Q1.

SELECT LD.LOCATION\_NAME,

TD.TIME\_YEAR CONTRACT\_YEAR,

TD.TIME\_MONTH CONTRACT\_MONTH,

SUM(JF.QUANTITY\_ORDERED)\* SUM(JF.UNIT\_PRICE) JOB\_AMOUNT\_ORDERED,

ROUND(CUME\_DIST() OVER (PARTITION BY LD.LOCATION\_NAME, TD.TIME\_YEAR ORDER BY TD.TIME\_MONTH,SUM(JF.QUANTITY\_ORDERED)\* SUM(JF.UNIT\_PRICE)),2) AS CUMM\_JOB\_AMOUNT\_ORDERED

FROM W\_JOB\_F JF,

W\_LOCATION\_D LD,

W\_TIME\_D TD

WHERE JF.LOCATION\_ID = LD.LOCATION\_ID

AND JF.CONTRACT\_DATE = TD.TIME\_ID

GROUP BY LD.LOCATION\_NAME,

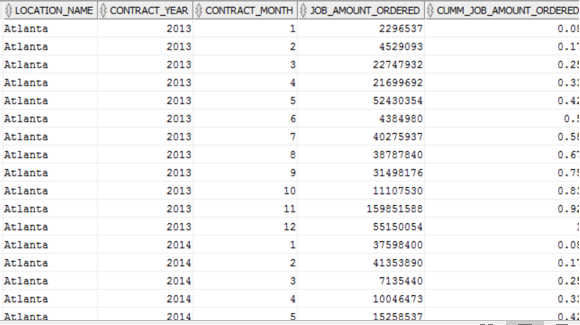
TD.TIME\_YEAR ,

TD.TIME\_MONTH

ORDER BY LOCATION\_NAME,

CONTRACT\_YEAR,

CONTRACT\_MONTH;



Q2.

SELECT LD.LOCATION\_NAME,

TD.TIME\_YEAR CONTRACT\_YEAR,

TD.TIME\_MONTH CONTRACT\_MONTH,

SUM(JF.QUANTITY\_ORDERED) \* SUM(JF.UNIT\_PRICE) JOB\_AMOUNT\_ORDERED,

ROUND(AVG(SUM(JF.QUANTITY\_ORDERED)\* SUM(JF.UNIT\_PRICE)) OVER (PARTITION BY LD.LOCATION\_NAME ORDER BY TD.TIME\_YEAR, TD.TIME\_MONTH ROWS 11 PRECEDING),2) AS MOVING\_AVG\_JOB\_AMOUNT\_ORDERED

FROM W\_JOB\_F JF,

W\_LOCATION\_D LD,

W\_TIME\_D TD

WHERE JF.LOCATION\_ID = LD.LOCATION\_ID

AND JF.CONTRACT\_DATE = TD.TIME\_ID

GROUP BY LD.LOCATION\_NAME,

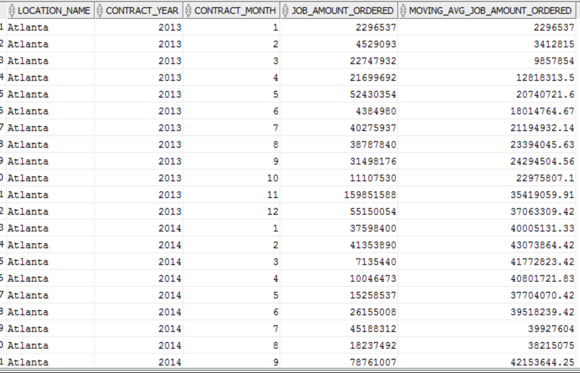
TD.TIME\_YEAR ,

TD.TIME\_MONTH

ORDER BY LOCATION\_NAME,

CONTRACT\_YEAR,

CONTRACT\_MONTH;



Q3.

SELECT INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH ,

SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL) AS PROFIT ,

RANK() OVER (PARTITION BY INS.CONTRACT\_YEAR ORDER BY (SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))DESC) ANNUAL\_RANK

FROM VW\_INVOICEREVENUESUMMARY INS,

VW\_SUBJOBCOSTSUMMARY CS

WHERE INS.JOB\_ID = CS.JOB\_ID

GROUP BY INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH

ORDER BY INS.CONTRACT\_YEAR,

ANNUAL\_RANK ;



Q4.

