



צהרונים- דוח

מיני פרוייקט בסיסי נתונים

מיכל אלגאוי ושוהם שרווי

תוכן

3.....	שלב א
3.....	הקדמה
3.....	תרשים ERD
4.....	תרשים DSD
4.....	פקודות CREATE TABLE
5.....	פקודת DESC
6.....	הכנסת נתונים לטבלאות
13.....	גיבוי ושחזור
14.....	שלב ב
14.....	שאלות בחירה
14.....	בלי פרמטרים
18.....	עם פרמטרים
22.....	שאלות עדכון
25.....	שאלות מחיקה
27.....	אילוצים
29.....	שלב ג
29.....	תוכנית מספר 1
29.....	פונקציה
29.....	פרוצדורה
31.....	תוכנית ראשית
32.....	תוכנית מספר 2
32.....	פונקציה
33.....	פרוצדורה
33.....	תוכנית ראשית
36.....	שלב ד
37.....	תרשים ERD
38.....	תרשים DSD
39.....	תרשים ERD אחרי אינטגרציה
40.....	תרשים DSD אחרי אינטגרציה
40.....	החלטות
42.....	מבט 1
42.....	תאור מילולי

42.....	שליפת נתונים
43.....	מבט 2
43.....	תאור מילולי
44.....	שליפת נתונים

שלב א

הקדמה

ערכת לניהול צהרונים - DAYCARE

מערכת ניהול הצהרונים המציעה מעקב מקיף ואפשרויות ניהול מגוונות. המערכת מאפשרת איכות ויעילות בניהול הצהרון על ידי איחסון וניהול נתונים מגוונים.

המערכת מאפשרת ניהול של קבוצות ילדים- שכבת הגיל, גודל הקבוצה וכדומה.

מאפשרת שמירה על נתוני החוגים בקבוצות השונות, פרטי המדריכים, פרטי הגננות ופרטי הילדים.

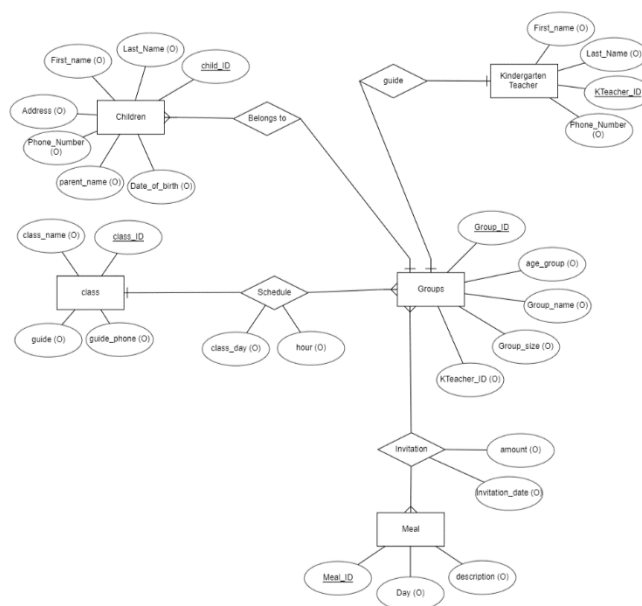
בנוסף מנהלת את רישום ההזמנות עבור המזון לצהרונים ופירוט המנות על בסיס יומי.

המערכת מספקת פונקציונליות של ניהול מורחבת ופרטים מקיפים של כל אוספי הצהרונים והילדים בקבוצות, כך שההנהלה תוכל לעקוב ולנהל ביעילות את הפעילויות והארועים במוסד.

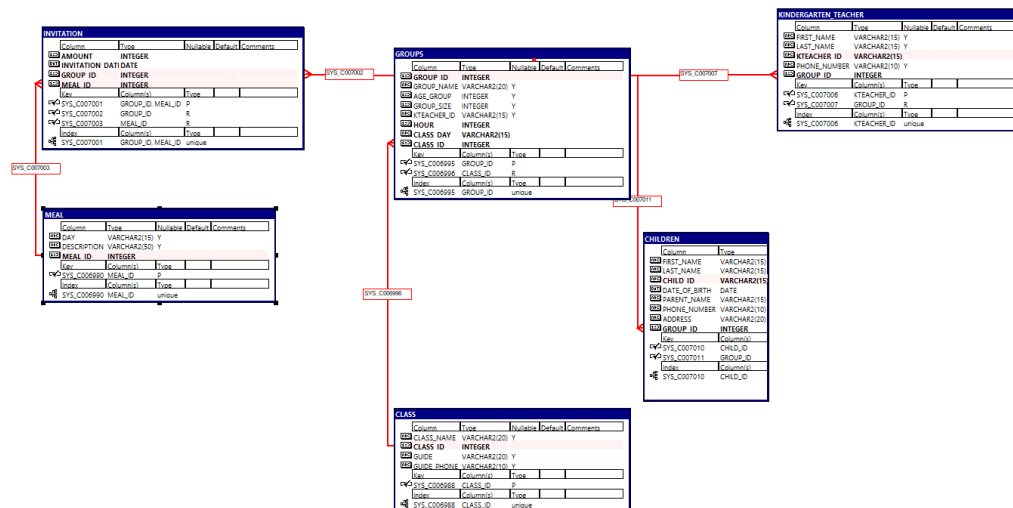
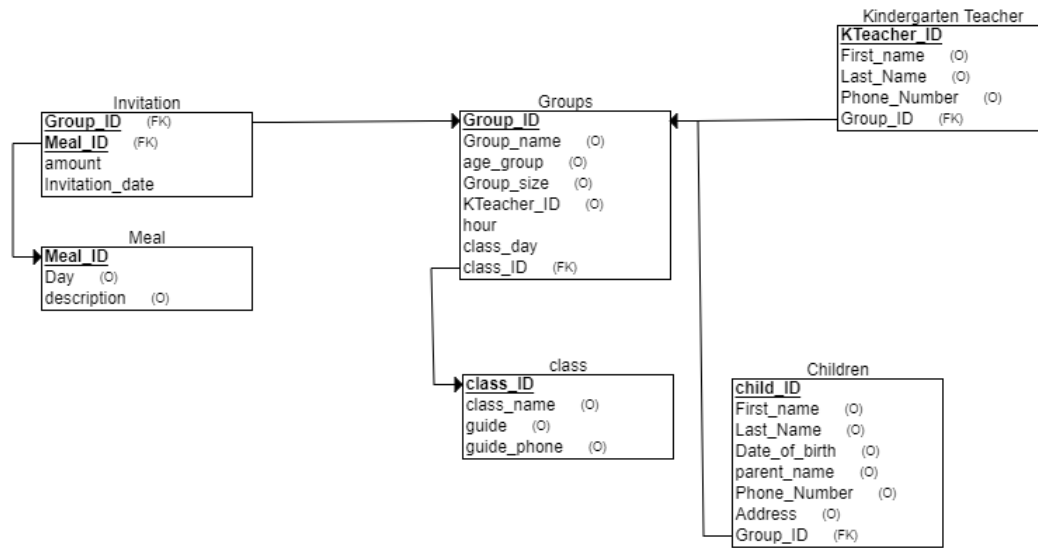
#הערה חשובה: בגלל הדרישה של 400 נתונים בטבלה נוצר עניין לא הגיוני בין הקבוצות- ילדים בקבוצה- הזמנות, וכן עניין של חוסר התאמה בין גילאים בקבוצה וגננות בתוך קבוצה (לכן השארתי חלק מהשדות כ NULL)

החל משלב 2 ביצעתי תיקון של הנתונים בצורה הגיונית על ידי שימוש בשאליות עדכון שעוד לא נדרשו בשלב 1 ועל ידי צמצום בסיס הנתונים בצורה הגיונית.

תרשים ERD



תרשים DSD



פקודות CREATE TABLE

CREATE TABLE class (class_name VARCHAR(40), class_ID INT NOT NULL, guide ;VARCHAR(20), guide_phone VARCHAR(15), PRIMARY KEY (class_ID))

CREATE TABLE Meal (Day VARCHAR(15), description VARCHAR(60), Meal_ID INT ;NOT NULL, PRIMARY KEY (Meal_ID))

CREATE TABLE Groups (Group_ID INT NOT NULL, Group_name VARCHAR(20), age_group INT, Group_size INT, KTeacher_ID VARCHAR(15), hour INT NOT NULL,

```
class_day VARCHAR(15) NOT NULL, class_ID INT NOT NULL, PRIMARY KEY
;(Group_ID), FOREIGN KEY (class_ID) REFERENCES class(class_ID) )
```

```
CREATE TABLE Invitation ( amount INT , Invitation_date DATE NOT NULL,
Group_ID INT NOT NULL, Meal_ID INT NOT NULL, PRIMARY KEY (Group_ID,
Meal_ID), FOREIGN KEY (Group_ID) REFERENCES Groups(Group_ID), FOREIGN
;KEY (Meal_ID) REFERENCES Meal(Meal_ID) )
```

```
CREATE TABLE Kindergarten_Teacher ( First_name VARCHAR(15), Last_Name
VARCHAR(15), KTeacher_ID VARCHAR(15) NOT NULL, Phone_Number
VARCHAR(10), Group_ID INT NOT NULL, PRIMARY KEY (KTeacher_ID), FOREIGN
;KEY (Group_ID) REFERENCES Groups(Group_ID) )
```

```
CREATE TABLE Children ( First_name VARCHAR(15), Last_Name VARCHAR(15),
child_ID VARCHAR(15) NOT NULL, Date_of_birth DATE, parent_name
VARCHAR(15), Phone_Number VARCHAR(10), Address VARCHAR(20), Group_ID
INT NOT NULL, PRIMARY KEY (child_ID), FOREIGN KEY (Group_ID) REFERENCES
;Groups(Group_ID) )
```

פקודת DESC

```
SQL> desc kindergarten_teacher
Name          Type          Nullable Default Comments
-----
FIRST_NAME    VARCHAR2(15)  Y
LAST_NAME     VARCHAR2(15)  Y
KTEACHER_ID   VARCHAR2(15)
PHONE_NUMBER  VARCHAR2(10)  Y
GROUP_ID      INTEGER

SQL> desc children
Name          Type          Nullable Default Comments
-----
FIRST_NAME    VARCHAR2(15)  Y
LAST_NAME     VARCHAR2(15)  Y
CHILD_ID      VARCHAR2(15)
DATE_OF_BIRTH DATE          Y
PARENT_NAME   VARCHAR2(15)  Y
PHONE_NUMBER  VARCHAR2(10)  Y
ADDRESS       VARCHAR2(20)  Y
GROUP_ID      INTEGER

SQL>
```

Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
 Connected as SYSTEM@XE

```
SQL> desc class
Name          Type          Nullable Default Comments
-----
CLASS_NAME    VARCHAR2(40)   Y
CLASS_ID      INTEGER
GUIDE         VARCHAR2(20)   Y
GUIDE_PHONE   VARCHAR2(15)   Y
```

```
SQL> desc meal
Name          Type          Nullable Default Comments
-----
DAY           VARCHAR2(15)   Y
DESCRIPTION   VARCHAR2(60)   Y
MEAL_ID       INTEGER
```

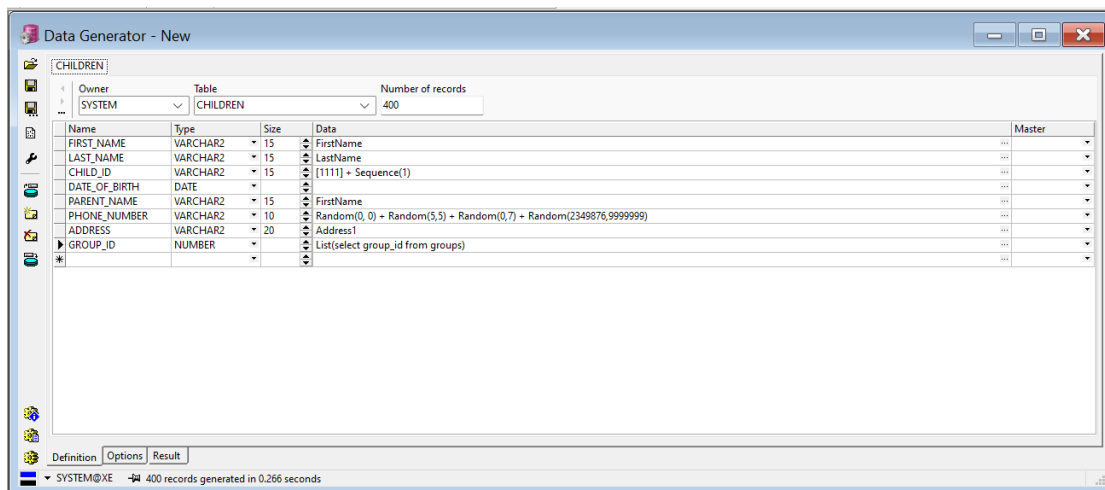
```
SQL> desc groups
Name          Type          Nullable Default Comments
-----
GROUP_ID      INTEGER
GROUP_NAME    VARCHAR2(20)   Y
AGE_GROUP     INTEGER        Y
GROUP_SIZE    INTEGER        Y
KTEACHER_ID   VARCHAR2(15)   Y
HOUR          INTEGER
CLASS_DAY     VARCHAR2(15)
CLASS_ID      INTEGER
```

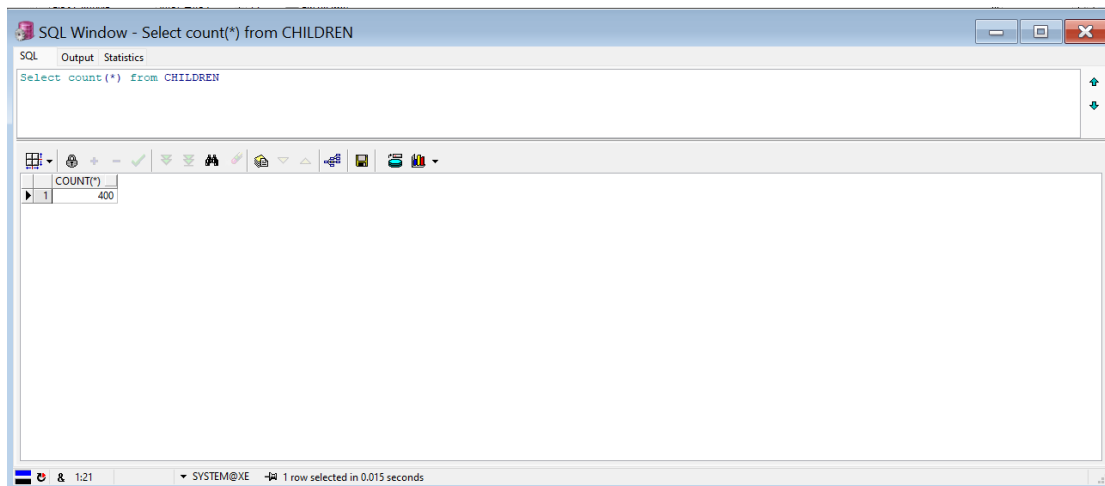
```
SQL> desc invitation
Name          Type          Nullable Default Comments
-----
AMOUNT        INTEGER        Y
INVITATION_DATE DATE
GROUP_ID      INTEGER
MEAL_ID       INTEGER
```

