דוח הפרוייקט שלב א

פרויקט על ארגון בתחום הפנאי פנאי - קולנוע

מגישות:

רחלי וינר – 325265221, viderracheli@gmail.com

נעמה מייפן – 214429193, naamameypen@gmail.com

הסבר:

מטרת המערכת:

מערכת ניהול הקולנוע נועדה לטפל בהיבטים התפעוליים של בית קולנוע. זה מקל על ניהול סרטים, זמני הצגה, קולנוע, מנויים ומכירת כרטיסים. המערכת שמה לה למטרה לייעל את תפעול הקולנוע, לשפר את חווית הלקוח ולייעל את ניצול המשאבים. על ידי שמירה על רישומים מפורטים ומאפשרת תזמון יעיל, המערכת מבטיחה תפקוד חלק ומשפרת את היעילות הכוללת של פעולות הקולנוע.

1. סרטים: מזהה סרט, כותרת, ז'אנר, במאי, תאריך יציאה, משך, דירוג.

טבלה זו מאחסנת מידע על הסרטים המוצגים בקולנוע, כולל תעודת הזהות, הכותרת, הז'אנר, הבמאי, תאריך היציאה, משך הזמן והדירוג שלהם.

2. **תאריכים**: DateID, HourH, DayD, DateD.

טבלה זו עוקבת אחר התאריכים והשעות שבהם הסרטים מתוכננים להיות מוצגים.

3. **אולמות**: TheaterID, TheaterName, Capacity.

טבלה זו מכילה מידע על בתי הקולנוע בתוך הקולנוע, כולל תעודת הזהות, השם ויכולת הישיבה של כל אולם.

4. **לוחות זמנים**: ScheduleID, TheaterID, AvailableSeats, MovieID, DateID.

טבלה זו מקשרת בין סרטים, תאריכים ובתי קולנוע, ומציינת איזה סרט מוצג באיזה קולנוע באיזו שעה וכמה מושבים זמינים לכל הצגה.

5. **מנויים**: מזהה מנוי, אחוזי הנחה, תאריך תפוגה.

טבלה זו מאחסנת מידע על המנויים, כולל תעודת הזהות שלהם, אחוז ההנחה ותאריך תפוגה של המנוי.

6. **מכירות כרטיסים**: TicketID, Price, IsSold, ScheduleID, SubscriberID.

טבלה זו מתעדת את מכירת הכרטיסים, כולל מזהה כל כרטיס, מחיר, סטטוס מכירה והקשרים עם לוחות זמנים ומנויים ספציפיים.

פונקציות עיקריות של המערכת:

1. **ניהול סרטים:**

- הוספה, עדכון והסרה של סרטים.

- מעקב אחר פרטי סרט כגון ז'אנר, במאי, תאריך יציאה, משך ודירוג.

2. **תזמון מועדי הופעה**:

- הגדרת זמני הקרנה לסרטים.

- הקצאת בתי קולנוע לסרטים שונים.

- ניהול המושבים הפנויים לכל שעת הופעה.

3. **ניהול אולמות**:

- שמירה על מידע על כל אולם בתוך הקולנוע.

- ניהול קיבולת וזמינות האולם.

4. **ניהול מנויים**:

- ניהול מידע מנויים.

- החלת הנחות למנויים.

- מעקב אחר תאריכי תפוגה של מנוי.

5. **מכירת כרטיסים**:

- מכירת כרטיסים למועדי הופעה קבועים.

- מעקב אחר מכירת כרטיסים וזמינות.

