

חברת מסיבות



שם המגיש: ירדן בצרי

ת.ז: 213217268

תאריך הגשה: 3/5/2022

שם המנחה: רות ברקוביץ אשר

פרטי מכללה: מכללת הדסה נעורים

סמל מוסד: 470146

3.....	הצעת פרויקט:
4.....	תיאור הארגון:
5.....	מבנה הארגון:
6.....	היררכיה ארגונית:
6.....	מהות המשימה והרקע שלה:
7.....	מטרות המערכת המוצעת:
7.....	דרישות פונקציונליות:
8.....	דרישות לא פונקציונליות:
8.....	תיחום המערכת:
9.....	תרשימים:
15.....	יצירת טבלאות:
24.....	שאלות בעלות רציונל עסקי:
38.....	סביבת הפיתוח:

הצעת פרויקט:

משרד החינוך והתרבות

הצעת נושא לפרויקט SQL לטכנאים
במגמת הנדסת תוכנה

סמל מוסד: 470146

ת"ז: 213217268

שם הסטודנט: ירדן בצרי

שנה"ל: תשפ"ב 2022

שם המנחה: רוית ברקוביץ-אשר

תאור הנושא: ארגון וניהול אירועי מסיבות. המערכת תכלול מידע עבור כל הנוכחים במסיבה ואת כל המידע הקשור לארגון וניהול המסיבה.

סביבת עבודה: sql server

אישור הצעת פרויקט:

 חתימה:

שם המנחה: רוית ברקוביץ-אשר

שם הסטודנט: ירדן בצרי חתימה: 

תיאור הארגון:

מהות הארגון –

ניהול מסיבות ברחבי הארץ.

ארגון זה הינו ארגון עסקי אשר מספק שירות לבעלי המועדונים והאולמות למטרות רווח.

מטרות הארגון –

מטרת על:

להביא רווח מרבי לחברת המסיבות

מטרות משנה:

- הגדלת מספר המסיבות אשר לוקחות חלק בארגון.
- פרסום ושיווק העסק באמצעי המדיה השונים.
- סיפוק צורכי הלקוחות על ידי פיתוח מערכת חכמה וידידותית.
- המלצות טובות על החברה בקרב בעלי העסקים.
- ניהול הספקים והעובדים בחברה בצורה מספקת עבור בעלי העסקים.

סביבת הארגון –

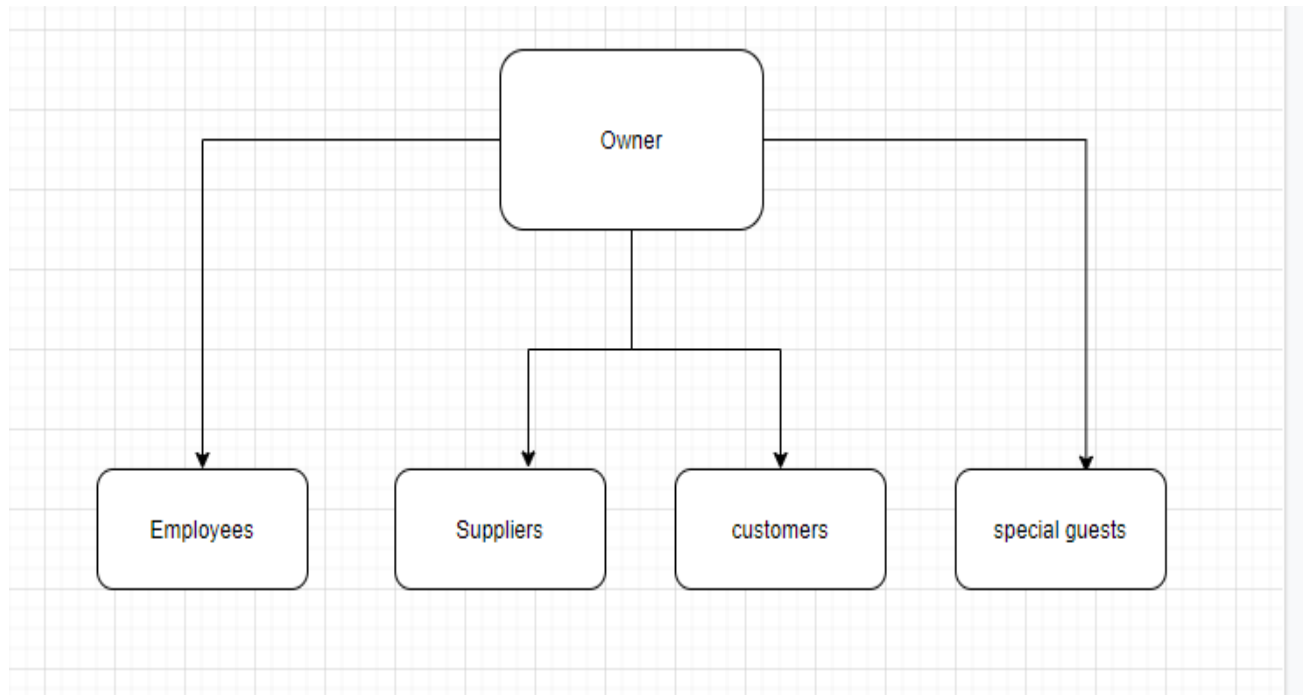
ארגון חברת המסיבות יספק מערכת אשר תפנה אל:

- בעלי העסקים.
- לקוחות.
- ספקים.
- עובדים.
- אורחים מיוחדים.