SELECT INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH ,

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END AS PROFIT\_MARGIN ,

RANK() OVER (PARTITION BY INS.CONTRACT\_YEAR ORDER BY

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END DESC) ANNUAL\_RANK

FROM VW\_INVOICEREVENUESUMMARY INS,

VW\_SUBJOBCOSTSUMMARY CS

WHERE INS.JOB\_ID = CS.JOB\_ID

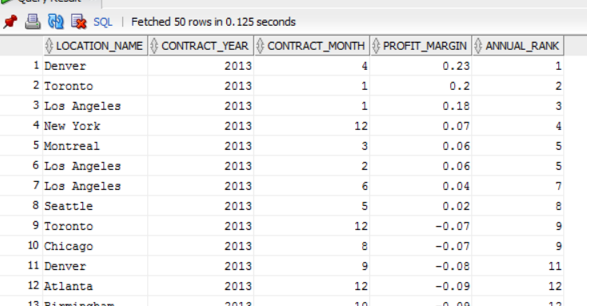
GROUP BY INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH

ORDER BY INS.CONTRACT\_YEAR,

ANNUAL\_RANK ;



Q5.

SELECT INS.JOB\_ID,

INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH ,

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END AS PROFIT\_MARGIN ,

PERCENT\_RANK() OVER (ORDER BY

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END) JOBS\_PERCENT\_RANK

FROM VW\_INVOICEREVENUESUMMARY INS,

VW\_SUBJOBCOSTSUMMARY CS

WHERE INS.JOB\_ID = CS.JOB\_ID

GROUP BY INS.JOB\_ID,

INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH

ORDER BY JOBS\_PERCENT\_RANK DESC;



SELECT JOB\_ID,

LOCATION\_NAME ,

CONTRACT\_YEAR ,

CONTRACT\_MONTH ,

PROFIT\_MARGIN ,

JOBS\_PERCENT\_RANK

FROM

(SELECT INS.JOB\_ID,

INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH ,

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END AS PROFIT\_MARGIN ,

PERCENT\_RANK() OVER (ORDER BY

CASE

WHEN SUM(INS.INVOICE\_AMOUNT)>0

THEN ROUND((SUM(INS.INVOICE\_AMOUNT) - SUM(CS.COST\_TOTAL))/SUM(INS.INVOICE\_AMOUNT),2)

ELSE 0

END) JOBS\_PERCENT\_RANK

FROM VW\_INVOICEREVENUESUMMARY INS,

VW\_SUBJOBCOSTSUMMARY CS

WHERE INS.JOB\_ID = CS.JOB\_ID

GROUP BY INS.JOB\_ID,

INS.LOCATION\_NAME ,

INS.CONTRACT\_YEAR ,

INS.CONTRACT\_MONTH

) Profit\_Percent\_Rank

WHERE JOBS\_PERCENT\_RANK >= 0.95;



Q7.

SELECT SCD.SALES\_CLASS\_DESC,

TD.TIME\_YEAR INVOICE\_SENT\_YEAR,

SUM(INF.QUANTITY\_SHIPPED) - SUM(INF.INVOICE\_QUANTITY) RETURN\_QUANTITY,

RANK() OVER(PARTITION BY TD.TIME\_YEAR ORDER BY (SUM(INF.QUANTITY\_SHIPPED) - SUM(INF.INVOICE\_QUANTITY)) DESC) ANNUAL\_RANK

FROM W\_JOB\_F JF,

W\_LOCATION\_D LD,

W\_INVOICELINE\_F INF,

W\_JOB\_SHIPMENT\_F JSF,

W\_SUB\_JOB\_F SJF,

W\_TIME\_D TD,

W\_SALES\_CLASS\_D SCD

WHERE JF.LOCATION\_ID = LD.LOCATION\_ID

AND JF.JOB\_ID = SJF.JOB\_ID

AND SJF.SUB\_JOB\_ID = JSF.SUB\_JOB\_ID

AND JSF.INVOICE\_ID = INF.INVOICE\_ID

AND INF.INVOICE\_SENT\_DATE = TD.TIME\_ID

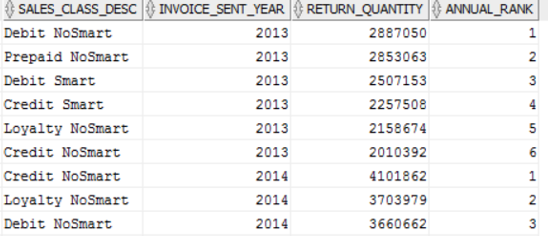
AND JF.SALES\_CLASS\_ID = SCD.SALES\_CLASS\_ID

GROUP BY SCD.SALES\_CLASS\_DESC,

TD.TIME\_YEAR

ORDER BY INVOICE\_SENT\_YEAR,

ANNUAL\_RANK;



Q8.

