

## דוח פרויקט שלב ד'

פונקציות:

2 מקבל מוצר ומחזיר את הכמות לכל ספק

```

4
5 CREATE OR REPLACE FUNCTION public.check_product_availability(
6     p_pid integer)
7     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
23     GROUP BY s.sname, s.sid;
24 END;
25 $BODY$;

```

דוגמת הרצה:

Query Query History

```
1 SELECT * FROM check_product_availability(101);
```

2

Data Output Messages Notifications

A toolbar containing several icons: a plus sign in a square, a document icon, a dropdown arrow, a clipboard icon, another dropdown arrow, a trash can icon, a database cylinder icon, a download arrow icon, a line graph icon, and a button labeled 'SQL'.

	sname character varying	sid integer	total_amount bigint
1	דינמיקה טק	1	13
2	שירותי פוקס	12	13

2 מקבל לקוח ומחזיר את הרכישות שלו

```

✓ 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;
$BODY$

```

דוגמת הרצה:

The screenshot shows a database IDE interface. At the top, a SQL query is entered: `1 SELECT * FROM get_customer_purchases(2);`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with two columns: 'get\_customer\_purchases' and 'refcursor'. The first row of data shows the value '<unnamed portal 1>' under the 'refcursor' column.

מכיון שחוזר אובייקט מסוג refcursor וזה לא טבלה.

פונקציה נוספת: מדפיסה מוצר וכמות במלאי של מוצר:

```

1  -- פונקציה שמקבלת מזהה מוצר ומדפיסה את שמו והכמות במלאי
2  CREATE OR REPLACE FUNCTION print_product_info(p_pid INT)
3  RETURNS VOID AS $$
4  DECLARE
5      v_name VARCHAR;
6      v_quantity INT;
7  BEGIN
8      -- שליפת שם המוצר והכמות למשתנים מקומיים (קורסור מרומז)
9      SELECT pname, quantity
10     INTO v_name, v_quantity
11     FROM product
12     WHERE pid = p_pid;
13
14     -- הדפסת המידע על המוצר
15     RAISE NOTICE 'Product: %, Quantity: %', v_name, v_quantity;
16 END;
17 $$ LANGUAGE plpgsql;
18
Data Output  Messages  Notifications
CREATE FUNCTION
lity Query returned successfully in 66 msec.
as(

```

## פרוצדורות: הסבר בתיקוד.

```
1 -- Procedure 1: Mark old 'ordered' purchases as 'cancelled'
2 CREATE OR REPLACE PROCEDURE mark_cancelled_purchases()
3 AS $$
4 BEGIN
5     UPDATE purchase
6     SET purstatus = 'cancelled'
7     WHERE purstatus = 'ordered'
8     AND purdate < CURRENT_DATE - INTERVAL '30 days';
9 END;
10 $$ LANGUAGE plpgsql;
```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 59 msec.

```
1 -- Procedure 2: Assign eid = 1 to all purchases without employee
2 -- Enhanced Procedure 2: Assign employee to purchases with NULL eid, with EXCEPTION handling
3 CREATE OR REPLACE PROCEDURE assign_employee_to_purchases()
4 AS $$
5 BEGIN
6     UPDATE purchase
7     SET eid = 1
8     WHERE eid IS NULL;
9
10    RAISE NOTICE 'All purchases without employee were assigned to eid=1';
11 EXCEPTION
12     WHEN OTHERS THEN
13         RAISE NOTICE 'An error occurred while assigning employee: %', SQLERRM;
14 END;
15 $$ 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  CREATE OR REPLACE FUNCTION trg_check_stock()
3  RETURNS trigger AS $$
4  DECLARE
5      available INT;
6  BEGIN
7      SELECT quantity INTO available
8      FROM product
9      WHERE pid = NEW.pid;
10
11     IF available < NEW.amount THEN
12         RAISE EXCEPTION 'Not enough stock for product %', NEW.pid;
13     END IF;
14
15     RETURN NEW;
16 END;
17 $$ LANGUAGE plpgsql;
18
19 CREATE TRIGGER trg_before_purchase
20 BEFORE INSERT ON purprod
21 FOR EACH ROW
22 EXECUTE FUNCTION trg_check_stock();
```

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

```
1 -- Main Program 1: Call mark_cancelled_purchases() and get_customer_purchases()
2 DO $$
3 DECLARE
4     c refcursor;
5     r RECORD;
6 BEGIN
7     -- Call the procedure
8     CALL mark_cancelled_purchases();
9
10    -- Call the function and loop over results
11    c := get_customer_purchases(1);
12    LOOP
13        FETCH c INTO r;
14        EXIT WHEN NOT FOUND;
15        RAISE NOTICE 'Purchase ID: %, Date: %, Cost: %', r.purid, r.purdate, r.purcost;
16    END LOOP;
17    CLOSE c;
18 END;
19 $$;
```

Data Output Messages Notifications

NOTICE: Purchase ID: 1, Date: 2024-12-07, Cost: 436  
NOTICE: Purchase ID: 3001, Date: 2025-03-31, Cost: 100  
DO

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.

```
1  -- Main Program 2: Call assign_employee_to_purchases() and check_product_availability()
2  DO $$
3  DECLARE
4      rec RECORD;
5  BEGIN
6      -- Call the procedure
7      CALL assign_employee_to_purchases();
8
9      -- Call the function and loop over results
10     FOR rec IN SELECT * FROM check_product_availability(3)
11     LOOP
12         RAISE NOTICE 'Supplier: %, SID: %, Quantity: %', rec.sname, rec.sid, rec.total_amount;
13     END LOOP;
14 END;
15 $$;
```

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  
NOTICE: Supplier: אוריון שיווק, SID: 6, Quantity: 5  
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