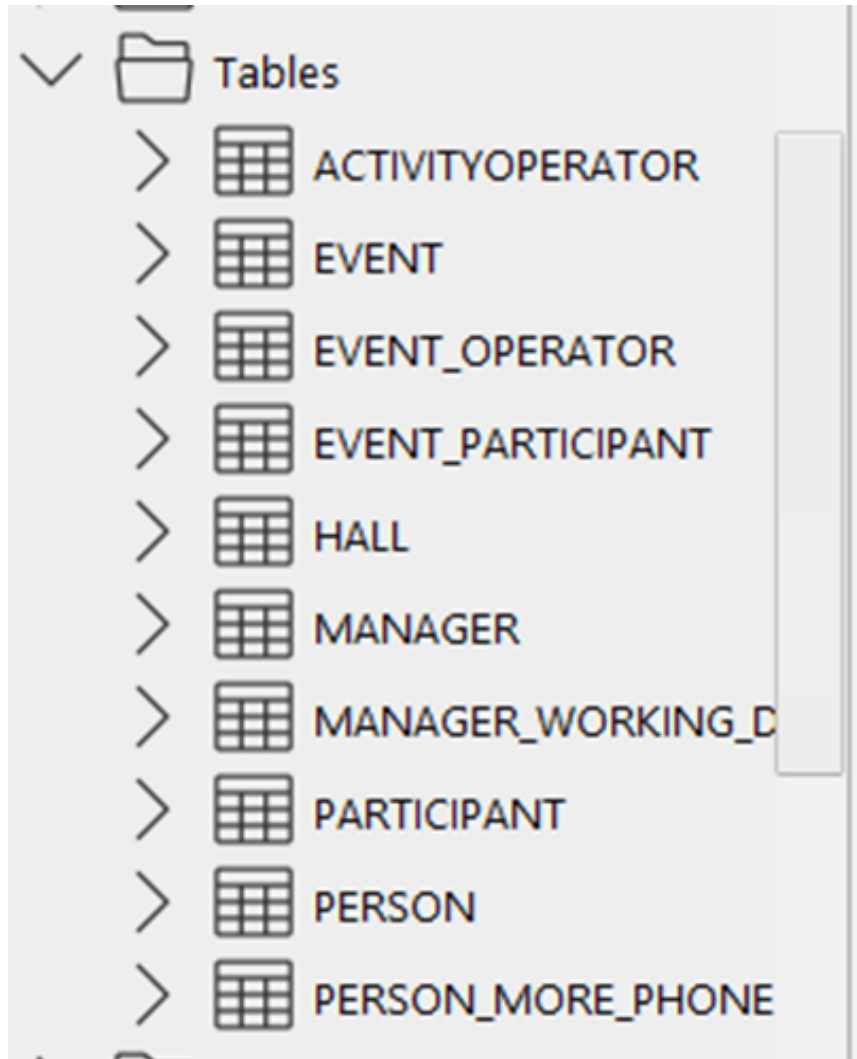


חלק ד

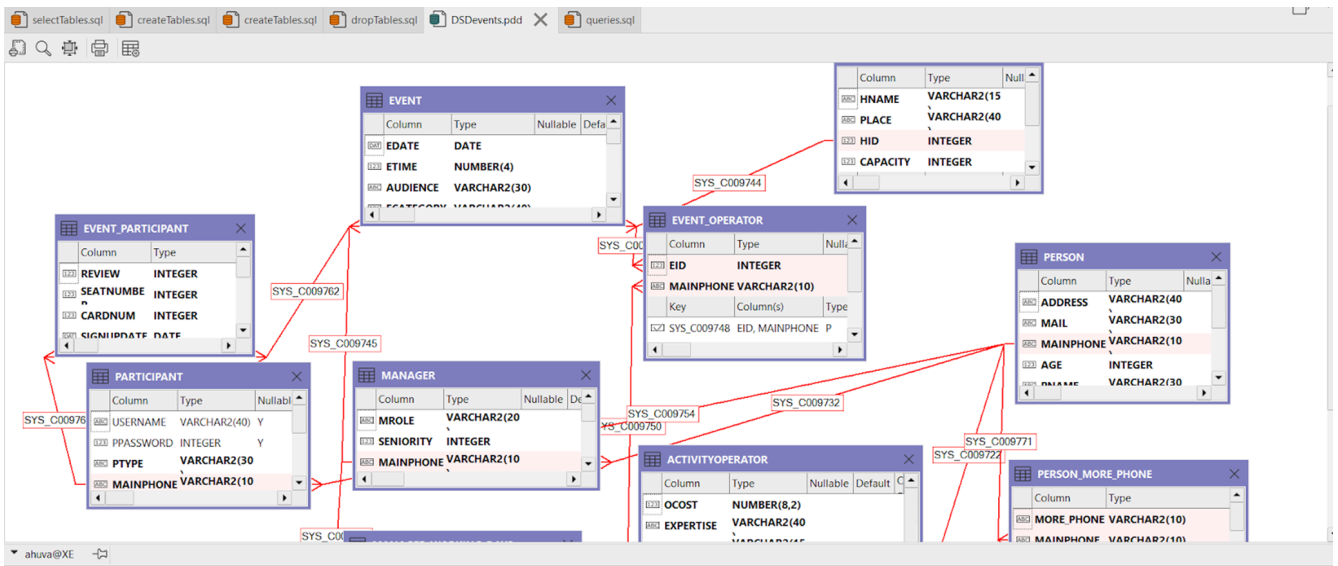
אינטגרציה ומבטים

# מהגיבוי לDSD

## הרצנו את הגיבוי של הפרויקט שקיבלנו בsql וזאת התוצאה:

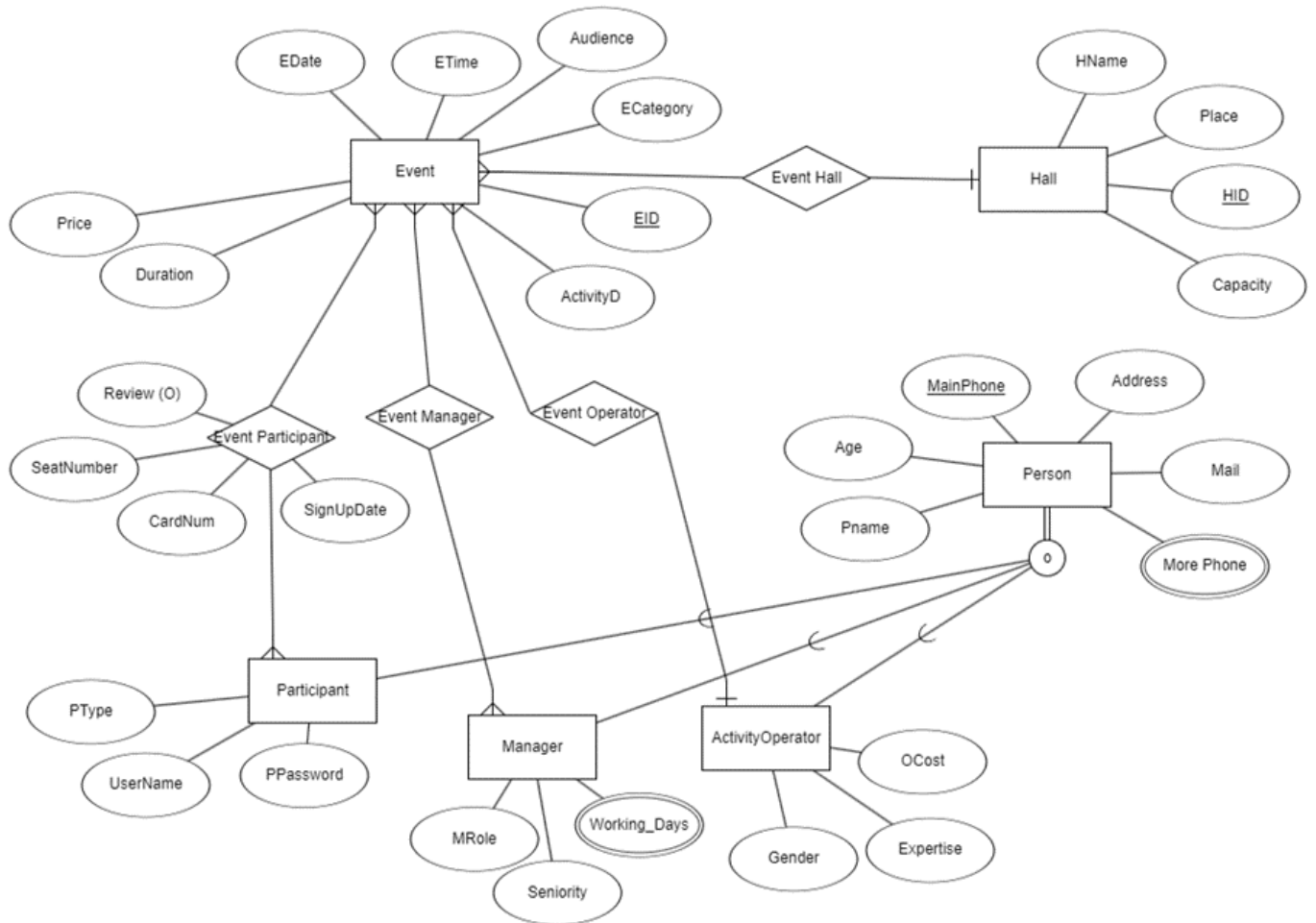


לאחר מכן יצרנו דרך ה־pl/sql בצורה אוטומטית את תרשימים ה־DSD:



# מכד DSD ל ERD

בעקבות ניתוח של התרשים DSD יצרנו את תרשים ה ERD הבא:



בתרשים DSD ראינו שקיבלנו את הטבלאות הבאות:

**טבלת Event:** ובה השדות הבאים: edate, etime, audience, ecategory,

activityid, price, duration, eid, hid, mainphone

המפתח הראשי הוא eid

מפתחות זרים: hid, mainphone.

**טבלת Hall:** ובה השדות הבאים: capacity, hid, hname, place.

המפתח הראשי הוא hid.

מסקנה: קיים קשר של יחיד לרבים בין Event ל Hall (כלומר כל ארוע מתקיים

באולם אחד בלבד)

**טבלת Person:** ובה השדות הבאים: age, mainphone, mail, address, pname,

המפתח הראשי הוא mainphone.

**טבלת Person\_More\_Phone:** ובה השדות הבאים: More\_Phone, MainPhone.

המפתח הראשי הוא More\_Phone, MainPhone

מפתח זר: MainPhone

מסקנה: התכונה More\_Phone היא Multivalued, כלומר אפשר להכניס שם כמה מספרי טלפון.

**טבלת Manager:** ובה השדות הבאים: MainPhone, Seniority, MRole acticityid.

המפתח הראשי הוא MainPhone.

**טבלת Manager\_Working\_Days:** ובה השדות הבאים: MainPhone,

Working\_Days.

