code,Qty,Facility cost center,GL code,UoM,ENG2/ENG4(Store code),MI,WO owner,No,temp2

###Asianpaints --- **Approval**
UAT_ASP_ApprovalURL = https://apmiidevapp.asianpaints.com:443/XMII/WSDLGen/AP_MAINTWIZ/Transactions/Tnx_Status_Update?wsdl

UAT_ASP_Apr_username = ap_sapuser
UAT_ASP_Apr_password = ap123

1104,KAS-WO014243,0027706216,MA,temp1,temp2,temp3

Plant code,WOid, Reservation ID,MA,temp1,temp2,temp3

AP Reservation UAT:

https://apmiidevapp.asianpaints.com:443/XMII/SOAPRunner/AP_MAINTWIZ/Transactions/Tnx_Mai ntwiz_Reservation_Creation?wsdl

  UAT_ASP_Res_username = ap_sapuser
  UAT_ASP_Res_password = ap123

Sample Payload:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xmii="http://www.sap.com/xMII">
  <soapenv:Header/>
  <soapenv:Body>
      <xmii:XacuteRequest>
      <!--Optional:-->
      <xmii:LoginName>ap_sapuser</xmii:LoginName>
      <!--Optional:-->
      <xmii:LoginPassword>ap123</xmii:LoginPassword>
      <!--Optional:-->
      <xmii:InputParams>
      <!--Optional:-->
      <xmii:Input>1104,KAS-WO014243,M77953,LL-K054,1,13-OHSF,
      L01004,NOS,ENG2,MI,P Mahendiran,No,temp2</xmii:Input>
      </xmii:InputParams>
      </xmii:XacuteRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

https://maintwizservices.in/standard/services/WOPartsComplete?wsdl

DB connection:

Apollo DB:

Host - apollo.ccauhzk9zobd.ap-south-1.rds.amazonaws.com
Password - MWsvrLegamcc833

UAT:

Please send UAT DB details
url   - 13.234.0.8
port   - 5432
username - postgres
password - Uat@postgres

QAT:

QAT-DB : IP address - 65.1.17.126
password - Uat@postgres

Pre-Live:

DB server IP - 3.7.115.245
Username - postgres
password - M8zg)3^DpW

Live:

Host - maintwizdbserver2023-cluster.cluster-ccauhzk9zobd.ap-south-1.rds.amazonaws.com
Password - MWsvrLe55&2023$$#

NP Zoom:

Link: https://zoom.us/j/5813869154#success
Mail ID: npmurali@maintwiz.com
PWD: Ceg@1992

To Delete Equipment:

Delete from equipmentgenerelinfo where eqpgi_eqpm_id in (xx);
Delete from equipmentmeterreading where eqpmtrd_eqpm_id in (xx);
Delete from equipmentnotification WHERE eqpnt_eqpm_id in (xx);
Delete from equipmentdocuments where eqpdc_eqpm_id in (xx);
Delete from equipmentpurchaseinfo where eqpprin_eqpm_id in (xx);
Delete from equipmentitemsattention where eqpita_eqpm_id in (xx);
Delete from equipmentvendors where eqpven_eqpm_id in (xx);
Delete from equipmentcriticalparts2 where eqpmtcp_eqpm_id in (xx);
Delete from equipmentoperations where eqpops_eqpm_id in (xx);
Delete from equipmentmaintenance where eqpman_eqpm_id in (xx);
Delete from equipmenttrainingcertificate where eqptrcer_eqpm_id in (xx);
Delete from equipmentfailurecodes where eqpficd_eqpm_id in (xx);
Delete from equipmentsafetypermit where eqpsfpmt_eqpm_id in (xx);
Delete from equipmentmaster where eqpm_id in (xx);

Equipment table:

Equipment Group -- equipmenttypes_m
Engineering Group -- equipmentcategory_m
Equipment Sub Group -- equipmentsubgroup_m
Equipment Class -- equipmentstatus_m
Equipment Criticality -- equipmentcriticality_m
Equipment Type -- assettype_m
Equipment Status -- equipmentclass_m

MR query for ASIANPAINTS

```
with eqp_mr_cte as (
 select eqpmtrd_id,eqpmtrd_eqpm_id,eqpmtrd_eqpm_code, eqpgi_atlocation, eqpmtrd_meter_code,
 facm_code
 from equipmentmeterreading
        INNER JOIN equipmentgenerelinfo ON eqpgi_id = eqpmtrd_eqpm_id
        inner join facilitymaster on facm_id = eqpgi_atlocation
 where eqpmtrd_isactive_status = true
),
meter_his_cte as (
 select facm_code, mrh_meter_code metercode, eqpmtrd_eqpm_code eqpmcode, eqpmtrd_eqpm_id
 eqpmid, max(mrh_mrm_create_date) mxdate
 from meterreadinghistory_y2022,eqp_mr_cte
 where mrh_meter_code = eqpmtrd_meter_code and mrh_eqpm_id = eqpmtrd_eqpm_id
        and mrh_meter_code not like '%EMR%'
 group by 1,2,3,4
)
select facm_code, eqpmcode, mrh_meter_code, mrh_mrm_create_date, mrh_lastreading
from meterreadinghistory_y2022, meter_his_cte
where mrh_mrm_create_date = mxdate and mrh_meter_code = metercode
 and eqpmid = mrh_eqpm_id
order by 1,2,3,4 desc
```

Order Number Manual Update:

```
select w.worprt_orderno,worm_facm_id,* from workorderparts w,workordermaster where
 worprt_orderno is null and worprt_complete ='f' and worm_id = w.worprt_worm_id order by 1 desc;

select w.worprt_orderno,* from workorderparts w where worprt_orderno is null and worprt_complete
 ='f' and worprt_isactive ='t' order by 1 desc;
```

CM Query for ASIANPAINTS

```
with eqp_cm_cte as (
 select eqpcbm_id, facm_code, eqpcbm_eqpm_id,eqpcbm_meter_code, eqpcbm_eqpm_code
 from equipmentcontrolmonitoring
        INNER JOIN equipmentgenerelinfo ON eqpgi_id = eqpcbm_eqpm_id
        inner join facilitymaster on facm_id = eqpgi_atlocation
 where eqpcbm_isactive_status = true
                and lower(eqpcbm_meter_code) not like '%ecm%'
),
cm_meter_his_cte as (
 select eqpcbm_eqpm_id mxeqpid, facm_code, eqpcbm_eqpm_code eqpcode, eqpcbm_meter_code
 mtrcode,
        max(cast(cmrh_cbmm_create_date as timestamp)) mxreaddate
 from condmtrreadhistory_y2022,eqp_cm_cte
```

where cmrh_eqpm_id = eqpcbm_eqpm_id and cmrh_meter_code = eqpcbm_meter_code
group by 1,2,3,4
)
select facm_code, eqpcode, mtrcode, mxreaddate, cmrh_lastreading
from cm_meter_his_cte, condmtrreadhistory_y2022
where cmrh_eqpm_id = mxeqpid and cmrh_meter_code = mtrcode
 and cast(cmrh_cbmm_create_date as timestamp) = mxreaddate;

If master available & history not available, use this

*****************CM select:**********************
with cmmeterdata as(
select cbmm_id as id from conditionalmonitormaster where cbmm_status = 'Closed'
except
select cmrh_cbmm_id from condmeterreadinghistory
)
select id,cbmm_create_date from cmmeterdata,conditionalmonitormaster where cbmm_id = id

*****************CM Delete:************************
with cmmeterdata as(
select cbmm_id as id from conditionalmonitormaster where cbmm_status = 'Closed'
except
select cmrh_cbmm_id from condmeterreadinghistory
)
Delete FROM conditionalmonitormaster USING cmmeterdata
WHERE cbmm_id = id
*****************MR select:**********************
with meterdata as(
select mrm_id as id from meterreadingmaster where mrm_status = 'Closed'
except
select mrh_mrm_id from meterreadinghistory
)
select id,mrm_create_date from meterdata,meterreadingmaster where mrm_id = id

*****************MR Delete:************************
with mrmeterdata as(
select mrm_id as id from meterreadingmaster where mrm_status = 'Closed'
except
select mrh_mrm_id from meterreadinghistory
)
--select id from mrmeterdata,meterreadingmaster where mrm_id = id
Delete FROM meterreadingmaster USING mrmeterdata
WHERE mrm_id = id

AP Store code

Ank - 1102-ENG2, 1102-ENG4

PAT - 1103-ENG2, 1103-ENG4

KAS - 1104-ENG2, 1104-ENG4

SRIP - 1106-ENG2, 1106-ENG4

KHN - 1107-ENG2, 1107-ENG4

MYS - 1108-ENG2, 1108-ENG4

ROH - 1109-ENG2, 1109-ENG4

VIZ - 1110-ENG2, 1110-ENG4

Query to check if SIR level is different:

select sirad_approval_status,sirad_approval_level,sir_approvalstatus,sir_approve_lvl,sirad_req_id,*
 from stockitemrequest s, stockitemrequest_approval_details s2 where sir_reqid = s2.sirad_req_id and
 sir_approve_lvl != s2.sirad_approval_level and sir_approvalstatus != 'Approved' and
 sirad_approval_status != 'Approved';

Query to check if SICR level is different:

select s2.sicrad_approval_status,s2.sicrad_approval_level,sicr_appr_sts ,sicr_appr_lvl
 ,s2.sicrad_req_id,* from stockitemchangereq s, stockitemchangerequest_approval_details s2 where
 sicr_reqid = s2.sicrad_req_id and sicr_appr_lvl != s2.sicrad_approval_level and sicr_appr_sts !=
 'Approved' and s2.sicrad_approval_status != 'Approved';

select * from stockitemchangerequest_approval_details where sicrad_req_id in ('');

select * from stockitemchangereq s where sicr_reqid in ('');

Updated Query SIR, SICR and MMR:

select sicrad_req_id, count(sicrad_approval_level),sicrad_approval_level from
 stockitemchangerequest_approval_details sad group by sicrad_req_id, sicrad_approval_level order by
 2 desc;

select * from stockitemchangerequest_approval_details where sicrad_req_id in ('');

```sql
select sirad_req_id, count(sirad_approval_level),sirad_approval_level from
 stockitemrequest_approval_details sad group by sirad_req_id , sirad_approval_level order by 2 desc;

select * from stockitemrequest_approval_details where sirad_req_id in ('');

select * from stockitemrequest ,stockitemrequest_approval_details s where sir_reqid = s.sirad_req_id
 and sir_approve_lvl != s.sirad_approval_level and s.sirad_approval_status = 'Waiting for Approval';

select * from stockitemchangereq ,stockitemchangerequest_approval_details s where sicr_reqid =
 s.sicrad_req_id  and sicr_appr_lvl != s.sicrad_approval_level  and s.sicrad_approval_status = 'Waiting
 for Approval';

select ommr_modifiedtime, ommr_date,ommr_sts_change_time, * from opnmechinemodifirequest o
 where ommr_date isnull or ommr_date = '' or ommr_sts_change_time isnull or
 ommr_sts_change_time::text = '';

update opnmechinemodifirequest set ommr_sts_change_time = ommr_modifiedtime where
 ommr_sts_change_time isnull or ommr_sts_change_time::text = '';

select ommrad_req_id , count(ommrad_approval_level),ommrad_approval_level from
 openmechinemodireq_approval_details oad group by ommrad_approval_level ,ommrad_req_id order
 by 2 desc;

select ommrmad_req_id ,count (ommrmad_approval_level),ommrmad_approval_level from
 openmechinemodireq_mainapproval_details group by ommrmad_approval_level,ommrmad_req_id
 order by 2 desc;

select * from openmechinemodireq_mainapproval_details where ommrmad_req_id = '';

select count(ommr_reqid),ommr_reqid from opnmechinemodifirequest o  group by ommr_reqid order
 by 1 desc;

select ommrmad_req_id,ommrmad_approval_level, count(ommrmad_req_id) from
 openmechinemodireq_mainapproval_details group by ommrmad_req_id,ommrmad_approval_level
 order by 3 desc;

select sir_exist_inv_value,sir_req_material_val,sir_tot_inv_val,* from stockitemrequest s where
 sir_exist_inv_value isnull or sir_req_material_val isnull or sir_tot_inv_val isnull;

update stockitemrequest set sir_exist_inv_value = 0 where sir_exist_inv_value isnull;
update stockitemrequest set sir_req_material_val = 0 where sir_req_material_val isnull;
update stockitemrequest set sir_tot_inv_val = 0 where sir_tot_inv_val isnull;

select sir_item_desc_check,* from stockitemrequest s where sir_approve_lvl > 'L2' and
 sir_item_desc_check not in ('Yes','No');
```

update stockitemrequest s set sir_item_desc_check ='No' where sir_approve_lvl > 'L2' and sir_item_desc_check not in ('Yes','No');

Thermax O&M User Logon details query:

select ulth_user_name as UserName,usr_personalcode as Personalcode,usr_type as UserRole,usr_billingidentity as Site,ulth_logon_time as LogonTime,ulth_logout_time as LogoutTime,ulth_event_type as Status,ulth_failure_reason as FailureReason from user_logon_track_history left join users_m on ulth_user_id = usr_id where usr_isactive ='Active' and usr_billingidentity in ('SGSL',   'KPC', 'TOESL',   'TML', 'Corporate',   'Utility',   'SCH',   'JSA', 'SLSL','BEL', 'JSW', 'ABD', 'UBL', 'RMPL',   'PCPL',   'KPW','BIFLK',   'CIPLA',   'CFL', 'KIPL','ABB', 'FAPL','TCL', 'KPW KPC', 'VIL', 'RIL', 'SMGPL',   'White house') and
cast(ulth_logon_time as date)>='2023-04-01' and cast(ulth_logon_time as date)<='2023-10-31' order by 5 asc;

To remove duplicate in CM history:

with cm_master as (
select cbmm_id as cbmmid,cbmm_select_sch_meterid::integer as meterid from conditionalmonitormaster where
cm_mom_id = 'CM061998'
)
Delete FROM condmeterreadinghistory WHERE cmrh_id in (select cmrh_id from condmeterreadinghistory,cm_master
where cmrh_cbmm_id = cbmmid and cmrh_eqpm_meter_id != meterid)

