

דוח פרויקט

מיני פרויקט בבסיסי נתונים

המערכת: מערכת לניהול ספריה
היתידה הנבחרת: ניהול אירועים בספריה



שירה קהלני 325283026
הלל אישון 212455570

תוכן עניינים

(שלב א')

- עמוד 3 - מבוא
- עמוד 4 - תרשים ERD
- עמוד 5 - תרשים DSD
- עמוד 6 - פקודות createTable
- עמוד 8 - פקודת desc
- עמוד 9 - יצירת נתונים ב data generator
- עמוד 10 - יצירת נתונים בקוד פייתון
- עמוד 12 - יצירת נתונים באתר Mockaroo
- עמוד 13 – גיבוי ושחזור

(שלב ב' - שאלות)

- עמוד 14 – שאלות select
- עמוד 18 – שאלות delete
- עמוד 20 – שאלות update
- עמוד 22 – שאלות עם פרמטרים
- עמוד 26 - אילוצים

(שלב ג' - תכניות)

- עמוד 29 – תכנית א'
- עמוד 35 – תכנית ב'

(שלב ד' - אינטגרציה ומבטים)

- עמוד 36 – תהליך האינטגרציה
- עמוד 49 - מבטים

- שלב א' -

מבוא

המערכת הכללית היא מערכת לניהול ספריה. היחידה בה אנו מתמקדות היא ניהול האירועים בספריה.

מטעם הספרייה מתארגנים אירועים שונים ופעילויות שונות, בתאריכים שונים, באולמות שונים הקיימים בספריה, ולקהלי יעד שונים.

המערכת שלנו שומרת מידע רלוונטי אודות האירועים (בעבר או בעתיד), האולמות, מנהלי האירועים, המפעילים והמשתתפים.

מטרת המערכת היא לאפשר ניהול וסנכרון נוח ומסודר בין הגורמים השונים, וכן לאפשר מעקב והסקת מסקנות על סמך נתוני עבר, על מנת לייעל את מדור הפעילויות של הספרייה.

נרצה לתשאל את המערכת שאלות כגון:

כל המידע אודות פעילות מסוימת (פרטי המיקום, המנהל, המשתתפים, מפעיל וכו')

באיזה אולם שביעות הרצון מההפעלות היא הטובה ביותר – על פי ציון המשוב שסיפקו המשתתפים.

מהו האירוע שגרף את הרווחים המקסימליים (נחשב לפי כמות המשתתפים, המחיר לכרטיס, ומחיר המפעיל)

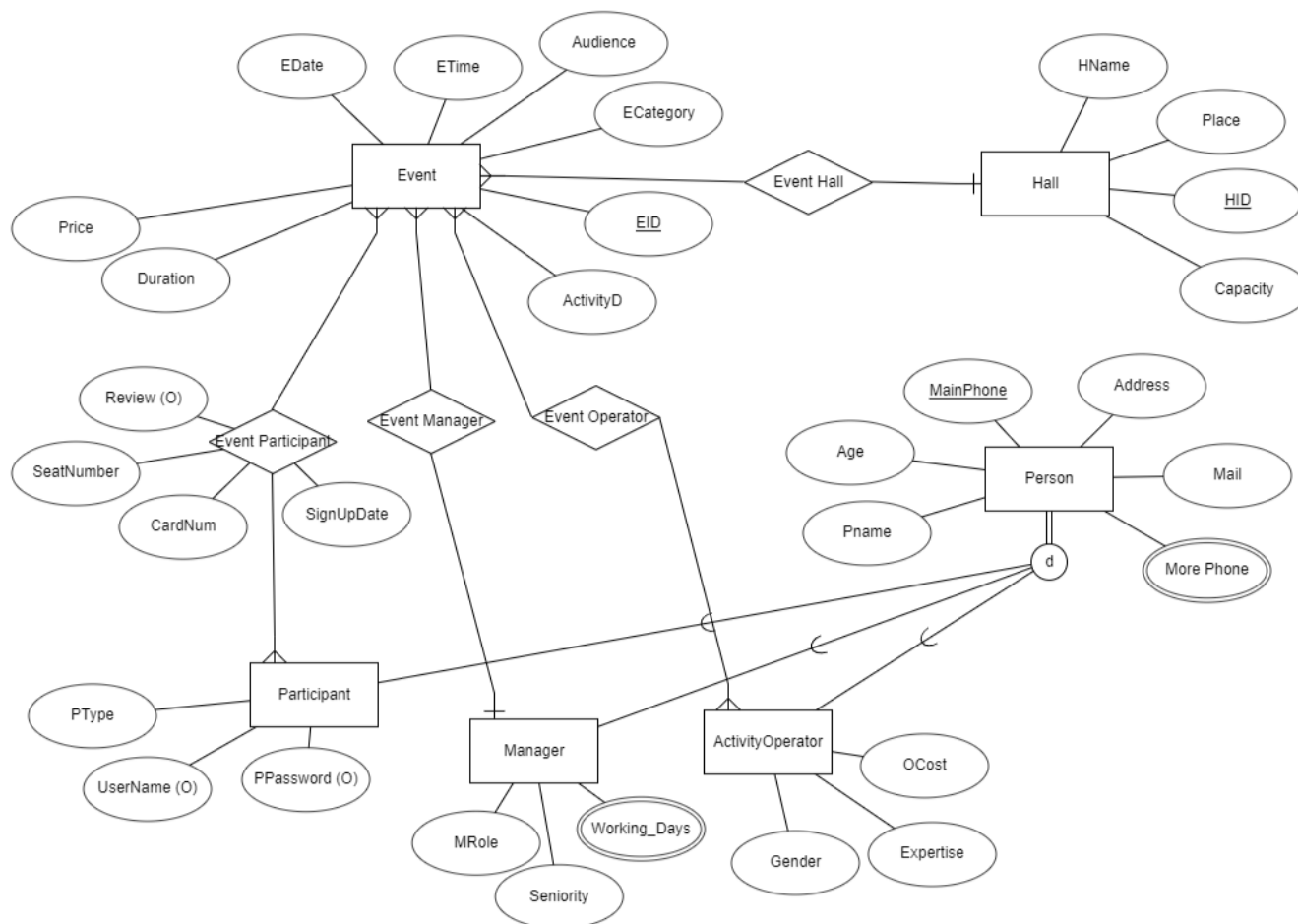
איזה סוג פעילות הוא המועדף בקרב ילדים מתחת לגיל 15 (נראה עבור איזה פעילות כמות הנרשמים מטווח גילאים זה היא הגבוהה ביותר)

מהו הגיל הממוצע של המשתתפים בהרצאות

חישוב הנחה עבור כל הנרשמים עד שבוע לפני יום האירוע

וכן הלאה...

תרשים ERD



נתייחס לכמה החלטות עיצוב שקיבלנו:

קשר "רבים לרבים" בין ישות מפעיל לישות אירוע: מאחר שיתכן שיוזמן יותר ממפעיל יחיד לאירוע מסוים, הקשר הוא כזה שמאפשר זאת.

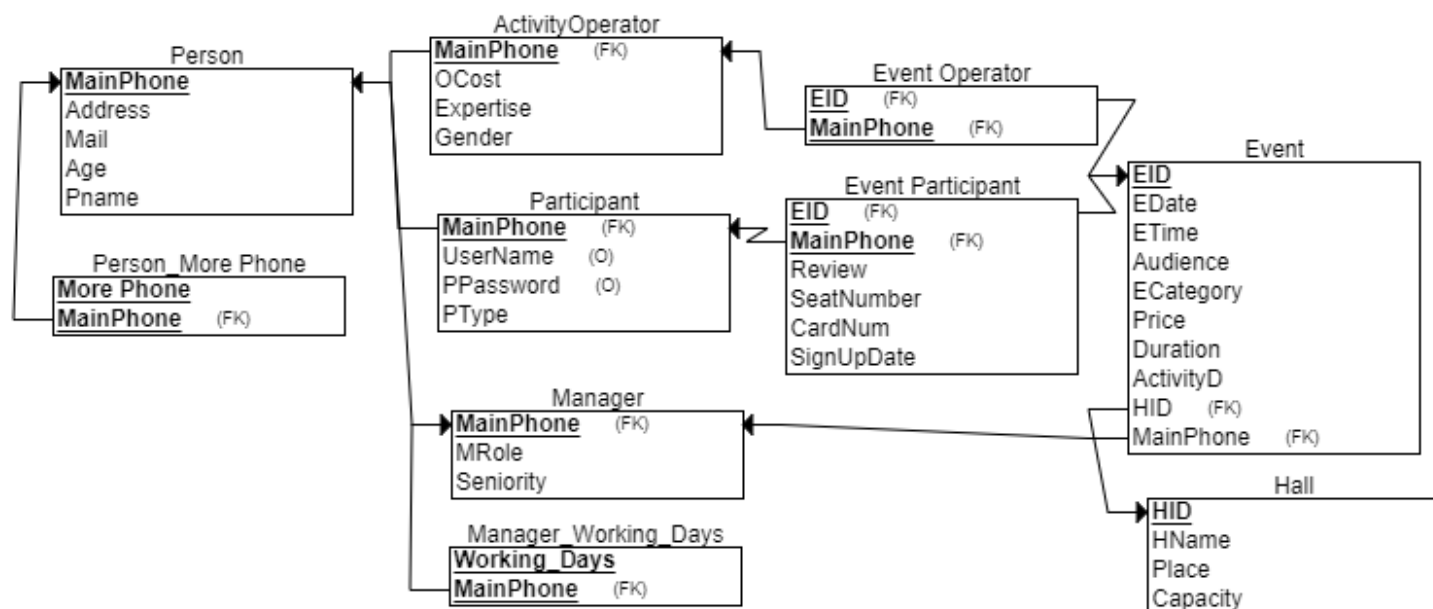
לעומת זאת: קשר "יחיד לרבים" בין ישות נהל לישות אירוע: בחרנו לאפשר שיבוץ של מנהל אחד בלבד לכל אירוע שיהיה האחראי הראשי, כדי לשמור על הגיון ועל סדר.

משתתף יכול להיות מסוג "מנוי" או "לא מנוי". עבור משתתף מנוי בלבד נשמרים שם המשתמש וסיסמא, ולכן תכונות אלו אופציונאליות בלבד.

עבור ישות מנהל ישנה תכונה ששומרת את הימים בשבוע בהם הוא עובד, משום שדבר זה תורם להסקת מסקנות שקשורות לניהול אירועים, ולשיבוץ תקין של מנהל ואירוע זה לזה.

מאחר ויש לנו שלוש ישויות (מפעיל, מנהל ומשתתף) שמייצגות אנשים, ארגנו את הקשרים בעזרת ירושה, כך שיש ישות "person" כללית, שלה תכונות הנוגעות לכל האנשים. וממנה יורשת כל ישות ספציפית יותר, ומכילה תכונות נוספות שרלוונטיות עבורה בלבד.

תרשים DSD



פיקודת CREATE TABLE

```

CREATE TABLE Hall
(
    HName varchar2(15) NOT NULL,
    Place varchar2(40) NOT NULL,
    HID INT NOT NULL,
    Capacity INT NOT NULL,
    PRIMARY KEY (HID)
);

CREATE TABLE Person
(
    Address varchar2(40) NOT NULL,
    Mail varchar2(30) NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    Age INT NOT NULL,
    Pname varchar2(30) NOT NULL,
    PRIMARY KEY (MainPhone)
);

CREATE TABLE Manager
(
    MRole varchar2(20) NOT NULL,
    Seniority INT NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Person(MainPhone)
);

CREATE TABLE ActivityOperator
(
    OCost number(8,2) NOT NULL,
    Expertise varchar2(40) NOT NULL,
    Gender varchar2(15) NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Person(MainPhone)
);

CREATE TABLE Participant
(
    UserName varchar2(40),
    PPassword INT,
    PType varchar2(30) NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Person(MainPhone)
);

CREATE TABLE Person_More_Phone
(
    More_Phone varchar2(10) NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (More_Phone, MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Person(MainPhone)
);

```

```

CREATE TABLE Manager_Working_Days
(
    Working_Days varchar2(10) NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (Working_Days, MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Manager(MainPhone)
);

CREATE TABLE Event
(
    EDate date NOT NULL,
    ETime number(4,0) NOT NULL,
    Audience varchar2(30) NOT NULL,
    ECategory varchar2(40) NOT NULL,
    ActivityD varchar2(300) NOT NULL,
    Price NUMBER(6,2) NOT NULL,
    Duration INT NOT NULL,
    EID INT NOT NULL,
    HID INT NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (EID),
    FOREIGN KEY (HID) REFERENCES Hall(HID),
    FOREIGN KEY (MainPhone) REFERENCES Manager(MainPhone)
);

CREATE TABLE Event_Operator
(
    EID INT NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (EID, MainPhone),
    FOREIGN KEY (EID) REFERENCES Event(EID),
    FOREIGN KEY (MainPhone) REFERENCES ActivityOperator(MainPhone)
);

CREATE TABLE Event_Participant
(
    Review INT NOT NULL,
    SeatNumber INT NOT NULL,
    CardNum INT NOT NULL,
    SignUpDate date NOT NULL,
    EID INT NOT NULL,
    MainPhone varchar2(10) NOT NULL,
    PRIMARY KEY (EID, MainPhone),
    FOREIGN KEY (EID) REFERENCES Event(EID),
    FOREIGN KEY (MainPhone) REFERENCES Participant(MainPhone)
);

```

Desc

Columns of HALL

Name	Type	Nullable	Default	Comments
▶ HNAME	VARCHAR2(15)			
2 PLACE	VARCHAR2(40)			
3 HID	INTEGER			
4 CAPACITY	INTEGER			

