מיני פרויקט בבסיסי נתונים

שם הארגון: חנות

שם האגף: נשים

מגישות:

תהילה בן עזרא 323845321
tasaraf@g.jct.ac.il
מיכל ישששכר 213686496
Michalisr17@gmail.com

מיני פרויקט בבסיסי נתונים

תוכן עניינים

מיני פרויקט בבסיסי נתונים

מיני פרויקט בבסיסי נתונים

פרק א'

תיאור כללי:

בפרויקט זה נממש מערכת ניהול מידע עבור חנות. המערכת מכילה ישויות שונות וקשרים ביניהם על מנת למפות את כל המידע בצורה מיטבית. מטרת הפרויקט היא לעזור לנהל את החנות בצורה יעילה ומתועדת היטב.

הישויות במערכת והקשרים ביניהם:

לקוחות – אנשים שקונים בחנות

עובדים – אנשים העובדים בחנות

ספקים – ספקי סחורה לחנות

מוצרים – המוצרים הנמכרים בחנות

קטגוריות – שמות הקטגוריות שעל פיהן נמיין את המוצרים שבחנות. לכל מוצר יש קטגוריה אחת לפחות.

הזמנות – הזמנת מוצרים חדשים בין העובדים לספקים

רכישות – רכישת מוצרים על ידי הלקוחות מהעובדים

Entities:

Client = Client_ID, Client_Name, Is_Club_Member

Worker = Worker_Id, Worker_Name, Start_of_Work_Date

Supplier = Supplier_Id, Supplier_Name, Region

Products = Product_Id, Product_Name, Quantity

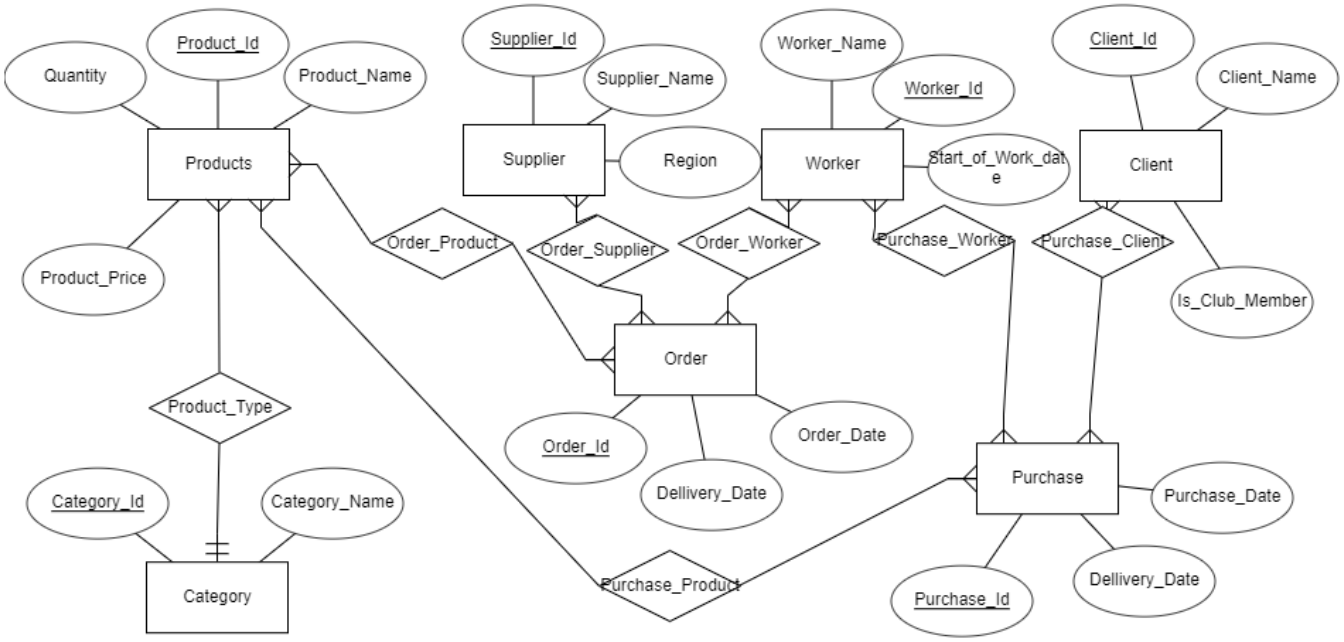
Category = Category_Id, Category_Name

Order = Order_Id, Order_Date, Delivery_Date, Quantity

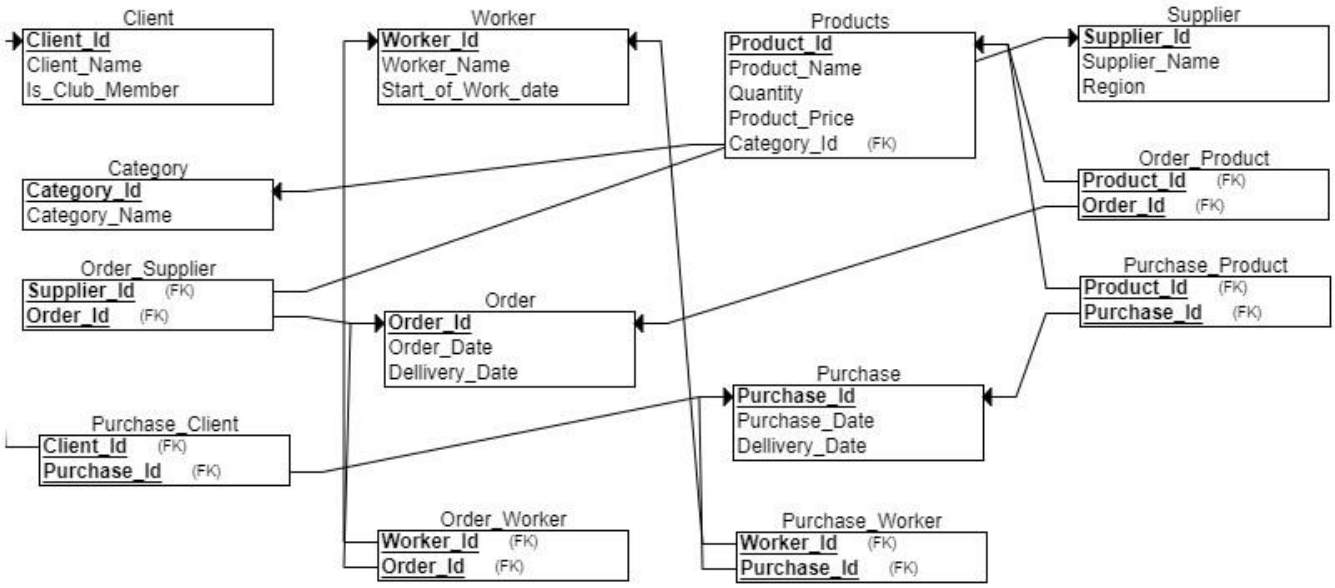
Purchase = Purchase_Id, Purchase_Date, Delivery_Date, Quantity