To check user is assigned:

select * from users_m um where um.usr_id in (uu,iii)

select eqpgi_owner_usr_id,eqpgi_equipment_owner,* from equipmentgenerelinfo where eqpgi_owner_usr_id =uu;

update equipmentgenerelinfo set eqpgi_owner_usr_id = iii, eqpgi_equipment_owner = 'Name' where eqpgi_owner_usr_id =uu;

select mrm_assignedto,* from meterreadingmaster where mrm_assignedto =uu and mrm_status not in ('Closed','Completed','Cancelled');

update meterreadingmaster set mrm_assignedto = iii where mrm_assignedto =uu and mrm_status not in ('Closed','Completed','Cancelled');

select mrsd_assignedto,* from meterreadingschedule where mrsd_assignedto =uu and mrsd_status not in ('Closed','Completed','Cancelled');

```sql
select cbmm_assignedto,* from conditionalmonitormaster where cbmm_assignedto =uu and
  cbmm_status not in ('Closed','Completed','Cancelled');

select cbms_assignedto,* from conditionalmonitorschedule where cbms_assignedto =uu and
  cbms_status not in ('Closed','Completed','Cancelled');

select smgi_assignedto,string_agg(smm_id::text,',') from scheduledmaintenancemaster left join
  scheduledmaintenancegenerelinfo on smm_id=smgi_smm_id where smgi_assignedto =uu and
  smm_activestatus='t' group by smgi_assignedto;

update scheduledmaintenancegenerelinfo set smgi_assignedto = iii where smgi_smm_id in (dd);

select smlt_assignedto,string_agg(s.smlt_id::text,',')  from scheduledmaintenancemaster left join
  schedulemaintlabourtask s on smm_id=smlt_smm_id where smlt_assignedto =uu and
  smm_activestatus='t' group by smlt_assignedto;

update schedulemaintlabourtask set smlt_assignedto = iii, smlt_assigneduser ='Name' where smlt_id in
  (cc);

select worgi_assignedto,string_agg(worm_id :: text,',') from workordermaster left join
  workordergenerelinfo on worm_id=worgi_worm_id where worgi_assignedto =uu and worm_status not
  in('Closed','Completed','Cancelled') group by worgi_assignedto;

update workordergenerelinfo set worgi_assignedto = iii where worgi_worm_id in ();

select worlbt_usrid,worlbt_assignedto,string_agg(w.worlbt_id :: text,',') from workordermaster left join
  workorderlabortasks w on worm_id=worlbt_worm_id where worlbt_usrid =uu and worm_status not
  in('Closed','Completed','Cancelled') group by w.worlbt_usrid,w.worlbt_assignedto;

update workorderlabortasks set worlbt_usrid = iii, worlbt_assignedto ='Name' where worlbt_id in (bb);

select wrbngi_assignedto,string_agg(wrbnm_id :: text,',') from workbenchmaster left join
  workbenchgenerelinfo on wrbnm_id=wrbngi_wrbnm_id where wrbngi_assignedto =uu and
  wrbnm_active_status='t' group by wrbngi_assignedto;

update workbenchgenerelinfo set wrbngi_assignedto = iii where wrbngi_wrbnm_id in (zz);

select wrbnlbt_usrid,string_agg(w.wrbnlbt_id:: text,',') from workbenchmaster left join
  workbenchlabortasks w on wrbnm_id=wrbnlbt_wrbnm_id where wrbnlbt_usrid =uu and
  wrbnm_active_status='t' group by wrbnlbt_usrid;

update workbenchlabortasks set wrbnlbt_usrid = iii, wrbnlbt_assignedto ='Name' where wrbnlbt_id in
  (11863974);

select calpm_assignto_usr_id,* from calibrationprocessmaster where calpm_assignto_usr_id =uu and
  calpm_activestatus='t';
```

```sql
select calom_assignto_usr_id,* from calibrationordermaster where calom_assignto_usr_id =uu and
 calom_status not in('Completed','Closed','Completed','Cancelled');

select cowrbnmcockls_usrid,cowrbnmcockls_assignedto,* from cowbchecklist where
 cowrbnmcockls_usrid =uu;

select cowrbnm_assignto_usr_id,* from coworkbenchmaster where cowrbnm_assignto_usr_id =uu and
 cowrbnm_isactive='t';

select cockls_usrid,cockls_assignedto,* from calibrationorder_checklist where cockls_usrid =uu;

select calprochkls_assignedto_id,calprochkls_assignedto_from,* from calibrationprocesschecklist
 where calprochkls_assignedto_id =uu;

select som_assignedto,string_agg(som_id::text,',') from serviceordermaster where som_assignedto =uu
 and som_status not in ('Closed','Completed','Cancelled') group by som_assignedto;

update serviceordermaster set som_assignedto = iii where som_id in (ff);

select insos_assignedto,* from ins_serviceorderschedule where insos_assignedto =uu  and
 insos_activestatus ='t';

select sos_assignedto,* from serviceorderschedule where sos_assignedto =uu and sos_activestatus ='t';

select inswogi_assignedto,* from instrumentwomaster,instrumentwogenerelinfo where
 inswogi_assignedto =uu and inswom_status not in ('Closed','Completed','Cancelled');

select inssom_assignedto,* from instrumentserviceordermaster where inssom_assignedto =uu and
 inssom_status not in ('Closed','Completed','Cancelled');

select usr_id,* from facilitynotification f where usr_id =uu;

select eqpnt_usr_id,* from equipmentnotification e where eqpnt_usr_id =uu;

update equipmentnotification set eqpnt_usr_id = iii where eqpnt_usr_id = uu;

select prtnt_usr_id,* from partsnotification p where prtnt_usr_id =uu;

select insnt_usr_id,* from instrumentnotification i where insnt_usr_id =uu;

select wornt_usr_id,* from workordernotification w where wornt_usr_id =uu;

select wrbnnt_usr_id,* from workbenchnotification w where wrbnnt_usr_id =uu;

select inswont_usr_id,* from instrumentwonotification i where inswont_usr_id =uu;
```

```sql
select insant_usr_id,* from instrumentallocationnotification i where insant_usr_id =uu;

select * from escalation_usrmgt where esc_usr_ids::text like '{%1597%}';

select string_agg(wfcd_id::text,',') from mw_workflow_config_details m where wfcd_approve_by::text
 like '%uu%';

select wfcd_approve_by,* from mw_workflow_config_details mwcd where wfcd_id in
 (2138,2142,2144,2207);

update mw_workflow_config_details set wfcd_approve_by = cast(REPLACE(cast (wfcd_approve_by
 as varchar(10000)) , 'uu' , 'iii')as integer[]) where wfcd_approve_by::text like '%uu%';

select string_agg(wospwf_id::text,',') from workorder_spareparts_workflow wsw where
 wospwf_wfcd_id in (WFCD-ID) and wospwf_approved_status = 'Waiting for Approval';

update workorder_spareparts_workflow w set wospwf_approve_usrid = d.wfcd_approve_by from
 mw_workflow_config_details d where wfcd_id = w.wospwf_wfcd_id and wospwf_wfcd_id in
 (WFCD-ID) and wospwf_approved_status = 'Waiting for Approval';

update users_m set usr_notify_medium=null, usr_notify_lang=null, usr_consolidate_msg=null,
 usr_messageopt='f', usr_msgopt_update='f', usr_transaction_msgs=null where usr_isactive =
 'DeActive';
```

Shift Configuration:

//common
```sql
update avail_daywise_workhours set adwwh_daywise_hours
 =cast(fn_getfinyear_facwise_allday_shifthours_new(adwwh_facm_id ,2023, '') as json)
where adwwh_adfyd_id in (select adfyd_id from availdays_financeyeardetails where adfyd_fym_year
 =2023);
```

// for 3 shift
```sql
update availabledaysmaster_new
set
adm_sunday = true,
adm_monday =true,
adm_tuesday = true,
adm_wednesday = true,
adm_thursday =true,
adm_friday =true,
adm_saturday = true,
adm_noofshifts = 3 ,
adm_shift_start_time ='06:00',
adm_shift1_start_time='06:00',adm_shift1_end_time='14:00',adm_shift1_total_time='08:00' ,
```

```
adm_shift2_start_time='14:00' ,adm_shift2_end_time='22:00' ,adm_shift2_total_time='08:00',
adm_shift3_start_time='22:00' ,adm_shift3_end_time='06:00' ,adm_shift3_total_time='08:00' ,
adm_total_shift_time='24:00' where adm_facm_id in (xx);
```

// for 2 shift
```
update availabledaysmaster_new
set
adm_sunday = true,
adm_monday =true,
adm_tuesday = true,
adm_wednesday = true,
adm_thursday =true,
adm_friday =true,
adm_saturday = true,
adm_noofshifts = 2 ,
adm_shift_start_time ='07:00',
adm_shift1_start_time='07:00',adm_shift1_end_time='15:00',adm_shift1_total_time='08:00' ,
adm_shift2_start_time='16:30' ,adm_shift2_end_time='00:30' ,adm_shift2_total_time='08:00',
adm_shift3_start_time='00:00' ,adm_shift3_end_time='00:00' ,adm_shift3_total_time='00:00' ,
adm_total_shift_time='16:00' where adm_facm_id in (xx)
```

// for 1 shift
```
update availabledaysmaster_new
set
adm_sunday = true,
adm_monday =true,
adm_tuesday = true,
adm_wednesday = true,
adm_thursday =true,
adm_friday =true,
adm_saturday = true,
adm_noofshifts = 1 ,
adm_shift_start_time ='08:30',
adm_shift1_start_time='08:30',adm_shift1_end_time='17:00',adm_shift1_total_time='08:30' ,
adm_shift2_start_time='00:00' ,adm_shift2_end_time='00:00' ,adm_shift2_total_time='00:00',
adm_shift3_start_time='00:00' ,adm_shift3_end_time='00:00' ,adm_shift3_total_time='00:00' ,
adm_total_shift_time='08:30' where adm_facm_id in (xx);
```

Maint. Mgmt. -> PM Compliance Report:

Show All – All generated orders.
Plan Compliance – All completed orders as per plan (Includes all Completed orders).
Schedule Compliance – All completed orders as per schedule (Completed within the expected closure
 date).

Plan but not schedule compliance – All completed orders executed after the schedule date (Completed after the expected closure date).
Plan Non Compliance – All incomplete Open orders.

KPI Dashboard Formulas:

1) MTTR - Sum of LH / No Of BD Occurrence
2) MTBF - (Total AH - Sum of LH) / No Of BD Occurrence
3) BD % - Sum of LH / (Available Hours-Sum of PM LH) * 100
 Mech BD % - Sum of Mech LH / (Available Hours-Sum of Mech PM LH) * 100
4) PM Compliance - Completed / (Total no of records-Cancelled)
5) PM Effective - Sum of PM LH/ (Sum of PM LH + Sum of BD LH)*100
6) AH - (No of parent equipment 'Yes' machine - Deleted) * No of days * Shift Hours
7) AH% - ((AH-Sum of LH)/AH)*100

AH- AVAILABLE HOURS
LH- LOST HOURS
BD- BREAKDOWN

To deactivate Instrument:

select * from calibrationprocessmaster where calpm_insm_code in ('EI-1088') and calpm_activestatus = true

select * from coworkbenchmaster where cowrbnm_isactive ='T' AND cowrbnm_calpm_serial_id in ('ATCJB-SM-CO000840")

select * from ins_serviceorderschedule where insos_insm_code in ('EI-1088') and insos_activestatus = 'T'

Dashboard Config tables:

mw_custom_first_level_facility_group_config_m;
mw_custom_facility_grouping_config_m;
mw_facilitywise_module_config_m;
update mw_custom_first_level_facility_group_config_m set cmflfgcm_group_ids ='{}';
update mw_custom_facility_grouping_config_m set cmfgcm_group_ids = '{}';
update mw_facilitywise_module_config_m set mwfwmcm_facm_ids = '{}' where mwfwmcm_id in (xx);

To move shift config from year to year:

//3 Shift
update availabledaysmaster_new

```sql
set adm_shift1_start_time=adm_shift_start_time,
adm_shift1_end_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+(cast (concat(adm_prodhourspershift, 'hours') as interval))),'HH24:MI'),
adm_shift1_total_time=( to_char(cast (concat(adm_prodhourspershift, 'hours') as interval),'HH24:MI')),
adm_shift2_start_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift, 'hours') as interval))),'HH24:MI'),
adm_shift2_end_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift*2, 'hours') as interval))),'HH24:MI'),
adm_shift2_total_time=( to_char(cast (concat(adm_prodhourspershift, 'hours') as interval),'HH24:MI')),
adm_shift3_start_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift*2, 'hours') as interval))),'HH24:MI'),
adm_shift3_end_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift*3, 'hours') as interval))),'HH24:MI'),
adm_shift3_total_time=( to_char(cast (concat(adm_prodhourspershift, 'hours') as interval),'HH24:MI')),
adm_total_shift_time=( to_char(cast (concat(adm_prodhourspershift*adm_noofshifts, 'hours') as
 interval),'HH24:MI'))
where adm_noofshifts=3 and adm_facm_id in (xx);
```

//2 Shift
```sql
update availabledaysmaster_new
set adm_shift1_start_time=adm_shift_start_time,
adm_shift1_end_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+(cast (concat(adm_prodhourspershift, 'hours') as interval))),'HH24:MI'),
adm_shift1_total_time=( to_char(cast (concat(adm_prodhourspershift, 'hours') as interval),'HH24:MI')),
adm_shift2_start_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift, 'hours') as interval))),'HH24:MI'),
adm_shift2_end_time=to_char((TO_TIMESTAMP(concat('2000-01-01
 ',adm_shift_start_time),'YYYY-MM-DD HH:MI')
+( cast (concat(adm_prodhourspershift*2, 'hours') as interval))),'HH24:MI'),
adm_shift2_total_time=( to_char(cast (concat(adm_prodhourspershift, 'hours') as interval),'HH24:MI')),
adm_shift3_start_time='00:00',
adm_shift3_end_time='00:00',
adm_shift3_total_time='00:00',
adm_total_shift_time=( to_char(cast (concat(adm_prodhourspershift*adm_noofshifts, 'hours') as
 interval),'HH24:MI'))
where adm_noofshifts=2 and adm_facm_id in (xx);
```

Duplicate in SM Checklist:

```sql
with CTE as
        (select  concat(s.smlt_smm_id, s.smlt_tasks, s.smlt_assignedto, s.smlt_natureofwork,
        s.smlt_estimatedefffort, s.smlt_notes, s.smlt_expectedresult) as task_concat,
        s.*,
        sm.smm_smid
        from    schedulemaintlabourtask s
        join    scheduledmaintenancemaster sm on sm.smm_id = s.smlt_smm_id
        where   sm.smm_activestatus = true )
select   sub1.task_count,
        sub1.task_concat,
        sub1.smlt_id,
        sub1.smlt_createdtime,
        sub1.row_no,
        sub1.smm_smid
from
        (select count(task_concat) over (partition by task_concat) as task_count,
        row_number () over (partition by task_concat order by task_concat ) as row_no,
        *
        from CTE) as sub1
where   sub1.task_count > 1
and             sub1.row_no != 1
order by task_concat, task_count
```

Count in duplicate SM Checklist:

```sql
with CTE as (
 select concat(s.smlt_smm_id, s.smlt_tasks, s.smlt_assignedto, s.smlt_natureofwork,
        s.smlt_estimatedefffort, s.smlt_notes, s.smlt_expectedresult) as task_concat,
            s.*
 from   schedulemaintlabourtask s
 join    scheduledmaintenancemaster sm on sm.smm_id = s.smlt_smm_id
 where          sm.smm_activestatus = true )
select sub1.task_count, sub1.task_concat,sub1.smlt_id, sub1.smlt_createdtime
from    (
        select count(task_concat) over (partition by task_concat) as task_count,
        *
        from CTE) as sub1
where sub1.task_count > 1
order by task_count,task_concat
```

To get the mismatch between Work order Task Count and Schedule Task Count

```sql
select * from
 (
 select worm_id ,count(worlbt_description ) as workorder_task_count, s.smlt_smm_smid  ,
 s.Schedule_Task_Count
```

```
, case
      when count(worlbt_description ) = s.Schedule_Task_Count then 'matching'
      else 'not_matching'
   end as taskslist_count
, w.worm_createdtime, sm.smm_activestatus ,sm.smm_createdtime
 from workorderlabortasks wt
 full join workordermaster w on w.worm_id = wt.worlbt_worm_id
 join
    (
    select smlt_smm_smid ,smlt_smm_id   ,count (smlt_tasks) as Schedule_Task_Count
    from schedulemaintlabourtask s
    group by smlt_smm_id, smlt_smm_smid
    ) as s
    on smlt_smm_smid = worm_som_smm_id
 full join scheduledmaintenancemaster sm on sm.smm_id = s.smlt_smm_id
 where worm_som_smm_id != '' and worm_status not in ('Completed','Cancelled','Closed')  --> Change
 this "worm_status" to get results respectively
 group by worm_id, s.smlt_smm_smid  ,s.Schedule_Task_Count, sm.smm_activestatus
 ,sm.smm_createdtime
 ) as wt
where wt.taskslist_count = 'not_matching'
order by 6 desc;
```

To get the mismatch between Workbench Task Count and Schedule Task Count

```
select * from
(
 select w2.wrbnm_id, count(wrbnlbt_description) as workbench_task_count
 ,w2.wrbnm_som_smm_id,s.Schedule_Task_Count
 , case
       when count(wrbnlbt_description) = s.Schedule_Task_Count then 'matching'
       else 'not_matching'
   end as taskslist_count
, w2.wrbnm_createdtime,  sm.smm_activestatus ,sm.smm_createdtime
 from workbenchlabortasks w
 full join workbenchmaster w2 on w2.wrbnm_id =w.wrbnlbt_wrbnm_id
 join
       (
       select smlt_smm_smid ,smlt_smm_id ,count (smlt_tasks) as Schedule_Task_Count
       from schedulemaintlabourtask
       group by smlt_smm_id, smlt_smm_smid
       ) as s
       on smlt_smm_smid = w2.wrbnm_som_smm_id
 full join scheduledmaintenancemaster sm on sm.smm_id = s.smlt_smm_id
 where w2.wrbnm_active_status = true                --> Change this "where condition" to get
 results for inactive/active workbench
```

group by w2.wrbnm_id, w2.wrbnm_som_smm_id,s.Schedule_Task_Count, sm.smm_activestatus
,sm.smm_createdtime
) as wbt
where wbt.taskslist_count = 'not_matching'
order by 6 desc

To move checklist from SM to WB

Delete WB checklist:

select string_agg(wrbnm_id::text,',') from workbenchmaster where wrbnm_som_smm_id in (SMID)and
wrbnm_active_status = true

Select * from workbenchlabortasks where wrbnlbt_wrbnm_id in (WBID)

Delete from workbenchlabortasks where wrbnlbt_wrbnm_id in (WBID)

Insert query:

select fn_insert_sm_to_wb_chklst_bulkrecords(1, 'WBID');

To move checklist from SM to WO

Delete WO checklist:

select string_agg(worm_id::text,',') from workordermaster where worm_som_smm_id in (SMID) and
worm_status not in ('Closed','Completed','Cancelled')

Select * from workorderlabortasks where worlbt_worm_id in (WOID)

Delete from workorderlabortasks where worlbt_worm_id in (WOID)

Insert query:

select fn_insert_wb_to_wo_chklst_bulkrecords(1, 'WOID');
Conditional Mointer Meter data upload:

Step 1 : Download the import records excel copy from open tab conditional monitoring.

