

דו"ח הפרויקט- שלב שני

שאלות select

1) מוצא את שמות כל המוצרים שנמכרו ביותר מהכמות הממוצעת של כל ההזמנות

Query
Query History

```

1  -- שמות כל המוצרים שנמכרו ביותר מהכמות הממוצעת של כל ההזמנות
2
3  SELECT p.pName, SUM(op.amount) AS total_sold
4  FROM product p
5  JOIN orderProd op ON p.pId = op.pId
6  GROUP BY p.pName
7  HAVING SUM(op.amount) > (SELECT AVG(amount) FROM orderProd);

```

Data Output
Messages
Notifications

Showing rows: 1 to 129
Page No: 1

	pname character varying (30)	total_sold bigint
1	Dumbbells	28
2	Broccoli	25
3	Toothpaste	29
4	Dress	38
5	TV	43
6	Toy Car	8

2) מחזיר את שמות כל הלקוחות שביצעו הזמנה בחודש האחרון

Query

Query History

1

--לקוחות שעשו הזמנה בחודש האחרון

2

3

SELECT c.cid, c.cName, c.cphone, o.ordDate

4

FROM customer c

5

JOIN orders o ON c.cId = o.cId

6

WHERE EXTRACT(MONTH FROM o.ordDate) = EXTRACT(MONTH FROM CURRENT_DATE)

7

AND EXTRACT(YEAR FROM o.ordDate) = EXTRACT(YEAR FROM CURRENT_DATE);

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 20

✎️

Page No: 1

	cid numeric (5)	cname character varying (15)	cphone numeric (12)	orddate date
1	20	Customer20	370062237812	2025-03-29
2	21	Customer21	764136434115	2025-03-02
3	24	Customer24	333257007042	2025-03-27
4	38	Customer38	539227273660	2025-03-27
5	45	Customer45	171884161354	2025-03-29
6	83	Customer83	578317484129	2025-03-14
7	00	Customer00	502012266275	2025-03-20

3) מראה את המחיר הכולל ששולם עבור כל מוצר שהוזמן ביותר מ-10 פעמים

מראה את המחיר הכולל ששולם עבור כל מוצר שהוזמן ביותר מ-10 פעמים--

SELECT p.pName, SUM(p.price * op.amount) AS total_revenue

FROM product p

JOIN orderProd op ON p.pId = op.pId

GROUP BY p.pName

HAVING SUM(op.amount) > 10;

Data OutputMessagesNotifications

Showing rows: 1 to 126

Page No: 1

	pname character varying (30)	total_revenue double precision
1	Dumbbells	3371.4799999999996
2	Broccoli	3920.75000000000005
3	Toothpaste	2398.88
4	Dress	7528.94
5	TV	1278.82
6	Butter	6377.76
7	Shrimp	2741.25

4) מראה כמה מוצרים יש בכל מחלקה, ומסדר אותם לפי כמות יורדת

Query Query History

```
1  --מראה כמה מוצרים יש בכל מחלקה, ומסדר אותם לפי כמות יורדת-
2
3  SELECT d.depName, COUNT(p.pId) AS total_products
4  FROM department d
5  JOIN product p ON d.depId = p.depId
6  GROUP BY d.depName
7  ORDER BY total_products DESC;
```

Data Output Messages Notifications

Showing rows: 1 to 20 Page No: 1

	depname character varying (30)	total_products bigint
1	Clothing	12
2	Electronics	12
3	Stationery	9
4	Music	9
5	Pet Supplies	7
6	Fish	7
7	Jewelrv	7

5) מציג את שם הלקוח שביצע את ההזמנה היקרה ביותר

Query Query History

```
1  --מציג את שם הלקוח שביצע את ההזמנה היקרה ביותר--
2
3  SELECT c.cid, c.cName, c.cphone, o.ordCost
4  FROM customer c
5  JOIN orders o ON c.cId = o.cId
6  WHERE o.ordCost = (SELECT MAX(ordCost) FROM orders);
7
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

	cid numeric (5)	cname character varying (15)	cphone numeric (12)	ordcost integer
1	84	Customer84	865376181014	499

(6) מוצא את שמות העובדים שעובדים במחלקה עם הכי הרבה מוצרים

Query Query History