מיני פרויקט בבסיסי נתונים

ERD Diagram

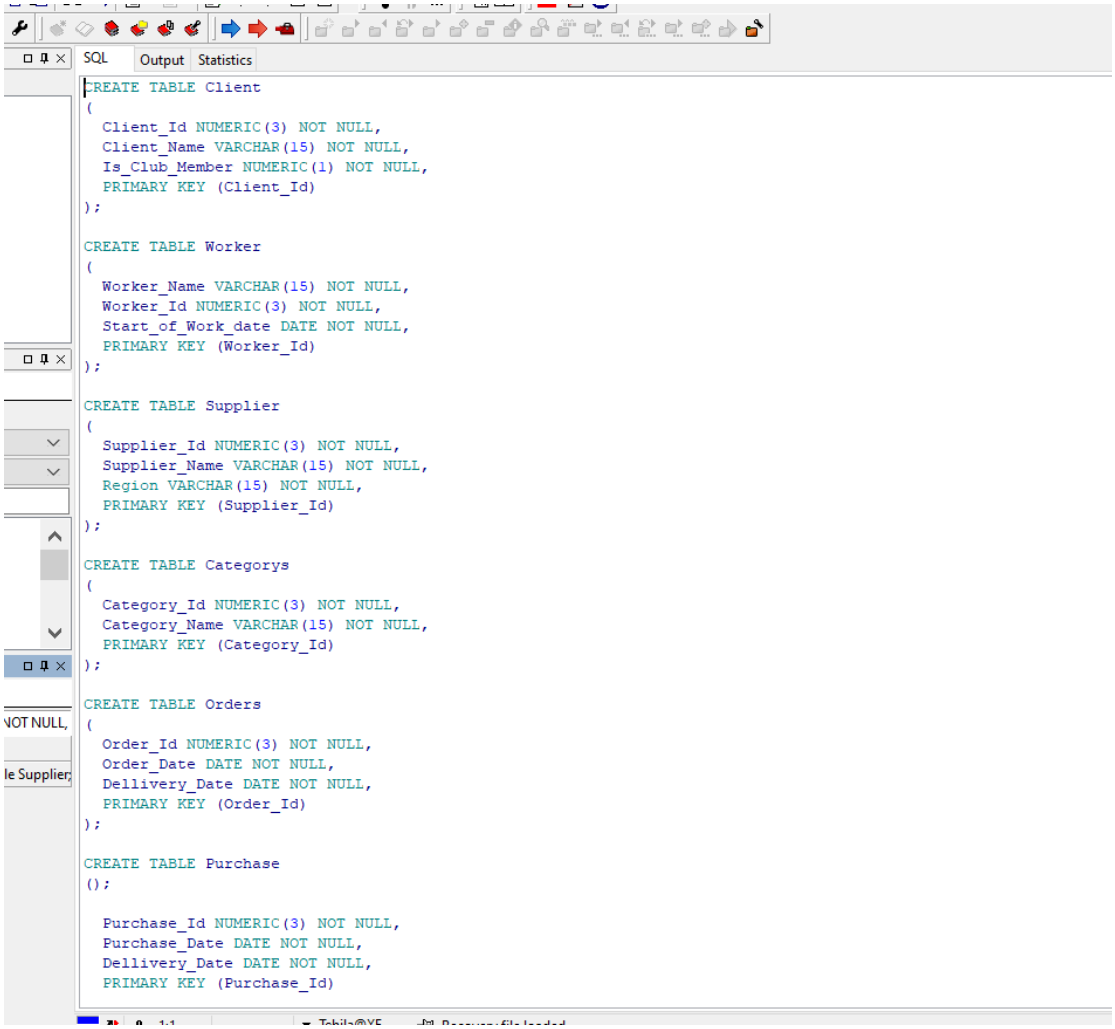


DSD Diagram



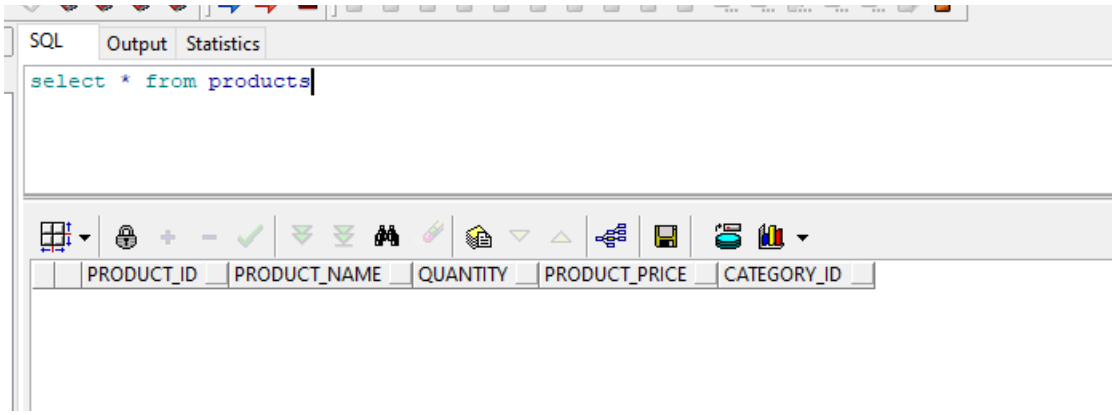
מיני פרויקט בבסיסי נתונים

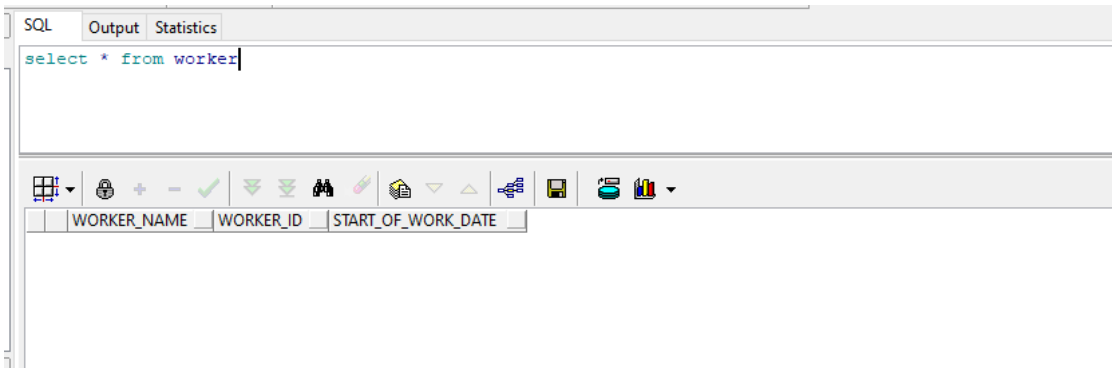
יצירת הטבלאות ע"י createTable



מיני פרויקט בבסיסי נתונים

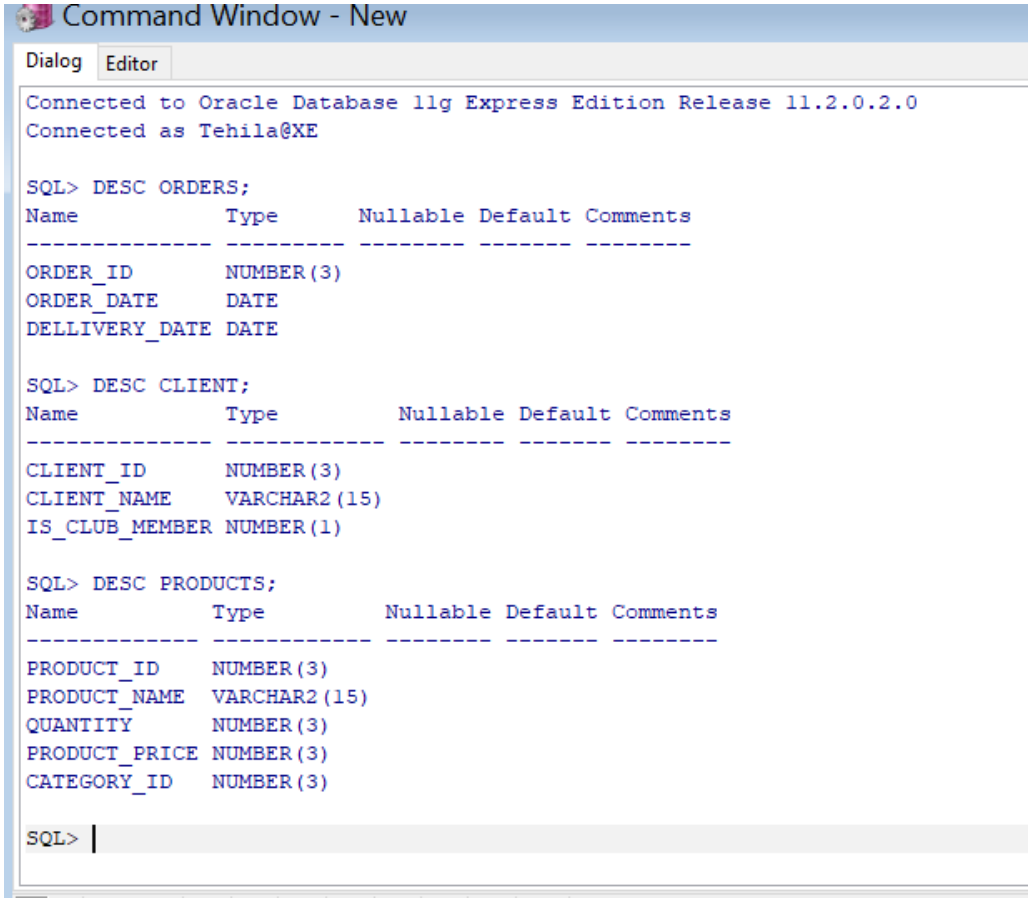
בדקנו שאכן הטבלאות נוצרו על ידי הרצה הסקריפט הזה:





מיני פרויקט בבסיסי נתונים

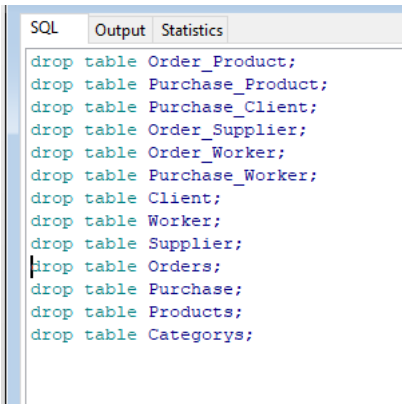
פקודת DESC:



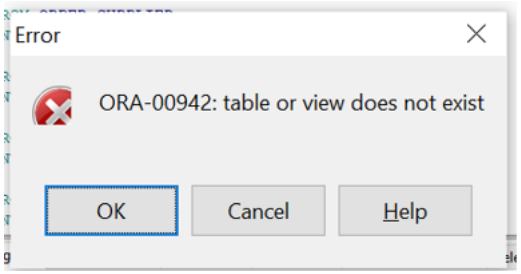
מיני פרויקט בבסיסי נתונים

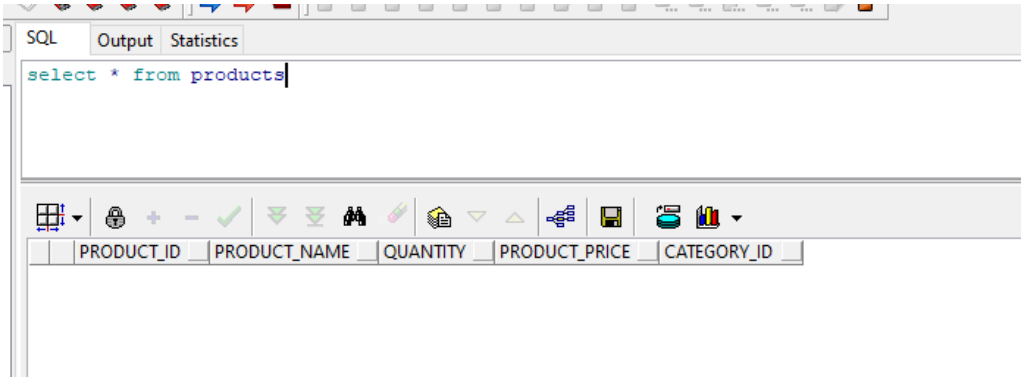
:DROP_TABLE

כמובן על פי הסדר כך שהטבלאות התלויות בטבלאות אחרות נמחקות קודם



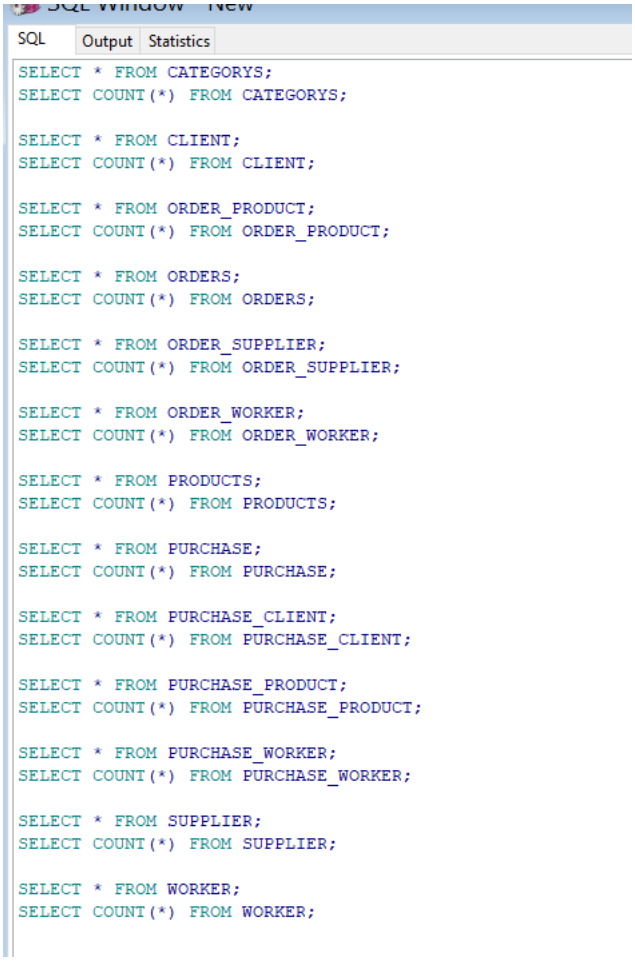
בדקנו שהטבלאות נמחקות ולאחר מכן יצרנו אותן שוב מחדש





