

בג"ד

מועדה

נעמה אילוז 212259204

ידידה כהן 322235250

מבוא

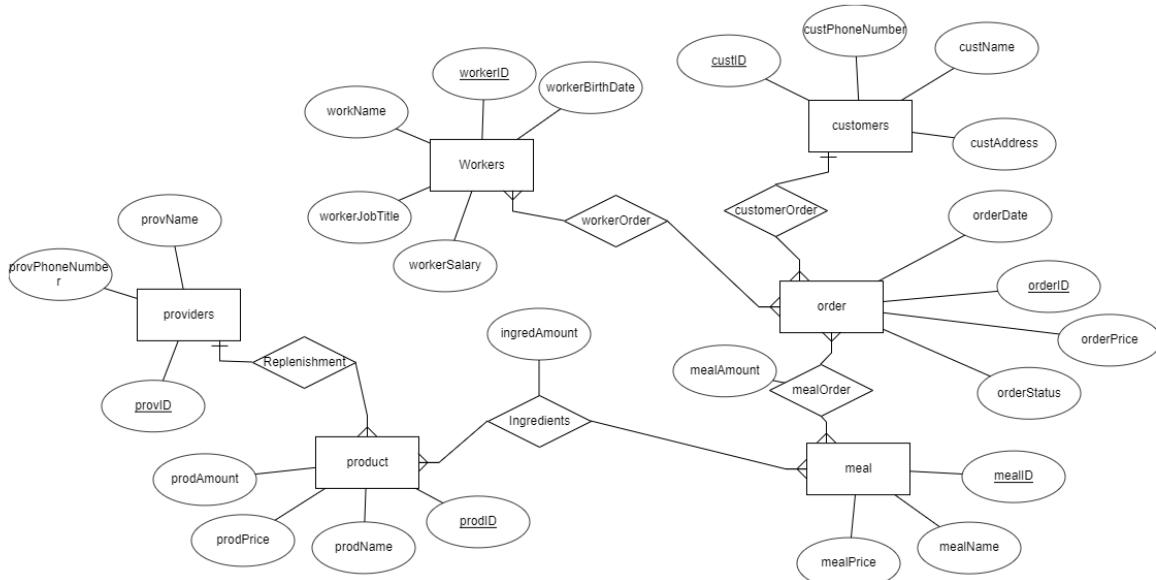
ניהול בית עסק היא אינה משימה קלה בפרט כאשר מדובר במסעדת, כמוות האתגרים איתם מתמודד בעל מסעדת והידע והניסיונו הנדרשים ממנו הם רבים יותר מכל בית עסק אחר, אחד הכלים היכולים להקל מעט בניהול המסעדת היא תוכנה ייעודית המותאמת לאופי הפעולות של המסעדת.

שירותי המערכת

המערכת מאפשר הזמנות מצד הלוקוחות, חידוש המלאי מהספקים, הכנת הזמנה מצד העובדים ומשלוח נריצה להיות מסוגלים לעשوت שאלות על כמות המוצרים במלאי, סכום הזמנה, מיקומה, משכורות העובדים, הכנסות והוצאות.

- ניהול ספקים – המערכת תשמור עבור כל ספק את המוצרים אותו הוא מספק, וכן פרטיים על הספק.
- חומר גלם – המערכת תשמר מידע על חומר הגלם, כמותם ומחירם.
- ארכחות – המערכת תשמר מידע על הארכחות ומחיירם.
- לוקוחות – המערכת תשמר את פרטי הלוקוח כמו שם, טלפון וכתובות.
- עובדים – המערכת תשמר פרטיים כמו משרה, משכורת שם ות.ז.

תרשים ERD



תרשים DSD

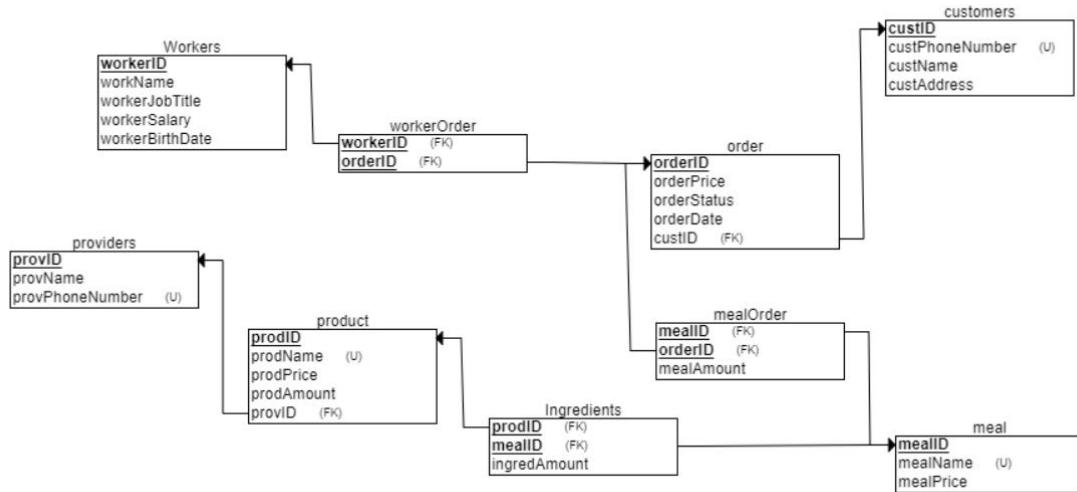
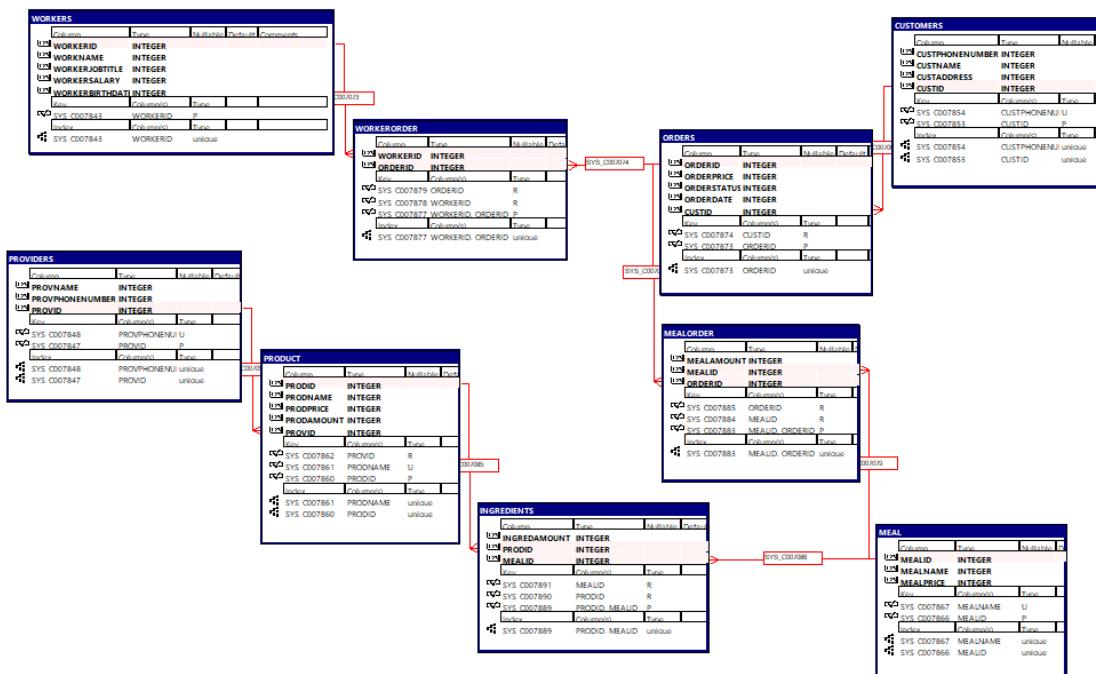