המפתח הראשי הוא Working\_Days, MainPhone

מפתח זר: MainPhone

מסקנה: התכונה Working\_Days היא Multivalued, כלומר אפשר להכניס שם כמה ימי עבודה.

מסקנה נוספת: Manager יורש מ-Person. (כי יש לו את אותו מפתח ראשי כמו

person, רק שאצלו זה מפתח זר)

**טבלת ActivityOperator:** ובה השדות הבאים: Gender, Expertise, OCost,

MainPhone.

המפתח הראשי שהוא גם מפתח זר MainPhone.

מסקנה: ActivityOperator יורש מ-Person. (כי יש לו את אותו מפתח ראשי כמו

person, רק שאצלו זה מפתח זר)

**טבלת Participant:** ובה השדות הבאים: PType, PPassword, UserName,

MainPhone.

המפתח הראשי שהוא גם מפתח זר MainPhone.

מסקנה: Participant יורש מ-Person. (כי יש לו את אותו מפתח ראשי כמו

person, רק שאצלו זה מפתח זר)

בנוסף יש לנו 2 טבלאות שמתארות קשרים:

**טבלת Event\_Participant:** ובה השדות הבאים: SeatNumber, Review, CardNum,

SignUpDate, EID, MainPhone.

המפתח הראשי שהוא גם מפתח זר: EID, MainPhone

מסקנה: קיים קשר של רבים לרבים בין event ל participant (מכיוון שבטבלת

Event\_Participant יש מפתח זר מכל אחת מהטבלאות האלו, ואין לה מפתח משל

עצמה)

**טבלת Event\_Operator:** ובה השדות הבאים: EID, MainPhone.

המפתח הראשי שהוא גם מפתח זר: EID, MainPhone

מסקנה: קיים קשר של רבים לרבים בין event ל ActivityOperator.