מבנה הארגון:

שם הגורם:	תיאור הגורם:	דרג:	רמת מעורבות:
בעל העסק	עומד בראש העסק, אחראי להוסיף למערכת מסיבות חדשות במועדים הרצויים. בנוסף יכול לאשר או לסרב להצעות של אורחים מיוחדים למסיבה. קובע מחירים עבור ספקים למוצרים מסוימים ומחיר לשעה עבור עובדים בעבודה מסוימת.	ניהולי	יזם + משתמש
חברת האשראי	ישות חיצוני אשר אחראית על חיוב התשלומים והוצאת חשבוניות. ישות זו לא משתמשת במערכת.	תפעולי	גורם חיצוני
ספק	יכול לרשום את עצמו לספק את המוצר אשר אותו הוא מספק במידה ועדיין יש צורך במוצר.	תפעולי	משתמש
עובד	יכול לרשום את עצמו בתאריך מסוים לעבוד במסיבה מסוימת במידה ועדיין יש צורך בעובד מהסוג המתאים.	תפעולי	משתמש
לקוח	יכול לרשום את עצמו ואת חבריו (אם מעוניין) למסיבה מסוימת במידה ועומד בדרישות הגיל והמכירה לא sold out .	תפעולי	משתמש
אורח מיוחד	יכול לרשום את עצמו במערכת ולתת הצעת מחיר להשתתפות במסיבה מסוימת.	תפעולי	משתמש

היררכיה ארגונית:



מהות המשימה והרקע שלה:

- ✚ מסיבות הם חלק עיקרי מהחיים של אנשים ברחבי הארץ ובעולם.
- ✚ על מנת שהמסיבה תתרחש על הצד הטוב ביותר יש להקים מערכת אשר תספק ניהול לכל הישויות הדרושות במסיבה.
- ✚ מהות המשימה תהיה תמיכה בכל אירועי המסיבות ברחבי הארץ ובכל הגורמים השונים הלוקחים חלק במסיבה.

מטרות המערכת המוצעת:

- ✚ המטרה היא לממש מערכת שתוכל להתאים לכל מסיבה כך שתינתן האפשרות ללקוחות להזמין ברטיסים למסיבה.
- ✚ המערכת תצטרך לדאוג למזער את הטעויות עם הלקוחות המזמינים כמו לאשר לקוחות למסיבה בתפוסה מלאה או לאשר לקוחות למסיבה למרות שהם בגיל הלא מתאים.
- ✚ בנוסף המערכת תצטרך לנהל את הספקים ולדאוג שכל המוצרים השונים יגיעו למסיבה.
- ✚ המערכת תצטרך לתמוך בניהול של העובדים במערכת.
- ✚ המערכת תצטרך לוודא שלכל מסיבה יש ספקים לכל המוצרים השונים ועובדים בכל הסוגים.
- ✚ המערכת תנהל את שיבוץ האורחים המיוחדים לכל מסיבה.

דרישות פונקציונליות:

- ✚ המערכת תתמוך בניהול הלקוחות והזמנותיהם למסיבה.
- ✚ המערכת תתמוך בשליחת תמונות המסיבה ללקוחותיה.
- ✚ המערכת תתמוך בניהול הספקים ומתן אפשרות רישום למסיבה אך ורק אם יש צורך במוצר אותו מספקים.
- ✚ המערכת תתמוך בניהול עובדים, תדאג להעלות את המשכורת לעובדים הפעילים ביותר, בנוסף תוכל להציג דוח של הצלם הפעיל ביותר במסיבה.
- ✚ המערכת תתמוך בהפקת דוח לבעל העסק שיציג לו את הצעות המחיר הטובות ביותר ואת המסיבות הקרובות ביותר כך שיוכל בזריזות לשבץ אורח מיוחד במסיבה.

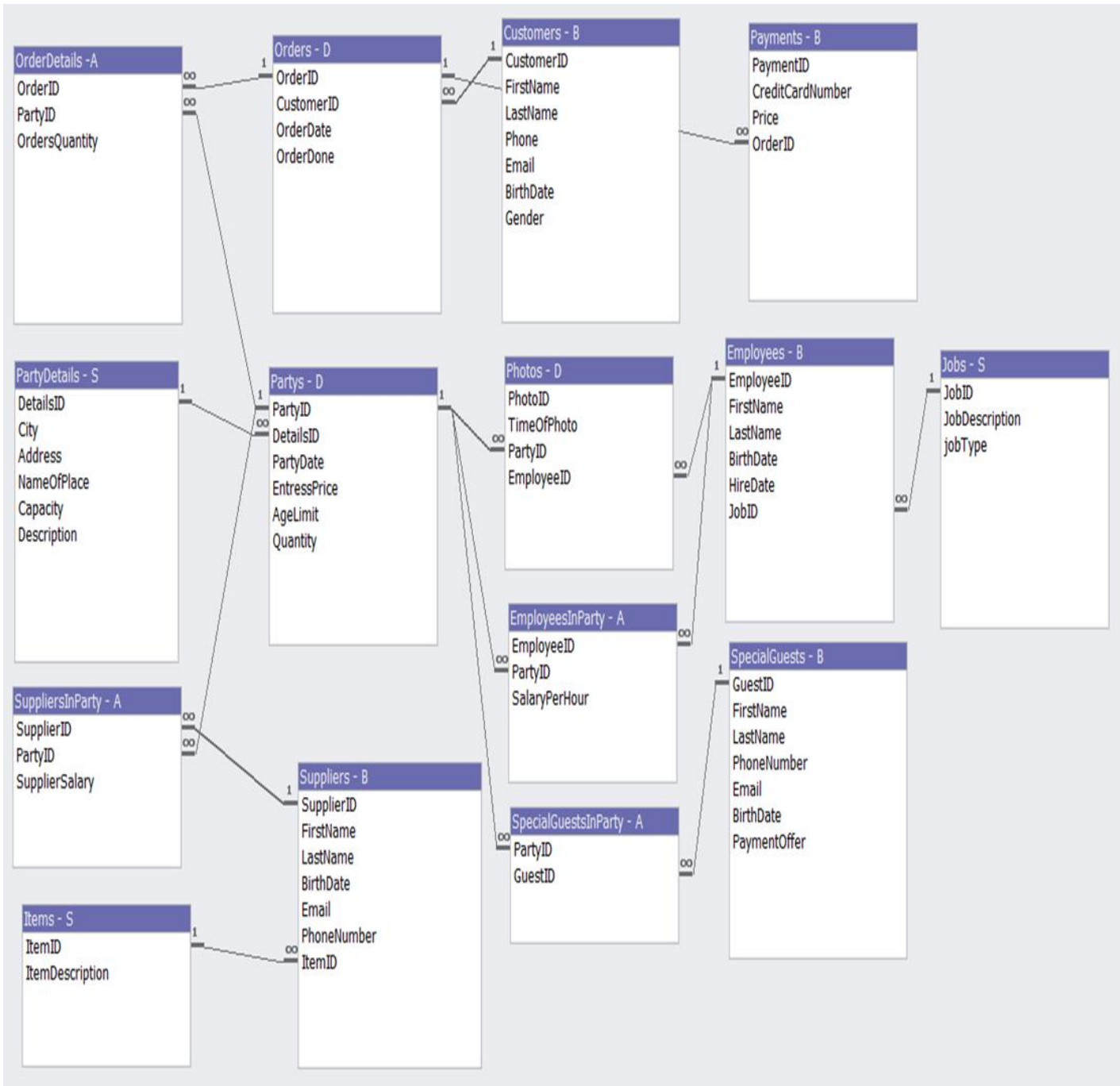
דרישות לא פונקציונליות:

- ✚ המערכת לא תתמוך בחיוב ובקבלת חשבונות. דבר אשר יעשה על ידי ישות חיצונית שהיא חברת האשראי.
- ✚ המערכת לא תתמוך בהכנסה של לקוחות נוספים למסיבה במקרה ויש תפוסה מלאה מבחינת כרטיסים שהוזמנו אך אנשים שהזמינו כרטיס לא הגיעו למסיבה.
- ✚ המערכת לא תתמוך במתן הצעת מחיר לספק ולקוח, הצעת המחיר תגיע דרך בעל העסק מחוץ למערכת.

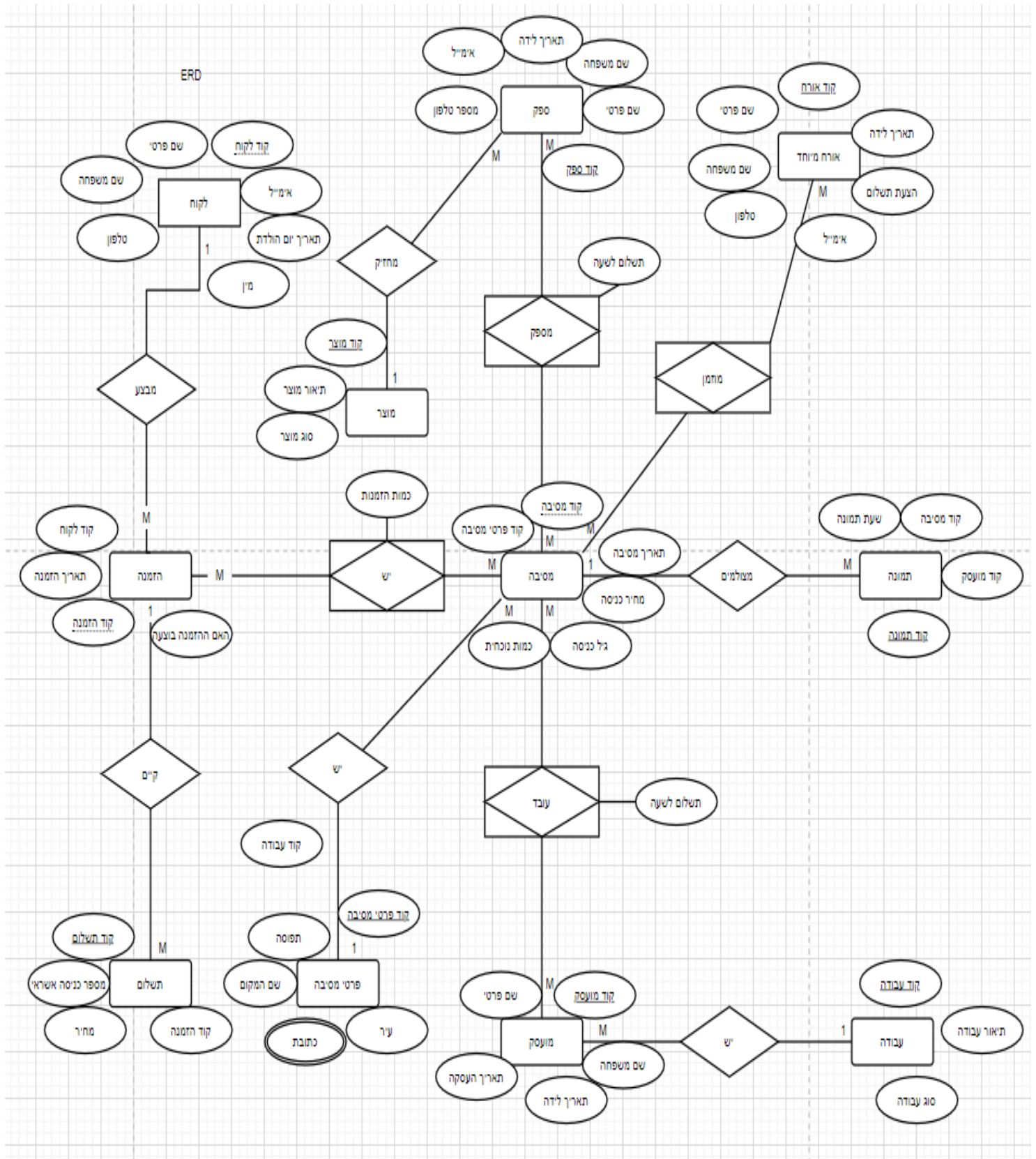
תיחום המערכת:

- ✚ המערכת תנהל את כל הישויות : ספקים, עובדים, לקוחות, אורחים מיוחדים.
- ✚ המערכת תספק מידע רחב לבעל העסק, כמו דוחות של מסיבות בתפוסה חלקית, עובדים מצטיינים ועוד.
- ✚ המערכת תספק לספקים ולעובדים מידע על המסיבות שיש צורך בסוג העבודה או המוצר שהם עוסקים בו.
- ✚ המערכת תעזור לבעל העסק לשבץ אורחים מיוחדים למסיבות.
- ✚ לסיום, המערכת גם תעדכן נתונים לבדה כמו, סכום סופי לתשלום ללקוחות, אישור הזמנה, העלה לעובדים מצטיינים במשכורת ועוד.