Diagram Window



CREATE TABLES נטיר

```
SQL    Output  Statistics
CREATE TABLE Workers
(
    workerID INT NOT NULL,
    workName VARCHAR2(20) NOT NULL,
    workerJobTitle VARCHAR2(20) NOT NULL,
    workerSalary INT NOT NULL,
    workerBirthDate DATE NOT NULL,
    PRIMARY KEY (workerID)
);

CREATE TABLE Providers
(
    provName VARCHAR2(20) NOT NULL,
    provPhoneNumber INT NOT NULL,
    provID INT NOT NULL,
    PRIMARY KEY (provID)
);

CREATE TABLE Customers
(
    custPhoneNumber INT NOT NULL,
    custName VARCHAR2(20) NOT NULL,
    custAddress VARCHAR2(20) NOT NULL,
    custID INT NOT NULL,
    PRIMARY KEY (custID)
);

CREATE TABLE Product
(
    prodID INT NOT NULL,
    prodName VARCHAR2(20) NOT NULL,
    prodPrice INT NOT NULL,
    prodAmount INT NOT NULL,
    provID INT NOT NULL,
    PRIMARY KEY (prodID),
    FOREIGN KEY (provID) REFERENCES Providers(provID)
);

CREATE TABLE Meal
(
    mealID INT NOT NULL,
    mealName VARCHAR2(20) NOT NULL,
    mealPrice INT NOT NULL,
    PRIMARY KEY (mealID)
);
```

```
CREATE TABLE Orders
(
    orderID INT NOT NULL,
    orderPrice INT NOT NULL,
    orderStatus VARCHAR2(20) NOT NULL,
    orderDate DATE NOT NULL,
    custID INT NOT NULL,
    PRIMARY KEY (orderID),
    FOREIGN KEY (custID) REFERENCES Customers(custID)
);

CREATE TABLE WorkerOrder
(
    workerID INT NOT NULL,
    orderID INT NOT NULL,
    PRIMARY KEY (workerID, orderID),
    FOREIGN KEY (workerID) REFERENCES Workers(workerID),
    FOREIGN KEY (orderID) REFERENCES Orders(orderID)
);

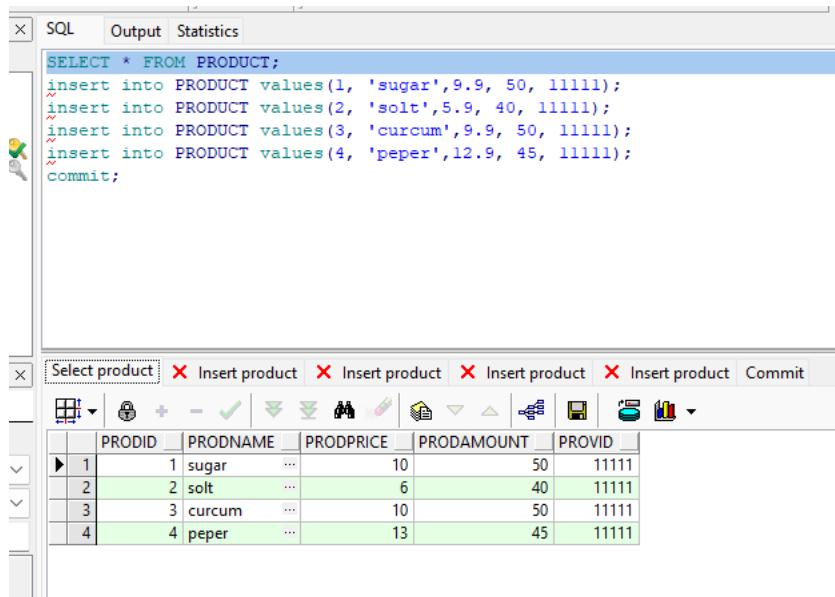
CREATE TABLE MealOrder
(
    mealAmount INT NOT NULL,
    mealID INT NOT NULL,
    orderID INT NOT NULL,
    PRIMARY KEY (mealID, orderID),
    FOREIGN KEY (mealID) REFERENCES Meal(mealID),
    FOREIGN KEY (orderID) REFERENCES Orders(orderID)
);

CREATE TABLE Ingredients
(
    ingredAmount INT NOT NULL,
    prodID INT NOT NULL,
    mealID INT NOT NULL,
    PRIMARY KEY (prodID, mealID),
    FOREIGN KEY (prodID) REFERENCES Product(prodID),
    FOREIGN KEY (mealID) REFERENCES Meal(mealID)
);
```

פקודת INSERT TABLE

פקודה זו מכניסה נתונים לatableות, בהתאם לסוג הנוצר בכל טבלה.