הכנסת נתונים לטבלאות

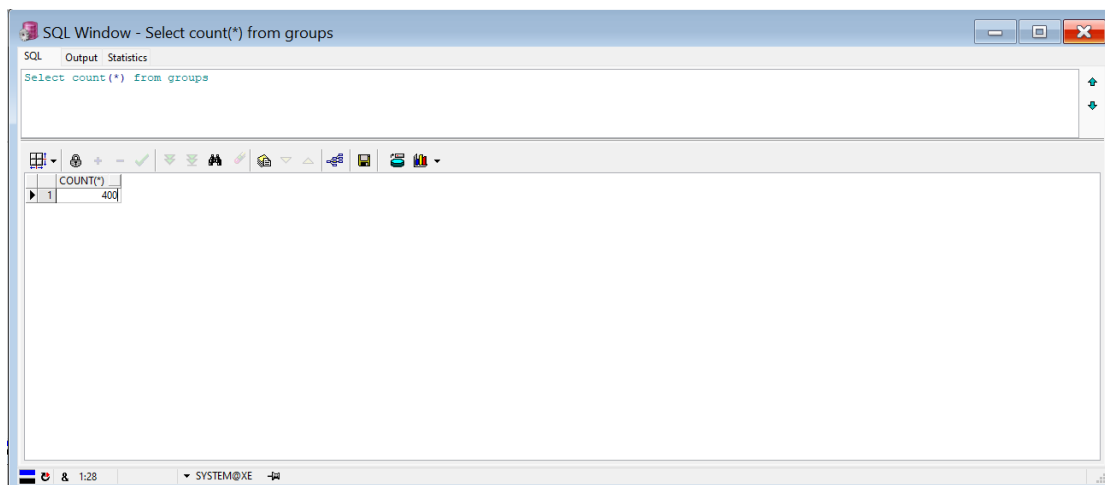
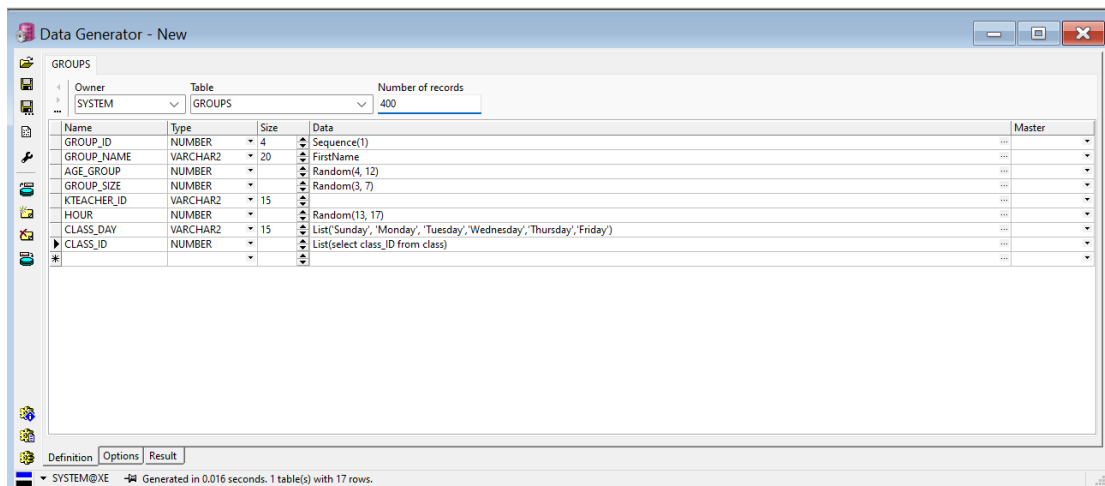
DATA GENERATOR

הטבלה-CHILDREN

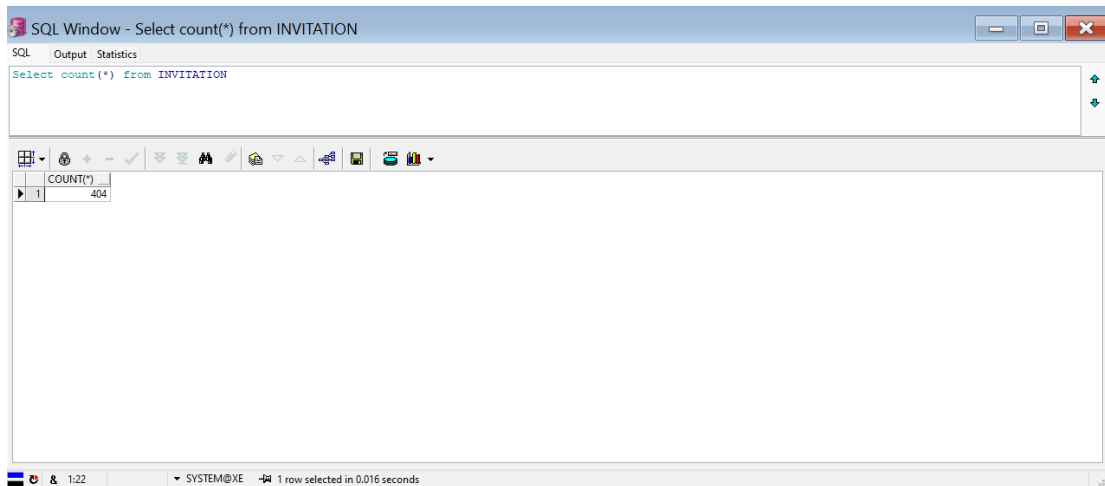
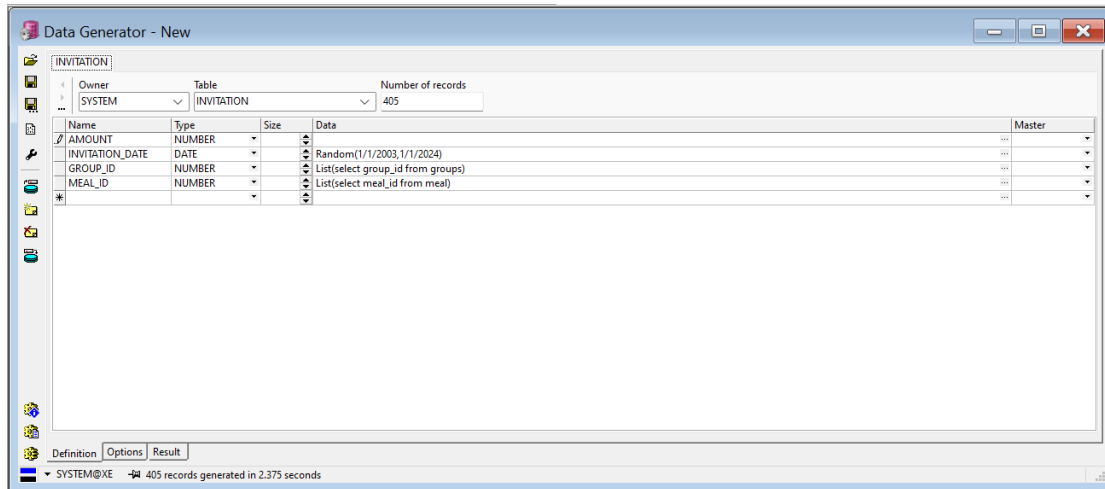




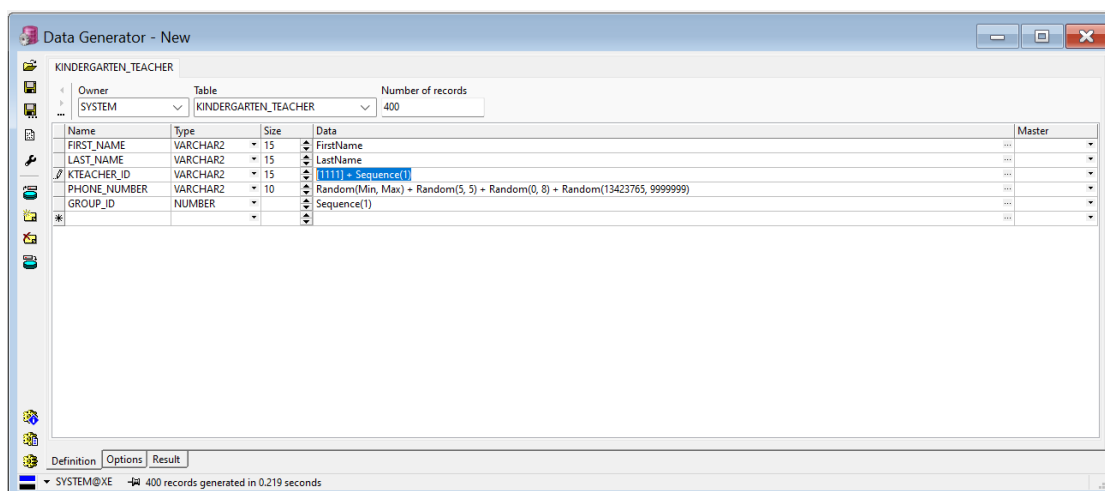
הטבלה-GROUPS

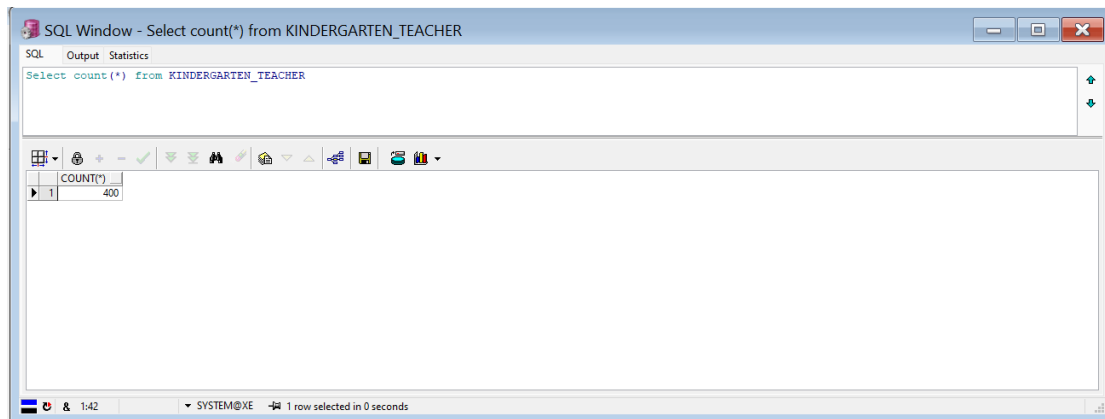


הטבלה INVITATIONS-

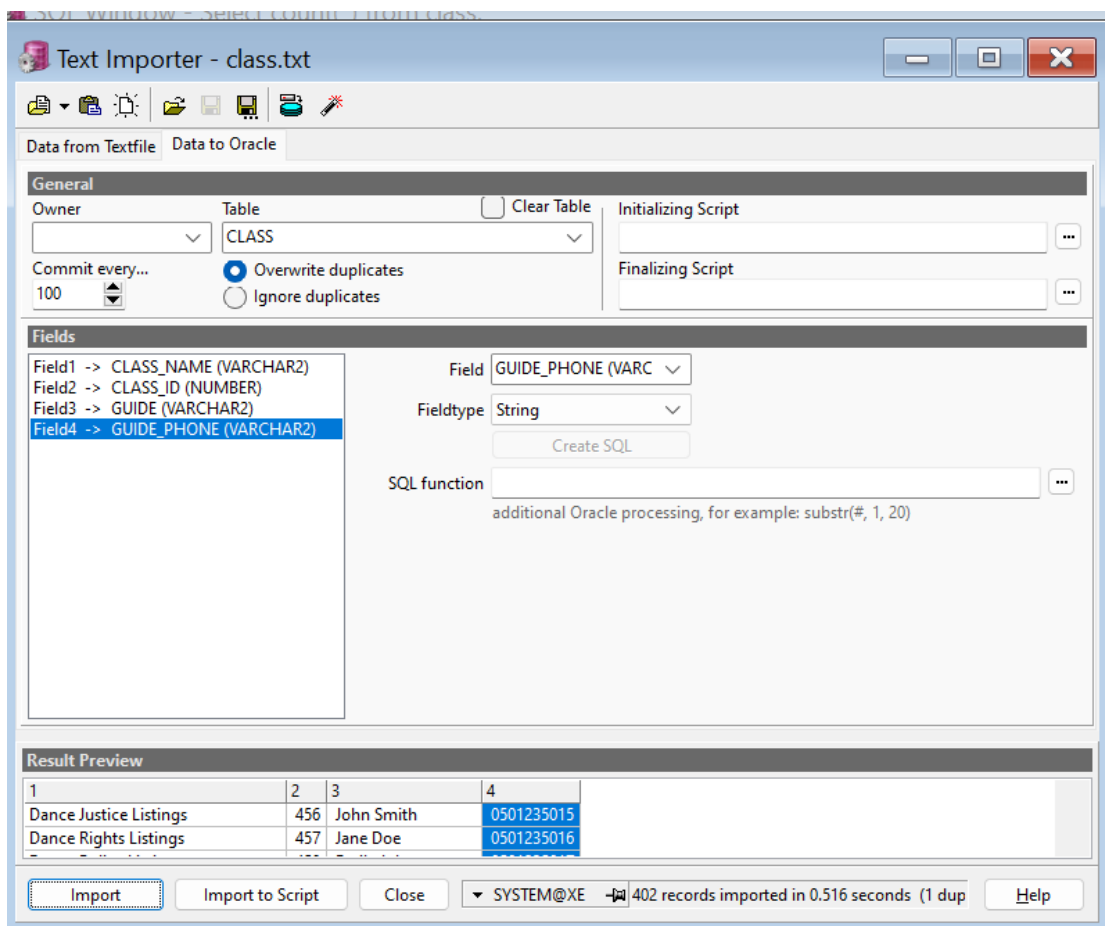


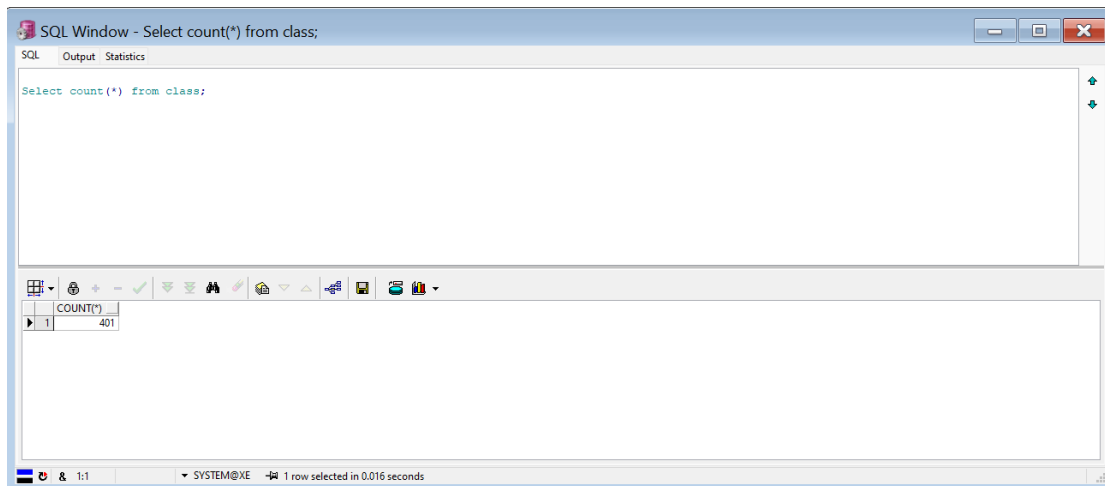
הטבלה-Kindergarten_Teacher





קובץ טקסט
הטבלה CLASS





קוד בפיתון

הטבלה

MEAL-

```
C: > Users > M0538 > Desktop > stageA > Programing > meal.py > ...
1 import random
2
3 FOOD_FILE = "C:/Users/M0538/minip/FOOD.txt"
4 OUTPUT_FILE = "C:/Users/M0538/minip/meal_schedule.txt"
5
6 DAYS_OF_WEEK = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday"]
7
8 def create_meal_schedule(food_file, output_file):
9
10     with open(food_file, "r", encoding="utf-8") as file:
11         foods = file.readlines()
12
13         foods = [food.strip() for food in foods if food.strip()]
14
15         random.shuffle(foods)
16
17         with open(output_file, "w", encoding="utf-8") as file:
18
19             for i, food in enumerate(foods, start=1):
20                 day = random.choice(DAYS_OF_WEEK)
21                 file.write(f"{day}, {food}, {i}\n")
22
23 if __name__ == "__main__":
24     create_meal_schedule(FOOD_FILE, OUTPUT_FILE)
25
```

SQL WINDOW - CREATE TABLE MEAL (DAY VARCHAR(2), DESCRIPTION VARCHAR(100), MEAL_ID INT(1))

Text Importer - meal_schedule.txt

Data from Textfile | Data to Oracle

General

Owner: Table: MEAL ☐ Clear Table

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

Initializing Script: ...

Finalizing Script: ...