# ERD משולב

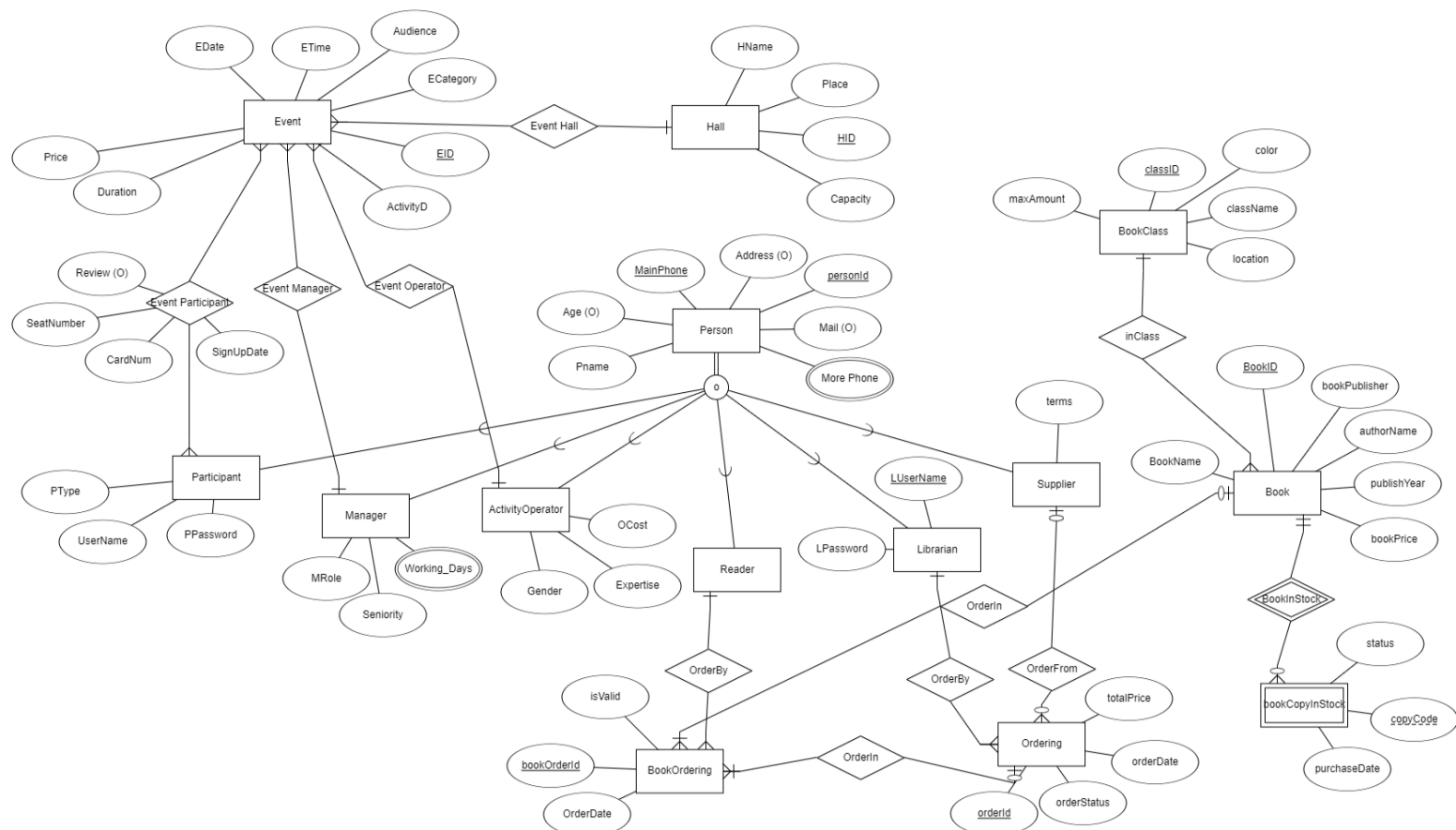
מכיוון שלא היו ישויות משותפות בין הדיאגרמות, כלומר שני הדיאגרמות עסקו בטיפול בישויות שונות לחלוטין, לא היתה לנו כמעט התלבטות לגבי איחוד ישויות, או הוספת קשר מסוים בין ישויות.

התלבטנו האם לאחד את הישות "קורא" ו"משתתף" אך החלטנו לאפשר גם למי שאינו קורא קבוע בספרייה להשתתף בפעילויות הספרייה. ומכיוון ש"מפעיל פעילות" ו"ספק" עוסקים בתחומים שונים וזקוקים לתכונות שונות, לא איחדנו בין השניים.