מיני פרויקט בבסיסי נתונים

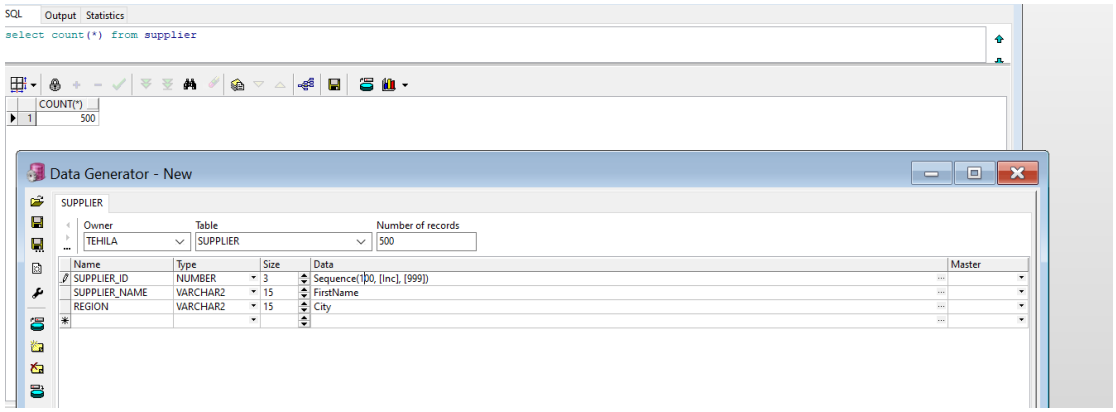
קובץ SELECT

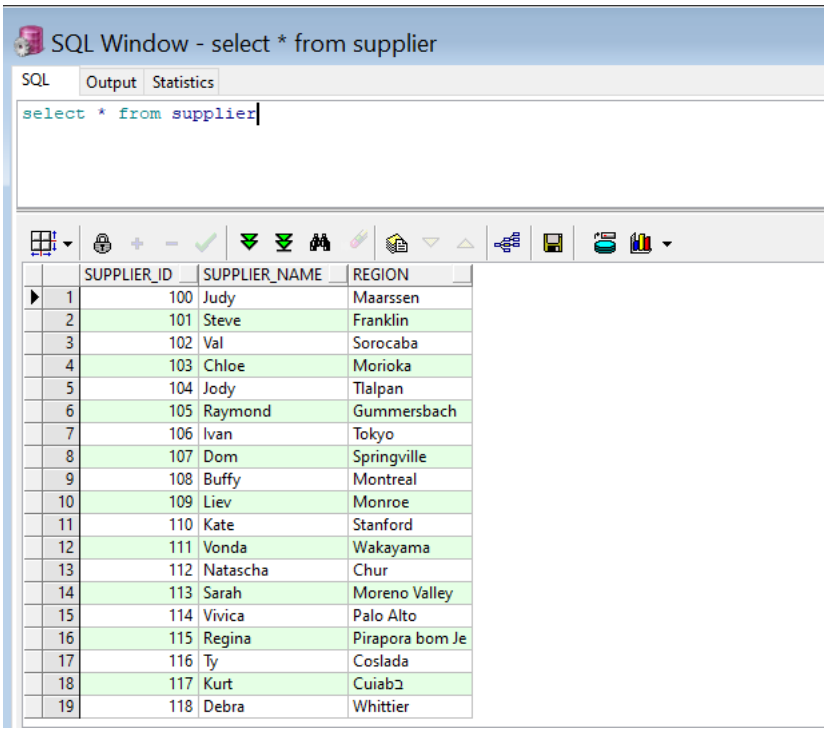


מיני פרויקט בבסיסי נתונים

הכנסת נתונים על ידי שלושת הדרכים המבוקשות:

הכנסת נתונים על ידי DATA GENERATOR:





	SUPPLIER_ID	SUPPLIER_NAME	REGION
1	100	Judy	Maarssen
2	101	Steve	Franklin
3	102	Val	Sorocaba
4	103	Chloe	Morioka
5	104	Jody	Tlalpan
6	105	Raymond	Gummersbach
7	106	Ivan	Tokyo
8	107	Dom	Springville
9	108	Buffy	Montreal
10	109	Liev	Monroe
11	110	Kate	Stanford
12	111	Vonda	Wakayama
13	112	Natascha	Chur
14	113	Sarah	Moreno Valley
15	114	Vivica	Palo Alto
16	115	Regina	Pirapora bom Je
17	116	Ty	Coslada
18	117	Kurt	Cuiabב
19	118	Debra	Whittier

מיני פרויקט בבסיסי נתונים

הכנסת נתונים על ידי קוד בפיתון ופקודות INSERT:

```python
def generate_insert_statement(existing_ids):
    purchase_id = generate_unique_purchase_id(existing_ids)
    existing_ids.add(purchase_id)
    purchase_date = generate_random_date()
    dellivey_date = generate_random_date()
    return f"INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES({purchase_id}, (TO_DATE('{purchase

if __name__ == "__main__":
    # Set to keep track of existing IDs
    existing_ids = set()

    # Generate multiple insert statements
    for _ in range(400):  # Change the range to generate more or fewer statements
        print(generate_insert_statement(existing_ids))
```

```
"C:\Users\Tehila Benezra\PycharmProjects\test2022b\venv\Scripts\python.exe" C:\assembly\xordll.py
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(560, (TO_DATE('2027/10/03', 'yyyy/mm/dd')), (TO_DATE('2030/07/04', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(949, (TO_DATE('2025/07/29', 'yyyy/mm/dd')), (TO_DATE('2024/06/18', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(207, (TO_DATE('2024/01/29', 'yyyy/mm/dd')), (TO_DATE('2028/10/27', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(248, (TO_DATE('2025/06/08', 'yyyy/mm/dd')), (TO_DATE('2030/08/27', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(668, (TO_DATE('2027/12/27', 'yyyy/mm/dd')), (TO_DATE('2029/03/19', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(740, (TO_DATE('2030/08/19', 'yyyy/mm/dd')), (TO_DATE('2025/07/25', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(304, (TO_DATE('2025/07/25', 'yyyy/mm/dd')), (TO_DATE('2028/05/30', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(960, (TO_DATE('2027/11/23', 'yyyy/mm/dd')), (TO_DATE('2026/07/31', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(751, (TO_DATE('2026/01/11', 'yyyy/mm/dd')), (TO_DATE('2028/03/24', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(799, (TO_DATE('2028/07/10', 'yyyy/mm/dd')), (TO_DATE('2030/02/01', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(829, (TO_DATE('2026/01/12', 'yyyy/mm/dd')), (TO_DATE('2024/05/06', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(643, (TO_DATE('2029/09/11', 'yyyy/mm/dd')), (TO_DATE('2026/09/21', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(880, (TO_DATE('2028/08/07', 'yyyy/mm/dd')), (TO_DATE('2030/09/30', 'yyyy/mm/dd')));
INSERT INTO TEHILA.purchase(PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE) VALUES(858, (TO_DATE('2029/11/19', 'yyyy/mm/dd')), (TO_DATE('2024/10/29', 'yyyy/mm/dd')));
```

SQL Window - New

SQL | Output | Statistics

```
insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (604, to_date('27-03-2025', 'dd-mm-yyyy'), to_date('16-10-2027', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (319, to_date('08-02-2025', 'dd-mm-yyyy'), to_date('02-05-2030', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (348, to_date('28-03-2025', 'dd-mm-yyyy'), to_date('27-01-2025', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (179, to_date('15-08-2025', 'dd-mm-yyyy'), to_date('14-08-2027', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (388, to_date('23-12-2026', 'dd-mm-yyyy'), to_date('07-05-2029', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (863, to_date('24-07-2025', 'dd-mm-yyyy'), to_date('14-03-2026', 'dd-mm-yyyy'));

insert into purchase (PURCHASE_ID, PURCHASE_DATE, DELLIVERY_DATE)
values (116, to_date('21-05-2029', 'dd-mm-yyyy'), to_date('23-12-2025', 'dd-mm-yyyy'));
```

SQL | Output | Statistics

```
select * from purchase
```

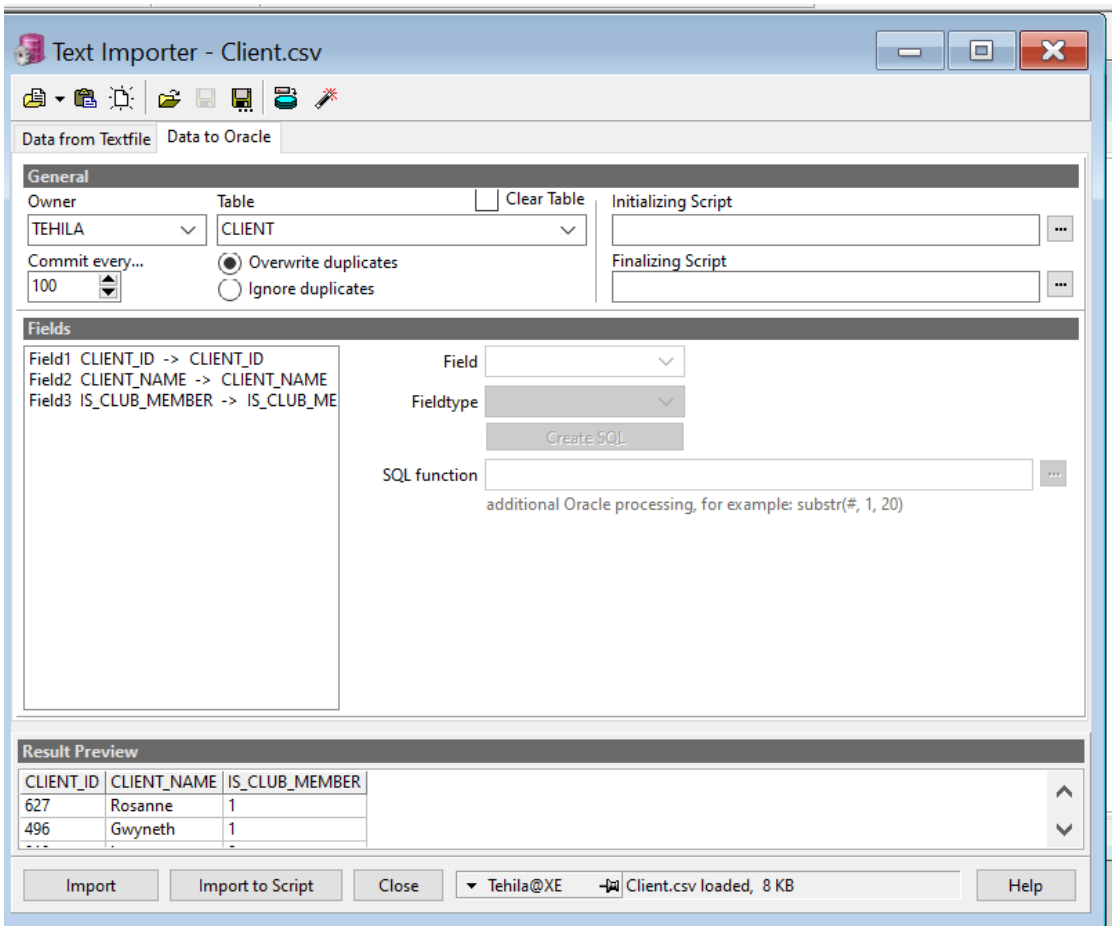
	PURCHASE_ID	PURCHASE_DATE	DELLIVERY_DATE
1	604	27/03/2025	16/10/2027
2	319	08/02/2025	02/05/2030
3	348	28/03/2025	27/01/2025
4	179	15/08/2025	14/08/2027
5	388	23/12/2026	07/05/2029
6	863	24/07/2025	14/03/2026
7	116	21/05/2029	23/12/2025
8	814	22/07/2029	05/08/2026
9	474	04/01/2024	01/06/2024
10	436	02/08/2028	21/10/2024
11	339	11/09/2025	17/01/2030
12	464	19/11/2028	06/02/2025
13	676	29/05/2025	20/01/2030