Fields

Field1 -> DAY (VARCHAR2)
Field2 -> DESCRIPTION (VARCHAR2)
Field3 -> MEAL_ID (NUMBER)

Field: MEAL_ID (NUMBER)
Fieldtype: Number

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

Result Preview

1	2	3
Monday	Chicken and cheese sliders	1
Friday	Rice cakes with tuna salad	2

SYSTEM@XE 230 records imported in 0.172 seconds

SQL Window - Select count(*) from meal

SQL Output Statistics

```
Select count(*) from meal
```

COUNT(*)
230

SYSTEM@XE 1 row selected in 0.015 seconds

C: > Users > M0538 > Desktop > stageA > Programing > ≡ FOOD.txt

```
1  Grilled cheese sandwich
2  Chicken nuggets
3  Peanut butter and jelly sandwich
4  Fruit salad
5  Veggie sticks with hummus
6  Macaroni and cheese
7  Quesadillas
8  Cheese and crackers
9  Mini pizzas
10 Turkey and cheese roll-ups
11 Pasta salad
12 Tuna salad sandwich
13 Cheese quesadillas
14 Egg salad sandwich
15 Chicken salad sandwich
16 Ham and cheese sandwich
17 Yogurt with granola
18 Smoothies with fruit and yogurt
19 English muffin pizzas
20 Turkey and avocado wraps
21 Bagel with cream cheese
22 Rice cakes with almond butte
23 Ants on a log (celery with peanut butter and raisins)
24 Cheese and tomato skewers
25 Mini muffins (banana, blueberry, etc.)
26 Waffle sticks with syrup for dipping
27 Cheese and spinach triangles
28 Turkey and cranberry wrap
29 Mini meatballs
```

Friday: Cheese and ham baguette slices 1
 Thursday: Cheese and ham salad wraps 2
 Tuesday: Cheese quesadillas 3
 Monday: Turkey and cranberry croissants 4
 Tuesday: Cheese and ham crescent rolls 5
 Monday: Turkey and cranberry pita pockets 6
 Monday: Cheese and ham croissants 7
 Friday: Veggie and cream cheese crostini 8
 Tuesday: Cheese and tomato bagels 9
 Monday: Veggie and cream cheese baguette rolls 10
 Friday: Cheese and ham wraps 11
 Friday: Veggie and cream cheese salad wraps 12
 Monday: Mini chicken paninis 13
 Thursday: Chicken and avocado wraps 14
 Thursday: Chicken and vegetable spring rolls 15
 Sunday: Chicken and vegetable skewers 16
 Monday: Cheese and tomato pita bread 17
 Sunday: Turkey and cranberry pinwheels 18
 Thursday: Turkey and cranberry salad pitas 19
 Tuesday: Chicken and vegetable quesadillas 20
 Friday: Grilled cheese sandwich 21
 Thursday: Mini chicken sandwiches 22
 Friday: Veggie and cream cheese pinwheels 23
 Friday: Veggie and cream cheese pinwheels 24
 Monday: Cheese and ham baguette rolls 25
 Tuesday: Chicken and cheese sliders 26
 Friday: Cucumber sandwiches 27
 Thursday: Smoothies with fruit and yogurt 28
 Tuesday: Turkey and cranberry salad slices 29

גיבוי ושחזור

ביצוע גיבוי:

Name	Type	Compiled
LOGMNR_TABSUBPARTS	TABLE	27/08/2011 08:24:20
LOGMNR_TSS	TABLE	27/08/2011 08:24:20
LOGMNR_TYPES	TABLE	27/08/2011 08:24:20
LOGMNR_UIDS	TABLE	27/08/2011 08:24:20
LOGMNR_USERS	TABLE	27/08/2011 08:24:20
LOGSTDBYAPPLY_MILESTONE	TABLE	27/08/2011 08:26:18
LOGSTDBYAPPLY_PROGRESS	TABLE	27/08/2011 08:26:18
LOGSTDBYEVENTS	TABLE	27/08/2011 08:26:18
LOGSTDBYFLASHBACK_SCN	TABLE	27/08/2011 08:26:18
LOGSTDBYPARAMETERS	TABLE	27/08/2011 08:26:18
LOGSTDBYVPSQL	TABLE	27/08/2011 08:26:18
LOGSTDBYVSSCN	TABLE	27/08/2011 08:26:18
LOGSTDBYVSSKIP	TABLE	27/08/2011 08:26:18
LOGSTDBYVSSKIP_TRANSACTION	TABLE	27/08/2011 08:26:18
LOGSTDBYVSSKIP_TABLES	TABLE	27/08/2011 08:26:19
LOGSTDBYVSHISTORY	TABLE	27/08/2011 08:26:19
LOGSTDBYVSSKIP_SUPPORT	TABLE	27/08/2011 08:26:19
DEFS_PROPAGATOR	TABLE	27/08/2011 08:32:57
DEFS_ACEERROR	TABLE	27/08/2011 08:33:09
DEFS_AQCALL	TABLE	27/08/2011 08:33:09
DEFS_CALLDEST	TABLE	27/08/2011 08:33:09
DEFS_DESTINATION	TABLE	27/08/2011 08:33:09
DEFS_ERROR	TABLE	27/08/2011 08:33:09
REPCATS_REPCATALOG	TABLE	27/08/2011 08:33:09
REPCATS_REPPROP	TABLE	27/08/2011 08:33:09
REPCATS_REPSHEMA	TABLE	27/08/2011 08:33:09
SQLPLUS_PRODUCT_PROFILE	TABLE	27/08/2011 08:36:13
HELP	TABLE	27/08/2011 08:36:18
CLASS	TABLE	02/06/2024 02:35:47
CHILDREN	TABLE	02/06/2024 03:38:29
GROUPS	TABLE	02/06/2024 03:38:29
MEAL	TABLE	02/06/2024 03:38:29
KINDERGARTEN_TEACHER	TABLE	02/06/2024 04:54:50
INVITATION	TABLE	02/06/2024 05:09:42

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

Output file: C:\Users\MO338\minipl\backup1.sql

SYSTEM@XE Exporting... Done

ביצוע שחזור לאחר מחיקת הטבלאות :

PL/SQL Developer - SYSTEM@XE : [SQL Window - select * from class; select * from meal; select * from groups; select * from invitation; select ...]

SQL Window - select * from class; select * from meal; select * from groups; select * from invitation; select ...

```

select * from class;
select * from meal;
select * from groups;
select * from invitation;
select * from kindergarten_teacher;
select * from children;

```

CLASS_NAME	CLASS_ID	NAME	PHONE
Dance Justice Listings	456	John Smith	0901230910
Dance Rights Listings	457	Jane Doe	0901230915
Dance Policy Listings	458	Emily Johnson	0901230917
Dance Legislation Listings	459	Michael Brown	0901230918
Dance Advocacy Group Listings	460	Jessica Davis	0901230919
Dance Union Listings	461	Daniel Garcia	0901230920
Dance Association Listings	462	Sarah Martinez	0901230921
Dance Organization Listings	463	David Hernandez	0901230922
Dance Conference Listings	464	Ashley Lopez	0901230923
Dance Symposium Listings	465	Christopher Wilson	0901230924
Dance Journal Listings	466	Amanda Lewis	0901230925
Dance Magazine Listings	467	Joshua Lee	0901230926
Dance Blog Listings	468	Megan Walker	0901230927
Dance Vlog Listings	469	Andrew Hall	0901230928
Dance Podcast Listings	470	Elizabeth Young	0901230929
Dance Website Listings	471	Ryan King	0901230930
Dance App Listings	472	Nicole Wright	0901230931
Dance Game Listings	473	Brandon Scott	0901230932
Dance Software Listings	474	Samantha Green	0901230933
Dance Equipment Listings	475	Matthew Adams	0901230934
Dance Supply Listings	476	Lauren Baker	0901230935
Dance Costume Listings	477	Tyler Gonzalez	0901230936
Dance Shoe Listings	478	Olivia Perez	0901230937
Dance Accessory Listings	479	James Roberts	0901230938
Dance Prep Listings	480	Sophia Thompson	0901230939
Dance Sock Listings	481	Alexander Rivera	0901230940
Dance Lighting Listings	482	Ava Carter	0901230941
Dance Sound Listings	483	Daniel Torres	0901230942

Find: 401 rows selected in 2.001 seconds

שלב ב

שאלות בחירה

בלי פרמטרים

(1

מספר ארוחות שהזמנו בחודש מסוים עבור כל קבוצה

```

SELECT

,G.Group_name AS Group_Name

,EXTRACT(MONTH FROM I.Invitation_date) AS Month

SUM(I.amount) AS Total_Meals_Ordered

FROM

Invitation I

JOIN Groups G ON I.Group_ID = G.Group_ID

GROUP BY

G.Group_name, EXTRACT(MONTH FROM I.Invitation_date)

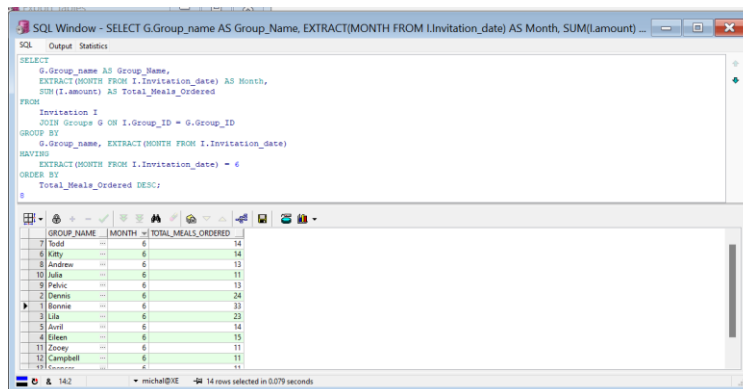
HAVING

EXTRACT(MONTH FROM I.Invitation_date) = 6

ORDER BY

;Total_Meals_Ordered DESC

```



	GROUP_NAME	MONTH	TOTAL_MEALS_ORDERED
7	Todd	6	14
6	Kitty	6	14
8	Andrew	6	13
10	Julia	6	11
9	Patric	6	11
7	Dennis	6	10
1	Bonnie	6	33
3	Lila	6	23
5	April	6	14
4	Eileen	6	15
11	Dorey	6	11
12	Campbell	6	11
13	Connors	6	11

(2

פרטי ההורים וכתובת של ילדים שיום ההולדת שלהם בחודש הנוכחי

```

SELECT

,C.First_name AS Child_First_Name

,C.Last_Name AS Child_Last_Name

,C.Date_of_birth AS Birth_Date

,C.parent_name AS Parent_Name

```


C.Address AS Address

FROM

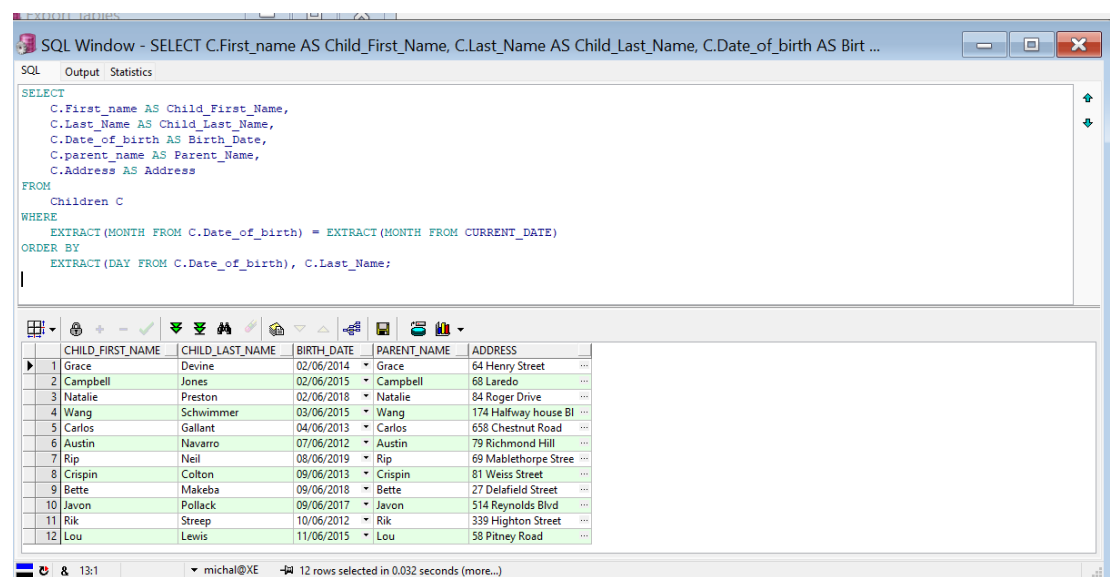
Children C

WHERE

EXTRACT(MONTH FROM C.Date_of_birth) = EXTRACT(MONTH FROM
CURRENT_DATE)

ORDER BY

;EXTRACT(DAY FROM C.Date_of_birth), C.Last_Name



The screenshot shows a SQL window titled "SQL Window - SELECT C.First_name AS Child_First_Name, C.Last_Name AS Child_Last_Name, C.Date_of_birth AS Birt ...". The query is as follows:

```
SELECT
  C.First_name AS Child_First_Name,
  C.Last_Name AS Child_Last_Name,
  C.Date_of_birth AS Birth_Date,
  C.parent_name AS Parent_Name,
  C.Address AS Address
FROM
  Children C
WHERE
  EXTRACT(MONTH FROM C.Date_of_birth) = EXTRACT(MONTH FROM CURRENT_DATE)
ORDER BY
  EXTRACT(DAY FROM C.Date_of_birth), C.Last_Name;
```

The results are displayed in a table with 12 rows and 5 columns: CHILD_FIRST_NAME, CHILD_LAST_NAME, BIRTH_DATE, PARENT_NAME, and ADDRESS. The data is as follows:

	CHILD_FIRST_NAME	CHILD_LAST_NAME	BIRTH_DATE	PARENT_NAME	ADDRESS
1	Grace	Devine	02/06/2014	Grace	64 Henry Street
2	Campbell	Jones	02/06/2015	Campbell	68 Laredo
3	Natalie	Preston	02/06/2018	Natalie	84 Roger Drive
4	Wang	Schwimmer	03/06/2015	Wang	174 Halfway house Bl
5	Carlos	Gallant	04/06/2013	Carlos	658 Chestnut Road
6	Austin	Navarro	07/06/2012	Austin	79 Richmond Hill
7	Rip	Neil	08/06/2019	Rip	69 Mablethorpe Stree
8	Crispin	Colton	09/06/2013	Crispin	81 Weiss Street
9	Bette	Makeba	09/06/2018	Bette	27 Delafield Street
10	Javon	Pollack	09/06/2017	Javon	514 Reynolds Blvd
11	Rik	Streep	10/06/2012	Rik	339 Highton Street
12	Lou	Lewis	11/06/2015	Lou	58 Pitney Road

(3

סך מספר הילדים בכל קבוצה ומידע על הגננת

SELECT

,G.Group_name

,KT.First_name || ' ' || KT.Last_Name AS KTeacher

COUNT(C.child_ID) AS Number_of_Children

FROM

Groups G

JOIN Kindergarten_Teacher KT ON G.KTeacher_ID = KT.KTeacher_ID

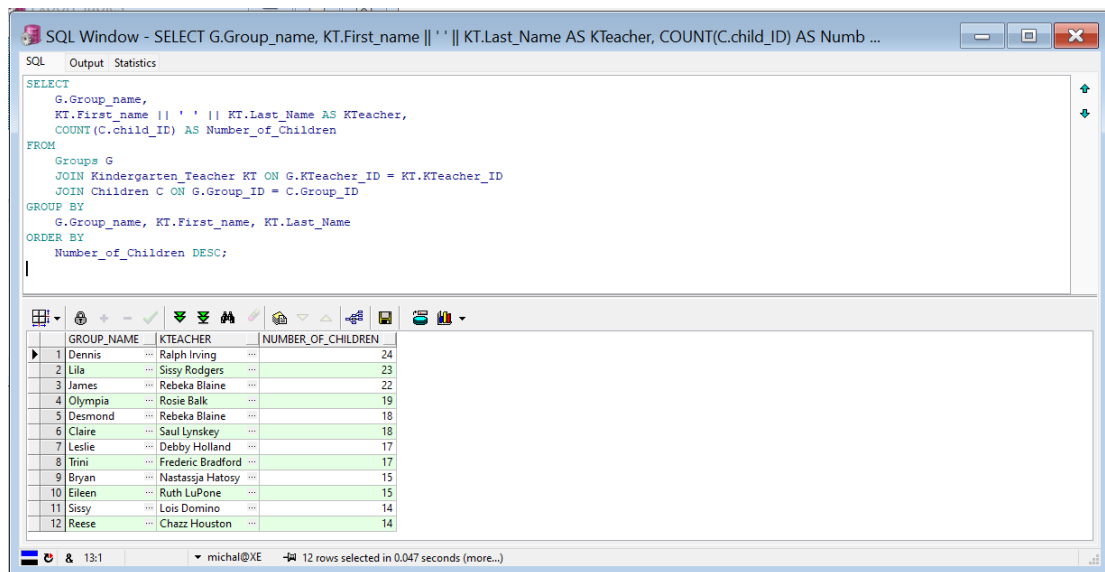
JOIN Children C ON G.Group_ID = C.Group_ID

GROUP BY

G.Group_name, KT.First_name, KT.Last_Name

ORDER BY

;Number_of_Children DESC



The screenshot shows an SQL Window with the following query:

```
SELECT
  G.Group_name,
  KT.First_name || ' ' || KT.Last_Name AS KTeacher,
  COUNT(C.child_ID) AS Number_of_Children
FROM
  Groups G
  JOIN Kindergarten_Teacher KT ON G.KTeacher_ID = KT.KTeacher_ID
  JOIN Children C ON G.Group_ID = C.Group_ID
GROUP BY
  G.Group_name, KT.First_name, KT.Last_Name
ORDER BY
  Number_of_Children DESC;
```

The results are displayed in a table with the following columns: GROUP_NAME, KTEACHER, and NUMBER_OF_CHILDREN. The table contains 12 rows of data, sorted by the number of children in descending order.