PRODUCT



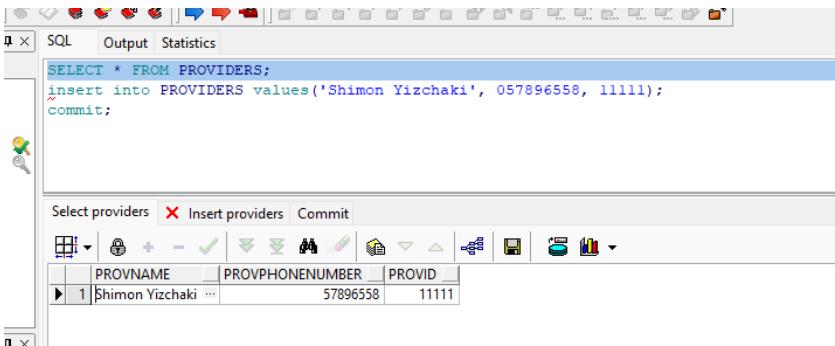
```

SELECT * FROM PRODUCT;
insert into PRODUCT values(1, 'sugar',9.9, 50, 11111);
insert into PRODUCT values(2, 'solt',5.9, 40, 11111);
insert into PRODUCT values(3, 'curcum',9.9, 50, 11111);
insert into PRODUCT values(4, 'peper',12.9, 45, 11111);
commit;

```

PRODID	PRODNAME	PRODPRI	PRODAMOUNT	PROVID
1	sugar	10	50	11111
2	solt	6	40	11111
3	curcum	10	50	11111
4	peper	13	45	11111

PROVIDERS



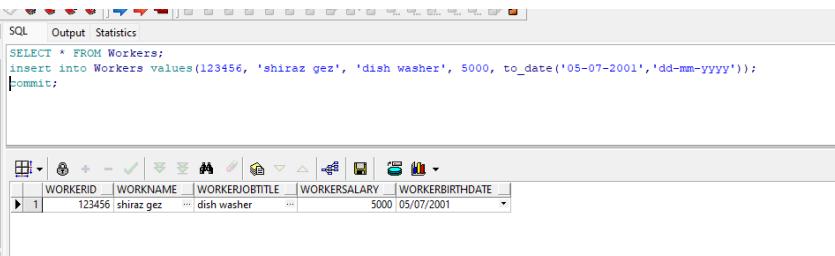
```

SELECT * FROM PROVIDERS;
insert into PROVIDERS values('Shimon Yizchaki', 057896558, 11111);
commit;

```

PROVNAME	PROVPHONE	PROVID
Shimon Yizchaki	57896558	11111

WOKERS



```

SELECT * FROM Workers;
insert into Workers values(123456, 'shiraz gez', 'dish washer', 5000, to_date('05-07-2001','dd-mm-yyyy'));
commit;

```

WORKERID	WORKNAME	WORKERJOBTITLE	WORKERSALARY	WORKERBIRTHDATE
123456	shiraz gez	dish washer	5000	05/07/2001

Customers

Customers

```

SQL Output Statistics
SELECT * FROM Customers;
insert into Customers values(04376448, 'ayala gez', 'Tel Aviv', 3143315);
insert into Customers values(02814297, 'galit malkiel', 'Elad', 4554662 );
insert into Customers values(0942563, 'shiran jib', 'Elad', 289785144 );
insert into Customers values(037284, 'shirav levi', 'Netanya', 5267256 );
commit;

```

	CUSTPHONE NUMBER	CUSTNAME	CUSTADDRESS	CUSTID
1	4376448	ayala gez	Tel Aviv	3143315
2	2814297	galit malkiel	Elad	4554662
3	942563	shiran jib	Elad	289785144
4	37284	shirav levi	Netanya	5267256

Meal

MEAL

```

SQL Output Statistics
SELECT * FROM MEAL;
insert into MEAL values(1, 'shawarma', 50);
insert into MEAL values(2, 'chips', 15);
insert into MEAL values(3, 'pargit', 45);
insert into MEAL values(4, 'shnizl', 45);
commit;

```

	MEALID	MEALNAME	MEALPRICE
1	1	shawarma	50
2	2	chips	15
3	3	pargit	45
4	4	shnizl	45

ORDERS

ORDERS

```

SQL Output Statistics
SELECT * FROM ORDERS;
Insert into ORDERS values(1, 103,'Deliversd', to_date('26-05-2024','dd-mm-yyyy'), 3143315);
insert into ORDERS values(2, 55.9,'Sent', to_date('25-05-2024','dd-mm-yyyy'), 4554662);
insert into ORDERS values(3, 89,'Payed', to_date('26-05-2024','dd-mm-yyyy'), 289785144);
insert into ORDERS values(4, 25,'Deliversd', to_date('26-05-2024','dd-mm-yyyy'), 5267256);
commit;

```

	ORDERID	ORDERPRICE	ORDERSTATUS	ORDERDATE	CUSTID
1	1	103	Deliversd	26/05/2024	3143315
2	2	56	Sent	25/05/2024	4554662
3	3	89	Payed	26/05/2024	289785144
4	4	25	Deliversd	26/05/2024	5267256

WORKERORDER

SQL Output Statistics

```
SELECT * FROM WORKERORDER;
insert into WORKERORDER values(123456, 1);
insert into WORKERORDER values(123456, 2);
insert into WORKERORDER values(123456, 3);
insert into WORKERORDER values(123456, 4);
commit;
```

	WORKERID	ORDERID
▶	1	123456
	2	123456
	3	123456
	4	123456

MEALAMOUNT

SQL Output Statistics

```
SELECT * FROM MEALORDER;
insert into MEALORDER values(2, 1, 1);
insert into MEALORDER values(1, 3, 1);
insert into MEALORDER values(3, 1, 2);
insert into MEALORDER values(5, 4, 3);
insert into MEALORDER values(1, 2, 3);
commit;
```

	MEALAMOUNT	MEALID	ORDERID
▶	1	2	1
	2	3	1
	3	5	4
	4	1	2
	5	1	3

INGREDIENT

SQL Output Statistics

```

SELECT * FROM INGREDIENTS;
insert into INGREDIENTS values(2, 1, 1);
insert into INGREDIENTS values(1, 2, 1);
insert into INGREDIENTS values(3, 3, 2);
insert into INGREDIENTS values(5, 4, 3);
insert into INGREDIENTS values(1, 2, 4);
commit;

