*This queries was created using SQL Developer and Oracle.*

*1. Count the number of active notifications in the Store with id = 193*

*The result should have next view: name of   Store, type of transaction and it amount. Order the result by amount.*

select st.STORE\_NAME, nt.Name, count(nt.Name)as AMOUNT FROM STORES st

join NOTIFICATIONS ns on st.ID\_STORE = ns.ID\_STORE

join NOTIFICATION\_TYPES nt on ns.ID\_NOTIFICATION\_TYPE = nt.ID\_NOTIFICATION\_TYPE

Where st.ID\_STORE = 193 and ns.ENABLED = 'Y'

GROUP BY st.STORE\_NAME, nt.Name

Order BY AMOUNT; *2. Count the number of active notifications with description 'Currency Fund Count....' (all types) in the Store with id = 198*

*The result should have next view: name of   Store, type of transaction and amount.*

select st.STORE\_NAME, nt.DESCRIPTION, COUNT(nt.DESCRIPTION) as AMOUNT FROM STORES st

join NOTIFICATIONS ns on st.ID\_STORE = ns.ID\_STORE

join NOTIFICATION\_TYPES nt on ns.ID\_NOTIFICATION\_TYPE = nt.ID\_NOTIFICATION\_TYPE

where st.ID\_STORE = 198 and nt.DESCRIPTION like 'Currency Fund Count%' and ns.ENABLED = 'Y'

group by st.STORE\_NAME, nt.DESCRIPTION; *3. Count the number of not active drawers in all active stores.*

*The result should have next view: names of   Stores, amount of drawers.*

*Order the result by store name.*

select st.STORE\_NAME, COUNT(dr.ID\_STORE)as AMOUNT FROM STORES st

join DRAWERS dr on st.ID\_STORE = dr.ID\_STORE

where st.ENABLED = 'Y' and dr.ENABLED = 'N'

group by st.STORE\_NAME

order by st.STORE\_NAME;

*4. Count the number of drawers which assigned to users (locate not in the safe).*

*The result should have next view: names of Stores, amount of drawers.*

*Order the result by store name*

select st.STORE\_NAME, COUNT(dr.ID\_STORE) as AMOUNT FROM STORES st

join DRAWERS dr on st.ID\_STORE = dr.ID\_STORE

where dr.ID\_DRAWER\_STATUS = 1

group by st.STORE\_NAME

order by st.STORE\_NAME;

*5. Perform with your own user next transaction: PM Count with reviewer*

*The result should have next view: names of   Stores, transaction status, transaction step, step status, creator of the step (login), step comment.*

*Order the result by creation date of the transaction steps.*

select st.STORE\_NAME, tt.NAME as NameTranz, ts.NAME as StatusTranz, stp.NAME as StepTranzac, sts.NAME as StatusStep, uss.LOGIN, stco.STEP\_COMMENT, tst.ADD\_DATE, tst.IS\_WITH\_REVIEWER

FROM STORES st

join TRANSACTIONS tr ON st.ID\_STORE = tr.ID\_STORE

join TRANSACTION\_TYPES tt on tt.ID\_TRANSACTION\_TYPE = tr.ID\_TRANSACTION\_TYPE

join TRANSACTION\_STATUSES ts on tr.ID\_TRANSACTION\_STATUS = ts.ID\_TRANSACTION\_STATUS

join TRANSACTION\_STEPS tst on tst.ID\_TRANSACTION = tr.ID\_TRANSACTION

join STEPS stp on tst.ID\_STEP = stp.ID\_STEP

--tt.ID\_TRANSACTION\_TYPE = stp.ID\_TRANSACTION\_TYPE

join STEP\_STATUSES sts on sts.ID\_STEP\_STATUS = tst.ID\_STEP\_STATUS

join STEP\_EXECUTORS stex on stex.ID\_TRANSACTION\_STEP = tst.ID\_TRANSACTION\_STEP

join USERS uss on uss.ID\_USER = stex.ID\_USER

left join STEP\_COMMENTS stco on stco.ID\_STEP\_EXECUTOR = stex.ID\_STEP\_EXECUTOR

where

tt.ID\_TRANSACTION\_TYPE = 3

--and st.ID\_STORE = 199 or stex.ID\_USER = 4049

ORDER BY tst.ADD\_DATE desc;