Columns of PERSON

Name	Type	Nullable	Default	Comments
▶ ADDRESS	VARCHAR2(40)			
2 MAIL	VARCHAR2(30)			
3 MAINPHONE	VARCHAR2(10)			
4 AGE	INTEGER			
5 PNAME	VARCHAR2(30)			

Columns of ACTIVITYOPERATOR

Name	Type	Nullable	Default	Comments
▶ OCOST	NUMBER(8,2)			
2 EXPERTISE	VARCHAR2(40)			
3 GENDER	VARCHAR2(15)			
4 MAINPHONE	VARCHAR2(10)			

Columns of MANAGER

Name	Type	Nullable	Default	Comments
▶ MROLE	VARCHAR2(20)			
2 SENIORITY	INTEGER			
3 MAINPHONE	VARCHAR2(10)			

Columns of PARTICIPANT

Name	Type	Nullable	Default	Comments
▶ USERNAME	VARCHAR2(40)	Y		
2 PPASSWORD	INTEGER	Y		
3 PTYPE	VARCHAR2(30)			
4 MAINPHONE	VARCHAR2(10)			

Columns of PERSON_MORE_PHONE

Name	Type	Nullable	Default	Comments
▶ MORE_PHONE	VARCHAR2(10)			
2 MAINPHONE	VARCHAR2(10)			

Columns of MANAGER_WORKING_DAYS

Name	Type	Nullable	Default	Comments
▶ WORKING_DAYS	VARCHAR2(10)			
2 MAINPHONE	VARCHAR2(10)			

Columns of EVENT

Name	Type	Nullable	Default	Comments
▶ EDATE	DATE			
2 ETIME	NUMBER(4)			
3 AUDIENCE	VARCHAR2(30)			
4 ECATEGORY	VARCHAR2(40)			
5 ACTIVITYD	VARCHAR2(300)			
6 PRICE	NUMBER(6,2)			
7 DURATION	INTEGER			
8 EID	INTEGER			
9 HID	INTEGER			
10 MAINPHONE	VARCHAR2(10)			

Columns of EVENT_OPERATOR

Name	Type	Nullable	Default	Comments
▶ EID	INTEGER			
2 MAINPHONE	VARCHAR2(10)			

Columns of EVENT_PARTICIPANT

Name	Type	Nullable	Default	Comments
▶ REVIEW	INTEGER			
2 SEATNUMBER	INTEGER			
3 CARDNUM	INTEGER			
4 SIGNUPDATE	DATE			
5 EID	INTEGER			
6 MAINPHONE	VARCHAR2(10)			

הכנסת נתונים ע"י data generator

לדוגמא:

EVENT				
Owner	Table	Number of records		
SYSTEM	EVENT	420		
Name	Type	Size	Data	
EDATE	DATE		Random(1/1/2000, 1/1/2016)	
ETIME	NUMBER	4	list(1000, 1030, 1100, 1200, 1230, 1300, 1330, 1400, 1430, 1500, 1530, 1600, 1630, 1700, 1730, 1800,	
AUDIENCE	VARCHAR2	30	list('Children', 'Adults', 'Elderly', 'Youth', 'Children and youth')	
ECATEGORY	VARCHAR2	40	list('Lecture', 'workshop', 'performance', 'story', 'educational activity', 'play', 'conversation with an au	
ACTIVITYD	VARCHAR2	300	Components.Description	
PRICE	NUMBER	6,2	Random(10.00, 300.00)	
DURATION	NUMBER		Random(30, 360)	
EID	NUMBER		Sequence(1, [Inc])	
HID	NUMBER		List(select HID from HALL)	
MAINPHONE	VARCHAR2	10	List(select MAINPHONE from MANAGER)	

HALL				
Owner	Table	Number of records		
SYSTEM	HALL	450		
Name	Type	Size	Data	
HNAME	VARCHAR2	15	List('Iris', 'Tavor', 'Galilee', 'Hermon', 'Carmel', 'Negev', 'Sharon', 'Golan', 'Eilat', 'Sinai')	
PLACE	VARCHAR2	40	List('floor: 0 room: 0001', 'floor: 1 room: 1023', 'floor: 2 room: 2034', 'floor: 3 room: 3045', 'floor: 4 room: 4056', 'floor: 5	
HID	NUMBER		Sequence(1, [Inc], [WithinParent])	
CAPACITY	NUMBER		List(30,50,80,100,150,200,250,300,500)	

EVENT										
EDATE	ETIME	AUDIENCE	ECATEGORY	ACTIVITYD	PRICE	DURA	EID	HID	MAINPHONE	
04-05-2012	1630	Adults	exhibition	ADS USB2000 USB Turbo 2.0 pci (53.77	224	1	8	5550004	
17-04-2001	1330	Adults	performance	AMD XP 2000+ Box/with AMD Fa	218.51	52	2	5	5550008	
24-03-2010	1630	Elderly	story	AMD XP 2600+ Retail box. (with	201.20	134	3	2	5550004	
20-03-2007	1630	Youth	sports	GA-SINXP1394(GA-8SQ800 Ultra	152.14	107	4	2	5550008	
25-08-2013	1530	Children ar	story	AMD XP 2200+ & Gigabyte 7N40	147.54	65	5	6	5550001	
17-06-2001	1630	Elderly	film screening	Black Complete Computer System	190.68	110	6	6	5550001	
30-11-2000	1030	Adults	performance	Cable/DSL Broadband Router (w/	209.14	161	7	1	5550004	
16-01-2002	1200	Elderly	film screening	ATI Radeon 7000 32MB DDR PCI	265.19	343	8	4	5550001	
31-12-2012	1230	Elderly	sports	8 inch Neon blue lamp	36.09	201	9	3	5550003	
22-02-2014	1000	Elderly	exhibition	Combo AMD XP 2600+ CPU; Giga	82.78	317	10	6	5550008	
01-02-2014	1900	Children ar	exhibition	Maxtor External 80Gb Hard drive	287.55	96	11	7	5550006	
16-12-2000	1930	Children ar	exhibition	Intel Pentium4 2.0ghz & Gigabyte	66.50	345	12	6	5550001	
17-05-2000	1630	Youth	volunteering	14inch LCD MG-015A Xgen2 Mon	236.00	158	13	7	5550008	
28-07-2000	1300	Adults	exhibition	USB to Parallel Printer Cable.	269.12	360	14	4	5550008	
15-05-2007	1300	Children ar	exhibition	Biostar M7TDR P4 M/B Skt 423 (w	184.18	162	15	7	5550008	
16-10-2011	1500	Elderly	training	Intel Celeron 1.3Ghz & Biostar M6	175.97	303	16	2	5550006	
25-12-2011	1430	Elderly	story	TV Tuner card PCI (KWorld M878	265.58	207	17	6	5550003	
05-07-2004	1900	Elderly	educational activity	Intel Celeron 1.3GHz 256k S370 F	244.54	282	18	7	5550008	
03-08-2014	1030	Youth	exhibition	AMD XP 2000+ (OEM)	54.86	282	19	7	5550001	
24-04-2005	1200	Children ar	Lecture	AMD XP 1800+ & Biostar M7VKQ	80.72	134	20	4	5550003	
26-07-2006	1200	Adults	film screening	Intel Pentium4 3.0ghz & Asus P4C	127.18	207	21	2	5550003	
14-03-2011	1230	Elderly	sports	Sony 16X DVD-CDRom (OEM)	162.86	58	22	3	5550001	
30-06-2006	1800	Children	Lecture	AMD XP 1700+ & Gigabyte GA-7	199.23	212	23	5	5550001	
22-04-2015	1630	Elderly	performance	ADS Technologies Firewire Extern	151.69	295	24	7	5550001	
26-05-2007	2000	Adults	film screening	AMD XP 2100+ CPU (oem)(AXDA	236.29	338	25	6	5550003	
23-06-2008	1500	Children	exhibition	Cyber Power 725VA Battery Back	226.61	139	26	8	5550003	

HALL				
HNAME	PLACE	HID	CAPACITY	
Iris	floor: 2 room: 2034	1	200	
Galilee	floor: 0 room: 0001	2	250	
Eilat	floor: 2 room: 2034	3	200	
Golan	floor: 4 room: 4056	4	300	
Sharon	floor: 0 room: 0001	5	250	
Sinai	floor: 2 room: 2034	6	30	
Hermon	floor: 3 room: 3045	7	80	
Iris	floor: 2 room: 2034	8	80	
Negev	floor: 6 room: 6078	9	150	
Negev	floor: 2 room: 2034	10	500	
Galilee	floor: 1 room: 1023	11	50	
Sinai	floor: 3 room: 3045	12	80	
Hermon	floor: 5 room: 5067	13	200	
Hermon	floor: 1 room: 1023	14	30	
Sharon	floor: 4 room: 4056	15	50	
Negev	floor: 0 room: 0001	16	30	
Eilat	floor: 5 room: 5067	17	50	
Carmel	floor: 3 room: 3045	18	50	
Sinai	floor: 1 room: 1023	19	250	
Eilat	floor: 0 room: 0001	20	500	
Iris	floor: 1 room: 1023	21	300	
Eilat	floor: 3 room: 3045	22	150	
Galilee	floor: 1 room: 1023	23	80	
Eilat	floor: 2 room: 2034	24	500	

הכנסת נתונים עם קוד python

חלק מן הקוד:

```
import random
from datetime import datetime, timedelta

def random_date(start, end):
    return start + timedelta(days=random.randint(0, (end - start).days))

def generate_phone_number():
    return "555" + "".join([str(random.randint(0, 9)) for _ in range(4)])

def main():
    # File to write SQL inserts
    with open("InsertTablesByPython.sql", "w") as file:
        # Parameters for data generation
        num_entries = 15

        # Generating and writing data for Hall table
        for i in range(1, num_entries + 1):
            hname = f"Hall {i}"
            place = f"Location {i}"
            hid = i
            capacity = random.randint(20, 300)
            file.write(f"INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('{hname}', '{place}', {hi

        # Generating and writing data for Person table
        persons = []
        for i in range(1, num_entries + 1):
            address = f"{i * 123} Elm St"
            mail = f"person{i}@example.com"
            main_phone = generate_phone_number()
            age = random.randint(20, 60)
            pname = f"Person {i}"
            persons.append((address, mail, main_phone, age, pname))
            file.write(f"INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('{address}'

        # Generating and writing data for Manager table
        for i in range(1, num_entries + 1):
            mrole = random.choice(['Event Coordinator', 'Senior Manager', 'Assistant Manager', 'Operat

        # Generating and writing data for Event table
        start_date = datetime.strptime('2024-06-01', '%Y-%m-%d')
        end_date = datetime.strptime('2024-06-30', '%Y-%m-%d')
        for i in range(1, num_entries + 1):
            edate = random_date(start_date, end_date).strftime('%Y-%m-%d')
            etime = random.randint(1000, 1800)
            audience = random.choice(['Children', 'Adults', 'Family', 'Teens', 'Seniors'])
            category = random.choice(['Education', 'Health', 'Entertainment', 'Literature', 'History', 'Mus
            activityd = f"Activity Description {i}"
            price = round(random.uniform(5.00, 20.00), 2)
            duration = random.randint(45, 180)
            eid = i
            hid = random.randint(1, num_entries)
            main_phone = random.choice(persons)[2]
            file.write(f"INSERT INTO Event (EDate, ETime, Audience, ECategory, ActivityD, Price, Duration, EI

        # Generating and writing data for Event_Operator table
        for i in range(1, num_entries + 1):
            eid = i
            main_phone = random.choice(persons)[2]
            file.write(f"INSERT INTO Event_Operator (EID, MainPhone) VALUES ({eid}, '{main_phone}');\n")

        # Generating and writing data for Event_Participant table
        for i in range(1, num_entries + 1):
            review = random.randint(1, 5)
            seat_number = random.randint(1, 200)
            card_num = random.randint(10000000000, 99999999999)
            signup_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
            eid = i
            main_phone = random.choice(persons)[2]
            file.write(f"INSERT INTO Event_Participant (Review, SeatNumber, CardNum, SignUpDate, EID, MainPho
```