מיני פרויקט בבסיסי נתונים

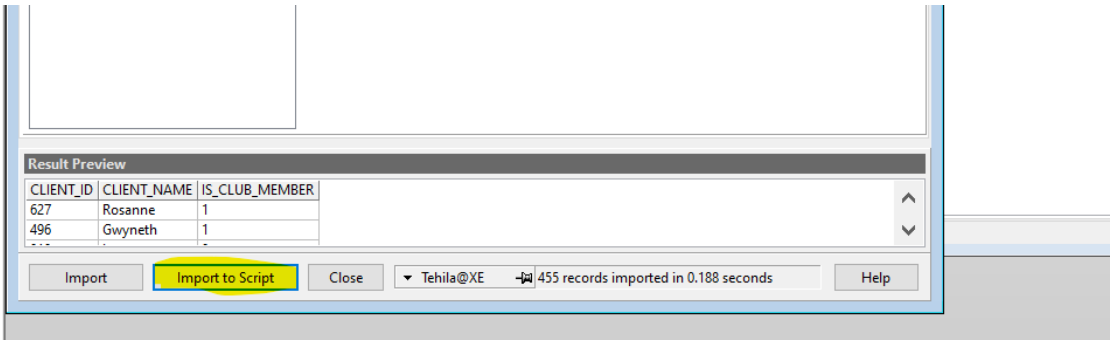
הכנסת נתונים מקובץ:

	E	D	C	B	A	
			IS_CLUB_	CLIENT_N	CLIENT_I	1
			1	Rosanne	627	2
			1	Gwyneth	496	3
			0	Larry	818	4
			0	Joey	206	5
			0	Eliza	563	6
			1	Robin	126	7
			1	Gina	615	8
			0	Nastassja	551	9
			0	Angela	592	10
			0	Domingo	898	11
			1	Adam	700	12
			0	Simon	727	13
			0	Leon	115	14
			1	Anita	349	15
			1	Lauren	244	16
			0	Henry	487	17
			1	Famke	685	18
			0	Mia	462	19
			0	Celia	544	20
			0	Kevn	193	21
			0	Jack	951	22

Client



מיני פרויקט בבסיסי נתונים



SQL Output Statistics

```
insert into tehila.client (client_id, client_name, is_club_member)
values (627, 'Rosanne', 1);

insert into tehila.client (client_id, client_name, is_club_member)
values (496, 'Gwyneth', 1);

insert into tehila.client (client_id, client_name, is_club_member)
values (818, 'Larry', 0);

insert into tehila.client (client_id, client_name, is_club_member)
values (206, 'Joey', 0);

insert into tehila.client (client_id, client_name, is_club_member)
values (563, 'Eliza', 0);

insert into tehila.client (client_id, client_name, is_club_member)
values (126, 'Robin', 1);

insert into tehila.client (client_id, client_name, is_club_member)
values (615, 'Gina', 1);

insert into tehila.client (client_id, client_name, is_club_member)
values (551, 'Nastassja', 0);

insert into tehila.client (client_id, client_name, is_club_member)
values (592, 'Angela', 0);
```

SQL Window - SELECT * FROM CLIENT

SQL Output Statistics

```
SELECT * FROM CLIENT
```

	CLIENT_ID	CLIENT_NAME	IS_CLUB_MEMBER
1	627	Rosanne	1
2	496	Gwyneth	1
3	818	Larry	0
4	206	Joey	0
5	563	Eliza	0
6	126	Robin	1
7	615	Gina	1
8	551	Nastassja	0
9	592	Angela	0
10	898	Domingo	0
11	700	Adam	1
12	727	Simon	0
13	115	Leon	0
14	349	Anita	1
15	244	Lauren	1
16	487	Henry	0
17	685	Famke	1
18	462	Mia	0

מיני פרויקט בבסיסי נתונים

יצירת קובץ INSERT

שכולל כ- 10 שורות לכל אחת מהטבלאות כנדרש

```
insert into client (CLIENT_ID, CLIENT_NAME, IS_CLUB_MEMBER)
values (727, 'Simon', 0);

insert into client (CLIENT_ID, CLIENT_NAME, IS_CLUB_MEMBER)
values (115, 'Leon', 0);

insert into client (CLIENT_ID, CLIENT_NAME, IS_CLUB_MEMBER)
values (349, 'Anita', 1);

insert into client (CLIENT_ID, CLIENT_NAME, IS_CLUB_MEMBER)
values (244, 'Lauren', 1);

insert into CATEGORYS (CATEGORY_ID, CATEGORY_NAME)
values (100, 'irure ipsum do.');

insert into CATEGORYS (CATEGORY_ID, CATEGORY_NAME)
values (101, 'rerum sit. ');

insert into CATEGORYS (CATEGORY_ID, CATEGORY_NAME)
values (102, 'officiis. ');
```

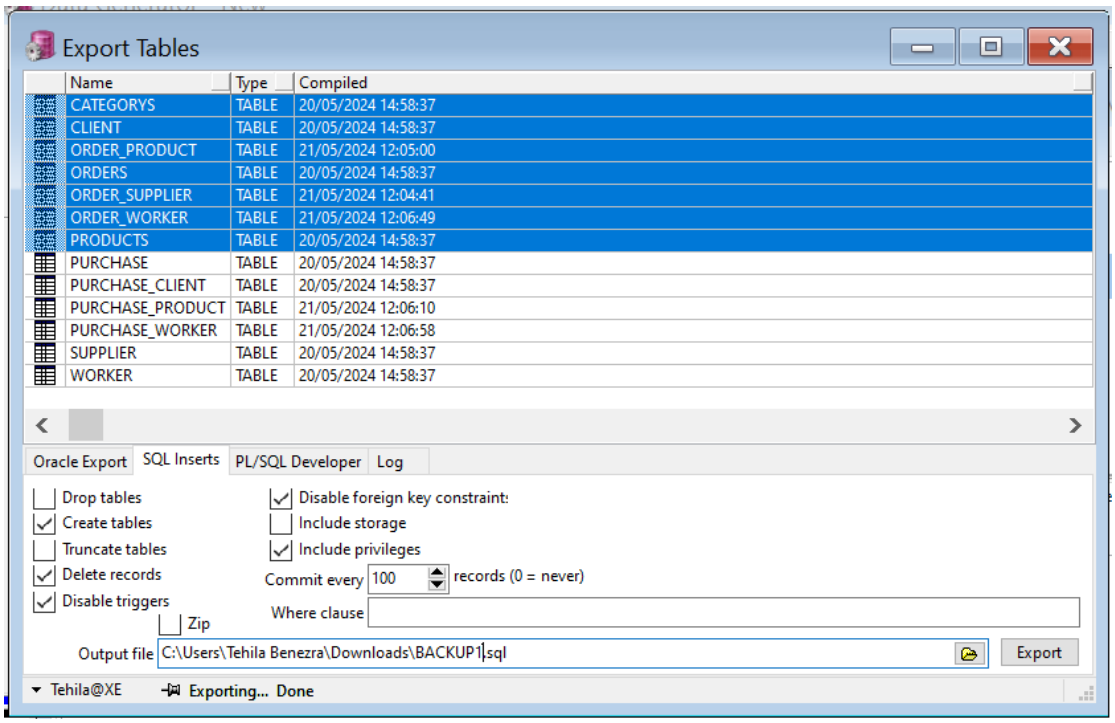
SQL | Output | Statistics

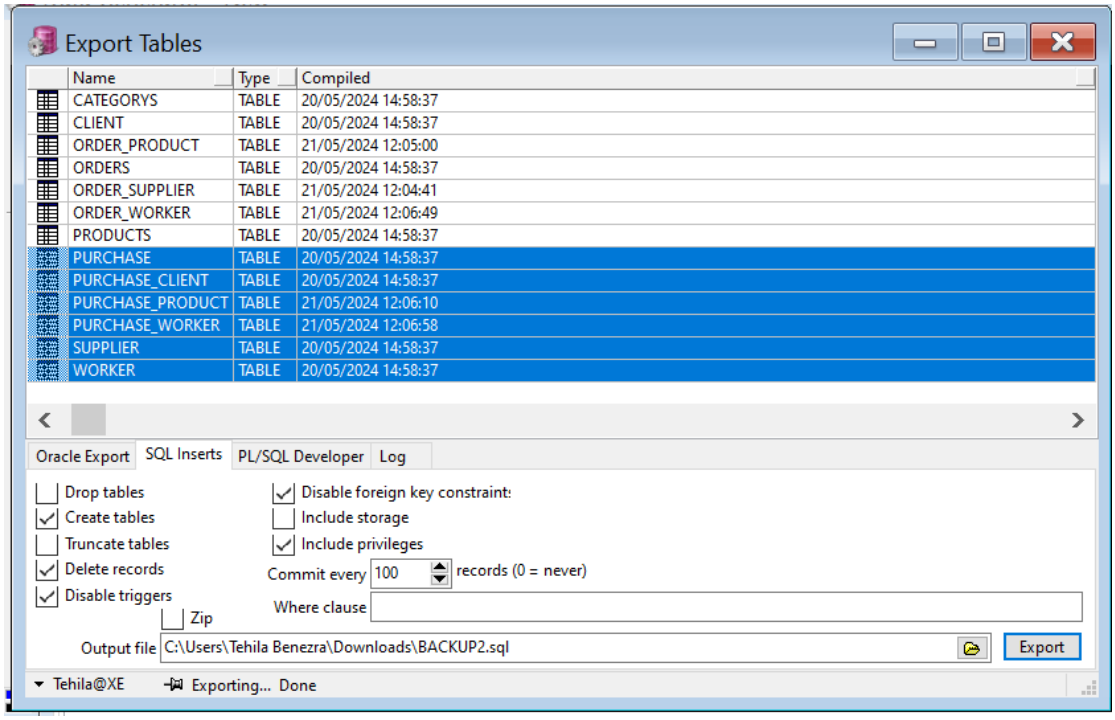
```
insert into ORDER_PRODUCT (PRODUCT_ID, ORDER_ID)
values (111, 611);

insert into ORDER_PRODUCT (PRODUCT_ID, ORDER_ID)
values (117, 617);

insert into ORDER_PRODUCT (PRODUCT_ID, ORDER_ID)
values (120, 120);

insert into ORDER_PRODUCT (PRODUCT_ID, ORDER_ID)
values (124, 124);

insert into ORDER_PRODUCT (PRODUCT_ID, ORDER_ID)
values (126, 626);
insert into ORDERS (ORDER_ID, ORDER_DATE, DELLIVERY_DATE)
values (493, to_date('25-03-2025', 'dd-mm-yyyy'), to_date('03-06-2024', 'dd-mm-yyyy'));

insert into ORDERS (ORDER_ID, ORDER_DATE, DELLIVERY_DATE)
values (377, to_date('04-02-2026', 'dd-mm-yyyy'), to_date('23-05-2025', 'dd-mm-yyyy'));

insert into ORDERS (ORDER_ID, ORDER_DATE, DELLIVERY_DATE)
values (323, to_date('12-06-2026', 'dd-mm-yyyy'), to_date('10-04-2029', 'dd-mm-yyyy'));

insert into ORDERS (ORDER_ID, ORDER_DATE, DELLIVERY_DATE)
values (703, to_date('12-08-2025', 'dd-mm-yyyy'), to_date('24-12-2025', 'dd-mm-yyyy'));
```

501:53 | Tehila@XE | SQL script saved successfully

מיני פרויקט בבסיסי נתונים

גיבוי (הגיבוי בוצע ב2 חלקים כיוון שלא היה מקום)





מיני פרויקט בבסיסי נתונים

BACKUP1.sql

C: > Users > Tehila Benezra > Downloads > BACKUP1.sql

```
prompt PL/SQL Developer import file
prompt Created on יום חמישי 23 אוק 2024 by Tehila Benezra
set feedback off
set define off
prompt Creating CATEGORYS...
create table CATEGORYS
(
  category_id   NUMBER(3) not null,
  category_name VARCHAR2(15) not null
)
;
alter table CATEGORYS
  add primary key (CATEGORY_ID);

prompt Creating CLIENT...
create table CLIENT
(
  client_id      NUMBER(3) not null,
  client_name    VARCHAR2(15) not null,
  is_club_member NUMBER(1) not null
)
;
alter table CLIENT
  add primary key (CLIENT_ID);

prompt Creating ORDERS...
create table ORDERS
(
  order_id       NUMBER(3) not null,
  order_date     DATE not null,
```