```

	INGREDAMOUNT	PRODID	MEALID
▶	1	2	1
	2	3	2
	3	5	3
	4	1	2
	5	1	1

DATA GENERATOR נידרף

CUSTOMERS

CUSTPHONENUMBER CUSTNAME CUSTADDRESS CUSTID

1 066-3490189	... Parker-Tisdale	... 81 Dustin Road	... 522940481
2 098-8871935	... Buddy-Whitmore	... 75 Dom Street	... 241870241
3 029-5330814	... Robert-Wilkinson	... 22 Golden Ave	... 517234236
4 033-8058781	... Elisabeth-Pleasence	... 67 Brno Road	... 333242291
5 038-6311550	... Willem-Weston	... 98 Lyonne Drive	... 931068657
6 059-1411969	... Adina-Schock	... 52 Karlstad Ave	... 485770015
7 029-3638923	... Armin-Minogue	... 24 Fisher Drive	... 605005533
8 017-8155631	... Tracy-Wilson	... 224 Hudson	... 146196348
9 059-5516331	... William-Mirren	... 328 Morrison Drive	... 233491041
10 043-7924042	... Clive-Streep	... 99 Stills	... 448975366
11 011-2335739	... Carrie-Northam	... 26 Savage Ave	... 585495930
12 071-3355694	... Mary Beth-Malkovich	... 777 Bracknell Drive	... 163377240
13 059-6863938	... Kirk-Nakai	... 91 Dorn Street	... 723324367
14 086-7329939	... Laura-Hurley	... 804 Crow Ave	... 175533054
15 080-2578367	... Ali-Robbins	... 9 Keen Ave	... 998742670
16 024-2680232	... Samuel-Secada	... 53rd Street	... 958232104
17 032-3056923	... Vince-Carilton	... 92nd Street	... 671094388
18 071-1165339	... Larenz-Roy Parnell	... 62nd Street	... 489370868
19 062-5150026	... Samantha-Colman	... 84 Whitford Drive	... 142670763
20 057-7019986	... Rachel-Mraz	... 62nd Street	... 297939538
21 039-2707140	... Ruth-Midler	... 486 Chely Ave	... 719631207
22 060-9299876	... Celia-Bandy	... 62nd Street	... 445829789
23 077-9105557	... Gene-Tah	... 279 Curfman Drive	... 808696837
24 082-4412408	... Carlos-Creek	... 90 Cesena Drive	... 722292162
25 097-4676724	... Ernie-Peebles	... 50 Rozenburg Ave	... 236078123
26 099-1298110	... Stewart-Hirsch	... 75 Glenn Road	... 400797963
27 074-2135523	... Sam-Azaria	... 17 Vai Drive	... 565851808
28 023-7696873	... Belinda-Rockwell	... 58 Glenn Blvd	... 664047979
29 057-8834690	... Ani-Wopat	... 8 Ray Road	... 840036663
30 075-8154445	... Sally-Rhymes	... 67 Happy Valley Str	... 418093205
31 051-4150035	... Arnold-Tanor	... 61 Maury	... 296796858
32 095-2352474	... Sylvester-Lapointe	... 94 Jean	... 424695939
33 059-3565607	... Samuel-Weaving	... 752 Ossie Road	... 958755725

CUSTOMERS

Owner	Table	Number of records
NAAMA	CUSTOMERS	1

Name	Type	Size	Data
CUSTPHONENUMBER	VARCHAR2	20	0' + Random(11, 99) ' - Random(1111111, 9999999)
CUSTNAME	VARCHAR2	20	FirstName - LastName
CUSTADDRESS	VARCHAR2	20	Address1
CUSTID	NUMBER		Random(11111111, 99999999)
*			

WORKERS

WORKERS

	WORKERID	WORKERNAME	WORKERJOBTITLE	WORKERSALARY	WORKERBIRTHDATE
▶	1	583440085 Nikka-Scaggs	Dishwasher	7886	20/07/1999
	2	961086423 Andrae-Kweller	Delivery-guy	9650	31/12/1991
	3	303427211 Maury-Paltrow	Director	6690	05/12/1966
	4	997916800 Jessica-Scheider	Director	9748	07/10/1970
	5	268935709 Terry-Landau	Dishwasher	7871	19/06/1980
	6	926632629 Marina-McGregor	Cleaner	9439	24/04/1982
	7	700042433 Alec-Furay	Sous-chef	7530	28/10/1994
	8	424818453 Irene-Simpson	Dishwasher	5119	24/03/1956
	9	439954182 Loren-McNeice	Sous-chef	8706	25/05/1966
	10	631270973 Gilbert-Aglukark	Delivery-guy	8776	07/05/1960
	11	764505236 Vanessa-Vicious	Sous-chef	9686	17/09/1972
	12	343318925 Kyle-Eckhart	Chef	5435	06/08/1963
	13	395147369 Selma-Travers	Chef	9292	19/06/1993
	14	916399906 Laura-Holland	Delivery-guy	9493	30/08/1983
	15	608030150 Aidan-Wen	Director	8248	25/01/1977
	16	935338584 Patricia-Borden	Sous-chef	7172	04/06/1962
	17	289093734 Bernard-Holden	Director	7052	10/08/1990
	18	171215188 Roy-Coughlan	Sous-chef	8644	26/11/1957
	19	145850293 Roddy-Colon	Chef	9619	21/01/1969
	20	385172141 Vendetta-Herndon	Cleaner	7924	27/12/1950
	21	578214853 Larenz-Beckinsale	Sous-chef	7586	21/10/1955
	22	484508657 Lisa-Richardson	Dishwasher	5621	30/07/1971
	23	840618001 Rupert-Dysart	Dishwasher	9416	19/07/1970
	24	211257982 Victor-Kretschmann	Chef	8085	25/06/1988
	25	684649220 Chad-Russo	Chef	8519	01/02/1964
	26	816909755 Ian-McClinton	Director	5399	13/12/1975
	27	307363267 Jean-Claude-Wopat	Dishwasher	6771	10/01/1966
	28	658344377 Jet-Witt	Delivery-guy	9181	12/05/1964
	29	240616467 Hank-Simpson	Sous-chef	9104	16/02/1967
	30	582218859 Aaron-Cassidy	Chef	8307	22/09/1958
	31	268872702 Laurence-MacLachlan	Director	5159	08/11/1973
	32	754665173 Ernie-von Sydow	Cleaner	9575	08/01/1965
	33	688155235 Sonny-Cobbs	Chef	6548	08/01/1984