חלק מן הפקודות שנוצרו:

```
C: > Users > halel > OneDrive - gjct.ac.il > מסמכים > לימודים > מיניפ בסיסי נתונים > InsertTablesByPython.sql

1  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 1', 'Location 1', 1, 243);
2  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 2', 'Location 2', 2, 25);
3  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 3', 'Location 3', 3, 211);
4  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 4', 'Location 4', 4, 265);
5  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 5', 'Location 5', 5, 246);
6  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 6', 'Location 6', 6, 58);
7  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 7', 'Location 7', 7, 173);
8  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 8', 'Location 8', 8, 104);
9  INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 9', 'Location 9', 9, 295);
10 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 10', 'Location 10', 10, 128);
11 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 11', 'Location 11', 11, 98);
12 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 12', 'Location 12', 12, 88);
13 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 13', 'Location 13', 13, 135);
14 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 14', 'Location 14', 14, 286);
15 INSERT INTO Hall (HName, Place, HID, Capacity) VALUES ('Hall 15', 'Location 15', 15, 129);
16 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('123 Elm St', 'person1@example.com', '555
17 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('246 Elm St', 'person2@example.com', '555
18 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('369 Elm St', 'person3@example.com', '555
19 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('492 Elm St', 'person4@example.com', '555
20 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('615 Elm St', 'person5@example.com', '555
21 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('738 Elm St', 'person6@example.com', '555
22 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('861 Elm St', 'person7@example.com', '555
23 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('984 Elm St', 'person8@example.com', '555
24 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1107 Elm St', 'person9@example.com', '55
25 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1230 Elm St', 'person10@example.com', '5
26 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1353 Elm St', 'person11@example.com', '5
27 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1476 Elm St', 'person12@example.com', '5
28 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1599 Elm St', 'person13@example.com', '5
29 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1722 Elm St', 'person14@example.com', '5
30 INSERT INTO Person (Address, Mail, MainPhone, Age, Pname) VALUES ('1845 Elm St', 'person15@example.com', '5
31 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Assistant Manager', 14, '5555832');
32 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Senior Manager', 16, '5552340');
33 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Event Coordinator', 10, '5553380');
34 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Operations Manager', 10, '5553368');
35 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Event Coordinator', 12, '5551119');
36 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Operations Manager', 13, '5559394');
37 INSERT INTO Manager (MRole, Seniority, MainPhone) VALUES ('Junior Manager', 17, '5552448');

INSERT INTO ActivityOperator (OCost, Expertise, Gender, MainPhone) VALUES (121.65, 'Science educator', 'Female
INSERT INTO ActivityOperator (OCost, Expertise, Gender, MainPhone) VALUES (129.03, 'Yoga guide', 'Male', '5555
INSERT INTO ActivityOperator (OCost, Expertise, Gender, MainPhone) VALUES (151.64, 'Yoga guide', 'Female', '55
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user1', 1444, 'Child', '5555832');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user2', 6240, 'Senior', '5552340');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user3', 7097, 'Adult', '5553380');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user4', 3597, 'Senior', '5553368');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user5', 2017, 'Adult', '5551119');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user6', 2010, 'Child', '5559394');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user7', 2307, 'Senior', '5552448');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user8', 1861, 'Teen', '5551604');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user9', 8738, 'Adult', '5552568');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user10', 7032, 'Teen', '5550968');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user11', 8630, 'Adult', '5559933');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user12', 5450, 'Child', '5553686');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user13', 6976, 'Adult', '5557172');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user14', 2355, 'Child', '5559074');
INSERT INTO Participant (UserName, PPassword, PType, MainPhone) VALUES ('user15', 3378, 'Senior', '5558034');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5557095', '5555832');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5556526', '5552340');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5556835', '5553380');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5553134', '5553368');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5552321', '5551119');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5554131', '5559394');
INSERT INTO Person_More_Phone (More_Phone, MainPhone) VALUES ('5557095', '5552448');
```


הכנסת נתונים דרך Mockaroo

שימוש באתר:

The screenshot shows the Mockaroo website interface. At the top, there's a navigation bar with links: SCHEMAS, DATASETS, MOCK APIS, SCENARIOS, PROJECTS, FUNCTIONS, and buttons for SIGN IN and UPGRADE NOW. Below the navigation bar, a message states: "Need more data? Plans start at just \$60/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud." The main area is a configuration table for a dataset named "Event_Participant".

Field Name	Type	Options
Review	ICD9 Proc Desc (Sh...	blank: 0% [Σ] [X]
SeatNumber	Number	min: 12 max: 999 decimals: 0 blank: 0% [Σ] [X]
CardNum	Number	min: 1000 max: 9000 decimals: 0 blank: 0% [Σ] [X]
FlightDate	Datetime	05/22/2023 to 05/22/2024 format: m/d/yyyy blank: 0% [Σ] [X]
ZIP	Phone	format: #####-#### blank: 0% [Σ] [X]
MainPhone	Phone	format: #####-#### blank: 0% [Σ] [X]

Below the table, there are buttons: "+ ADD ANOTHER FIELD" and "GENERATE FIELDS USING AI...". At the bottom, there's a section for "# Rows: 400", "Format: SQL", "Table Name: Event_Participant", and a checkbox "Include CREATE TABLE". At the very bottom, there are buttons: "GENERATE DATA", "PREVIEW", "SAVE AS...", "DERIVE FROM EXAMPLE...", and "MORE".

חלק מן הפקודות שנוצרו:

```
C: > Users > daniel > Desktop > Person.sql
1 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('59746 Mallard Avenue', 'jyepiskov0@netlog.com', '7938262589', '
2 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('72678 Roxbury Alley', 'gmerdew1@etsy.com', '1794405930', '14-34
3 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('074 8th Drive', 'orapps2@hatena.ne.jp', '8209392764', '80-896-2
4 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('7 Meadow Valley Way', 'ehathway3@phpbb.com', '2447576877', '76-
5 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('6465 Walton Road', 'tjohl14@about.com', '5663046222', '63-070-07
6 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('58 Sullivan Junction', 'syoun5@technorati.com', '2711538487',
7 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('404 Beilfuss Park', 'fpapierz6@digg.com', '4461171750', '95-836
8 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('7577 Sullivan Lane', 'tplimmer7@ocn.ne.jp', '8125534679', '13-6
9 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('892 Mendota Junction', 'htease8@timesonline.co.uk', '4512841262
10 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('9422 Sullivan Crossing', 'khilllock9@gov.uk', '7735645405', '53-
11 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('86 Clemons Plaza', 'plyosika@de.vu', '5799439646', '35-002-8560
12 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('575 Derek Trail', 'mbutterfieldb@squarespace.com', '2091681671'
13 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('63837 Erie Terrace', 'morisc@woothemes.com', '9762559797', '77-
14 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('15624 Superior Parkway', 'lperisd@marketwatch.com', '2072913923
15 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('057 Swallow Pass', 'kbingee@dailymail.co.uk', '2352130089', '90
16 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('8395 Arkansas Court', 'nmethamf@hatena.ne.jp', '6166375696', '9
17 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('02775 Anzinger Court', 'nkinsong@upenn.edu', '4637527717', '40-
18 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('819 Village Green Hill', 'bcantrillh@columbia.edu', '4563471949
19 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('11 Luster Avenue', 'ahurlstoni@xing.com', '1484501201', '20-858
20 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('63 Crowley Crossing', 'cvanz@army.mil', '1822359497', '36-023-4
21 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('0 Saint Paul Terrace', 'kkittlessk@canalblog.com', '3765529169'
22 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('24 Claremont Way', 'jbarringer1@hugedomains.com', '5419266601',
23 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('5919 Tennyson Plaza', 'adem@gizmodo.com', '9001060105', '18-812
24 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('4 Cottonwood Terrace', 'tpressn@sitemeter.com', '2638940170', '
25 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('14494 Messerschmidt Terrace', 'mwolferso@foxnews.com', '3039574
26 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('68 Oakridge Circle', 'rwillgrassp@forbes.com', '1546210565', '4
27 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('1249 Heath Terrace', 'ecolqueranq@blog.com', '1094143040', '89-
28 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('5969 Spaight Place', 'rgumleyr@aboutads.info', '7316622972', '4
29 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('32546 Holmberg Alley', 'fdencss@fema.gov', '4705691089', '81-15
30 insert into Person (Address, Mail, MainPhone, Age, Pname) values ('471 Hazelcrest Pass', 'mboltt@bluehost.com', '4456145153', '06-
```

גיבוי ושחזור הטבלאות

גיבוי:

Name	Type	Compiled
ACTIVITY_OPERATOR	TABLE	22/05/2024 21:58:34
EVENT	TABLE	22/05/2024 21:58:34
EVENT_OPERATOR	TABLE	22/05/2024 21:58:34
EVENT_PARTICIPANT	TABLE	22/05/2024 21:58:34
HALL	TABLE	22/05/2024 21:58:34
MANAGER	TABLE	22/05/2024 21:58:34
MANAGER_WORKING_DAYS	TABLE	22/05/2024 21:58:34
MVIEWS_ADV_AUG	TABLE	04/02/2019 04:48:34
MVIEWS_ADV_BASSETABLE	TABLE	04/02/2019 04:48:34
MVIEWS_ADV_CLIQUE	TABLE	04/02/2019 04:48:34
MVIEWS_ADV_ELIGIBLE	TABLE	04/02/2019 04:48:34
MVIEWS_ADV_EXCEPTIONS	TABLE	04/02/2019 04:48:34
PARTICIPANT	TABLE	22/05/2024 21:58:34
PERSON	TABLE	22/05/2024 21:58:34
PERSON_MORE_PHONE	TABLE	22/05/2024 21:58:34
SQLPLUS_PRODUCT_PROFILE	TABLE	04/02/2019 05:51:04

User: <CURRENT USER>

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)

☐ Zip
 Where clause:

Output file: C:\Users\halel\OneDrive - g.jct.ac.il\מחברים\מייניפ בסיסי נתונים\backup22.05.24.sql

Export

SYSTEM@XE Exporting... Done

שחזור:

שלב ב' - שאלות ואילוצים

שאלות select:

1. שאלת שמחזירה את רשימת המנהלים ה"מוצלחים" ביותר. כלומר, כל המנהלים שממוצע המשובים שנתנו משתתפים שנרשמו לפעילויות שניהלו גבוה מ 6. הרשימה תכיל את השם, הטלפון וממוצע המשובים של כל מנהל, ותמוין מהמוצלח ביותר כלפי מטה.

```
SELECT
    p.Pname AS ManagerName,
    m.MainPhone AS ManagerPhone,
    ROUND(AVG(ep.Review), 2) AS AverageReview
FROM
    Manager m
JOIN
    Person p ON m.MainPhone = p.MainPhone
JOIN
    Event e ON m.MainPhone = e.MainPhone
JOIN
    Event_Participant ep ON e.EID = ep.EID
GROUP BY
    p.Pname, m.MainPhone
HAVING
    ROUND(AVG(ep.Review), 2) > 6
ORDER BY
    AverageReview DESC;
```

הרצה ותוצאות השאלת:

		MANAGERNAME	MANAGERPHONE	AVERAGEREVIEW
▶	1	Radney Ruffalo	0577661021	6.63
	2	Delroy Epps	0593215466	6.63
	3	Holland D'Onofrio	0555493632	6.57
	4	Howie Colin Young	0593311899	6.45
	5	Jim Hawn	0549751262	6.43

< [11:40:06] 21 rows selected in 0.120 seconds

2. שאילתא שמחזירה מידע על אירועים שיש בהם בעיית מקום. כלומר שמספר הרשומים אליהם גבוה מתכולת האולם בו הם משובצים.
עבור כל אירוע כזה נקבל את מס' האירוע, מס' האולם, מספר הנרשמים, ותכולת האולם.

```
SELECT
    e.EID AS EventNumber,
    e.HID AS HallNumber,
    h.Capacity,
    COUNT(ep.MainPhone) AS NumberOfRegistrants
FROM
    Event e
JOIN
    Hall h ON e.HID = h.HID
JOIN
    Event_Participant ep ON e.EID = ep.EID
GROUP BY
    e.EID, e.HID, h.Capacity;
HAVING
    COUNT(ep.MainPhone) > h.Capacity;
```

הרצה ותוצאות השאילתא:

	EVENTNUMBER	HALLNUMBER	CAPACITY	NUMBEROFREGISTRANTS
1	136	266	30	33
2	377	257	30	42
3	300	13	30	42
4	17	181	30	31
5	67	243	30	34

13:29 SYSTEM@XE [11:41:11] 30 rows selected in 0.077 seconds

3. שאילתא שמחזירה מידע אודות "הרשמה מוקדמת". כלומר, רישומים לאירועים שהתבצעו לפחות שבוע מראש מתאריך האירוע. נקבל את שם המשתתף, הטלפון שלו, מספר האירוע, תאריך האירוע, ותאריך הרישום.

```
SELECT
    p.Pname AS ParticipantName,
    ep.MainPhone,
    e.EID AS EventNumber,
    e.edate,
    ep.signupdate
FROM
    Event_Participant ep
JOIN
    Participant par ON ep.MainPhone = par.MainPhone
JOIN
    Person p ON par.MainPhone = p.MainPhone
JOIN
    Event e ON ep.EID = e.EID
WHERE
    ep.SignUpDate <= e.EDate - 7;
```

הרצה ותוצאות השאילתא:

	PARTICIPANTNAME	MAINPHONE	EVENTNUMBER	EDATE	SIGNUPDATE
1	Julie Diesel	0563453847	52	03/08/2011	09/12/2009
2	Liquid Schiavelli	0506289997	364	03/02/2008	02/09/2001
3	Elias Armatrading	0515943261	386	11/04/2012	12/10/2007
4	Russell Spacey	0546101500	73	25/07/2012	19/10/2001
5	Lila Miller	0590145621	6	03/11/2004	06/02/2002

10:51 0:01 SYSTEM@XE [11:41:53] 4921 rows selected in 1.110 seconds

4. שאילתא שמחזירה את סוג הפעילות המועדפת על כל קבוצת גיל. כלומר, עבור כל קבוצת גיל את סוג הפעילות שיש הכי הרבה הרשמות לפעילויות מסוג זה. נקבל רשימה של קבוצות הגיל, ממוינות בסדר עולה, והקטגוריה המועדפת לכל קבוצה.

```
WITH AgeGroups AS (
    SELECT
        CASE
            WHEN Age BETWEEN 7 AND 10 THEN 'Mini-Club'
            WHEN Age BETWEEN 11 AND 14 THEN 'Kids-Club'
            WHEN Age BETWEEN 15 AND 18 THEN 'Teen-Club'
            WHEN Age BETWEEN 19 AND 60 THEN 'Adults-Club'
            WHEN Age BETWEEN 61 AND 90 THEN 'Elders-Club'
            ELSE 'Unknown'
        END AS AgeGroup,
        MainPhone
    FROM
        Person
)

SELECT
    AgeGroup,
    (
        SELECT
            e.ECategory
        FROM
            Event e
        JOIN
            Event_Participant ep ON e.EID = ep.EID
        JOIN
            AgeGroups ag ON ep.MainPhone = ag.MainPhone
        WHERE
            ag.AgeGroup = ag_outer.AgeGroup
        GROUP BY
            e.ECategory
        ORDER BY
            COUNT(*) DESC
        FETCH FIRST 1 ROW ONLY
    ) AS MostPopularActivity
FROM
    AgeGroups ag_outer
GROUP BY
    AgeGroup
ORDER BY
    CASE
        WHEN AgeGroup = 'Mini-Club' THEN 1
        WHEN AgeGroup = 'Kids-Club' THEN 2
        WHEN AgeGroup = 'Teen-Club' THEN 3
        WHEN AgeGroup = 'Adults-Club' THEN 4
        WHEN AgeGroup = 'Elders-Club' THEN 5
        ELSE 6
    END;
```

הרצה ותוצאות השאילתא:

	AGEGROUP	MOSTPOPULARACTIVITY	
▶ 1	Mini-Club	educational activity	...
2	Kids-Club	educational activity	...
3	Teen-Club	educational activity	...
4	Adults-Club	educational activity	...
5	Elders-Club	educational activity	...