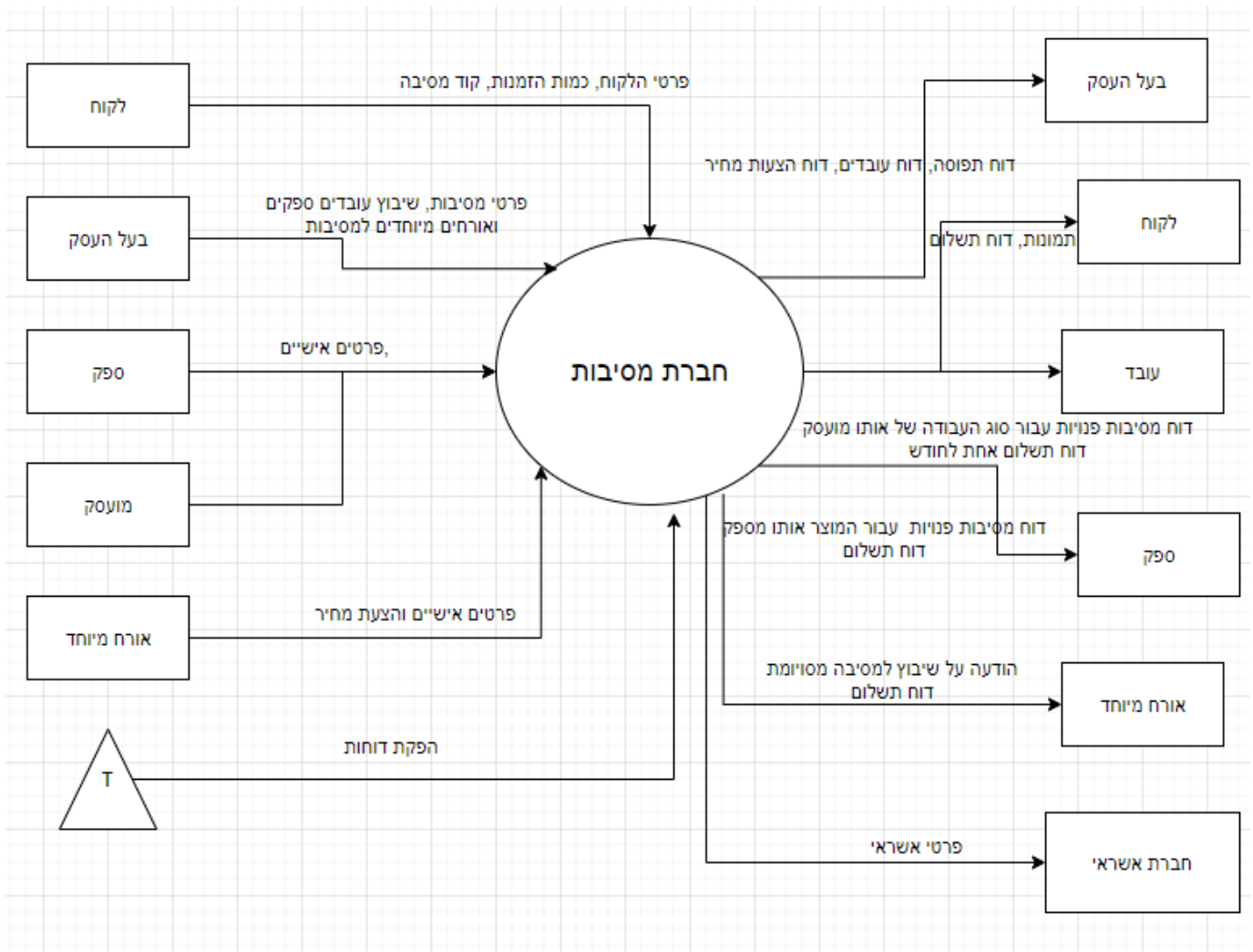
DSD



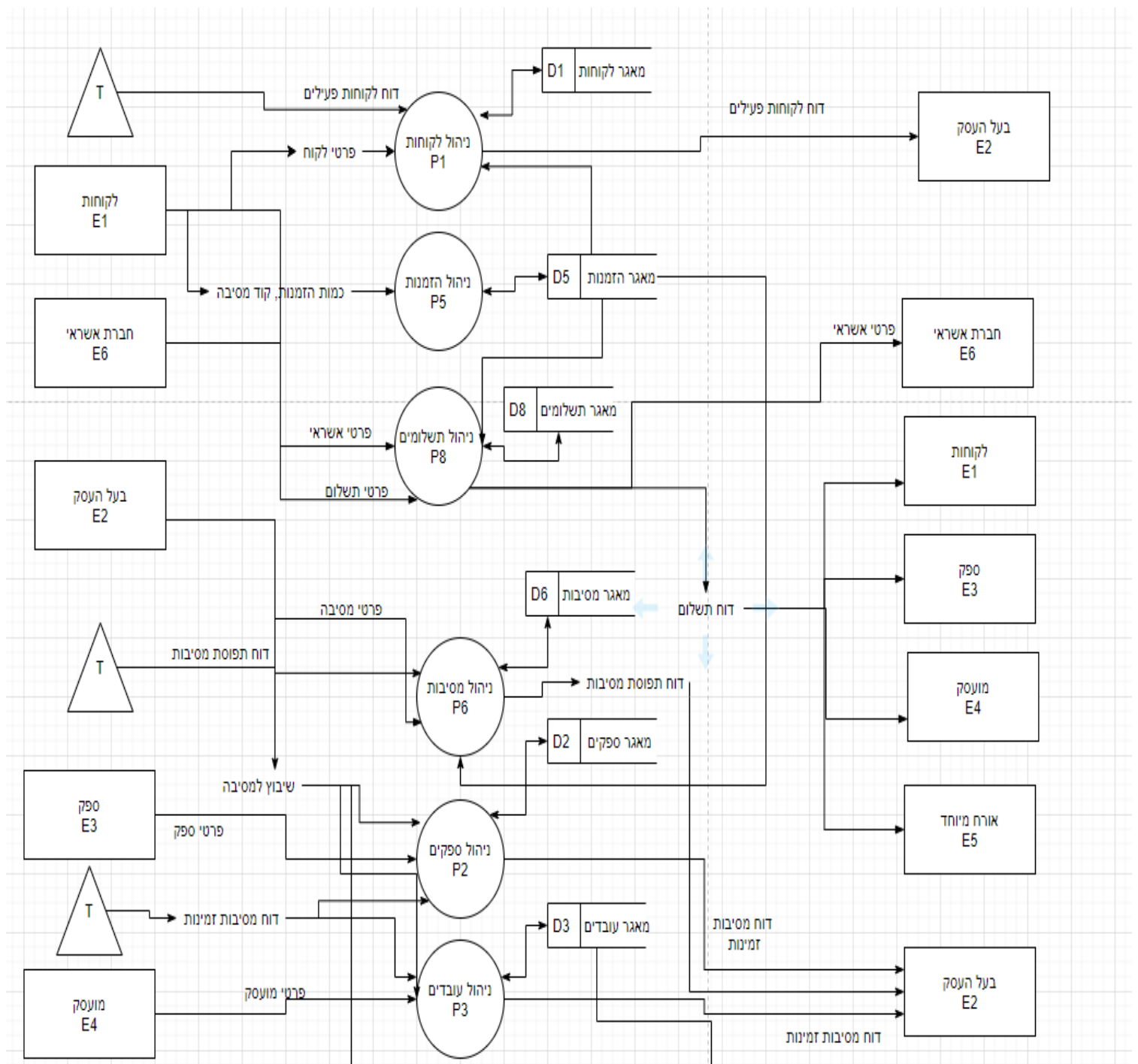
ERD

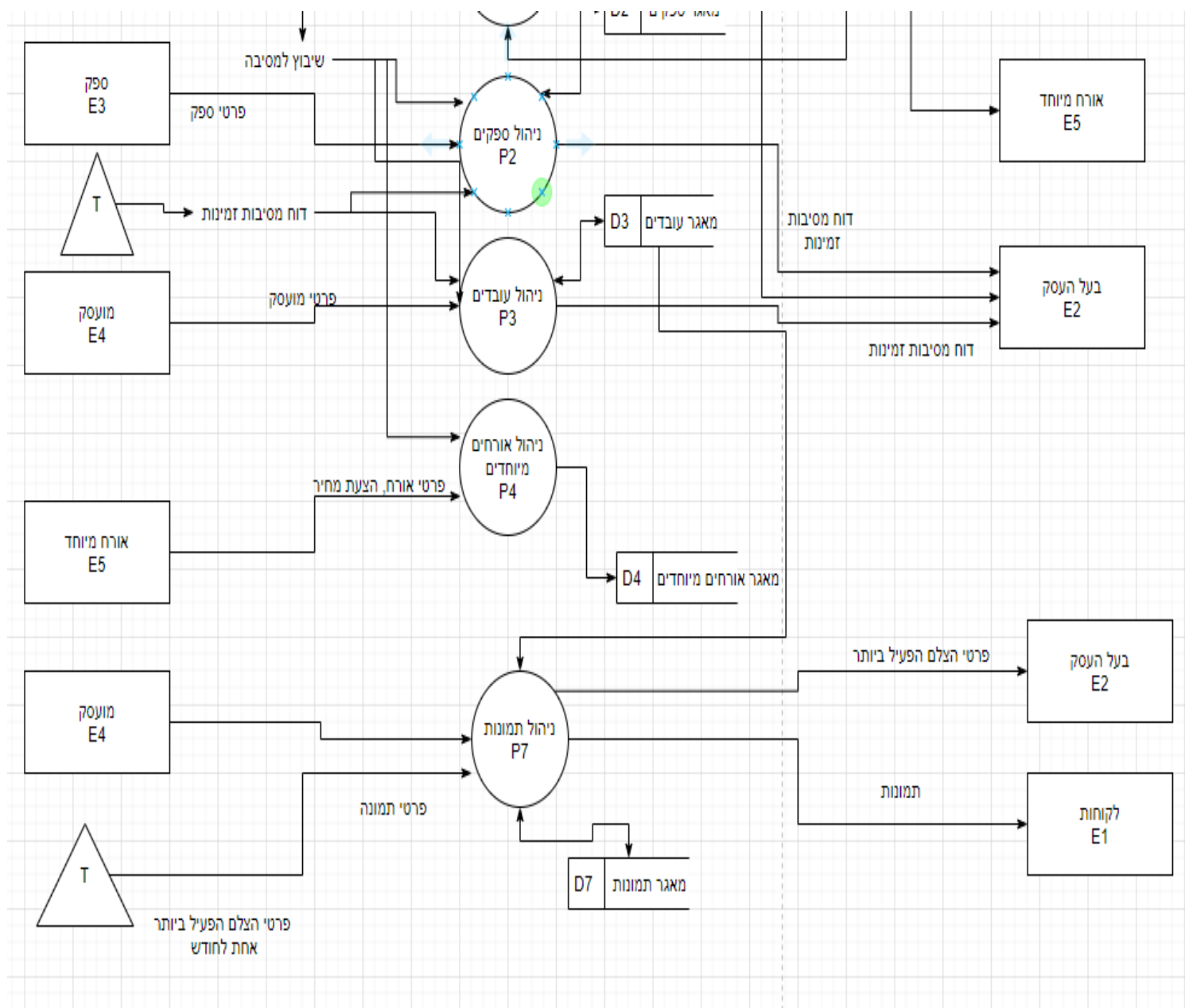


DFD DETAILS



DFDO





DFD1 – ניהול לקוחות

DFD1 – ניהול ספקים

DFD1 – ניהול עובדים

DFD1 – ניהול אורחים מיוחדים

DFD1 – ניהול הזמנות

DFD1 – ניהול מסיבות

DFD1 – ניהול תמונות

DFD1 – ניהול תשלומים

Create PartyDetails:

```
CREATE TABLE PartyDetails(  
DetailsID AUTOINCREMENT NOT NULL,  
City varchar(20),  
Address varchar(30),  
NameOfPlace varchar(15),  
Capacity int NOT NULL,  
Quantity int NOT NULL,  
AgeLimit int,  
Description varchar(100),  
PRIMARY KEY(DetailsID)  
);
```

