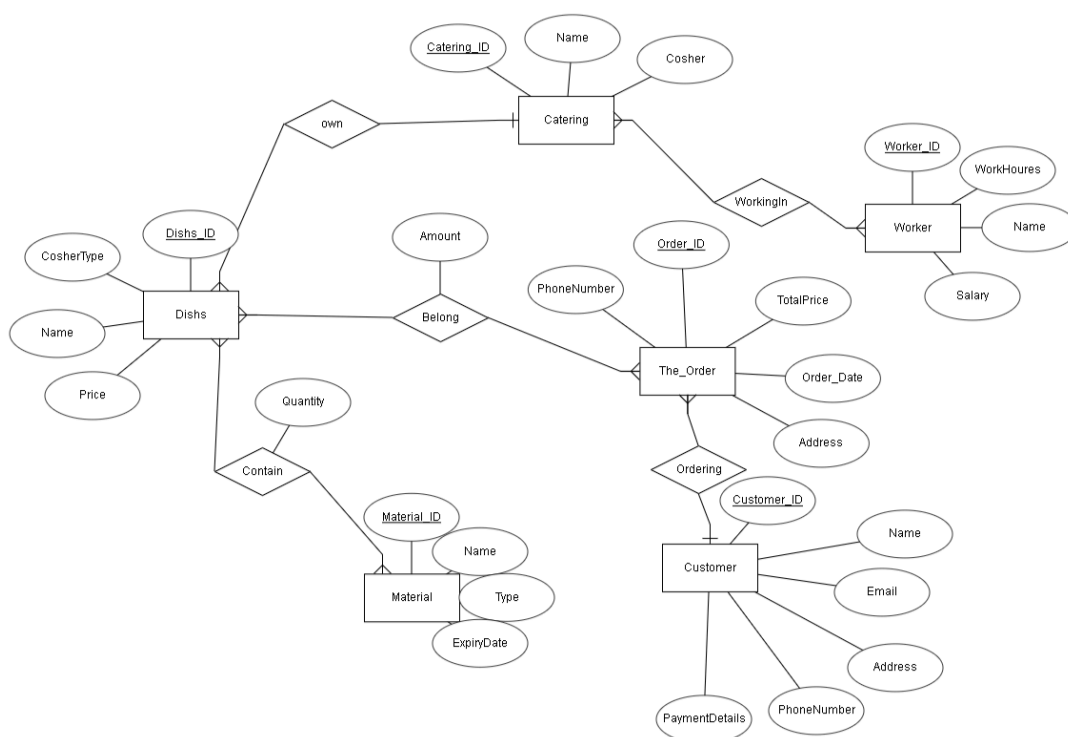


דו"ח מיני פרויקט בבסיסי נתונים

שרית תיק - 213230048 ואיטי ישראלי - 213712367

אוכל - קייטרינג

יצרנו דיאגרמת ERD המתארת פרויקט קייטרינג שבו בחרנו בכיתה.



catering - ישות הקייטרינג מכילה בתוכה 3 אטריביוטים. אטריביוט לזיהוי, לשם הקייטרינג וכשרות הקייטרינג קישרנו ישות זו לישות **Worker** בקשר של רבים לרבים משום שלכל קייטרינג יכול להיות אפס עובדים או יותר ויכול להיות עובש שעובד באפס קייטרינגים או יותר. וישות **Dishes** קישרנו אותה לקייטרינג בקשר של לכל היותר אחד כי לכל קייטרינג יכול להיות אפס או יותר מנות ולכל תבשיל חייב להיות במקסימום קייטרינג אחד שמכין אותה (תבשיל מיוחד לקייטרינג ספציפי):

Worker - ישות עובד מכילה בתוכה 4 אטריביוטים. אטריביוט לזיהוי, שם העובד, שעות עבודה ומשכורת
Dishes - ישות מנות מכילה בתוכה 4 אטריביוטים. אטריביוט לזיהוי, חלבי/בשרי, שם המאכל, ומחיר המאכל
 קישרנו ישות זו לישות **Material** בקשר של רבים ליחיד משום שלכך תבשיל יש אפס או יותר מרכיבים ולכל מרכיב הוא יכול להיות באפס או תבשיל אחד בדיוק, וישות **The_Order** בקשר של רבים לרבים משום שלכל הזמנה יכול להיות אפס או יותר תבשילים וכל תבשיל יכול להיות באפס או יותר הזמנות:

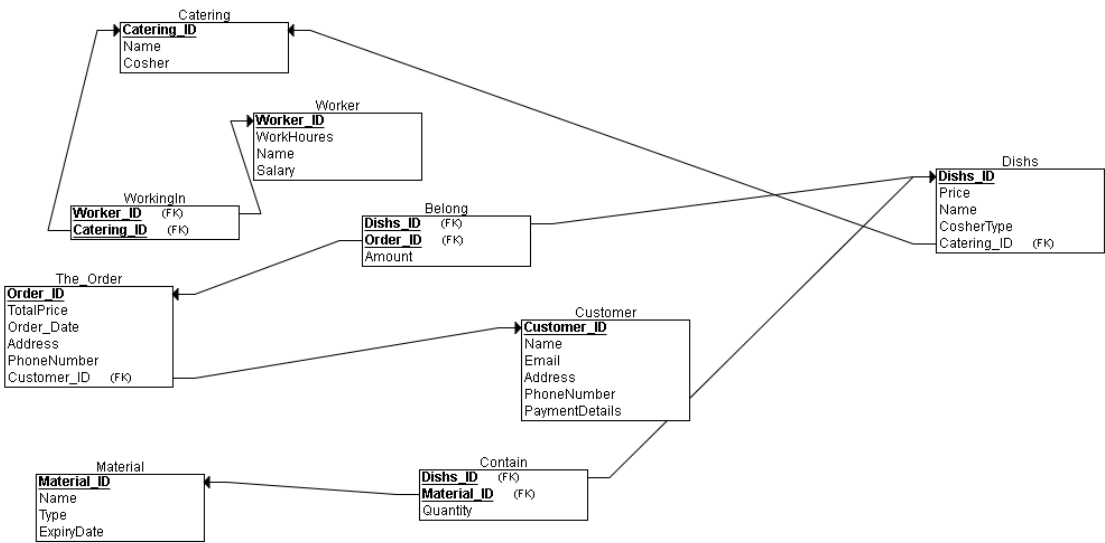
Material - ישות חומר גלם מכילה בתוכה 4 אטריביוטים. אטריביוט לזיהוי, לשם החומר גלם, לסוג החומר גלם (צומח, אביזרי מכולת וכו') ותאריך תפוגת החומר גלם.

The_Order - ישות הזמנות מכילה בתוכה חמישה אטריביוטים. אטריביוט לזיהוי, כתובת המזמין, טלפון המזמין, תאריך ההזמנה ומחיר כולל של ההזמנה

קישרנו ישות זו לישות **Customer** בקשר של בדיוק אחד משום שכל הזמנה שייכת ללקוח אחד בלבד:

Customer - ישות לקוח מכילה בתוכה שישה אטריביוטים. אטריביוט לזיהוי, שם הלקוח, כתובת מייל של הלקוח, כתובת הלקוח, מספר הטלפון של הלקוח, ופרטי תשלום (אשראי, פיפאל וכו').

תרשים הDSD:



פקודת הCreateTable:

הוספנו אילוצים על שני טבלאות:

Catering - על שדה הכשרות הוספנו את האופציה שיוכלו רק להכניס Y/N, הכוונה האם הקייטרינג כשר או לא.

Dishes - על שדה הסוג כשרות, שיוכלו להכניס את האופציות חלבי/ בשרי/ או אף אחד - פרווה, וככה ידעו