Step 2 : Download the conditional monitoring reading upload template from OPERATION / METER
READING IMPORT/Equipment Condition Monitoring Import - Schedule Records ------Download
this template and feed the client data into this.

To check whether any duplicate exist in PM WO for particular date:

```
select * from workordermaster where cast(worm_createdtime as date) = '2022-03-17' and
 worm_createfrom = 'sm';
```

To update instrument code:

```
update instrumentgenerelinfo set insgi_insm_code  = 'xx' where insgi_insm_id in (xx);
update instrumentaccuracy set inacy_insm_code  = 'xx' where inacy_insm_id  in (xx);
update instrumentnotification set insnt_insm_code  = 'xx' where insnt_insm_id  in (xx);
update instrumentdocuments set insdc_insm_code  = 'xx' where insdc_insm_id  in (xx);
update instrumentcalibration set insclb_insm_code  = 'xx' where insclb_insm_id  in (xx);
update instrumentcalibrationrange set inscr_insm_code  = 'xx' where inscr_insm_id  in (xx);
update instrumentpurchaseinfo set insprin_insm_code  = 'xx' where insprin_insm_id in (xx);
update instrumentspecification_main set insspm_insm_code  = 'xx' where insspm_insm_id  in (xx);
update instrumentspecifications set inssp_insm_code  = 'xx' where inssp_insm_id  in (xx);
update instrumentvendors set insven_insm_code  = 'xx' where insven_insm_id in (xx);
update instrumentcalibrationrangeoutput set inscr_out_insm_code  = 'xx' where inscr_out_insm_id  in
 (xx);
update instrumentfailurecodes set insficd_insm_code  = 'xx' where insficd_insm_id in (xx);
update instrumentissuecodes set insicd_insm_code  = 'xx' where insicd_insm_id in (xx);
update instrumentcalibrationandmaintenance set inscalmaint_insm_code  = 'xx' where
 inscalmaint_insm_id in (xx);
update instrumentmasterinformation set insmaster_insm_code  = 'xx' where insmaster_insm_id  in (xx);
update instrumentmaster set insm_code = 'xx' where insm_id IN(xx);
```

Instrument Code change:

```
select * from instrumentgenerelinfo where insgi_insm_id in (xx);
select * from instrumentcalibration where insclb_insm_id  in (xx);
select * from instrumentcalibrationrange where inscr_insm_id  in (xx);
select * from instrumentspecification_main where insspm_insm_id  in (xx);
select * from instrumentcalibrationrangeoutput where inscr_out_insm_id  in (xx);
select * from instrumentcalibrationandmaintenance where inscalmaint_insm_id in (xx);
select * from instrumentmaster where insm_id IN(xx) ;
```


To Delete Instrument:

```
Delete from instrumentgenerelinfo where insgi_insm_id in (xx);
Delete from instrumentaccuracy where inacy_insm_id  in (xx);
Delete from instrumentnotification where insnt_insm_id  in (xx);
Delete from instrumentdocuments where insdc_insm_id  in (xx);
Delete from instrumentcalibration where insclb_insm_id  in (xx);
Delete from instrumentcalibrationrange where inscr_insm_id  in (xx);
Delete from instrumentpurchaseinfo where insprin_insm_id in (xx);
Delete from instrumentspecification_main where insspm_insm_id  in (xx);
Delete from instrumentspecifications where inssp_insm_id  in (xx);
```

Delete from instrumentvendors where insven_insm_id in (xx);
Delete from instrumentcalibrationrangeoutput where inscr_out_insm_id in (xx);
Delete from instrumentfailurecodes where insficd_insm_id in (xx);
Delete from instrumentissuecodes where insicd_insm_id in (xx);
Delete from instrumentcalibrationandmaintenance where inscalmaint_insm_id in (xx);
Delete from instrumentallocationhistory where insahy_insm_id in (xx);
Delete from instrumentmasterinformation where insmaster_insm_id in (xx);
Delete from instrumentmaster where insm_id IN(xx) ;

To Delete CalibrationSchedule:

Delete from calbrationscheduler where calsch_calpm_id in (xx) ;
Delete from calbrationschedulelatest where calschlat_calpm_id in (xx) ;
Delete from calbrationscheduleall where calschall_calpm_id in (xx) ;
Delete from calibrationprocessprereading where calprocper_calom_id in (xx) ;
Delete from calibrationprocesspostreading where calprocpor_calom_id in (xx);
Delete from calibrationprocessvendor where calprovn_calom_id in (xx) ;
Delete from calibrationprocesschecklist where calprochkls_calom_id in (xx) ;
Delete from calibrationprocessmaster where calpm_id in (xx);

To Delete CalibrationOrder:

Delete from calibrationorcompletenotes where calorcn_calom_id in (xx);
Delete from calibrationorder_checklist where cockls_calom_id in (xx);
Delete from calibrationorder_recommendations where corecom_calom_id in (xx);
Delete from calibrationorder_readingmaster where calordrmas_calom_id in (xx);
Delete from calibrationordervendor where calomvn_calom_id in (xx);
Delete from calibrationobservation where calobs_calom_id in (xx);
Delete from calibrationpostreading where calpor_calom_id in (xx);
Delete from calibrationprereading where calper_calom_id in (xx);
Delete from calibrationraisefallreading where calrfr_calom_id in (xx);
Delete from calibrationorder_preparation where coprep_calom_id in (xx);
Delete from calibrationorder_history where cohis_calom_id in (xx);
Delete from calibrationordermaster where calom_id in (xx);

To update Eqpm. Code Change:

update equipmentgenerelinfo set eqpgi_eqpm_code ='' where eqpgi_eqpm_id in (xx);
update equipmentmeterreading set eqpmtrd_eqpm_code = '' where eqpmtrd_eqpm_id in (xx);
update equipmentnotification set eqpnt_eqpm_code = '' WHERE eqpnt_eqpm_id in (xx);
update equipmentdocuments set eqpdc_eqpm_code = '' where eqpdc_eqpm_id in (xx);
update equipmentpurchaseinfo set eqpprin_eqpm_code = '' where eqpprin_eqpm_id in (xx);
update equipmentitemsattention set eqpita_eqpm_code = '' where eqpita_eqpm_id in (xx);
update equipmentvendors set eqpven_eqpm_code ='' where eqpven_eqpm_id in (xx);
update equipmentcriticalparts2 set eqpmtcp_eqpm_code ='' where eqpmtcp_eqpm_id in (xx);
update equipmentoperations set eqpops_eqpm_code ='' where eqpops_eqpm_id in (xx);

```sql
update equipmentmaintenance set eqpman_eqpm_code ='' where eqpman_eqpm_id in (xx);
update equipmenttrainingcertificate set eqptrcer_eqpm_code ='' where eqptrcer_eqpm_id in (xx);
update equipmentfailurecodes set eqpficd_eqpm_code ='' where eqpficd_eqpm_id in (xx);
update equipmentsafetypermit set eqpsfpmt_eqpm_code ='' where eqpsfpmt_eqpm_id in (xx);
update workbenchmaster set wrbnm_eqpm_code ='' where wrbnm_eqpm_id in (xx);
update workordermaster set worm_eqpm_code ='' where worm_eqpm_id in (xx);
update serviceordermaster set som_eqpm_code = '' where som_eqpm_id in (xx);
update meterreadingmaster set mrm_eqpm_code = '' where mrm_eqpm_id in (xx);
update conditionalmonitormaster set cbmm_eqpm_code = '' where cbmm_eqpm_id in (xx);
update equipmentmaster set eqpm_code ='' where eqpm_id in (xx);
```

To Delete Conditional Monitor Open Order:

```sql
Delete from conditionalmonitormaster c where cbmm_id in (xx);
Delete from condmeterreadinghistory c where cmrh_cbmm_id in (xx);
```

To delete Reservation from a WO:

```sql
delete from workorder_spareparts_workflow where wospwf_worm_id = xx and wospwf_reservation_id
 = 'ResID';
delete from workorder_spareparts_reservation_info where wospri_worm_id = xx and
 wospri_sap_reservation_id = 'ResID';
delete from workorder_partsissue_return where wopir_worm_id = xx and wopir_worprt_reservation_id
 = 'ResID';
delete from workorderparts where worprt_worm_id = xx and worprt_reservation_id = 'ResID';
```

Bulk closure WO of Shift:

```sql
with woshift_data_cte as (select worsftd_worm_id, coalesce(worsftd_off_datetime,
 worsftd_created_datetime) comp_date_time from workordershiftdetails), wdata as (select worm_id,
 worm_woid, worm_facm_id, worm_eqpm_id, worm_createdtime, comp_date_time compdate,
        fn_get_shiftdate_data(worm_facm_id, worm_eqpm_id, cast(to_char(comp_date_time,'YYYY')
 as integer),
        to_char(comp_date_time,'Mon-YYYY'), to_char(comp_date_time,'dd-Mon-YYYY HH24:MI'))
 compsftdata
 from woshift_data_cte, workordermaster
 where worsftd_worm_id = worm_id
        and worm_status not in ('Closed', 'Completed')
        --and (worm_type like 'Preventive%' or worm_createfrom not like 'sm%')
        --and cast(worm_createdtime as date) >= '2018-08-01'
        --and cast(worm_createdtime as date) <= '2018-12-31'
        and worm_id in (34483,34484) ----------Here mention the list of worm _id.
)
update workordershiftdetails
set worsftd_end_shift_date = cast((compsftdata).shiftdate as date),
 worsftd_end_shift = (compsftdata).shifttype,
```

worsftd_closed_datetime = compdate,
 worsftd_closed_shift = (compsftdata).shifttype
from wdata
where worm_id = worsftd_worm_id;

----Completion Data Update

with woshift_data_cte as (select worsftd_worm_id, coalesce(worsftd_off_datetime,
 worsftd_created_datetime) comp_date_time from workordershiftdetails), wdata as (select worm_id,
 worm_woid, worm_facm_id, worm_eqpm_id, worm_createdtime, comp_date_time compdate,
 fn_get_shiftdate_data(worm_facm_id, worm_eqpm_id, cast(to_char(comp_date_time,'YYYY') as
 integer), to_char(comp_date_time,'Mon-YYYY'), to_char(comp_date_time,'dd-Mon-YYYY
 HH24:MI')) compsftdata from woshift_data_cte, workordermaster
 where worsftd_worm_id = worm_id
        and worm_status not in ('Closed', 'Completed')
        --and (worm_type like 'Preventive%' or worm_createfrom not like 'sm%')
        --and cast(worm_createdtime as date) >= '2018-08-01'
        --and cast(worm_createdtime as date) <= '2018-12-31'
        and worm_id in (128245,   128246) ----------Here mention the list of worm _id)
update workordercompletionnotes wcn set worcn_completiondate = compdate, worcn_modifiedtime =
 now(xx), worcn_usr_from = 'SU', worcn_completedby = 'SAKTHIVEL R'-----Mention the completed
 by user from wdata where worm_id = worcn_worm_id;

---WO Master Update

with woshift_data_cte as (select worsftd_worm_id, coalesce(worsftd_off_datetime,
 worsftd_created_datetime) comp_date_time from workordershiftdetails), wdata as (select worm_id
 woid, worm_woid, worm_facm_id, worm_eqpm_id, worm_createdtime, comp_date_time compdate,
 fn_get_shiftdate_data(worm_facm_id, worm_eqpm_id, cast(to_char(comp_date_time,'YYYY') as
 integer), to_char(comp_date_time,'Mon-YYYY'), to_char(comp_date_time,'dd-Mon-YYYY
 HH24:MI')) compsftdata from woshift_data_cte, workordermaster
 where worsftd_worm_id = worm_id and worm_status not in ('Closed', 'Completed')
        --and (worm_type like 'Preventive%' or worm_createfrom not like 'sm%')
        --and cast(worm_createdtime as date) >= '2018-08-01'
        --and cast(worm_createdtime as date) <= '2018-12-31'
        and worm_id in (34483,34484) ----------Here mention the list of worm _id) update
 workordermaster set worm_status = 'Completed' from wdata where worm_id = woid;

_____

   _____

Then finally update the completed status in workorderhistory table

insert into workorderhistory (worhis_worm_id, worhis_status_change, worhis_user,
 worhis_createdtime, worhis_modifiedtime, worhis_createduser_id, worhis_modifieduser_id) values
 (34483, 'Completed', 'GOVINDARAJULU A', now(), now(), 69, 69);

To edit user count in maintwiz database:

SELECT mw_sdur_adminusers, mw_sdur_productionusers, mw_sdur_normalusers,
 mw_sdur_totalusers
            FROM mw_sudomain_user_restriction
              WHERE mw_sdur_subdomain ='tpicomposites'

update mw_sudomain_user_restriction set mw_sdur_adminusers = 20, mw_sdur_productionusers
 =100, mw_sdur_normalusers =100 where mw_sdur_subdomain = 'tpicomposites'

Calculation:

Super user = (Normal user * 33%) + (Production user * 50%) + (Admin user * 100%)

To add or edit approver of level 1, 2, 3:

Select role ID to add or edit :

select * from roletypes_m where rlty_type = 'VIZ_SENIOR OFFICER'

Find Facilty mapped config ID:

select * from mw_workflow_config_master, mw_workflow_config_details where wfcm_workflow_for
 = 'SP' and wfcm_id = wfcd_wfcm_id and wfcm_id = 1

Update query to add role:

update mw_workflow_config_details set wfcd_approve_by_role=wfcd_approve_by_role||'{98,94,506}'
 where wfcd_level = 'L2' and wfcd_wfcm_id = 1

QUERY TO CHECK WHICH FACILITY HAS ROLE AND LEVEL APPROVAL RANGE:

SELECT wfcd_id,(SELECT unnest(wfcd_condition_desc) AS sp_cond_id), wfcd_approve_by,
 wfcd_approve_by_grp, wfcd_approve_by_role,
 wfcd_level,wfcn_wfcp_param,wfcd_approve_send_msg
FROM mw_workflow_config_details wcd
LEFT JOIN mw_workflow_config_master ON wfcm_id =wfcd_wfcm_id
LEFT JOIN mw_workflow_conditions ON (SELECT unnest(wfcd_condition_desc) AS f)=wfcn_id
WHERE 577 = ANY (CAST(string_to_array(wfcm_facm_id, ',') AS INT[])) AND
 wfcm_workflow_for='SP'
 ORDER BY wfcd_level asc

