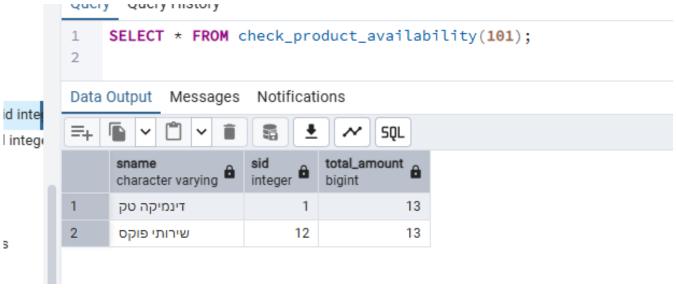
<u>דוח פרויקט שלב ד'</u>

פונקציות:

מקבל מוצר ומחזיר את הכמות לכל ספק $\underline{2}$

```
5 • CREATE OR REPLACE FUNCTION public.check_product_availability(
         RETURNS TABLE(sname character varying, sid integer, total_amount bigint)
8
         LANGUAGE 'plpgsql'
9
         COST 100
10
         VOLATILE PARALLEL UNSAFE
11
         ROWS 1000
12
13
    AS $BODY$
14 BEGIN
15
         RETURN QUERY
16
         SELECT s.sname, s.sid, SUM(pp.amount)
17
         FROM supplier s
18
         JOIN "order" o ON s.sid = o.sid
19
         JOIN ordprod op ON o.ordid = op.ordid
20
         JOIN product p ON op.pid = p.pid
21
         JOIN purprod pp ON pp.pid = p.pid
22
         WHERE p.pid = p_pid
         GROUP BY s.sname, s.sid;
23
24
   END;
25
    $BODY$;
```

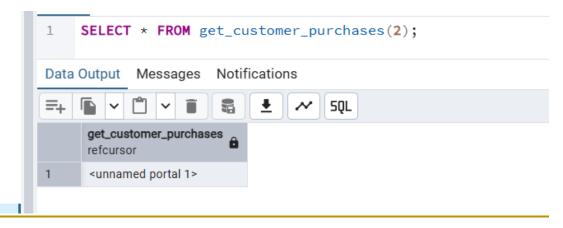
דוגמת הרצה:



מקבל לקוח ומחזיר את הרכישות שלו 2

```
CREATE OR REPLACE FUNCTION public.get_customer_purchases(
     p cid integer)
     RETURNS refcursor
     LANGUAGE 'plpgsql'
     COST 100
     VOLATILE PARALLEL UNSAFE
  AS $BODY$
  DECLARE
     cur refcursor;
 BEGIN
     OPEN cur FOR
          SELECT purid, purdate, purcost
          FROM purchase
          WHERE cid = p_cid;
     RETURN cur;
  END;
  $BODV$ .
```

דוגמת הרצה:



מכיון שחוזר אובייקט מסוג refcursor וזה לא טבלה.

פונקציה נוספת: מדפיסה מוצר וכמות במלאי של מוצר:

```
פונקציה שמקבלת מזהה מוצר ומדפיסה את שמו והכמות במלאי --
     2 • CREATE OR REPLACE FUNCTION print_product_info(p_pid INT)
        RETURNS VOID AS $$
     3
         DECLARE
             v_name VARCHAR;
     6
             v_quantity INT;
     7 V BEGIN
             שליפת שם המוצר והכמות למשתנים מקומיים (קורסור מרומז) ---
     9
             SELECT pname, quantity
    10
             INTO v_name, v_quantity
    11
             FROM product
    12
             WHERE pid = p_pid;
    13
    14
             -- הדפסת המידע על המוצר
    15
             RAISE NOTICE 'Product: %, Quantity: %', v_name, v_quantity;
        END;
    16
         $$ LANGUAGE plpgsql;
    17
    Data Output Messages Notifications
    CREATE FUNCTION
lity
    Query returned successfully in 66 msec.
es(
```

פרוצדורות: הסבר בתיעוד.

```
1 -- Procedure 1: Mark old 'ordered' purchases as 'cancelled'
2 • CREATE OR REPLACE PROCEDURE mark_cancelled_purchases()
3
    AS $$
4
     BEGIN
5
         UPDATE purchase
         SET purstatus = 'cancelled'
6
7
         WHERE purstatus = 'ordered'
8
          AND purdate < CURRENT_DATE - INTERVAL '30 days';</pre>
9
   END;
10
    $$ LANGUAGE plpgsql;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 59 msec.
```

```
-- Procedure 2: Assign eid = 1 to all purchases without employee
     -- Enhanced Procedure 2: Assign employee to purchases with NULL eid, with EXCEPTION handling
3 • CREATE OR REPLACE PROCEDURE assign_employee_to_purchases()
    AS $$
     BEGIN
        UPDATE purchase
6
7
        SET eid = 1
8
        WHERE eid IS NULL;
        RAISE NOTICE 'All purchases without employee were assigned to eid=1';
10
11 V EXCEPTION
         WHEN OTHERS THEN
12
13
             RAISE NOTICE 'An error occurred while assigning employee: %', SQLERRM;
14
     END;
   $$ LANGUAGE plpgsql;
Data Output Messages Notifications
CREATE PROCEDURE
Query returned successfully in 68 msec.
```