C: > Users > Tehila Benezra > Downloads > BACKUP2.sql

```
prompt PL/SQL Developer import file
prompt Created on יום חמישי 23 אוק 2024 by Tehila Benezra
set feedback off
set define off
prompt Creating PURCHASE...
create table PURCHASE
(
  purchase_id    NUMBER(3) not null,
  purchase_date  DATE not null,
  dellivery_date DATE not null
)
;
alter table PURCHASE
  add primary key (PURCHASE_ID);

prompt Creating PURCHASE_CLIENT...
create table PURCHASE_CLIENT
(
  client_id   NUMBER(3) not null,
  purchase_id NUMBER(3) not null
)
;
alter table PURCHASE_CLIENT
  add primary key (CLIENT_ID, PURCHASE_ID);
alter table PURCHASE_CLIENT
  add foreign key (CLIENT_ID)
  references CLIENT (CLIENT_ID);
alter table PURCHASE_CLIENT
  add foreign key (PURCHASE_ID)
  references PURCHASE (PURCHASE ID);
```

מיני פרויקט בבסיסי נתונים

מחיקת הפרויקט

SQL Output Statistics

```
drop table Order_Product;
drop table Purchase_Product;
drop table Purchase_Client;
drop table Order_Supplier;
drop table Order_Worker;
drop table Purchase_Worker;
drop table Client;
drop table Worker;
drop table Supplier;
drop table Orders;
drop table Purchase;
drop table Products;
drop table Categorys;
```

SQL Output Statistics

```
SELECT * FROM CATEGORYS;
SELECT COUNT(*) FROM CATEGORYS;

SELECT * FROM CLIENT;
SELECT COUNT(*) FROM CLIENT;

SELECT * FROM ORDER_PRODUCT;
SELECT COUNT(*) FROM ORDER_PRODUCT;

SELECT * FROM ORDERS;
SELECT COUNT(*) FROM ORDERS;
```

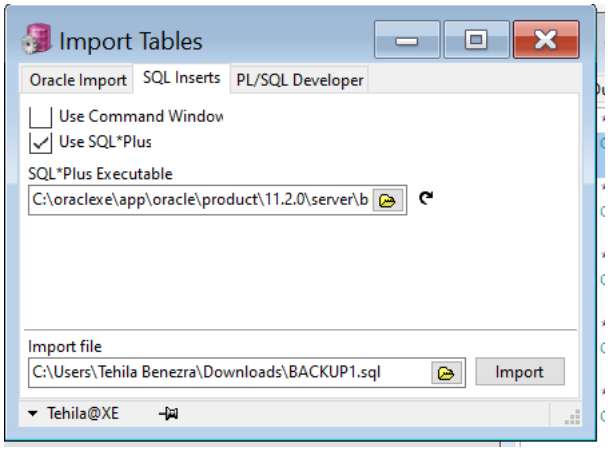
Error

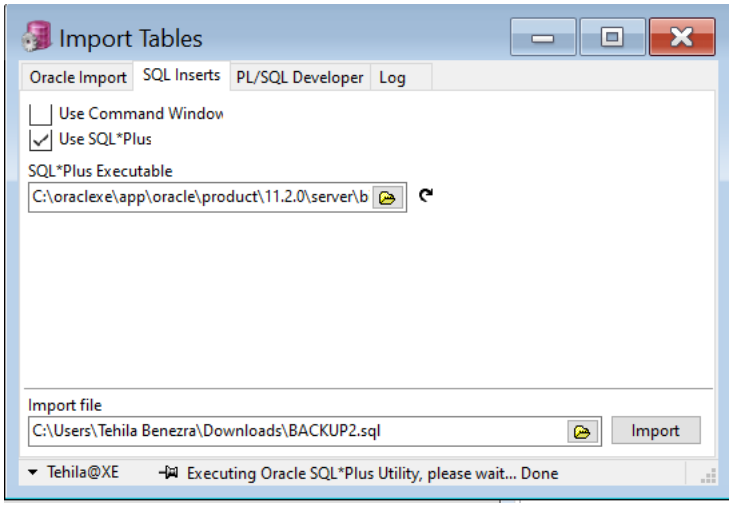
ORA-00942: table or view does not exist

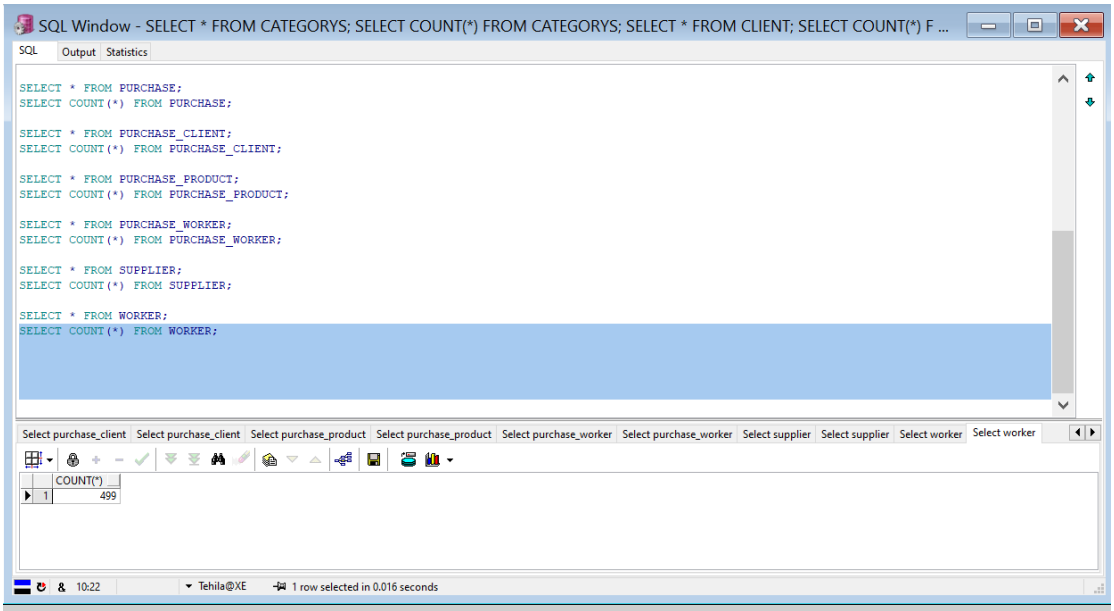
OK Cancel Help

מיני פרויקט בבסיסי נתונים

שחזור הפרויקט (כמובן עלינו לייבא את שני קבצי הגיבוי)





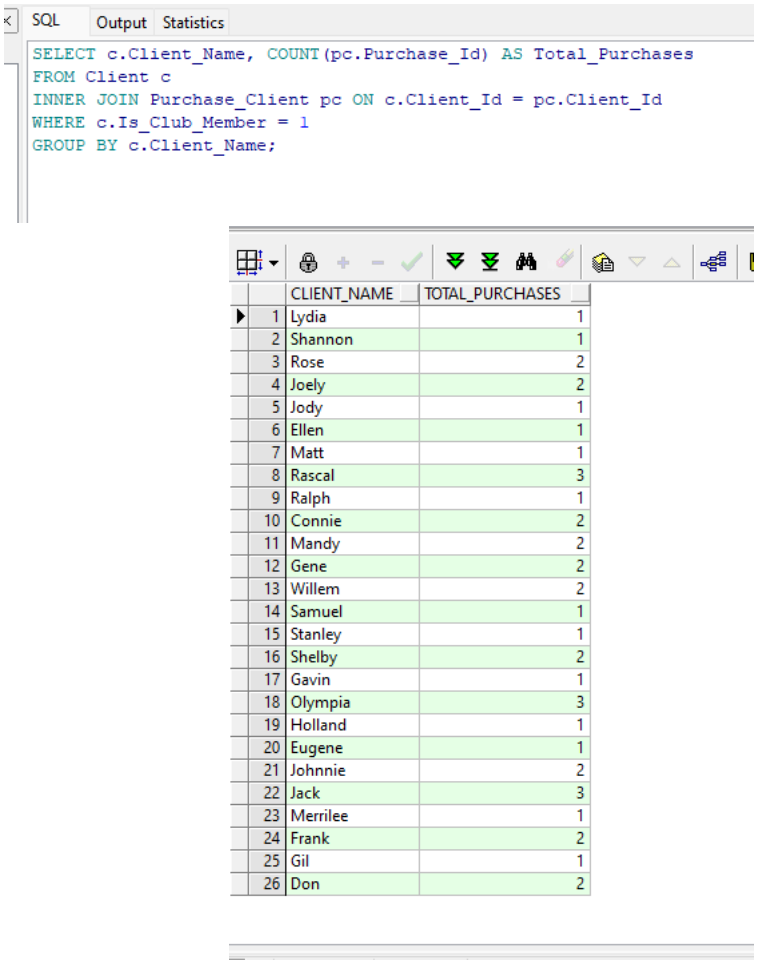


הפרויקט שוחזר בהצלחה.

פרק ב

Select query 1:

מחזיר את מספר הלקוחות שיש להם מועדון , ואת מספר ההזמנות שרכשו



	CLIENT_NAME	TOTAL_PURCHASES
1	Lydia	1
2	Shannon	1
3	Rose	2
4	Joely	2
5	Jody	1
6	Ellen	1
7	Matt	1
8	Rascal	3
9	Ralph	1
10	Connie	2
11	Mandy	2
12	Gene	2
13	Willem	2
14	Samuel	1
15	Stanley	1
16	Shelby	2
17	Gavin	1
18	Olympia	3
19	Holland	1
20	Eugene	1
21	Johnnie	2
22	Jack	3
23	Merrilee	1
24	Frank	2
25	Gil	1
26	Don	2

Select query 2:

השאילתה מחזירה רשימה של קטגוריות מוצרים יחד עם המחיר הממוצע של המוצרים בכל קטגוריה, ממוין מהמחיר הממוצע הגבוה ביותר לנמוך ביותר. זה מאפשר השוואה קלה של מחירי המוצרים הממוצעים בקטגוריות שונות.