באיזה מאכל מדובר, מאכל חלבי, מאכל בשרי או מאכל פרווה.

```
CREATE TABLE Catering
(
    Catering_ID NUMERIC(5) NOT NULL,
    Name VARCHAR(15) NOT NULL,
    Coshier CHAR(1) NOT NULL CHECK(coshier='Y' or coshier='N'),
    PRIMARY KEY (Catering_ID)
);

CREATE TABLE Worker
(
    Worker_ID NUMERIC(5) NOT NULL,
    WorkHoures NUMERIC(2) NOT NULL,
    Name VARCHAR(15) NOT NULL,
    Salary NUMERIC(6) NOT NULL,
    PRIMARY KEY (Worker_ID)
);

CREATE TABLE Dishes
(
    Dishes_ID NUMERIC(5) NOT NULL,
    Price NUMERIC(4) NOT NULL,
    Name VARCHAR(15) NOT NULL,
    CoshierType VARCHAR(7) NOT NULL CHECK(coshierType='Dairy' or coshierType='Meat' or coshierType='None'),
    Catering_ID NUMERIC(5) NOT NULL,
    PRIMARY KEY (Dishes_ID),
    FOREIGN KEY (Catering_ID) REFERENCES Catering(Catering_ID)
);

CREATE TABLE Material
(
    Material_ID NUMERIC(5) NOT NULL,
    Name VARCHAR(15) NOT NULL,
    Type VARCHAR(15) NOT NULL,
    ExpiryDate DATE NOT NULL,
    PRIMARY KEY (Material_ID)
);

CREATE TABLE Customer
(
    Customer_ID NUMERIC(5) NOT NULL,
    Name VARCHAR(15) NOT NULL,
    Email VARCHAR(20) NOT NULL,
    Address VARCHAR(20) NOT NULL,
    PhoneNumber VARCHAR(11) NOT NULL,
    PaymentDetails VARCHAR(10) NOT NULL,
    PRIMARY KEY (Customer_ID)
);
```

```

Email VARCHAR(20) NOT NULL,
Address VARCHAR(20) NOT NULL,
PhoneNumber VARCHAR(11) NOT NULL,
PaymentDetails VARCHAR(10) NOT NULL,
PRIMARY KEY (Customer_ID)
);

CREATE TABLE WorkingIn
(
    Worker_ID NUMERIC(5) NOT NULL,
    Catering_ID NUMERIC(5) NOT NULL,
    PRIMARY KEY (Worker_ID, Catering_ID),
    FOREIGN KEY (Worker_ID) REFERENCES Worker(Worker_ID),
    FOREIGN KEY (Catering_ID) REFERENCES Catering(Catering_ID)
);

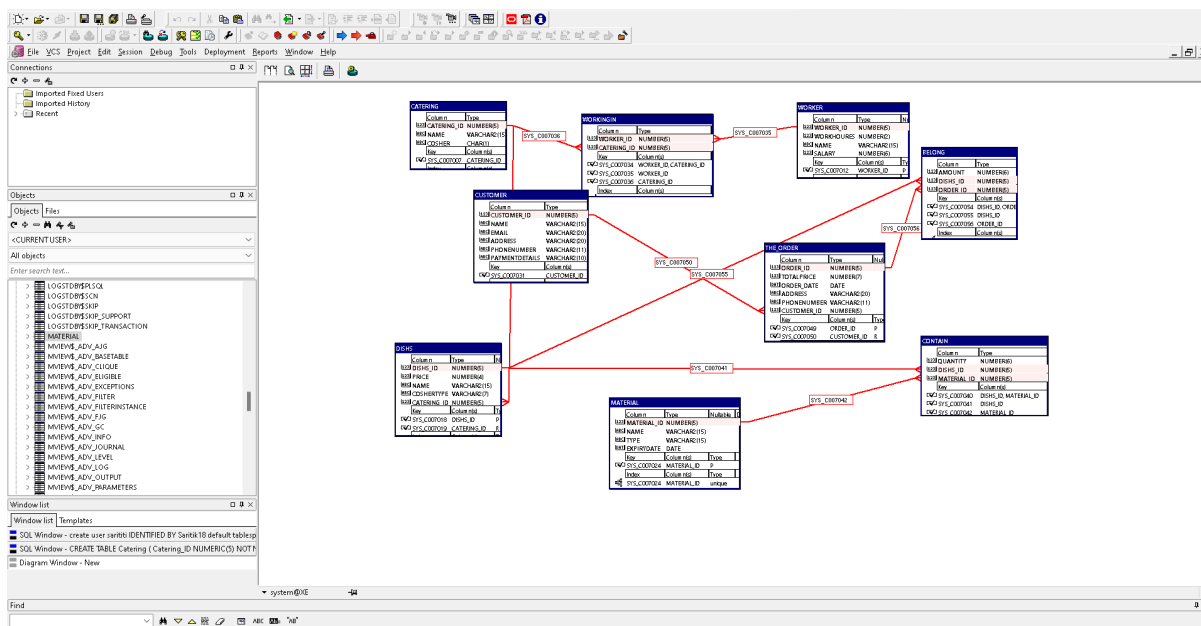
CREATE TABLE Contain
(
    Quantity NUMERIC(6) NOT NULL,
    Dishes_ID NUMERIC(5) NOT NULL,
    Material_ID NUMERIC(5) NOT NULL,
    PRIMARY KEY (Dishes_ID, Material_ID),
    FOREIGN KEY (Dishes_ID) REFERENCES Dishes(Dishes_ID),
    FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID)
);

CREATE TABLE The_Order
(
    Order_ID NUMERIC(5) NOT NULL,
    TotalPrice NUMERIC(7) NOT NULL,
    Order_Date DATE NOT NULL,
    Address VARCHAR(20) NOT NULL,
    PhoneNumber VARCHAR(11) NOT NULL,
    Customer_ID NUMERIC(5) NOT NULL,
    PRIMARY KEY (Order_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);

CREATE TABLE Belong
(
    Amount NUMERIC(6) NOT NULL,
    Dishes_ID NUMERIC(5) NOT NULL,
    Order_ID NUMERIC(5) NOT NULL,
    PRIMARY KEY (Dishes_ID, Order_ID),
    FOREIGN KEY (Dishes_ID) REFERENCES Dishes(Dishes_ID),
    FOREIGN KEY (Order_ID) REFERENCES The_Order(Order_ID)
);