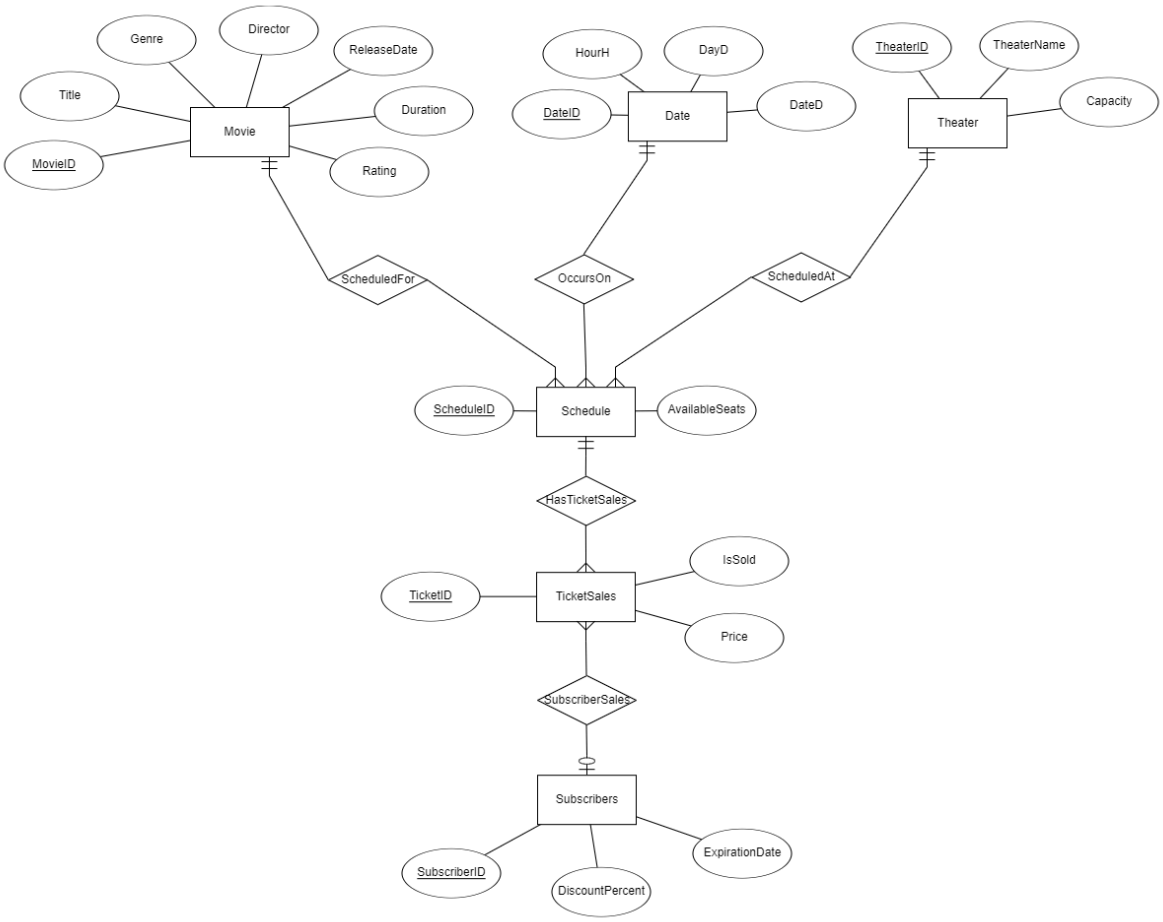
- החלת הנחות מנויים על מכירת כרטיסים.

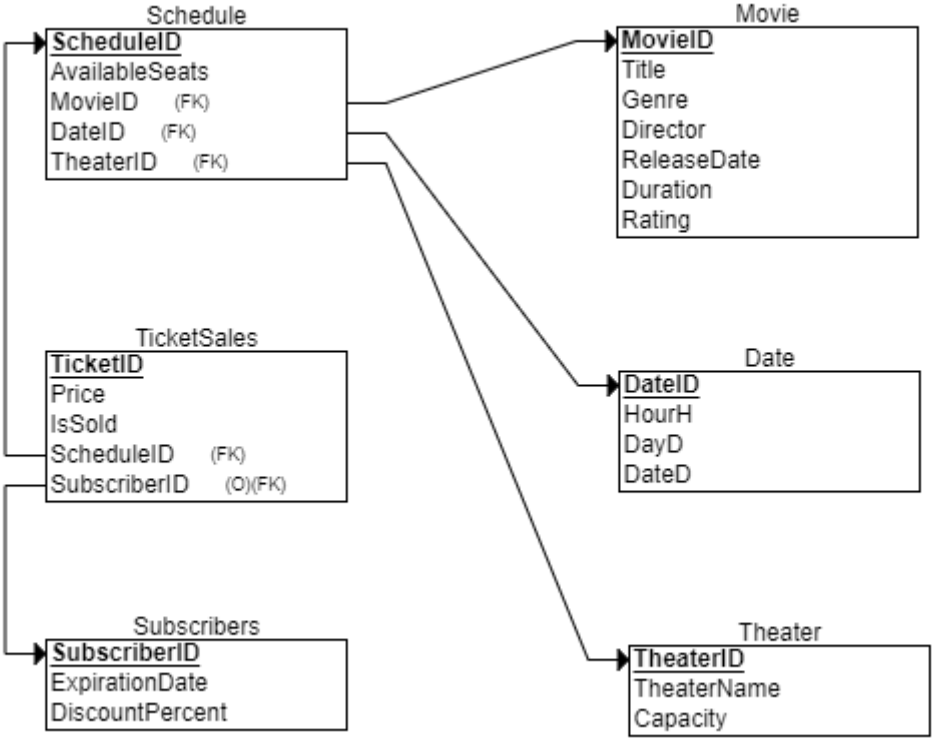
6. **ניהול תאריך וזמן**:

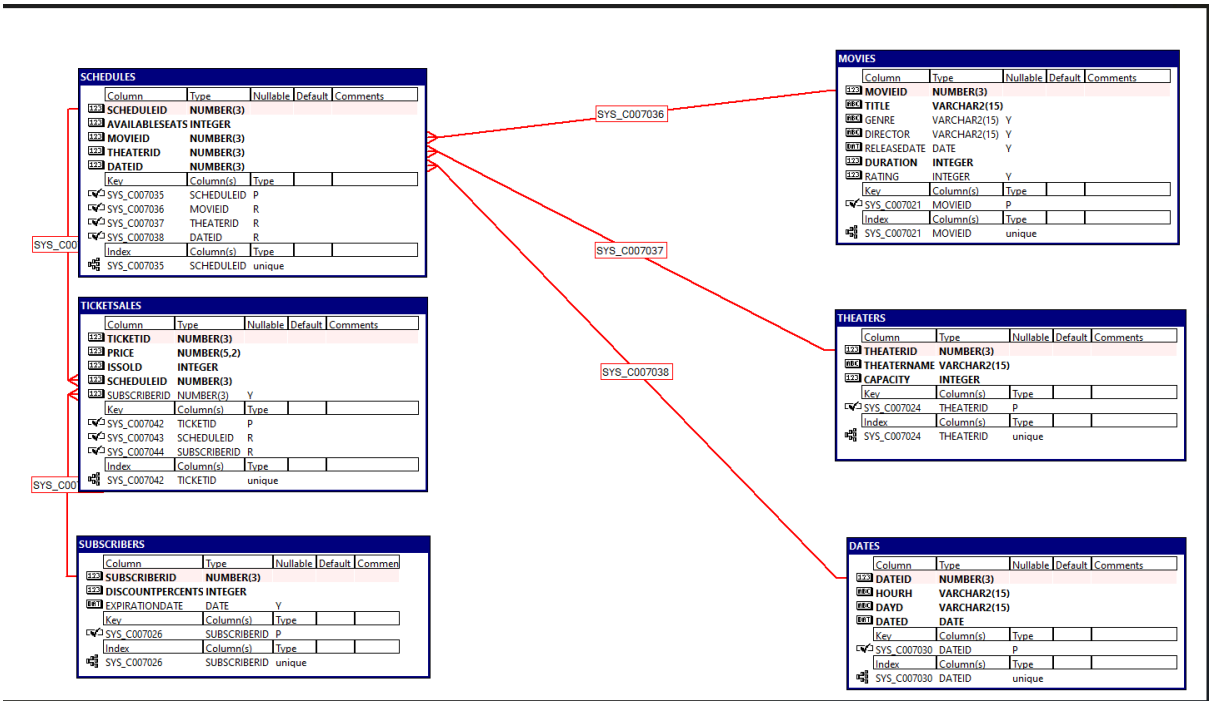
- רישום של כל התאריכים והשעות שבהם מוצגים סרטים.

- הבטחת תזמון מדויק והימנעות מעימותים.

תרשימי ה ERD ו ה DSD:


Genre
Director
ReleaseDate
Title
Movie
Duration
MovieID
Rating
HourH
DayD
DateID
Date
DateD
TheaterID
TheaterName
Theater
Capacity
ScheduledFor
OccursOn
ScheduledAt
ScheduleID
Schedule
AvailableSeats
HasTicketSales
IsSold
TicketID
TicketSales
Price
SubscriberSales
Subscribers
ExpirationDate
SubscriberID
DiscountPercent