SELECT
  wfcm_id, **string_agg**(um.usr_name::**text**,**','**) **as** app_name,
  wfcm_facm_id, wcd.wfcd_approve_by,

```
  wcd.wfcd_level, wfcn_wfcp_param
FROM
  mw_workflow_config_details wcd
JOIN mw_workflow_config_master wcm ON wcd.wfcd_wfcm_id = wcm.wfcm_id
JOIN mw_workflow_conditions wfc ON (SELECT unnest(wcd.wfcd_condition_desc)) =
wfc.wfcn_id
JOIN users_m um ON um.usr_id = ANY(wcd.wfcd_approve_by)
WHERE wcm.wfcm_workflow_for = 'SP'
group by wfcm_id,wfcm_facm_id, wcd.wfcd_approve_by,
  wcd.wfcd_level, wfcn_wfcp_param
  SELECT
  wfcm_id, wfcm_facm_id, wcd.wfcd_approve_by, wcd.wfcd_approve_by_grp,
  wcd.wfcd_approve_by_role, wcd.wfcd_level, wcd.wfcd_approve_send_msg,
  wfcn_wfcp_param, um.usr_name
FROM
  mw_workflow_config_details wcd
JOIN
  mw_workflow_config_master wcm ON wcd.wfcd_wfcm_id = wcm.wfcm_id
JOIN
  mw_workflow_conditions wfc ON (SELECT unnest(wcd.wfcd_condition_desc)) = wfc.wfcn_id
JOIN
  users_m um ON um.usr_id = ANY(wcd.wfcd_approve_by)
WHERE
  wcm.wfcm_workflow_for = 'SP'
ORDER BY
  1, 6 ASC;
```

To Delete last(only) meter reading from schedule:

Delete from meterreadinghistory where mrh_id in (1021576,1080655)

update meterreadingmaster set mrm_status = 'Assigned' where mrm_id in (1172575,1252156)

To add related store:

select * from facility_link_stores where flkst_id =1630
insert into facility_link_stores (flkst_id,
 flkst_facm_id,flkst_link_stores,flkst_createdtime,flkst_modifiedtime,flkst_createduser_id,flkst_modifi
 eduser_id,flkst_linkstores_id) values (1630,1861,'TamilNadu Facility',Now(xx),Now(xx),69,69,1858)

To find Labour work hours:

—Scheduled:

select (worlbt_assignedto),(worlbt_actualefforts) from workordermaster left join workorderlabortasks on worm_id=worlbt_worm_id where cast(worlbt_completed_datetime as date)='2021-11-25' and worm_type='Preventive'

—Unscheduled:

select (worlbt_assignedto),(worlbt_actualefforts) from workordermaster left join workorderlabortasks on worm_id=worlbt_worm_id where cast(worlbt_completed_datetime as date)='2021-11-25' and worm_type!='Preventive'

To deactivate Schedule with Equipment:

select eqpgi_atlocation, eqpm_code, eqpm_id from equipmentgenerelinfo left join equipmentmaster on eqpm_id = eqpgi_eqpm_id where eqpgi_atlocation = 608 and eqpm_code like 'PGL%'

select * from workbenchmaster where wrbnm_active_status = true and wrbnm_facm_id = 608 and wrbnm_eqpm_id not in (xx)

select * from workordermaster where worm_status not in ('Closed','Completed','Cancelled') and worm_facm_code = 'MYS-WPB-PKG' and worm_eqpm_code not like '%PGL%'

select * from scheduledmaintenancemaster where smm_activestatus = true and smm_facm_id =608 and smm_eqpm_id not in (xx)

Delete workordermaster where worm_id in (xx)

update workbenchmaster set wrbnm_active_status = false where wrbnm_id in (xx)

update scheduledmaintenancemaster set smm_activestatus = false where smm_id in (xx)
Dashboard setting:

--Overall config
INSERT INTO public.mw_custom_facility_grouping_config_m (cmfgcm_config_for,cmfgcm_conf_type,cmfgcm_group_ids,cmfgcm_group_codes,cmfgcm_createdtime,cmfgcm_modifiedtime,cmfgcm_createduser_id, cmfgcm_modifieduser_id) VALUES ('DASH','FAC','{xx}','{yy}',NULL,now(),1,1);

update mw_custom_facility_grouping_config_m set cmfgcm_group_ids = '{xx}' , cmfgcm_group_codes = '{yy}';

-- userwise config

Insert into mw_custom_first_level_facility_group_config_m (cmflfgcm_config_for, cmflfgcm_conf_type, cmflfgcm_user_ids, cmflfgcm_role_ids, cmflfgcm_group_ids, cmflfgcm_group_codes, cmflfgcm_facgrp_level, cmflfgcm_createdtime, cmflfgcm_modifiedtime,

cmflfgcm_createduser_id, cmflfgcm_modifieduser_id) values ('DASH', 'FAC', '{ID}', null, '{xx}', '{yy}', 1, now(), now(), 1, 1);

--Overall config, while new user created

select * from mw_facilitywise_module_config_m

PM Compliance
CO Compliance

Query set 1:

—WO

1. Select fn_getusername(worgi_assignedto), (worm_createdtime), (worgi_assignedto), (worcn_worm_id), (worcn_completedby), (worcn_completiondate) from workordermaster left join workordergenerelinfo on worm_id=worgi_worm_id left join workordercompletionnotes on worm_id = worcn_worm_id where worm_status = 'Completed' and worm_type = 'Preventive' and cast(worm_createdtime as date) >= '2021-01-01' and cast(worm_createdtime as date) <= '2021-12-09' and worm_woid like 'ATCJB%' and worm_failuredetails = 'PM completed';

2. update workordercompletionnotes set worcn_completedby = 'worgi_assignedto', worcn_completiondate = 'worm_createdtime' where worcn_worm_id = xx;

—Checklist

3. select fn_getusername(worgi_assignedto), (worm_createdtime), (worgi_assignedto), (worlbt_worm_id), (worlbt_assignedto), (worlbt_completed_datetime), (worlbt_complete), (worlbt_usrid) from workordermaster left join workordergenerelinfo on worm_id = worgi_worm_id left join workorderlabortasks on worm_id = worlbt_worm_id where worm_status! = 'Completed' and worm_type = 'Preventive' and cast(worm_createdtime as date) >= '2021-01-01' and cast(worm_createdtime as date) >= '2021-12-09' and worm_woid like 'ATCJB%' and worm_failuredetails = 'PM completed';

4. update workorderlabortasks set worlbt_complete = 't', worlbt_assignedto = 'worgi_assignedto(username)', worlbt_usrid = 'worgi_assignedto', worlbt_completed_datetime = 'worm_createdtime' where worlbt_worm_id = xx;

select * from workordermaster left join workordershiftdetails on worm_id = worsftd_worm_id where worm_status = 'Completed' and worm_type = 'Preventive' and cast(worm_createdtime as date)>='2021-01-01' and cast(worm_createdtime as date)<='2021-12-09' and worm_woid like 'ATCJB%' and worm_failuredetails='PM completed'

5. update workordershiftdetails set worsftd_end_shift_date='2020-04-13',worsftd_end_shift='SHIFTA',worsftd_closed_datetime='2020-04-13 06:30',worsftd_closed_shift='SHIFTA' where worsftd_id=3963;

6. insert into workorderhistory
 (worhis_worm_id,worhis_status_change,worhis_user,worhis_createdtime,worhis_modifiedtime,worhi
 s_createduser_id,worhis_modifieduser_id)
values (34483,'Cancelled','Dattataraya Gurme',now(xx) ,now(xx) ,69,69) ;

select * from workordercompletionnotes

update workordercompletionnotes set
 worcn_completedby='worgi_assignedto',worcn_completiondate='worm_createdtime' where
 worcn_worm_id=;

Query set 2:

```
with woshift_data_cte as (
select worsftd_worm_id, coalesce(worsftd_off_datetime, worsftd_created_datetime) comp_date_time
from workordershiftdetails
),
wodata as (
select worm_id, worm_woid, worm_facm_id, worm_eqpm_id, worm_createdtime, current_date
 compdate,
fn_get_shiftdate_data(worm_facm_id, worm_eqpm_id, cast(to_char(comp_date_time,'YYYY') as
 integer),
to_char(comp_date_time,'Mon-YYYY'), to_char(comp_date_time,'dd-Mon-YYYY HH24:MI'))
 compsftdata
from woshift_data_cte, workordermaster
where worsftd_worm_id = worm_id
and worm_status not in ('Closed', 'Completed')
--and (worm_type like 'Preventive%' or worm_createfrom not like 'sm%')
--and cast(worm_createdtime as date) >= '2018-08-01'
--and cast(worm_createdtime as date) <= '2018-12-31'
and worm_id in (xx)
)select worm_id, worm_woid, worm_createdtime, compdate,
 (compsftdata).shiftdate,(compsftdata).shifttype
from wodata;
```

_____

----Shift data Update

```
update workordershiftdetails set
 worsftd_end_shift_date='2020-04-13',worsftd_end_shift='SHIFTA',worsftd_closed_datetime='2020-0
 4-13 06:30',worsftd_closed_shift='SHIFTA' where worsftd_id=3963;
```

_____

---WO Master Update

```
with woshift_data_cte as (
select worsftd_worm_id, coalesce(worsftd_off_datetime, worsftd_created_datetime) comp_date_time
from workordershiftdetails
),
wdata as (
select worm_id woid, worm_woid, worm_facm_id, worm_eqpm_id, worm_createdtime,
 comp_date_time compdate,
fn_get_shiftdate_data(worm_facm_id, worm_eqpm_id, cast(to_char(comp_date_time,'YYYY') as
 integer),
to_char(comp_date_time,'Mon-YYYY'), to_char(comp_date_time,'dd-Mon-YYYY HH24:MI'))
 compsftdata
from woshift_data_cte, workordermaster
where worsftd_worm_id = worm_id
and worm_status not in ('Closed', 'Completed')
--and (worm_type like 'Preventive%' or worm_createfrom not like 'sm%')
--and cast(worm_createdtime as date) >= '2018-08-01'
--and cast(worm_createdtime as date) <= '2018-12-31'
and worm_id in (xx)
)update workordermaster
set worm_status = 'Cancelled',worm_reasonforcancel='Cancelled work order due to shut down of plant
 because of corona virus lockdown.' from wdata
where worm_id = woid;
```

_____

Then finally update the completed status in workorderhistory table

```
insert into workorderhistory
 (worhis_worm_id,worhis_status_change,worhis_user,worhis_createdtime,worhis_modifiedtime,worhi
 s_createduser_id,worhis_modifieduser_id)
values (34483,'Cancelled','Dattataraya Gurme',now(xx) ,now(xx) ,69,69 ) ;
```

PM CBM Import

```
update scheduledmaintenancemaster set smm_cm_wotype = 'Preventive' where smm_id in(1,2)
```

```
update workordermaster set worm_type = 'Preventive' where worm_id in (1,2)
```

To config facility to handover

update mw_custom_first_level_facility_group_config_m set
 cmflfgcm_group_ids='{}',cmflfgcm_group_codes='{}' where cmflfgcm_user_ids ='{}'

update mw_custom_facility_grouping_config_m set cmfgcm_group_ids='{}'
 ,cmfgcm_group_codes='{}'

update mw_facilitywise_module_config_m set mwfwmcm_facm_ids= '{}' where mwfwmcm_id in
 (37,4) –PM,CO

DB ->Schema -> Public -> function -> search fn_update_shifthandover_notes_summary -> Source ->
 comment ("RAISE EXCEPTION 'Nonexistent ID --> %', sqlqry;")

To Update Facility Code:

update facilitymaster set facm_code = 'xx' where facm_id = 12 ;
update workordermaster set worm_facm_code = facm_code from facilitymaster where worm_facm_id
 = facm_id;

To Delete Facility:

Delete from facilitycustomfields **where** faccf_facm_id **in** (*xx*);
Delete from facilitydocuments **where** facdc_facm_id **in** (*xx*);
Delete from facilityevents **where** facev_facm_id **in** (*xx*);
Delete from facilityfailurecodes **where** facficd_facm_id **in** (*xx*);
Delete from facilitygenerelinfo **where** facgi_facm_id **in**(*xx*);
Delete from facilityitemsattention **where** facita_facm_id **in** (*xx*);
Delete from facilitylocation **where** facln_facm_id **in** (*xx*);
Delete from facilitymaintenance **where** facman_facm_id **in** (*xx*);
Delete from facilitymeterreading **where** facmtrd_facm_id **in** (*xx*);
Delete from facilityderivedmeterreadingmaster **where** fdmr_facm_id **in** (*xx*);
Delete from facilitynotification **where** facnt_facm_id **in** (*xx*);
Delete from facilityonofftoggel **where** fcoftg_facm_id **in** (*xx*);
Delete from facilityoperations **where** facops_facm_id **in** (*xx*);
Delete from facilitysafetypermit **where** facsfpmt_facm_id **in** (*xx*);
Delete from facilitytrainingcertificate **where** factrcer_facm_id **in** (*xx*);
Delete from facilityvendors **where** facven_facm_id **in** (*xx*);
Delete from facility_link_stores **where** flkst_facm_id **in** (*xx*);
Delete from general_shifthandover_notes_summary **where** gshons_gshonm_id **in** (**Select** gshonm_id
from general_shifthandover_notes_master **where** gshonm_facm_id **in** (*xx*));
Delete from general_shifthandover_notes_master **where** gshonm_facm_id **in** (*xx*);
Delete **from** users_facility **where** usrfac_facm_id **in** (*xx*);
Delete from costcentre_facilitywise_m **where** cc_fac_facility_id **in** (*xx*);
Delete from facilitymaster **where** facm_id **in** (*xx*);

To Delete WorkOrder:

select string_agg(worm_id::text,',') from workordermaster w where worm_woid in ();

Delete FROM workorsafetyderdocuments **where** worsfdc_worm_id **in** (xx);
Delete FROM workordervendor **where** wovn_worm_id **in** (xx);
Delete FROM workordertools **WHERE** wotl_worm_id **in** (xx);
Delete FROM workordersafetyjob **where** worsfjb_worm_id **in** (xx);
Delete FROM workordersafetyequipment **WHERE** worsfeqp_worm_id **in** (xx);
Delete FROM workorderrootcause **WHERE** worrc_worm_id **in** (xx);
Delete FROM workorderrccorrectiveaction **WHERE** worrcca_worm_id **in** (xx);
Delete From workorderrcpreventiveactive w **where** worcpa_worm_id **in** (xx);
Delete FROM workorderprogressupdate **WHERE** worpu_worm_id **in** (xx);
Delete FROM workorderpartstool **WHERE** woprtl_worm_id **in** (xx);
Delete FROM workorderparts **WHERE** worprt_worm_id **in** (xx);
Delete FROM workorderonofftoggel **WHERE** wooftg_worm_id **in** (xx);
Delete FROM workordernotification **where** wordc_worm_id **in** (xx);
Delete FROM workordermeterreading **WHERE** wormtr_worm_id **in** (xx);
Delete FROM workorderlabortasks **WHERE** worlbt_worm_id **in** (xx);
Delete FROM workorderitemforattention **WHERE** woria_worm_id **in** (xx);
Delete FROM workorderhistory **WHERE** worhis_worm_id **in** (xx);
Delete FROM workordergenerelinfo **WHERE** worgi_worm_id **in** (xx);
Delete FROM workorderdocuments **WHERE** wordc_worm_id **in** (xx);
Delete FROM workordercompletionnotes **WHERE** worcn_worm_id **in** (xx);
Delete FROM avail_ttr_tbf_data **where** attd_eqoftg_id **in** (**select** eqoftg_id **from** equipmentonofftoggel **where** eqoftg_worm_id **in** (xx));
Delete FROM equipmentonofftoggel **where** eqoftg_worm_id **in** (xx);
Delete FROM workorder_offhistory **where** wo_off_worm_id **in** (xx);
Delete FROM workorder_onhistory **where** wo_on_worm_id **in** (xx);
Delete from workordershiftdetails w **where** worsftd_worm_id **in** (xx);
Delete FROM workordermaster **WHERE** worm_id **in** (xx);

To Cancel WorkOrder:

select * from workordercompletionnotes where worcn_worm_id =xx;

UPDATE public.workordercompletionnotes
SET worcn_completedby='xx', worcn_completiondate='dd' WHERE worcn_worm_id in (xx)

select * from workordershiftdetails where worsftd_worm_id =xx;