```
3 SELECT e.eName, d.depName
4 FROM employee e
5 JOIN department d ON e.depId = d.depId
6 WHERE d.depId = (
7     SELECT p.depId
8     FROM product p
9     GROUP BY p.depId
10    ORDER BY COUNT(*) DESC
11    LIMIT 1
12 )
```

Data Output Messages Notifications

Showing rows: 1 to 11 Page No: 1

	ename character varying (15)	depname character varying (30)
1	Employee11	Electronics
2	Employee12	Electronics
3	Employee32	Electronics
4	Employee49	Electronics
5	Employee52	Electronics

(7) מחזיר את מספר ההזמנות שבוצעו בכל חודש בשנה הנוכחית

Query Query History

```
1 --מחזיר את מספר ההזמנות שבוצעו בכל חודש בשנה הנוכחית
2
3 SELECT EXTRACT(MONTH FROM ordDate) AS order_month, COUNT(*) AS total_orders
4 FROM orders
5 WHERE EXTRACT(YEAR FROM ordDate) = EXTRACT(YEAR FROM CURRENT_DATE)
6 GROUP BY order_month
7 ORDER BY order_month;
8
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1

	order_month numeric	total_orders bigint
1	1	29
2	2	25
3	3	20

8) מוצא את המחלקות שאין בהן מוצרים בתוקף

הטבלה ריקה, כיון שיש אילוץ שאסור להכניס למערכת מוצר עם תאריך שכבר היה בעמודת התוקף, עם זאת, אחרי שיעבור הזמן ובתוקף יעבור השאילתא תהיה יעילה ונצרכת.

The screenshot shows the SQL Developer interface. The query editor contains the following SQL code:

```
1  --מוצא את המחלקות שאין בהן מוצרים בתוקף (validity)
2
3  SELECT d.depName
4  FROM department d
5  WHERE NOT EXISTS (
6      SELECT 1
7      FROM product p
8      WHERE p.depId = d.depId
9      AND p.validity >= CURRENT_DATE
10 );
```

Below the query editor, the 'Data Output' tab is active, showing the data type for the 'depname' column: 'depname character varying (30)'.

9) מציאת המחלקות עם הכי הרבה מכירות

The screenshot shows the SQL Developer interface. The query editor contains the following SQL code:

```
1  --מציאת המחלקות עם הכי הרבה מכירות
2
3  SELECT d.depId, d.depName, SUM(op.amount) AS total_sold
4  FROM department d
5  JOIN product p ON d.depId = p.depId
6  JOIN orderProd op ON p.pId = op.pId
7  GROUP BY d.depId, d.depName
8  ORDER BY total_sold DESC
9  LIMIT 1;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The columns are 'depId', 'depname', and 'total_sold'. The first row shows '1', 'Electronics', and '498'.

depId	depname	total_sold
1	Electronics	498

שאלות update

(1) העלאת מחיר כל המוצרים ב-10% אם הם נמכרו מעל 50 פעמים

```
1  --העלאת מחיר כל המוצרים ב-10% אם הם נמכרו מעל 50 פעמים
2
3  UPDATE product
4  SET price = price * 1.1
5  WHERE pId IN (
6      SELECT pId FROM orderProd GROUP BY pId HAVING SUM(amount) > 50
7  );
8
```

Data Output Messages Notifications

UPDATE 27

Query returned successfully in 98 msec.

לפני:

vs

Query Query History

1 SELECT * FROM public.product
2 ORDER BY pid ASC

Data Output Messages Notifications

Showing rows: 1 to 130 Page No: 1 of 1

	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)
1	1	Laptop	13	72.49	2025-11-29	20
2	2	Shirt	27	149.17	2025-12-15	15
3	3	Apple	2	70.42	2026-02-13	14
4	4	Sofa	27	18	2025-06-15	14
5	5	Toy Car	32	107.5	2026-03-13	4
6	6	Book	37	113.37	2025-11-22	14
7	7	TV	48	29.74	2025-10-17	11
8	8	Jeans	1	14.76	2026-01-04	4
9	9	Banana	25	86.20	2025-10-04	1

3

Query

Query History

Scratch

1

SELECT *

FROM public.product

2

ORDER BY pid ASC

Data Output

Messages

Notifications

Showing rows: 1 to 130

Page No: 1

of

	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)
1	1	Laptop	13	72.49	2025-11-29	20
2	2	Shirt	27	164.087	2025-12-15	15
3	3	Apple	2	70.42	2026-02-13	14
4	4	Sofa	27	18	2025-06-15	14
5	5	Toy Car	32	107.5	2026-03-13	4
6	6	Book	37	113.37	2025-11-22	14
7	7	TV	48	29.74	2025-10-17	11
8	8	Jeans	1	14.76	2026-01-04	4
9	9	Banana	25	86.29	2025-10-04	1
10	10	Couch	37	68.02	2026-02-13	19

(2) עדכון סטטוס של עובדים במחלקה מסוימת ל "inactive" אם לא בוצעו הזמנות דרך מחלקתם
בחצי השנה האחרונה

Query Query History