Schedule
ScheduleID
AvailableSeats
MovieID (FK)
DateID (FK)
TheaterID (FK)
Movie
MovieID
Title
Genre
Director
ReleaseDate
Duration
Rating
TicketSales
TicketID
Price
IsSold
ScheduleID (FK)
SubscriberID (O)(FK)
Date
DateID
HourH
DayD
DateD
Subscribers
SubscriberID
ExpirationDate
DiscountPercent
Theater
TheaterID
TheaterName
Capacity


SCHEDULES
Column Type Nullable Default Comments
SCHEDULEID NUMBER(3)
AVAILABLESEATS INTEGER
MOVIEID NUMBER(3)
THEATERID NUMBER(3)
DATEID NUMBER(3)
Key Column(s) Type
SYS_C007035 SCHEDULEID P
SYS_C007036 MOVIEID R
SYS_C007037 THEATERID R
SYS_C007038 DATEID R
Index Column(s) Type
SYS_C007035 SCHEDULEID unique
TICKETSALES
Column Type Nullable Default Comments
TICKETID NUMBER(3)
PRICE NUMBER(5,2)
ISSOLD INTEGER
SCHEDULEID NUMBER(3)
SUBSCRIBERID NUMBER(3) Y
Key Column(s) Type
SYS_C007042 TICKETID P
SYS_C007043 SCHEDULEID R
SYS_C007044 SUBSCRIBERID R
Index Column(s) Type
SYS_C007042 TICKETID unique
SUBSCRIBERS
Column Type Nullable Default Commen
SUBSCRIBERID NUMBER(3)
DISCOUNTPERCENTS INTEGER
EXPIRATIONDATE DATE Y
Key Column(s) Type
SYS_C007026 SUBSCRIBERID P
Index Column(s) Type
SYS_C007026 SUBSCRIBERID unique
SYS_C007036
SYS_C007037
SYS_C007038
MOVIES
Column Type Nullable Default Comments
MOVIEID NUMBER(3)
TITLE VARCHAR2(15)
GENRE VARCHAR2(15) Y
DIRECTOR VARCHAR2(15) Y
RELEASEDATE DATE Y
DURATION INTEGER
RATING INTEGER Y
Key Column(s) Type
SYS_C007021 MOVIEID P
Index Column(s) Type
SYS_C007021 MOVIEID unique
THEATERS
Column Type Nullable Default Comments
THEATERID NUMBER(3)
THEATERNAME VARCHAR2(15)
CAPACITY INTEGER
Key Column(s) Type
SYS_C007024 THEATERID P
Index Column(s) Type
SYS_C007024 THEATERID unique
DATES
Column Type Nullable Default Comments
DATEID NUMBER(3)
HOURH VARCHAR2(15)
DAYD VARCHAR2(15)
DATED DATE
Key Column(s) Type
SYS_C007030 DATEID P
Index Column(s) Type
SYS_C007030 DATEID unique

הכנסת נתונים לטבלאות ב3 שיטות:

1. Data generator: (TicketSales, Schedules)

TICKETSALES

Owner: RACHELI | Table: TICKETSALES | Number of records: 400

Name	Type	Size	Data	Master
TICKETID	NUMBER	3	Sequence(100)	
PRICE	NUMBER	5,2	List(30, 30, 30, 30, 50, 90)	
ISSOLD	CHAR	1	List(N)	
SCHEDULEID	NUMBER	3	List(select scheduleid from schedules)	
SUBSCRIBERID	NUMBER	3	List(select subscriberid from subscribers)	

SCHEDULES

Owner: RACHELI | Table: SCHEDULES | Number of records: 400

Name	Type	Size	Data
SCHEDULEID	NUMBER	3	Sequence(100)
AVAILABLESEATS	NUMBER		List(0)
MOVIEID	NUMBER	3	List(select MOVIEID from MOVIES)
THEATERID	NUMBER	3	List(select THEATERID from THEATERS)
DATEID	NUMBER	3	List(select DATEID from DATES)

```
insert into RACHELI.SCHEDULES (SCHEDULEID, AVAILABLESEATS, MOVIEID, THEATERID, DATEID)
values (498, 0, 424, 297, 190);

insert into RACHELI.SCHEDULES (SCHEDULEID, AVAILABLESEATS, MOVIEID, THEATERID, DATEID)
values (499, 0, 120, 428, 228);

UPDATE Schedules s
SET s.AvailableSeats = (
    SELECT t.Capacity
    FROM Theaters t
    WHERE t.TheaterID = s.TheaterID
);

commit;
```

לאחר הכנסת מקומות הישיבה כ 0 בכל הטבלאות, הוספנו בסוף הקובץ עדכון לערך תכולה, שיתאים לאולם שנבחר על פי ה TheatherID ניקח את התכולה של כל שורה בטבלה Theaters.