SQL Output Statistics

```
SELECT
    C.Category_Name,
    AVG(PR.Product_Price) AS Average_Price
FROM
    Categorys C
JOIN
    Products PR ON C.Category_Id = PR.Category_Id
GROUP BY
    C.Category_Name
ORDER BY
    Average_Price DESC;
```

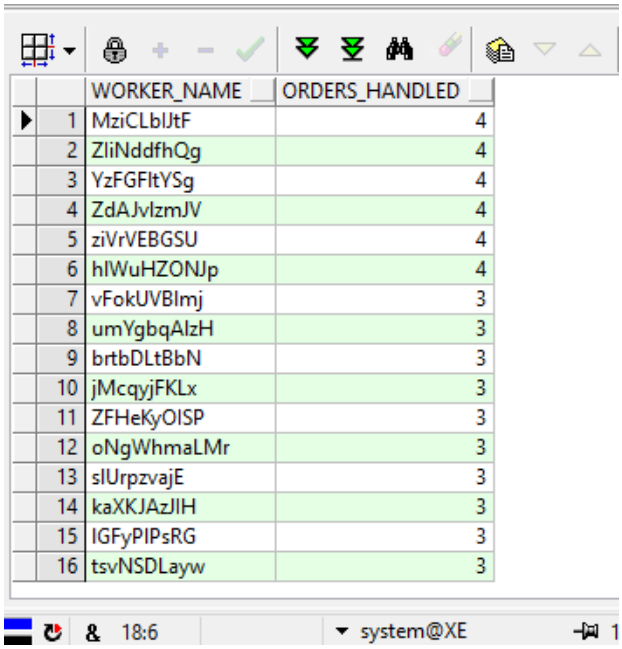
	CATEGORY_NAME	AVERAGE_PRICE
1	irure ipsum do.	200
2	amet do	99
3	alias aut	99
4	minus at	99
5	laborum	97
6	dolores	96
7	sunt	95
8	ut culpa	94
9	non libero	94
10	quod	93
11	saepe in	93
12	earum	91
13	in qui	90
14	do	89.8571428571428
15	vero ut	88
16	proident ut	87
17	et vero qui	85
18	amet aute	84
19	nostrud	83.5

6:6 system@XE 19 rc

Select query 3:

שאילתה לרשימת עובדים שטיפלו ביותר הזמנות ממספר ההזמנות הממוצע שטופלו על ידי כל העובדים, מסודרים לפי מספר ההזמנות שטופלו בסדר יורד:

```
SELECT
    w.Worker_Name,
    COUNT(ow.Order_Id) AS Orders_Handled
FROM
    Worker w
JOIN
    Order_Worker ow ON w.Worker_Id = ow.Worker_Id
GROUP BY
    w.Worker_Name
HAVING
    COUNT(ow.Order_Id) > (
        SELECT AVG(Order_Count)
        FROM (
            SELECT COUNT(Order_Id) AS Order_Count
            FROM Order_Worker
            GROUP BY Worker_Id
        )
    )
ORDER BY
    Orders_Handled DESC;
```



	WORKER_NAME	ORDERS_HANDLED
1	MziCLblJtF	4
2	ZliNddfhQg	4
3	YzFGFltYSg	4
4	ZdAJvlzmJV	4
5	ziVrVEBGSU	4
6	hlWuHZONJp	4
7	vFokUVBlmj	3
8	umYgbqAlzH	3
9	brtbDLtBbN	3
10	jMcqyjFKLx	3
11	ZFHeKyOISP	3
12	oNgWhmaLMr	3
13	slUrpzvajE	3
14	kaXKJAzJlH	3
15	lGFyPlPsRG	3
16	tsvNSDLayw	3

Select query 4:

השאילתה מחזירה רשימה של עובדים יחד עם ספירת ההזמנות להן הוקצו

```
SQL  Output  Statistics
SELECT w.Worker_Name, COUNT(ow.Order_Id) AS Assigned_Order_Count
FROM Worker w
LEFT JOIN Order_Worker ow ON w.Worker_Id = ow.Worker_Id
GROUP BY w.Worker_Name;
```

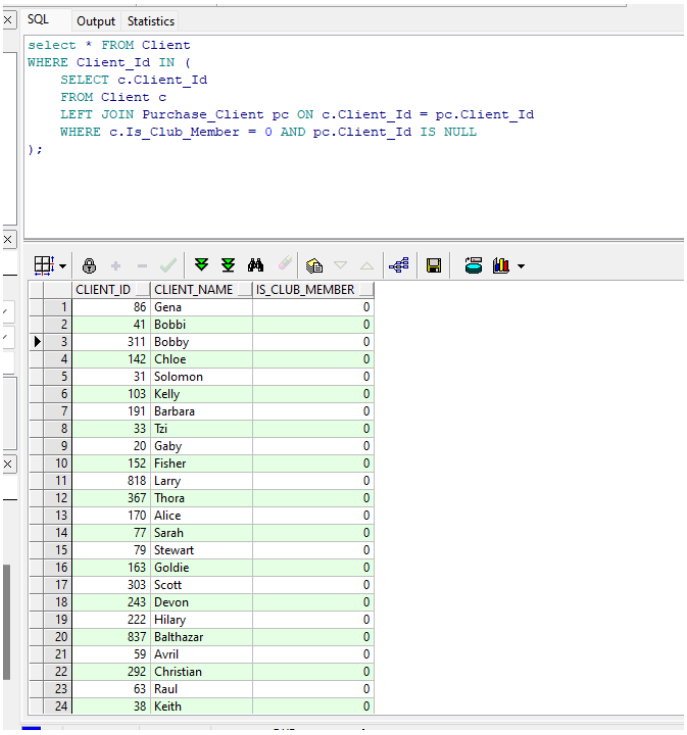
	WORKER_NAME	ASSIGNED_ORDER_COUNT
1	puxetLwKil	1
2	qHyXvQubmi	0
3	LMHneGcjSs	0
4	qBDWviOnPd	1
5	oTFsXBdyST	2
6	RzzphUKGOv	1
7	IGFyPIPsRG	3
8	HxnwyDaczK	1
9	rWiNIKuDwj	2
10	KgZbtaNcPG	2
11	DTHJVLijsL	0
12	aALtGlqGdO	0
13	tsvNSDLayw	3
14	qsrpJdGOqW	0
15	oHWpEQZeSd	2
16	SoEJLkEctp	2
17	xSxBrgiPfY	1

4:23 system@XE 17 ro

Delete query 1:

יצרנו שאילתת מחיקה , מוחקת את הלקוחות שלא חברי מועדון וביצעו רכישות בחנות .

נתון במידע לפני המחיקה:

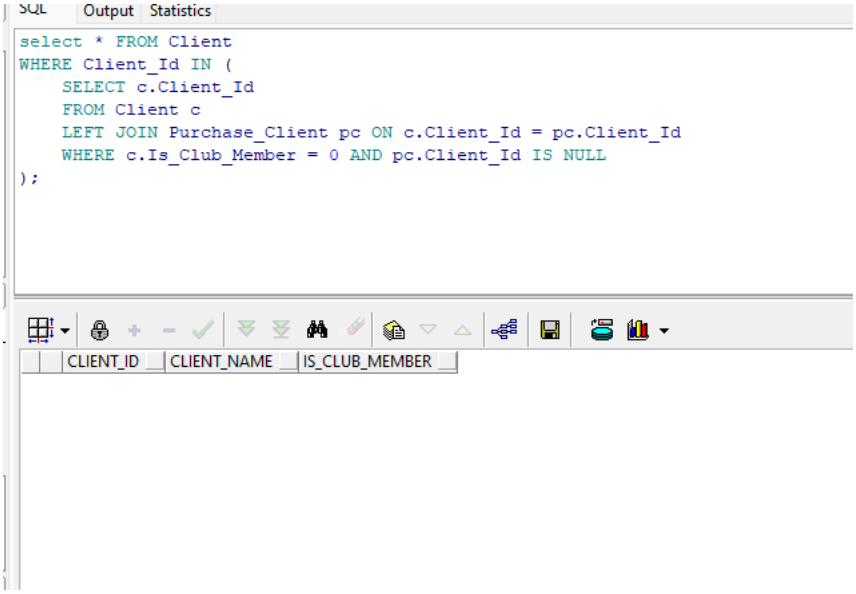


SQL Output Statistics

```
select * FROM Client
WHERE Client_Id IN (
    SELECT c.Client_Id
    FROM Client c
    LEFT JOIN Purchase_Client pc ON c.Client_Id = pc.Client_Id
    WHERE c.Is_Club_Member = 0 AND pc.Client_Id IS NULL
);
```

	CLIENT_ID	CLIENT_NAME	IS_CLUB_MEMBER
1	86	Gena	0
2	41	Bobbi	0
3	311	Bobby	0
4	142	Chloe	0
5	31	Solomon	0
6	103	Kelly	0
7	191	Barbara	0
8	33	Tzi	0
9	20	Gaby	0
10	152	Fisher	0
11	818	Larry	0
12	367	Thora	0
13	170	Alice	0
14	77	Sarah	0
15	79	Stewart	0
16	163	Goldie	0
17	303	Scott	0
18	243	Devon	0
19	222	Hilary	0
20	837	Balthazar	0
21	59	Avril	0
22	292	Christian	0
23	63	Raul	0
24	38	Keith	0

נעשה צילום מסך אחרי המחיקה :

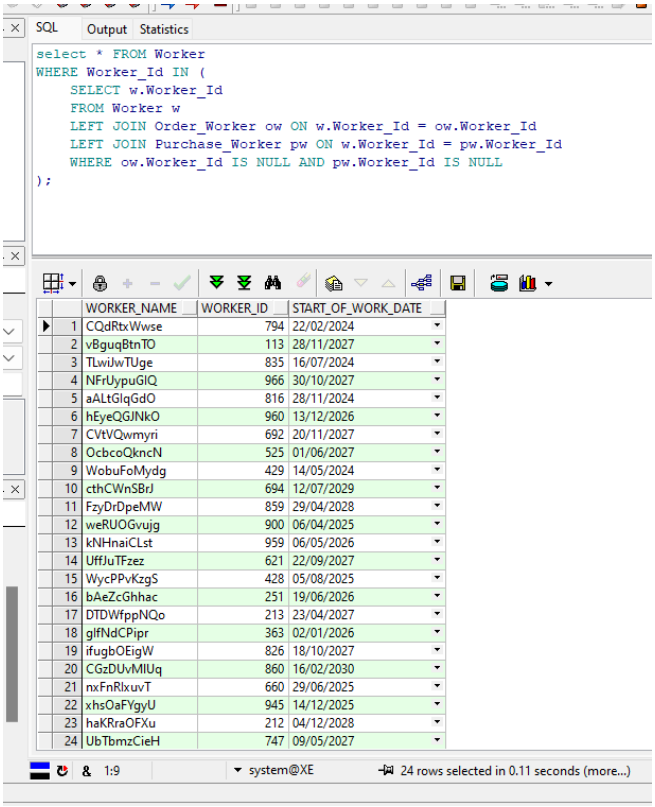


Delete query 2:

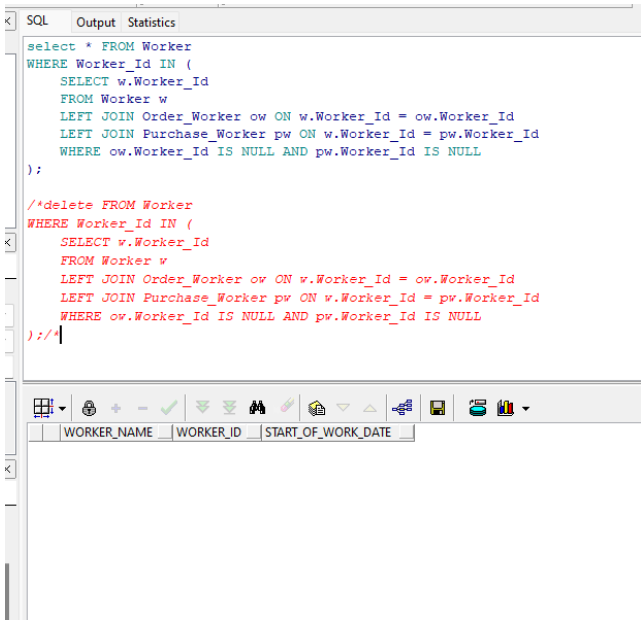
יצרנו שאילתת מחיקה נוספת .