Create Partys:

```
CREATE TABLE Partys(  
    PartyID AUTOINCREMENT NOT NULL ,  
    DetailsID int NOT NULL,  
    PartyDate DATE NOT NULL,  
    EntressPrice currency NOT NULL,  
    PRIMARY KEY(PartyID),  
    CONSTRAINT FK_DetailsID FOREIGN KEY (DetailsID)  
    REFERENCES PartyDetails(DetailsID));
```

Create Photos:

```
CREATE TABLE Photos(  
    PhotoID AUTOINCREMENT NOT NULL,  
    TimeOfPhoto TIME,  
    PartyID int NOT NULL,  
    PRIMARY KEY (PhotoID),  
    CONSTRAINT FK_PhotoPartyID FOREIGN KEY(PartyID)  
    REFERENCES Partys(PartyID)  
);
```


Create Customers:

```
CREATE TABLE Customers
(CustomerID AUTOINCREMENT NOT NULL,
FirstName varchar(15),
LastName varchar(15),
Phone varchar(10),
Email varchar(30),
BirthDate DATE,
Gender varchar(6),
PRIMARY KEY(CustomerID)
);
```

Create Orders:

```
CREATE TABLE Orders(
OrderID AUTOINCREMENT NOT NULL,
CustomerID int NOT NULL,
OrderDate DATE,
OrderDone int,
PRIMARY KEY(OrderID),
CONSTRAINT FK_CustomerID FOREIGN KEY (CustomerID)
REFERENCES Customers(CustomerID)
);
```

Create OrderDetails:

```
CREATE TABLE OrderDetails(  
    OrderID int NOT NULL,  
    PartyID int NOT NULL,  
    OrdersQuantity int NOT NULL,  
    CONSTRAINT PK_OrderDetails PRIMARY KEY (OrderID, PartyID),  
    CONSTRAINT FK_OrderIDDetails FOREIGN KEY (OrderID)  
    REFERENCES Orders(OrderID),  
    CONSTRAINT FK_PartyIDDetails FOREIGN KEY (PartyID)  
    REFERENCES Partys(PartyID)  
);
```

Create Payments:

```
CREATE TABLE Payments(  
    PaymentID AUTOINCREMENT NOT NULL,  
    CreditCardNumber int NOT NULL,  
    Price currency,  
    OrderID int NOT NULL,  
    PRIMARY KEY (PaymentID),  
    CONSTRAINT FK_OrderID FOREIGN KEY (OrderID)  
    REFERENCES Orders(OrderID)  
);
```

Create Jobs:

```
CREATE TABLE Jobs(  
  JobID AUTOINCREMENT NOT NULL,  
  JobDescription varchar(70),  
  jobType varchar(10),  
  PRIMARY KEY (JobID)  
);
```

Create Employees:

```
CREATE TABLE Employees(  
  EmployeeID AUTOINCREMENT NOT NULL,  
  FirstName varchar(15),  
  LastName varchar(15),  
  BirthDate DATE,  
  HireDate DATE,  
  JobID int NOT NULL,  
  PRIMARY KEY (EmployeeID),  
  CONSTRAINT FK_JobID FOREIGN KEY (JobID)  
  REFERENCES Jobs(JobID)  
);
```

Alter Photos:

```
ALTER TABLE Photos  
ADD CONSTRAINT FK_EmpID FOREIGN KEY (EmployeeID)  
REFERENCES Employees(EmployeeID);
```

Create EmployeesInParty:

```
CREATE TABLE EmployeesInParty(  
EmployeeID int NOT NULL,  
PartyID int NOT NULL,  
SalaryPerHour currency,  
CONSTRAINT PK_EmployeesInParty PRIMARY KEY(EmployeeID,  
PartyID),  
CONSTRAINT FK_EmployeeIDParty FOREIGN KEY (EmployeeID)  
REFERENCES Employees(EmployeeID),  
CONSTRAINT FK_PartyIDEmployees FOREIGN KEY (PartyID)  
REFERENCES Partys(PartyID)  
);
```

Create SpecialGuests:

```
CREATE TABLE SpecialGuests(  
  GuestID AUTOINCREMENT NOT NULL,  
  FirstName varchar(15),  
  LastName varchar(15),  
  PhoneNumber int,  
  Email varchar(20),  
  BirthDate DATE,  
  PaymentOffer currency NOT NULL,  
  PRIMARY KEY(GUESTID)  
);
```

Create SpecialGuestsInParty:

```
CREATE TABLE SpecialGuestsInParty(  
  PartyID int NOT NULL,  
  GuestID int NOT NULL,  
  PaymentOffer currency NOT NULL,  
  CONSTRAINT PK PRIMARY KEY (PartyID, GuestID),  
  CONSTRAINT FK_PartyID FOREIGN KEY (PartyID)  
  REFERENCES Partys(PartyID),  
  CONSTRAINT FK_GuestID FOREIGN KEY (GuestID)  
  REFERENCES SpecialGuests (GuestID)  
);
```

Create Suppliers:

```
CREATE TABLE Suppliers(  
SupplierID AUTOINCREMENT NOT NULL,  
FirstName varchar(15),  
LastName varchar(15),  
BirthDate DATE,  
Email varchar(30),  
PhoneNumber int,  
ItemID int NOT NULL,  
PRIMARY KEY (SupplierID)  
);
```

AlterSuppliers:

```
ALTER TABLE Suppliers  
ADD CONSTRAINT FK_ItemID FOREIGN KEY (ItemID)  
REFERENCES Items(ItemID);
```

Create Items:

```
CREATE TABLE Items(  
ItemID AUTOINCREMENT NOT NULL,  
ItemDescription varchar(70),  
PRIMARY KEY (ItemID)  
);
```

Create SuppliersInParty:


```
CREATE TABLE SuppliersInParty(  
  SupplierID int NOT NULL,  
  PartyID int NOT NULL,  
  SupplierSalary currency,  
  CONSTRAINT PK_SuppliersInParty PRIMARY KEY (SupplierID,  
  PartyID),  
  CONSTRAINT FK_SupplierIDParty FOREIGN KEY (SupplierID)  
  REFERENCES Suppliers (SupplierID),  
  CONSTRAINT FK_PartyIDSupplier FOREIGN KEY (PartyID)  
  REFERENCES Partys(PartyID)  
);
```

שאלתה 1 –

מטרת השאלתה: 

לאשר הזמנה של לקוח. 