UPDATE public.workordershiftdetails
SET worsftd_closed_datetime='dd', worsftd_closed_shift='SHIFTA',
 worsftd_end_shift_date='2023-05-08', worsftd_end_shift='SHIFTA'
WHERE worsftd_worm_id in (xx);

select * from workorderhistory where worhis_worm_id =xx;

INSERT INTO public.workorderhistory
(worhis_worm_id, worhis_status_change, worhis_user, worhis_createdtime, worhis_modifiedtime,
 worhis_createduser_id, worhis_modifieduser_id, worhis_duration_status,
 worhis_max_permissible_duration, worhis_reason_for_delay)VALUES(xx, 'Cancelled', 'RAMESH
 ITHEPALLI', now(), now(), 969, 969, NULL, NULL, NULL);

update workordermaster set worm_status ='Cancelled',worm_reasonforcancel = 'Cancelled' where
 worm_id in (xx);

To Delete WorkBench:

Delete from workbenchcompletionnotes WHERE wrbncn_wrbnm_id In(xx) ;
Delete from workbenchgenerelinfo WHERE wrbngi_wrbnm_id In(xx);
Delete from workbenchhistory WHERE wrbnhis_wrbnm_id In(xx);
Delete from workbenchitemforattention WHERE wrbnia_wrbnm_id In(xx);
Delete from workbenchlabortasks WHERE wrbnlbt_wrbnm_id In(xx);
Delete from workbenchmeterreading WHERE wrbnmtr_wrbnm_id In(xx);
Delete from workbenchnotification WHERE wrbnnt_wrbnm_id In(xx);
Delete from workbenchonofftoggel WHERE wrbnoftg_wrbnm_id In(xx);
Delete from workbenchparts WHERE wrbnprt_wrbnm_id In(xx);
Delete from workbenchpartstool WHERE wrbnprtl_wrbnm_id In(xx);
Delete from workbenchprogressupdate WHERE wrbnpu_wrbnm_id In(xx);
Delete from workbenchrccorrectiveaction WHERE wrbnrcca_wrbnm_id In(xx);
Delete from workbenchrootcause WHERE wrbnrc_wrbnm_id In(xx);
Delete from workbenchsafetyderdocuments WHERE wrbnsfdc_wrbnm_id In(xx);
Delete from workbenchsafetyequipment WHERE wrbnsfeqp_wrbnm_id In(xx);
Delete from workbenchsafetyjob WHERE wrbnsfjb_wrbnm_id In(xx);
Delete from workbenchtools WHERE wrbntl_wrbnm_id In(xx);
Delete from workbenchvendor WHERE wrbnvn_wrbnm_id In(xx);
Delete from workbenchmaster where wrbnm_id In(xx);

To Delete schedule:

Delete FROM schedulemainttools where smtl_smm_id in (xx);
Delete FROM schedulemaintpartstool where smprtl_smm_id in (xx);
Delete FROM schedulemaintparts where smprt_smm_id in (xx);
Delete FROM schedulemaintlabourtask where smlt_smm_id in (xx);
Delete FROM scheduledmaintenancevendor where smvn_smm_id in (xx);
Delete FROM scheduledmaintenancegenerelinfo WHERE smgi_smm_id in (xx);
Delete FROM sm_cbm_schedule_all WHERE smcbmsa_smm_id in (xx);
Delete FROM sm_cbm_schedule_latest where smcbmsl_smm_id in (xx);
Delete FROM sm_cbm_scheduler where smcbms_smm_id in (xx);
Delete FROM sm_ebm_scheduler where smebms_smm_id in (xx);
Delete FROM sm_tbm_schedule_all WHERE smtbmsa_smm_id in (xx);
Delete FROM sm_tbm_schedule_latest where smtbmsl_smm_id in (xx);
Delete FROM sm_tbm_scheduler where smtbms_smm_id in (xx);

Delete FROM scheduledmaintenancemaster where smm_id in (xx);

To Delete User:

Delete from users_dashboard_settings where usrds_usr_id in (86,88);
Delete from users_facility where usrfac_usr_id in (86,88);
Delete from usergroup where urgr_usr_id in (86,88);
Delete from usersaccounts where uract_usr_id in (86,88);
Delete from users_m where usr_id in (86,88);

To Delete resource:

Delete from additionalresource_facility where arfac_arm_id = 49;
Delete from additionalresourcemaster where arm_id = 49;
Delete from additionalresourceskills where ars_arm_id = 49;

additionalresource_pastEmpHis
additionalresource_eduqualification
additionalresource_certification
additionalresource_credentials
additionalresource_licenses
additionalresource_workhis
additionalresource_healthhis
additionalresource_incidenthis
additionalresourceskills_master
additionalresource_training
additionalresource_trainertraining
additionalresource_extrnaltraining
additionalrolesresponsibility_master

To Delete Parts:

Delete from partsgenerelinfo where prtgi_prtm_id in (xx);
Delete from partsdocuments where prtdc_prtm_id in (xx);
Delete from partshistory where prthis_prtm_id in (xx);
Delete from partsinimumstocklevel where prtmsl_prtm_id in (xx);
Delete from partslocation where prtln_prtm_id in (xx);
Delete from partsnotification where prtnt_prtm_id in (xx);
Delete from partsreceipts where prtrc_prtm_id in (xx);
Delete from partsvendors where pacven_pacm_id in (xx);
Delete from partslocationhistory where prtlochis_prtm_id in (xx);
Delete from partsmaster where prtm_id in (xx);
Delete from partsinimumstocklevel where prtmsl_prtm_id in (xx);

To Update part code:

Update partsgenerelinfo set prtgi_prtm_code = 'zz' where prtgi_prtm_id in (xx);
Update partsdocuments set prtdc_prtm_code = 'zz' where prtdc_prtm_id in (xx);
Update partshistory set prthis_prtm_code = 'zz' where prthis_prtm_id in (xx);
Update partsinimumstocklevel set prtmsl_prtm_code = 'zz' where prtmsl_prtm_id in (xx);
Update partslocation set prtln_prtm_code = 'zz' where prtln_prtm_id in (xx);
Update partsnotification set prtnt_prtm_code = 'zz' where prtnt_prtm_id in (xx);
Update partsreceipts set prtrc_prtm_code = 'zz' where prtrc_prtm_id in (xx);
Update partslocationhistory set prtlochis_prtm_code = 'zz' where prtlochis_prtm_id in (xx);
Update partsmaster set prtm_code = 'zz' where prtm_id in (xx);
Update partsinimumstocklevel set prtmsl_prtm_code = 'zz' where prtmsl_prtm_id in (xx);

Delete Instrument SO:

Delete FROM ins_so_schedule_all where ins_soshall_insos_smm_id='SM-ISO002148';
Delete FROM ins_serviceorderschedule where insos_id=210;
Delete FROM ins_so_scheduler where insso_sch_insos_smm_id='SM-ISO002148';
Delete FROM ins_so_schedule_latest where ins_soshlat_insos_smm_id='SM-ISO002148';

Delete Equipment SO:

Delete from serviceorderschedule where sos_id in (2360);
Delete from so_scheduler where so_som_id in (2360);
Delete from so_schedule_latest where soshlat_som_id in (2360);
Delete from so_schedule_all where soshall_som_id in (2360);

To Delete SIR:

Delete from stockitemrequest where sir_reqid = 'xx' ;
Delete from stockitemrequest_approval_details where sirad_req_id = 'xx' ;
Delete from stockitemrequest_review_details where sirrd_req_id = 'xx' ;
Delete from stockitemrequestdocuments where sirdc_req_id = 'xx' ;

To Delete MMR:

DELETE FROM opnmechinemodifirequest WHERE ommr_reqid = 'value';
DELETE FROM opnmechinemodifireqdocuments WHERE ommrdc_req_id = 'value';
DELETE FROM openmechinemodireq_approval_details WHERE ommrad_req_id = 'value';
DELETE FROM openmechinemodireq_initialapproval_details WHERE ommriad_req_id = 'value';
DELETE FROM openmechinemodireq_mainapproval_details WHERE ommrmad_req_id = 'value';

To Delete CSO order:

Delete from customserviceorder_usr_feilds where cso_usr_cso_m_id = xx;
Delete from customserviceorder_text_feilds where cso_text_cso_m_id = xx;
Delete from customserviceorder_equipment_fields where cso_equipment_cso_m_id = xx;
Delete from customserviceorder_dt_fields where cso_dt_cso_m_id = xx;

Delete from customserviceorder_dropdown_feilds where cso_dropdown_cso_m_id = xx;
Delete from customserviceorder_documents where csom_doc_cso_m_id = xx;
Delete from customserviceorder_checklist where cso_checklist_cso_m_id = xx;
Delete from cso_historydetails where csh_cso_m_id = xx;
Delete from customserviceordermaster where cso_m_id = xx;

RCA Condition and flow:

lsthrs - WO Lost hours > n hours for a WO
monlsthrs - WO Lost hours > n hours for a Month for a Equipment
monoccurance. - Occurrences > n nos for a Month for a Equipment
Issue - Issue Occurrenes > n nos for a Month for a Equipment
parts. - Parts Price > n price for a WO
manuchk - Manual Check

Weekly Customer Usage:

select * from workordermaster where worm_gen_from = 'wor' and worm_type = 'Breakdown' and
 cast(worm_createdtime as date)>='2022-07-24' and worm_woid like 'SAVW-AGB%'

select * from workordermaster where worm_gen_from = 'wo' and worm_type = 'Breakdown' and
 cast(worm_createdtime as date)>='2022-07-24' and worm_woid like 'SAVW-AGB%'

select * from workorderparts left join workordermaster on worprt_worm_id = worm_id where
 worm_type = 'Breakdown' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
 'SAVW-AGB%'

select * from workorderlabortasks left join workordermaster on worlbt_worm_id = worm_id where
 worm_type = 'Breakdown' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
 'SAVW-AGB%'

select * from workordervendor left join workordermaster on wovn_worm_id = worm_id where
 worm_type = 'Breakdown' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
 'SAVW-AGB%'

select * from scheduledmaintenancemaster where smm_smid like 'SAVW-AGB%' and
 smm_activestatus ='t'

select * from workbenchmaster where wrbnm_type = 'Preventive' and cast(wrbnm_createdtime as date)
 >= '2022-07-24' and wrbnm_wbid like 'SAVW-AGB%' and wrbnm_active_status = 't'

select * from workordermaster where worm_type = 'Preventive' and cast(worm_createdtime as
 date)>='2022-07-01' and worm_woid like 'SAVW-AGB%'

```sql
select * from workorderparts left join workordermaster on worprt_worm_id = worm_id where
worm_type = 'Preventive' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
'SAVW-AGB%'

select * from workorderlabortasks left join workordermaster on worlbt_worm_id = worm_id where
worm_type = 'Preventive' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
'SAVW-AGB%' and worlbt_actualhours > '00:00'

select * from workordervendor left join workordermaster on wovn_worm_id = worm_id where
worm_type = 'Preventive' and cast(worm_createdtime as date)>='2022-07-24' and worm_woid like
'SAVW-AGB%'

select * from calibrationprocessmaster where calpm_serial_id like 'SAVW-AGB%' and
calpm_activestatus = 't'

select * from coworkbenchmaster where cowrbnm_calpm_serial_id like 'SAVW-AGB%' and
cowrbnm_isactive ='t'

select * from calibrationordermaster where calom_serial_id like 'SAVW-AGB%' and
cast(calom_createdtime as date)>='2022-07-01'

select * from calibrationordervendor where calomvn_calom_serial_id like 'SAVW-AGB%'

select * from serviceordermaster where som_facm_code like '3_3%'

select * from instrumentserviceordermaster where inssom_facm_code like '3_3%'

select * from instrumentwomaster where inswom_facm_code like '3_3%'

select * from workordermaster where worm_type not in ('Preventive','Breakdown') and
cast(worm_createdtime as date)>='2022-07-24' and worm_woid like 'SAVW-AGB%'

select * from workorderparts left join workordermaster on worprt_worm_id = worm_id where
worm_type not in ('Preventive','Breakdown') and cast(worm_createdtime as date)>='2022-07-24' and
worm_woid like 'SAVW-AGB%'

select * from workorderlabortasks left join workordermaster on worlbt_worm_id = worm_id where
worm_type not in ('Preventive','Breakdown') and cast(worm_createdtime as date)>='2022-07-24' and
worm_woid like 'SAVW-AGB%' and worlbt_actualhours > '00:00'

select * from workordervendor left join workordermaster on wovn_worm_id = worm_id where
worm_type not in ('Preventive','Breakdown') and cast(worm_createdtime as date)>='2022-07-24' and
worm_woid like 'SAVW-AGB%'

select * from workordermaster where worm_type = 'Preventive' and worm_status not in
('Closed','Completed','Cancelled') and worm_woid like 'SAVW-AGB%'
```

select * from oppmaintenancetrack where omt_facm_code like '3_3%'

select * from meterreadingschedule where mrsd_facm_code like '3_3%' and mrsd_activestatus ='t'

select * from meterreadingmaster where mrm_facm_code like '3_3%'

select * from conditionalmonitorschedule where cbms_facm_code like '3_3%' and cbms_activestatus ='t'

select * from conditionalmonitormaster where cm_mom_id like 'SAVW-AGB%'

select * from cbm_scheduler where cbm_cbms_smid not like '%CM%'

select * from partshistory where prthis_facm_code like '3_3%' and cast(prthis_createdtime as date) >='2022-07-24'

select eqpgi_atlocationname,* from equipmentcriticalparts2 left join equipmentgenerelinfo on eqpgi_eqpm_id = eqpmtcp_eqpm_id

select * from partsinimumstocklevel where prtmsl_minimumstocklevel > '0'

To change equipment status as Deleted:

select count(eqpm_id),**string_agg**(eqpm_id::**text**,',') **from** equipmentmaster **where** eqpm_code **in** (***EQ-Code***);
update equipmentmaster **set** eqpm_equipmentclass ='Deleted' **where** eqpm_id **in** (EQ-ID);
update scheduledmaintenancemaster **set** smm_activestatus ='f' **where** smm_eqpm_id **in** (EQ-ID) **and** smm_activestatus = 't';
update workbenchmaster **set** wrbnm_active_status ='f' **where** wrbnm_eqpm_id **in** (EQ-ID) **and** wrbnm_active_status ='t';
update meterreadingschedule **set** mrsd_activestatus ='f',mrsd_select_sch_meterids='' **where** mrsd_eqpm_id **in** (EQ-ID) **and** mrsd_activestatus ='t';
update conditionalmonitorschedule **set** cbms_activestatus ='f', cbms_select_sch_meterids ='' **where** cbms_eqpm_id **in** (EQ-ID) **and** cbms_activestatus ='t';
update serviceorderschedule **set** sos_activestatus ='f' **where** sos_eqpm_id **in** (EQ-ID) **and** sos_activestatus ='t';
select string_agg(worm_id::**text**,','),**string_agg**(worm_woid::**text**,'','') **from** workordermaster *w* **where** worm_eqpm_id **in** (***EQ-ID***) **and** worm_status **not in** ('Closed','Completed','Cancelled');

To deactivate WO schedule:

select smm_activestatus,* from scheduledmaintenancemaster s where smm_activestatus ='t' and smm_smid in (SM-ID);
update scheduledmaintenancemaster set smm_activestatus ='f' where smm_smid in (SM-ID);

select string_agg(wrbnm_id::text,',') from workbenchmaster w where wrbnm_som_smm_id in (SM-ID)
 and wrbnm_active_status = 't';
update workbenchmaster set wrbnm_active_status ='f' where wrbnm_som_smm_id in (SM-ID);
select string_agg(worm_id::text,','),string_agg(worm_woid::text,''','') from workordermaster w where
 worm_som_smm_id in (SM-ID) and worm_status not in ('Closed','Completed','Cancelled');