	GROUP_NAME	KTEACHER	NUMBER_OF_CHILDREN
1	Dennis	Ralph Irving	24
2	Lila	Sissy Rodgers	23
3	James	Rebeka Blaine	22
4	Olympia	Rosie Balk	19
5	Desmond	Rebeka Blaine	18
6	Claire	Saul Lynskey	18
7	Leslie	Debby Holland	17
8	Trini	Frederic Bradford	17
9	Bryan	Nastassja Hatossy	15
10	Eileen	Ruth LuPone	15
11	Sissy	Lois Domino	14
12	Reese	Chazz Houston	14

The status bar at the bottom indicates that 12 rows were selected in 0.047 seconds.

(4

מנות מוזמנות לפי תאריכים ותיאורי מנות

SELECT

,I.Invitation_date

,M.description

SUM(I.amount) AS Total_Amount

FROM

Invitation I

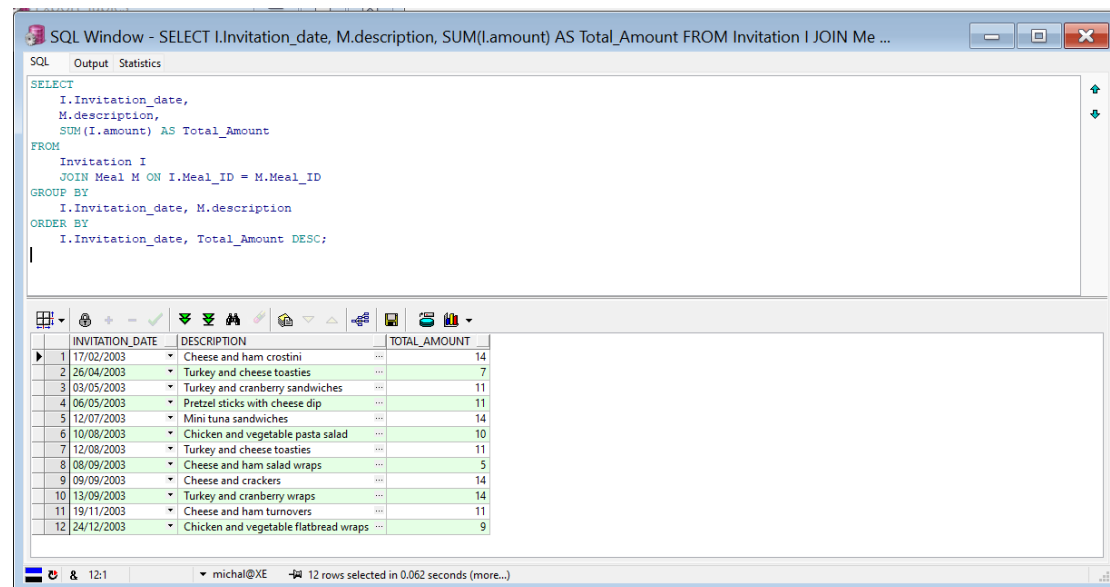
JOIN Meal M ON I.Meal_ID = M.Meal_ID

GROUP BY

I.Invitation_date, M.description

ORDER BY

;I.Invitation_date, Total_Amount DESC



The screenshot shows a SQL Window with the following query:

```
SELECT
  I.Invitation_date,
  M.description,
  SUM(I.amount) AS Total_Amount
FROM
  Invitation I
JOIN Meal M ON I.Meal_ID = M.Meal_ID
GROUP BY
  I.Invitation_date, M.description
ORDER BY
  I.Invitation_date, Total_Amount DESC;
```

The results are displayed in a table with the following data:

INVITATION_DATE	DESCRIPTION	TOTAL_AMOUNT
17/02/2003	Cheese and ham crostini	14
26/04/2003	Turkey and cheese toasties	7
03/05/2003	Turkey and cranberry sandwiches	11
06/05/2003	Pretzel sticks with cheese dip	11
12/07/2003	Mini tuna sandwiches	14
10/08/2003	Chicken and vegetable pasta salad	10
12/08/2003	Turkey and cheese toasties	11
08/09/2003	Cheese and ham salad wraps	5
09/09/2003	Cheese and crackers	14
13/09/2003	Turkey and cranberry wraps	14
19/11/2003	Cheese and ham turnovers	11
24/12/2003	Chicken and vegetable flatbread wraps	9

עם פרמטרים

(1) הזמנות בשנה מסוימת -

מטרה: שאילתה זו מחזירה מידע על הזמנות שנעשו בשנה מסוימת.

עמודות שנבחרו: תאריך ההזמנה, כמות, שם הקבוצה, ותיאור הארוחה.

טבלאות: הטבלאות שמשתתפות הן Invitation, Groups ו-Meal.

תנאי: השאילתה מסננת לפי שנה של תאריך ההזמנה באמצעות פונקציית EXTRACT.

פרמטר:

invitation_year: השנה של ההזמנות (נכנס כפרמטר).

SQL Window - SELECT i.Invitation_date, i.amount, g.Group_name, m.description FROM Invitation i JOIN Groups g ...

```

SELECT i.Invitation_date, i.amount, g.Group_name, m.description
FROM Invitation i
JOIN Groups g ON i.Group_ID = g.Group_ID
JOIN Meal m ON i.Meal_ID = m.Meal_ID
WHERE EXTRACT(YEAR FROM i.Invitation_date) = &name = "invitation_year" type = "integer" hint = "Enter the year of the invitations">
ORDER BY i.Invitation_date, g.Group_name;

```

Variables

Name	Value
invitation_year	2023

Enter the year of the invitations

	INVITATION_DATE	AMOUNT	GROUP_NAME	DESCRIPTION
1	10/01/2023	9	Spike	Veggie and cream cheese puff pastry rolls
2	20/02/2023	6	Sara	Cheese and tomato salad rolls
3	04/03/2023	12	Ben	Mini vegetable burgers
4	09/03/2023	14	Kitty	Veggie and hummus wraps
5	03/04/2023	18	Desmond	Cheese and ham turnovers

(2)

שאלתה: מידע מפורט על ילדים בקבוצת גיל מסוימת -

מטרה: שאלתה זו מחזירה מידע מפורט על ילדים בקבוצת גיל מסוימת.

עמודות שנבחרו: שם פרטי, שם משפחה, תאריך לידה, שם הורה, מספר טלפון, כתובת, ושם הקבוצה.

טבלאות: הטבלאות שמתפתחות הן Children ו-Groups.

תנאי: השאלתה מסוננת לפי קבוצת גיל של הילדים.

פרמטר:

age_group: קבוצת הגיל של הילדים (נכנס כפרמטר).

SQL Window - SELECT i.Invitation_date, i.amount, g.Group_name, m.description FROM Invitation i JOIN Groups g ...

```

SELECT ch.First_name, ch.Last_Name, ch.Date_of_birth, ch.parent_name, ch.Phone_Number, ch.Address, g.Group_name
FROM Children ch
JOIN Groups g ON ch.Group_ID = g.Group_ID
WHERE g.age_group = &<name = "age_group" type = "integer" hint = "Enter the age group">
ORDER BY ch.Last_name, ch.First_name;

```

Variables

Name	Value
age_group	3

OK Cancel Clear

Enter the age group

	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	PARENT_NAME	PHONE_NUMBER	ADDRESS	GROUP_NAME
1	Trini	Alda	23/05/2020	Trini	0529125257	379 Penn Drive	Rebecca
2	Denise	Ali	14/06/2019	Denise	0515125229	100 Lecanto Ave	Kyra
3	Grace	Arquette	03/08/2012	Grace	0514830268	824 Emily Road	Rebecca
4	Vincent	Assante	30/03/2018	Vincent	0543285763	73 Kylie Road	Bryan
5	Terence	Austin	09/11/2019	Terence	0569969909	33 Trejo Street	Kurtwood
6	Robby	Bale	11/07/2019	Robby	0546048357	13 Annandale Blvd	Famke
7	Shawn	Begley	28/05/2012	Shawn	0563115425	14 Gates Ave	Bryan
8	Avenged	Belushi	13/03/2017	Avenged	0512669668	73rd Street	Famke
9	Buffy	Benson	18/12/2017	Buffy	0508018933	43rd Street	Bryan
10	Tramaine	Berkoff	24/06/2020	Tramaine	0548104582	229 LaMond Ave	Kyra
11	Lara	Boothe	05/12/2021	Lara	0518353483	57 Utsunomiya Ave	Rory
12	Patrick	Branagh	14/10/2013	Patrick	0557094095	36 Birkenhead Ave	Famke
13	Holly	Carmen	20/01/2021	Holly	0544162690	13 Whitley Drive	Bryan
14	Vince	Carrere	24/10/2015	Vince	0575359946	35 Winans Ave	Sander
15	Humberto	Cazale	11/01/2021	Humberto	0566595071	493 Dafoe Street	Sander
16	Marina	Cetera	30/08/2018	Marina	0525778327	53 Rooker Road	Kurtwood
17	Ming-Na	Child	14/10/2019	Ming-Na	0515731046	337 Maxwell Drive	Kurtwood
18	Shannyn	Choice	08/04/2017	Shannyn	0558581921	35 Pelvic Road	Rory

(3

שאלתה: שלפת כל הקבוצות לפי יום מסוים וקבוצת גיל -

מטרה: שאלתה זו מחזירה מידע על קבוצות לפי יום מסוים וקבוצת גיל.

עמודות שנבחרו: מזהה קבוצה, שם קבוצה, יום שיעור, קבוצת גיל ושם מדריך.

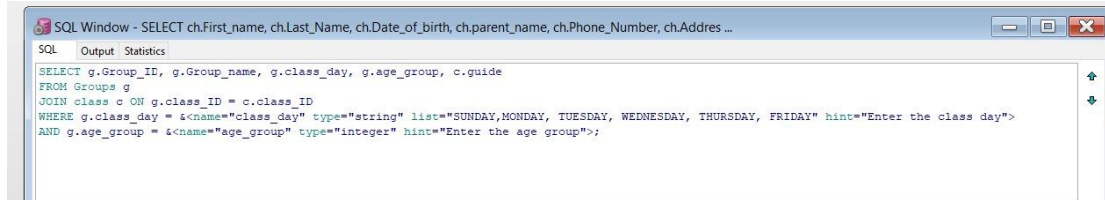
טבלאות: הטבלאות שמשתתפות הן Groups ו-class.

תנאי: השאלתה מסננת לפי יום השיעור וקבוצת הגיל של הקבוצות.

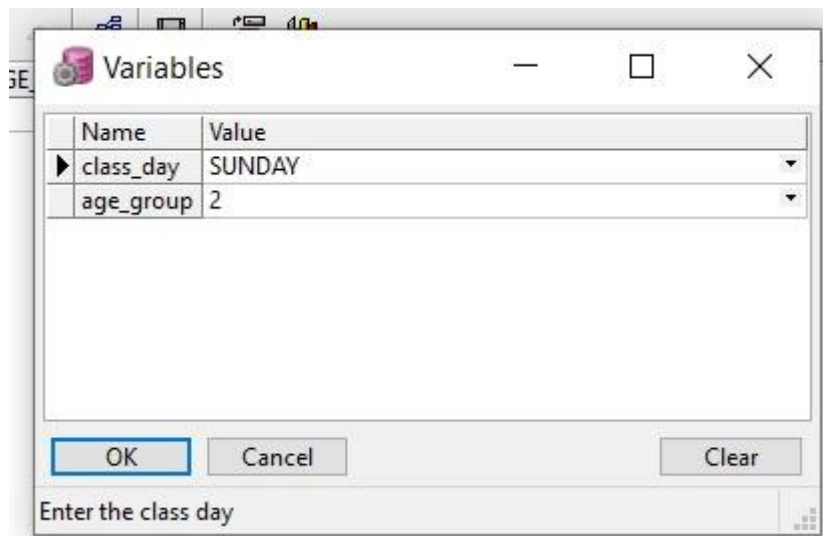
פרמטרים:

class_day: יום השיעור (נכנס כפרמטר).

age_group: קבוצת הגיל של הקבוצות (נכנס כפרמטר).



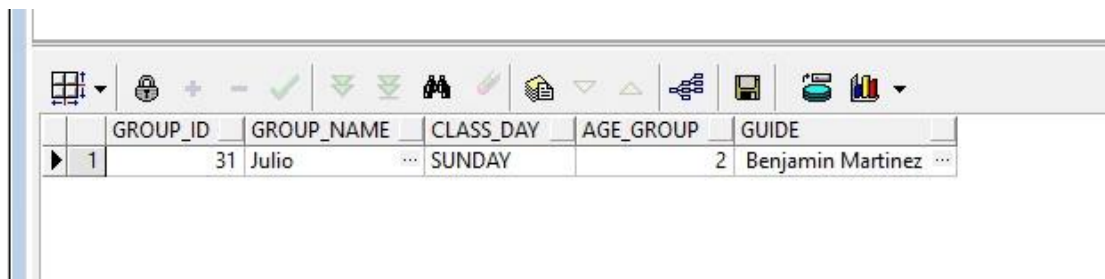
```
SQL Window - SELECT ch.First_name, ch.Last_Name, ch.Date_of_birth, ch.parent_name, ch.Phone_Number, ch.Address ...
SQL Output Statistics
SELECT g.Group_ID, g.Group_name, g.class_day, g.age_group, c.guide
FROM Groups g
JOIN class c ON g.class_ID = c.class_ID
WHERE g.class_day = :<name="class_day" type="string" list="SUNDAY,MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY" hint="Enter the class day">
AND g.age_group = :<name="age_group" type="integer" hint="Enter the age group">;
```



Name	Value
class_day	SUNDAY
age_group	2

OK Cancel Clear

Enter the class day



	GROUP_ID	GROUP_NAME	CLASS_DAY	AGE_GROUP	GUIDE
▶ 1	31	Julio	SUNDAY	2	Benjamin Martinez

(4

שאלתה: שליפת כל ההזמנות משנת מסוימת עם קבוצת גיל מסוימת -

מטרה: שאלתה זו מחזירה מידע על הזמנות שנעשו בשנה מסוימת עם קבוצת גיל מסוימת.

עמודות שנבחרו: תאריך ההזמנה, שם הקבוצה, תיאור הארוחה וכמות ההזמנות.

טבלאות: הטבלאות שמשתתפות הן Invitation, Groups ו-Meal.

תנאי: השאלתה מסננת לפי שנה של תאריך ההזמנה וקבוצת גיל.