שמוחקת את כל העובדים שמעולם לא הוקצו להזמנות ורכישות

אלא הנתונים לפני המחיקה :



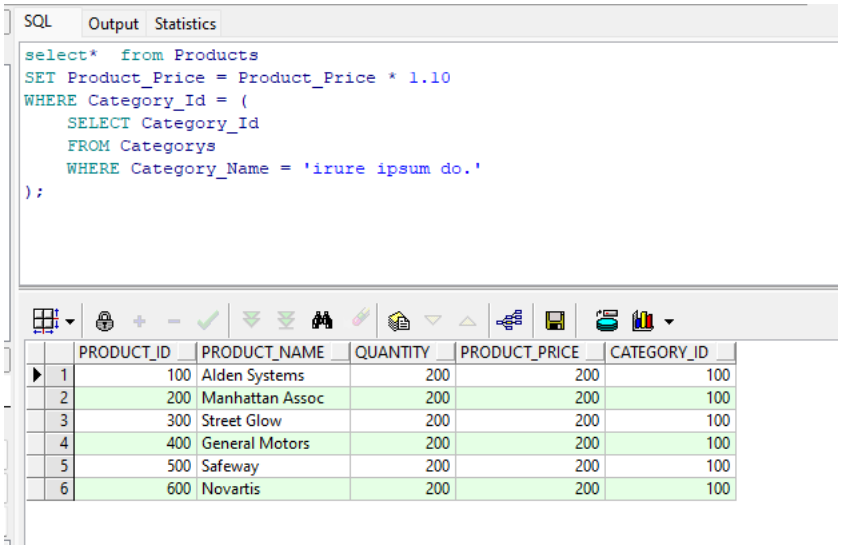
לאחר המחיקה כך נראה בסיס הנתונים שאליו ניסינו לגשת :



Update query 1

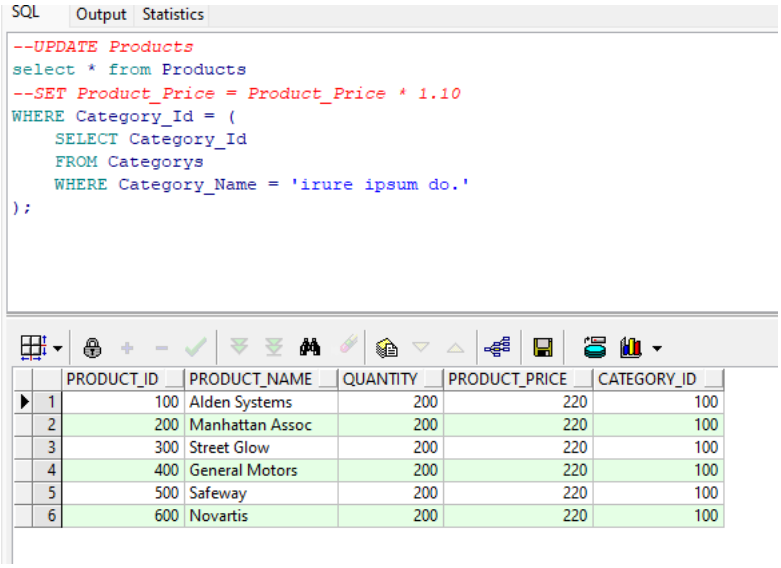
שאילתה שמעלה את המחיר מוצר בקטגוריה מסוימת ב10 אחוז

לפני:



	PRODUCT_ID	PRODUCT_NAME	QUANTITY	PRODUCT_PRICE	CATEGORY_ID
1	100	Alden Systems	200	200	100
2	200	Manhattan Assoc	200	200	100
3	300	Street Glow	200	200	100
4	400	General Motors	200	200	100
5	500	Safeway	200	200	100
6	600	Novartis	200	200	100

אחרי:



	PRODUCT_ID	PRODUCT_NAME	QUANTITY	PRODUCT_PRICE	CATEGORY_ID
1	100	Alden Systems	200	220	100
2	200	Manhattan Assoc	200	220	100
3	300	Street Glow	200	220	100
4	400	General Motors	200	220	100
5	500	Safeway	200	220	100
6	600	Novartis	200	220	100

Update query 2:

שאילתת עדכון שמוסיפה לתאריך משלוח 7 ימים לפני

```
select * from Orders
WHERE Order_Id IN (
    SELECT os.Order_Id
    FROM Order_Supplier os
    JOIN Supplier s ON os.Supplier_Id = s.Supplier_Id
    WHERE s.Region = 'Boucherville'
);
```

	ORDER_ID	ORDER_DATE	DELLIVERY_DATE
1	996	12/08/2026	09/10/2025
2	362	13/02/2025	15/12/2027
3	432	11/10/2028	13/10/2029

אחרי

```
select *
from Orders
WHERE Order_Id IN (
    SELECT os.Order_Id
    FROM Order_Supplier os
    JOIN Supplier s ON os.Supplier_Id = s.Supplier_Id
    WHERE s.Region = 'Boucherville'
);
```

	ORDER_ID	ORDER_DATE	DELLIVERY_DATE
1	996	12/08/2026	16/10/2025
2	362	13/02/2025	22/12/2027
3	432	11/10/2028	20/10/2029

השאילתה

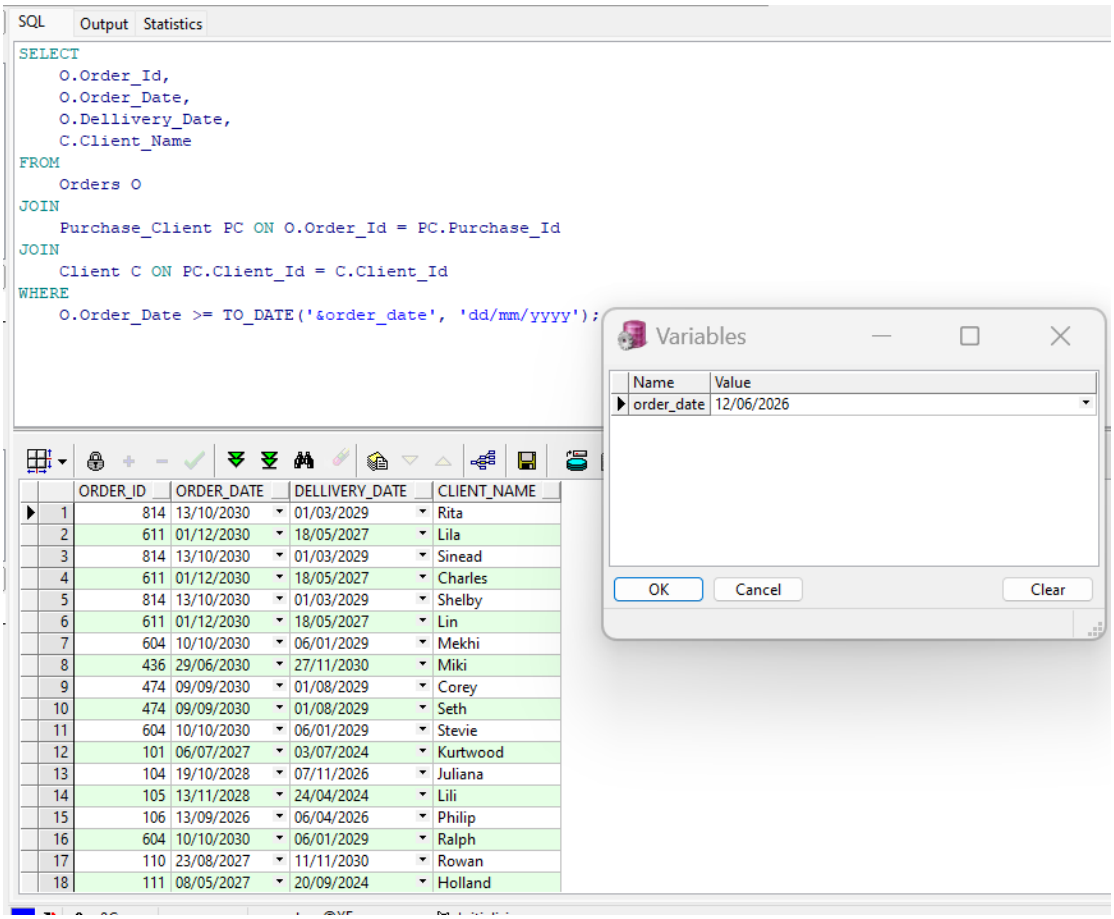
```sql
UPDATE Orders
SET Dellivery_Date = Dellivery_Date + 7
WHERE Order_Id IN (
    SELECT os.Order_Id
    FROM Order_Supplier os
    JOIN Supplier s ON os.Supplier_Id = s.Supplier_Id
    WHERE s.Region = 'Boucherville'
```

Queries with parameters:

1.query with data parameter

שאילתה שמחזירה את כל הנתונים של ההזמנה מתאריך מסוים או אחריו

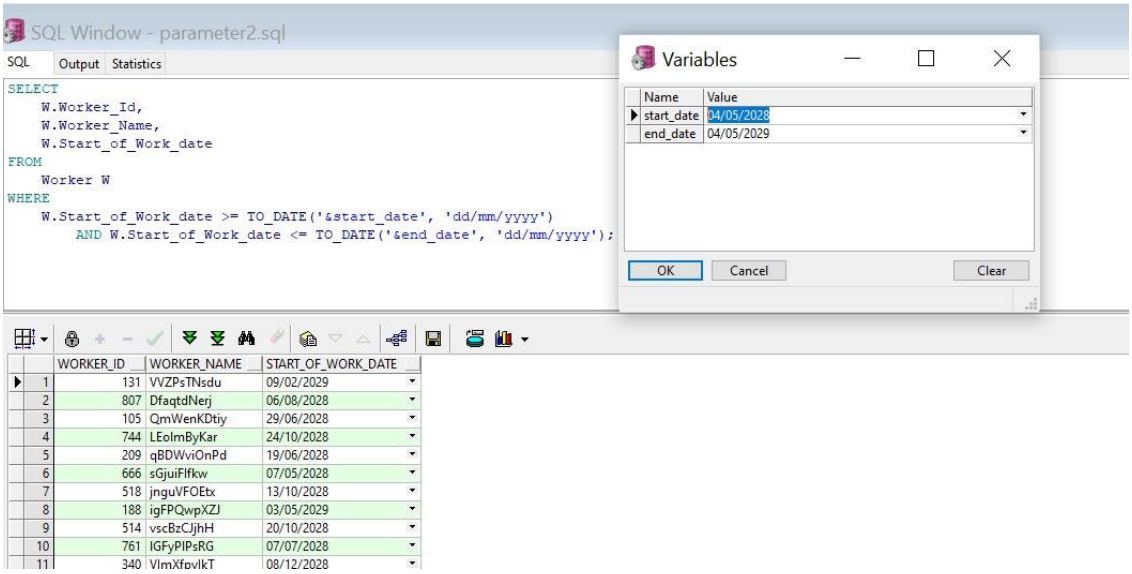
לדוג: כאן קיבלנו את כל ההזמנות שבוצעו מתאריך 12.06.2026