הלקוח יאושר במידה ועומד בדרישות הגיל של המסיבה
ותפוסת האנשים במקום עולה על כמות הכרטיסים שנמכרו
עד כה.

הרציונל העסקי: 

לעמוד בהבטחה של גיל מסוים כך שיישמר אמינות עם
הלקוחות ולא לאשר סתם אנשים שבסופו של דבר יתגלה
שאין להם מקום במסיבה.

Approve order:

```
UPDATE Orders, OrderDetails, Partys, Customers,  
PartyDetails SET Orders.OrderDone = true,  
Partys.Quantity = Partys.Quantity +1  
WHERE Customers.CustomerID =  
Orders.CustomerID  
AND Orders.OrderID = OrderDetails.OrderID  
AND OrderDetails.PartyID =Partys.PartyID  
AND DateDiff("yyyy", Customers.BirthDate, Now())  
>= Partys.AgeLimit  
AND Partys.DetailsID = PartyDetails.DetailsID  
AND PartyDetails.Capacity > Partys.Quantity;
```


שאלתה 2 –

מטרת השאלתה:

להציג בפני בעל העסק את הצלם שהיה הכי פעיל במסיבה
בכך שצילם הכי הרבה תמונות.

הרציונל העסקי:

במידה ובעל העסק יגלה מי הצלם המצטיין ויחליט להביא לו
צופר או בונוס, הגיוני ששאר הצלמים יחליטו לגלות יותר
רצינות בעבודתם ולצלם יותר תמונות וככה הסיכוי שלקוח
יופיע בתמונה יגדל מה שיגרום לתחושה נעימה וטובה בקרב
הלקוחות עם פוטנציאל גבוה יותר לחזור למקום פעם
נוספת.

Best photographer:

```
SELECT E.FirstName, E.LastName,  
Count(P.EmployeeID) AS MaximumPictures  
FROM Employees AS E, Photos AS P, Jobs AS J  
WHERE E.EmployeeID = P.EmployeeID  
AND J.JobID = E.JobID  
GROUP BY E.FirstName, LastName  
HAVING Count(P.EmployeeID) =  
(SELECT MAX(BestPhotographer)  
FROM(  
SELECT E.FirstName, E.LastName,  
COUNT(P.EmployeeID) AS BestPhotographer  
FROM Employees E, Photos P, Jobs J  
WHERE E.EmployeeID = P.EmployeeID  
AND J.JobID = E.JobID  
GROUP BY E.FirstName, LastName));
```

שאלתה 3 –

✚ מטרת השאלתה:

✚ לעזור לבעל העסק לשבץ אורחים מיוחדים למסיבות.

✚ השאלתה תחזיר לבעל העסק את כל הקומבינציות של שלוש המסיבות הקרובות יחד עם האורחים המיוחדים בעלי הצעת המחיר הזולה ביותר.

✚ רציונל עסקי:

✚ הרציונל הוא שבתור חברת מסיבות חדשה, האינטרס הוא להביא את האורחים המיוחדים הזולים ביותר, ולחשוף בפני בעל העסק הוא העובדים תחתיו את שלושת המסיבות הקרובות כדי שייקבע למסיבות אלה אורח מיוחד בהקדם האפשרי.

✚ המטרה היא לא לקבוע במקום בעל העסק את האורחים המיוחדים למסיבה מסוימת אלה לתת לבעל העסק עזרה / הכוונה כדי להגיע להצלחה העסקית הטובה ביותר.

Schedule special guest:

```
SELECT DISTINCT SG.GuestID, P.PartyID
FROM SpecialGuests SG, Partys P
WHERE SG.PaymentOffer = ANY
(SELECT MinPayment
FROM(SELECT TOP 3 SG.GuestID,
Min(SG.PaymentOffer) AS MinPayment
FROM SpecialGuests SG
GROUP BY SG.GuestID))
AND P.PartyDate = ANY
(SELECT MinDate
FROM(SELECT TOP 3 P.PartyID, Min(P.PartyDate)
AS MinDate
FROM Partys P
GROUP BY P.PartyID
HAVING Date() < Min(P.PartyDate))));
```

שאלתה 4 -

מטרת השאלתה:

לעדכן את טבלת התשלומים בהתאם לכמות ההזמנות שביצע משתמש בו זמנית.

לדוגמא: אם משתמש רכש שני כרטיסים למסיבה אז בטבלת התשלומים צריך להופיע למשתמש זה תשלום בעבור מחיר המסיבה * 2.

רציונל עסקי:

הרציונל העסקי הוא לחסוך מבעל העסק או מאנשים תחתיו להזין באופן ידני את המחיר בעבור כרטיס למסיבה ככה שהמערכת תעשה זאת באופן אוטומטי לאחר ביצוע ההזמנה.

Update payment price:

UPDATE Payments, Partys, Orders, OrderDetails

SET Payments.Price = Partys.EntressPrice *
OrderDetails.OrdersQuantity


WHERE Payments.OrderID = Orders.OrderID


AND OrderDetails.OrderID = Orders.OrderID


AND OrderDetails.PartyID = Partys.PartyID;

שאלתה 5 –

מטרת השאלתה: 

להציג את כל המקומות שהמסיבות שהתקיימו בו היו 
בתפוסה אשר קטנה מ-60 אחוז שלוש פעמים לפחות.

הרציונל העסקי: 


הרציונל הוא לעקוב אחר מקומות של מסיבות שלא באים 
אליהם מספיק אנשים באופן קבוע כדי או לפרסם את
מקומות אילו יותר, או לוותר על נוכחותם בחברת המסיבות.


Worst partys:

```
SELECT PD.DetailsID, PD.NameOfPlace
FROM PartyDetails PD
WHERE PD.NameOfPlace = ANY(
SELECT PD.NameOfPlace
FROM PartyDetails PD, Partys P
WHERE PD.DetailsID = P.DetailsID
AND Date() > P.PartyDate
AND P.Quantity * 0.4 <= PD.Capacity
GROUP BY PD.NameOfPlace
HAVING COUNT(PD.DetailsID) >= 3);
```

שאלתה 6 –

מטרת השאלתה: 

להציג לכל ספק את המסיבות בהם המוצר שאותו הוא 
מייבא אינו מסופק למסיבה. כלומר לתת לספק רשימה של
מסיבות שבהם יש צורך במוצר שאותו הוא מספק.