פרמטרים:

year: השנה של ההזמנות (נכנס כפרמטר).

age_group: קבוצת הגיל של הקבוצות (נכנס כפרמטר).

```
SQL Window - SELECT g.Group_ID, g.Group_name, g.class_day, g.age_group, c.guid FROM Groups g JOIN class c ON ...
SQL Output Statistics
SELECT i.Invitation_date, g.Group_name, m.description AS Meal_Description, i.amount AS Number_Of_Invitations
FROM Invitation i
JOIN Groups g ON i.Group_ID = g.Group_ID
JOIN Meal m ON i.Meal_ID = m.Meal_ID
WHERE EXTRACT(YEAR FROM i.Invitation_date) = <name="year" type="integer" hint="Enter the year">
AND g.age_group = <name="age_group" type="integer" hint="Enter the age group">;
```

Variables

Name	Value
year	2020
age_group	2

Enter the year

	INVITATION_DATE	GROUP_NAME	MEAL_DESCRIPTION	NUMBER_OF_INVITATIONS
1	26/08/2020	Rawlins	Cheese and spinach stuffed chicken thighs	5
2	22/08/2020	Johnette	Tortilla chips with salsa and guacamole	12

שאלות עדכון

(1)

עדכון מס' המנות בהזמנה לפי גודל הקבוצה בפועל

UPDATE Invitation

) = SET amount

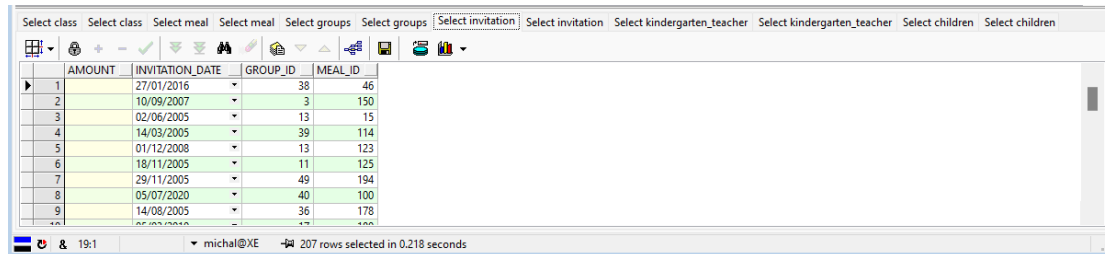
(*)SELECT COUNT

FROM Children

WHERE Children.Group_ID = Invitation.Group_ID

;(

לפני הרצת השאילתה:



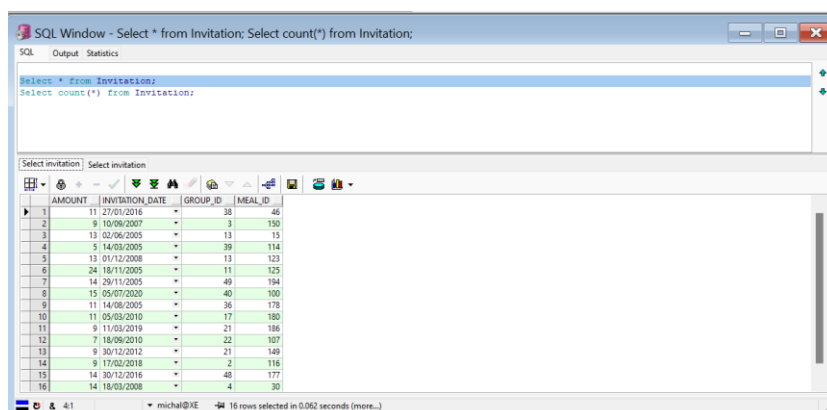
	AMOUNT	INVITATION_DATE	GROUP_ID	MEAL_ID
1		27/01/2016	38	46
2		10/09/2007	3	150
3		02/06/2005	13	15
4		14/03/2005	39	114
5		01/12/2008	13	123
6		18/11/2005	11	125
7		29/11/2005	49	194
8		05/07/2020	40	100
9		14/08/2005	36	178

הרצה:



```
SQL Window - UPDATE Invitation SET amount = ( SELECT COUNT(*) FROM Children WHERE Children.Group_ID = Invitat ...  
SQL Output Statistics  
UPDATE Invitation  
SET amount = (  
  SELECT COUNT(*)  
  FROM Children  
  WHERE Children.Group_ID = Invitation.Group_ID  
);
```

אחרי ההרצה:



```
SQL Window - Select * from Invitation; Select count(*) from Invitation;  
SQL Output Statistics  
Select * from Invitation;  
Select count(*) from Invitation;
```

	AMOUNT	INVITATION_DATE	GROUP_ID	MEAL_ID
1	11	27/01/2016	38	46
2	9	10/09/2007	3	150
3	13	02/06/2005	13	15
4	5	14/03/2005	39	114
5	13	01/12/2008	13	123
6	24	18/11/2005	11	125
7	14	29/11/2005	49	194
8	15	05/07/2020	40	100
9	11	14/08/2005	36	178
10	11	05/03/2010	17	180
11	9	11/03/2019	21	106
12	7	18/09/2010	22	107
13	9	30/12/2012	21	149
14	9	17/02/2018	2	116
15	14	30/12/2016	48	177
16	14	18/03/2008	4	30

(2

עדכון גילאי הקבוצות להיות בין 2 ל 11

UPDATE Groups

;SET age_group = FLOOR(DBMS_RANDOM.VALUE(2, 11))

לפני ההרצה :

SQL Window - Select * from class; Select count(*) from class; Select * from Meal; Select count(*) from Meal; ...

SQL Output Statistics

```

Select * from Groups;
Select count(*) from Groups;

Select * from Invitation;
Select count(*) from Invitation;

Select * from Kindergarten_Teacher;
Select count(*) from Kindergarten_Teacher;

```

Select class Select class Select meal Select meal Select groups Select groups Select invitation Select invitation Select kindergarten_teacher Select kindergarten_teacher Select children Select children

	GROUP_ID	GROUP_NAME	AGE_GROUP	KTEACHER_ID	HOURL	CLASS_DAY	CLASS_ID
1	1	Johnette	3	428283	16	THURSDAY	507
2	2	Spike	8	678542	17	FRIDAY	829
3	3	Trick	6	51274	14	TUESDAY	856
4	4	Kyra	9	241345	15	WEDNESDAY	742
5	5	Ethan	5	517314	13	WEDNESDAY	745
6	6	Leslie	10	719535	16	WEDNESDAY	725
7	7	Ben	6	367784	16	WEDNESDAY	823
8	8	Casey	2	574911	17	SUNDAY	616
9	9	Famke	8	9727100	15	THURSDAY	701
10	10	Heath	6	163546	17	TUESDAY	672
11	11	Dennis	5	792747	13	MONDAY	599
12	12	Johnnie	5	524373	13	TUESDAY	811
13	13	Andrew	8	578595	14	SUNDAY	481
14	14	Holly	3	82426	13	TUESDAY	646
15	15	Desmond	3	191553	17	FRIDAY	801
16	16	Sander	8	871418	13	MONDAY	470

16 rows selected in 0.047 seconds (more...)

הרצה:

SQL Window - UPDATE Groups SET age_group = FLOOR(DBMS_RANDOM.VALUE(2, 11));

SQL Output Statistics

```

UPDATE Groups
SET age_group = FLOOR(DBMS_RANDOM.VALUE(2, 11));

```

50 rows updated in 0.016 seconds

אחרי ההרצה:

SQL Window - Select * from Groups; Select count(*) from Groups;

SQL Output Statistics

```
Select * from Groups;
Select count(*) from Groups;
```

Select groups Select groups

	GROUP_ID	GROUP_NAME	AGE_GROUP	KTEACHER_ID	HOURL	CLASS_DAY	CLASS_ID
1	1	Johnette	...	2 428283	16	THURSDAY	507
2	2	Spike	...	6 678542	17	FRIDAY	829
3	3	Trick	...	9 51274	14	TUESDAY	856
4	4	Kyra	...	3 241345	15	WEDNESDAY	742
5	5	Ethan	...	5 517314	13	WEDNESDAY	745
6	6	Leslie	...	6 719535	16	WEDNESDAY	725
7	7	Ben	...	4 367784	16	WEDNESDAY	823
8	8	Casey	...	6 574911	17	SUNDAY	616
9	9	Famke	...	3 9727100	15	THURSDAY	701
10	10	Heath	...	10 163546	17	TUESDAY	672
11	11	Dennis	...	8 792747	13	MONDAY	599
12	12	Johnnie	...	2 524373	13	TUESDAY	811
13	13	Andrew	...	9 578595	14	SUNDAY	481
14	14	Holly	...	4 82426	13	TUESDAY	646
15	15	Desmond	...	5 191553	17	FRIDAY	801
16	16	Sander	...	3 871418	13	MONDAY	470
17	17	Bonnie	...	7 916780	15	FRIDAY	654

3:1 | michal@XE | 17 rows selected in 0.063 seconds (more...)

שאלות מחיקה

(

מחיקת קבוצות ללא ילדים

DELETE FROM Groups

) WHERE Group_ID NOT IN

SELECT DISTINCT Group_ID

FROM Children

;

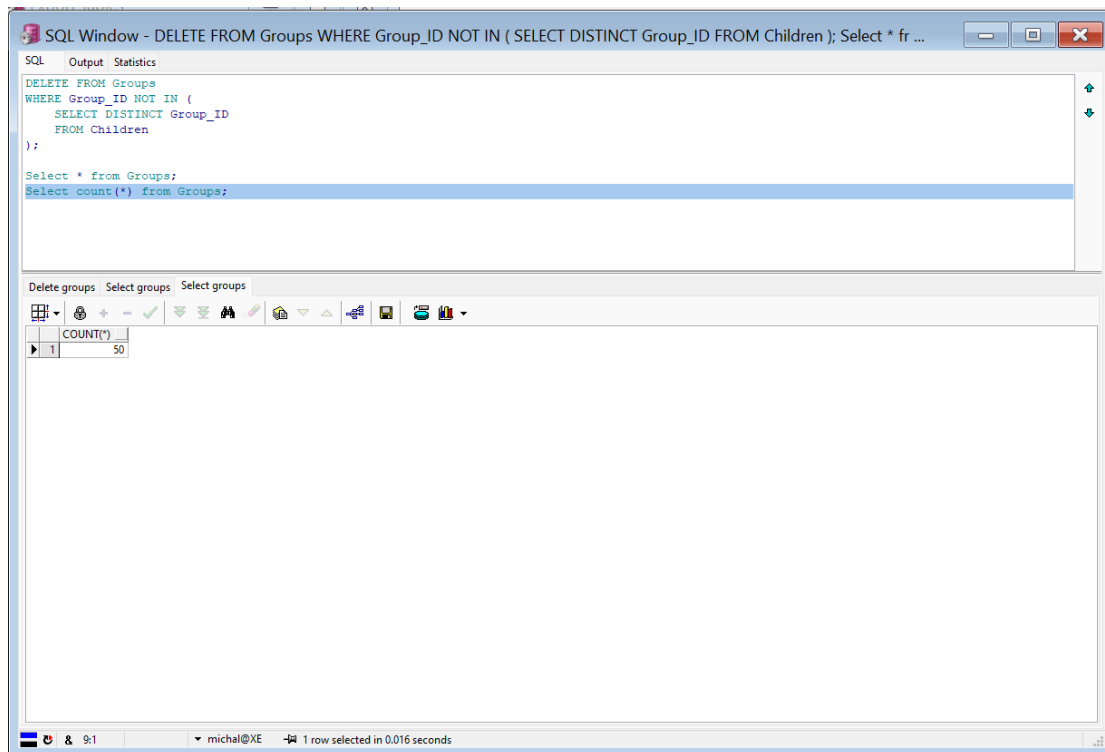
לפני ההרצה:

Select groups Select groups

	COUNT(*)
1	50

אחרי ההרצה:

(נשאר אותו דבר מאחר ואין קבוצות ללא ילדים)

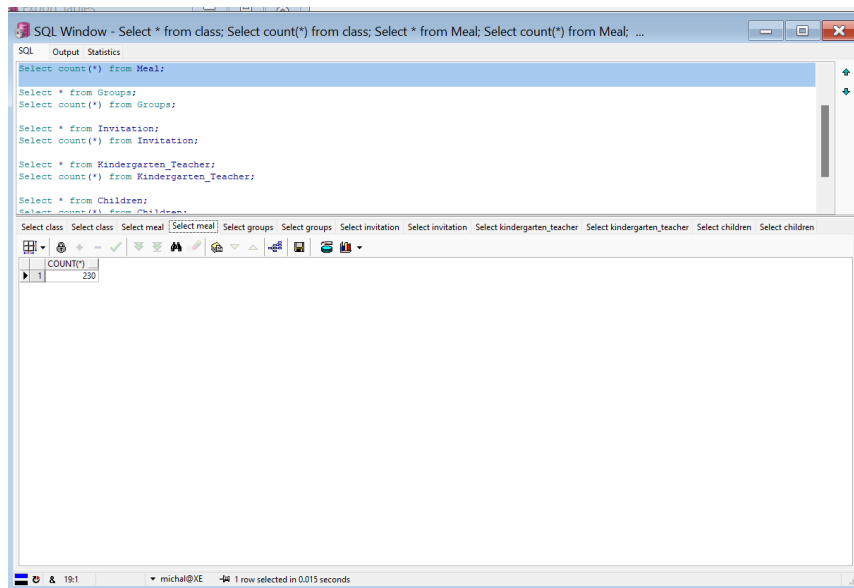


(2

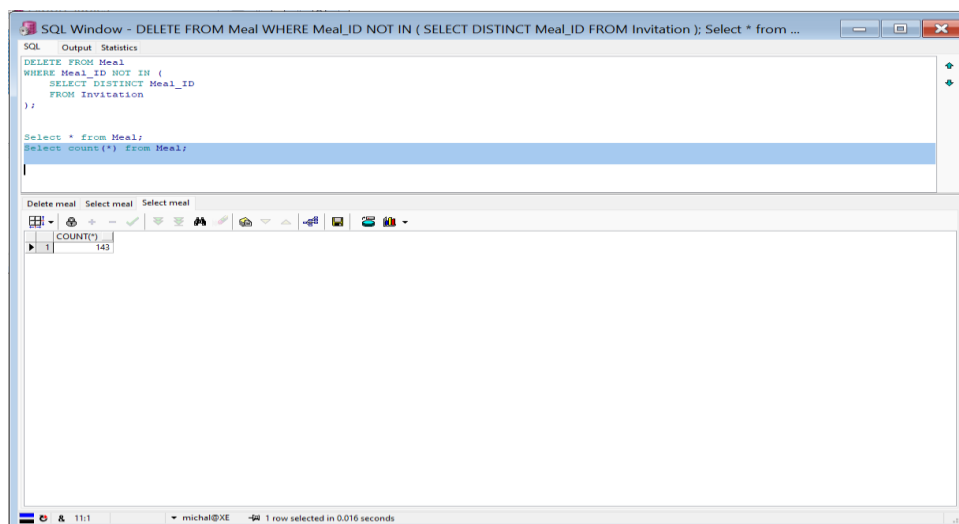
מחיקת מנות שלא הוזמנו באף הזמנה

```
DELETE FROM Meal
) WHERE Meal_ID NOT IN
SELECT DISTINCT Meal_ID
FROM Invitation
; (
```

לפני ההרצה :



אחרי ההרצה:



אילוצים

(1

האילוץ מוודא שגיל הילד אינו שלילי, כלומר תאריך הלידה של הילד חייב להיות קטן מהתאריך הנוכחי.

```
CREATE OR REPLACE TRIGGER trg_check_date_of_birth
BEFORE INSERT OR UPDATE ON Children
FOR EACH ROW
BEGIN
```

IF :NEW.Date_of_birth > SYSDATE THEN

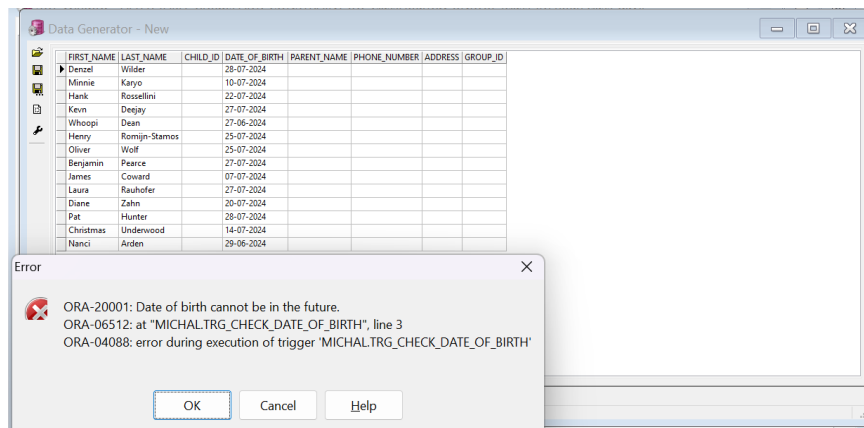
RAISE_APPLICATION_ERROR(-20001, 'Date of birth cannot be in the
;future.')