SYSTEM@XE [11:43:03] 5 rows selected in 0.150 seconds

שאלות delete:

1. שאלתא שמוחקת אירועים שנקבעו ליום שאינו יום עבודה של המנהל האחראי עליהם. בהתאם נמחקים כל הרישומים לאירוע זה, וכן הקשר עם המפעיל שהוזמן לאירוע זה.

```
DELETE FROM Event_Participant ep
WHERE ep.EID IN (
    SELECT e.EID
    FROM Event e
    WHERE TO_CHAR(e.EDate, 'Day') NOT IN (
        SELECT mw.Working_Days
        FROM Manager_Working_Days mw
        WHERE mw.MainPhone = e.MainPhone
    )
);

DELETE FROM Event_Operator eo
WHERE eo.EID IN (
    SELECT e.EID
    FROM Event e
    WHERE TO_CHAR(e.EDate, 'Day') NOT IN (
        SELECT mw.Working_Days
        FROM Manager_Working_Days mw
        WHERE mw.MainPhone = e.MainPhone
    )
);

DELETE FROM Event e
WHERE TO_CHAR(e.EDate, 'Day') NOT IN (
    SELECT mw.Working_Days
    FROM Manager_Working_Days mw
    WHERE mw.MainPhone = e.MainPhone
);
```

לפני ההרצה, נוכל לראות אירועים שהיום שנקבע להם אינו מתוך ימי העבודה של המנהל:

	EVENTNUMBER	DAYOFWEEK	MANAGERPHONE	WORKINGDAYS
▶ 1	1	Friday	0557673859	Tuesday
2	2	Wednesday	0557707249	Thursday
3	3	Friday	0577661021	Tuesday
4	4	Wednesday	0509619309	Thursday, Wednesday

הרצה:

SYSTEM@XE [11:24:35] 406 rows deleted in 0.015 seconds

לאחר ההרצה, נותרו רק אירועים שהם באחד מימי העבודה של המנהל שלהם:

	EVENTNUMBER	DAYOFWEEK	MANAGERPHONE	WORKINGDAYS
▶ 1	5	Wednesday	0509619309	Thursday, Wednesday
2	7	Wednesday	0591946782	Wednesday
3	59	Wednesday	0599777765	Wednesday
4	73	Wednesday	0583206356	Monday, Wednesday

2. הספרייה החליטה שהתנדבות אינה סוג פעילות מתאים לילדים מתחת לגיל עשר. השאילתא להלן מוחקת את השתתפויותיהם של ילדים מתחת לגיל 10 מאירועים שבקטגוריה "התנדבות".

```
DELETE FROM
    Event_Participant
WHERE
    EID IN (
        SELECT e.EID
        FROM Event e
        WHERE e.ECategory = 'volunteering'
    )
AND
    MainPhone IN (
        SELECT p.MainPhone
        FROM Person p
        WHERE p.Age < 10
    );
```

לפני ההרצה ניתן למצוא רישומים של ילדים עד גיל 10, שרשומים לאירוע מסוג התנדבות:

	EID	AGE	ECATEGORY
1	3	9	volunteering
2	233	7	volunteering
3	326	8	volunteering
4	326	8	volunteering
5	27	7	volunteering

נריץ:

```
[11:34:18] 39 rows deleted in 0.031 seconds
```

לאחר ההרצה לא נמצאו ילדים עד גיל 10 שרשומים להתנדבות:

EID	AGE	ECATEGORY

SYSTEM@XE [11:35:09] 0 rows selected in 0.075 seconds

שאלות update:

1. שאילתא שמעדכנת את המחיר של אירועים שרוצים להגדיל את הרישום עליהם. מפחיתה ב 20% את מחיר הרישום לכל אירוע שבו מספר הנרשמים עד כה קטן מ-50, וכן נותר עוד מקום באולם בו מתקיים האירוע.

```
-- Update the price of events with fewer than 50 participants and
available space in the hall
UPDATE Event e
SET e.Price = CEIL(e.Price * 0.80)
WHERE e.EID IN (
    SELECT e.EID
    FROM Event e
    JOIN Hall h ON e.HID = h.HID
    LEFT JOIN (
        SELECT ep.EID, COUNT(ep.MainPhone) AS ParticipantCount
        FROM Event_Participant ep
        GROUP BY ep.EID
    ) epc ON e.EID = epc.EID
    WHERE epc.ParticipantCount < 50
    AND h.Capacity - COALESCE(epc.ParticipantCount, 0) > 0
);
```

נראה את מחירם של אירועים מיועדים להנחה לפני ההרצה.
ניתן לראות שמספר הנרשמים קטן מ-50, ושיש עוד מקום באולם:

		EVENTNUMBER	EVENTPRICE	NUMBEROFREGISTRANTS	NUMBEROFSEATS
▶	6	141	51.07	40	300
	7	231	39.12	41	200
	8	237	96.66	37	200
	9	272	17.90	39	300
	10	289	213.44	39	200

הרצה:

Update event

Commit

21:1

SYSTEM@XE [11:46:07] 390 rows updated in 0.031 seconds

לאחר ההרצה- ניתן לראות שהמחיר לכרטיס עבור אירועים אלו הוזל ב 20%:

6	141	41.00	40	300
7	231	32.00	41	200
8	237	78.00	37	200
9	272	15.00	39	300

2. שאילתא שמעדכנת אירועים ששובצו להתקיים בקומה גבוהה (קומה 2 ומעלה) וקהל היעד שלהם הוא גיל הזהב, להיות מיועדים למבוגרים ולא לגיל הזהב. מכיוון שהעלייה במדרגות לא מתאימה לקהל יעד זה.

```
UPDATE Event e
SET e.Audience = 'Adults'
WHERE e.Audience = 'Elderly'
AND e.HID IN (
  SELECT h.HID
  FROM Hall h
  WHERE REGEXP_SUBSTR(h.Place, 'floor: ([0-9]+)') IS NOT NULL
  AND TO_NUMBER(REGEXP_SUBSTR(h.Place, 'floor: ([0-9]+)', 1, 1,
NULL, 1)) >= 2
);
```

לפני ההרצה- ישנם אירועים שמתקיימים בקומה גבוהה ומיועדים לגיל הזהב:

	EVENTNUMBER	TARGETAUDIENCE	HALLLOCATION
▶ 1	5	Elderly	floor: 6 room: 6078
2	7	Children	floor: 5 room: 5067
3	59	Children	floor: 4 room: 4056
4	73	Elderly	floor: 5 room: 5067
5	120	Adults	floor: 2 room: 2034

הרצה:

Update event
Commit

SYSTEM@XE [11:50:22] 69 rows updated in 0.009 seconds

לאחר ההרצה:

	EVENTNUMBER	TARGETAUDIENCE	HALLLOCATION
▶ 1	5	Adults	floor: 6 room: 6078
2	7	Children	floor: 5 room: 5067
3	59	Children	floor: 4 room: 4056
4	73	Adults	floor: 5 room: 5067
5	120	Adults	floor: 2 room: 2034

שאלות עם פרמטרים:

1. השאלתא מחזירה את כל המנהלים שממוצע הדירוגים של אירועים שניהלו גבוהים מהמספר המבוקש.
המשתמש יכניס דירוג (בין 1 ל 10), ויקבל את רשימת כל המנהלים שממוצע הדירוג של אירועים שניהלו גבוה מציון זה. עבור כל מנהל יוחזר שמו, מס' הפלאפון שלו, והממוצע. אם המשתמש לא בוחר אחרת, ברירת המחדל היא 6.

```
--Query 1: high rating managers
SELECT
    p.Pname AS ManagerName,
    m.MainPhone AS ManagerID,
    ROUND(AVG(ep.Review), 2) AS AverageReview
FROM
    Manager m
JOIN
    Person p ON m.MainPhone = p.MainPhone
JOIN
    Event e ON m.MainPhone = e.MainPhone
JOIN
    Event_Participant ep ON e.EID = ep.EID
GROUP BY
    p.Pname, m.MainPhone
HAVING
    ROUND(AVG(ep.Review), 2) > <name="min rating" hint="min rating
between 3-10" type="integer" default=6>
ORDER BY
    AverageReview DESC;
```

הרצה:

Name	Value
min rating	6

min rating between 3-10

OK Cancel Clear

תוצאות השאלתא (עבור 6):

	MANAGERNAME	MANAGERID	AVERAGEREVIEW
1	Radney Ruffalo	0577661021	6.63
2	Delroy Epps	0593215466	6.63
3	Holland D'Onofrio	0555493632	6.57
4	Howie Colin Young	0593311899	6.45
5	Jim Brown	0540751262	6.42

SYSTEM@XE [12:51:32] 21 rows selected in 0.077 seconds

2. השאילתא מחזירה את פרטי המשתתפים מבין טווח גילאים רצוי, שמשתתפים בפעילויות מהקטגוריות המבוקשות. המשתמש בוחר קטגוריות (ניתן לבחור יותר מאחת), בוחר גיל מינימלי וגיל מקסימלי (ברירת המחדל היא 10 ו-50), ומקבל את פרטי הרישומים של אנשים מטווח גילאים זה לאירועים מקטגוריות אלו.

```
--Query 2: participant in range of ages categories
select * FROM Event_Participant
WHERE
    EID IN (
        SELECT e.EID
        FROM Event e
        WHERE e.ECategory = &<multiselect="yes" name="Event category"
list="'Lecture', 'workshop', 'performance', 'story', 'educational
activity', 'play', 'conversation with an author', 'educational
activity', 'training', 'film screening', 'exhibition', 'sports',
'volunteering' required=true >
    )
AND
    MainPhone IN (
        SELECT p.MainPhone
        FROM Person p
        WHERE p.Age > &<name="min age" type="integer" default=10> and
p.Age < &<name="max age" type="integer" default=50>
    );
```

הרצה:

Select values

☐ 'Lecture'
☒ 'workshop'
☐ 'performance'
☐ 'story'
☐ 'educational activity'
☐ 'play'
☐ 'conversation with an author'
☐ 'educational activity'
☐ 'training'
☐ 'film screening'
☐ 'exhibition'
☒ 'sports'
☐ 'volunteering'

OK

Cancel

Clear

Variables

Name	Value
Event category	'workshop','sports'
min age	10
max age	50

OK

Cancel

Clear

תוצאות: כל הרישומים של משתתפים מגיל 10 עד 50 לאירועים מסוג ספורט ומסוג סדנא-

	REVIEW	SEATNUMBER	CARDNUM	SIGNUPDATE	EID	MAINPHONE
1	8	115	716	19/10/2001	73	0546101500
2	2	127	227	03/09/2000	318	0582556077
3	3	132	744	27/05/2021	121	0530325327
4	9	79	590	06/06/2015	328	0547939892
5	8	285	385	18/08/2013	272	0562033925

36:1

SYSTEM@XE [13:06:41] 945 rows selected in 0.471 seconds

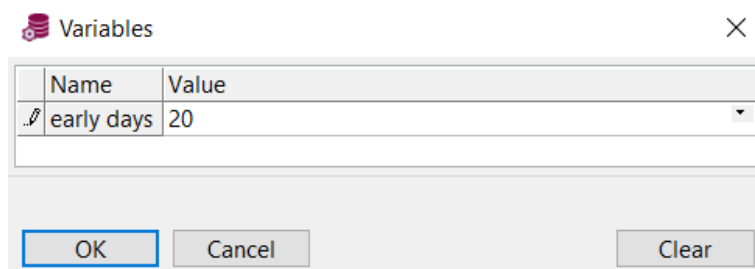
3. השאילתא מחזירה רשימה של רישומים מוקדמים לאירועים.

המשתמש מכניס מספר ימים, נניח X, ומקבל את הפרטים של כל הרישומים (שם ומס' טלפון של המשתתף, מס' האירוע, תאריך הרישום ותאריך האירוע) שהתבצעו X ימים מראש. ברירת המחדל היא 7 ימים.

--Query 3: early sign up

```
SELECT
    p.Pname AS ParticipantName,
    ep.MainPhone,
    e.EID AS EventNumber,
    e.edate,
    ep.signupdate
FROM
    Event_Participant ep
JOIN
    Participant par ON ep.MainPhone = par.MainPhone
JOIN
    Person p ON par.MainPhone = p.MainPhone
JOIN
    Event e ON ep.EID = e.EID
WHERE
    ep.SignUpDate <= e.EDate - &<name="early days" type=integer
default=7>;
```

הרצה:



Name	Value
early days	20

OK Cancel Clear

ביקשנו הרשמות של 20 ימים מראש.