הרציונל העסקי: לאפשר לספק לראות מראש את המסיבות 
הזמינות בשבילו וכך לחסוך זמן של שיחת טלפון עם בעל
העסק ושיבוץ באופן מהיר יותר.

Available partys for suppliers:

```
SELECT DISTINCT S.SupplierID, P.PartyID
FROM Suppliers S, Partys P, (
    SELECT SP.PartyID AS PartyID, S.ItemID AS ItemID
    FROM SuppliersInParty SP, Suppliers S, Items I
    WHERE SP.SupplierID = S.SupplierID
    AND S.ItemID = I.ItemID
    GROUP BY SP.PartyID, S.ItemID) DifItems
WHERE P.PartyID = DifItems.PartyID
AND S.ItemID <> DifItems.ItemID;
```

שאלתה 7 –

מטרת השאלתה:

להעלות את שכר העובדים בחמישה אחוזים במידה ובחודש שעבר עבדו לפחות ב25 אחוזים מהעבודות בחברה (בהנחה שביום יש מקסימום 4 מסיבות) ושכרם לא הגיע לשכר עובד מקסימלי שהוחלט כ55 שקל לשעה. השאלתה תבוצע אחת לחודש על ידי המערכת.

רציונל עסקי:


העלאת משכורת בקרב עובדים עשויה לגרום לעובדים להשקיע ולהיות מסורים יותר לעבודתם.
בכך הסיכוי לכמות עובדים מספקת באירוע יהיה ככל הנראה גבוה יותר משמעותית.
כך לצורך העניין, בשביל שעובד שמקבל 30 שקל לשעה יגיע לשכר המקסימלי ייקח לו כחצי שנה.


Promote hard workers:


```
UPDATE EmployeesInParty AS EP
SET EP.SalaryPerHour = EP.SalaryPerHour * 1.1
WHERE EP.EmployeeID = ANY(
SELECT E.EmployeeID
FROM Employees E, (
    SELECT COUNT(EP.PartyID) AS Counter,
    EP.EmployeeID AS EmployeeID
    FROM EmployeesInParty AS EP, Partys P
    WHERE P.PartyID = EP.PartyID
    AND DateDiff("m", P.PartyDate, Date()) = 1
    GROUP BY EP.EmployeeID) PartysCounter
WHERE E.EmployeeID = PartysCounter.EmployeeID
AND PartysCounter.Counter >= (
SELECT COUNT(*)
FROM Partys P
WHERE DateDiff("m", P.PartyDate, Date()) = 1 ) *
0.25);
```

שאלתה 8 –

מטרת השאלתה: 

להציג את המשתמשים שבחודש האחרון הלכו ליותר 
מסיבות ממוצע האנשים שהזמינו כרטיס למסיבה כלשהי
בחודש האחרון.

רציונל עסקי: 

כך תהיה אפשרות לראות מי הם המשתמשים הקבועים 
שהולכים למסיבות וכך לדאוג להפיץ להם מסיבות חדשות
ברשתות החברתיות.

Active customers:

```
SELECT O.CustomerID, C.FirstName, C.LastName,  
COUNT(O.CustomerID) AS NumOfPartys  
FROM Orders O, Customers C, OrderDetails OD,  
Partys P  
WHERE O.CustomerID = C.CustomerID  
AND O.OrderID = OD.OrderID  
AND OD.PartyID = P.PartyID  
AND DateDiff("m", P.PartyDate, Now()) = 1  
GROUP BY O.CustomerID, C.FirstName, C.LastName  
HAVING COUNT(O.CustomerID) > (  
SELECT AVG(CountPartys)  
FROM (SELECT O.CustomerID,  
COUNT(O.CustomerID) AS CountPartys  
FROM Orders O, OrderDetails OD, Partys P  
WHERE O.OrderID = OD.OrderID  
AND OD.PartyID = P.PartyID  
AND DateDiff("m", P.PartyDate, Now()) = 1  
GROUP BY O.CustomerID));
```

שאלתה 9 –

מטרת השאלתה:

להציג עבור כל עובד את המסיבות בהם יש צורך בעובד מהסוג שלו. כלומר לתת רשימה של מסיבות שבהם יש צורך בסוג העבודה של אותו מועסק.

הרציונל העסקי: לאפשר בעל הסק לראות מראש את המסיבות הזמינות בשבילו וכך לחסוך זמן של בדיקה והשיבוץ יתבצע באופן מהיר יותר.

Available partys for employees:

```
SELECT DISTINCT E.EmployeeID, P.PartyID
,FROM Employees AS E, Partys AS P
SELECT EP.PartyID AS PartyID, E.JobID AS JobID)
FROM EmployeesInParty AS EP, Employees AS E,
Jobs AS J
WHERE EP.EmployeeID = E.EmployeeID
AND E.JobID = J.JobID
GROUP BY EP.PartyID, E.JobID) AS DifJobs
WHERE P.PartyID = DifJobs.PartyID AND E.JobID <>
DifJobs.JobID;
```

שאילתה 10 –

מחיקת לקוחות לא פעילים

שאילתה 11 –

מחיקת מסיבות מלפני יותר מחודש

שאילתה 12 –

**הצגת המסיבות הכי יקרות שהגיעו אליהם הכי פחות אנשים
בסדר יורד**

שאילתה 13 –

**הצגת כל העובדים שהחודש הם עובדים שנה על מנת שיהיה
אפשר לתת להם בonus**

שאילתה 14 –

**הצגת כל המסיבות בחודש הקרוב שמתחת לגיל 18 , בדיקה
האם נשלח ספק של אלכוהול בטעות למקום , אם כן למחוק את
השיבוץ של אותו הספק.**

שאילתה 15 –

**הוחלט בחברה מתן הנחה של עשר אחוז למי שמבצע מעל 6
הזמנות בו זמנית, לכן המערכת תעדכן את טבלת התשלומים
במידת הצורך.**

Microsoft Access 2010 -