To deactivate ConditionMonitor Schedule:

select cbms_activestatus,* from conditionalmonitorschedule c where cs_mom_id in (SM-CM-ID);
update conditionalmonitorschedule set cbms_activestatus = 'f', cbms_select_sch_meterids ='' where
 cs_mom_id in (SM-CM-ID);
select string_agg(cbmm_id::text,',') from conditionalmonitormaster c where cbmm_cs_mom_id in
 (SM-CM-ID) and cbmm_createdtime :: date >= 'date';
update conditionalmonitormaster set cbmm_status = 'Closed' where cbmm_id in (CBMM-ID);

To delete generated ConditionMonitor order:

Delete from conditionalmonitormaster c where cbmm_id in (CBMM-ID);
Delete from condmeterreadinghistory c where cmrh_cbmm_id in (CBMM-ID);

To deactivate Calibration schedule:

select calpm_activestatus,* from calibrationprocessmaster c where calpm_serial_id in ('SM-CO');
update calibrationprocessmaster set calpm_activestatus = 'f' where calpm_serial_id in ('SM-CO');
select string_agg(cowrbnm_wbid::text,''','') from coworkbenchmaster c where
 cowrbnm_calpm_serial_id in ('SM-CO');
update coworkbenchmaster set cowrbnm_isactive = 'f' where cowrbnm_wbid in ('CO-WB');
select count(calom_id),string_agg(calom_serial_id::text,''','') from calibrationordermaster c where
 calom_calpm_serial_id in ('CO-WB') and calom_status not in ('Closed','Completed','Cancelled');
update calibrationordermaster set calom_status = 'Cancelled' where calom_serial_id in('CO');

To change equipment location:

update equipmentgenerelinfo set eqpgi_atlocation = F-iD,eqpgi_atlocationname ='F-NaMe' where
 eqpgi_eqpm_code = 'CoDe'; -- not needed if facility change done in frontend.
UPDATE scheduledmaintenancemaster s SET smm_facm_id = eqpgi_atlocation from
 equipmentgenerelinfo e WHERE s.smm_eqpm_id = eqpgi_eqpm_id and s.smm_activestatus = 't';
UPDATE workbenchmaster w SET wrbnm_facm_id = eqpgi_atlocation from equipmentgenerelinfo e
 WHERE w.wrbnm_eqpm_id = eqpgi_eqpm_id and w.wrbnm_active_status = 't';
UPDATE workordermaster w SET worm_facm_id = eqpgi_atlocation from equipmentgenerelinfo e
 WHERE w.worm_eqpm_id = eqpgi_eqpm_id and w.worm_status not in
 ('Closed','Completed','Cancelled'); –not needed if client asked for.
update meterreadingschedule m SET mrsd_facm_id = eqpgi_atlocation from equipmentgenerelinfo e
 WHERE m.mrsd_eqpm_id = eqpgi_eqpm_id and mrsd_activestatus ='t';
update conditionalmonitorschedule c SET cbms_facm_id = eqpgi_atlocation from
 equipmentgenerelinfo e WHERE c.cbms_eqpm_id = eqpgi_eqpm_id and cbms_activestatus ='t';

update serviceorderschedule s SET sos_facm_id = eqpgi_atlocation from equipmentgenerelinfo e
  WHERE s.sos_eqpm_id = eqpgi_eqpm_id and sos_activestatus ='t';

To delete reservation ID:

select * from workorder_workflow wsw where wospri_worm_id in (WO-ID) and
  wospri_sap_reservation_id in ('ResID')
delete from workorder_spareparts_workflow where wospwf_worm_id = WO-ID and
  wospwf_reservation_id = 'ResID';
delete from workorder_spareparts_reservation_info where wospri_worm_id = WO-ID and
  wospri_sap_reservation_id = 'ResID';
delete from workorder_partsissue_return where wopir_worm_id = WO-ID and
  wopir_worprt_reservation_id = 'ResID';
delete from workorderparts where worprt_worm_id = WO-ID and worprt_reservation_id = 'ResID';

To delete schedule checklist:

select * from scheduledmaintenancemaster where smm_smid in (SM-ID-CL);
select * from schedulemaintlabourtask where smlt_smm_smid in (SM-ID-CL);
delete from schedulemaintlabourtask where smlt_smm_smid in (SM-ID-CL);

Schedule deactivate for Testing server:

update scheduledmaintenancemaster set smm_activestatus ='f';
update workbenchmaster set wrbnm_active_status ='f' ;
update conditionalmonitorschedule set cbms_activestatus = 'f', cbms_select_sch_meterids ='' ;
update calibrationprocessmaster set calpm_activestatus = 'f' ;
update coworkbenchmaster set cowrbnm_isactive = 'f' ;
update meterreadingschedule set mrsd_activestatus ='f',mrsd_select_sch_meterids='';
update serviceorderschedule set sos_activestatus ='f';
update ins_serviceorderschedule set insos_activestatus ='f';

Data Templates:

1. Facilities Master
2. User Management
3. Equipment Master
4. PM Schedule
5. PM Checklist
6. Parts Master
7. Equipment Issue Code
8. General Issue Code
9. Instrument Master
10. Calibration Schedule

11. Equipment Meters
12. Condition Monitoring Parameters
13. CBM Schedules

Approval:

RCA Approval - Full User, Combo User
PM & CO Compliance Report Approval - Full User, Combo User
WO Parts Approval - Full User, Combo User
PM WO Workflow Approval - Full User, Combo User
CO Workflow Approval - Full User
Safety Permit Approval - Full User, Combo User
Asset Mgmt Approval - Full User, Combo User
SIR, SICR and MMR - Full User, Combo User
WB Preactor - Full User

WO Sequence ID Correction:

select * from mw_sequence_id_m where mwseq_initialchar='SM'
update mw_sequence_id_m set mwseq_serialvalue='217587' where mwseq_initialchar='SM'

Query to update data from one table to another table:

UPDATE additionalresourceskills
SET ars_arm_resourceid = additionalresourcemaster.arm_resourceid
from additionalresourcemaster WHERE ars_arm_id = additionalresourcemaster.arm_id;

INSERT INTO additionalresourceskills (ars_arm_id, ars_arm_resourceid)
SELECT arm_id, arm_resourceid
FROM additionalresourcemaster
WHERE arm_id NOT IN (SELECT ars_arm_id FROM additionalresourceskills);

New facility addition:

select * from facilitymaster order by 1 desc;

Shift configuration
Transaction numbering

select * from mw_dm_transaction_numbering_m;

with test as(
 select unnest(mwdmtnm_parentchildfacm_id) as fid from mw_dm_transaction_numbering_m order by
 1 asc
)select * from facilitymaster where facm_id not in (select fid from test);

To find duplicate in Transaction Numbering of a facility ID:

```
with cte as(
        select unnest(mwdmtnm_parentchildfacm_id) as fid, * from
mw_dm_transaction_numbering_m
        ) select * from cte where cte.fid in
                (select unnest(mwdmtnm_parentchildfacm_id) as fid from
mw_dm_transaction_numbering_m
                group by unnest(mwdmtnm_parentchildfacm_id)
                having count(*) > 1)

update mw_dm_transaction_numbering_m set mwdmtnm_parentchildfacm_id =
 mwdmtnm_parentchildfacm_id ||'{xx}' where mwdmtnm_id = xz;
```

CC and order number:

```
select * from costcentre_facilitywise_m order by 3 desc;

INSERT INTO public.costcentre_facilitywise_m
(cc_fac_costcentre, cc_fac_facility_id, cc_fac_isdefault, cc_fac_createdtime, cc_fac_modifiedtime,
 cc_fac_createduser_id, cc_fac_modifieduser_id)
VALUES('CC',facmID, false, now(), now(), 1, 1);

select * from costcenter_orderno where cco_orderno = '';
```

User management - facility access
Store facility config

```
select * from facility_link_stores order by 2 desc;

INSERT INTO public.facility_link_stores
(flkst_facm_id, flkst_link_stores, flkst_createdtime, flkst_modifiedtime, flkst_createduser_id,
 flkst_modifieduser_id, flkst_linkstores_id)
VALUES(facmID, 'StoreName', now(),now(), 1, 1, storeID);
```

Approval Workflow

```
select * from facilitymaster f where facm_code = '1007-BU5';

select * from facilitymaster f where facm_id in ();

select * from mw_workflow_config_master mwcm where wfcm_facm_id like '%925%';

select * from mw_workflow_config_details mwcd where wfcd_wfcm_id = 148;

select * from mw_workflow_config_details mwcd where wfcd_approve_by::text like '%159%';
```

```sql
update workorder_spareparts_workflow set wospwf_approve_usrid = wfcd_approve_by from
 mw_workflow_config_details where wospwf_wfcd_id = wfcd_id and wospwf_approved_status =
 'Waiting for Approval';

select * from mw_workflow_conditions mwc where wfcn_id = 209;

select * from mw_workflow_config_master m,mw_workflow_config_details mwcd where
 wfcm_workflow_for like 'SP%' and wfcd_wfcm_id = m.wfcm_id;
```

PM, CP Planning Horizon
Mandatory fields update (Issue code)
RCA facility selection (escalation condition)

```sql
select string_agg(facm_id::text,','),string_agg(facm_name::text,',') from facilitymaster;

select * from Mw_custom_first_level_facility_group_config_m; --User wise dash

select string_agg(usr_id::text,',')  from users_m um where usr_isactive ='Active' and usr_billingidentity
 ='Chennai' and usr_fac_access != 'LTU';

INSERT INTO mw_custom_first_level_facility_group_config_m
(cmflfgcm_config_for, cmflfgcm_conf_type, cmflfgcm_user_ids, cmflfgcm_role_ids,
 cmflfgcm_group_ids, cmflfgcm_group_codes, cmflfgcm_facgrp_level, cmflfgcm_createdtime,
 cmflfgcm_modifiedtime, cmflfgcm_createduser_id, cmflfgcm_modifieduser_id)
VALUES('DASH', 'FAC', '{1}', NULL, '{facm_id}','{facm_name}', 1, now(), now(), 1, 1);

select * from Mw_custom_facility_grouping_config_m; -- add all facility

INSERT INTO mw_custom_facility_grouping_config_m
(cmfgcm_config_for, cmfgcm_conf_type, cmfgcm_group_ids, cmfgcm_group_codes,
 cmfgcm_createdtime, cmfgcm_modifiedtime, cmfgcm_createduser_id, cmfgcm_modifieduser_id,
 cmfgcm_facgrp_level)
VALUES('DASH', 'FAC', '{facm_id}','{facm_name}', now(), now(), 1, 1, 0);

select * from mw_facilitywise_module_config_m; -- PM, CO
```

To create new menu:

```sql
select * from menu_m;
select * from submenu_m;
select * from mw_menumanagement_m;
select * from users_rolewise_menuaccess;

INSERT INTO public.menu_m (menu_topmenuname, menu_topmenuid, menu_createdtime,
 menu_modifiedtime,
```

```sql
    menu_createduser_id, menu_modifieduser_id, menu_available, menu_icon_class, menu_li_id,
 menu_order)
VALUES('TPM', 17, now(), now(), 1, 1, true, 'fas fa-user-shield', 'tpm-tm', 17);

--delete from submenu_m where smenu_topmenu_id = 17 and smenu_submenuid = 1;

INSERT INTO public.submenu_m (smenu_topmenu_id, smenu_submenuname, smenu_createdtime,
 smenu_modifiedtime, smenu_createduser_id, smenu_modifieduser_id,
 smenu_access_right, smenu_submenuid, smenu_page_link, smenu_icon_class, smenu_add_link,
 smenu_li_id,smenu_order)
Values (17,'Planned Maintenance',now(),now(),1,1,true,1,'/mymaintwiz/mwpagemenu/pm','fa
 fa-viadeo-square','','tpm-m1',1);
INSERT INTO public.submenu_m (smenu_topmenu_id, smenu_submenuname, smenu_createdtime,
 smenu_modifiedtime, smenu_createduser_id, smenu_modifieduser_id,
 smenu_access_right, smenu_submenuid, smenu_page_link, smenu_icon_class, smenu_add_link,
 smenu_li_id,smenu_order)
Values (17,'Jishu Hozen',now(),now(),1,1,true,2'/mymaintwiz/mwpagemenu/jh','fa
 fa-viadeo-square','','tpm-m2',2);
INSERT INTO public.submenu_m (smenu_topmenu_id, smenu_submenuname, smenu_createdtime,
 smenu_modifiedtime, smenu_createduser_id, smenu_modifieduser_id,
 smenu_access_right, smenu_submenuid, smenu_page_link, smenu_icon_class, smenu_add_link,
 smenu_li_id,smenu_order)
Values (17,'Kobetsu Kaizen',now(),now(),1,1,true,3,'/mymaintwiz/mwpagemenu/kk','fa
 fa-viadeo-square','','tpm-m3',3);
INSERT INTO public.submenu_m (smenu_topmenu_id, smenu_submenuname, smenu_createdtime,
 smenu_modifiedtime, smenu_createduser_id, smenu_modifieduser_id,
 smenu_access_right, smenu_submenuid, smenu_page_link, smenu_icon_class, smenu_add_link,
 smenu_li_id,smenu_order)
Values (17,'Quality Maintenance',now(),now(),1,1,true,4,'/mymaintwiz/mwpagemenu/qm','fa
 fa-viadeo-square','','tpm-m4',4);

--delete from mw_menumanagement_m where concat(mmmgt_topmenu_id,'_',mmmgt_submenu_id) in
 ('1_13');

INSERT INTO public.mw_menumanagement_m
 (mmmgt_la_code,mmmgt_topmenu_id,mmmgt_submenu_id,mmmgt_menuview,mmmgt_createdtime,
 mmmgt_modifiedtime,
mmmgt_createduser_id,mmmgt_modifieduser_id,mmmgt_menu_add,mmmgt_menu_edit,mmmgt_men
 u_delete,mmmgt_menu_view)
(select 'FU', smenu_topmenu_id, smenu_submenuid, true,now(),now(),1,1,true,true,true,true
 from submenu_m
 where concat(smenu_topmenu_id,'_',smenu_submenuid) in ('1_13')
 union
 select 'VOU', smenu_topmenu_id, smenu_submenuid,true,now(),now(),1,1,false,false,false,true
 from submenu_m
 where concat(smenu_topmenu_id,'_',smenu_submenuid) in ('1_13')
```

```
union
select 'CU', smenu_topmenu_id, smenu_submenuid, true,now(),now(),1,1,true,true,true,true
from submenu_m
where concat(smenu_topmenu_id,'_',smenu_submenuid) in ('1_13')
union
select 'LTU', smenu_topmenu_id, smenu_submenuid, true,now(),now(),1,1,true,true,true,true
from submenu_m
where concat(smenu_topmenu_id,'_',smenu_submenuid) in ('1_13')
)
;

delete from users_rolewise_menuaccess where concat(urwma_topmenu_id,'_',urwma_submenu_id) in
 ('1_13');
INSERT INTO public.users_rolewise_menuaccess
(urwma_roletype, urwma_user_id, urwma_topmenu_id, urwma_submenu_id, urwma_add, urwma_edit,
 urwma_delete, urwma_view, urwma_createdtime, urwma_modifiedtime, urwma_createduser_id,
 urwma_modifieduser_id,
 urwma_menuview, urwma_lc_code)
(
select rtm.rlty_type, rtm.rlty_id, mwmm.mmmgt_topmenu_id, mwmm.mmmgt_submenu_id,
 mwmm.mmmgt_menu_add, mwmm.mmmgt_menu_edit, mwmm.mmmgt_menu_delete,
 mwmm.mmmgt_menu_view,
 now(), now(), 1,1, mwmm.mmmgt_menuview, rtm.rlty_lc_access_code
from roletypes_m rtm
 inner join mw_menumanagement_m mwmm on mwmm.mmmgt_la_code =  rtm.rlty_lc_access_code
        and concat(mmmgt_topmenu_id,'_',mmmgt_submenu_id) in ('1_13')
order by 2,14,3,4
);
```

Check schedule moved from WB to WO:

```
select count(worm_id) from workordermaster where worm_createfrom = 'sm' and
 cast(worm_schedule_date as date) = current_date
select distinct   count(wrbnm_id) from workbenchmaster
left join workordermaster on worm_wrbnm_id = wrbnm_id
where wrbnm_createfrom = 'sm' and wrbnm_active_status = 't' and cast(wrbnm_schedule_date as date)
 = current_date+1
```