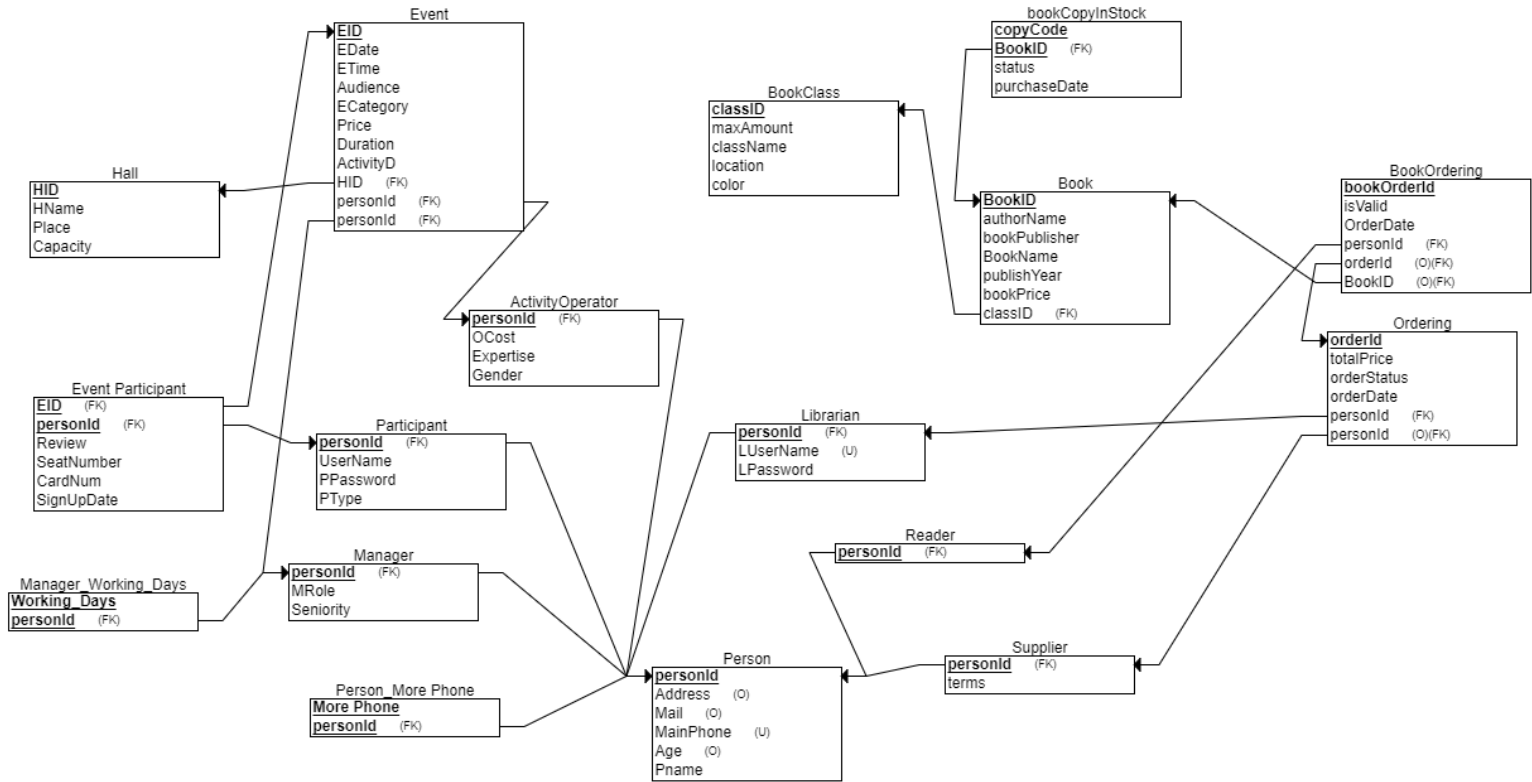
אך ב-ERD השני כל הישויות שמהוות בן אדם באופן כלשהו חוברו לירושה מישות "אדם". ואצלנו לא היתה ירושה כזאת משום שכל דמות החזיקה במזהה יחודי שונה,

אך החלטנו למען הסדר הטוב כן לחבר גם את הישויות המתאימות שלנו לירושה הנ"ל.

תוך הפיכת המזהה הייחודי של הירושה למזהה ייחודי שיהיה לכל אדם, והפיכת המזהים של שם משתמש לספרן ומספר קורא לקורא, לערכים ייחודים בלבד אך שלא מהווים מפתח ראשי. וידאנו שזה לא מפר את צורת 3NF ששמרנו עליה בבניית הדיאגרמה, וחיברנו את הירושה לישויות המתאימות.



# DSD משולב



# תהליך האינטגרציה

תהליך האינטגרציה היה מורכב מאוד, מכיוון שהחלטנו שלמען הסדר הטוב יש להעביר את המפתח הראשי של person להיות personid ולא mainphone, כפי שהיה מלכתחילה.

זה דרש לשנות את המפתח הראשי בPerson, אך לפני כן היה צורך לשנות את כל המפתחות הזרים להם הוא משויך, סה"כ 8 מפתחות זרים שהינם גם מפתחות ראשיים ברוב הטבלאות.

התחלנו בלהוסיף את העמודה personid לטבלאות הרלוונטיות, ולמלא אותה: ראשית הוספנו את העמודה לטבלה person:

```
--Add the column personid to person
```

```
ALTER TABLE Person ADD personId NUMBER;
```

```
DECLARE
```

```
    max_id NUMBER;
```

```
BEGIN
```

```
    SELECT MAX(readernumber) INTO max_id FROM reader;
```

```
    EXECUTE IMMEDIATE 'CREATE SEQUENCE seq_id START WITH ' ||  
(max_id + 1) || ' INCREMENT BY 1';
```

```
END;
```

```
UPDATE Person
```

```
SET personId = seq_id.NEXTVAL;
```

כעת הוספנו אותה בכל שאר הטבלאות הרלוונטיות ומילאנו אותה בהתאמה לעמודה mainphone:

```
--Add the column personid to Person_More_Phone
```

```
ALTER TABLE Person_More_Phone ADD personId NUMBER;
```

```
--Filling the personid column values in the Person_More_Phone table  
according to the personid column values in the person table
```

```
MERGE INTO Person_More_Phone p
```

```
USING person pe
```

```
ON (p.mainphone = pe.mainphone)
```

```
WHEN MATCHED THEN
```

```
UPDATE SET p.personID = pe.personID;
```

--Add the column personid to participant table  
ALTER TABLE participant ADD personId NUMBER;

--Filling the personid column values in the Participant table according to the personid column values in the person table

```
MERGE INTO participant p
USING person pe
ON (p.mainphone = pe.mainphone)
WHEN MATCHED THEN
UPDATE SET p.personID = pe.personID;
```

--Add the column personid to Event\_Participant table  
ALTER TABLE Event\_Participant ADD personId NUMBER;

--Filling the personid column values in the Event\_Participant table according to the personid column values in the person table

```
MERGE INTO Event_Participant ep
USING participant p
ON (ep.mainphone = p.mainphone)
WHEN MATCHED THEN
UPDATE SET ep.personID = p.personID;
```

--Add the column personid to Manager table  
ALTER TABLE Manager ADD personId NUMBER;

--Filling the personid column values in the Manager table according to the personid column values in the person table

```
MERGE INTO Manager m
USING person pe
ON (m.mainphone = pe.mainphone)
WHEN MATCHED THEN
UPDATE SET m.personID = pe.personID;
```

--Add the column personid to ActivityOperator table  
ALTER TABLE ActivityOperator ADD personId NUMBER;

--Filling the personid column values in the ActivityOperator table according to the personid column values in the person table

```
MERGE INTO ActivityOperator a
USING person pe
ON (a.mainphone = pe.mainphone)
WHEN MATCHED THEN
UPDATE SET a.personID = pe.personID;
```



--Add the column personid to Manager\_Working\_Days table  
ALTER TABLE Manager\_Working\_Days ADD personId NUMBER;

--Filling the personid column values in the Manager\_Working\_Days table according to the personid column values in the Manager table  
MERGE INTO Manager\_Working\_Days mwd  
USING Manager m  
ON (mwd.mainphone = m.mainphone)  
WHEN MATCHED THEN  
UPDATE SET mwd.personID = m.personID;