```

טבלאות הSQL:



Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
Connected as Saritit18XE

```
SQL> desc Belong
Name      Type      Nullable Default Comments
-----
AMOUNT    NUMBER(5)
DISHS_ID  NUMBER(5)
ORDER_ID  NUMBER(5)
```

```
SQL> desc Catering
Name      Type      Nullable Default Comments
-----
CATERING_ID NUMBER(5)
NAME      VARCHAR2(15)
COSHER    CHAR(1)
```

```
SQL> desc Contain
Name      Type      Nullable Default Comments
-----
QUANTITY  NUMBER(6)
DISHS_ID  NUMBER(5)
MATERIAL_ID NUMBER(5)
```

```
SQL> desc Customer
Name      Type      Nullable Default Comments
-----
CUSTOMER_ID NUMBER(5)
NAME      VARCHAR2(15)
EMAIL     VARCHAR2(20)
ADDRESS   VARCHAR2(20)
PHONENUMBER VARCHAR2(11)
PAYMENTDETAILS VARCHAR2(10)
```

```
SQL> desc Dishes
Name      Type      Nullable Default Comments
-----
DISHS_ID  NUMBER(5)
PRICE     NUMBER(4)
NAME      VARCHAR2(15)
COSHERTYPE VARCHAR2(7)
CATERING_ID NUMBER(5)
```

```
SQL> desc Material
Name      Type      Nullable Default Comments
-----
MATERIAL_ID NUMBER(5)
```

```
EMAIL     VARCHAR2(20)
ADDRESS   VARCHAR2(20)
PHONENUMBER VARCHAR2(11)
PAYMENTDETAILS VARCHAR2(10)
```

```
SQL> desc Dishes
Name      Type      Nullable Default Comments
-----
DISHS_ID  NUMBER(5)
PRICE     NUMBER(4)
NAME      VARCHAR2(15)
COSHERTYPE VARCHAR2(7)
CATERING_ID NUMBER(5)
```

```
SQL> desc Material
Name      Type      Nullable Default Comments
-----
MATERIAL_ID NUMBER(5)
NAME      VARCHAR2(15)
TYPE      VARCHAR2(15)
EXPIRYDATE DATE
```

```
SQL> desc The_Order
Name      Type      Nullable Default Comments
-----
ORDER_ID  NUMBER(5)
TOTALPRICE NUMBER(7)
ORDER_DATE DATE
ADDRESS   VARCHAR2(20)
PHONENUMBER VARCHAR2(11)
CUSTOMER_ID NUMBER(5)
```

```
SQL> desc Worker
Name      Type      Nullable Default Comments
-----
WORKER_ID NUMBER(5)
WORKHOURES NUMBER(2)
NAME      VARCHAR2(15)
SALARY    NUMBER(6)
```

```
SQL> desc WorkingIn
Name      Type      Nullable Default Comments
-----
WORKER_ID NUMBER(5)
CATERING_ID NUMBER(5)
```

צילומי מסך של 3 השיטות

שימוש ב-Data Generator: (שמתי טבלה אחת לדוגמא כך בוצעו כל שאר הטבלאות)