```

1  --מחלקתם בחצי השנה האחרונה "inactive"-עדכון סטטוס של עובדים במחלקה מסוימת ל-
2
3  UPDATE employee
4  SET rId = (SELECT rId FROM role WHERE alias = 'inactive' LIMIT 1)
5  WHERE depId NOT IN (
6    SELECT DISTINCT p.depid
7    FROM product p
8    JOIN orderProd op ON p.pId = op.pId
9    JOIN orders o ON op.ordId = o.ordId

```

Data Output Messages Notifications

UPDATE 0

Query returned successfully in 57 msec.

לפני:

Query		Query History	Scratch F		
1	SELECT *	FROM public.employee			
2	ORDER BY	eid ASC			
Data Output		Messages	Notifications		
				Showing rows: 1 to 100	Page No: 1 of 1
	eid [PK] numeric (5)	ename character varying (15)	ephone numeric (12)	depid numeric (5)	rid numeric (5)
1	1	Employee1	233643715257	3	6
2	2	Employee2	566215199523	12	5
3	3	Employee3	497194325901	18	6
4	4	Employee4	324594896756	16	3
5	5	Employee5	835452799492	18	4
6	6	Employee6	957818532404	19	5
7	7	Employee7	649414632007	19	9
8	8	Employee8	327820124447	9	1
9	9	Employee9	132471733668	18	2

אחרי:

Query		Query History	Scratch F		
1	SELECT *	FROM public.employee			
2	ORDER BY	eid ASC			
Data Output		Messages	Notifications		
				Showing rows: 1 to 100	Page No: 1 of 1
	eid [PK] numeric (5)	ename character varying (15)	ephone numeric (12)	depid numeric (5)	rid numeric (5)
1	1	Employee1	233643715257	3	6
2	2	Employee2	566215199523	12	5
3	3	Employee3	497194325901	18	6
4	4	Employee4	324594896756	16	3
5	5	Employee5	835452799492	18	4
6	6	Employee6	957818532404	19	5
7	7	Employee7	649414632007	19	9
8	8	Employee8	327820124447	9	1
9	9	Employee9	132471733668	18	2
10	10	Employee10	272528171115	9	2

3) עדכון מחיר כל המוצרים שתוקפם פג ל-0 (נחשבים כלא זמינים)

1
2
3
4
5
6

--(נחשבים כלא זמינים) עדכון מחיר כל המוצרים שתוקפם פג ל-0

UPDATE product
SET price = 0
WHERE validity < CURRENT_DATE;

Data Output

Messages

Notifications

UPDATE 0

Query returned successfully in 49 msec.

לפני:

Query

Query History

1
2

SELECT * FROM public.product
ORDER BY pid ASC

Data Output

Messages

Notifications

Showing rows: 1 to 130

Page No: 1

	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)
1	1	Laptop	13	72.49	2025-11-29	20
2	2	Shirt	27	149.17	2025-12-15	15
3	3	Apple	2	70.42	2026-02-13	14
4	4	Sofa	27	18	2025-06-15	14
5	5	Toy Car	32	107.5	2026-03-13	4
6	6	Book	37	113.37	2025-11-22	14

Query

Query History

1

SELECT * FROM public.product

2

ORDER BY pid ASC

Data Output

Messages

Notifications

Showing rows: 1 to 130

Page No: 1

of

	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)
