

דוח פרויקט שלב ד'

פונקציות(פונקציות נוספות בסוף הדף):

1. 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
	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 client 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, showing a table with two columns: "get_customer_purchases" and "refcursor". The first row of data is labeled "1" and contains the value "<unnamed portal 1>".

מכיון שחוזר אובייקט מסוג refcursor וזה לא טבלה.

3. 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 בתיעוד.

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

פונקציות נוספות:

```
Query Query History
1  -- FUNCTION: public.get_customer_summary(integer)
2
3  -- DROP FUNCTION IF EXISTS public.get_customer_summary(integer);
4
5  CREATE OR REPLACE FUNCTION public.get_customer_summary(
6      p_cid integer)
7      RETURNS text
8      LANGUAGE 'plpgsql'
9      COST 100
10     VOLATILE PARALLEL UNSAFE
11     AS $BODY$
12     DECLARE
13         total_orders INT;
14         total_items INT;
15         avg_cost NUMERIC;
16         total_cost INT;
17     BEGIN
18         -- מספר רכישות
19         SELECT COUNT(*) INTO total_orders
20         FROM purchase
21         WHERE cid = p_cid;
22
23         -- מספר מוצרים
24         SELECT COALESCE(SUM(pp.amount), 0) INTO total_items
25         FROM purchase pu
26         JOIN purprod pp ON pu.purid = pp.purid
27         WHERE pu.cid = p_cid;
28
29         -- ממוצע עלות ורכישות
30         SELECT COALESCE(AVG(purcost), 0), COALESCE(SUM(purcost), 0)
31         INTO avg_cost, total_cost
32         FROM purchase
33         WHERE cid = p_cid;
34
35         RETURN format(
36             '%s לקוח, מוצרים, ממוצע עלות רכישה %s רכישות, רכש %s ביצע %s לקוח',
37             p_cid,
38             total_orders,
39             total_items,
40             ROUND(avg_cost, 2),
41             CASE WHEN total_orders > 5 AND total_cost > 1000 THEN 'חשוב' ELSE 'רגיל' END
42         );
43     END;
44     $BODY$;
45
46 ALTER FUNCTION public.get_customer_summary(integer)
47     OWNER TO cotehila;
```

Data Output Messages Notifications

ALTER FUNCTION

הרצה:

Data Output Messages Notifications



get_customer_summary
text

1 2... רכישות, רכש 3 מוצרים, ממוצע עלות רכישה: 268.00, לקוח רגיל

Query

Query History

```

1 CREATE OR REPLACE FUNCTION reassign_if_employee_unsuitable(p_purid INT)
2 RETURNS TEXT AS $$
3 DECLARE
4     current_role TEXT;
5     current_eid INT;
6     new_eid INT;
7 BEGIN
8     -- שלב 1: שליפת העובד והתפקיד הנוכחי ברכישה
9     SELECT e.eid, r.rolename INTO current_eid, current_role
10    FROM purchase pu
11   JOIN employee e ON pu.eid = e.eid
12   JOIN role r ON e.rid = r.rid
13  WHERE pu.purid = p_purid;
14
15     -- שלב 2: בדיקה אם התפקיד לא מתאים
16     IF current_role NOT IN ('מסלולן', 'קופאי', 'Customer Service') THEN
17         -- שלב 3: חיפוש עובד אחר שכן מתאים
18         SELECT e2.eid INTO new_eid
19        FROM employee e2
20   JOIN role r2 ON e2.rid = r2.rid
21  WHERE r2.rolename IN ('מסלולן', 'קופאי', 'Customer Service')
22     LIMIT 1;
23
24     -- שלב 4: אם נמצא - עדכון הרכישה
25     IF FOUND THEN
26         UPDATE purchase
27        SET eid = new_eid
28       WHERE purid = p_purid;
29
30         RETURN format('העובד הלא מתאים הוחלף לעובד', new_eid);
31     ELSE
32         RETURN 'לא נמצא עובד מתאים להחלפה';
33     END IF;
34 ELSE
35     RETURN 'העובד הנוכחי כבר בתפקיד מתאים - אין צורך בהחלפה';
36 END IF;
37
38 EXCEPTION
39     WHEN NO_DATA_FOUND THEN
40         RETURN 'לא נמצאה רכישה או שאין לה עובד משויך';
41 END;
42 $$ LANGUAGE plpgsql;

```

Data Output

Messages

Notifications

CREATE FUNCTION

Query returned successfully in 143 msec.

הרצה:

reassign_if_employee_unsuitable

reassign_if_employee_unsuitable

```

1 SELECT reassign_if_employee_unsuitable(3010);

```

Data Output

Messages

Notifications

reassign_if_employee_unsuitable

text

1

העובד הלא מתאים הוחלף לעובד 23

bility(p_

(p_pid ir

ses(p_ci

pid integ

_unsuita

purchas