SELECT SCD.SALES\_CLASS\_DESC,

TD.TIME\_YEAR INVOICE\_SENT\_YEAR,

SUM(INF.QUANTITY\_SHIPPED) - SUM(INF.INVOICE\_QUANTITY) RETURN\_QUANTITY,

RATIO\_TO\_REPORT(SUM(INF.QUANTITY\_SHIPPED) - SUM(INF.INVOICE\_QUANTITY)) OVER(PARTITION BY TD.TIME\_YEAR) RETURNQTY\_RATIO

FROM W\_JOB\_F JF,

W\_LOCATION\_D LD,

W\_INVOICELINE\_F INF,

W\_JOB\_SHIPMENT\_F JSF,

W\_SUB\_JOB\_F SJF,

W\_TIME\_D TD,

W\_SALES\_CLASS\_D SCD

WHERE JF.LOCATION\_ID = LD.LOCATION\_ID

AND JF.JOB\_ID = SJF.JOB\_ID

AND SJF.SUB\_JOB\_ID = JSF.SUB\_JOB\_ID

AND JSF.INVOICE\_ID = INF.INVOICE\_ID

AND INF.INVOICE\_SENT\_DATE = TD.TIME\_ID

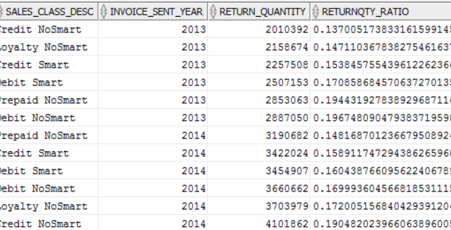
AND JF.SALES\_CLASS\_ID = SCD.SALES\_CLASS\_ID

GROUP BY SCD.SALES\_CLASS\_DESC,

TD.TIME\_YEAR

ORDER BY INVOICE\_SENT\_YEAR,

RETURNQTY\_RATIO;



Q9.

SELECT SDS.LOCATION\_NAME,

TD.TIME\_YEAR YEAR\_PROMISED,

SUM(SDS.BUSINESS\_DAYS\_DIFF) BUSINESS\_DAYS\_DIFF,

RANK() OVER(PARTITION BY TD.TIME\_YEAR ORDER BY SUM(SDS.BUSINESS\_DAYS\_DIFF) DESC) RANK\_DAYS\_DELAYED,

DENSE\_RANK() OVER(PARTITION BY TD.TIME\_YEAR ORDER BY SUM(SDS.BUSINESS\_DAYS\_DIFF) DESC) DENSERANK\_DAYS\_DELAYED

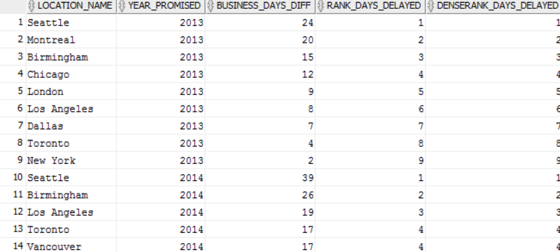
FROM VW\_SHIPPEDDATESTATS SDS,

W\_TIME\_D TD

WHERE SDS.DATE\_SHIP\_BY = TD.TIME\_ID

GROUP BY SDS.LOCATION\_NAME,

TD.TIME\_YEAR;



Q10.

SELECT SDS.LOCATION\_NAME,

TD.TIME\_YEAR YEAR\_PROMISED,

COUNT(DISTINCT JOB\_ID) JOBS\_DELAYED\_COUNT,

SUM(SDS.BUSINESS\_DAYS\_DIFF) BUSINESS\_DAYS\_DIFF,

CASE

WHEN SUM(QUANTITY\_ORDERED)>0

THEN (SUM(QUANTITY\_ORDERED)-SUM(SHIPPED\_QTY\_DELAYED))/SUM(QUANTITY\_ORDERED)

ELSE 0

END AS DELAY\_RATE,

RANK() OVER(PARTITION BY TD.TIME\_YEAR ORDER BY

CASE

WHEN SUM(QUANTITY\_ORDERED)>0

THEN (SUM(QUANTITY\_ORDERED)-SUM(SHIPPED\_QTY\_DELAYED))/SUM(QUANTITY\_ORDERED)

ELSE 0

END DESC) RANK\_DAYS\_DELAYED

FROM VW\_SHIPMENTDELAYS\_DATEPROMISED SDS,

W\_TIME\_D TD

WHERE SDS.DATE\_PROMISED = TD.TIME\_ID

GROUP BY SDS.LOCATION\_NAME,

TD.TIME\_YEAR;