1	1	Laptop	13	72.49	2025-11-29	20
2	2	Shirt	27	164.087	2025-12-15	15
3	3	Apple	2	70.42	2026-02-13	14
4	4	Sofa	27	18	2025-06-15	14
5	5	Toy Car	32	107.5	2026-03-13	4
6	6	Book	37	113.37	2025-11-22	14
7	7	TV	48	29.74	2025-10-17	11
8	8	Jeans	1	14.76	2026-01-04	4
9	9	Banana	25	86.29	2025-10-04	1
10	10	Couch	37	68.02	2026-02-13	19

שאלות delete

(1) מחיקת כל ההזמנות שבוצעו יותר משנה אחורה

Query		Query History		 	
1	מחיקת כל ההזמנות שבוצעו יותר משנה אחורה				
2					
3	▼	DELETE FROM	orders		
4		WHERE	ordDate < CURRENT_DATE - INTERVAL '1 year';		
5					
Data Output		Messages		Notifications	
DELETE 314					
Query returned successfully in 80 msec.					

לפני:

Query

Query History

Scratch Pa

1

SELECT * FROM public.orders

2

ORDER BY ordid ASC

Data Output

Messages

Notifications

Showing rows: 1 to 613

Page No: 1 of 1

	ordid [PK] numeric (5)	orddate date	ordstatus character varying (10)	ordcost integer	cid numeric (5)
1	1	2024-12-07	cancelled	436	1
2	2	2024-10-25	cancelled	348	2
3	3	2025-02-05	delivered	281	2
4	4	2023-06-02	ordered	449	3
5	5	2024-09-17	delivered	73	3
6	6	2024-07-20	cancelled	246	3
7	7	2024-10-27	ordered	435	4
8	8	2024-12-14	ordered	223	5
9	9	2024-03-22	delivered	326	5

אחרי:

(2) מחיקת לקוחות שלא ביצעו הזמנה מעולם

QueryQuery History

1

--הזמנה מעולם-

2

3

▼DELETE FROM customer

4WHERE cId NOT IN (SELECT DISTINCT cId FROM orders);

5

Data OutputMessagesNotifications

DELETE 186

Query returned successfully in 47 msec.

לפני:

QueryQuery History

1

▼SELECT * FROM public.customer

2ORDER BY cid ASC

Data OutputMessagesNotifications

Showing rows: 1 to 400Page No: 1 of 1

	cid [PK] numeric (5)	cname character varying (15)	cphone numeric (12)
1	1	Customer1	806715987455
2	2	Customer2	860121600253
3	3	Customer3	188597830061
4	4	Customer4	134717068926
5	5	Customer5	387696126771
6	6	Customer6	624480379433
7	7	Customer7	710407155966
8	8	Customer8	842696148500
9	9	Customer9	390734286734

Query Query History

```

1 SELECT * FROM public.customer
2 ORDER BY cid ASC

```

Data Output Messages Notifications

Showing rows: 1 to 214 Page No: 1

	cid [PK] numeric (5)	cname character varying (15)	cphone numeric (12)
1	1	Customer1	806715987455
2	2	Customer2	860121600253
3	3	Customer3	188597830061
4	4	Customer4	134717068926
5	5	Customer5	387696126771
6	7	Customer7	710407155966
7	8	Customer8	842696148500
8	9	Customer9	390734286734
9	10	Customer10	855340212494
10	11	Customer11	130278793711

(3) מחיקת מוצרים שאין מהם מלאי בכלל ואין הזמנות שלהם

Query Query History