תוצאות:

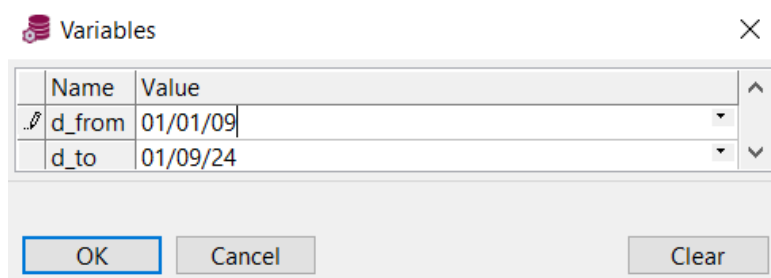
		PARTICIPANTNAME	MAINPHONE	EVENTNUMBER	EDATE	SIGNUPDATE
▶	1	Julie Diesel	0563453847	52	03/08/2011	09/12/2009
	2	Liquid Schiavelli	0506289997	364	03/02/2008	02/09/2001
	3	Elias Armatrading	0515943261	386	11/04/2012	12/10/2007
	4	Russell Spacey	0546101500	73	25/07/2012	19/10/2001

< 39:1 0:01 SYSTEM@XE [13:21:09] 4901 rows selected in 1.552 seconds

4. השאילתא מבקשת מהמשתמש טווח תאריכים (תאריך התחלה ותאריך סיום), ומחזירה את כל המפעילים שהוזמנו בתקופה זו ולא היה כדאי להזמין אותם: כל המפעילים שמחירים גבוה מ 2000 ש, ובפעילויות שהעבירו בתקופה הזו השתתפו בממוצע פחות מ 50 איש. עבור כל מפעיל כזה נקבל את מס' הפלאפון שלו, עלותו, ומס' המשתתפים הממוצע בפעילויות שלו בתקופה זו.

```
--Query 4: Operators that should not have been ordered, in a specific
period
SELECT AO.MainPhone, AO.OCost, round(AVG(EP.ParticipantCount),0) AS
AvgParticipants
FROM ActivityOperator AO
JOIN Event_Operator EO ON AO.MainPhone = EO.MainPhone
JOIN Event E ON EO.EID = E.EID
JOIN (
    SELECT EID, COUNT(MainPhone) AS ParticipantCount
    FROM Event_Participant
    GROUP BY EID
) EP ON E.EID = EP.EID
WHERE E.EDate BETWEEN <name="d_from" type="date" default="01/01/24">
and <name="d_to" type="date" default="01/09/24">
AND AO.OCost > 2000
GROUP BY AO.MainPhone, AO.OCost
HAVING AVG(EP.ParticipantCount) < 50;
```

הרצה:



Name	Value
d_from	01/01/09
d_to	01/09/24

תוצאות (עבור ערכי ברירת המחדל, שבצילום לעיל):

		MAINPHONE	OCOST	AVGPARTICIPANTS
▶	1	0516940381	9147.54	32
	2	0592988124	6765.61	32
	3	0516056811	2417.05	37
	4	0501194171	6779.04	31
	5	0550017077	6100.10	31

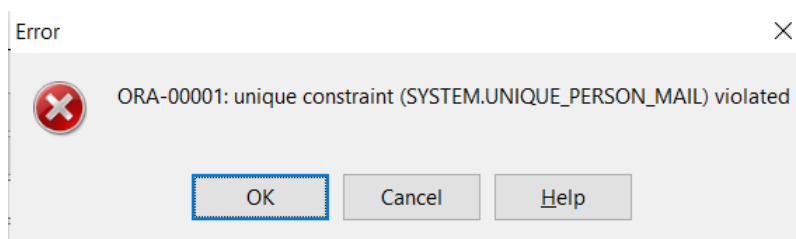
59:1 SYSTEM@XE [13:30:55] 150 rows selected in 0.131 seconds

אילוצים:

1. הוספת אילוץ unique עבור כתובת אימייל בטבלה Person.
כלומר, לא יתאפשר להכניס כתובת אימייל שכבר שמורה עבור אדם אחר.

```
ALTER TABLE Person
ADD CONSTRAINT unique_person_mail UNIQUE (Mail);
```

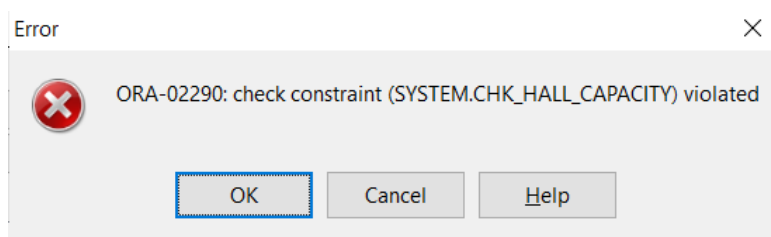
ננסה להכניס עבור שני אנשים את אותה כתובת אימייל, וקיבלנו את השגיאה הבאה:



2. הוספת אילוץ check לתכונת ה capacity בטבלה Hall. כך שקיבולת האולם חייבת להיות גדולה מ 0 אנשים.

```
ALTER TABLE Hall
ADD CONSTRAINT chk_hall_capacity CHECK (Capacity > 0);
```

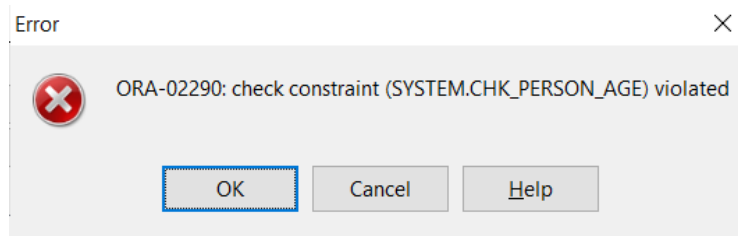
ננסה להכניס אולם עם קיבולת 0 ונקבל שגיאה:



3. הוספת אילוץ check לתכונה age בטבלה Person, כך שהגיל של אדם חייב להיות מעל 0.

```
ALTER TABLE Person
ADD CONSTRAINT chk_person_age CHECK (Age > 0);
```

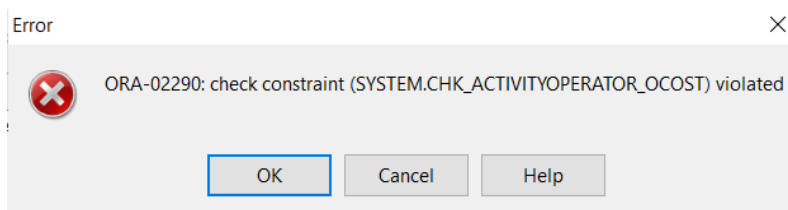
ננסה להכניס אדם שגילו 0 ונקבל שגיאה:



4. הוספת אילוץ check לתכונה cost בטבלה operator, כך שעלות של מפעיל לא יכולה להיות מספר שלילי.

```
ALTER TABLE ActivityOperator
ADD CONSTRAINT chk_activityoperator_ocost CHECK (OCost >= 0);
```

ננסה להכניס מפעיל שעלותו 1- ונקבל שגיאה:



5. הוספת default עבור התכונה price בטבלה event, כך שמחיר ברירת מחדל של אירוע יהיה 50 ₪.

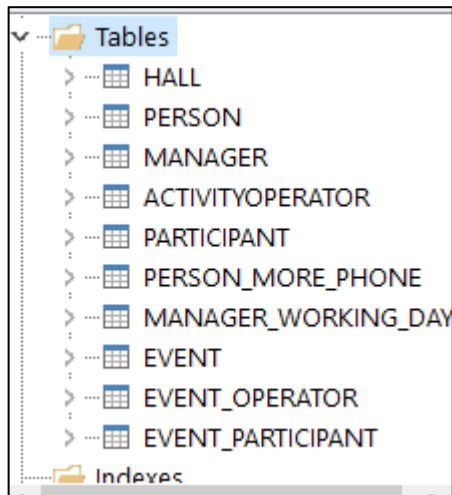
נכניס אולם חדש בלי לשלוח ערך עבור עלות האירוע, ונראה שנכנס הערך 50 בתכונה :PRICE

```
INSERT INTO Event (EDate, ETime, Audience, ECategory, ActivityD, Duration, EID, HID, MainPhone)
VALUES (TO_DATE('2024-06-01', 'YYYY-MM-DD'),
1800, 'General', 'Music Concert', 'Live concert by popular band', 120, 1, 9, '5559876641');

select* from event where eid=1;
```

	EDATE	ETIME	AUDIENCE	ECATEGORY	ACTIVITYD	PRICE	DURATION	EID	HID	MA
1	01/06/2024	1800	General	Music Concert	Live concert by popular band	50.00	120	1	9	555

Oracle Import	SQL Inserts	PL/SQL Developer	Log
<input type="checkbox"/> Use Command Window <input checked="" type="checkbox"/> Use SQL*Plus SQL*Plus Executable C:\app\halel\product\18.0.0\dbhomeXE\bin\sqlplus.exe			



Import file
C:\Users\halel\OneDrive - g.jct.ac.il\מיון נתונים\מיון נתונים\מיון נתונים\backup22.05.24.sql
SYSTEM@XE Executing Oracle SQL*Plus Utility, please wait... Done

שלב ג' - תכניות

תכנית א'-updateOperatorPrace :

התכנית מבצעת שינויים באירועים עתידיים לפי אימות היותו של מפעיל מסוים "מפעיל ברמה גבוהה"

התכנית מקבלת מהמשתמש מספר פלאפון של מפעיל (אם המספר אינו של מפעיל, תיזרק שגיאה מתאימה).

ראשית, היא בודקת האם המפעיל הנ"ל הוא מפעיל מוצלח, על ידי קריאה לפונקציה DeservesRaiseOperator. הפונקציה בודקת האם ממוצע כל המשובים שנתנו משתתפים עבור אירועים שהעביר מפעיל זה הינו 6 ומעלה. אם כן, הפונקציה תחזיר אמת. אחרת- תחזיר שקר.

לאחר מכן התכנית תבצע הסתעפות על פי הערך הבוליאני המוחזר. אם חזר שקר (המפעיל לא נחשב "ברמה גבוהה") – יודפס שהמפעיל אינו עונה להגדרה, והתכנית תסתיים. אם יוחזר אמת – תיקרא הפרוצדורה ManageOperatorEvents. הפרוצדורה תקבל את מספר המפעיל ותבצע מספר שינויים בנתונים:

א. תמצא מבין האירועים העתידיים שמועברים על ידי מפעיל זה את שלושת האירועים שאליהם רשום המספר הגבוה ביותר של משתתפים, ותעלה את המחיר לכרטיס ב 20%.

ב. תמצא את המפעיל הגרוע ביותר (זה שממוצע הדירוגים שנתנו המשתתפים באירועים שהעביר הוא הנמוך ביותר), ותחליף אותו במפעיל הנ"ל (שנמצא כי הוא "ברמה גבוהה"). כלומר, עבור כל אירוע עתידי של המפעיל הגרוע – יוחלף המפעיל למפעיל שלנו, באם הדבר מתאפשר ואין התנגשות של תאריכים.

קוד התכנית:

```
DECLARE
    --v_OperatorPhone ActivityOperator.MainPhone%TYPE :=
    '&operator_phone';
    v_DeservesRaise BOOLEAN;
BEGIN
    -- Call the function
    v_DeservesRaise := DeservesRaiseOperator(p_OperatorPhone=>
:p_OperatorPhone);
    IF v_DeservesRaise THEN
        -- Call the procedure
        DBMS_OUTPUT.PUT_LINE('The operator deserves to raise' ||
SQLERRM);
        ManageOperatorEvents(p_OperatorPhone=> :p_OperatorPhone);
    else
        DBMS_OUTPUT.PUT_LINE('The operator does not deserves to
raise' || SQLERRM);
    END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
```

קוד הפונקציה DeservesRaiseOperator:

```
CREATE OR REPLACE FUNCTION DeservesRaiseOperator(p_OperatorPhone IN
VARCHAR2)
RETURN BOOLEAN
AS
    v_AvgRating NUMBER := 0;
    v_PhoneExists NUMBER;
    e_OperatorNotFound EXCEPTION;
BEGIN
    -- Check if the phone number belongs to an operator
    SELECT COUNT(*)
    INTO v_PhoneExists
    FROM ActivityOperator
    WHERE MainPhone = p_OperatorPhone;

    IF v_PhoneExists = 0 THEN
        RAISE e_OperatorNotFound; -- Raise custom exception
    END IF;

    -- Calculate the average rating using the CalculateAverageRating
    function
    v_AvgRating := CalculateAverageRating(p_OperatorPhone);

    RETURN v_AvgRating > 5;
EXCEPTION
    WHEN e_OperatorNotFound THEN
        RAISE_APPLICATION_ERROR(-20001, 'The cell phone number
entered does not belong to an operator');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20012, 'An error occurred while
calculating the average rating: ' || SQLERRM);
END DeservesRaiseOperator;
```

קוד הפרוצדורה ManageOperatorEvents:

```
CREATE OR REPLACE PROCEDURE ManageOperatorEvents(
    p_OperatorPhone IN VARCHAR2
)
AS
    CURSOR upcoming_events IS
        SELECT e.EID, e.Price
        FROM Event e
        WHERE e.OpMainPhone = p_OperatorPhone
        AND e.EDate > SYSDATE
        ORDER BY (SELECT COUNT(*) FROM Event_Participant ep WHERE
ep.EID = e.EID) DESC
        FETCH FIRST 3 ROWS ONLY;

    CURSOR worst_operator IS
        SELECT ao.MainPhone
        FROM ActivityOperator ao;

    v_EID Event.EID%TYPE;
    v_Price Event.Price%TYPE;
    v_WorstOperatorPhone ActivityOperator.MainPhone%TYPE;
    v_EventConflict NUMBER;
    v_EventsUpdated NUMBER := 0;
```

```

v_EventsReassigned NUMBER := 0;
v_TotalEvents NUMBER := 0;
v_LowestAvgRating NUMBER := 1000;
v_CurrentAvgRating NUMBER;
BEGIN
    -- Debug print: Start of the procedure
    DBMS_OUTPUT.PUT_LINE('Starting ManageOperatorEvents procedure for
operator phone: ' || p_OperatorPhone);

    -- Count the total number of upcoming events for the operator
    SELECT COUNT(*)
    INTO v_TotalEvents
    FROM Event e
    WHERE e.OpMainPhone = p_OperatorPhone
    AND e.EDate > SYSDATE;

    DBMS_OUTPUT.PUT_LINE('The operator is responsible for ' ||
v_TotalEvents || ' upcoming events.');
```