WORKER

Owner

SARITITI

Table

WORKER

Number of records

400

Name	Type	Size	Data	Master
WORKER_ID	NUMBER	5	Sequence(10, 10)	
WORK_HOURS	NUMBER	2	Random(4, 12)	
NAME	VARCHAR2	15	FirstName + LastName	
SALARY	NUMBER	6	Random(4000, 20000)	

```
insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (10, 8, 'JonnyLaw', 12405);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (20, 11, 'JillShawn', 7380);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (30, 5, 'MacleyRobinson', 17094);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (40, 8, 'ChuckByars', 18417);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (50, 7, 'CharlesApplegate', 17406);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (60, 7, 'ValCarwen', 11853);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (70, 10, 'DavidOchit', 13603);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (80, 11, 'BillyGarr', 7561);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (90, 7, 'DonShames', 4366);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (100, 6, 'DiamondChanning', 13256);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (110, 10, 'Rhyssuvail', 6892);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (120, 4, 'EctApplegate', 13085);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (130, 7, 'NellyApple', 18779);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (140, 7, 'Goddametrading', 17215);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
values (150, 12, 'MacnameBrown', 5126);

insert into SARITITI.WORKER (WORKER_ID, WORK_HOURS, NAME, SALARY)
```

שימוש בקוד python: (קוד הpython פלט לי קובץ טקסט)

```
CustomerFile.py
1 import random
2 import string
3
4
5 # Function to generate a random full name with a maximum length of 15 characters
6 def random_name():
7     first_names = ['John', 'Jane', 'Alex', 'Emily', 'Chris', 'Katie', 'Michael', 'Sarah', 'David', 'Laura']
8     last_names = ['Smith', 'Johnson', 'Williams', 'Jones', 'Brown', 'Davis', 'Miller', 'Wilson', 'Moore', 'Taylor']
9     while True:
10         name = f'{random.choice(first_names)} {random.choice(last_names)}'
11         if len(name) <= 15:
12             return name
13         if len(first_names) > 0:
14             first_names.pop() # Reduce the pool to ensure we don't get into an infinite loop
15
16
17 # Function to generate a random email with a maximum length of 20 characters
18 def random_email(name):
19     domains = ['gmail.com', 'yahoo.com', 'hotmail.com', 'outlook.com', 'example.com']
20     email_prefix = name.replace(' ', '.').lower()
21     email_suffix = random.choice(domains)
22     email = f'{email_prefix}@{email_suffix}'
23     if len(email) > 20:
24         email = email[:20 - len(email_suffix) - 1] + '@' + email_suffix
25     return email[:20]
26
27
28 # Function to generate a random address with a maximum length of 20 characters
29 def random_address():
```

```
CustomerFile.py
25     return email[:20]
26
27
28 # Function to generate a random address with a maximum length of 20 characters
29 def random_address():
30     streets = ['Main', 'High', 'Broadway', 'Elm', 'Maple', 'Oak', 'Pine', 'Cedar', '2nd', '3rd']
31     cities = ['Springfield', 'Riverside', 'Greenfield', 'Franklin', 'Clinton', 'Fairview', 'Greenville', 'Bristol', 'Madison', 'Georgetown']
32     states = ['CA', 'TX', 'NY', 'FL', 'IL', 'PA', 'OH', 'MI', 'GA', 'NC']
33     address = f"{random.randint(100, 999)} {random.choice(streets)} {random.choice(cities)} {random.choice(states)}"
34     return address[:20]
35
36
37 # Function to generate a random phone number
38 def random_phone():
39     return f"{random.randint(100, 999)}-{random.randint(100, 999)}{random.randint(1000, 9999)}"
40
41
42 # Function to generate random payment details
43 def random_payment():
44     methods = ['CreditCard', 'Cash', 'Bit']
45     return random.choice(methods)
46
47
48 # Function to generate a unique ID
49 def unique_id(existing_ids):
50     while True:
51         new_id = random.randint(1, 99999)
52         if new_id not in existing_ids:
53             existing_ids.add(new_id)
54             return new_id
```

```
CustomerFile.py
49 def unique_id(existing_ids):
50     while True:
51         new_id = random.randint(1, 99999)
52         if new_id not in existing_ids:
53             existing_ids.add(new_id)
54             return new_id
55
56
57 def generate_data_file(filename, num_rows):
58     existing_ids = set()
59     with open(filename, 'w') as file:
60         for _ in range(num_rows):
61             id = unique_id(existing_ids)
62             name = random_name()
63             email = random_email(name)
64             address = random_address()
65             phone = random_phone()
66             payment = random_payment()
67             row = f"{id},{name},{email},{address},{phone},{payment}\n"
68             file.write(row)
69
70
71 # Generate the data file with 500 rows
72 generate_data_file('Customer.txt', 500)
73
74 print("Data file generated successfully.")
75
76
```

```
David Moore,david.mo@outlook.com,467 Pine Riverside,T.183-9124185,CreditCard,8436
Alex Taylor,alex.tay@outlook.com,357 Main Fairview,PA.163-5823945,Bit,96392
John Smith,john.sm@icloud.com,904 Elm Clinton,TX.756-6263699,CreditCard,98984
Katie Brown,katie.br@example.com,331 3rd Madison,IL.743-8999983,Bit,33584
Alex Brown,alex.brown@yahoo.com,738 Pine Madison,OH.800-7192006,Bit,51400
Laura Wilson,laura.w@outlook.com,370 Elm Springfield,330-3236814,Cash,74647
John Brown,john.br@outlook.com,235 Pine Madison,CA.436-9646337,Cash,39945
Katie Jones,katie.jo@outlook.com,923 Broadway Greenfield,624-6472791,CreditCard,93298
Laura Brown,laura.br@outlook.com,746 2nd Fairview,NY.382-4681095,Cash,11706
Katie Taylor,katie.ta@hotmail.com,322 Pine Greenfield,252-7183804,CreditCard,1067
Alex Miller,alex.mil@hotmail.com,615 3rd Greenville,CA.89-8874555,Bit,40393
Alex Wilson,alex.w@outlook.com,918 Elm Georgetown,T.340-5918073,Cash,69664
Michael Jones,michael.jo@outlook.com,261 Pine Riverside,C.558-9971312,CreditCard,83545
John Smith,john.smith@yahoo.com,760 Maple Fairview,M.180-1945599,Bit,29721
John Brown,john.brown@gmail.com,122 Oak Springfield,366-5374829,Cash,79861
Emily Brown,emily.br@outlook.com,628 Main Clinton,PA.612-5818247,CreditCard,46666
Sarah Moore,sarah.mo@outlook.com,774 Cedar Greenville,584-9235980,Cash,15581
Emily Brown,emily.br@hotmail.com,673 Cedar Greenville,565-8913519,Bit,81596
Chris Taylor,chris.ta@hotmail.com,721 Main Franklin,OH.487-4362221,Bit,8358
Chris Moore,chris.mo@outlook.com,896 Elm Springfield,430-7376615,Cash,24296
Michael Jones,michael.jo@gmail.com,795 Elm Madison,NC.854-1817479,CreditCard,62454
Jane Davis,jane.dav@outlook.com,834 Broadway Clinton,808-9288402,Cash,47195
John Johnson,john.john@gmail.com,839 Elm Greenville,N.272-7644227,Bit,3305
Laura Johnson,laura.jo@hotmail.com,704 Broadway Franklin,650-1058156,Cash,33920
Jane Johnson,jane.jh@outlook.com,578 Pine Fairview,PA.696-4353335,Cash,20720
Jane Moore,jane.mo@hotmail.com,507 3rd Springfield,299-5322274,Bit,57611
Katie Jones,katie.jo@outlook.com,155 3rd Franklin,NY.598-9106113,CreditCard,27791
Emily Jones,emily.jo@example.com,244 Elm Bristol,IL.747-1244794,Cash,26189
John Smith,john.sm@example.com,431 Elm Springfield,429-1484917,Bit,48155
Katie Miller,katie.mil@yahoo.com,979 2nd Greenfield,C.299-2474396,Cash,33040
Sarah Smith,sarah.sm@example.com,837 Pine Fairview,CA.859-2905573,Cash,49645
Alex Moore,alex.moore@gmail.com,500 Elm Greenville,T.632-2388448,CreditCard,55703
Katie Davis,katie.da@hotmail.com,486 Cedar Clinton,PA.159-2643007,Bit,30591
Michael Wilson,michael.wi@yahoo.com,282 Pine Bristol,TX.799-7762339,Bit,54590
Sarah Miller,sarah.mil@yahoo.com,928 Pine Madison,OH.210-3775555,Cash,51101
```

The screenshot displays the Oracle Data Loader (ODL) interface, specifically the 'File Data' tab. The interface is divided into several sections: 'File Data', 'Configuration', 'Result Preview', and 'Import'.

