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.

1. Table to copy

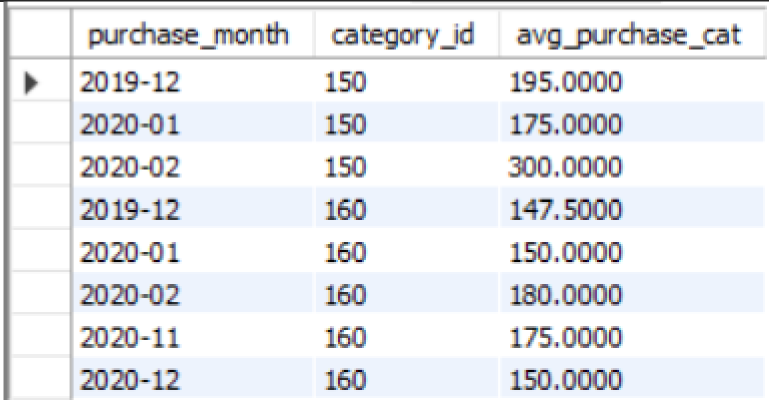
```
CREATE TABLE  product(
product_id INTEGER(5) NOT NULL unique,
category_id   INTEGER(4) NOT NULL);


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');
```

2. Merging tables

```
SELECT
    *
FROM
    purchase
        LEFT JOIN
    product ON product.product_id = purchase.item_code;
```

3. Counting average of purchase_qty and grouping by year-mont and category_id

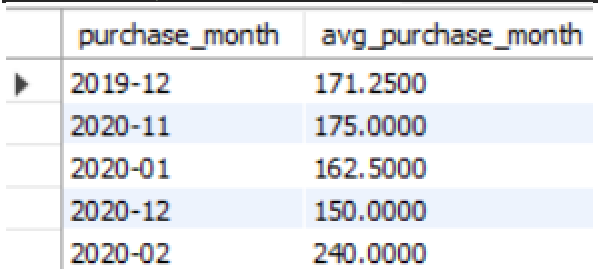
```
with merged as (SELECT
    *
FROM
    purchase
        LEFT JOIN
    product ON product.product_id = purchase.item_code)

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;
```



purchase_month	category_id	avg_purchase_cat
2019-12	150	195.0000
2020-01	150	175.0000
2020-02	150	300.0000
2019-12	160	147.5000
2020-01	160	150.0000
2020-02	160	180.0000
2020-11	160	175.0000
2020-12	160	150.0000

4. 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;
```

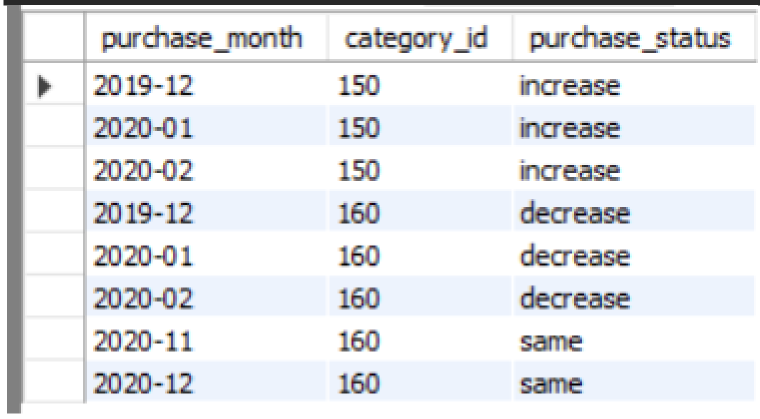


purchase_month	avg_purchase_month
2019-12	171.2500
2020-11	175.0000
2020-01	162.5000
2020-12	150.0000
2020-02	240.0000

5. 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),
```

```sql
    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;
```



purchase_month	category_id	purchase_status
2019-12	150	increase
2020-01	150	increase
2020-02	150	increase
2019-12	160	decrease
2020-01	160	decrease
2020-02	160	decrease
2020-11	160	same
2020-12	160	same