NAAMA@XE 1:22 400 rows selected in 0.234 seconds

PROVIDERS

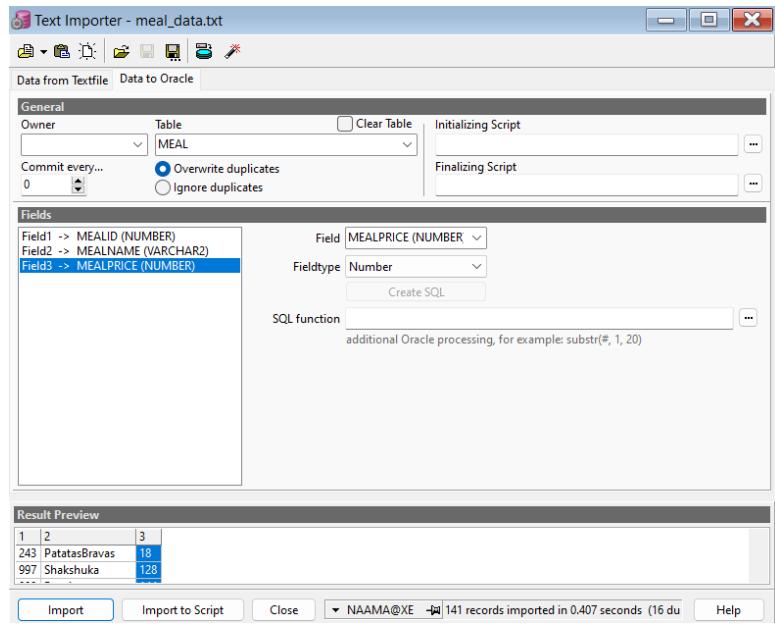
PROVIDERS

	PROVNAME	PROVPHONENUMBER	PROVID
▶	1 Jesus-LaSalle	032-9261177	67629442
	2 Avril-Levert	042-6815247	135836379
	3 Dick-Phoenix	077-5649844	439647872
	4 Diane-Tempest	079-5095358	965300293
	5 Olga-Coolidge	039-3964462	625197018
	6 Hugh-Haynes	018-8447548	752709756
	7 Hank-Krieger	089-3257255	633048403
	8 Ewan-Ponce	043-3568967	612887658
	9 Shawn-Meyer	044-2394165	174032265
	10 Vertical-Ricci	029-2405329	134116135
	11 Trini-Greenwood	077-8745089	772804873
	12 Chi-Barnett	030-3983449	112109750
	13 Kieran-Lowe	070-8224619	460094011
	14 Tyrone-Kinski	090-4487174	197359450
	15 Seth-Allen	084-6677859	540261844
	16 Isabella-Gyllenhaal	058-9733636	423186695
	17 Neil-Rickles	075-6260422	147618326
	18 Rueben-Keener	059-5673720	315354122
	19 Clay-Jackman	046-9324511	291897478
	20 Fisher-Webb	039-7827133	816197152
	21 Johnnie-Goodall	014-4727570	517788028
	22 Max-Popper	026-4668349	864747888
	23 Kiefer-Clarkson	040-7981665	220944628
	24 Diane-Cassidy	069-6793933	214774059

PROVIDERS

	PROVNAME	PROVPHONENUMBER	PROVID
◀	1 Jesus-LaSalle	032-9261177	67629442
	2 Avril-Levert	042-6815247	135836379
	3 Dick-Phoenix	077-5649844	439647872
	4 Diane-Tempest	079-5095358	965300293
	5 Olga-Coolidge	039-3964462	625197018
	6 Hugh-Haynes	018-8447548	752709756
	7 Hank-Krieger	089-3257255	633048403
	8 Ewan-Ponce	043-3568967	612887658
	9 Shawn-Meyer	044-2394165	174032265
	10 Vertical-Ricci	029-2405329	134116135
	11 Trini-Greenwood	077-8745089	772804873
	12 Chi-Barnett	030-3983449	112109750
	13 Kieran-Lowe	070-8224619	460094011
	14 Tyrone-Kinski	090-4487174	197359450
	15 Seth-Allen	084-6677859	540261844
	16 Isabella-Gyllenhaal	058-9733636	423186695
	17 Neil-Rickles	075-6260422	147618326
	18 Rueben-Keener	059-5673720	315354122
	19 Clay-Jackman	046-9324511	291897478
	20 Fisher-Webb	039-7827133	816197152
	21 Johnnie-Goodall	014-4727570	517788028
	22 Max-Popper	026-4668349	864747888
	23 Kiefer-Clarkson	040-7981665	220944628
	24 Diane-Cassidy	069-6793933	214774059