-- Increase the price of the top 3 upcoming events by 20%

```

FOR event_rec IN upcoming_events LOOP
    v_EID := event_rec.EID;
    v_Price := event_rec.Price * 1.2;

    UPDATE Event
    SET Price = v_Price
    WHERE EID = v_EID;

    v_EventsUpdated := v_EventsUpdated + 1;
    DBMS_OUTPUT.PUT_LINE('Updated price for event ID: ' || v_EID
|| ', new price: ' || v_Price);
END LOOP;

DBMS_OUTPUT.PUT_LINE('The price of ' || v_EventsUpdated || '
events has been updated.');
```

-- Find the worst operator by calculating average ratings

```

OPEN worst_operator;
LOOP
    FETCH worst_operator INTO v_WorstOperatorPhone;
    EXIT WHEN worst_operator%NOTFOUND;

    v_CurrentAvgRating :=
CalculateAverageRating(v_WorstOperatorPhone);

    IF v_CurrentAvgRating < v_LowestAvgRating THEN
        v_LowestAvgRating := v_CurrentAvgRating;
        v_WorstOperatorPhone := v_WorstOperatorPhone;
    END IF;
END LOOP;
CLOSE worst_operator;

DBMS_OUTPUT.PUT_LINE('Worst operator phone: ' ||
v_WorstOperatorPhone);

IF v_WorstOperatorPhone IS NOT NULL THEN
    -- Reassign future events of the worst operator to the
specified operator
    FOR event_rec IN (
        SELECT EID, EDate, ETime
        FROM Event
```

```

        WHERE OpMainPhone = v_WorstOperatorPhone
        AND EDate > SYSDATE
    ) LOOP
        -- Check for scheduling conflicts
        SELECT COUNT(*)
        INTO v_EventConflict
        FROM Event e
        WHERE e.OpMainPhone = p_OperatorPhone
        AND e.EDate = event_rec.EDate
        AND e.ETime = event_rec.ETime;

        IF v_EventConflict = 0 THEN
            UPDATE Event
            SET OpMainPhone = p_OperatorPhone
            WHERE EID = event_rec.EID;

            v_EventsReassigned := v_EventsReassigned + 1;
            DBMS_OUTPUT.PUT_LINE('Reassigned event ID: ' ||
event_rec.EID || ' to operator phone: ' || p_OperatorPhone);
        ELSE
            DBMS_OUTPUT.PUT_LINE('Event ID ' || event_rec.EID ||
' on date ' || event_rec.EDate || ' at time ' || event_rec.ETime || '
cannot be reassigned due to a scheduling conflict.');


---



```


קוד פונקציית עזר לחישוב ממוצע הדירוגים של מפעיל – CalculateAverageRating:

```
CREATE OR REPLACE FUNCTION CalculateAverageRating(p_OperatorPhone IN
VARCHAR2)
RETURN NUMBER
AS
    v_TotalRating NUMBER := 0;
    v_Count NUMBER := 0;
    v_AvgRating NUMBER := 0;
BEGIN
    FOR rating_rec IN (
        SELECT ep.Review
        FROM Event e
        JOIN Event_Participant ep ON e.EID = ep.EID
        WHERE e.OpMainPhone = p_OperatorPhone
    )
    LOOP
        v_TotalRating := v_TotalRating + rating_rec.Review;
        v_Count := v_Count + 1;
    END LOOP;

    IF v_Count > 0 THEN
        v_AvgRating := v_TotalRating / v_Count;
    ELSE
        v_AvgRating := 0;
    END IF;

    RETURN v_AvgRating;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20014, 'An error occurred while
calculating the average rating: ' || SQLERRM);
END CalculateAverageRating;
```

דוגמאות הרצה:

מקרה א': מפעיל שכן ענה להגדרה "מפעיל ברמה גבוהה" ובוצעו עבורו העדכונים:

לפני ההרצה- אלו האירועים העתידיים שמפעיל המפעיל שמספרו 0596383100, ממיינים לפי מספר הנרשמים.

	EID	EDATE	ETIME	PRICE	MAINPHONE	OPMAINPHONE	NUMBEROFREGISTRANTS
1	431	13/06/2028	1000	151.30	0565313461	0596383100	4
2	271	24/07/2026	1930	544.18	0518436881	0596383100	3
3	5	12/07/2025	1600	116.74	0551442035	0596383100	2
4	132	04/02/2029	1200	229.50	0502526488	0596383100	1
5	185	24/04/2025	1600	61.64	0525467580	0596383100	0
6	177	19/09/2027	1830	165.93	0517368286	0596383100	0
7	382	13/11/2025	1500	139.76	0551442035	0596383100	0

הרצה:

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 [x] Enabled
The operator deserves to raiseORA-0000: normal, successful completion
Starting ManageOperatorEvents procedure for operator phone: 0596383100
The operator is responsible for 7 upcoming events.
Updated price for event ID: 431, new price: 181.56
Updated price for event ID: 271, new price: 653.02
Updated price for event ID: 5, new price: 140.09
The price of 3 events has been updated.
Worst operator phone: 0598398528
Reassigned event ID: 130 to operator phone: 0596383100
The operator with the lowest average reviews has been reassigned for 1 events.
4:32 SYSTEM@XE [11:26:30] Executed in 0.02 seconds

```

אחרי ההרצה- ניתן לראות שהמחיר של שלושת האירועים הראשונים השתנה, וכן שהתווסף אירוע נוסף (שהיה שייך למפעיל הגרוע ביותר):

	EID	EDATE	ETIME	PRICE	MAINPHONE	OPMAINPHONE	NUMBEROFREGISTRANTS
1	431	13/06/2028	1000	181.56	0565313461	0596383100	4
2	130	11/04/2027	1530	94.82	0573277445	0596383100	3
3	271	24/07/2026	1930	653.02	0518436881	0596383100	3
4	5	12/07/2025	1600	140.09	0551442035	0596383100	2
5	132	04/02/2029	1200	229.50	0502526488	0596383100	1
6	185	24/04/2025	1600	61.64	0525467580	0596383100	0
7	177	19/09/2027	1830	165.93	0517368286	0596383100	0
8	382	13/11/2025	1500	139.76	0551442035	0596383100	0

מקרה ב'- מפעיל שלא ענה להגדרה

תודפס הודעה מתאימה ולא יתבצעו שינויים בבסיס הנתונים:

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 [x] Enabled
The operator does not deserves to raiseORA-0000: normal, successful completion
2:1 SYSTEM@XE [11:31:11] Executed in 0.001 seconds

```

מקרה ג'- מקרה של שגיאה – אם הוכנס מספר פלאפון שאינו של מפעיל

תודפס הודעה מתאימה, ולא יתבצעו שינויים בבסיס הנתונים (פונקציית העדכון כלל לא נקראת):

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 [x] Enabled
An error occurred: ORA-20001: The cell phone number entered does not belong to an operator

```

תכנית ב'-hallsAndTrainingDay :

הנהלת הספרייה מעוניינת לקבוע יום השתלמות למנהלים.

ראשית, התכנית קוראת לפונקציה FindSuitableTrainingDay שתמצא תאריך מתאים ואולם מתאים להשתלמות. הפונקציה תחפש את היום מימי השבוע שבו יש מינימום מנהלים שזהו יום עבודה שלהם. לאחר מכן, תמצא אולם שבו אפשר לערוך את ההשתלמות בהקדם ביום זה בשבוע, אשר התכולה שלו מתאימה להכיל את כלל המנהלים של הספרייה. הפונקציה מחזירה את התאריך והאולם שנבחרו עבור יום ההשתלמות, והתוכנית תדפיס את מה שהתקבל.

בנוסף, ההנהלה מעוניינת בבדיקת העדפות הקהל ביחס למגדר המפעילים בסוגי הפעילויות השונות. לשם כך, התכנית מזמנת את הפרוצדורה PrintPreferredOperatorType. הפרוצדורה בודקת עבור כל קטגוריית אירועים (למשל- סדנאות/ ספורט/ הצגה/ וכו') האם מספר הנרשמים הממוצע גבוה יותר כאשר המפעיל הוא גבר או כאשר זו מפעילה אישה. התוצאות מודפסות במסך הקלט.

אם לא נמצא אולם מתאים או התרחשה כל שגיאה אחרת – תיזרק שגיאה.

קוד התכנית:

```
DECLARE
    v_TrainingDetails VARCHAR2(3000); -- Increase the size to handle
    larger return values
BEGIN
    -- Call the function FindSuitableTrainingDay
    v_TrainingDetails := FindSuitableTrainingDay;
    DBMS_OUTPUT.PUT_LINE('Training Details: ' || v_TrainingDetails);

    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE('');

    -- Call the procedure PrintPreferredOperatorType
    PrintPreferredOperatorType();

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
```

קוד הפונקציה FindSuitableTrainingDay:

```
CREATE OR REPLACE FUNCTION FindSuitableTrainingDay
RETURN VARCHAR2
IS
    v_min_managers_count INT := 900;
    v_min_managers_day VARCHAR2(10);
    v_managers_count INT;
    v_training_date DATE;
    v_hall_id INT;
    v_capacity INT;
    v_manager_count INT;
    v_event_count INT;

    CURSOR c_days IS
```

```

SELECT DISTINCT Working_Days
FROM Manager_Working_Days;

CURSOR c_managers(day IN VARCHAR2) IS
SELECT COUNT(*) AS manager_count
FROM Manager_Working_Days
WHERE Working_Days = day;

CURSOR c_halls(capacity_needed IN INT) IS
SELECT HID, Capacity
FROM Hall
WHERE Capacity >= capacity_needed
ORDER BY Capacity;

CURSOR c_events(hall_id IN INT, training_date IN DATE) IS
SELECT COUNT(*) AS event_count
FROM Event
WHERE HID = hall_id AND EDate = training_date; -- Ensure
EDate is the correct column name

BEGIN
    -- Find the day of the week with the minimum number of managers
    working
    FOR day_rec IN c_days LOOP
        OPEN c_managers(day_rec.Working_Days);
        FETCH c_managers INTO v_managers_count;
        CLOSE c_managers;

        IF v_managers_count < v_min_managers_count THEN
            v_min_managers_count := v_managers_count;
            v_min_managers_day := day_rec.Working_Days;
        END IF;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Minimum managers working day: ' ||
v_min_managers_day);

    -- Find the nearest date starting from today which is the desired
    day of the week.
    v_training_date := NEXT_DAY(SYSDATE, v_min_managers_day);

    DBMS_OUTPUT.PUT_LINE('Closest training date: ' ||
TO_CHAR(v_training_date, 'DD-MON-YYYY'));

    -- Find an available hall with sufficient capacity and no event
    on the training date
    SELECT COUNT(*) INTO v_manager_count FROM Manager;

    OPEN c_halls(v_manager_count);
    LOOP
        FETCH c_halls INTO v_hall_id, v_capacity;
        EXIT WHEN c_halls%NOTFOUND;

        -- Check if there is any event scheduled in the hall on the
        training date
        OPEN c_events(v_hall_id, v_training_date);
        FETCH c_events INTO v_event_count;
        CLOSE c_events;

        -- If no event is scheduled, select this hall
        IF v_event_count = 0 THEN

```

```

        EXIT;
    END IF;
END LOOP;
CLOSE c_halls;

IF v_event_count = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Selected hall ID: ' || v_hall_id || '
with capacity: ' || v_capacity);
    RETURN 'Training Date: ' || TO_CHAR(v_training_date, 'DD-MON-
YYYY') || ', Hall ID: ' || v_hall_id;
ELSE
    RETURN 'No suitable hall found.';
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No suitable hall found.');
```

RETURN 'No suitable hall found.';

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' ||

SQLERRM);

RETURN 'An unexpected error occurred.';

END FindSuitableTrainingDay;

קוד הפרוצדורה :PrintPreferredOperatorType

```

CREATE OR REPLACE PROCEDURE PrintPreferredOperatorType
IS
    v_avg_registrants_men NUMBER;
    v_avg_registrants_women NUMBER;

    -- Cursor to get the categories.
    CURSOR c_categories IS
        SELECT DISTINCT ECategory
        FROM Event;

    -- Cursor to get the average number of registrants for events by
    male operators
    CURSOR c_avg_registrants_men(category VARCHAR2) IS
        SELECT AVG(participant_count)
        FROM (
            SELECT e.EID, COUNT(p.mainPhone) AS participant_count
            FROM Event e
            JOIN Event_Participant p ON e.EID = p.EID
            JOIN ActivityOperator o ON e.OpMainPhone = o.MainPhone
            WHERE e.ECategory = category AND o.Gender = 'male'
            GROUP BY e.EID
        );

    -- Cursor to get the average number of registrants for events by
    female operators
    CURSOR c_avg_registrants_women(category VARCHAR2) IS
        SELECT AVG(participant_count)
        FROM (
            SELECT e.EID, COUNT(p.mainPhone) AS participant_count
            FROM Event e
            JOIN Event_Participant p ON e.EID = p.EID
            JOIN ActivityOperator o ON e.OpMainPhone = o.MainPhone
            WHERE e.ECategory = category AND o.Gender = 'female'
            GROUP BY e.EID
        );

```

```
);

BEGIN
  -- Loop through each category
  FOR category_rec IN c_categories LOOP
    -- Get the average number of registrants for male operators
    OPEN c_avg_registrants_men(category_rec.ECategory);
    FETCH c_avg_registrants_men INTO v_avg_registrants_men;
    CLOSE c_avg_registrants_men;