File Data: This section shows a list of data rows. The columns are: ID, Name, Address, and Credit Card. The data is as follows:

ID	Name	Address	Credit Card
58416	David Moore	david.mo@outlook.com	467 Pine Riverside T, 103-9124185
90382	Alex Taylor	alex.ty@outlook.com	957 Main Fairview PA, 163-5823945
89894	John Smith	john.smith@hotmail.com	904 Elm Clinton T2, 756-6263689
33586	Katie Brown	katie.br@example.com	331 3rd Madison IL, 743-8999983
62400	Alex Brown	alex.brown@yahoo.com	735 Pine Madison OH, 800-7150046
74647	Laura Wilson	laura.wilson@yahoo.com	370 Elm Springfield, 930-3236014
39946	John Brown	john.brown@outlook.com	235 Pine Madison CA, 436-9264337
93259	Katie Jones	katie.j@outlook.com	953 Broadway Greenville, 624-672791
11706	Laura Brown	laura.br@outlook.com	746 2nd Fairview NY, 382-4681095
1067	Katie Taylor	katie.ta@hotmail.com	322 Pine Greenwood, 252-7183804
40393	Alex Miller	alex.mil@hotmail.com	615 3rd Greenville C, 489-8874555
69564	Alex Wilson	alex.wilson@yahoo.com	918 Elm Georgetown T, 343-5918073
83545	Michael Jones	michael.j@outlook.com	361 Pine Riverside C, 558-9971912
2871	John Smith	john.smith@yahoo.com	760 Maple Fairview M, 180-1844599
79861	John Brown	john.brown@gmail.com	122 Oak Springfield, 346-5374829
6656	Emily Brown	emily.br@outlook.com	558 Main Clinton PA, 616-5810247
15661	Nathan Moore	nathan.moore@yahoo.com	774 Cedar Greenville, 884-9525980
81596	Emily Brown	emily.br@hotmail.com	673 Cedar Greenville, 561-8913159
9358	Chris Taylor	chris.ta@hotmail.com	721 Main Franklin OH, 487-4362221
24296	Chris Moore	chris.mo@outlook.com	696 Elm Springfield, 490-7376615
62454	Michael Jones	michael.j@gmail.com	795 Elm Madison NC, 854-1617479
47356	Jane Davis	jane.davis@outlook.com	874 Broadway Clinton, 808-9308402
3305	John Johnson	john.john@gmail.com	839 Elm Greenville M, 272-694227
33920	Laura Johnson	laura.j@hotmail.com	704 Broadway Franklin, 650-1058156
20720	John Johnson	john.john@outlook.com	378 Pine Fairview PA, 696-4353333
67611	Jane Moore	jane.mo@hotmail.com	507 3rd Springfield, 299-9322274

Configuration: This section shows settings for the 'File Data' tab. The 'Field Start' is set to 'Relative position' and 'Field End' is set to 'Character'. The 'Result Preview' section shows a preview of the data rows, including columns for ID, Name, Address, and Credit Card.

Result Preview: This section shows a preview of the data rows, including columns for ID, Name, Address, and Credit Card. The data is as follows:

ID	Name	Address	Credit Card
58416	David Moore	david.mo@outlook.com	467 Pine Riverside T, 103-9124185
90382	Alex Taylor	alex.ty@outlook.com	957 Main Fairview PA, 163-5823945

Import: This section shows the 'Import' button and the 'Import to Script' button. The 'Import' button is disabled, and the 'Import to Script' button is enabled. The 'Import to Script' button is labeled 'Import to Script'.

Configuration (Table): This section shows settings for the 'Table' tab. The 'Table' is set to 'CUSTOMER'. The 'Field Start' is set to 'Relative position' and 'Field End' is set to 'Character'. The 'Result Preview' section shows a preview of the data rows, including columns for ID, Name, Address, and Credit Card.