Text importer



Python script

```
e3.py - C:/Users/Naama/Downloads/e3.py (3.10.2)
File Edit Format Run Options Window Help
import random

# List of unique meal names without spaces (ensure there are at least 400 unique
meal_names = [
    "Schnitzel", "Pizza", "Burger", "Pasta", "Salad", "Soup", "Steak", "Tacos",
    "Sandwich", "Curry", "Burrito", "Sushi", "Ramen", "Paella", "Quiche",
    "Kebab", "Falafel", "DimSum", "Goulash", "Risotto", "ChowMein", "Pho",
    "Dumplings", "Lasagna", "Pierogi", "Crepes", "Fajitas", "HotDog", "Shawarma",
    "Ceviche", "Empanadas", "Moussaka", "Enchiladas", "Bruschetta", "Biryani",
    "Jambalaya", "Omelette", "Tapas", "Tagine", "Ratatouille", "Chili", "FriedRi
    "Gumbo", "ClamChowder", "BakedBeans", "Macaroni", "PotRoast", "Bangers",
    "BeefStew", "Cassoulet", "Chowder", "FishTacos", "Lobster", "Meatloaf", "Pol
    "ShepherdsPie", "TunaSalad", "WaldorfSalad", "ChickenWings", "ChiliDog",
    "CrabCakes", "EggRoll", "FishandChips", "FrenchToast", "HushPuppies",
    "JalapenoPoppers", "PekingDuck", "StuffedPeppers", "TeriyakiChicken",
    "TortillaSoup", "TurkeySandwich", "VeggieBurger", "BBQRibs", "ChickenFarm",
    "Cornbread", "GrilledCheese", "LambChops", "MashedPotatoes", "Meatballs",
    "Nachos", "Oysters", "Pancakes", "PulledPork", "Quesadilla", "Ravioli",
    "RedBeans", "Salmon", "ShrimpCocktail", "SloppyJoes", "Spaghetti", "Tofu",
    "ZucchiniBread", "Arancini", "BabaGanoush", "Bibimbap", "Calzone", "Carbonar
    "Chimichanga", "ChowMein", "Coquelin", "DuckConfit", "EggsBenedict", "Fajita
    "FishCurry", "Frittata", "Gazpacho", "Gnocchi", "Gravlax", "HuevosRancheros"
    "Kimchi", "Laksa", "MakiRoll", "Mole", "Moussaka", "NasiGoreng", "Onigiri",
    "PadThai", "Pastrami", "PatatasBravas", "PolloAsado", "Porchetta", "Poutine"
    "Ratatouille", "Roti", "Satay", "ScotchEgg", "Shakshuka", "Souvlaki", "Spana
    "StuffedCabbage", "Sukiyaki", "SurfandTurf", "TandooriChicken", "TarteTatin"
    "Tempura", "Tiramisu", "TomYum", "TunaMelt", "Vichyssoise", "Vindaloo", "Won
]

# Shuffle the meal names to randomize them
random.shuffle(meal_names)

data = []
for i in range(len(meal_names)):
    random_number = random.randint(111, 999)
    meal_name = meal_names[i]
    random_price = random.randint(15, 150)
    data.append(f"{random_number},{meal_name},{random_price}")

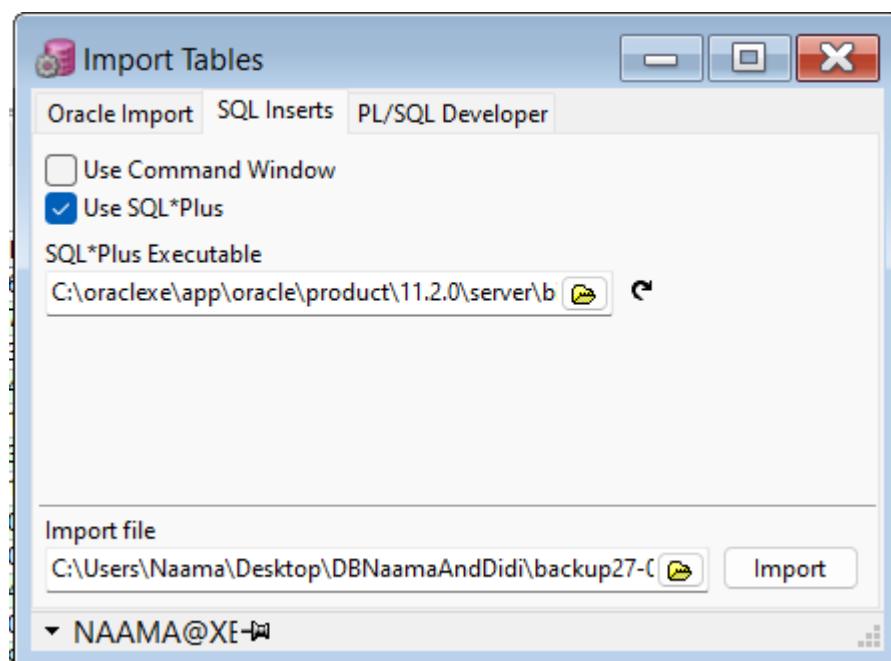
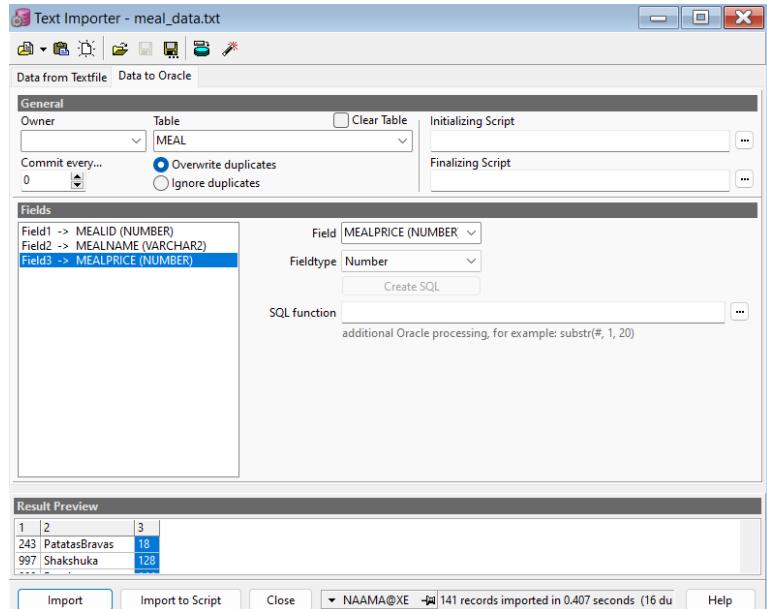
# Write data to a text file
with open("meal_data.txt", "w") as file:
    for line in data:
        file.write(line + "\n")

print("141 lines of data have been written to meal_data.txt")
```

הערות:

בטבלה product אין 400 נתונים כיון שדי קשה למצאו 400 שמות של מאכלים וחומר גלם 😊

גיבוי ו恢復



שאלות

4 SELECT

.1. בחר את העובדים מעל גיל 50 שעשו פחות מ 4 הזמנות יחד עם הפרטים שלהם.

```
--Select the employees over the age of 50 who made less than 4 orders along with their details.
SELECT workname, workerid, workerbirthdate
FROM workers w
WHERE 4 >= (SELECT COUNT(*) 
FROM workerorder wo
WHERE w.workerid = wo.workerid) AND (EXTRACT (year from w.workerbirthdate)) < 1975;
```

--Select the average order price per month for the current year.

	WORKNAME	WORKERID	WORKERBIRTHDATE
1	Colm-Nicks	333067647	18/10/1952
2	Tramaine-Milsap	718339906	25/02/1961
3	Penelope-Lopez	744468530	05/11/1965
4	Nastassja-Vannelli	611069104	01/01/1951
5	Joaquin-MacNeil	962903534	18/08/1965
6	Spike-McCready	544633195	19/05/1963
7	Claire-Jenkins	334885243	17/07/1960
8	Carrie-Anne-Stanley	314986077	03/08/1955
9	Beth-McCain	419094984	06/09/1966
10	Ernie-Shawn	550647367	01/11/1950
11	Lynette-Black	207467856	07/01/1952
12	Shirley-Webb	701137116	27/04/1951
13	Nils-Rebhorn	811740389	18/11/1964
14	Gin-Jeter	812761051	24/04/1959
15	Earl-Dayne	122887112	04/03/1971
16	Pelvic-Byrne	727731831	12/06/1966
17	Emm-Reubens	387317698	27/05/1964
18	Tom-King	168794272	13/04/1954

.2. בחר את המספר הכללי של הזמנות שביצעו כל לקוחות יחד עם הפרטים שלהם ומין מהגובה לנמו.

```
--4 Select Queries
--Select the total number of orders placed by each customer along with their details.
SELECT c.custName, c.custPhoneNumber, COUNT(o.orderID) AS totalOrders
FROM Customers c
JOIN Orders o ON c.custID = o.custID
GROUP BY c.custName, c.custPhoneNumber
ORDER BY totalOrders DESC;
```

```
--Select the workers who have managed the highest number of orders and their details.
SELECT w.workName, w.workerJobTitle, COUNT(wo.orderID) AS totalOrdersManaged
FROM Workers w
JOIN WorkerOrder wo ON w.workerID = wo.workerID
```

Select customers Select workers			
	CUSTNAME	CUSTPHONENUMBER	TOTALORDERS
1	Miki-Tsettos	051-8296868	6
2	Mika-Cagle	015-5239687	5
3	Ceili-Mantegna	029-8464688	4
4	Hank-Rickles	069-4765539	4
5	Larenz-Elwes	087-9815553	4
6	Paula-Palmieri	061-4590865	4
7	Charlize-Napolitano	067-7846134	4
8	Chazz-Folds	039-6797761	3