;END IF

;END

/

ניסיון להכניס ערך אשר סותר את האילוץ:



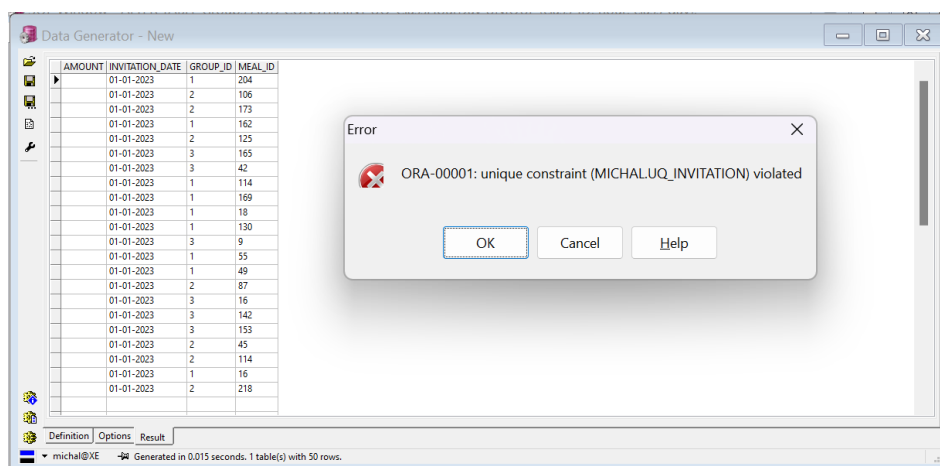
(2

**אילוץ UNIQUE על השדה Invitation_date ו-Group_ID יחד, כדי לוודא שאין יותר
מהזמנה אחת לקבוצה באותו יום**

ALTER TABLE Invitation

;ADD CONSTRAINT UQ_Invitation UNIQUE (Invitation_date, Group_ID)

ניסיון להכניס ערך אשר סותר את האילוץ:



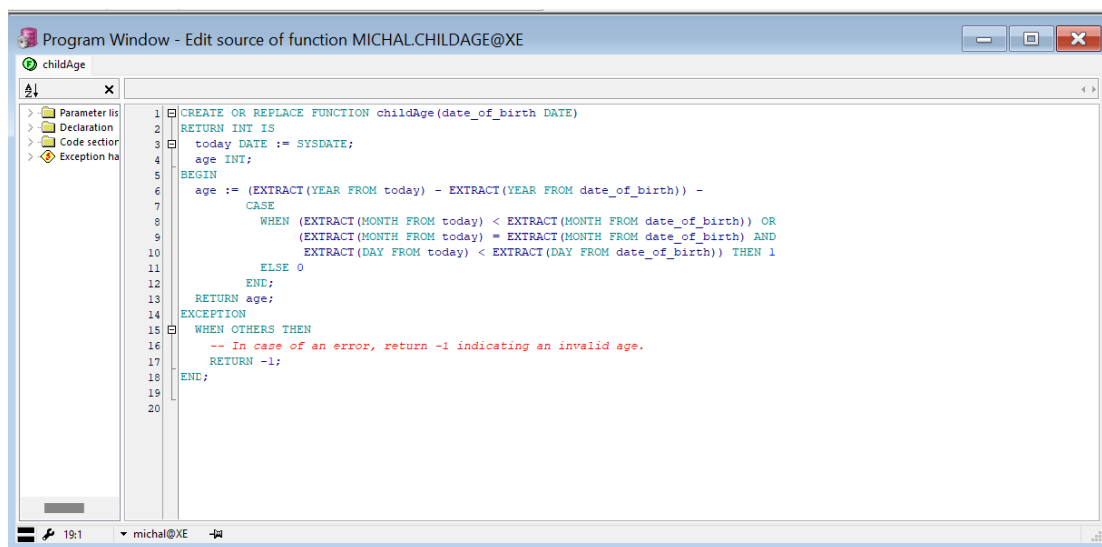
שלב ג

תוכנית מספר 1

פונקציה

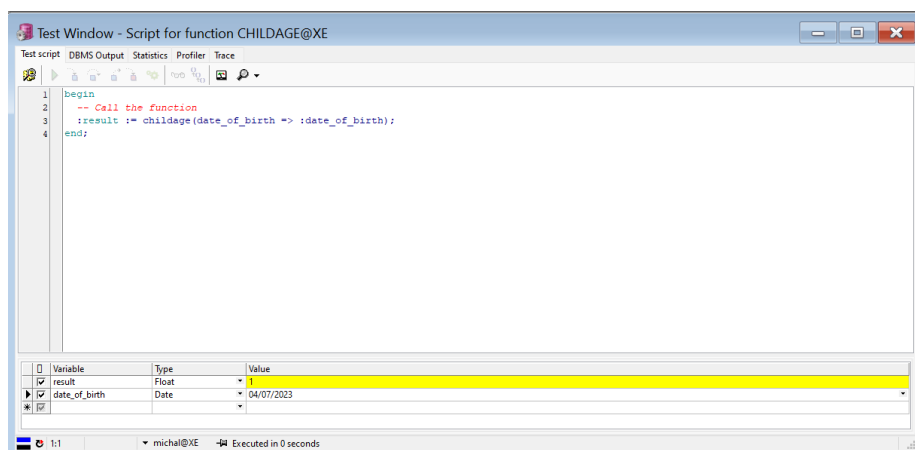
מחשבת את גיל הילד על סמך תאריך הלידה

הפונקציה לחישוב גיל :



```
1 CREATE OR REPLACE FUNCTION childAge(date_of_birth DATE)
2 RETURN INT IS
3   today DATE := SYSDATE;
4   age INT;
5 BEGIN
6   age := (EXTRACT(YEAR FROM today) - EXTRACT(YEAR FROM date_of_birth)) -
7   CASE
8     WHEN (EXTRACT(MONTH FROM today) < EXTRACT(MONTH FROM date_of_birth)) OR
9          (EXTRACT(MONTH FROM today) = EXTRACT(MONTH FROM date_of_birth) AND
10           EXTRACT(DAY FROM today) < EXTRACT(DAY FROM date_of_birth)) THEN 1
11     ELSE 0
12   END;
13   RETURN age;
14 EXCEPTION
15   WHEN OTHERS THEN
16     -- In case of an error, return -1 indicating an invalid age.
17     RETURN -1;
18 END;
```

דוגמאת הרצה :



```
1 begin
2   -- Call the function
3   :result := childAge(date_of_birth => :date_of_birth);
4 end;
```

Variable	Type	Value
result	Float	1
date_of_birth	Date	04/07/2023

פרוצדורה

מקבלת ת"ז של ילד ומעדכנת לו את מס' הקבוצה בהתאם לגיל שלו. נניח עבור ילד בגיל 5 יבחר באופן רנדומלי צהרון מתוך כל הצהרונים שקבוצת הגיל שלהם 5

הרצת הפרוצדורה: נתוני הילד לפני הרצה:

10	Holland	Palmieri	7716810	11/09/2021	Holland	0518400150	60 The Woodlands	...	49
----	---------	----------	---------	------------	---------	------------	------------------	-----	----

הפרוצדורה:

Program Window - Edit source of procedure MICHAL.UPDATECHILDGROUP@XE

```

20 SELECT Group_ID INTO v_group_id
21 FROM (
22     SELECT Group_ID
23     FROM Groups
24     WHERE age_group = v_child_age
25     ORDER BY DBMS_RANDOM.VALUE
26 )
27 WHERE ROWNUM = 1;
28
29 DBMS_OUTPUT.PUT_LINE('Assigned new group ID: ' || v_group_id);
30
31 -- עדכון מספר הקבוצה בטבלת הילדים
32 UPDATE Children
33 SET Group_ID = v_group_id
34 WHERE child_ID = p_child_id;
35
36 -- אישור עדכונים
37 COMMIT;
38
39 -- הדפסת אישור עדכון
40 DBMS_OUTPUT.PUT_LINE('Child ' || p_child_id || ' moved from group ' || v_original_group_id || ' to ' || v_group_id);
41 EXCEPTION
42 WHEN NO_DATA_FOUND THEN
43     -- הודעה ומקרה ולא נמצאה קבוצה מתאימה
44     DBMS_OUTPUT.PUT_LINE('No suitable group found for the child.');
```

אחרי הרצת הטסט:

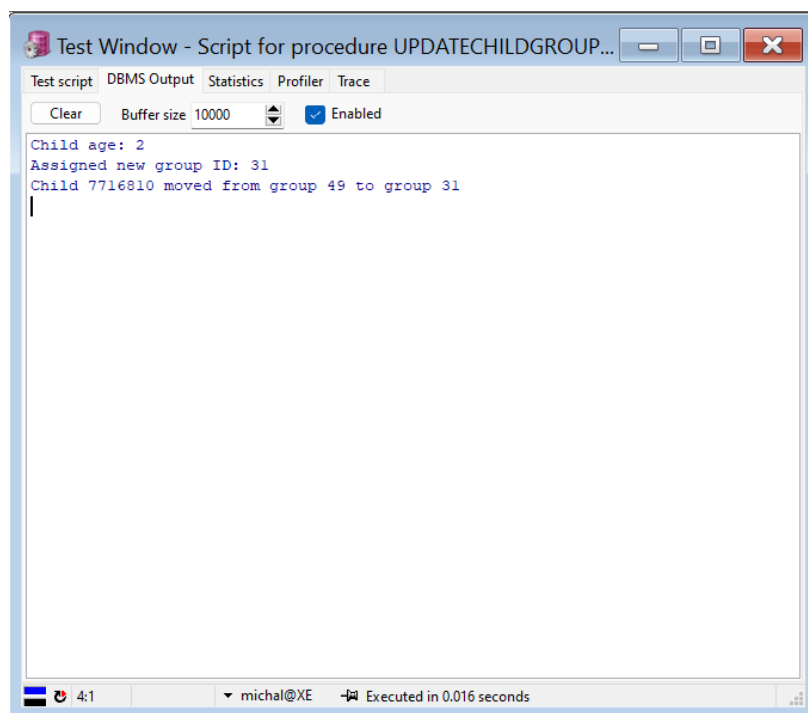
Test Window - Script for procedure UPDATECHILDGROUP...

```

1 begin
2     -- Call the procedure
3     updatechildgroup(p_child_id => :p_child_id);
4 end;
```

Variable	Type	Value
p_child_id	String	7716810

4:5 micahal@XE Executed in 0.016 seconds



העדכון בטבלה:

10	Holland	Palmieri	7716810	11/09/2021	Holland	0518400150	60 The Woodlands	...	31
----	---------	----------	---------	------------	---------	------------	------------------	-----	----

תוכנית ראשית

עוברת על כל טבלת הילדים ומעדכנת עבורם את מס' הקבוצה בהתאם לגיל.

לפני הרצה-

	FIRST_NAME	LAST_NAME	CHILD_ID	DATE_OF_BIRTH	PARENT_NAME	PHONE_NUMBER	ADDRESS	GROUP_ID
1	Frances	Lerner	142341	16/12/2012	Frances	0554495143	31st Street	42
2	Chad	Van Damme	553762	26/04/2021	Chad	0508672962	46 Bloomington Stree	33
3	Roscoe	Lonsdale	485513	11/06/2020	Roscoe	0513927528	61 Nanaimo Street	14
4	Bradley	Crimson	764244	28/01/2019	Bradley	0539108186	18 Sarsgaard Street	46
5	Petula	Summer	967935	12/03/2021	Petula	0561879033	37 Dushku Road	26
6	Holly	Carmen	693396	20/01/2021	Holly	0544162690	13 Whitley Drive	33
7	Rhona	McKellen	169697	17/03/2021	Rhona	0573618630	97 Hedaya Road	33
8	Benjamin	Klein	293198	29/08/2017	Benjamin	0561924672	50 Beals Road	27
9	Ike	Pepper	755639	03/07/2014	Ike	0517277947	84 Allison Ave	10
10	Holland	Palmieri	7716810	11/09/2021	Holland	0518400150	60 The Woodlands	31
11	Timothy	Englund	2126911	15/03/2016	Timothy	0577740922	68 Richie Ave	18
12	Powers	Hopkins	6965212	26/03/2013	Powers	0524952434	99 Burke Street	45
13	Natasha	Roy Parnell	9887213	12/02/2016	Natasha	0519060777	42 Debra Ave	11
14	Mac	Scorsese	3944314	01/07/2015	Mac	0563455306	92nd Street	48
15	Cevin	Utada	3346513	02/02/2017	Cevin	0566262106	63rd Street	32
16	Red	McElhone	7314116	23/01/2012	Red	0502587459	15 North Bethesda Dr	34
17	Wayne	Pitney	1356717	29/09/2016	Wayne	0525511296	38 Rochester Road	32
18	Tommy	Rivers	1145418	03/11/2018	Tommy	0532602705	901 Germantown Stree	34

התוכנית-

```
SQL Window - DECLARE CURSOR child_cursor IS SELECT child_ID FROM Children; v_child_id VARCHAR(15); BEGIN -- פ ...
SQL Output Statistics
DECLARE
CURSOR child_cursor IS
SELECT child_ID
FROM Children;

v_child_id VARCHAR(15);
BEGIN
-- פתיחת הקורסור
OPEN child_cursor;

LOOP
-- טעימת רשומה מהקורסור
FETCH child_cursor INTO v_child_id;

-- יציאה מהלולאה כאשר אין יותר רשומות
EXIT WHEN child_cursor%NOTFOUND;

-- קריאה לפונקציה שבודקת כל ילד
UpdateChildGroup(v_child_id);
END LOOP;

-- סגירת הקורסור
CLOSE child_cursor;
END;
/
```

אחרי הרצה-

SQL Window - Select * from Children; Select count(*) from Children;

```
SQL Output Statistics
Select * from Children;
Select count(*) from Children;
```

Select children Select children

	FIRST_NAME	LAST_NAME	CHILD_ID	DATE_OF_BIRTH	PARENT_NAME	PHONE_NUMBER	ADDRESS	GROUP_ID
1	Frances	Lerner	142341	16/12/2012	Frances	0554495143	31st Street	2
2	Chad	Van Damme	553762	26/04/2021	Chad	0508672962	46 Bloomington Stree	17
3	Roscoe	Lonsdale	485513	11/06/2020	Roscoe	0513927528	61 Nanaimo Street	33
4	Bradley	Crimson	764244	28/01/2019	Bradley	0539108186	18 Sarsgaard Street	9
5	Petula	Sumner	967935	12/03/2021	Petula	0561879033	37 Dushku Road	5
6	Holly	Carmen	693396	20/01/2021	Holly	0544162690	13 Whitley Drive	24
7	Rhona	McKellen	169697	17/03/2021	Rhona	0573618630	97 Hedaya Road	24
8	Benjamin	Klein	293198	29/08/2017	Benjamin	0561924672	50 Beals Road	41
9	Ike	Pepper	755639	03/07/2014	Ike	0517277947	84 Allison Ave	48
10	Holland	Palmieri	7716810	11/09/2021	Holland	0518400150	60 The Woodlands	35
11	Timothy	Englund	2126911	15/03/2016	Timothy	0577740922	68 Richie Ave	17
12	Powers	Hopkins	6965212	26/03/2013	Powers	0524952434	99 Burke Street	2
13	Natasha	Roy Parnell	9807213	12/02/2016	Natasha	0519060777	42 Debra Ave	27
14	Mac	Scorsese	3944314	01/07/2015	Mac	0563455306	92nd Street	16
15	Cevin	Utada	3346515	02/02/2017	Cevin	0566262106	63rd Street	8
16	Red	McElhone	7314116	23/01/2012	Red	0502587459	15 North Bethesda Dr	32
17	Wayne	Pitney	1356717	29/09/2016	Wayne	0525511296	38 Rochester Road	13
18	Tommy	Rivers	1145418	03/11/2018	Tommy	0532602705	901 Germantown Stree	9

תוכנית מספר 2

פונקציה

הפונקציה מחזירה cursor עם כל הילדים מעל גיל 10.

יצירת הפונקציה

CREATE OR REPLACE FUNCTION GetChildrenAbove10

RETURN SYS_REFCURSOR

IS

```

;children_cursor SYS_REFCURSOR

BEGIN

OPEN children_cursor FOR

SELECT c.child_id, c.first_name, c.last_name, c.date_of_birth

FROM CHILDREN c

;WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, c.date_of_birth) / 12) > 10


;RETURN children_cursor

;END

```

פרוצדורה

פרוצדורה: UpdatePhoneNumbers

הפרוצדורה משתמשת בקורסור המוחזר מהפונקציה GetChildrenAbove10 כדי לעדכן את מספרי הטלפון של הילדים.

הפרוצדורה מקבלת את הקורסור מהפונקציה GetChildrenAbove10.

עוברת בלולאה על כל רשומה בקורסור.

מחשבת את מספר הטלפון החדש בהתבסס על תעודת הזהות של הילד, ומעדכנת את השדה phone_number בטבלה.