Result Preview (Table): This section shows a preview of the data rows, including columns for ID, Name, Address, and Credit Card. The data is as follows:

ID	Name	Address	Credit Card
58416	David Moore	david.mo@outlook.com	467 Pine Riverside T, 103-9124185
90382	Alex Taylor	alex.ty@outlook.com	957 Main Fairview PA, 163-5823945

שימוש בקבצי Excel: (שוב שמתי טבלה אחת לדוגמא כך בוצעו כל שאר הטבלאות)

[illegible]

Data from TextfileData to Oracle

The Data

Encoding

"", "CATERING_ID", "NAME", "COSHER"
"1", "30", "Charlotte", "N"
"2", "46", "Chubby", "N"
"3", "60", "Poppy", "Y"
"4", "70", "Gean", "Y"
"5", "73", "Maynard", "Y"
"6", "87", "Ido", "Y"
"7", "102", "Clarence", "Y"
"8", "113", "Alannah", "Y"
"9", "135", "Radosy", "N"
"10", "184", "Rose", "N"
"11", "200", "Goran", "N"
"12", "202", "Mike", "Y"
"13", "209", "Samuel", "Y"
"14", "257", "Fonessa", "Y"
"15", "290", "Bennis", "N"
"16", "291", "Ela", "Y"
"17", "313", "Luka", "N"
"18", "324", "Ellie", "Y"
"19", "326", "Billey", "Y"
"20", "344", "Nicola", "N"
"21", "349", "Fiona", "N"
"22", "401", "Gladys", "N"
"23", "430", "Howard", "N"
"24", "433", "Roberta", "Y"
"25", "468", "Borace", "N"

Configuration

General

Fieldcount
4

Quote character
"

Comment line
--

Import lines
1

Field1 (+0 -1)
Field2 (+0 -1) CATERING_ID
Field3 (+0 -1) NAME
Field4 (+0 -1) COSHER

Field Start
☐ Relative position
☐ Absolute position
☐ Character

Field End
☐ Length
☐ Character

Filter

Apply

Result Preview

	CATERING_ID	NAME	COSHER
1	30	Charlotte	N
2	46	Chubby	N

ImportImport to ScriptCloseSarith@XE Catering.csv loaded, 9 KBHelp

Data from TextfileData to Oracle

General

Owner
SARITH

Table
CATERING

Clear Table

Initializing Script

Commit every...
0

Overwrite duplicates
☐

Ignore duplicates
☒

Finalizing Script

Fields

Field1 ->
Field2 CATERING_ID -> CATERING_ID
Field3 NAME -> NAME
Field4 COSHER -> COSHER

Field
Fieldtype
SQL function

Field COSHER
Fieldtype String
Create SQL
additional Oracle processing, for example: substr@, 1, 20

Result Preview

	CATERING_ID	NAME	COSHER
1	30	Charlotte	N
2	46	Chubby	N

ImportImport to ScriptCloseSarith@XE Catering.csv loaded, 9 KBHelp

NameTypeCompiled

BELOFHTABLE20/05/2024 14:50:11

CATERINGTABLE20/05/2024 14:50:11

CONTRINTABLE20/05/2024 14:50:11

CUSTOMERTABLE20/05/2024 14:50:11

DISHCTABLE20/05/2024 14:50:11

MATERIALTABLE20/05/2024 14:50:11

THE_ORDERTABLE20/05/2024 14:50:11

WORKERTABLE20/05/2024 14:50:11

WORKINGINTABLE20/05/2024 14:50:11

Oracle Export: SQL Insets: PL/SQL Developer:

☐ Drop tables
☒ Create tables
☐ Truncate tables
☒ Disable triggers

☒ Disable foreign key constraints
☐ Include storage
☒ Include privileges
Commit every 100 records (0 = never)
Where clause

☐ Zip
Output file: C:\Users\emham\source\un1022 brm\backup\SQL.sql

Sarith@XEEExport

	Name	Type	Compiled
22	BELONG	TABLE	20/09/2024 14:50:11
23	CATERING	TABLE	20/09/2024 14:50:11
24	CARPON	TABLE	20/09/2024 14:50:11
25	CUSTOMER	TABLE	20/09/2024 14:50:11
26	DIGES	TABLE	20/09/2024 14:50:11
27	MARSHAL	TABLE	20/09/2024 14:50:11
28	THE_ORDER	TABLE	20/09/2024 14:50:11
29	WORKER	TABLE	20/09/2024 14:50:11
30	WORKINGON	TABLE	20/09/2024 14:50:11

Oracle Export

SQL Inserts

PL/SQL Developer

Log

☐ Drop tables

☒ Create tables

☐ Truncate tables

☒ Delete records

☒ Disable triggers

☒ Disable foreign key constraints

☐ Include storage

☒ Include privileges

Commit every 100 records (0 = never)

Where clause

☐ Zip

Output file C:\Users\מחשב\source\מסגרת\מסגרת\backupSQL.sql

Export

▼ Santh@XE