--Add the column personid to Event\_Operator table  
ALTER TABLE Event\_Operator ADD personId NUMBER;

--Filling the personid column values in the Event\_Operator table according to the personid column values in the ActivityOperator table  
MERGE INTO Event\_Operator eo  
USING ActivityOperator ao  
ON (eo.mainphone = ao.mainphone)  
WHEN MATCHED THEN  
UPDATE SET eo.personID = ao.personID;

--Add the column personid to Event table  
ALTER TABLE Event ADD personId NUMBER;

--Filling the personid column values in the Event\_Operator table according to the personid column values in the Manager table  
MERGE INTO Event e  
USING Manager m  
ON (e.mainphone = m.mainphone)  
WHEN MATCHED THEN  
UPDATE SET e.personID = m.personID;

לאחר שהוספנו את העמודה personid, עלינו להכין את הקרקע כדי להפוך אותה למפתח ראשי ומפתח זר בטבלאות הרלוונטיות. לכן קודם כל יש למחוק את האילוץ של מפתח ראשי מהעמודה mainphone, וכדי לעשות זאת יש למחוק קודם כל את כל האילוצים של מפתחות זרים:

--Delete the foreign key constraint from the main\_phone column in all tables

```
ALTER TABLE Event_Participant DROP CONSTRAINT SYS_C009763;
```

```
ALTER TABLE Event DROP CONSTRAINT SYS_C009745;
```

```
ALTER TABLE Event_Operator DROP CONSTRAINT SYS_C009750;
```

```
ALTER TABLE Manager_Working_Days DROP CONSTRAINT  
SYS_C009767;
```

```
ALTER TABLE Manager DROP CONSTRAINT SYS_C009732;
```

```
ALTER TABLE ActivityOperator DROP CONSTRAINT SYS_C009722;
```

```
ALTER TABLE participant DROP CONSTRAINT SYS_C009754;
```

```
ALTER TABLE Person_More_Phone DROP CONSTRAINT SYS_C009771;
```

כעת ניתן לשנות את המפתח הראשי כרצוי בעמודה person, מכיוון שהוא לא תלוי כבר בשום אילוץ מפתח זר נוסף:

--Changing the primary key of the person table

```
ALTER TABLE Person DROP CONSTRAINT SYS_C009716;
```

```
ALTER TABLE Person ADD CONSTRAINT person_key PRIMARY KEY  
(personId);
```

כעת נחליף את המפתחות הראשיים בכל הטבלאות שבהם mainphone היווה מפתח ראשי

--Changing the primary key of all the tables

```
ALTER TABLE Manager DROP CONSTRAINT SYS_C009731;
```

```
ALTER TABLE ActivityOperator DROP CONSTRAINT SYS_C009721;
```

```
ALTER TABLE participant DROP CONSTRAINT SYS_C009753;
```

```
ALTER TABLE Manager ADD CONSTRAINT manager_key PRIMARY KEY (personId);
```

```
ALTER TABLE ActivityOperator ADD CONSTRAINT ao_key PRIMARY KEY (personId);
```

```
ALTER TABLE participant ADD CONSTRAINT participant_key PRIMARY KEY (personId);
```

--Deleting the constraint from tables where the primary key is a pair of values

```
ALTER TABLE Event_Participant  
DROP CONSTRAINT SYS_C009761;
```

```
ALTER TABLE Event_Participant  
ADD CONSTRAINT pk_event_participant PRIMARY KEY (EID, personId);
```

```
ALTER TABLE Event_Operator  
DROP CONSTRAINT SYS_C009748;
```

```
ALTER TABLE Event_Operator  
ADD CONSTRAINT pk_event_operator PRIMARY KEY (EID, personId);
```

```
ALTER TABLE Manager_Working_Days  
DROP CONSTRAINT SYS_C009766;
```

```
ALTER TABLE Manager_Working_Days  
ADD CONSTRAINT pk_manager_working_days PRIMARY KEY (Working_Days, personId);
```

```
ALTER TABLE Person_More_Phone  
DROP CONSTRAINT SYS_C009770;
```

```
ALTER TABLE Person_More_Phone  
ADD CONSTRAINT pk_person_more_phone PRIMARY KEY (More_Phone, personId);
```

כעת נוסיף את אילוצי המפתחות הזרים על העמודה personId בכל מקום  
שמחקנו אותם מהmainphone:

--Adding the foreign key constraint where we deleted itALTER TABLE

Event\_Participant

ADD CONSTRAINT fk\_Event\_Participant\_personId

FOREIGN KEY (personId) REFERENCES Participant(personId);

ALTER TABLE Event

ADD CONSTRAINT fk\_Event\_personId

FOREIGN KEY (personId) REFERENCES Manager(personId);

ALTER TABLE Event\_Operator

ADD CONSTRAINT fk\_Event\_Operator\_personId

FOREIGN KEY (personId) REFERENCES ActivityOperator(personId);

ALTER TABLE Manager\_Working\_Days

ADD CONSTRAINT fk\_mwd\_personId

FOREIGN KEY (personId) REFERENCES Manager(personId);

ALTER TABLE Manager

ADD CONSTRAINT fk\_Manager\_personId

FOREIGN KEY (personId) REFERENCES person(personId);

ALTER TABLE ActivityOperator

ADD CONSTRAINT fk\_ActivityOperator\_personId

FOREIGN KEY (personId) REFERENCES person(personId);

ALTER TABLE participant

ADD CONSTRAINT fk\_participant\_personId

FOREIGN KEY (personId) REFERENCES person(personId);

ALTER TABLE Person\_More\_Phone

ADD CONSTRAINT fk\_pmp\_personId

FOREIGN KEY (personId) REFERENCES person(personId);

הדבר האחרון שנותר הוא למחוק את העמודה mainphone מכל המקומות בהם היוותה מפתח זר (נשאיר אותה רק בטבלה person בתור עמודה רגילה)

--drop the mainphone column

```
ALTER TABLE Event_Participant DROP COLUMN mainphone;
```

```
ALTER TABLE Event DROP COLUMN mainphone;
```

```
ALTER TABLE Event_Operator DROP COLUMN mainphone;
```

```
ALTER TABLE Manager_Working_Days DROP COLUMN mainphone;
```

```
ALTER TABLE Manager DROP COLUMN mainphone;
```

```
ALTER TABLE ActivityOperator DROP COLUMN mainphone;
```

```
ALTER TABLE participant DROP COLUMN mainphone;
```

```
ALTER TABLE Person_More_Phone DROP COLUMN mainphone;
```

כעת לאחר שסיימנו לבצע את השינויים הנדרשים להתאמת המפתח הראשי בטבלאות שקיבלנו, נעבור לשלב ב של האינטגרציה שהוא חיבור הטבלאות שלנו לירושה מ-person.