```

1 --מחיקת מוצרים שאין מהם מלאי בכלל ואין הזמנות שלהם
2
3 DELETE FROM product
4 WHERE stock = 0
5 AND pId NOT IN (SELECT DISTINCT pId FROM orderProd);
6

```

Data Output Messages Notifications

DELETE 0

Query returned successfully in 42 msec.

לפני:

Query Query History		Scratch Pad :					
1 SELECT * FROM public.product							
2 ORDER BY pid ASC							
Data Output Messages Notifications							
		Showing rows: 1 to 130 Page No: 1 of 1					
	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)	
1	1	Laptop	13	72.49	2025-11-29	20	
2	2	Shirt	27	149.17	2025-12-15	15	
3	3	Apple	2	70.42	2026-02-13	14	
4	4	Sofa	27	18	2025-06-15	14	
5	5	Toy Car	32	107.5	2026-03-13	4	
6	6	Book	37	113.37	2025-11-22	14	
7	7	TV	48	29.74	2025-10-17	11	
8	8	Jeans	1	14.76	2026-01-04	4	
9	9	Banana	25	86.29	2025-10-04	1	

אחרי:

Query Query History		Scratch Pad					
1 SELECT * FROM public.product							
2 ORDER BY pid ASC							
Data Output Messages Notifications							
		Showing rows: 1 to 130 Page No: 1 of 1					
	pid [PK] numeric (5)	pname character varying (30)	stock integer	price double precision	validity date	depid numeric (5)	
1	1	Laptop	13	72.49	2025-11-29	20	
2	2	Shirt	27	164.087	2025-12-15	15	
3	3	Apple	2	70.42	2026-02-13	14	
4	4	Sofa	27	18	2025-06-15	14	
5	5	Toy Car	32	107.5	2026-03-13	4	
6	6	Book	37	113.37	2025-11-22	14	
7	7	TV	48	29.74	2025-10-17	11	
8	8	Jeans	1	14.76	2026-01-04	4	
9	9	Banana	25	86.29	2025-10-04	1	
10	10	Couch	37	68.02	2026-02-13	19	

קביעת אילוצים

טלפון חייב להיות באורך 12 ספרות.

אם לא מוזן תאריך הזמנה, ייקבע התאריך הנוכחי כברירת מחדל.

NOT NULL: שם המוצר הוא חובה.

Query Query History

```
1 ALTER TABLE employee
2
3 ADD CONSTRAINT chk_phone CHECK (ePhone >= 100000000000 AND ePhone <= 99999999999); -- 1.
4
5
6 ALTER TABLE orders
7 ALTER COLUMN ordDate SET DEFAULT CURRENT_DATE; -- 2. DEFAULT: ייקבע התאריך הנוכחי
8
9 ALTER TABLE product
10 ALTER COLUMN pName SET NOT NULL; -- 3. NOT NULL: שם המוצר הוא חובה
11
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 95 msec.

בדיקה 1: אין מספיק ספרות בטלפון

Query Query History

```
1 INSERT INTO employee (eId, eName, ePhone, depId, rId)
2 VALUES (2001, 'Alice', 1234567890, 1, 1);
3
```

Data Output Messages Notifications

ERROR: new row for relation "employee" violates check constraint "chk_phone"
Failing row contains (2001, Alice, 1234567890, 1, 1).

SQL state: 23514
Detail: Failing row contains (2001, Alice, 1234567890, 1, 1).

בדיקה 2: תאריך מתעדכן כברירת מחדל

Query Query History

```
1 INSERT INTO orders (ordId, ordStatus, ordCost, cId)
2 VALUES (3001, 'ordered', 100, 1);
3
4 SELECT ordId, ordDate FROM orders WHERE ordId = 3001;
5
```

Data Output Messages Notifications

	ordid [PK] numeric (5)	orddate date
1	3001	2025-03-31

בדיקה 3: ערך NULL

Te
igr
ctic
erie
rate
sed
jer
es
ust
epa
mp
rde
rde

Query Query History

```
1  ✓ INSERT INTO product (pId, stock, price, validity, depId)
2  VALUES (5001, 10, 49.99, '2026-01-01', 2);
3
```

Data Output **Messages** Notifications

```
ERROR: null value in column "pname" of relation "product" violates not-null constraint
Failing row contains (5001, null, 10, 49.99, 2026-01-01, 2).
```

SQL state: 23502

Detail: Failing row contains (5001, null, 10, 49.99, 2026-01-01, 2).