יצירת הפרוצדורה

```

CREATE OR REPLACE PROCEDURE UpdatePhoneNumbers AS

;children_cursor SYS_REFCURSOR

;child_id_val CHILDREN.child_id%TYPE

;first_name CHILDREN.first_name%TYPE

;last_name CHILDREN.last_name%TYPE

;date_of_birth CHILDREN.date_of_birth%TYPE

;new_phone_number CHILDREN.phone_number%TYPE

```

תוכנית ראשית

```

BEGIN

-- קבל את הקורסור מהפונקציה

```

```
;children_cursor := GetChildrenAbove10
```

```
LOOP
```

```
FETCH children_cursor INTO child_id_val, first_name, last_name,  
;date_of_birth
```

```
;EXIT WHEN children_cursor%NOTFOUND
```

```
-- כאן נחשב את מספר הטלפון
```

```
;new_phone_number := '05' || LPAD(SUBSTR(child_id_val, 1, 8), 8, '0')
```

```
-- עדכון מספר הטלפון
```

```
UPDATE CHILDREN
```

```
SET phone_number = new_phone_number
```

```
;WHERE child_id = child_id_val
```

```
-- הדפסת הודעה למסך
```

```
DBMS_OUTPUT.PUT_LINE('Updated phone number for ' || first_name || ' ' ||  
;last_name || ' (ID: ' || child_id_val || ') to ' || new_phone_number)
```

```
;END LOOP
```

```
;CLOSE children_cursor
```

```
-- ביצוע COMMIT לשמירת השינויים
```

```
;COMMIT
```

```
;END
```

```
/
```

```
-- הרצת הפרוצדורה
```

BEGIN

;UpdatePhoneNumbers

;END

	FIRST_NAME	LAST_NAME	CHILD_ID	DATE_OF_BIRTH	PARENT_NAME	PHONE_NUMBER	ADDRESS	GROUP_ID
1	Frances	Lerner	142341	16/12/2012	Frances	0554495143	31st Street	42
2	Chad	Van Damme	553762	26/04/2021	Chad	0508672962	46 Bloomington Stree	20
3	Roscoe	Lonsdale	485513	11/06/2020	Roscoe	0513927528	61 Nanaimo Street	42
4	Bradley	Crimson	764244	28/01/2019	Bradley	0539108186	18 Sarsgaard Street	8
5	Petula	Summer	967935	12/03/2021	Petula	0561879033	37 Dushku Road	32
6	Holly	Carmen	693396	20/01/2021	Holly	0544162690	13 Whitley Drive	26
7	Rhona	McKellen	169697	17/03/2021	Rhona	0573618630	97 Hedaya Road	24
8	Benjamin	Klein	293198	29/08/2017	Benjamin	0561924672	50 Beals Road	17
9	Ike	Pepper	755639	03/07/2014	Ike	0517277947	84 Allison Ave	29
10	Holland	Palmieri	7716810	11/09/2021	Holland	0518400150	60 The Woodlands	39
11	Timothy	Englund	2126911	15/03/2016	Timothy	0577740922	68 Richie Ave	9
12	Powers	Hopkins	6965212	26/03/2013	Powers	0524952434	99 Burke Street	45
13	Natasha	Roy Parnell	9887213	12/02/2016	Natasha	0519060777	42 Debra Ave	19
14	Mac	Scorsese	3944314	01/07/2015	Mac	0563455306	92nd Street	21

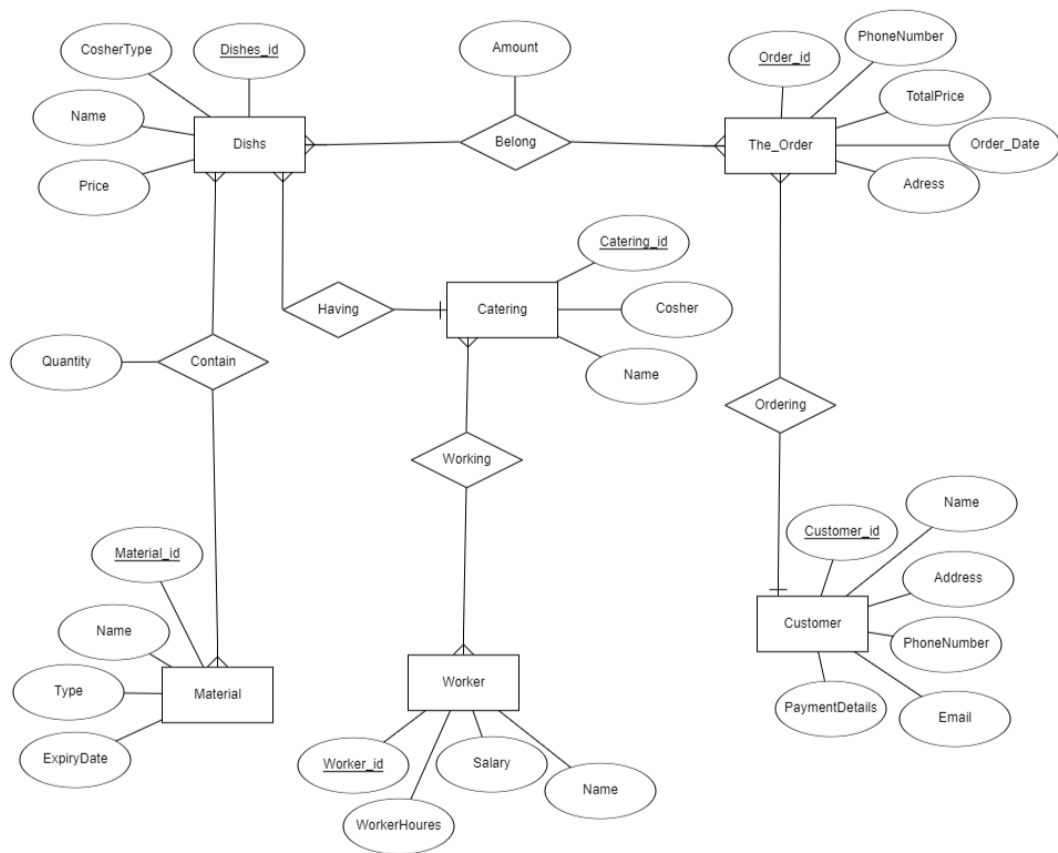
	CHILD_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	142341	Frances	Lerner	0500142341
2	6965212	Powers	Hopkins	0506965212
3	7314116	Red	McElhone	0507314116
4	7951322	Toni	Remar	0507951322
5	2295730	Nikki	Sandoval	0502295730
6	4469332	Garry	Joli	0504469332
7	3786647	Cliff	Chesnutt	0503786647
8	4594453	Owen	Aaron	0504594453
9	5864866	Jamie	Archer	0505864866
10	9556271	Maura	Herndon	0509556271
11	3529473	Stevie	Wincott	0503529473
12	1679575	Alessandro	Geldof	0501679575

	CHILD_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	142341	Frances	Lerner	0554495143
2	6965212	Powers	Hopkins	0524952434
3	7314116	Red	McElhone	0502587459
4	7951322	Toni	Remar	0514994047
5	2295730	Nikki	Sandoval	0545390936
6	4469332	Garry	Joli	0566279796
7	3786647	Cliff	Chesnutt	0544163233
8	4594453	Owen	Aaron	0576599720
9	5864866	Jamie	Archer	0525417421
10	9556271	Maura	Herndon	0509878949
11	3529473	Stevie	Wincott	0578841824
12	1679575	Alessandro	Geldof	0527952947

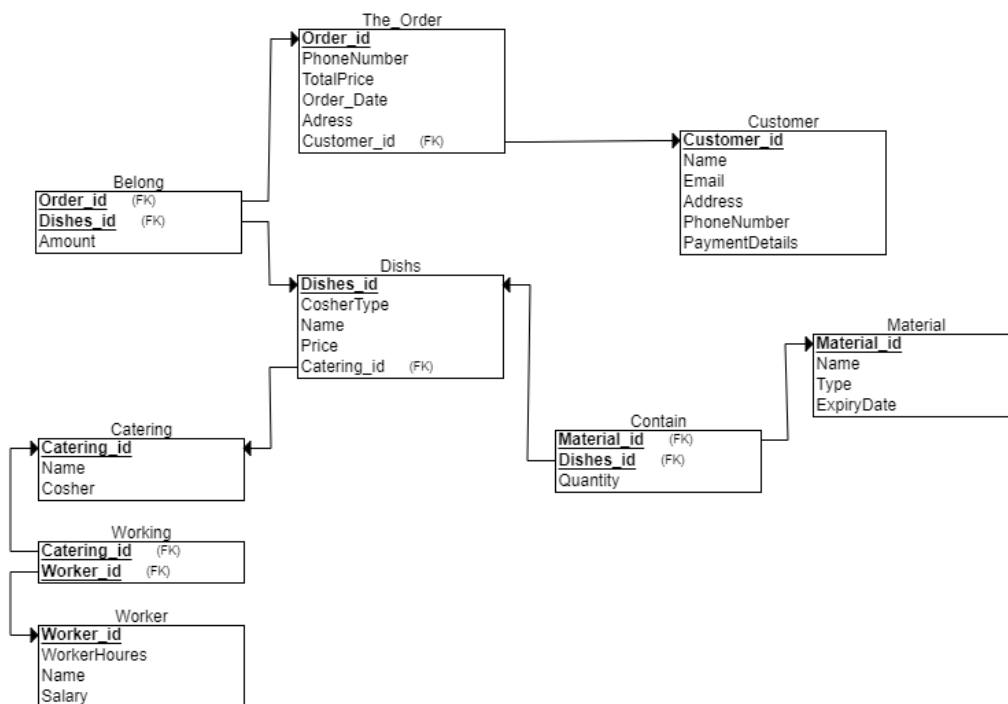
Updated phone number for Arturo Keaton (ID: 43364485) to 0543364485
 Updated phone number for Carlos Gallant (ID: 55647492) to 0555647492
 Updated phone number for Liv Orbit (ID: 18847494) to 0518847494
 Updated phone number for Crispin Colton (ID: 98139496) to 0598139496
 Updated phone number for Carrie Schneider (ID: 75116497) to 0575116497
 Updated phone number for Cameron Rossellini (ID: 15196508) to 0515196508
 Updated phone number for Winona Fierstein (ID: 64884524) to 0564884524
 Updated phone number for Lindsay McGriff (ID: 96138527) to 0596138527
 Updated phone number for Rueben Farrow (ID: 23264530) to 0523264530
 Updated phone number for Jamie Rossellini (ID: 25478542) to 0525478542
 Updated phone number for Jill Crewson (ID: 72452549) to 0572452549
 Updated phone number for Andrea Whitmore (ID: 53223550) to 0553223550
 Updated phone number for Simon Byrne (ID: 99816551) to 0599816551
 Updated phone number for Tanya Blanchett (ID: 75793553) to 0575793553
 Updated phone number for Grant Newton (ID: 36915556) to 0536915556
 Updated phone number for Jody Aglukark (ID: 95896559) to 0595896559
 Updated phone number for Shawn Begley (ID: 22817562) to 0522817562
 Updated phone number for Betty Cartlidge (ID: 65377565) to 0565377565
 Updated phone number for Clay Chappelle (ID: 56125566) to 0556125566
 Updated phone number for Patti Galecki (ID: 21195575) to 0521195575
 Updated phone number for Taylor Pitney (ID: 32266578) to 0532266578

שלב ד

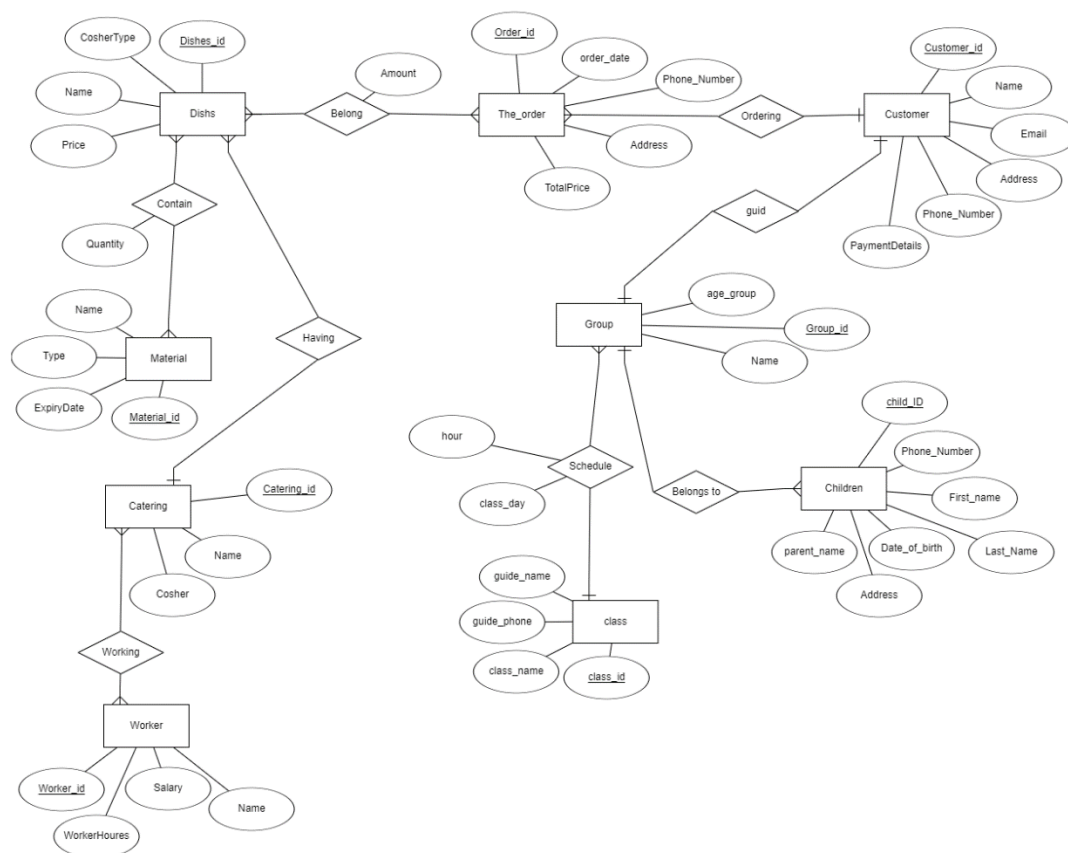
תרשים ERD



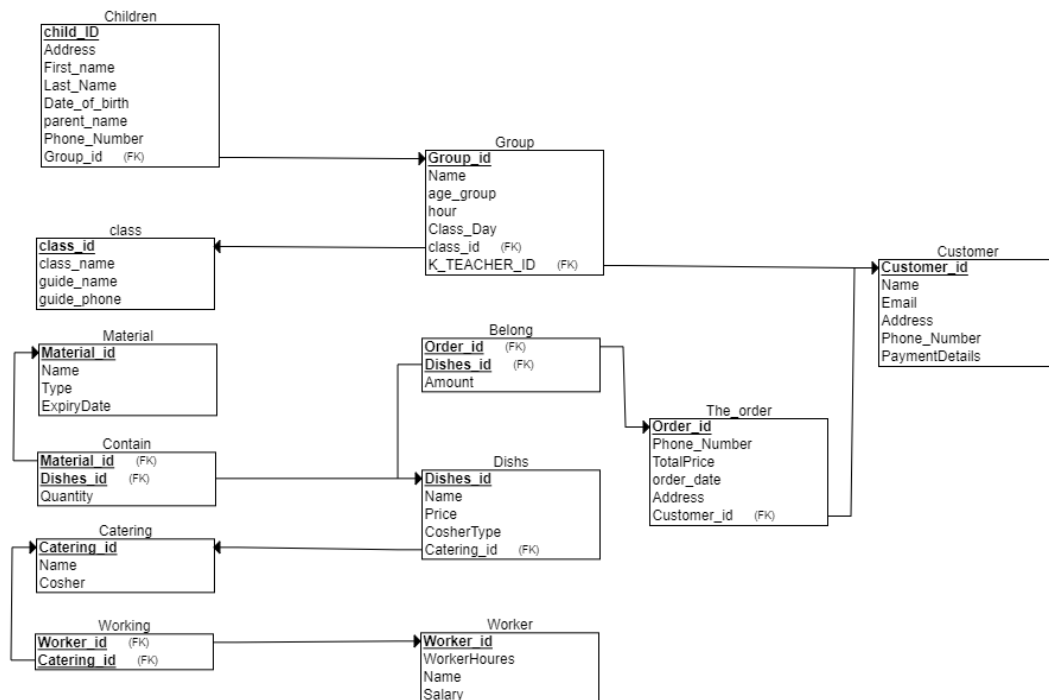
תרשים DSD



תרשים ERD אחרי אינטגרציה



תרשים DSD אחרי אינטגרציה



החלטות

- עבור הטבלה "מנות" שהייתה בפרויקט המקורי- נוריד את השדה "יום", הבחירה של המנות לא תהיה קבועה מראש אלא תתבצע על ידי הגננת
- מיזוג הטבלה "מנות" מ2 הפרוייקטים, הוספת השדות החסרים, מחיקת הטבלה מנות המיותרת.
- מחיקת טבלת הקשר INVITATION
- נמחק את הטבלה "גננות", במקומה נקשר את "לקוח" ל- קבוצות. מי שיבצע את ההזמנה לצהרון תהיה הגננת:
- מחיקת השדה EMAIL ו ADDRESS מ"לקוח" (אימייל לא רלוונטי וכתובת מופיעה בפרטי ההזמנה)
- מחיקת השדה "שם פרטי" ושינוי שם השדה "שם משפחה" ל- NAME
- מיזוג הטבלאות ומחיקת הטבלה "גננות"