ראשית נתאים את הטבלה person לטבלאות שלנו בכך שנהפוך שדות לא משמעותיים לnullable.

--put a null option in the tables of the people we don't want to fill in all the fields for them

```
ALTER TABLE Person
```

```
MODIFY Address varchar2(40) NULL;
```

```
ALTER TABLE Person
```

```
MODIFY Mail varchar2(30) NULL;
```

```
ALTER TABLE Person
```

```
MODIFY Age INT NULL;
```

בנוסף, נשנה את מגבלת התווים על שדה הטלפון מ10 ל11 שזו היתה  
הסטנדרטיזציה שהחלטנו עליה

--Increasing the phone number size to the standard size

```
ALTER TABLE Person
```

```
MODIFY mainPhone VARCHAR(11);
```

נכניס כעת את המידע מכל הטבלאות שאנו רוצים שירשו מהטבלה Person לתוך  
הטבלה בפועל:

--Inserting data from all the tables that inherit from person into the  
main table

```
INSERT INTO Person (personId, pname, mainPhone, Address, Mail, Age)
```

```
SELECT
```

```
    personId AS new_personId,
```

```
    LibrarianName AS pname,
```

```
    Lphone AS mainPhone,
```

```
    NULL AS Address, -- assuming other columns can be NULL or have  
default values
```

```
    NULL AS Mail,
```

```
    NULL AS Age
```

```
FROM
```

```
    Librarian;
```

```
INSERT INTO Person (personId, pname, mainPhone, Address, Mail, Age)
```

```
SELECT
```

```
    Supid AS new_personId,
```

```
    supname AS pname,
```

```
    sphone AS mainPhone,
```

```
    NULL AS Address, -- assuming other columns can be NULL or have  
default values
```

```
    NULL AS Mail,
```

```
    NULL AS Age
```

```
FROM
```

```
    supplier;
```

```

INSERT INTO Person (personId, pname, mainPhone, Address, Mail, Age)
SELECT
    readerNumber AS new_personId,
    readername AS pname,
    phone AS mainPhone,
    NULL AS Address, -- assuming other columns can be NULL or have
default values
    NULL AS Mail,
    NULL AS Age
FROM
    reader;

```

עתה נטפל טבלה טבלה בשינויי המפתח ובירושה. נתחיל מהטבלה librarian בה המפתח כרגע הוא username מסוג varchar. נרצה להוסיף לכל ספרן מזהה מהטבלה person ולהפוך את המזהה למפתח זר ומפתח ראשי בטבלה. נתחיל מלמחוק את האילוצים על המפתח הראשי הנוכחי מהטבלאות בהם הוא קיים - טבלת ההזמנות.

--A change for a librarian because until now his key was varchar and now it will be number  
ALTER TABLE ordering DROP CONSTRAINT SYS\_C009794; --for key  
in ordering

עכשיו נוסיף את העמודה personId בטבלה ונמלא אותה בהתאם

```

ALTER TABLE librarian ADD personId NUMBER;

```

```

MERGE INTO librarian l
USING person pe
ON (l.librarianname = pe.pname)
WHEN MATCHED THEN
UPDATE SET l.personID = pe.personID;

```

נשנה את המפתח הראשי בטבלה לpersonid

```
ALTER TABLE Librarian  
ADD CONSTRAINT fk_librarian_person  
FOREIGN KEY (personId) REFERENCES Person(personId)
```

```
ALTER TABLE Librarian DROP CONSTRAINT SYS_C009788;--drop  
primarykey constraint from librarian
```

```
ALTER TABLE Librarian ADD CONSTRAINT Librarian_key PRIMARY KEY  
(personId);
```

נתאים את טבלת ההזמנות למפתח הראשי החדש ונוסיף את אילוץ המפתח הזר

--Filling the personid column values in the ordering table according to  
the personid column values in the librarian table

```
ALTER TABLE ordering ADD personId NUMBER;
```

```
MERGE INTO ordering o  
USING librarian li  
ON (o.username = li.username)  
WHEN MATCHED THEN  
UPDATE SET o.personID = li.personID;
```

```
ALTER TABLE ordering  
ADD CONSTRAINT fk_librarian_ordering  
FOREIGN KEY (personId) REFERENCES librarian(personId)  
ON UPDATE CASCADE;
```

כעת נטפל בטבלאות קורא וספק. נוסיף למפתח הראשי הקיים שלהם (מספר  
קורא ומזהה ספק) אילוץ מפתח זר מpersonid, מכיוון שהם זהים באופיים לשדה  
זה, אין צורך להחליפם.



```
ALTER TABLE supplier  
ADD CONSTRAINT fk_supplier_person  
FOREIGN KEY (supid) REFERENCES Person(personId)
```

```
ALTER TABLE reader  
ADD CONSTRAINT fk_reader_person  
FOREIGN KEY (readernumber) REFERENCES Person(personId)
```

לסיום, נמחק את כל העמודות הכפולות שנוצרו עקב השכפול של הנתונים לטבלת  
person

--Deleting all duplicate columns

```
ALTER TABLE reader  
DROP COLUMN readername;
```

```
ALTER TABLE reader  
DROP COLUMN phone;
```

```
ALTER TABLE librarian  
DROP COLUMN librarianname;
```

```
ALTER TABLE librarian  
DROP COLUMN Iphone;
```

```
ALTER TABLE supplier  
DROP COLUMN supname;
```

```
ALTER TABLE supplier  
DROP COLUMN sphone;
```

```
ALTER TABLE ordering  
DROP COLUMN lusername;
```

כעת האינטרגציה בוצעה, והקשרים הקיימים הם כפי שהחלטנו והבענו בERD  
המשותף.

# VIEWS

## ordering view

--View to return details for each order:

--name of the orderer, name of the supplier, number of books ordered and price.