טריגרים: הסבר בתיעוד

```
1 -- Trigger 1: Check product stock before inserting to purprod
2 v CREATE OR REPLACE FUNCTION trg_check_stock()
   RETURNS trigger AS $$
    DECLARE
4
5
        available INT;
6 V BEGIN
7
        SELECT quantity INTO available
8
        FROM product
9
        WHERE pid = NEW.pid;
10
11 🗸
       IF available < NEW.amount THEN</pre>
           RAISE EXCEPTION 'Not enough stock for product %', NEW.pid;
12
        END IF;
13
14
15
        RETURN NEW;
16
   END;
17
    $$ LANGUAGE plpgsql;
19 v CREATE TRIGGER trg_before_purchase
20 BEFORE INSERT ON purprod
21 FOR EACH ROW
22 EXECUTE FUNCTION trg_check_stock();
```

```
-- Trigger 2: Reduce product quantity after purchase
 2 v CREATE OR REPLACE FUNCTION trg_decrease_stock()
     RETURNS trigger AS $$
     BEGIN
4
 5
         UPDATE product
 6
         SET quantity = quantity - NEW.amount
         WHERE pid = NEW.pid;
 7
 8
9
         RETURN NEW;
     END;
10
11
    $$ LANGUAGE plpgsql;
12
13 ∨ CREATE TRIGGER trg_after_purchase
    AFTER INSERT ON purprod
14
    FOR EACH ROW
15
16 EXECUTE FUNCTION trg_decrease_stock();
```

תוכניות ראשיות:

1. הקוד מריץ בלוק שבו הוא קורא תחילה לפרוצדורה שמסמנת רכישות שבוטלו, ואז מפעיל פונקציה שמחזירה את כל הרכישות של לקוח מסוים בעזרת קורסור, עובר עליהן בלולאה מדפיס כל רכישה דרך הודעת NOTICE:

```
-- Main Program 1: Call mark_cancelled_purchases() and get_customer_purchases()
      2 v DO $$
           DECLARE
               c refcursor;
              r RECORD;
      6 V BEGIN
               -- Call the procedure
      7
               CALL mark_cancelled_purchases();
      9
               -- Call the function and loop over results
     11
              c := get_customer_purchases(1);
     12 🗸
              LOOP
     13
                   FETCH c INTO r;
     14
                  EXIT WHEN NOT FOUND;
                   RAISE NOTICE 'Purchase ID: %, Date: %, Cost: %', r.purid, r.purdate, r.purcost;
     15
     16
               END LOOP;
     17
               CLOSE c;
          END;
     18
     19 $$;
     Data Output Messages Notifications
      NOTICE: Purchase ID: 1, Date: 2024-12-07, Cost: 436
      NOTICE: Purchase ID: 3001, Date: 2025-03-31, Cost: 100
bility
ses(
      Query returned successfully in 74 msec.
```

.2 הפונקציה מקצה לעובד ברירת מחדל את כל הרכישות שלא שובצו לעובד. לאחר מכן היא בודקת זמינות של מוצר מסוים אצל ספקים שונים. לבסוף, היא מציגה את שם הספק, מזהה הספק וכמות המוצר הזמינה.

```
-- Main Program 2: Call assign_employee_to_purchases() and check_product_availability()
        2 v DO $$
        3
             DECLARE
                 rec RECORD;
        5 V BEGIN
                 -- Call the procedure
                 CALL assign_employee_to_purchases();
                 -- Call the function and loop over results
        9
       10 🗸
                 FOR rec IN SELECT * FROM check_product_availability(3)
       11
       12
                      RAISE NOTICE 'Supplier: %, SID: %, Quantity: %', rec.sname, rec.sid, rec.total_amount;
       13
                 END LOOP;
       14
             END;
       15
             $$;
       Data Output Messages Notifications
        NOTICE: All purchases without employee were assigned to eid=1
        NOTICE: Supplier: דינמיקה טק, SID: 1, Quantity: 20
        NOTICE: Supplier: משאבים לוג, SID: 4, Quantity: 5
        NOTICE: Supplier: אריה טכנולוג, SID: 5, Quantity: 5
lability
        NOTICE: Supplier: אוריון שיווק, SID: 6, Quantity: 5
nases(
        NOTICE: Supplier: אספקה מרכז, SID: 9, Quantity: 5
        NOTICE: Supplier: קשרי תעשייה, SID: 10, Quantity: 10
        NOTICE: Supplier: נתיב פתרונות, SID: 11, Quantity: 5
        NOTICE: Supplier: שירותי פוקס, SID: 12, Quantity: 5
        NOTICE: Supplier: תפנית טכנולוג, SID: 13, Quantity: 5
```