.3. בחר את העובדים שניהלו את מספר ההזמנות הגבוה ביותר ואת הפרטים שלהם.

```
--Select the workers who have managed the highest number of orders and their details.
SELECT w.workName, w.workerJobTitle, COUNT(wo.orderID) AS totalOrdersManaged
FROM Workers w
JOIN WorkerOrder wo ON w.workerID = wo.workerID
GROUP BY w.workName, w.workerJobTitle
HAVING COUNT(wo.orderID) = (
    SELECT MAX(orderCount)
    FROM (
        SELECT COUNT(orderID) AS orderCount
        FROM WorkerOrder
        GROUP BY workerID
    )
);
--Select the top 3 most expensive meals along with their ingredients and their respective prices.
```

	WORKNAME	WORKERJOBTITLE	TOTALORDERSMANAGED
1	Elle-Mac	Dishwasher	5

.4. בחר את מחיר ההזמנה הממוצע לחודש עבור השנה הנוכחיות.

```
--Select the average order price per month for the current year.
SELECT EXTRACT(MONTH FROM o.orderDate) AS month, AVG(o.orderPrice) AS avgOrderPrice
FROM Orders o
WHERE EXTRACT(YEAR FROM o.orderDate) = EXTRACT(YEAR FROM SYSDATE)
GROUP BY EXTRACT(MONTH FROM o.orderDate)
ORDER BY month;
```

	MONTH	AVGORDERPRICE
1	1	109.164705882353
2	2	124.464788732394
3	3	134.826086956522
4	4	148
5	5	120.082191780822

2 UPDATE

.1. עדכן את שכר העובדים שניהלו יותר מ-10 ההזמנות על ידי העלאתו ב-10%.

```
--2 Update Queries
--Update the salary of workers who have managed more than 10 orders by increasing it by 10%.
UPDATE Workers
SET workerSalary = workerSalary * 1.1
WHERE workerID IN (SELECT workerID FROM WorkerOrder GROUP BY workerID HAVING COUNT(orderID) > 10)
```

משכורת העובדים שעובדים בתנאים לפני הרצת השאלה:

	WORKERID	WORKNAME	WORKERJOBTITLE	WORKERSALARY	WORKERBIRTHDATE
1	143274912	Elle-Mac	Dishwasher	16635	17/07/1987

אחרי:

	WORKERID	WORKNAME	WORKERJOBTITLE	WORKERSALARY	WORKERBIRTHDATE
1	143274912	Elle-Mac	Dishwasher	18299	17/07/1987

.2. עדכן את הסטטוס של כל ההזמנות שבוצעו לפני יותר מחודש ל"הושלם".

```
--Update the status of all orders placed more than a month ago to 'completed'.
UPDATE Orders
SET orderStatus = 'Delivered'
WHERE orderDate < ADD_MONTHS(SYSDATE, -1);
```

סטטוס ההזמנות לפני יותר מחודש לפני הרצת השאלה:

Select orders Update orders Select orders

	ORDERID	ORDERPRICE	ORDERSTATUS	ORDERDATE	CUSTID
▶ 1	59182	188	Received	... 04/01/2024	▼ 199640888
2	65592	90	Received	... 20/04/2024	▼ 296255366
3	61398	135	Received	... 11/01/2024	▼ 216541625
4	41599	299	Received	... 28/02/2024	▼ 296255366
5	80222	259	Received	... 04/02/2024	▼ 390110769
6	84259	145	Received	... 06/04/2024	▼ 789562188
7	56419	269	Received	... 27/01/2024	▼ 738961356
8	50051	1	Received	... 06/03/2024	▼ 974693239
9	88351	72	Received	... 22/02/2024	▼ 966414435
10	88653	179	Received	... 04/01/2024	▼ 656442110

אחרי:

Select orders Update orders Select orders

	ORDERID	ORDERPRICE	ORDERSTATUS	ORDERDATE	CUSTID
▶ 1	59182	188	Delivered	... 04/01/2024	▼ 199640888
2	65592	90	Delivered	... 20/04/2024	▼ 296255366
3	61398	135	Delivered	... 11/01/2024	▼ 216541625
4	41599	299	Delivered	... 28/02/2024	▼ 296255366
5	80222	259	Delivered	... 04/02/2024	▼ 390110769
6	84259	145	Delivered	... 06/04/2024	▼ 789562188
7	56419	269	Delivered	... 27/01/2024	▼ 738961356
8	50051	1	Delivered	... 06/03/2024	▼ 974693239
9	88351	72	Delivered	... 22/02/2024	▼ 966414435
10	88653	179	Delivered	... 04/01/2024	▼ 656442110

2 DELETE

(או צילום מסך של מסד הנתונים לפני ואחריו כו ראיית הסעיף רק אחרי שהרצתי כבר את השאלות ?).

1. מחק את כל ההזמנות בנות יותר משנה מהיום.

```
--2 Delete Queries
--Delete all orders that are older than one year from today.
DELETE FROM Orders
WHERE orderDate < ADD_MONTHS(SYSDATE, -12);
```

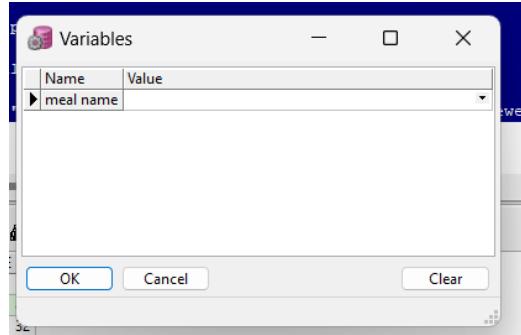
2. מחק את כל הלקוחות שלא ביצעו הזמנות.

```
--Delete all customers who have not placed any orders.
DELETE FROM Customers
WHERE custID NOT IN (SELECT DISTINCT custID FROM Orders);
```

שאלות עם פרמטרים

4 SELECT

1. נרצה לדעת את כל המרכיבים במנה מסוימת. הוספנו בשאלתה רשימה עם שמות כל המרכיבים כדי שהמרץ יוכל לבחור ממנה לבחירתו.

