# Compare the purchasing status of the average purchase

From the following tables write a query in SQL to compare the purchasing status of the average purchase quantity of products of a category to the average pruchase quantity of the distributor. Return purchase month, category\_id and purchase status.

## Table to copy

CREATE TABLE  product(

product\_id INTEGER(5) NOT NULL unique,

category\_id   INTEGER(4) NOT NULL);

insert into product values(8001,150);

insert into product values(8002,160);

insert into product values(8003,160);

insert into product values(8004,150);

insert into product values(8005,160);

CREATE TABLE purchase (

purchase\_no INTEGER(5) NOT NULL unique,

item\_code   INTEGER(4) NOT NULL,

purchase\_qty  integer(5),

purchase\_date  date,

foreign key (item\_code) references product(product\_id));

insert into purchase values(1001,8001,240,'2019-12-17');

insert into purchase values(1002,8002,150,'2019-12-17');

insert into purchase values(1003,8003,175,'2020-11-15');

insert into purchase values(1004,8004,150,'2019-12-17');

insert into purchase values(1005,8005,145,'2019-12-05');

insert into purchase values(1006,8001,150,'2020-01-05');

insert into purchase values(1007,8002,200,'2020-01-15');

insert into purchase values(1008,8003,150,'2020-12-17');

insert into purchase values(1009,8001,200,'2020-01-28');

insert into purchase values(1010,8002,180,'2020-02-07');

insert into purchase values(1011,8001,300,'2020-02-25');

insert into purchase values(1012,8005,100,'2020-01-27');

## Merging tables

SELECT

    \*

FROM

    purchase

        LEFT JOIN

    product ON product.product\_id = purchase.item\_code;

## Counting average of purchase\_qty and grouping by year-mont and category\_id

with merged as (SELECT

    \*

FROM

    purchase

        LEFT JOIN

    product ON product.product\_id = purchase.item\_code)

SELECT

    SUBSTR(purchase\_date, 1, 7) AS purchase\_month,

    category\_id,

    AVG(purchase\_qty) AS avg\_purchase\_cat

FROM

    merged

GROUP BY purchase\_month , category\_id

ORDER BY 2 , 1 ASC;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

## Counting average of purchase\_qty grouping by category\_id

with merged as (SELECT

    \*

FROM

    purchase

        LEFT JOIN

    product ON product.product\_id = purchase.item\_code),

    avg\_category as(SELECT

    SUBSTR(purchase\_date, 1, 7) AS purchase\_month,

    category\_id,

    AVG(purchase\_qty) AS avg\_purchase\_cat

FROM

    merged

GROUP BY purchase\_month , category\_id

ORDER BY 2 , 1 ASC)

SELECT

    SUBSTR(purchase\_date, 1, 7) AS purchase\_month,

    AVG(purchase\_qty) AS avg\_purchase\_month

FROM

    merged

GROUP BY 1;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

## Compering each average and write case statement

with merged as (SELECT

    \*

FROM

    purchase

        LEFT JOIN

    product ON product.product\_id = purchase.item\_code),

    avg\_category as(SELECT

    SUBSTR(purchase\_date, 1, 7) AS purchase\_month,

    category\_id,

    AVG(purchase\_qty) AS avg\_purchase\_cat

FROM

    merged

GROUP BY purchase\_month , category\_id

ORDER BY 2 , 1 ASC),

    avg\_month as(

 SELECT

    SUBSTR(purchase\_date, 1, 7) AS purchase\_month,

    AVG(purchase\_qty) AS avg\_purchase\_month

FROM

    merged

GROUP BY 1)

SELECT

    c.purchase\_month,

category\_id,

    CASE

        WHEN avg\_purchase\_cat > avg\_purchase\_month THEN 'increase'

        WHEN avg\_purchase\_cat < avg\_purchase\_month THEN 'decrease'

        WHEN avg\_purchase\_cat = avg\_purchase\_month THEN 'same'

    END AS purchase\_status

FROM

    avg\_category AS c

        LEFT JOIN

    avg\_month AS m ON c.purchase\_month = m.purchase\_month;

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie