

## **מיני פרויקט בבסיסי נתונים**

**שם הארגון: חנות**

**שם האגף: נשים**

## **מגישות:**

תהילה בן עזרא 323845321

[tasaraf@g.jct.ac.il](mailto:tasaraf@g.jct.ac.il)

מיכל יששכר 213686496

[Michalir17@gmail.com](mailto:Michalir17@gmail.com)

# מיני פרויקט בבסיסי נתונים

## תוכן עניינים

4	פרק א'
4	תיאור כללי:
4	הישויות במערכת והקשרים ביניהם:
5	ERD Diagram
5	DSD Diagram
6	יצירת הטבלאות ע"י createTable
8	פקודת DESC:
9	DROP_TABLE:
10	קובץ SELECT
11	הכנסת נתונים על ידי שלושת הדרכים המבוקשות:
11	הכנסת נתונים על ידי DATA GENERATOR:
12	הכנסת נתונים על ידי קוד בפיתון ופקודות INSERT:
13	הכנסת נתונים מקובץ:
15	יצירת קובץ INSERT
16	גיבוי
18	מחיקת הפרויקט
19	שחזור הפרויקט
20	פרק ב
20	Select query 1:
21	Select query 2:
22	Select query 3:
23	Select query 4:
24	Delete query 1:
25	Delete query 2:
26	Update query 1
27	Update query 2:
29	Queries with parameters:
30	1.query with data parameter
31	2.query with date parameter
32	3.query with hint parameter
32	4.query with name parameter
32	5.query with list parameter

## מיני פרויקט בבסיסי נתונים

34	..... Constraints
34	..... 1.alter table –CHECK
34	..... 2.alter table-DEFAULT

# 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 נמיינ 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

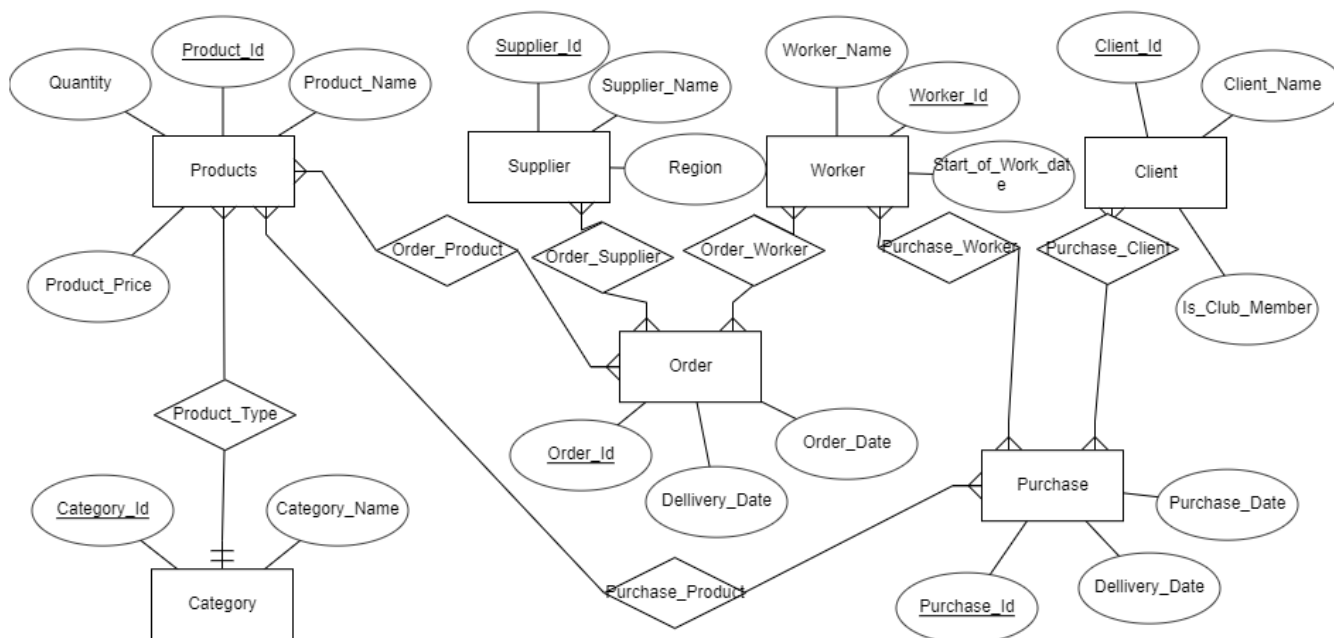
Category = Category\_Id, Category\_Name

Order = Order\_Id, Order\_Date, Delivery\_Date, Quantity

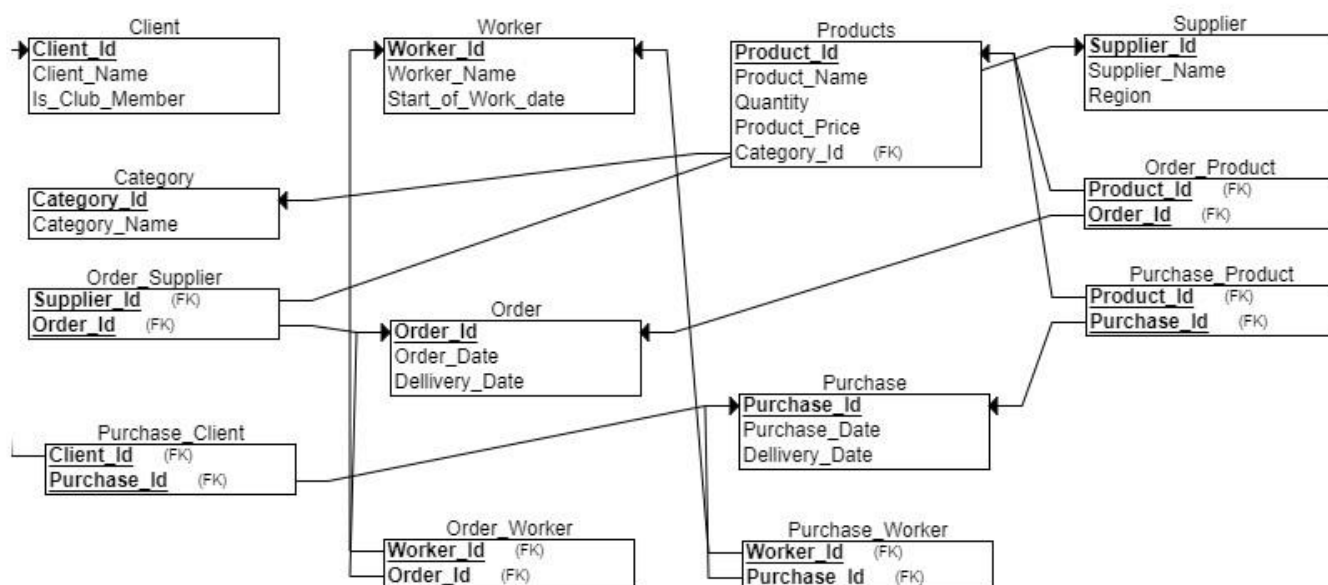
Purchase = Purchase\_Id, Purchase\_Date, Delivery\_Date, Quantity

# מיני פרויקט בבסיסי נתונים

## ERD Diagram

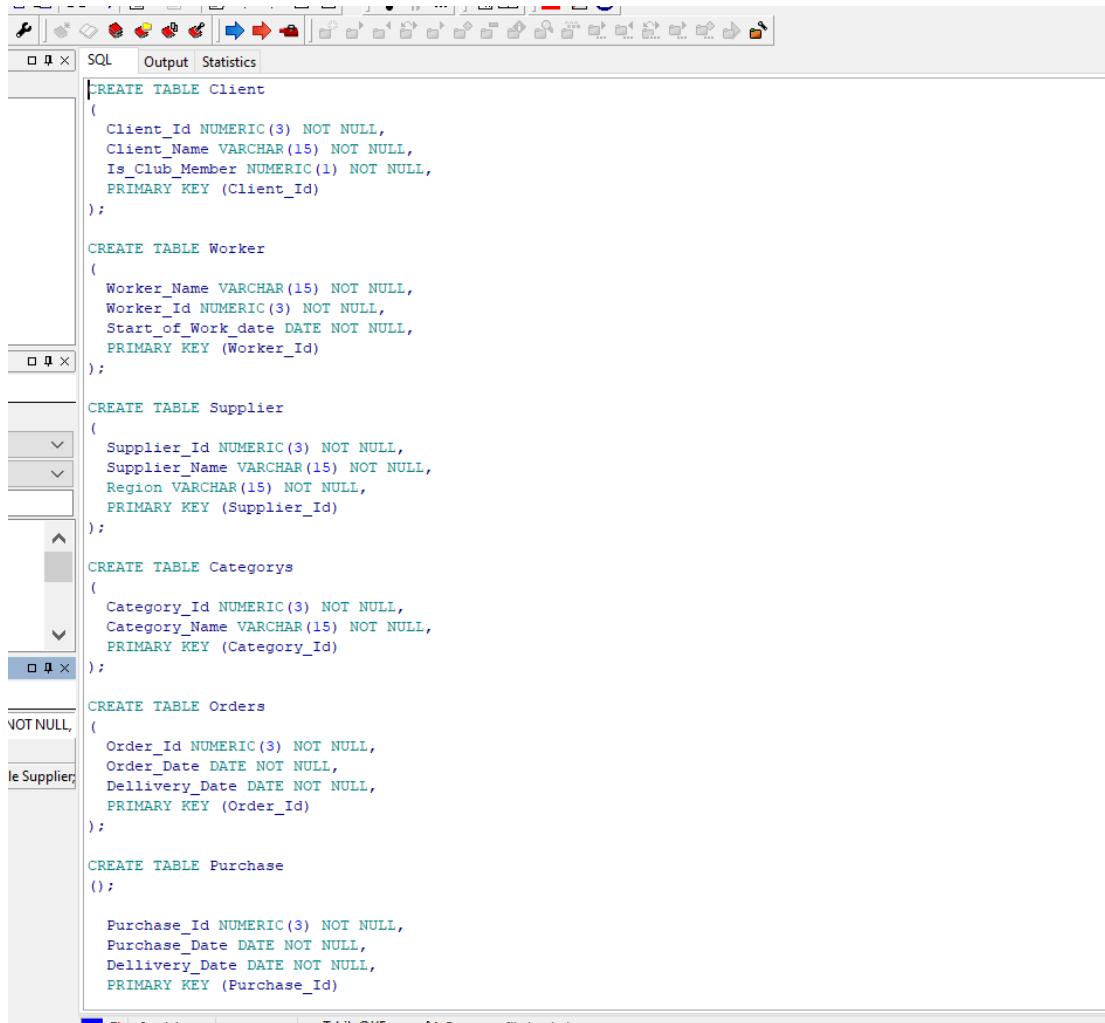


## DSD Diagram



# מיני פרויקט בבסיסי נתונים

## יצירת הטבלאות ע"י createTable



```
CREATE TABLE Client
(
  Client_Id NUMERIC(3) NOT NULL,
  Client_Name VARCHAR(15) NOT NULL,
  Is_Club_Member NUMERIC(1) NOT NULL,
  PRIMARY KEY (Client_Id)
);

CREATE TABLE Worker
(
  Worker_Name VARCHAR(15) NOT NULL,
  Worker_Id NUMERIC(3) NOT NULL,
  Start_of_Work_date DATE NOT NULL,
  PRIMARY KEY (Worker_Id)
);

CREATE TABLE Supplier
(
  Supplier_Id NUMERIC(3) NOT NULL,
  Supplier_Name VARCHAR(15) NOT NULL,
  Region VARCHAR(15) NOT NULL,
  PRIMARY KEY (Supplier_Id)
);

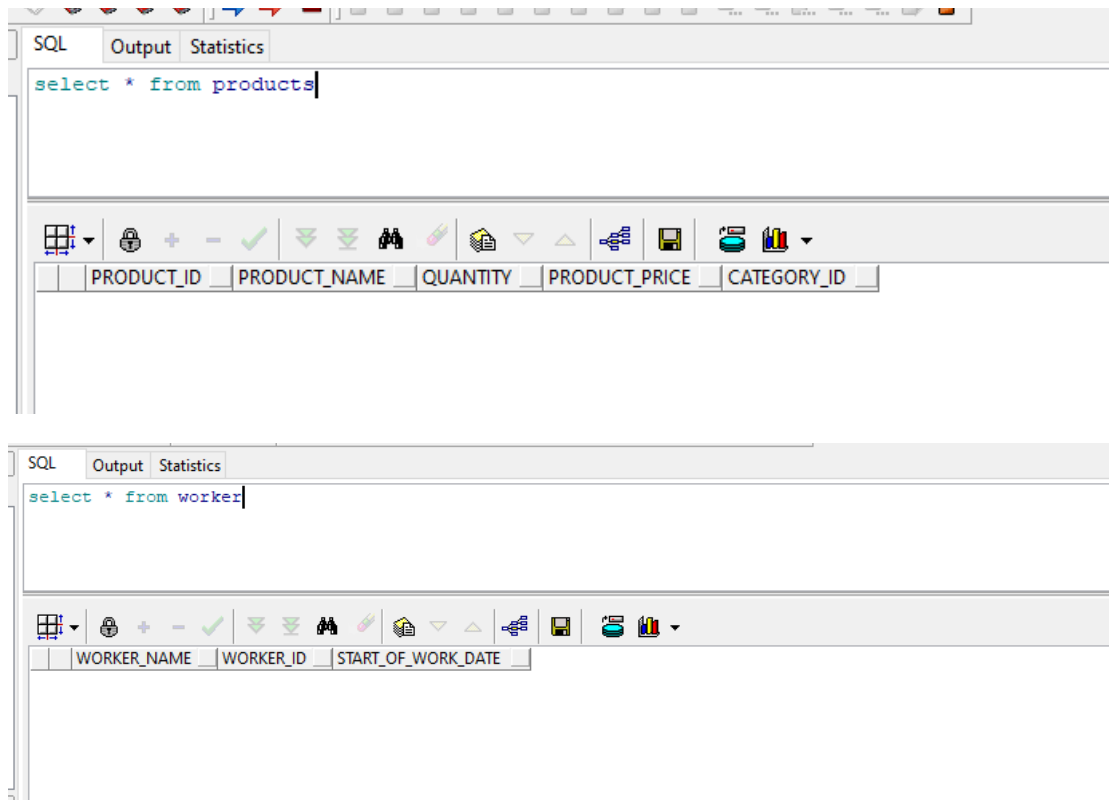
CREATE TABLE Categorys
(
  Category_Id NUMERIC(3) NOT NULL,
  Category_Name VARCHAR(15) NOT NULL,
  PRIMARY KEY (Category_Id)
);

CREATE TABLE Orders
(
  Order_Id NUMERIC(3) NOT NULL,
  Order_Date DATE NOT NULL,
  Dellivery_Date DATE NOT NULL,
  PRIMARY KEY (Order_Id)
);

CREATE TABLE Purchase
(
  Purchase_Id NUMERIC(3) NOT NULL,
  Purchase_Date DATE NOT NULL,
  Dellivery_Date DATE NOT NULL,
  PRIMARY KEY (Purchase_Id)
);
```

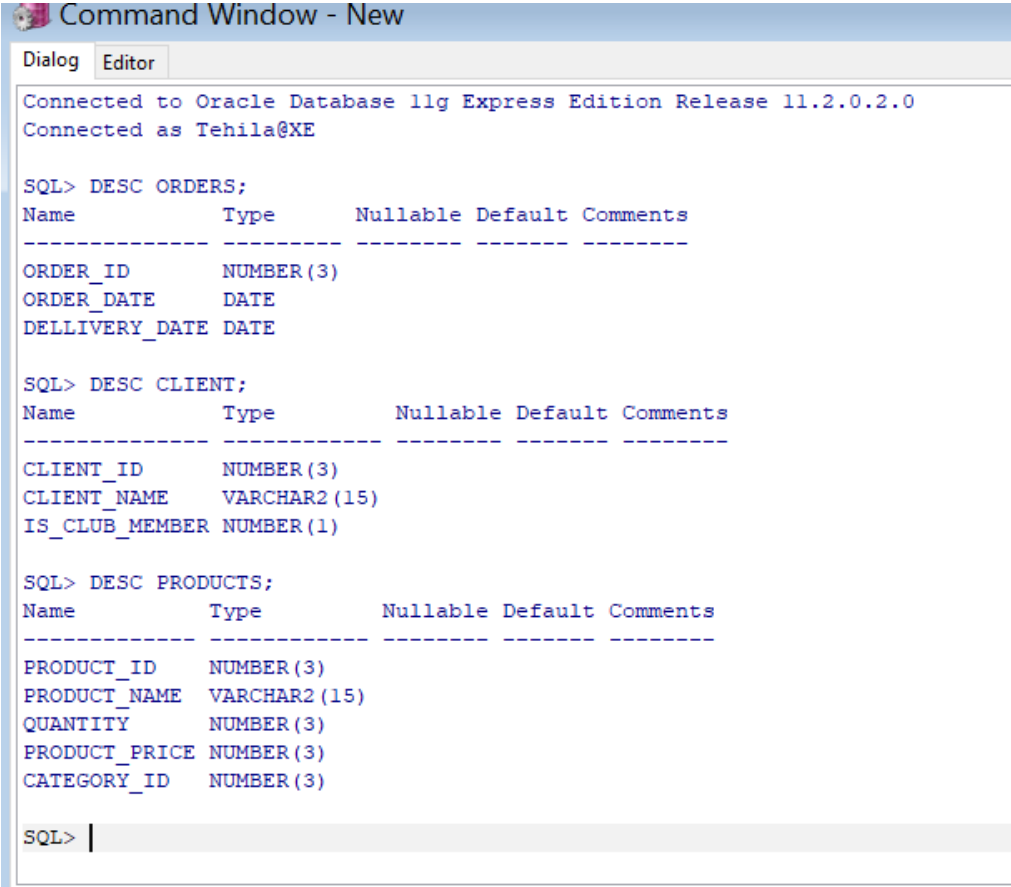
## מיני פרויקט בבסיסי נתונים

בדקנו שאכן הטבלאות נוצרו על ידי הרצה הסקריפט הזה:



# מיני פרויקט בבסיסי נתונים

## פקודת DESC:



```
Command Window - New
Dialog Editor
Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
Connected as Tehila@XE

SQL> DESC ORDERS;
Name          Type          Nullable Default Comments
-----
ORDER_ID      NUMBER(3)
ORDER_DATE    DATE
DELLIVERY_DATE DATE

SQL> DESC CLIENT;
Name          Type          Nullable Default Comments
-----
CLIENT_ID     NUMBER(3)
CLIENT_NAME   VARCHAR2(15)
IS_CLUB_MEMBER NUMBER(1)

SQL> DESC PRODUCTS;
Name          Type          Nullable Default Comments
-----
PRODUCT_ID    NUMBER(3)
PRODUCT_NAME  VARCHAR2(15)
QUANTITY      NUMBER(3)
PRODUCT_PRICE NUMBER(3)
CATEGORY_ID   NUMBER(3)

SQL> |
```



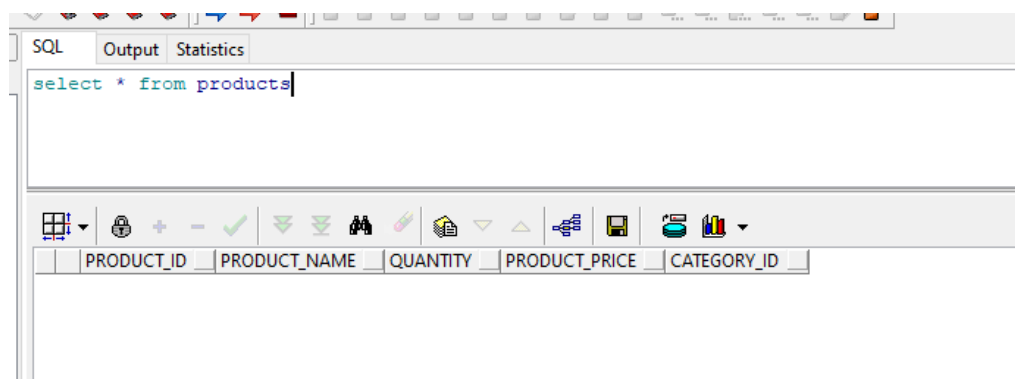
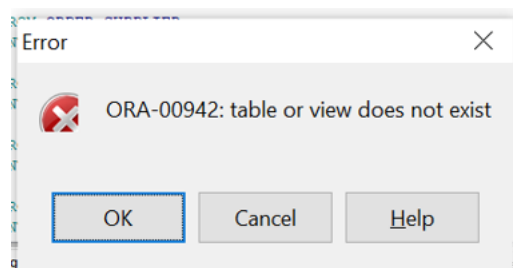
# מיני פרויקט בבסיסי נתונים

## :DROP\_TABLE

כמובן על פי הסדר כך שהטבלאות התלויות בטבלאות אחרות נמחקות קודם

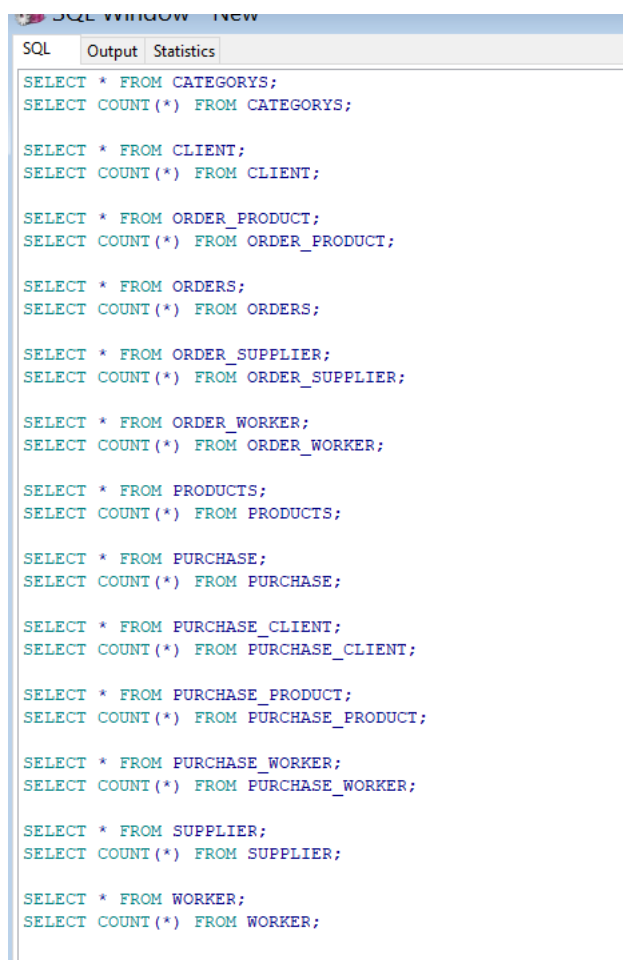
```
SQL      Output  Statistics
drop table Order_Product;
drop table Purchase_Product;
drop table Purchase_Client;
drop table Order_Supplier;
drop table Order_Worker;
drop table Purchase_Worker;
drop table Client;
drop table Worker;
drop table Supplier;
drop table Orders;
drop table Purchase;
drop table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

בדקנו שהטבלאות נמחקות ולאחר מכן יצרנו אותן שוב מחדש



# מיני פרויקט בבסיסי נתונים

## קובץ SELECT



```
SQL Window New
SQL Output Statistics
SELECT * FROM CATEGORIES;
SELECT COUNT(*) FROM CATEGORIES;

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

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

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

SELECT * FROM ORDER_SUPPLIER;
SELECT COUNT(*) FROM ORDER_SUPPLIER;

SELECT * FROM ORDER_WORKER;
SELECT COUNT(*) FROM ORDER_WORKER;

SELECT * FROM PRODUCTS;
SELECT COUNT(*) FROM PRODUCTS;

SELECT * FROM PURCHASE;
SELECT COUNT(*) FROM PURCHASE;

SELECT * FROM PURCHASE_CLIENT;
SELECT COUNT(*) FROM PURCHASE_CLIENT;

SELECT * FROM PURCHASE_PRODUCT;
SELECT COUNT(*) FROM PURCHASE_PRODUCT;

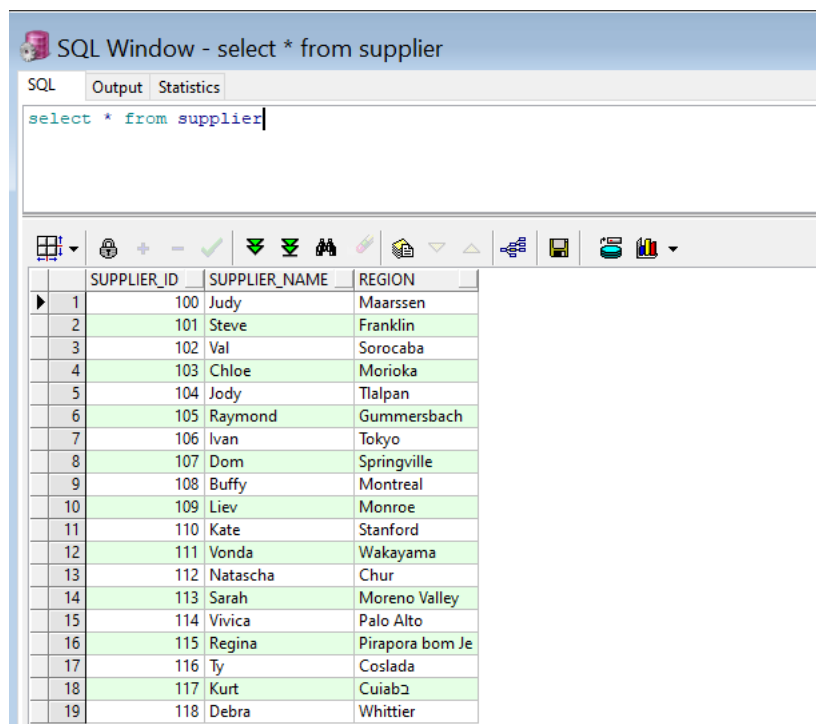
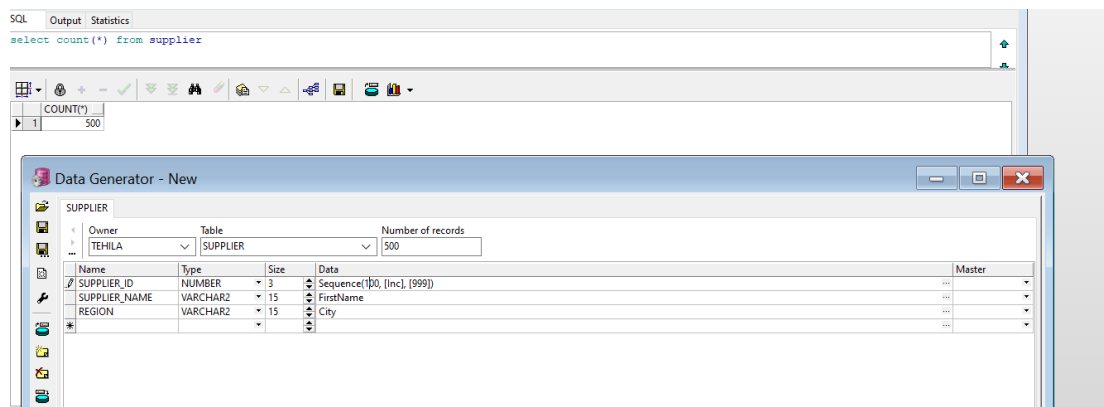
SELECT * FROM PURCHASE_WORKER;
SELECT COUNT(*) FROM PURCHASE_WORKER;

SELECT * FROM SUPPLIER;
SELECT COUNT(*) FROM SUPPLIER;

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

## מיני פרויקט בבסיסי נתונים

הכנסת נתונים על ידי שלושת הדרכים המבוקשות:  
הכנסת נתונים על ידי DATA GENERATOR:



## מיני פרויקט בבסיסי נתונים

### הכנסת נתונים על ידי קוד בפיתון ופקודות INSERT:

```
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_id}'

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

Text Importer - Client.csv

Data from Textfile | Data to Oracle

**General**

Owner: TEHILA | Table: CLIENT | Clear Table

Commit every...: 100 | ☒ Overwrite duplicates | ☐ Ignore duplicates

Initializing Script:  ...

Finalizing Script:  ...

**Fields**

Field1 CLIENT\_ID -> CLIENT\_ID  
Field2 CLIENT\_NAME -> CLIENT\_NAME  
Field3 IS\_CLUB\_MEMBER -> IS\_CLUB\_ME

Field:  | Fieldtype:  | Create SQL

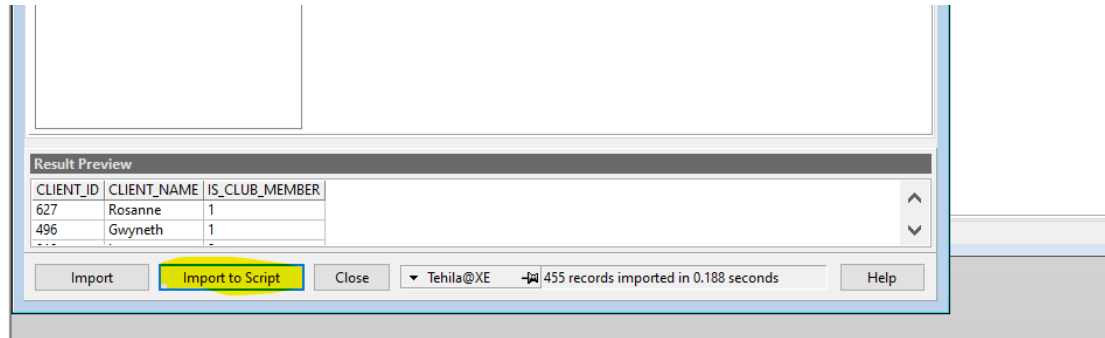
SQL function:  ...  
additional Oracle processing, for example: substr(#, 1, 20)

**Result Preview**

CLIENT_ID	CLIENT_NAME	IS_CLUB_MEMBER
627	Rosanne	1
496	Gwyneth	1

Import | Import to Script | Close | Tehila@XE | Client.csv loaded, 8 KB | Help

## 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 הנדרש

```
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 CATEGORIES (CATEGORY_ID, CATEGORY_NAME)
values (100, 'irure ipsum do. ');

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

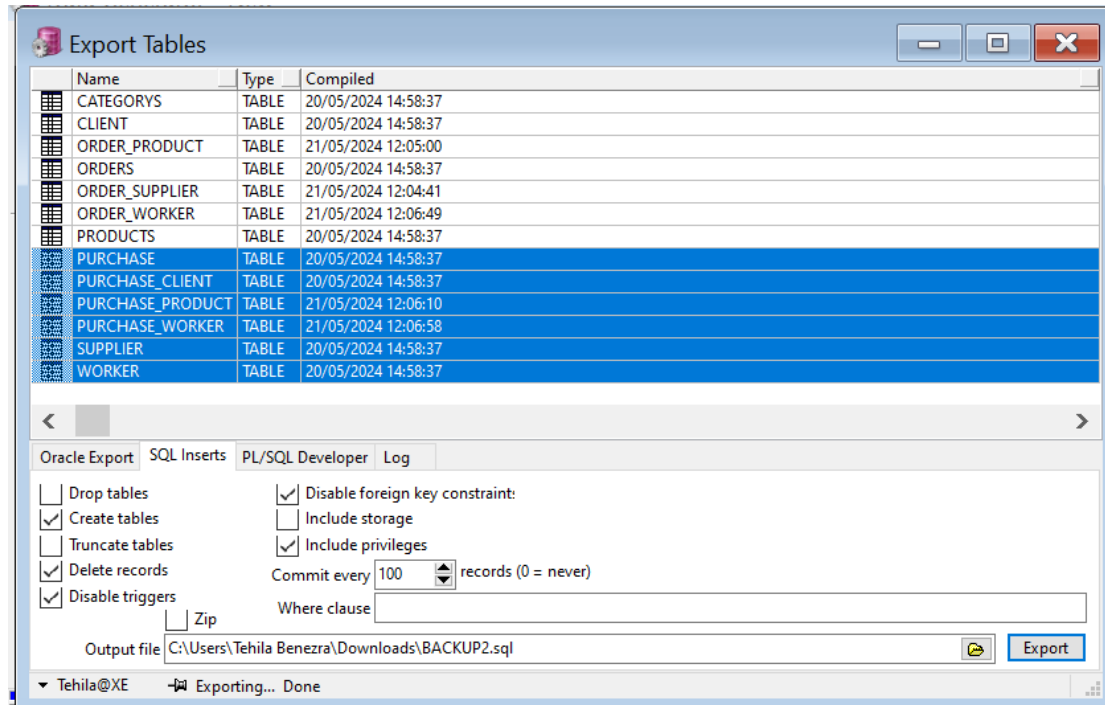
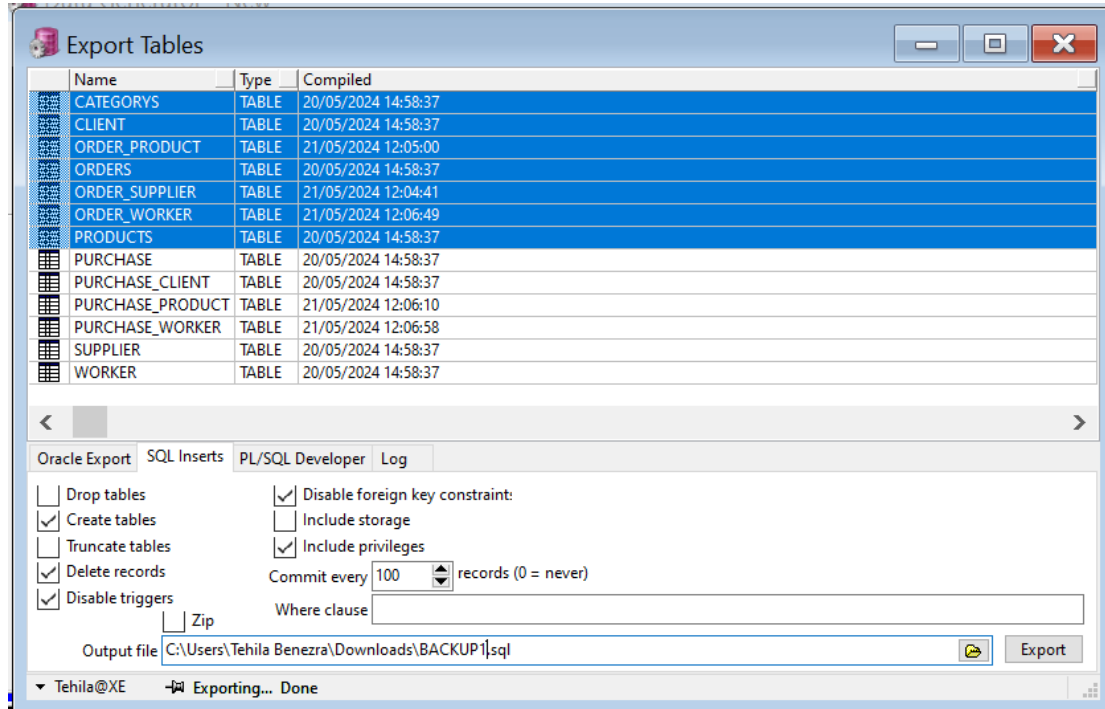
insert into CATEGORIES (CATEGORY_ID, CATEGORY_NAME)
values (102, 'officiis. ');
```

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, DELIVERY_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, DELIVERY_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, DELIVERY_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, DELIVERY_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)





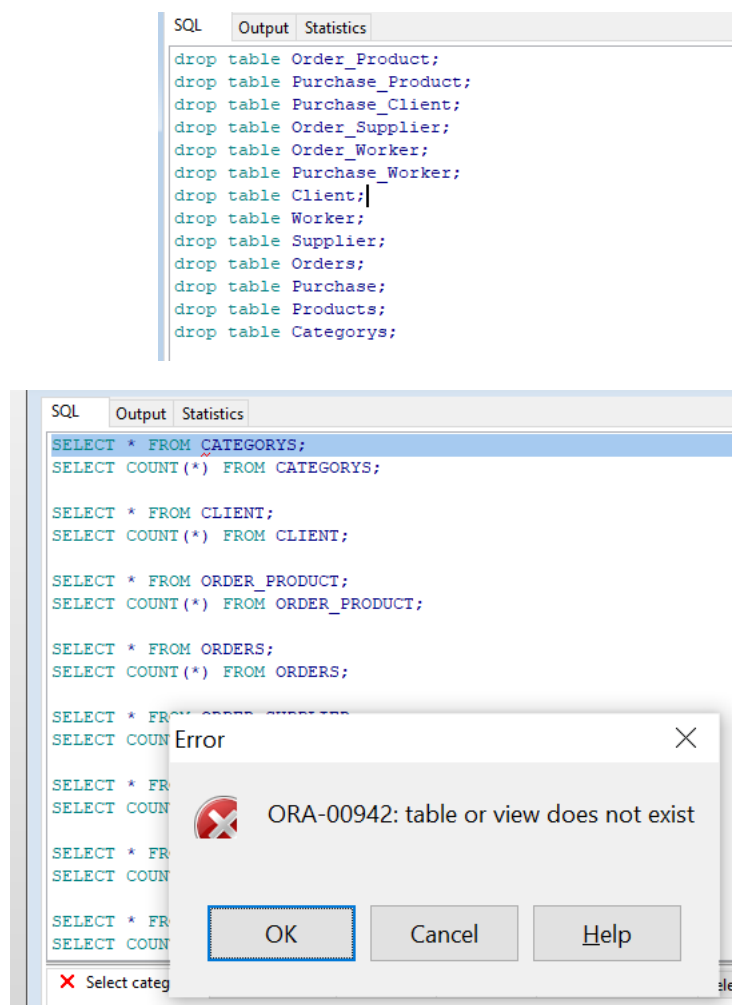
## מיני פרויקט בבסיסי נתונים

```
BACKUP1.sql X
C: > Users > Tehila Benezra > Downloads > BACKUP1.sql
1 | prompt PL/SQL Developer import file
2 | prompt Created on יום 23 מאי 2024 by Tehila Benezra
3 | set feedback off
4 | set define off
5 | prompt Creating CATEGORYS...
6 | create table CATEGORYS
7 | (
8 |   category_id  NUMBER(3) not null,
9 |   category_name VARCHAR2(15) not null
10 | )
11 | ;
12 | alter table CATEGORYS
13 |   add primary key (CATEGORY_ID);
14 |
15 | prompt Creating CLIENT...
16 | create table CLIENT
17 | (
18 |   client_id    NUMBER(3) not null,
19 |   client_name  VARCHAR2(15) not null,
20 |   is_club_member NUMBER(1) not null
21 | )
22 | ;
23 | alter table CLIENT
24 |   add primary key (CLIENT_ID);
25 |
26 | prompt Creating ORDERS...
27 | create table ORDERS
28 | (
29 |   order_id     NUMBER(3) not null,
30 |   order date   DATE not null,
```

```
C: > Users > Tehila Benezra > Downloads > BACKUP2.sql
1 | prompt PL/SQL Developer import file
2 | prompt Created on יום 23 מאי 2024 by Tehila Benezra
3 | set feedback off
4 | set define off
5 | prompt Creating PURCHASE...
6 | create table PURCHASE
7 | (
8 |   purchase_id  NUMBER(3) not null,
9 |   purchase_date DATE not null,
10 |   dellivery_date DATE not null
11 | )
12 | ;
13 | alter table PURCHASE
14 |   add primary key (PURCHASE_ID);
15 |
16 | prompt Creating PURCHASE_CLIENT...
17 | create table PURCHASE_CLIENT
18 | (
19 |   client_id  NUMBER(3) not null,
20 |   purchase_id NUMBER(3) not null
21 | )
22 | ;
23 | alter table PURCHASE_CLIENT
24 |   add primary key (CLIENT_ID, PURCHASE_ID);
25 | alter table PURCHASE_CLIENT
26 |   add foreign key (CLIENT_ID)
27 |     references CLIENT (CLIENT_ID);
28 | alter table PURCHASE_CLIENT
29 |   add foreign key (PURCHASE_ID)
30 |     references PURCHASE (PURCHASE ID);
```

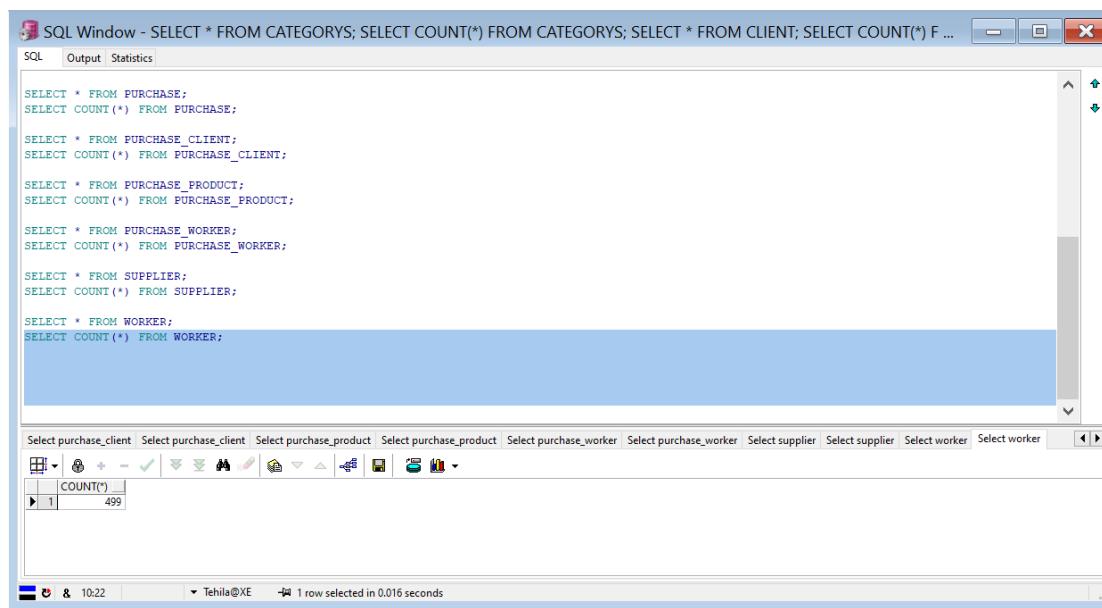
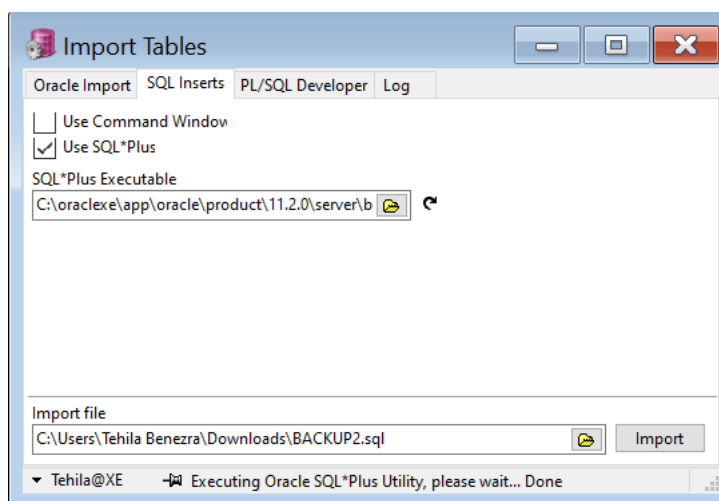
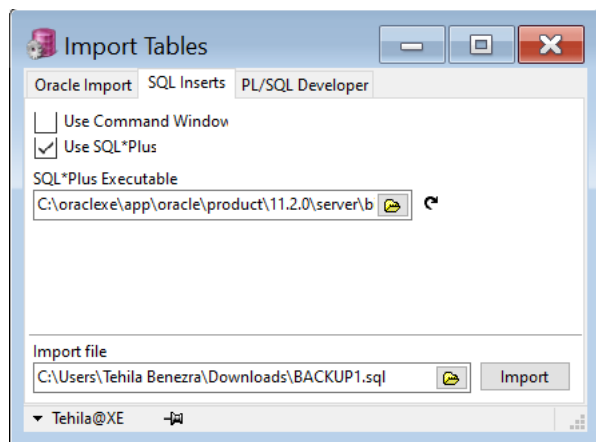
# מיני פרויקט בבסיסי נתונים

## מחיקת הפרויקט



## 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

SQL	Output	Statistics
<pre>SELECT c.Client_Name, COUNT(pc.Purchase_Id) AS Total_Purchases FROM Client c INNER JOIN Purchase_Client pc ON c.Client_Id = pc.Client_Id WHERE c.Is_Club_Member = 1 GROUP BY c.Client_Name;</pre>		
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:

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

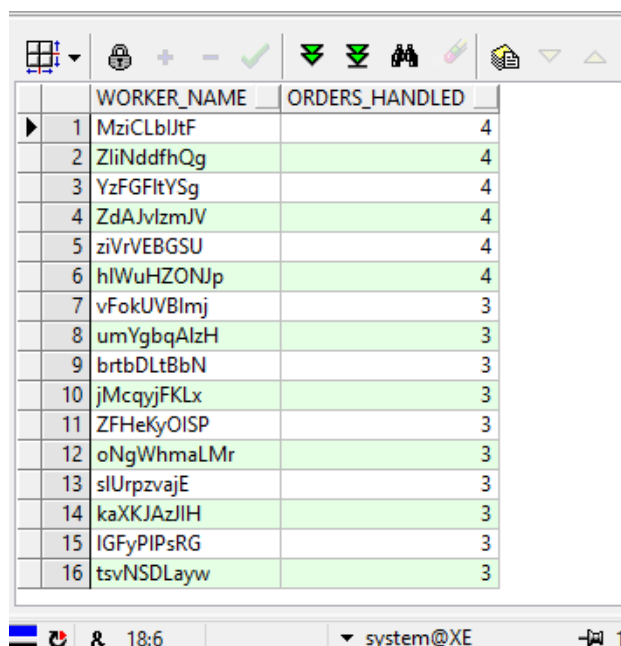
```
SQL Output Statistics
SELECT
  C.Category_Name,
  AVG(PR.Product_Price) AS Average_Price
FROM
  Categorys C
JOIN
  Products PR ON C.Category_Id = PR.Category_Id
GROUP BY
  C.Category_Name
ORDER BY
  Average_Price DESC;
```

	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

### Select query 3:

שאלתה 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	MziCLbJtF	4
2	ZliNddfhQg	4
3	YzFGFitYSg	4
4	ZdAJvlzmJV	4
5	ziVrVEBGsu	4
6	hIWuHZONJp	4
7	vFokUVBlmj	3
8	umYgbqAlzH	3
9	brtbDLtBbN	3
10	jMcqyjFKLx	3
11	ZFHeKyOISP	3
12	oNgWhmaLMr	3
13	slUrpzvajE	3
14	kaXKJAzJIH	3
15	IGFyPIPsRG	3
16	tsvNSDLayw	3

## Select query 4:

השאלתה מחזירה רשימה של עובדים יחד עם ספירת ההזמנות להן  
הוקצו

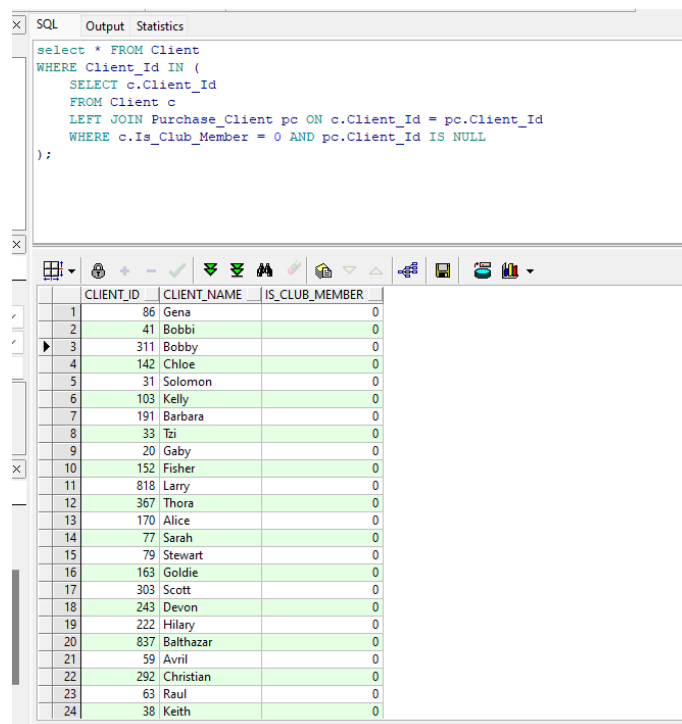
```
SQL Output Statistics
SELECT w.Worker_Name, COUNT(ow.Order_Id) AS Assigned_Order_Count
FROM Worker w
LEFT JOIN Order_Worker ow ON w.Worker_Id = ow.Worker_Id
GROUP BY w.Worker_Name;
```

	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

## Delete query 1:

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

נתון במידע לפני המחיקה:



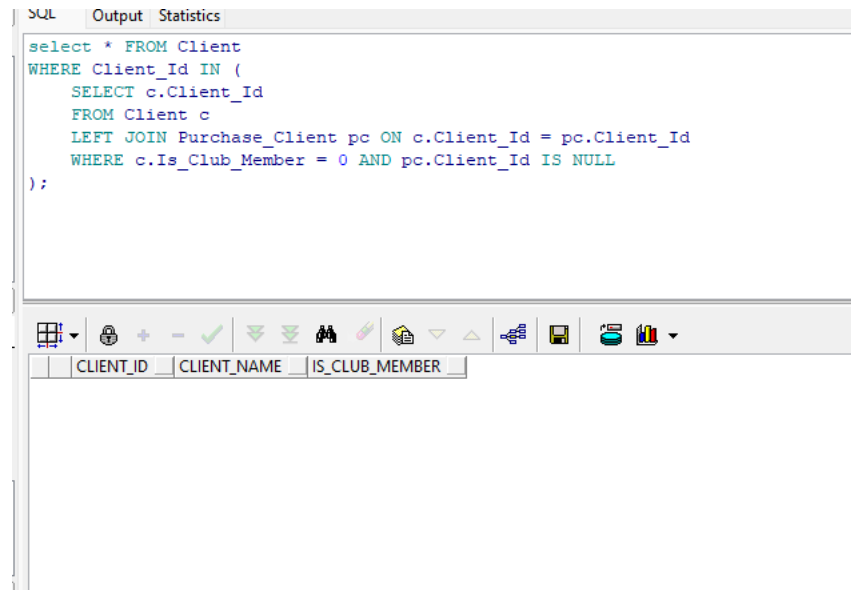
The screenshot shows the SQL Developer interface. The top pane displays a SQL query designed to delete clients who are not club members but have made purchases. The bottom pane shows the result set of the query, which lists 24 clients with their IDs, names, and club membership status (all 0).