Completed notes error:
```
select w2.worm_modifieduser_id,* from workordercompletionnotes w,workordermaster w2 where
 worcn_completiondate isnull and worcn_worm_id =worm_id and w2.worm_status ='Completed'

update workordercompletionnotes set worcn_completiondate = w.worm_modifiedtime from
 workordermaster w where worcn_completiondate is null and worcn_worm_id =worm_id and
 w.worm_status ='Completed'
```

select worcn_completedby, usr_name,* from workordercompletionnotes,users_m u,workordermaster w
 where u.usr_id = w.worm_modifieduser_id and worcn_completedby isnull and worcn_worm_id
 =worm_id and w.worm_status ='Completed'

update workordercompletionnotes set worcn_completedby = usr_name from users_m
 u,workordermaster w where u.usr_id = w.worm_modifieduser_id and worcn_completedby isnull and
 worcn_worm_id =worm_id and w.worm_status ='Completed'

Password:

select uract_usr_id, uract_loginname,uract_password,decode(uract_password,'base64') from
 usersaccounts where uract_usr_id = xx

SM Task Library Checklist update query:

select fn_insert_sm_tasklibrary('TL-000038','{SM036336,SM036337}')

Aggregate query:

SELECT STRING_AGG(wrbnm_id::text, ',') AS Concatenated_Values FROM workbenchmaster
 WHERE wrbnm_som_smm_id IN ('PER-SM003488') AND wrbnm_active_status = TRUE

SELECT STRING_AGG(facm_code, ',') AS concatenated_values FROM facilitymaster f WHERE
 f.facm_code LIKE '1001%'

Query for finding many schedule frequency:

with CTE as (
select smm_smid, f.facm_code, e.eqpm_code,smm_wocategory as
 Workorder_category,smm_natureofissue as Nature_of_Work ,smm_smdescription as SM_Description
 ,to_char(smtbms_tbm_rr_startdate, 'dd-mm-yyyy') as Start_Date ,smtbms_tbm_mn_dayof as Day ,
 smtbms_tbm_rpinterval as Every_Month,smtbms_tbm_wocompwithin as
 WO_should_be_completed_within_N_days
from scheduledmaintenancemaster s
join sm_tbm_scheduler sts on s.smm_id = sts.smtbms_smm_id
join equipmentmaster e on s.smm_eqpm_id = e.eqpm_id
join facilitymaster f on s.smm_facm_id = f.facm_id
where s.smm_activestatus = true
)

select * from CTE where eqpm_code in ('10050155')

Query need to test:

with CTE_1 as(
        with eqp_onoff as(

```sql
		select concat(eqoftg_eqpm_id,eqoftg_worm_id,onoff_status) as eqp_onoff_searchid
		,concat(eqoftg_eqpm_id,eqoftg_worm_id,onoff_status,'true') as
eqp_onoff_wo_on_searchid ,*
		from(
			select
			case when eqoftg_onoffstatus = true then 'true' else 'false' end as onoff_status,*
			from equipmentonofftoggel
			) as subquery_on_1
	),
	wo_on as(
		select concat(wo_on_eqpm_id,wo_on_worm_id,'true',wo_on_activestatus) as
wo_on_searchid,*
		from (
			select
			case when wo_onactivestatus = true then 'true' else 'false' end as
wo_on_activestatus,*
			from workorder_onhistory
			) as subquery_on_2
	)
select eqoftg_eqpm_id as eqp_id_on,eqoftg_worm_id as wo_onhistory_notinserted_worm_id
--,eqp_onoff_searchid,eqp_onoff_wo_on_searchid,wo_on_searchid
from eqp_onoff
left join wo_on on eqp_onoff_wo_on_searchid = wo_on_searchid
where eqp_onoff_searchid not like '%false%' and wo_on_searchid is null
),
CTE_2 as(
	with eqp_onoff as(
			select concat(eqoftg_eqpm_id,eqoftg_worm_id,onoff_status) as
eqp_onoff_searchid
			,concat(eqoftg_eqpm_id,eqoftg_worm_id,onoff_status,'true') as
eqp_onoff_wo_off_searchid ,*
			from(
				select
				case when eqoftg_onoffstatus = true then 'true' else 'false' end as
onoff_status,*
				from equipmentonofftoggel
				) as subquery_off_1
		),
		wo_off as(
		select concat(wo_off_eqpm_id,wo_off_worm_id,'false',wo_off_activestatus) as
wo_off_searchid,*
		from (
			select
			case when wo_offactivestatus = true then 'true' else 'false' end as
wo_off_activestatus,*
			from workorder_offhistory
```

```
            ) as subquery_off_2
        )
    select eqoftg_eqpm_id as eqp_id_off ,eqoftg_worm_id as wo_offhistory_notinserted_worm_id
 --,eqp_onoff_searchid,eqp_onoff_wo_off_searchid,wo_off_searchid
    from eqp_onoff
    left join wo_off on eqp_onoff_wo_off_searchid = wo_off_searchid
    where eqp_onoff_searchid not like '%true%' and wo_off_searchid is null
    )

select
case when eqp_id_on is null then eqp_id_off else eqp_id_on end as eqp_id
,w.worm_createdtime ::date
,wo_onhistory_notinserted_worm_id,wo_offhistory_notinserted_worm_id
from CTE_1
full outer join CTE_2 on wo_onhistory_notinserted_worm_id=wo_offhistory_notinserted_worm_id
left outer join workordermaster w on case when wo_onhistory_notinserted_worm_id is not null then
 wo_onhistory_notinserted_worm_id=worm_id else wo_offhistory_notinserted_worm_id=worm_id
 end
order by 2 desc
```

Query to find resource user count:

```
select distinct (af.arfac_arm_id), a.arm_resourceid ,a.arm_first_name ,a.arm_last_name  from
 additionalresourcemaster a
join additionalresource_facility af  on a.arm_id = af.arfac_arm_id
join facilitymaster f on f.facm_id = af.arfac_facm_id
where f.facm_code like '1007%'and a.arm_isactive_status = true and a.arm_sms_only = false
```

Query to get Order No and Cost Center against Facility

```
select f.facm_code , cc_fac_costcentre, string_agg(cco_orderno,',') from facilitymaster f
left join costcentre_facilitywise_m cfm  on cfm.cc_fac_facility_id =f.facm_id
left join (select unnest(cco_cc_description) as cc,cco_orderno from costcenter_orderno co ) as
cc_orderno
on cc= cc_fac_costcentre
where f.facm_code like '1007-TTC%'
group by facm_code,cc_fac_costcentre;
```

Query to get Order No and Cost Center against Facility and Equipment:

```
select f.facm_code , cc_fac_costcentre,eqpgi_eqpm_code,e.eqpgi_costcenter ,
string_agg(cco_orderno,',') from facilitymaster f
left join equipmentgenerelinfo e on e.eqpgi_atlocation = f.facm_id
left join costcentre_facilitywise_m cfm  on cfm.cc_fac_facility_id =f.facm_id
left join (select unnest(cco_cc_description) as cc,cco_orderno from costcenter_orderno co ) as
cc_orderno on cc= cc_fac_costcentre
```

where *f*.facm_code **like '1007-TTC%'**
group by facm_code,cc_fac_costcentre,eqpgi_eqpm_code,eqpgi_costcenter;

Equipment table name:

Equipment Group -- equipmenttypes_m
Engineering Group -- equipmentcategory_m
Equipment Sub Group -- equipmentsubgroup_m
Equipment Class -- equipmentstatus_m
Equipment Criticality -- equipmentcriticality_m
Equipment Type -- assettype_m
Equipment Status -- equipmentclass_m

JSON querys:

update customserviceorder_text_feilds set cso_text_fields = cast(REPLACE(cast (cso_text_fields as varchar(10000)) , '\n' , '') as json) where cso_text_feilds_id = xx

update equipmentmaster **set** eqpm_description = **REPLACE(cast** (eqpm_description **as varchar**(10000)) , **'&'** , **'AND')**

New key column addition:

UPDATE customserviceorder_text_feilds set cso_text_fields = cso_text_fields::jsonb || '{"Target Date": ""}' Where cso_text_feilds_id =xx

Key column data search:

select * from customserviceorder_text_feilds where cso_text_fields ->> 'CBM Recommendations' like 'Velocit%'

Duplicate reservation ID:

```
with    CTE as (select distinct wospri_worm_id , wospri_sap_reservation_id ,
        count(wospri_sap_reservation_id) over (partition by wospri_sap_reservation_id) as reser_count,
        Count(wospri_sap_reservation_id) over (partition by wospri_worm_id,
 wospri_sap_reservation_id) as reser_wo_count,
case
        when   (count(wospri_sap_reservation_id) over (partition by wospri_sap_reservation_id)) =
 Count(wospri_sap_reservation_id) over (partition by wospri_worm_id, wospri_sap_reservation_id)
        then    'True' else 'False'
        end as  Case
from    workorder_spareparts_reservation_info wsri )
select  CTE.*,
        worm_woid
from    CTE
```

```
join    workordermaster on cte.wospri_worm_id = worm_id
where  CTE.wospri_sap_reservation_id not in ('0','')
and            CTE.case = 'False'
```

Workbench active without schedule:

```
select * from workbenchmaster w
full outer join scheduledmaintenancemaster s on wrbnm_som_smm_id = smm_smid
where s.smm_id is null and w.wrbnm_active_status ='t' limit 50
```

Equipment With Schedule:

```
select facm_name Facility_Name, facm_code Facility_Code, e1.eqpm_name Equipment_Name,
  e1.eqpm_description Equipment_Description, e1.eqpm_code Equipment_Code, s.smm_smid SMID,
  s.smm_smdescription SM_Description, eqpgi_eqppartof, e2.eqpm_name Parent_Equipment_Name,
  e2.eqpm_description Parent_Equipment_Description, e2.eqpm_code Parent_Equipment_Code from
  equipmentgenerelinfo e inner join facilitymaster on facm_id=eqpgi_atlocation full outer join
  equipmentmaster e2 on eqpgi_eqppartof = cast(e2.eqpm_id as varchar) full outer join
  scheduledmaintenancemaster s on e.eqpgi_eqpm_id = s.smm_eqpm_id inner join equipmentmaster e1
  on eqpgi_eqpm_id = e1.eqpm_id where e.eqpgi_atlocation in (xx) and s.smm_id is not null and
  s.smm_activestatus ='t';
```

Equipment Without Schedule:

select facm_name *Facility_Name*, facm_code *Facility_Code*, e1.eqpm_name *Equipment_Name*,
e1.eqpm_description *Equipment_Description*, e1.eqpm_code *Equipment_Code*, e2.eqpm_name
Parent_Equipment_Name, e2.eqpm_description *Parent_Equipment_Description*, e2.eqpm_code
Parent_Equipment_Code **from** equipmentgenerelinfo *e* **inner join** facilitymaster **on** facm_id =
eqpgi_atlocation **full outer join** equipmentmaster *e2* **on** eqpgi_eqppartof = **cast(***e2*.eqpm_id **as
varchar) full outer join** scheduledmaintenancemaster *s* **on** e.eqpgi_eqpm_id = *s*.smm_eqpm_id **inner
join** equipmentmaster *e1* **on** eqpgi_eqpm_id = *e1*.eqpm_id **where** *e*.eqpgi_atlocation **in** (xx) **and**
s.smm_id **is null and** *e2*.eqpm_equipmentclass !=**'Deleted'**;

Instrument with calibration:

```
select facm_name Facility_Name, facm_code Facility_Code, i2.insm_name Instrument_Name,
  i2.insm_description Instrument_Description, i2.insm_code Instrument_Code from
  instrumentgenerelinfo i inner join facilitymaster on facm_id=i.insgi_atlocation full outer join
  calibrationprocessmaster c on i.insgi_insm_id = c.calpm_insm_id inner join instrumentmaster i2 on
  i.insgi_insm_id = i2.insm_id where i.insgi_atlocation in (xx) and c.calpm_id is not null;
```

Instrument without calibration:

```
select facm_name Facility_Name, facm_code Facility_Code, i2.insm_name Instrument_Name,
  i2.insm_description Instrument_Description, i2.insm_code Instrument_Code from
  instrumentgenerelinfo i inner join facilitymaster on facm_id=i.insgi_atlocation full outer join
```

calibrationprocessmaster c on i.insgi_insm_id = c.calpm_insm_id inner join instrumentmaster i2 on i.insgi_insm_id = i2.insm_id where i.insgi_atlocation in (xx) and c.calpm_id is null;

Issue code with equipment and facility:

select facm_name, facm_code, e1.eqpicd_eqpm_code, e1.eqpicd_code, e1.eqpicd_description from facilitymaster f, equipmentissuecodes e1 ,equipmentgenerelinfo e where f.facm_id = e.eqpgi_atlocation and eqpgi_atlocation in (xx) and eqpgi_eqpm_id = e1.eqpicd_eqpm_id;

General Issue code with group and subgroup:

select em2.eqty_type as group, em3.eqsbgrp_subgroupname as subgourp,bdic_description ,bdic_code from breakdownissuecode bd
left join equipmenttypes_m em2 on bd.bdic_eqgrp_pk_id = em2.eqty_id
left join equipmentsubgroup_m em3 on em3.eqsbgrp_id = bd.bdic_eqsbgrp_pk_id;

General Issue Code Duplicate Code:

with duplicate_issueCodes as (
 select concat(em2.eqty_type, em3.eqsbgrp_subgroupname, bd.bdic_description) as concat_data,
 count(*) from breakdownissuecode bd
 left join equipmenttypes_m em2 on bd.bdic_eqgrp_pk_id = em2.eqty_id
 left join equipmentsubgroup_m em3 on em3.eqsbgrp_id = bd.bdic_eqsbgrp_pk_id
 group by concat(em2.eqty_type, em3.eqsbgrp_subgroupname, bd.bdic_description) having count(*) >
 1
)
select bd.bdic_id, eqty_type as group_data, em3.eqsbgrp_subgroupname as subgroup,
 bd.bdic_description as issue_description,
bdic_code, wofic_code ,dic.concat_data
from breakdownissuecode bd
left join equipmenttypes_m em2 on bd.bdic_eqgrp_pk_id = em2.eqty_id
left join equipmentsubgroup_m em3 on em3.eqsbgrp_id = bd.bdic_eqsbgrp_pk_id
left join wofailurecode wfc on bd.bdic_id = wfc.wofic_bdic_pk_id
right join duplicate_issueCodes dic on concat(em2.eqty_type, em3.eqsbgrp_subgroupname,
 bd.bdic_description) = dic.concat_data
order by 1,2,3,4 asc

Failure code with equipment and facility:

Select facm_name, facm_code, e1.eqpicd_eqpm_code, e1.eqpicd_code, e1.eqpicd_description,
 eqpficd_code, eqpficd_description from equipmentfailurecodes f1, facilitymaster f,
 equipmentissuecodes e1, equipmentgenerelinfo e where eqpgi_eqpm_id = f1.eqpficd_eqpm_id and
 eqpficd_eqpicd_code = e1.eqpicd_code and f.facm_id = e.eqpgi_atlocation and eqpgi_atlocation in
 (xx) and eqpgi_eqpm_id = e1.eqpicd_eqpm_id;

Query to find facility spare level and its approver for apollo:

```sql
SELECT
    wfcm_id, string_agg(um.usr_name::text,',') as app_name,
    string_agg(facm_code::text,',') as facm_code, wcd.wfcd_approve_by,
    wcd.wfcd_level, wfcn_wfcp_param
FROM
    mw_workflow_config_details wcd
JOIN mw_workflow_config_master wcm ON wcd.wfcd_wfcm_id = wcm.wfcm_id
JOIN mw_workflow_conditions wfc ON (SELECT unnest(wcd.wfcd_condition_desc)) = wfc.wfcn_id
JOIN users_m um ON um.usr_id = ANY(wcd.wfcd_approve_by)
join facilitymaster f on facm_id::text = Any(string_to_array(wfcm_facm_id,','))
WHERE wcm.wfcm_workflow_for = 'SP'
group by wfcm_id,wfcm_facm_id, wcd.wfcd_approve_by,
    wcd.wfcd_level, wfcn_wfcp_param;
```