CREATE OR REPLACE VIEW LibrarianOrderDetails AS

SELECT

P.Pname AS LibrarianName,

O.orderId AS OrderNumber,

O.orderDate AS OrderDate,

SP.Pname AS SupplierName,

COUNT(BO.BookID) AS NumberOfBooksOrdered,

O.totalPrice AS PriceCharged

FROM

Librarian L

JOIN

Person P ON L.personId = P.personId

JOIN

Ordering O ON L.personId = O.personId

JOIN

Supplier S ON O.supid = S.supid

JOIN

Person SP ON S.supid = SP.personId

JOIN

BookOrdering BO ON O.orderId = BO.orderId

GROUP BY

P.Pname, O.orderId, O.orderDate, SP.Pname, O.totalPrice;

זהו מבט על הזמנות:

מבט זה מאחד את כל הנתונים הרלוונטיים על ההזמנות הקיימות במערכת:  
הוא מספק עבור כל הזמנה את שם הספרן המזמין, שם הספק, מספר ההזמנה ותאריכה, מספר הספרים שהוזמנו בה ומחירה.

SQL

Output Statistics

```
select * from LibrarianOrderDetails
```

	LIBRARIANNAME	ORDERNUMBER	ORDERDATE	SUPPLIERNAME	NUMBEROFBOOKSORDERED	PRICECHARGED
1	d	28	20/05/2020	f	5	349.00
2	h	31	15/03/2019	f	2	105.30
3	a	10	21/07/2021	g	5	306.10
4	c	3	24/01/2021	h	5	304.67
5	b	15	25/07/2021	i	5	399.25
6	a	5	19/02/2018	j	5	286.75
7	c	20	10/12/2021	j	5	420.13
8	c	22	29/09/2019	k	5	390.47
9	d	25	01/02/2022	k	5	312.02

--Returns the 10 most expensive orders placed

SELECT \*

FROM LibrarianOrderDetails

ORDER BY PriceCharged DESC

FETCH FIRST 10 ROWS ONLY;

שאלתה זו מחזירה את 10 ההזמנות היקרות ביותר הקיימות במערכת

```
--Returns the 10 most expensive orders placed
```

```
SELECT *
```

```
FROM LibrarianOrderDetails
```

```
ORDER BY PriceCharged DESC
```

```
FETCH FIRST 10 ROWS ONLY;
```

	LIBRARIANNAME	ORDERNUMBER	ORDERDATE	SUPPLIERNAME	NUMBEROFBOOKSORDERED	PRICECHARGED
1	c	20	10/12/2021	j	5	420.13
2	b	15	25/07/2021	i	5	399.25
3	c	22	29/09/2019	k	5	390.47
4	d	28	20/05/2020	f	5	349.00
5	d	25	01/02/2022	k	5	312.02
6	a	10	21/07/2021	g	5	306.10
7	c	3	24/01/2021	h	5	304.67
8	a	5	19/02/2018	j	5	286.75

--Returns how much money the library owes to each supplier

SELECT

SupplierName,

SUM(PriceCharged) AS TotalAmountPaid

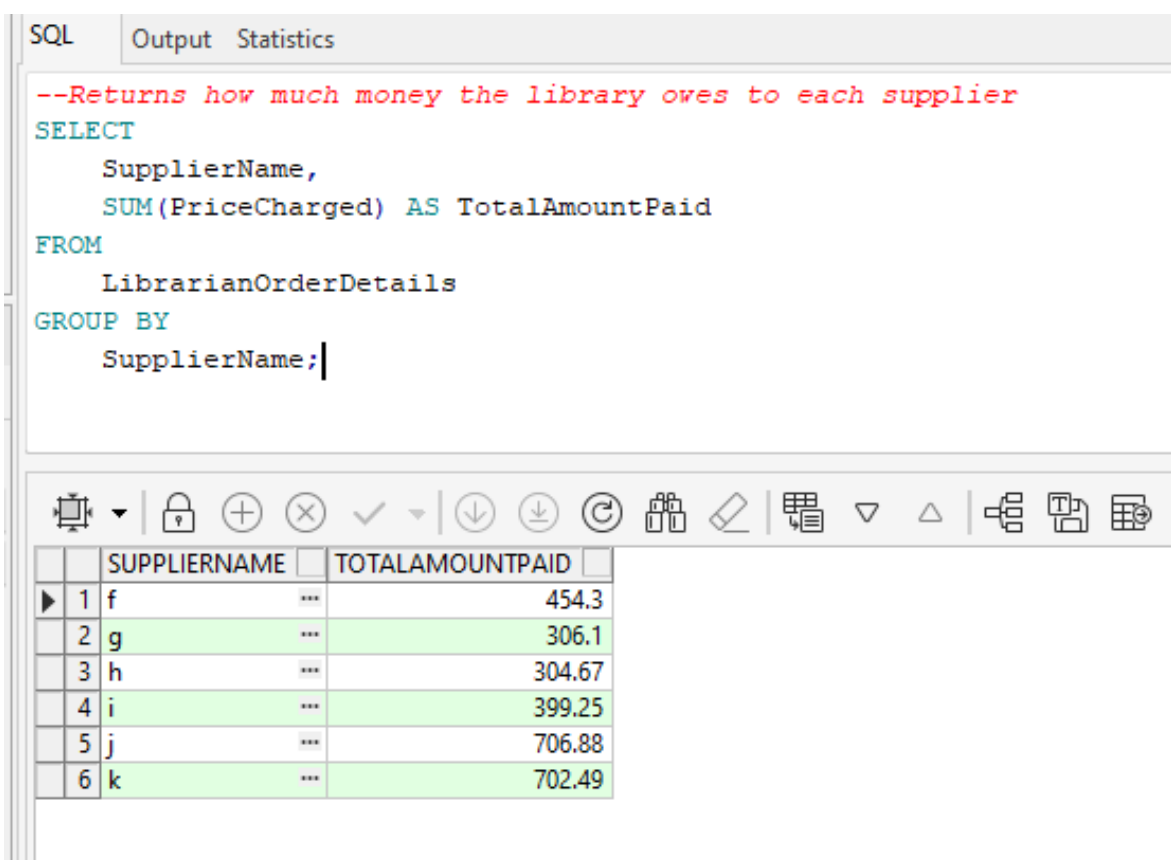
FROM

LibrarianOrderDetails

GROUP BY

SupplierName;

שאלתה זו מחזירה לפי נתוני ההזמנות, כמה כסף הספרייה צריכה לשלם לכל ספק.