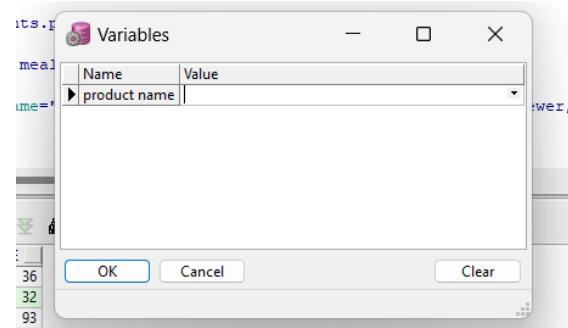


```
--Select all products that are in a specific meal
SELECT
    Product.prodName,
    Ingredients.ingredAmount
FROM
    Ingredients
JOIN
    Product ON Ingredients.prodID = Product.prodID
JOIN
    Meal ON Ingredients.mealID = Meal.mealID
WHERE
    Meal.mealName = &<name="meal name" list="Schnitzel, shawarma, chicken-skewer, entrecote-skewer,

```

PRODNAME	INGREDAMOUNT
1 chicken soup	169
2 salt	103
3 pickles	87
4 hummus	34
5 chicken-breast	21
6 cucumber	195
7 steak	47
8 tahini	75
9 eggplant	117
10 turmeric	136
11 paprika	190

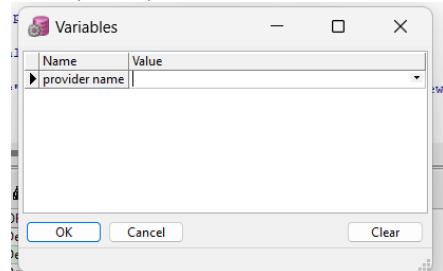
2. נרצה לבחור את כל המנות שמכילים מוצר מסוים – לדוגמא בשבייל אונשיים אלרגיים.



```
--Select the meals that contain a specific ingredient.
SELECT m.mealName, m.mealPrice
FROM Meal m
JOIN Ingredients i ON m.mealID = i.mealID
JOIN Product p ON i.prodID = p.prodID
WHERE p.prodName = &<name="product name" list="Sugar, salt, turmeric, chicken soup, paprika, black-pepper, chicken-breast, eggs, pot
```

MEALNAME	MEALPRICE
1 steak	52
2 kebab-skewer	86
3 Meatballs	32
4 Dumplings	37
5 water	34
6 beer	88
7 entrecote-skewer	20

3. נרצה לראות את כל המזמרים שמספק לנו ספק מסוים.



```
--Select products provided by a specific provider.
SELECT prodName, prodPrice
FROM Product
WHERE provID = (SELECT provID FROM Providers WHERE provName = &<name="provider name" list='

--Select the meals that contain a specific ingredient.
```

	PRODNAME	PRODPRI
▶ 1	juice	36
2	vanile	32
3	paprica	93
4	water	103

.4 נרצה לראות את כל ההזמנות שנעשו בטוויח תאריכים מסוימים.
השתמשנו בשאליתה `hint` שאומר למשתמש באיזה פורטט להכניס את הערכים. בזדב שווה `to_date` של `string` כדי שלא נדרש להמיר את הערכים למחוזות.

`ts.p` Variables

Name	Value
s_date	01-01-2024
e_date	01-06-2024

OK Cancel Clear

```
--Select orders within a specific date range.
SELECT orderID, orderPrice, orderStatus, orderDate
FROM Orders
WHERE orderDate BETWEEN TO_DATE(&<name="s_date" hint="In format DD-MM-YYYY" type="string"
&<name="e_date" hint="In format DD-MM-YYYY" type="string"

--Select products provided by a specific provider.
```

	ORDERID	ORDERPRICE	ORDERSTATUS	ORDERDATE
▶ 1	59182	188	Delivered	... 04/01/2024
2	65592	90	Delivered	... 20/04/2024
3	61398	135	Delivered	... 11/01/2024
4	41599	299	Delivered	... 28/02/2024
5	80222	259	Delivered	... 04/02/2024
6	84259	145	Delivered	... 06/04/2024
7	56419	269	Delivered	... 27/01/2024
8	50051	1	Delivered	... 06/03/2024
9	88351	72	Delivered	... 22/02/2024
10	88653	179	Delivered	... 04/01/2024
11	13604	75	Delivered	... 13/03/2024
12	52830	135	Delivered	... 10/02/2024
13	90296	149	Delivered	... 17/01/2024
14	81313	27	Delivered	... 26/04/2024
15	12869	179	Delivered	... 12/02/2024
16	53802	10	Delivered	... 06/02/2024
17	95864	179	Delivered	... 24/01/2024
18	26757	140	Delivered	... 02/03/2024

AILOZIM

רשמנן 3 אילוצים הכרחיים: שהמשכורת, מחירי המוצרים וסכום ההזמנה לא יוכל להיות שליליים.

```
ALTER TABLE Workers
ADD CONSTRAINT CHK_WorkerSalary CHECK (workerSalary > 0);

ALTER TABLE Product
ADD CONSTRAINT CHK_ProdPrice CHECK (prodPrice > 0);

ALTER TABLE Orders
ADD CONSTRAINT CHK_OrderPrice CHECK (orderPrice >= 0);
```

עכשו ננסה להוסיף מוצר עם מחיר שלילי

```
insert into PRODUCT (PRODID, PRODNAME, PRODPRIICE, PRODAMOUNT, PROVID)
values (087, 'eg0gs', 73, -96, 781585360);
commit;
```

נרים ונבדוק אם המוצר נכנס לatable ונראה שלא:

```
SELECT *
FROM PRODUCT P
WHERE P.PRODPRIICE <0;
```

	PRODID	PRODNAME	PRODPRIICE	PRODAMOUNT	PROVID

גיבוי

יצור קובץ גיבוי בשם backup2.sql:

The screenshot shows the Oracle SQL Developer interface. At the top, there is a toolbar with various icons. Below it is a table view showing nine database tables: CUSTOMERS, INGREDIENTS, MEAL, MEALORDER, ORDERS, PRODUCT, PROVIDERS, WORKERORDER, and WORKERS. Each row includes columns for Name, Type, and Compiled date.

Below the table view is a configuration panel for an export operation:

- Oracle Export** tab is selected.
- SQL Inserts**, **PL/SQL Developer**, and **Log** tabs are also visible.
- Options:**
 - Drop tables
 - Create tables
 - Truncate tables
 - Delete records
 - Disable triggers
 - Disable foreign key constraints
 - Include storage
 - Include privileges
- Commit every**: A dropdown menu set to 100 records (0 = never).
- Where clause**: An empty text input field.
- Zip**: A checkbox that is unchecked.

Output file: C:\Users\Naama\Desktop\DBNaamaAndDidi\backup2.sql

Buttons: A yellow "Export" button and a grey "Cancel" button.

At the bottom left, a status bar shows the connection information: NAAMA@XE and the progress: Exporting... Done.