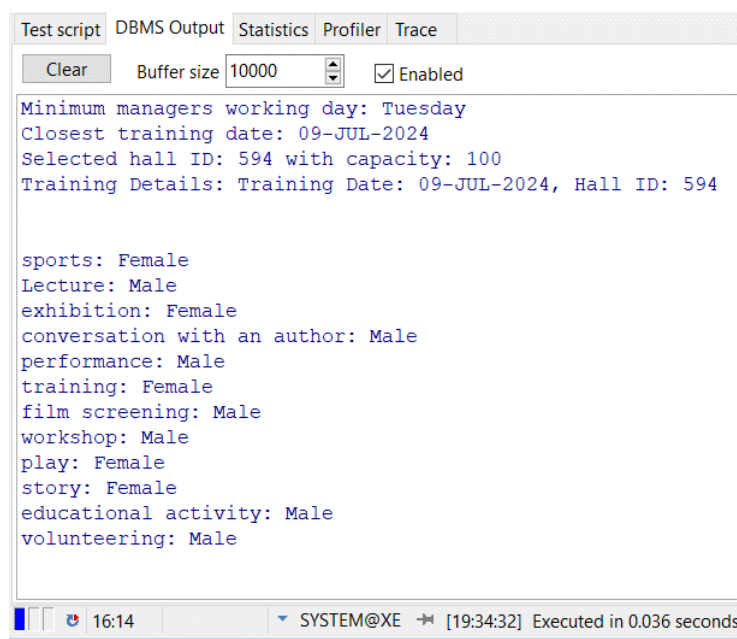
    -- Get the average number of registrants for female operators
    OPEN c_avg_registrants_women(category_rec.ECategory);
    FETCH c_avg_registrants_women INTO v_avg_registrants_women;
    CLOSE c_avg_registrants_women;

    -- Print the preferred operator type for the category
    IF v_avg_registrants_men > v_avg_registrants_women THEN
      DBMS_OUTPUT.PUT_LINE(category_rec.ECategory || ': Male');
    ELSE
      DBMS_OUTPUT.PUT_LINE(category_rec.ECategory || ':
Female');
    END IF;
  END LOOP;

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END PrintPreferredOperatorType;
```

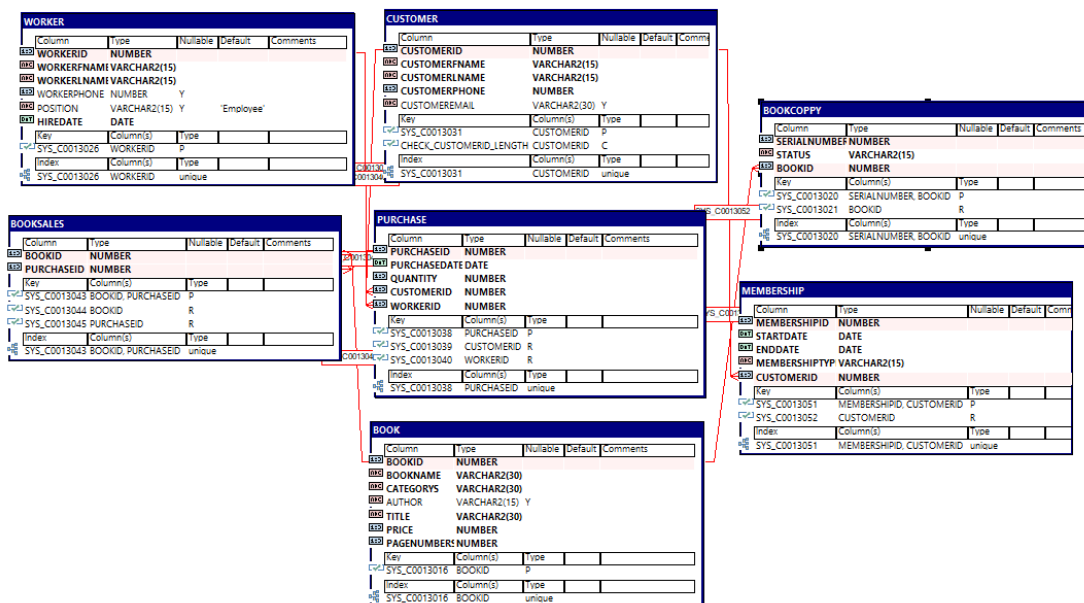
דוגמת הרצה:

ניתן לראות שהיום בשבוע שנבחר עבור ההשתלמות הוא יום שלישי, והתאריך שנקבע הוא ליום שלישי הקרוב (9.7.2024), באולם מספר 594. וכן ניתן לראות את הרשימה שמציגה עבור כל קטגוריית פעילות, איזה מגדר מפעיל מועדף יותר על הקהל.

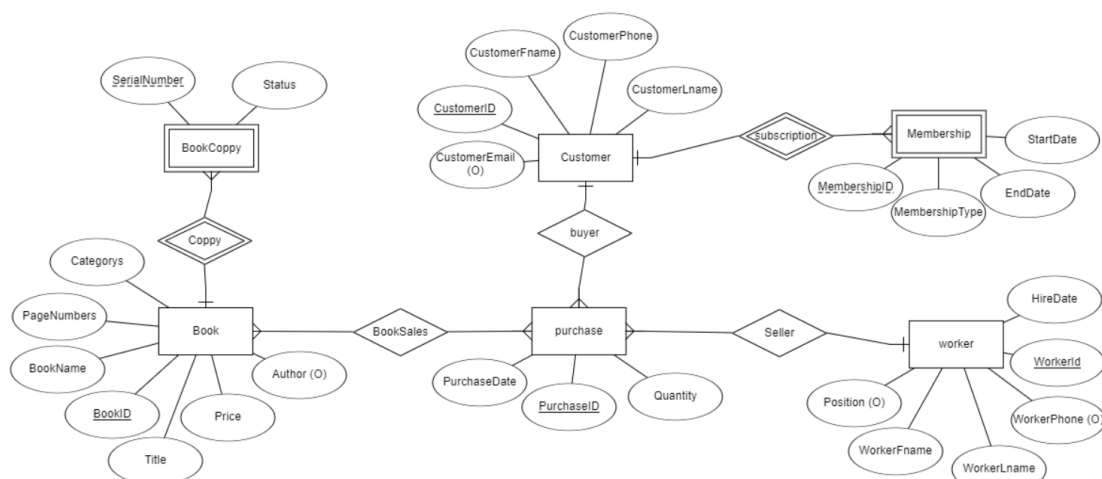


- שלב ד' -

DSD של האגף החדש:



ERD של האגף החדש:



החלטות אינטגרציה:

הטבלאות החדשות:

- נוסף לטבלאות האגף שלנו את כל הטבלאות של האגף השני. ונבצע את השינויים הבאים:
- את שם הטבלה "Worker" המקורית, נשנה ל "Librarian", משום שבאיחוד האגפים יש יותר מסוג עובד אחד, ולכן עלינו להיות ספציפיים יותר.
- ניצור טבלה אחרת בשם "Worker" שתייצג "עובד" מכל סוג שהוא.

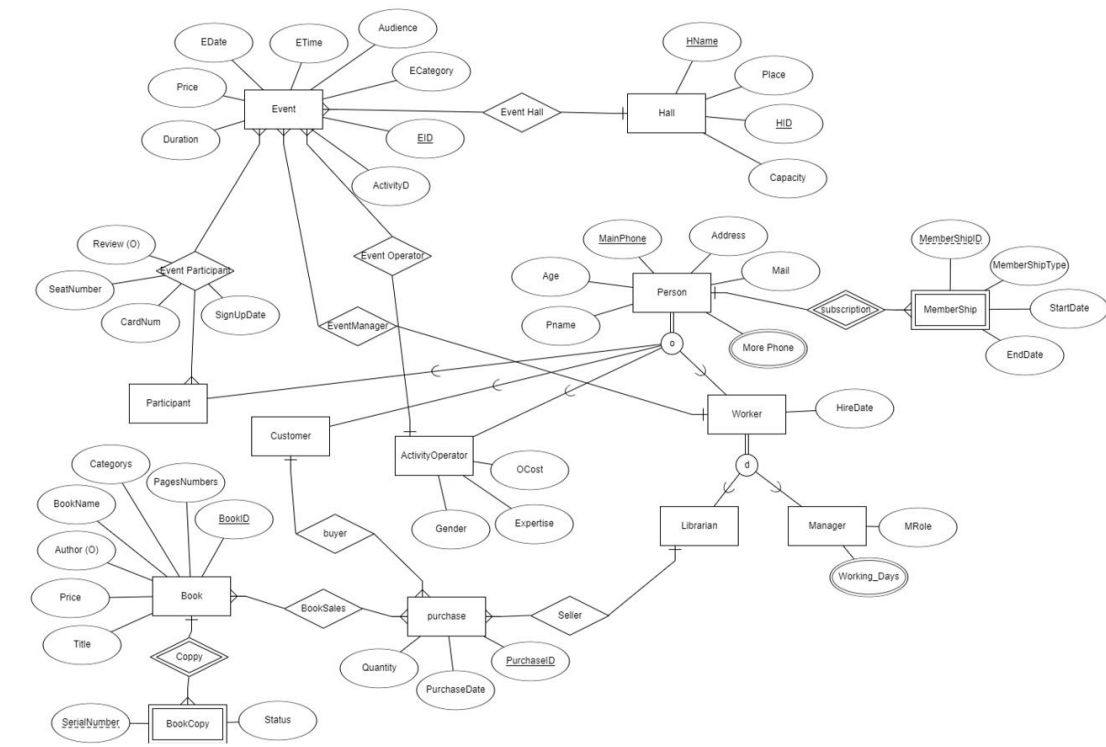
קשרים בין טבלאות:

- Manager ו Librarian ירשו מ Worker משום שאלו סוגים של עובדים.
- Worker ירש מ Person משום שמדובר באנשים.
- Customer ירש מ Person, משום שמדובר באנשים.
- נקבע את MemberShip להיות ישות חלשה מ Person

שינויים בתכונות:

- נוסף ל worker את התכונה hireDate שתחליף את ה hireDate של Librarian וה seniority של Manager (נחשב את תאריך הגיוס לפי שנות הותק בהתאם לתאריך הנוכחי).
- מכל הטבלאות שכעת יורשות מ Person נסיר תכונות של פרטים אישיים שנמצאות ב Person, ונקבע את התכונה mainPhone להיות המפתח.
- נסיר את התכונה "type" של participant שציינה האם משתתף הוא מנוי או לא. משום שיש לנו טבלה של membership שמציינת מנוי של אדם.
- בהתאם, נסיר את התכונות האופציונאליות של שם משתמש וסיסמא שהיו ב Participant.
- נסיר את התכונה position מ Librarian, משום שעכשיו יש פשוט שני סוגים של עובדים וקבענו שרק ספרן יכול לבצע מכירות.

ERD של מיוזג האגפים:



DSD של מיוזג האגפים:



הסבר מילולי של התהליך והפקודות:

מחיקת אילוצים לא רלוונטיים למבנה החדש:

```
SELECT constraint_name
FROM all_cons_columns
WHERE table_name = 'MEMBERSHIP' AND column_name='CUSTOMERID';
```

```
ALTER TABLE Membership DROP CONSTRAINT SYS_C0016027;
ALTER TABLE Membership DROP CONSTRAINT SYS_C0016028;
ALTER TABLE Membership DROP CONSTRAINT SYS_C0016029;
```

```
SELECT constraint_name
FROM all_cons_columns
WHERE table_name = 'PURCHASE' AND column_name='WORKERID';
```

```
ALTER TABLE purchase DROP CONSTRAINT SYS_C0016014;
ALTER TABLE purchase DROP CONSTRAINT SYS_C0016017;
```

```
SELECT constraint_name
FROM all_cons_columns
WHERE table_name = 'MANAGER' AND column_name='MAINPHONE';
```

```
ALTER TABLE MANAGER DROP CONSTRAINT SYS_C0015948;
ALTER TABLE MANAGER DROP CONSTRAINT SYS_C0015950;
```

שינוי שם הטבלה WORKER המקורית ל LIBRARIAN:

```
-- Rename the existing worker table to Librarian
ALTER TABLE Worker RENAME TO Librarian;
```

עדכון הטיפוס של התכונה mainPhone כך שתתאים ליותר מ 10 תווים (יחזור בכמה מקומות):

```
ALTER TABLE person MODIFY (mainphone VARCHAR2(15));
```

טיפול בטבלה LIBRARIAN:

```
--change the type of the phone number column of librarian
ALTER TABLE Librarian RENAME COLUMN workerphone TO workerphoneTemp;
```

```
ALTER TABLE Librarian ADD (workerphone VARCHAR2(15));
UPDATE Librarian SET workerphone = TO_CHAR(workerphoneTemp);
ALTER TABLE Librarian DROP COLUMN workerphoneTemp;
```

טיפול בטבלה CUSTOMER:

```
--change the type of the phone number column of customer
ALTER TABLE Customer RENAME COLUMN CustomerPhone TO
CustomerPhoneTemp;
ALTER TABLE Customer ADD (CustomerPhone VARCHAR2(15));
UPDATE Customer SET CustomerPhone = TO_CHAR(CustomerPhoneTemp);
ALTER TABLE Customer DROP COLUMN CustomerPhoneTemp;
```

WORKER יצירת הטבלה

```
-- Create the new Worker table
CREATE TABLE Worker
(
    MainPhone VARCHAR2(15) NOT NULL,
    HireDate DATE NOT NULL,
    PRIMARY KEY (MainPhone),
    FOREIGN KEY (MainPhone) REFERENCES Person(MainPhone)
```