SQL Output Statistics

```
--Returns how much money the library owes to each supplier
SELECT
    SupplierName,
    SUM(PriceCharged) AS TotalAmountPaid
FROM
    LibrarianOrderDetails
GROUP BY
    SupplierName;
```

	SUPPLIERNAME	TOTALAMOUNTPAID
1	f	454.3
2	g	306.1
3	h	304.67
4	i	399.25
5	j	706.88
6	k	702.49

# events view

--dislay details of events that took place in the months of July-August

CREATE OR REPLACE VIEW RECENTEVENTDETAILS AS

SELECT

e.EID,  
e.EDate,  
e.ETime,  
e.Audience,  
e.ECategory,  
e.ActivityD,  
e.Price,  
e.Duration,  
h.HName AS HallName,  
h.Place AS HallPlace,  
h.Capacity AS HallCapacity,  
p.Pname AS ManagerName,  
m.MRole AS ManagerRole,  
ao.Expertise AS OperatorExpertise,  
ao.Gender AS OperatorGender,  
part.UserName AS ParticipantUserName,  
part.PType AS ParticipantType,  
ep.Review AS ParticipantReview,  
ep.SeatNumber AS ParticipantSeatNumber,  
ep.SignUpDate AS ParticipantSignUpDate

FROM

Event e

JOIN

Hall h ON e.HID = h.HID

JOIN

Manager m ON e.personid = m.personid

JOIN

Person p ON m.personid = p.personid

JOIN

Event\_Operator eo ON e.EID = eo.EID

JOIN

ActivityOperator ao ON eo.personid = ao.personid

JOIN

Event\_Participant ep ON e.EID = ep.EID

JOIN

Participant part ON ep.personid = part.personid

WHERE

EXTRACT(MONTH FROM e.EDate) IN (7, 8)

ORDER BY

e.EDate DESC;

הview events מחזיר פרטים חשובים על הארועים שהתרחשו בחודשי הקיץ  
(יולי-אוגוסט): פרטים על הארוע עצמו-תאריך, מיקום,קטגוריה ועוד. ובנסוף פרטים  
על מנהל הארוע, המפעיל והמשתתפים.

שאלתא 1: הצגת המפעילים שהמומחיות שלהם היא "Yoga guide" - של  
הארועים שעלו פחות מ500 ש"ח, בסדר עולה מהארוע הזול לארוע היקר ביותר  
(אם נרצה להזמין ארוע בתקציב מוגבל)

--display the Activity operators of the cheapest events

SELECT

p.Pname AS OperatorName,  
p.Mail AS OperatorEmail,  
p.MainPhone AS OperatorPhone,  
red.ActivityD AS EventName,  
red.EDate AS EventDate,  
red.Price AS EventPrice

FROM

RECENTEVENTDETAILS red

JOIN

Event\_Operator eo ON red.EID = eo.EID

JOIN

Activityoperator ao ON eo.personid = ao.personid

JOIN

Person p ON ao.personid = p.personid

WHERE

red.Price < 500 AND red.OperatorExpertise = 'Yoga guide'

ORDER BY

red.Price ASC;

--display the Activity operators of the cheapest events

```
SELECT
  p.Pname AS OperatorName,
  p.Mail AS OperatorEmail,
  p.MainPhone AS OperatorPhone,
  red.ActivityD AS EventName,
  red.EDate AS EventDate,
  red.Price AS EventPrice
```

FROM  
RECENTEVENTDETAILS red

	OPERATORNAME	OPERATOREMAIL	OPERATORPHONE	EVENTNAME	EVENTDATE	EVENTPRICE
1	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
2	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
3	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
4	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
5	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
6	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
7	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
8	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
9	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
10	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
11	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
12	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
13	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
14	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51
15	Dustin Ruffalo	dustin.ruffalo@mitsubishimotor	0574211067	ADS External USB 2.0 Drive Kit	02 2014 יוני	15.51

שאלתא 2: הצגת המשתתפים בארועים השונים, מסודר בסדר יורד מהמשתתף שהשתתף בהכי הרבה ארועים ועד למשתתף שהשתתף בהכי מעט ארועים (על מנת לדעת מי המשתתפים הפעילים ביותר)

--display the participants in the events

```
SELECT
```

```
  part.UserName AS ParticipantUserName,
  p.MainPhone AS ParticipantPhone,
  p.Mail AS ParticipantEmail,
  COUNT(red.EID) AS NumberOfEvents
```

```
FROM
```

```
  RECENTEVENTDETAILS red
```

```
JOIN
```

```
  Event_Participant ep ON red.EID = ep.EID
```

```
JOIN
```

```
  Participant part ON ep.personid = part.personid
```

```
JOIN
```

```
  Person p ON part.personid = p.personid
```

```
GROUP BY
```

```
  part.UserName,
  p.MainPhone,
  p.Mail
```

```
ORDER BY
```

```
  NumberOfEvents DESC;
```

--dispay the participants in the events

SELECT

part.UserName AS ParticipantUserName,  
p.MainPhone AS ParticipantPhone,  
p.Mail AS ParticipantEmail,  
COUNT(red.EID) AS NumberOfEvents

FROM

RECENTEVENTDETAILS red

JOIN

Event\_Participant ep ON red.EID = ep.EID

JOIN

	PARTICIPANTUSERNAME	PARTICIPANTPHONE	PARTICIPANTEMAIL	NUMBEROFEVENTS
1	Davey10	0573585596	kitty.c@aco.com	950
2	France88	0585938946	amy@softworld.mx	734
3	Lucy93	0530325327	katie.nugent@mai.pt	696
4	Isabella53	0547705117	debbie.emmett@nobrainerblindsc	695
5	CeCe24	0500454771	bernie.w@linersdirect.mx	678
6	Ralph30	0598759179	tommy.sedaka@dis.com	634
7	Thin14	0571588597	linda.g@catamount.com	626
8	Isaiah21	0589008587	amandap@pra.uk	608
9	Gilbert38	0504353524	nancy.macneil@randomwalk.com	591
10	Irene20	0515342330	randy.deltoro@marlabs.com	581
11	Rawlins38	0505798712	vin.m@microtek.dk	569
12	Noah77	0509619309	lonnie.connick@lloydgroup.de	568
13	Sander09	0544524116	charlton.tate@ppr.li	534
14	Ned25	0522313715	emerson.stampley@hudsonriverba	531
15	Kasey36	0549350445	a.herrmann@cendant.za	523
16	Laura89	0584763434	miles.hawn@ufs.br	514