צילומי מסך של שינוי ומיזוג הטבלאות: (1)

```
Select * from Kindergarten_Teacher;
Select count(*) from Kindergarten_Teacher;

Select * from customer;
Select count(*) from customer;
```

	COUNT(*)
1	321

53:1 michal@XE 1 row selected in 0.016 seconds

```
Select * from Kindergarten_Teacher;
Select count(*) from Kindergarten_Teacher;

Select * from customer;
Select count(*) from customer;
```

	COUNT(*)
1	100

53:1 michal@XE 1 row selected in 0.016 seconds

```
Select kindergarten_teacher   Select kindergarten_teacher   Select customer   Select customer
```

	COUNT(*)
1	421

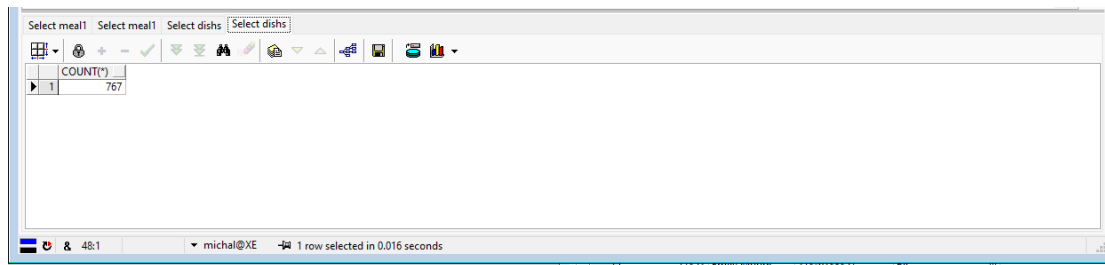
58:1 michal@XE 1 row selected in 0.016 seconds

(2)

```
Select meal1   Select meal1   Select dishes   Select dishes
```

	COUNT(*)
1	67

48:1 michal@XE 1 row selected in 0.016 seconds



מבט 1

תאור מילולי

המבט מציג מידע על הצהרונים והילדים הרשומים בהם

```
CREATE VIEW KindergartenOverview AS
SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class_day,
c.first_name, c.last_name, c.date_of_birth, c.parent_name
FROM GROUPS g
JOIN CHILDREN1 c ON g.group_id = c.group_id;
```

```
SELECT *
FROM KindergartenOverview;
```

SQL Window - CREATE VIEW KindergartenOverview AS SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class_day, c.first_name, c.last_name, c.date_of_birth, c.parent_name

```
CREATE VIEW KindergartenOverview AS
SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class_day, c.first_name, c.last_name, c.date_of_birth, c.parent_name
FROM GROUPS g
JOIN CHILDREN1 c ON g.group_id = c.group_id;

SELECT *
FROM KindergartenOverview;
```

	GROUP_ID	GROUP_NAME	AGE_GROUP	HOUR	CLASS_DAY	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	PARENT_NAME
1	2	Spike	...	11	FRIDAY	Frances	Lerner	16/12/2012	Frances
2	17	Bonnie	...	3	FRIDAY	Chad	Van Damme	26/04/2021	Chad
3	33	Rebecca	...	4	FRIDAY	Roscoe	Lonsdale	11/06/2020	Roscoe
4	9	Famke	...	5	THURSDAY	Bradley	Crimson	28/01/2019	Bradley
5	5	Ethan	...	3	WEDNESDAY	Petula	Summer	12/03/2021	Petula
6	24	Trini	...	3	SUNDAY	Holly	Carmen	20/01/2021	Holly
7	24	Trini	...	3	SUNDAY	Rhona	McKellen	17/03/2021	Rhona
8	41	Jarvis	...	6	THURSDAY	Benjamin	Klein	29/08/2017	Benjamin
9	48	Raymond	...	10	THURSDAY	Ike	Pepper	03/07/2014	Ike
10	35	Avril	...	2	MONDAY	Holland	Palmieri	11/09/2021	Holland
11	47	Susan	...	8	MONDAY	Timothy	Englund	15/03/2016	Timothy
12	2	Spike	...	11	FRIDAY	Powers	Hopkins	26/03/2013	Powers
13	27	Marc	...	8	THURSDAY	Natasha	Roy Parnell	12/02/2016	Natasha
14	16	Sander	...	9	MONDAY	Mac	Scorsese	01/07/2015	Mac
15	8	Casey	...	7	SUNDAY	Cevin	Utada	02/02/2017	Cevin
16	32	Olympia	...	12	SUNDAY	Red	McElhone	23/01/2012	Red
17	13	Andrew	...	7	SUNDAY	Wayne	Pitney	29/09/2016	Wayne
18	9	Famke	...	5	THURSDAY	Tommy	Rivers	03/11/2018	Tommy
19	25	Sara	...	7	TUESDAY	Chant	Brock	30/11/2016	Chant
20	48	Raymond	...	10	THURSDAY	Patrick	Branagh	14/10/2013	Patrick
21	27	Marc	...	8	THURSDAY	Judy	Gugino	28/02/2016	Judy
22	2	Spike	...	11	FRIDAY	Toni	Remar	04/04/2013	Toni
23	2	Spike	...	11	FRIDAY	F... ..	F... ..	08/11/2019	F... ..

587 rows selected in 0.296 seconds

שליפת נתונים

שאלתה 1: מציאת כל הילדים בקבוצה מסוימת, לפי שם הקבוצה ('Spike')

```
SELECT first_name, last_name, date_of_birth
FROM KindergartenOverview
```

WHERE group_name = 'Spike';

SQL Window - CREATE VIEW KindergartenOverview AS SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class ...

```

SQL      Output  Statistics
SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class_day, c.first_name, c.last_name, c.date_of_birth, c.parent_name
FROM GROUPS g
JOIN CHILDREN c ON g.group_id = c.group_id;

SELECT *
FROM KindergartenOverview;

SELECT first_name, last_name, date_of_birth
FROM KindergartenOverview
WHERE group_name = 'Spike';

```

	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH
1	Frances	Lerner	16/12/2012
2	Powers	Hopkins	26/02/2013
3	Toni	Remar	04/04/2013
4	Crispin	Blackmore	06/07/2013
5	Lloyd	Johnson	16/03/2013
6	Lin	Bening	28/04/2013
7	Dionne	Kilmer	19/03/2013
8	Victoria	Horton	07/11/2012
9	Alannah	Car	14/10/2012
10	Sandra	Noseworthy	27/01/2013
11	Kevin	Phoenix	15/03/2013
12	Joshua	Wierhorst	27/07/2012
13	Eugene	Neeson	24/04/2013

13 rows selected in 0.031 seconds (more...)

שאלתה 2: סופרת כמה ילדים יש בכל קבוצת גיל

SELECT age_group, COUNT(*)

GROUP BY age_group;

SQL Window - CREATE VIEW KindergartenOverview AS SELECT g.group_id, g.group_name, g.age_group, g.hour, g.class ...

```

SQL      Output  Statistics
SELECT first_name, last_name, date_of_birth
FROM KindergartenOverview
WHERE group_name = 'Spike';

SELECT age_group, COUNT(*)
FROM KindergartenOverview
GROUP BY age_group;

```

	AGE_GROUP	COUNT(*)
1	11	65
2	6	58
3	2	35
4	4	51
5	5	65
6	8	51
7	3	50
8	7	62
9	10	58
10	9	62
11	12	30

11 rows selected in 0.031 seconds

מבט 2

תאור מילולי

מבט שמתמקד בפרטים הכוללים של ההזמנות בפרויקט הקייטרינג, כולל פרטים על הלקוחות שביצעו את ההזמנות והמנות שהוזמנו.

```

CREATE VIEW CateringOrderDetails AS
SELECT o.order_id, o.totalprice, o.order_date, o.address, c.name AS
customer_name, c.phonenumber, d.description AS dish_description,
b.amount

```

```

FROM THE_ORDER o
JOIN CUSTOMER c ON o.customer_id = c.customer_id
JOIN BELONG b ON o.order_id = b.order_id
JOIN DISHS d ON b.dishes_id = d.dishes_id;

SELECT *
FROM CateringOrderDetails;

```

SQL Window - CREATE VIEW CateringOrderDetails AS SELECT o.order_id, o.totalprice, o.order_date, o.address, c...

SQL Output Statistics

```

CREATE VIEW CateringOrderDetails AS
SELECT o.order_id, o.totalprice, o.order_date, o.address, c.name AS customer_name, c.phonenumber, d.description AS dish_description, b.amount
FROM THE_ORDER o
JOIN CUSTOMER c ON o.customer_id = c.customer_id
JOIN BELONG b ON o.order_id = b.order_id
JOIN DISHS d ON b.dishes_id = d.dishes_id;

SELECT *
FROM CateringOrderDetails;

```

	ORDER_ID	TOTALPRICE	ORDER_DATE	ADDRESS	CUSTOMER_NAME	PHONENUMBER	DISH_DESCRIPTION	AMOUNT
1	40160	1000000	21/08/2021	56 Maintenon Ave	Katie Brown	659-2102763	Sate Lilit	655239
2	86420	3308944	16/12/2022	54 Ani Drive	Katie Brown	743-8999983	Sate Padang Aga	471115
3	80904	1000000	27/10/2021	20 Dartmouth Drive	Chris Smith	870-4296518	Sate Padang Sol	214624
4	54338	1000000	25/01/2022	52 Frost Blvd	Laura Williams	853-4227521	Sate Meranggi	508938
5	29984	1000000	15/10/2022	80 Haverhill Blvd	Jane Johnson	827-1983393	Ketupat	308320
6	91407	1000000	11/05/2022	85 Hjalperup	Katie Miller	813-6454142	Sate Padang Men	794745
7	60877	1000000	29/08/2022	62 Atlanta Road	Chris Davis	644-8928678	Sate Padang Aga	597745
8	91588	1000000	25/04/2022	36 Macy Road	David Moore	547-6274084	Sate Padang Aga	950038
9	84219	1000000	14/11/2020	26 Carlisle Drive	David Davis	251-1565707	Sayur Lodeh	931659
10	97948	486562	07/05/2022	80 Thorton Street	Chris Davis	643-3661921	Sate Padang Dha	804636
11	59136	1000000	02/12/2023	63rd Street	Michael Smith	518-3643990	Laksa	21273
12	20547	1000000	31/12/2021	61st Street	Laura Smith	667-7284259	Bihun	418661
13	63955	9464071	24/09/2020	3 Elizabeth Street	Katie Moore	400-3815204	Empanada	525475
14	24034	1000000	09/08/2022	358 Worell Street	Katie Miller	855-4843222	Sate Padang Sol	285298
15	10259	7714557	16/07/2020	66 Omaha Drive	Michael Smith	309-2558871	Sate Lilit	984079
16	69961	1000000	17/01/2022	30 McAnally Street	David Davis	251-1565707	Sate Padang Aga	960425
17	34506	1000000	06/06/2020	96 Santa Cruz Street	Emily Taylor	319-6111245	Sate Padang Sj	32009
18	83726	1000000	28/06/2020	39 Long Island City	Alex Smith	423-6402263	Sate Padang Pas	217804
19	37096	1000000	21/11/2020	40 Jesus Drive	John Williams	792-9002023	Crepe	884593

9:27 michal@XE 250 rows selected in 0.171 seconds

שליפת נתונים

שאלתה 1: חיפוש כל ההזמנות שבוצעו בתאריך מסוים

```

SELECT order_id, totalprice, customer_name, phonenumber, address
FROM CateringOrderDetails
WHERE order_date = TO_DATE('2021-08-21', 'YYYY-MM-DD');

```

SQL Window - CREATE VIEW CateringOrderDetails AS SELECT o.order_id, o.totalprice, o.order_date, o.address, c...

SQL Output Statistics

```

SELECT order_id, totalprice, customer_name, phonenumber, address
FROM CateringOrderDetails
WHERE order_date = TO_DATE('2021-08-21', 'YYYY-MM-DD');

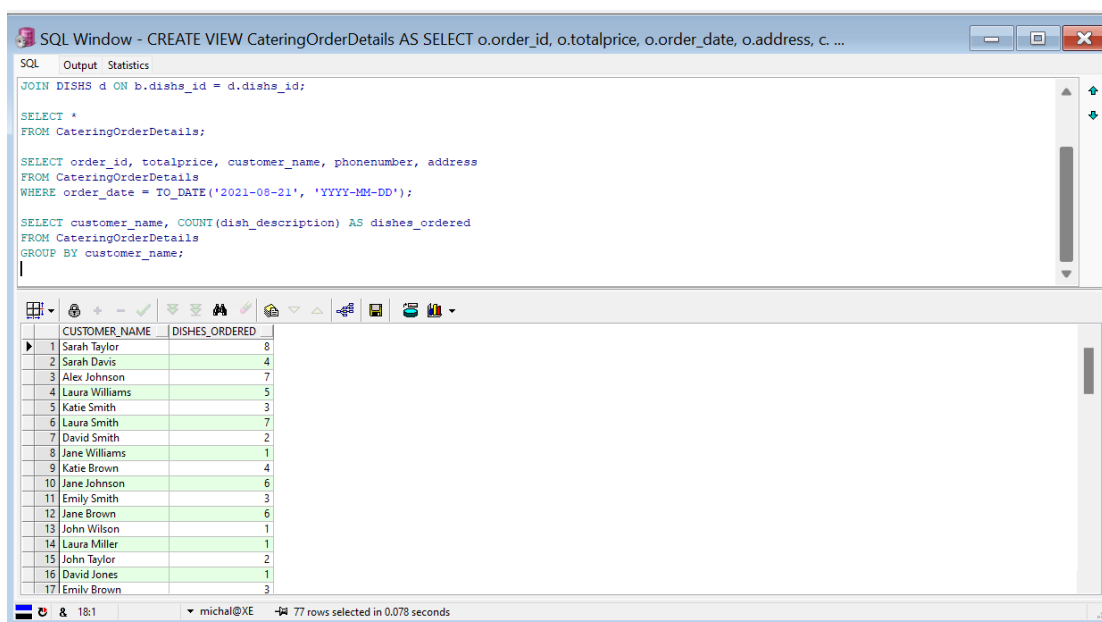
```

	ORDER_ID	TOTALPRICE	CUSTOMER_NAME	PHONENUMBER	ADDRESS
1	40160	1000000	Katie Brown	659-2102763	56 Maintenon Ave

12:26 michal@XE 1 row selected in 0.016 seconds

שאלתה 2: ספירת המנות שהוזמנו לכל לקוח

```
SELECT customer_name, COUNT(dish_description) AS dishes_ordered
FROM CateringOrderDetails
GROUP BY customer_name;
```



The screenshot shows an SQL Window titled "SQL Window - CREATE VIEW CateringOrderDetails AS SELECT o.order_id, o.totalprice, o.order_date, o.address, c. ...". The window has tabs for "SQL", "Output", and "Statistics". The SQL tab is active, displaying the following query:

```
JOIN DISHS d ON b.dishes_id = d.dishes_id;

SELECT *
FROM CateringOrderDetails;

SELECT order_id, totalprice, customer_name, phonenumber, address
FROM CateringOrderDetails
WHERE order_date = TO_DATE('2021-08-21', 'YYYY-MM-DD');

SELECT customer_name, COUNT(dish_description) AS dishes_ordered
FROM CateringOrderDetails
GROUP BY customer_name;
```

Below the query, the results are displayed in a table with two columns: "CUSTOMER_NAME" and "DISHES_ORDERED". The table contains 17 rows of data, with the first row highlighted in blue. The status bar at the bottom indicates "77 rows selected in 0.078 seconds".

	CUSTOMER_NAME	DISHES_ORDERED
1	Sarah Taylor	8
2	Sarah Davis	4
3	Alex Johnson	7
4	Laura Williams	5
5	Katie Smith	3
6	Laura Smith	7
7	David Smith	2
8	Jane Williams	1
9	Katie Brown	4
10	Jane Johnson	6
11	Emily Smith	3
12	Jane Brown	6
13	John Wilson	1
14	Laura Miller	1
15	John Taylor	2
16	David Jones	1
17	Emily Brown	3