```
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 :



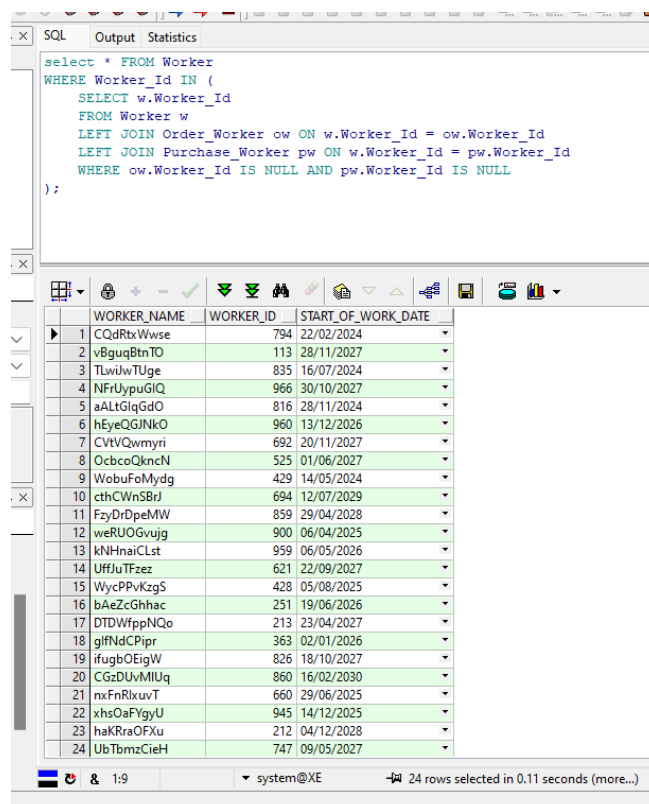


## Delete query 2:

יצרנו שאילתת מחיקה נוספת .

שמוחקת את כל העובדים שמעולם לא הוקצו להזמנות ורכישות

אלא הנתונים לפני המחיקה :



לאחר המחיקה כך נראה בסיס הנתונים שאלינו ניסינו לגשת :

```
SQL Output Statistics
select * FROM Worker
WHERE Worker_Id IN (
  SELECT w.Worker_Id
  FROM Worker w
  LEFT JOIN Order_Worker ow ON w.Worker_Id = ow.Worker_Id
  LEFT JOIN Purchase_Worker pw ON w.Worker_Id = pw.Worker_Id
  WHERE ow.Worker_Id IS NULL AND pw.Worker_Id IS NULL
);

/*delete FROM Worker
WHERE Worker_Id IN (
  SELECT v.Worker_Id
  FROM Worker v
  LEFT JOIN Order_Worker ow ON v.Worker_Id = ow.Worker_Id
  LEFT JOIN Purchase_Worker pw ON v.Worker_Id = pw.Worker_Id
  WHERE ow.Worker_Id IS NULL AND pw.Worker_Id IS NULL
);*/
```

WORKER_NAME	WORKER_ID	START_OF_WORK_DATE
-------------	-----------	--------------------

## Update query 1

שאלתה שמעלה את המחיר מוצר בקטגוריה מסוימת ב10 אחוז  
לפני:

```
SQL Output Statistics
select* from Products
SET Product_Price = Product_Price * 1.10
WHERE Category_Id = (
  SELECT Category_Id
  FROM Categorys
  WHERE Category_Name = 'irure ipsum do.'
);
```

	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:

SQLOutputStatistics

```
--UPDATE Products
select * from Products
--SET Product_Price = Product_Price * 1.10
WHERE Category_Id = (
    SELECT Category_Id
    FROM Categorys
    WHERE Category_Name = 'irure ipsum do.'
);
```

A horizontal toolbar containing various icons for data manipulation and visualization. From left to right, the icons include: a grid, a lock, a plus sign, a minus sign, a checkmark, a double arrow pointing down, a double arrow pointing up, a group of people, a pencil, a house, a downward arrow, an upward arrow, a bar chart, a floppy disk, a database cylinder, and a bar chart with a dropdown arrow.