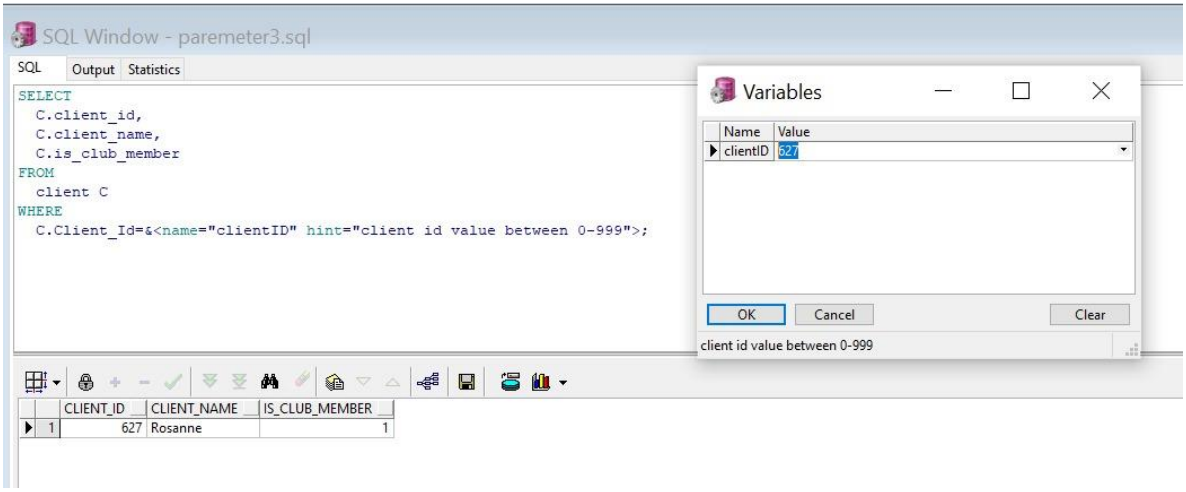
2.query with date parameter

השאילתה מחזירה נתונים של עובד שהחל לעבוד בטווח תאריכים מסוים.

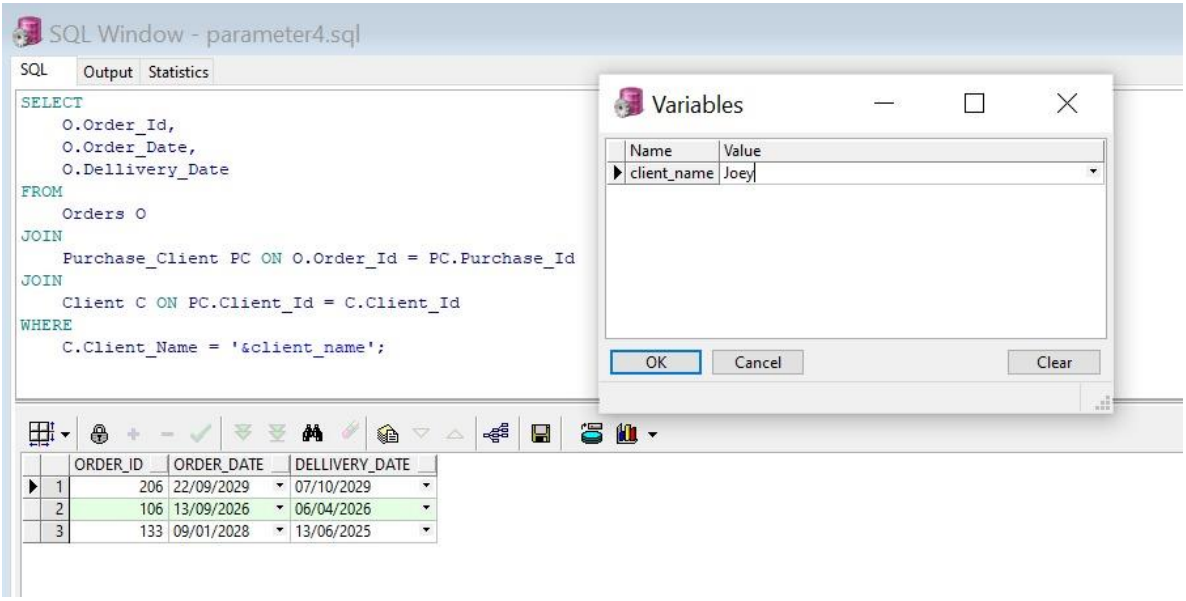
לדוג כאן הוא החזיר נתונים של כל העובדים שתאריך העבודה ההתחלתי שלהם הוא בין 4.5.28 ל 4.5.29



3.query with hint parameter

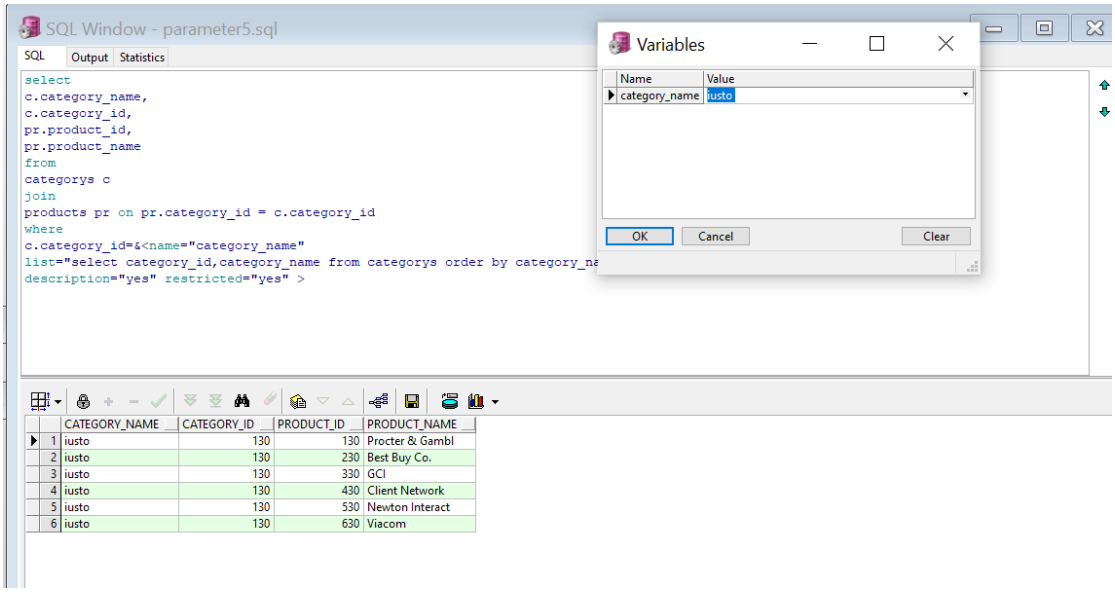


4.query with name parameter



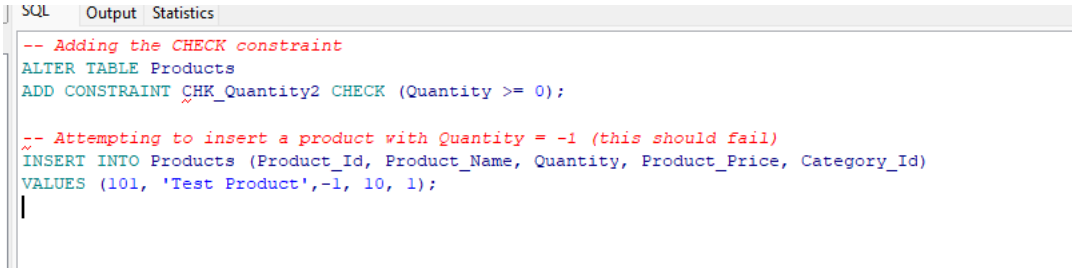
5.query with list parameter

מיכל יששכר
תהילה בן עזרא

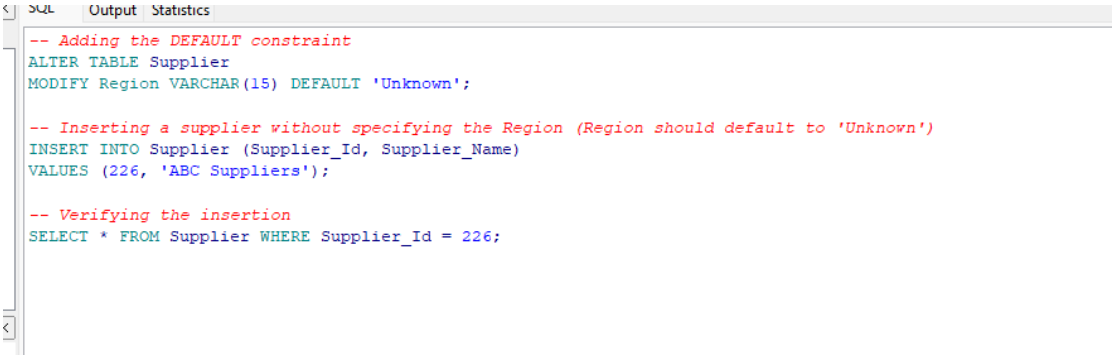


Constraints

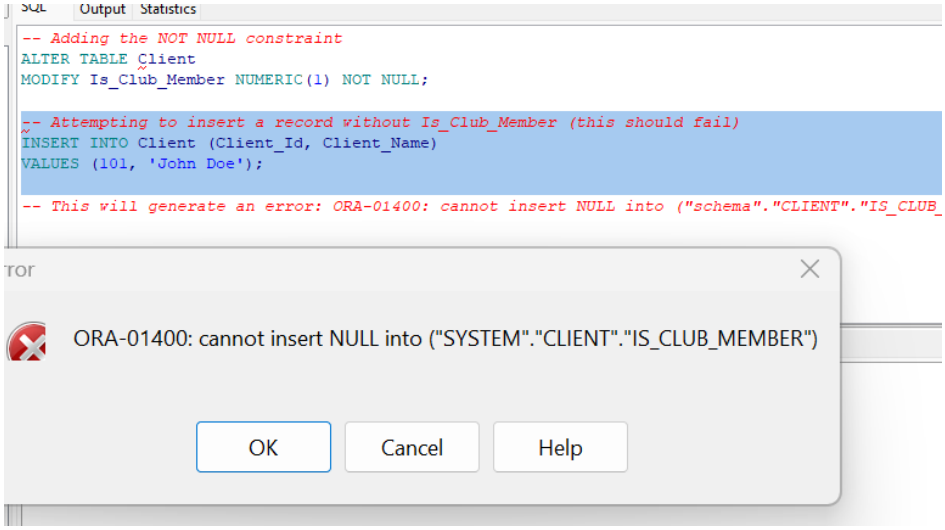
1.alter table –CHECK



2.alter table-DEFAULT



הוא לא נותן לי להוסיף לקוח בגלל שלא הכנסנו ערך ואין אפשרות של הוספת ערך NULL , הגדרנו בטבלאות שאין אפשרות הזנת ערך NULL


Output
Statistics
-- Adding the NOT NULL constraint
ALTER TABLE Client
MODIFY Is_Club_Member NUMERIC(1) NOT NULL;
-- Attempting to insert a record without Is_Club_Member (this should fail)
INSERT INTO Client (Client_Id, Client_Name)
VALUES (101, 'John Doe');
-- This will generate an error: ORA-01400: cannot insert NULL into ("schema"."CLIENT"."IS_CLUB_
ror
ORA-01400: cannot insert NULL into ("SYSTEM"."CLIENT"."IS_CLUB_MEMBER")
OK
Cancel
Help