

Tracking Running Totals with Window Functions

Business problem: Tracking Sales Quota Progress over Time

The sales team works diligently to sell the product, and they have quotas that they must reach in order to earn all of their commission. Because these goals are so intimately tied to revenue, the manager of the team wants to track each sales member's performance throughout the year. You suggest a **% of quota reached** metric that could be displayed on a dashboard, but the sales manager expresses her concern that a single metric won't give her visibility into their progress throughout the year. You suggest providing a **running_total** of sales revenue and a **percent_quota** metric that will be recalculated every time a sales member makes another sale. She agrees, and you get started!

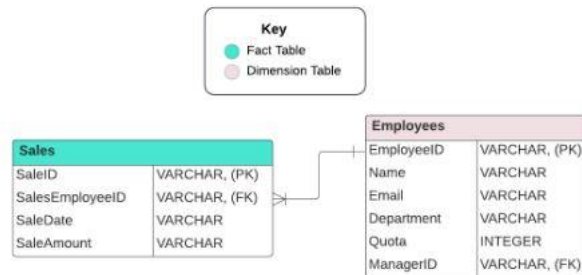
Task:

Calculate the running total of sales revenue, **running_total**, and the % of quota reached, **percent_quota**, for each sales employee on each date they make a sale. Use the **sales** and **employees** table to pull in and create the following fields:

- **salesemployeeid**
- **saledate**
- **saleamount**
- **quota**
- **running_total**
- **percent_quota**

Order the final output by **salesemployeeid** and **saledate**.

Sales Data Model



```
1  -- SQL code by John Uzoma
2  SELECT
3      S.salesemployeeid,
4      S.saledate,
5      S.saleamount,
6      SUM(S.saleamount) OVER(PARTITION BY S.salesemployeeid ORDER BY S.saledate) AS running_total,
7      CAST(SUM(S.saleamount) OVER(PARTITION BY S.salesemployeeid ORDER BY S.saledate) AS FLOAT) / E.quota as percent_quota
8  FROM SALES S
9  INNER JOIN EMPLOYEES E
10     ON S.salesemployeeid = E.employeeid
11  ORDER BY S.salesemployeeid, S.saledate;
```

Console output

Correct output

16 rows

SALESEMPLOYEEID	SALEDATE	SALEAMOUNT	RUNNING_TOTAL	PERCENT_QUOTA
E172	2023-03-11	9000	9000	0.18
E172	2023-06-11	11000	20000	0.4
E172	2023-06-15	15000	35000	0.7
E172	2023-07-17	3000	38000	0.76
E192	2023-02-09	14000	14000	0.23333333333333334
E192	2023-03-14	9000	23000	0.38333333333333336
E192	2023-05-16	16000	39000	0.65
E429	2023-01-30	4000	4000	0.05
E429	2023-04-03	7000	11000	0.1375
E429	2023-07-13	10000	21000	0.2625
E738	2023-01-04	18000	18000	0.36
E738	2023-03-18	10000	28000	0.56
E738	2023-03-26	5000	33000	0.66
E738	2023-04-16	1000	34000	0.68
E738	2023-07-16	12000	46000	0.92
E738	2023-07-28	11000	57000	1.14