Query to find facility spare level and its approver for AP:

```sql
SELECT
    wcm.wfcm_id,
    string_agg(um.usr_name::text, ',') AS app_name,
    string_agg(f.facm_code::text, ',') AS facm_code,
    wcd.wfcd_approve_by,
    wcd.wfcd_level,
    wfc.wfcn_wfcp_param
FROM
    mw_workflow_config_details wcd
JOIN
    mw_workflow_config_master wcm ON wcd.wfcd_wfcm_id = wcm.wfcm_id
JOIN
    mw_workflow_conditions wfc ON wfc.wfcn_id = ANY(wcd.wfcd_condition_desc::int[])
JOIN
    users_m um ON um.usr_id = ANY(wcd.wfcd_approve_by::int[])
JOIN
    facilitymaster f ON f.facm_id::text = ANY(string_to_array(wcm.wfcm_facm_id, ','))
WHERE
    wcm.wfcm_workflow_for = 'SP'
GROUP BY
    wcm.wfcm_id, wcd.wfcd_approve_by, wcd.wfcd_level, wfc.wfcn_wfcp_param;
```

Schedule without checklist:

```sql
select smm_id, smm_smid, smm_facm_id, smm_eqpm_id, smm_smdescription, smm_activestatus,
smm_schduletype, smm_nextscheduledate
from scheduledmaintenancemaster s
full outer join schedulemaintlabourtask l
on s.smm_id = l.smlt_smm_id
```

where l.smlt_tasks isnull
and smm_activestatus ='t';

Monthly report Query:

select worm_type,Count(worm_id) from workordermaster w where worm_createdtime::date >=
 '2023-06-01' and worm_createdtime::date <= '2023-06-30' and worm_woid like 'CHINP%' group by
 worm_type;

select count(worm_id) from workordermaster w where worm_type != 'Preventive' and
 worm_createdtime::date >= '2023-06-01' and worm_createdtime::date <= '2023-06-30' and
 worm_woid like 'CHINP%';

select count(som_id) from serviceordermaster s where som_smm_id like 'CHINP%' and
 som_createdtime::date >= '2023-06-01' and som_createdtime::date <= '2023-06-30';

select worm_category,Count(worm_id) from workordermaster w where worm_createdtime::date >=
 '2023-06-01' and worm_createdtime::date <= '2023-06-30' and worm_woid like 'CHINP%' group by
 worm_category;

select worm_status,worm_category,Count(worm_id) from workordermaster w where
 worm_createdtime::date >= '2023-06-01' and worm_createdtime::date <= '2023-06-30' and
 worm_woid like 'CHINP%' group by worm_status,worm_category;

Query to find columns used in other table:

SELECT *
FROM information_schema.columns
where column_name like '%facm_code%';

Duplicate reservation data correction:

select * from workorderparts w where worprt_reservation_id like '%yy';

select * from workorderparts w where worprt_reservation_id like '%yy' and worprt_complete = 'f';

select * from workorder_partsissue_return wpr where wopir_worprt_reservation_id like '%yy' and
 wopir_worm_id = xx ;

select * from workorder_spareparts_workflow wsw where wospwf_reservation_id like '%yy' and
 wospwf_worm_id = xx ;

select * from workorder_spareparts_workflow wsw where wospwf_worm_id = xx ;

select * from workorder_spareparts_reservation_info wsri where wospri_sap_reservation_id like '%yy'
 and wospri_worm_id = xx ;

select * from workorder_spareparts_reservation_info wsri where wospri_worm_id = xx ;

update workorderparts set worprt_reservation_id = '' where worprt_reservation_id like '%yy' and
 worprt_worm_id = xx ;

Delete from workorder_partsissue_return where wopir_worprt_reservation_id like '%yy' and
 wopir_worm_id = xx ;

update workorder_spareparts_workflow set wospwf_approved_by = null, wospwf_approved_status =
 'Waiting for Approval',
 wospwf_current_approved_status= 'Waiting for Approval', wospwf_level_approved_status='Waiting
 for Approval', wospwf_reservation_id='0',
wospwf_approved_datetime = null,wospwf_txn_statusupdate_message_type
 ='',wospwf_txn_statusupdate_message=''
where wospwf_reservation_id like '%yy' and wospwf_worm_id = xx ;

update workorder_spareparts_reservation_info set wospri_reservation_status
 ='',wospri_sap_reservation_message_type = '',wospri_sap_reservation_id = '0',
wospri_sap_reservation_message = '' where wospri_sap_reservation_id like '%yy' and wospri_worm_id
 = xx;

My Approval Query:

with shortlistlevel as (
            with chkrej1 as (
                    with chkrej as (
                            select distinct wospwf_worm_id as woid
                            from
workorder_spareparts_workflow,workordermaster,users_facility,users_m
                            where wospwf_isactive=true and wospwf_worm_id=worm_id and
usrfac_facm_id=worm_facm_id and usrfac_usr_id=usr_id and
                                usr_fac_access=any('{FU,CU}'::text[]) and usr_id=194 and 194 =
ANY(wospwf_approve_usrid)
) select distinct woid,string_agg(wospwf_approved_status,',') status from
chkrej,workorder_spareparts_workflow where wospwf_isactive=true and wospwf_worm_id=woid
group by woid
) select * from chkrej1 where status not like '%Rejected%'
) select count(*)
  from shortlistlevel,workorder_spareparts_workflow
        where wospwf_isactive=true and wospwf_approved_status='Waiting for Approval'
        and wospwf_worm_id=woid and 194 = ANY(wospwf_approve_usrid)
            and wospwf_approve_level = (
                    select wospwf_approve_level from workorder_spareparts_workflow

```
                    where wospwf_isactive=true and wospwf_worm_id = woid and
wospwf_approved_status='Waiting for Approval'
                                order by 1 asc limit 1
                    );
```

Dashboard Non-Conformance:

```
with cte_facmod as
      (
          select unnest(mwfwmcm_facm_ids) as facids, mwfwmcm_module_value
          from mw_facilitywise_module_config_m where mwfwmcm_module_opt_code =
'PM_CMP_ADD_CLS_ALL'
      ),
      pm_wo_period_cte as (
      select worm_facm_id pmfacid, worm_eqpm_id pmeqpid, worm_id pmwoid
      from workordermaster
        inner join facilitymaster on facm_id = worm_facm_id
          inner join cte_facmod on facids = worm_facm_id
        left join equipmentmaster on eqpm_id = worm_eqpm_id
      where lower(coalesce (worm_createfrom, '')) like 'sm%'
            AND upper (worm_status) not in ('CANCELLED','COMPLETED','CLOSED')
            and cast(worm_createdtime as date) >= to_date('102023', 'MMYYYY')
      and cast(worm_createdtime as date) < current_date
      and case when mwfwmcm_module_value in ('def','yes') then DATE_PART ('day', now () -
worm_expclosuredate) > 0 else 1=1
        end
      UNION ALL
      select worm_facm_id pmfacid, worm_eqpm_id pmeqpid, worm_id pmwoid
      from workordermaster
        inner join facilitymaster on facm_id = worm_facm_id
          inner join cte_facmod on facids = worm_facm_id
        left join equipmentmaster on eqpm_id = worm_eqpm_id
      where lower(coalesce (worm_createfrom, '')) like 'sm%'
            AND upper (worm_status) in ('COMPLETED','CLOSED')
            and cast(worm_createdtime as date) >= to_date('102023', 'MMYYYY')
      and cast(worm_createdtime as date) < current_date
       and case when mwfwmcm_module_value in ('yes') then DATE_PART ('day', now () -
worm_expclosuredate) > 0 else 1=1
          end
      ),
      pm_wo_cklst_non_confrm_cte as (
        select pmfacid, pmeqpid,
          cast(sum(all_nature) as numeric) all_nature,
          cast(sum(comp_all_nature) as numeric) comp_all_nature,
          cast(sum(mechanical) as numeric) mechanical,
          cast(sum(comp_mechanical) as numeric) comp_mechanical,
```

```
        cast(sum(electrical) as numeric) electrical,
        cast(sum(comp_electrical) as numeric) comp_electrical,
        cast(sum(instrumentation) as numeric) instrumentation,
        cast(sum(comp_instrumentation) as numeric) comp_instrumentation,
        cast(sum((all_nature - (mechanical + electrical + instrumentation))) as numeric) otherss,
        cast(sum((comp_all_nature - (comp_mechanical + comp_electrical + comp_instrumentation)))
as numeric) cmpothers,
        worlbt_natureofwork
      from
       (
        select pmfacid, pmeqpid, worm_id,
          sum(case lower(wlb.worlbt_natureofwork) when 'mechanical' then 1 else 0 end)
mechanical,
          sum(case when (lower(wlb.worlbt_natureofwork) = 'mechanical' and
wlb.worlbt_noncnf_resolve_status) then 1 else 0 end) comp_mechanical,
          sum(case lower(wlb.worlbt_natureofwork) when 'electrical' then 1 else 0 end) electrical,
          sum(case when (lower(wlb.worlbt_natureofwork) = 'electrical' and
wlb.worlbt_noncnf_resolve_status) then 1 else 0 end) comp_electrical,
          sum(case lower(wlb.worlbt_natureofwork) when 'instrumentation' then 1 else 0 end)
instrumentation,
          sum(case when (lower(wlb.worlbt_natureofwork) = 'instrumentation' and
wlb.worlbt_noncnf_resolve_status) then 1 else 0 end) comp_instrumentation,
          sum(1) all_nature,
          --sum(case when wlb.worlbt_complete then 1 else 0 end) comp_all_nature
          sum(case when wlb.worlbt_noncnf_resolve_status then 1 else 0 end) comp_all_nature ,
          worlbt_natureofwork
        from pm_wo_period_cte, workordermaster wm
         inner join workorderlabortasks wlb on wlb.worlbt_worm_id = worm_id
           and worlbt_checklist = true
        where pmwoid = worm_id and pmfacid = worm_facm_id
        group by 1,2,3,worlbt_natureofwork
       ) a
      group by 1,2,worlbt_natureofwork
     )select * from pm_wo_cklst_non_confrm_cte
```

Console query to fetch input, select and text area Only

```
// 1. Function to generate XPath for a given element
function getXPath(el) {
  if (el.id) {
    return '//*[@id="' + el.id + '"]';
  } else {
    const parts = [];
    while (el && el.nodeType === Node.ELEMENT_NODE) {
      let index = 1;
      let sibling = el.previousSibling;
      while (sibling) {
```

```javascript
            if (sibling.nodeType === Node.ELEMENT_NODE && sibling.nodeName ===
el.nodeName) {
                index++;
            }
            sibling = sibling.previousSibling;
        }
        const tagName = el.nodeName.toLowerCase();
        const pathIndex = index > 1 ? `[${index}]` : '';
        parts.unshift(`${tagName}${pathIndex}`);
        el = el.parentNode;
    }
    return '/' + parts.join('/');
  }
}

// 2. Select all form input elements (in document order)
const formFields = document.querySelectorAll("input[type='text'], select, textarea");

// 3. Map each field to its XPath
const fieldXPaths = Array.from(formFields).map((el, i) => {
    return `${i + 1}. ${getXPath(el)}`;
});

// 4. Print the numbered XPath list in console
console.log("Form Field XPaths:");
fieldXPaths.forEach(xpath => console.log(xpath));
```

To Find MW User If Relieved:

```sql
WITH db_list AS (
    SELECT unnest(ARRAY[
    'alstomtransport', 'amns','apollo', 'asianpaints','auromining',
    'coromandel', 'elkem', 'emerald', 'fenesta','flash',
    'gcpl','godrejastec', 'jkpaper', 'kone', 'laxmiorganic',
    'lifecell','lnttectonjv', 'lntwet','onrgy','opgpg',
    'parry', 'pidilite', 'ppgasianpaints','radico','raychemrpg',
    'royalenfield','rpl','savwlogisticsasset','sggroup','thermax',
    'uail','volkswagen', 'voss','vossauto','welspun',
    'yokohama','yokohama-india'
    ]) AS db_name
)
SELECT db_name AS "Customer_Name",
    -- SM-WO count
    (
    SELECT string_agg(usr_id::VARCHAR,',')
     FROM dblink('dbname=' || db_list.db_name || ' user=Un password=PwD',
```

'select u.usr_id from users_m u where u.usr_name ilike "%kalai%" and usr_isactive
="Active"')
       AS t(usr_id INTEGER)
) AS "USER_ID"
FROM db_list;

Query to check privilege:

SELECT grantee, privilege_type
FROM information_schema.role_table_grants
WHERE table_schema = 'ai'
 AND table_name = 'instrument_master_information';

Query to check HUNG language is blank or wrong:

SELECT * FROM workordermaster w
WHERE (coalesce(w.worm_status_language, '') = '' or
coalesce(w.worm_type_language, '') = '' or
coalesce(w.worm_category_language, '') = '' or
coalesce(w.worm_priority_language, '') = '' or
coalesce(w.worm_natureofissue_language, '') = '')
  AND w.worm_woid ILIKE '%HUNG%';

select * from scheduledmaintenancemaster s where (coalesce(s.smm_wocategory_language, '') = '' or
coalesce(s.smm_natureofissue_language, '') = '')
  AND s.smm_smid ILIKE '%HUNG%' and s.smm_activestatus ='t';

SELECT * FROM workbenchmaster w
WHERE w.wrbnm_active_status = 't'
  AND w.wrbnm_wbid ILIKE 'HUNG%'
  AND (
    COALESCE(w.wrbnm_status_language, '') = '' OR
    COALESCE(w.wrbnm_type_language, '') = '' OR
    COALESCE(w.wrbnm_natureofissue_language, '') = '' OR
    COALESCE(w.wrbnm_priority_language, '') = '' OR
    COALESCE(w.wrbnm_category_languge, '') = '');

select *w*.wrbnm_category, *w*.wrbnm_category_languge, worcat_category_en, worcat_category_hu
FROM workordercategory, workbenchmaster *w*
where *w*.wrbnm_category = worcat_category_en
and *w*.wrbnm_active_status = 't'
  AND *w*.wrbnm_wbid **ILIKE** 'HUNG%'
  group by *w*.wrbnm_category, *w*.wrbnm_category_languge, worcat_category_en,
worcat_category_hu;

select *w*.wrbnm_priority, *w*.wrbnm_priority_language, py_name_en, py_name_hu

FROM priority_m,workbenchmaster *w*
where *w*.wrbnm_priority = py_name_en
and *w*.wrbnm_active_status = 't'
  AND *w*.wrbnm_wbid **ILIKE** 'HUNG%'
  group by *w*.wrbnm_priority, *w*.wrbnm_priority_language, py_name_en, py_name_hu;

select wrbnm_natureofissue, wrbnm_natureofissue_language, bdnoi_en, bdnoi_hu
FROM breakdownnatureofissue_m,workbenchmaster *w*
where *w*.wrbnm_natureofissue = bdnoi_en
and *w*.wrbnm_active_status = 't'
  AND *w*.wrbnm_wbid **ILIKE** 'HUNG%'
  group by wrbnm_natureofissue, wrbnm_natureofissue_language, bdnoi_en, bdnoi_hu;

select wrbnm_type, wrbnm_type_language, bdtyp_type_en, bdtyp_type_hu
FROM breakdowntype_m,workbenchmaster *w*
where *w*.wrbnm_type = bdtyp_type_en
and *w*.wrbnm_active_status = 't'
  AND *w*.wrbnm_wbid **ILIKE** 'HUNG%'
  group by wrbnm_type, wrbnm_type_language, bdtyp_type_en, bdtyp_type_hu;

select wrbnm_status, wrbnm_status_language, bdsts_status_en, bdsts_status_hu
FROM breakdownstatus_m,workbenchmaster *w*
where *w*.wrbnm_status = bdsts_status_en
and *w*.wrbnm_active_status = 't'
  AND *w*.wrbnm_wbid **ILIKE** 'HUNG%'
  group by wrbnm_status, wrbnm_status_language, bdsts_status_en, bdsts_status_hu;