```
);

-- Update the Librarian table
-- Remove the Position column
ALTER TABLE Librarian DROP COLUMN Position;
```

טיפול במפתחות מספרי הפלאפון:

```
UPDATE Librarian L
SET L.WorkerPhone = (
    SELECT TO_CHAR(ROWNUM + 1000000000)
    FROM DUAL
    WHERE NOT EXISTS (SELECT 1 FROM Person P WHERE P.MainPhone =
TO_CHAR(ROWNUM + 1000000000))
    AND NOT EXISTS (SELECT 1 FROM Librarian L2 WHERE L2.WorkerPhone =
TO_CHAR(ROWNUM + 1000000000))
)
WHERE L.WorkerPhone IS NULL;

DROP SEQUENCE membershipid_seq;
CREATE SEQUENCE membershipid_seq START WITH 200000000 INCREMENT BY 1;

-- Add MainPhone to Membership table
ALTER TABLE Membership ADD MainPhone VARCHAR2(15);

-- Insert participants who are subscribers into Membership with
generated values
INSERT INTO Membership (MembershipID, StartDate, EndDate,
MembershipType, CustomerID, MainPhone)
SELECT membershipid_seq.NEXTVAL,
    TO_DATE('2023-01-01', 'YYYY-MM-DD') +
TRUNC(DBMS_RANDOM.VALUE(0, 365)),
    TO_DATE('2024-01-01', 'YYYY-MM-DD') +
TRUNC(DBMS_RANDOM.VALUE(0, 365)),
    CASE ROUND(DBMS_RANDOM.VALUE(1, 3))
        WHEN 1 THEN 'month'
        WHEN 2 THEN 'year'
        WHEN 3 THEN '6-month'
    END,
    -1 as CustomerID,
    MainPhone
FROM Participant
WHERE PType = 'subscription';

-- Update Membership to reference Person
DELETE FROM Customer
WHERE ROWID IN (
    SELECT ROWID FROM (
        SELECT ROWID, ROW_NUMBER() OVER (PARTITION BY CustomerPhone
ORDER BY CustomerID) AS rn
        FROM Customer
    ) WHERE rn > 1
);

commit;

-- Update MainPhone in Membership
UPDATE Membership SET MainPhone = (SELECT TO_CHAR(CustomerPhone) FROM
Customer WHERE Customer.CustomerID = Membership.CustomerID);
```

```

INSERT INTO Person (MainPhone, Pname, Address, Mail, Age)
SELECT
    C.CustomerPhone AS MainPhone,
    C.CustomerFname || ' ' || C.CustomerLname AS Pname,
    'Default Address' AS Address, -- Placeholder value for Address
    NVL(C.CustomerEmail, C.CustomerFname || '@gmail.com') AS Mail, --
    Use CustomerEmail or generate default email
    30 AS Age -- Placeholder value for Age
FROM Customer C
WHERE NOT EXISTS (
    SELECT 1 FROM Person P WHERE P.MainPhone = C.CustomerPhone
);

commit;

-- Add foreign key constraint only if it does not exist
DECLARE
    cnt INTEGER;
BEGIN
    SELECT COUNT(*) INTO cnt
    FROM user_constraints
    WHERE table_name = 'MEMBERSHIP'
    AND constraint_type = 'R'
    AND r_constraint_name = 'FK_Membership_Person';

    IF cnt = 0 THEN
        EXECUTE IMMEDIATE 'ALTER TABLE Membership ADD CONSTRAINT
FK_Membership_Person FOREIGN KEY (MainPhone) REFERENCES
Person(MainPhone)';
    END IF;
END;

-- Drop the CustomerID column
ALTER TABLE Membership DROP COLUMN CustomerID;

-- Ensure MainPhone in Event_Participant references Person
BEGIN
    EXECUTE IMMEDIATE 'ALTER TABLE Event_Participant ADD CONSTRAINT
FK_Event_Participant_Person FOREIGN KEY (MainPhone) REFERENCES
Person(MainPhone)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -2275 THEN
            NULL; -- Constraint already exists, ignore the error
        ELSE
            RAISE;
        END IF;
END;

ALTER TABLE Participant DROP COLUMN PType;
ALTER TABLE Participant DROP COLUMN UserName;
ALTER TABLE Participant DROP COLUMN PPassword;

--treat purchase
-- Add a new column LibrarianPhone to PURCHASE table
ALTER TABLE purchase ADD (LibrarianPhone VARCHAR2(15));

```

```

ALTER TABLE Librarian ADD CONSTRAINT uq_workerPhone UNIQUE
(workerPhone);

-- Delete duplicates in Librarian table if the phone number is
already in Person table
DELETE FROM Librarian L
WHERE EXISTS (SELECT 1 FROM Person P WHERE P.MainPhone =
L.WorkerPhone);

select* from purchase;

UPDATE purchase p
SET LibrarianPhone = (
    SELECT l.workerphone
    FROM Librarian l
    WHERE l.WorkerId = p.WorkerId
);

ALTER TABLE purchase ADD CONSTRAINT fk_purchase_librarian FOREIGN KEY
(LibrarianPhone) REFERENCES Librarian(workerPhone);

-- Insert Librarian data into Person table
INSERT INTO Person (MainPhone, Pname, Address, Mail, Age)
SELECT
    L.WorkerPhone AS MainPhone,
    L.WorkerFname || ' ' || L.WorkerLname AS Pname,
    'Default Address' AS Address, -- Placeholder value
    'defaultemail@example.com' AS Mail, -- Placeholder value
    30 AS Age -- Placeholder value
FROM Librarian L
WHERE L.WorkerPhone IS NOT NULL
AND NOT EXISTS (SELECT 1 FROM Person P WHERE P.MainPhone =
L.WorkerPhone);

-- Insert Librarian data into Worker table
INSERT INTO Worker (MainPhone, HireDate)
SELECT
    L.WorkerPhone AS MainPhone,
    L.HireDate AS HireDate
FROM Librarian L
WHERE L.WorkerPhone IS NOT NULL;

-- Insert Manager data into Worker table
INSERT INTO Worker (MainPhone, HireDate)
SELECT
    M.MainPhone AS MainPhone,
    ADD_MONTHS(SYSDATE, -12 * M.Seniority) AS HireDate
FROM Manager M;

-- Delete the Seniority column from Manager table
ALTER TABLE Manager DROP COLUMN Seniority;

```

```

ALTER TABLE Librarian ADD CONSTRAINT FK_Librarian_Worker FOREIGN KEY
(workerPhone) REFERENCES Worker(MainPhone);

-- Add a new column CustomerPhone to PURCHASE table
ALTER TABLE purchase ADD (CustomerPhone VARCHAR2(15));

ALTER TABLE customer RENAME COLUMN CustomerPhone TO MainPhone;

-- Update CustomerPhone with corresponding MainPhone from Customer
table
UPDATE purchase p
SET CustomerPhone = (
    SELECT c.MainPhone
    FROM Customer c
    WHERE c.CustomerID = p.CustomerID
);

-- Rename CustomerPhone to MainPhone
ALTER TABLE Customer ADD CONSTRAINT uq_customerPhone UNIQUE
(MainPhone);

-- Add foreign key constraint on the new MainPhone column referencing
the Person table
ALTER TABLE purchase ADD CONSTRAINT fk_purchase_customer FOREIGN KEY
(CustomerPhone) REFERENCES Customer(MainPhone);

ALTER TABLE librarian RENAME COLUMN workerPhone TO MainPhone;

-- Update LibrarianPhone with corresponding MainPhone from Librarian
table
UPDATE purchase p
SET LibrarianPhone = (
    SELECT l.MainPhone
    FROM Librarian l
    WHERE l.WorkerId = p.WorkerId
);

```

הוספת אילוצים של מפתחות זרים:

```

-- Add foreign key constraint to Manager table
ALTER TABLE Manager ADD CONSTRAINT fk_manager_worker FOREIGN KEY
(MainPhone) REFERENCES Worker(MainPhone);
ALTER TABLE Customer ADD CONSTRAINT fk_customer_person FOREIGN KEY
(MainPhone) REFERENCES Person(MainPhone);

-- Drop all columns except MainPhone from the CUSTOMER table

```

מחיקת עמודות לא נדרשות:

```

ALTER TABLE Customer DROP COLUMN CustomerFname;
ALTER TABLE Customer DROP COLUMN CustomerLname;
ALTER TABLE Customer DROP COLUMN CustomerEmail;
ALTER TABLE purchase DROP COLUMN WorkerId;
ALTER TABLE purchase DROP COLUMN CustomerID;
ALTER TABLE Customer DROP COLUMN CustomerID;
ALTER TABLE Librarian DROP COLUMN HireDate;
ALTER TABLE Librarian DROP COLUMN WorkerLname;
ALTER TABLE Librarian DROP COLUMN WorkerFname;
ALTER TABLE Librarian DROP COLUMN WorkerId;

```

מבטים

• על האגף שלנו:

מבט על האולמות שמכיל פרטים על אולמות האירועים בספרייה-

```
--create view
CREATE OR REPLACE VIEW Hall_View AS
SELECT
    HID,
    HName,
    Place,
    Capacity
FROM Hall;
```

הרצה:

4:1	SYSTEM@XE	[02:34:12] Done in 0.007 seconds
-----	-----------	----------------------------------

שליפת הנתונים (select*):

	HID	HNAME	PLACE	CAPACITY
1	1	Sinai	floor: 6 room: 6078 ...	500
2	2	Carmel	floor: 4 room: 4056 ...	30
3	3	Eilat	floor: 4 room: 4056 ...	250
4	4	Negev	floor: 3 room: 3045 ...	100
5	5	Sharon	floor: 0 room: 0001 ...	100
6	6	Negev	floor: 6 room: 6078 ...	500
7	7	Galilee	floor: 6 room: 6078 ...	80
8	8	Sinai	floor: 0 room: 0001 ...	30
9	9	Negev	floor: 5 room: 5067 ...	150
10	10	Carmel	floor: 6 room: 6078 ...	80
11	11	Eilat	floor: 0 room: 0001 ...	300

שאלת א' – מחזירה את הפרטים על אולמות שהתכולה שלהם גבוהה מ 100 איש:

```
--query1
SELECT
    HID,
    HName,
    Place,
    Capacity
FROM Hall_View
WHERE Capacity > 100;
```

הרצה:

	HID	HNAME	PLACE	CAPACITY
1	1	Sinai	floor: 6 room: 6078 ...	500
2	3	Eilat	floor: 4 room: 4056 ...	250

שאלת ב' – מחזירה את הפרטים על כל האולמות שממוקמים בקומה רביעית ומעלה:

```
--query2
SELECT
    HID,
```

```
HName,
Place,
Capacity
FROM Hall_View
WHERE
    TO_NUMBER(SUBSTR(Place, INSTR(Place, ':') + 2, 1)) BETWEEN 4 AND 6;
```

הרצה:

	HID	HNAME	PLACE	CAPACITY
▶ 1	1	Sinai	floor: 6 room: 6078 ...	500
2	2	Carmel	floor: 4 room: 4056 ...	30

• על האגף החדש:

מבט על ישויות הספרים בספריה-

```
--create view
CREATE OR REPLACE VIEW Book_View AS
SELECT
    BookID,
    BookName,
    Categorys,
    Author,
    Title,
    Price,
    PageNumbers
FROM Book;
```

הרצה:

38:1	SYSTEM@XE	[02:43:09] Done in 0.007 seconds
------	-----------	----------------------------------

שליפת הנתונים (select*):

	BOOKID	BOOKNAME	CATEGORYS	AUTHOR	TITLE	PRICE	PAGENUMBERS
1	112459	Gone with the Wind	Fantasy	J.R.R. Tolkien	Science Fiction	66	54
2	112463	The Girl on the Train	Self-Help	Jane Austen	Young Adult	854	501
3	112467	The Handmaid's Tale	Cookbook	Dan Brown	Fantasy	904	220
4	112471	The Handmaid's Tale	Historical Fiction	J.R.R. Tolkien	Thriller	87	622
5	112475	The Help	Business	George Orwell	Fantasy	43	2
6	112479	Educated	Biography	J.K. Rowling	Travel	630	53
7	112483	One Hundred Years of Solitude	Science Fiction	William Blake	Fantasy	115	93
8	112487	Sapiens	Self-Help	J.R.R. Tolkien	Business	292	5
9	112491	The Catcher in the Rye	Travel	Agatha Christie	Mystery	10	98
10	112495	The Sun Also Rises	Biography	Dan Brown	Biography	47	2

שאלתא א' – שאלתא שמחזירה את הספרים שמתחילים באות 'ד':

```
SELECT
    BookID,
    BookName,
    Author,
    Title,
    Price
FROM Book
WHERE BookName LIKE 'T%';
```

הרצה:

	BOOKID	BOOKNAME	AUTHOR	TITLE	PRICE
1	112463	The Girl on the Train	Jane Austen	Young Adult	854
2	112467	The Handmaid's Tale	Dan Brown	Fantasy	904

52:1 SYSTEM@XE [02:46:20] 195 rows selected in 0.061 seconds

שאלתא ב' – מחזירה מידע על הספרים שמחירם בין 10 ל 20 ₪, ומציגה אותם בסדר מחיר יורד.

```
SELECT
    BookID,
    BookName,
    Author,
    Title,
    Price
FROM Book_View
WHERE Price BETWEEN 10 AND 20
ORDER BY Price DESC;
```

הרצה:

	BOOKID	BOOKNAME	AUTHOR	TITLE	PRICE
1	111439	The Help	Charles Dickens	Memoir	20
2	111935	Moby-Dick	Margaret Atwood	Fantasy	20

65:1 SYSTEM@XE [02:47:09] 32 rows selected in 0.035 seconds