	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:

שאלת 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

השאלתה

```
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

שאלתה שמחזירה את כל הנתונים של ההזמנה מתאריך מסוים או אחריו

לדוג: כאן קיבלנו את כל ההזמנות שבוצעו מתאריך 12.06.2026

The screenshot displays the SQL Server Enterprise Manager interface. The SQL query editor shows a query that selects order details and client names, filtered by order date on or after 12/06/2026. The query is as follows:

```
SELECT
    O.Order_Id,
    O.Order_Date,
    O.Delivery_Date,
    C.Client_Name
FROM
    Orders O
JOIN
    Purchase_Client PC ON O.Order_Id = PC.Purchase_Id
JOIN
    Client C ON PC.Client_Id = C.Client_Id
WHERE
    O.Order_Date >= TO_DATE('&order_date', 'dd/mm/yyyy');
```

The query results are shown in a table with the following columns: ORDER\_ID, ORDER\_DATE, DELIVERY\_DATE, and CLIENT\_NAME. The results list 18 orders, with the first four rows highlighted in green. A 'Variables' dialog box is open, showing the variable 'order\_date' with the value '12/06/2026'.

	ORDER_ID	ORDER_DATE	DELIVERY_DATE	CLIENT_NAME
1	814	13/10/2030	01/03/2029	Rita
2	611	01/12/2030	18/05/2027	Lila
3	814	13/10/2030	01/03/2029	Sinead
4	611	01/12/2030	18/05/2027	Charles
5	814	13/10/2030	01/03/2029	Shelby
6	611	01/12/2030	18/05/2027	Lin
7	604	10/10/2030	06/01/2029	Mekhi
8	436	29/06/2030	27/11/2030	Miki
9	474	09/09/2030	01/08/2029	Corey
10	474	09/09/2030	01/08/2029	Seth
11	604	10/10/2030	06/01/2029	Stevie
12	101	06/07/2027	03/07/2024	Kurtwood
13	104	19/10/2028	07/11/2026	Juliana
14	105	13/11/2028	24/04/2024	Lili
15	106	13/09/2026	06/04/2026	Philip
16	604	10/10/2030	06/01/2029	Ralph
17	110	23/08/2027	11/11/2030	Rowan
18	111	08/05/2027	20/09/2024	Holland

## 2.query with date parameter

השאלתה 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

The screenshot shows the SQL Developer interface. The main window displays a SQL query titled 'parameter2.sql'. The query selects worker details from a table named 'Worker' based on a date range. A 'Variables' dialog box is open, showing the values for the parameters 'start\_date' and 'end\_date'.

**SQL Window - parameter2.sql**

```
SELECT
  W.Worker_Id,
  W.Worker_Name,
  W.Start_of_Work_date
FROM
  Worker W
WHERE
  W.Start_of_Work_date >= TO_DATE('&start_date', 'dd/mm/yyyy')
  AND W.Start_of_Work_date <= TO_DATE('&end_date', 'dd/mm/yyyy');
```

**Variables Dialog**

Name	Value
start_date	04/05/2028
end_date	04/05/2029

**Query Results**

WORKER_ID	WORKER_NAME	START_OF_WORK_DATE
1	131 VVZPsTnsdu	09/02/2029
2	807 DfaqtdNerj	06/08/2028
3	105 QmWenKDtjy	29/06/2028
4	744 LEolmByKar	24/10/2028
5	209 qBDWviOnPd	19/06/2028
6	666 sGjuiFfkW	07/05/2028
7	518 jnguVFOEtX	13/10/2028
8	188 igFPQwpXZJ	03/05/2029
9	514 vscBzCjHh	20/10/2028
10	761 IGFyPIPsRG	07/07/2028
11	340 VlmXfövlkT	08/12/2028

### 3.query with hint parameter

The screenshot shows the SQL Window for 'parameter3.sql'. The SQL query is:

```
SELECT
  C.client_id,
  C.client_name,
  C.is_club_member
FROM
  client C
WHERE
  C.Client_Id=&<name="clientID" hint="client id value between 0-999">;
```

The Variables dialog box is open, showing the variable 'clientID' with the value '627'. The OK button is highlighted. Below the dialog, the text 'client id value between 0-999' is visible.

The result set shows one row:

	CLIENT_ID	CLIENT_NAME	IS_CLUB_MEMBER
1	627	Rosanne	1

### 4.query with name parameter

The screenshot shows the SQL Window for 'parameter4.sql'. The SQL query is:

```
SELECT
  O.Order_Id,
  O.Order_Date,
  O.Delivery_Date
FROM
  Orders O
JOIN
  Purchase_Client PC ON O.Order_Id = PC.Purchase_Id
JOIN
  Client C ON PC.Client_Id = C.Client_Id
WHERE
  C.Client_Name = '&client_name';
```

The Variables dialog box is open, showing the variable 'client\_name' with the value 'Joey'. The OK button is highlighted.

The result set shows three rows:

	ORDER_ID	ORDER_DATE	DELIVERY_DATE
1	206	22/09/2029	07/10/2029
2	106	13/09/2026	06/04/2026
3	133	09/01/2028	13/06/2025

### 5.query with list parameter



The screenshot shows an SQL IDE window titled "SQL Window - parameter5.sql". The main window has tabs for "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying a query. A "Variables" dialog box is open in the foreground, showing a table with two columns: "Name" and "Value". The table contains one row: "category\_name" with the value "iusto". The "OK" button is highlighted. Below the dialog, the "Output" tab is active, displaying a table with the following data:

	CATEGORY_NAME	CATEGORY_ID	PRODUCT_ID	PRODUCT_NAME
1	iusto	130	130	Procter & Gambl
2	iusto	130	230	Best Buy Co.
3	iusto	130	330	GCI
4	iusto	130	430	Client Network
5	iusto	130	530	Newton Interact
6	iusto	130	630	Viacom

# Constraints

## 1.alter table –CHECK

```
SQL Output Statistics
-- Adding the CHECK constraint
ALTER TABLE Products
ADD CONSTRAINT CHK_Quantity2 CHECK (Quantity >= 0);

-- Attempting to insert a product with Quantity = -1 (this should fail)
INSERT INTO Products (Product_Id, Product_Name, Quantity, Product_Price, Category_Id)
VALUES (101, 'Test Product', -1, 10, 1);
```

## 2.alter table-DEFAULT

```
SQL Output Statistics
-- Adding the DEFAULT constraint
ALTER TABLE Supplier
MODIFY Region VARCHAR(15) DEFAULT 'Unknown';

-- Inserting a supplier without specifying the Region (Region should default to 'Unknown')
INSERT INTO Supplier (Supplier_Id, Supplier_Name)
VALUES (226, 'ABC Suppliers');

-- Verifying the insertion
SELECT * FROM Supplier WHERE Supplier_Id = 226;
```

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