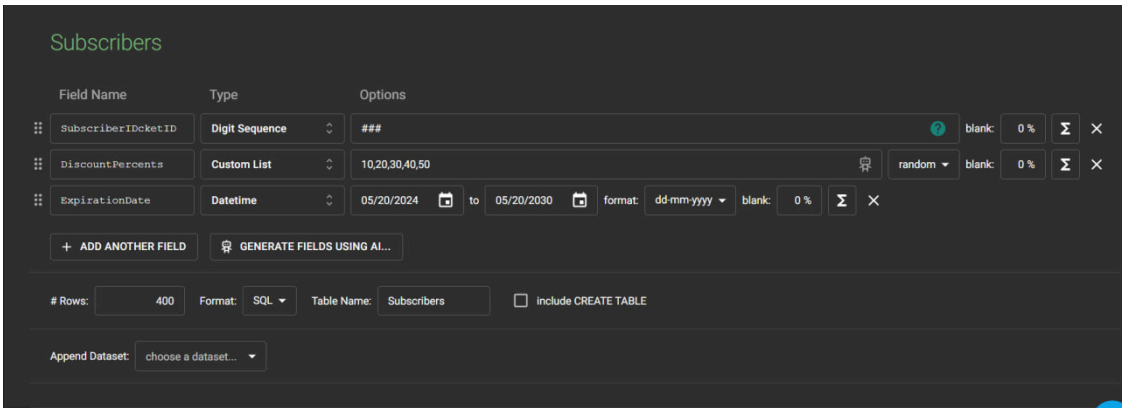
2. Python Script: (Theaters, Dates)

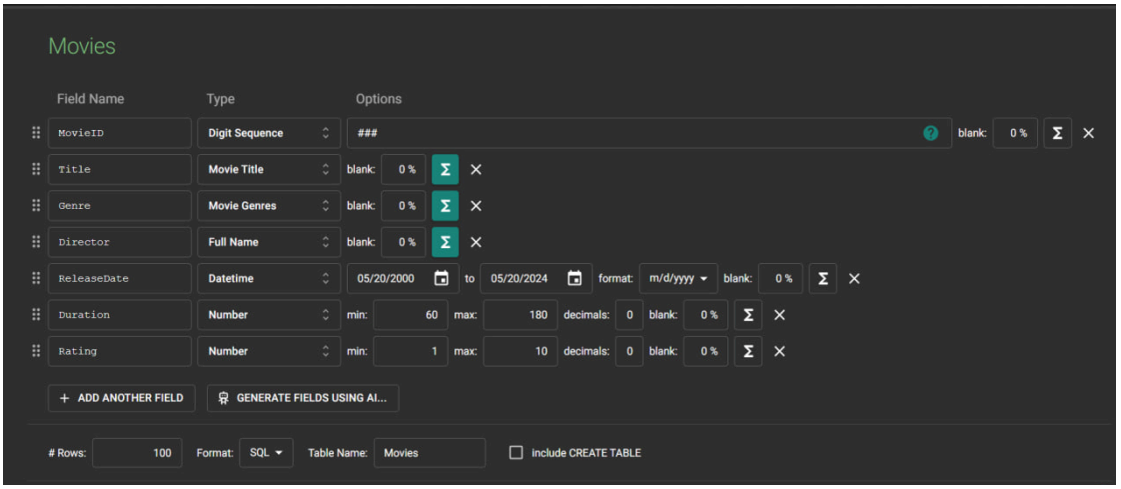
```python
def read_flower_names(filename):
    with open(filename, 'r') as file:
        flowers = [line.strip() for line in file.readlines()]
    return flowers

def generate_plsql_script(flowers, num_records, output_filename):
    theater_id_start = 111
    min_capacity = 20
    max_capacity = 300

    with open(output_filename, 'w') as file:
        file.write("BEGIN\n")
        for i in range(num_records):
            theater_id = theater_id_start + i
            theater_name = flowers[i % len(flowers)]
            capacity = random.randint(min_capacity, max_capacity)
            file.write(f"    INSERT INTO THEATERS (THEATERID, THEATERNAME, CAPACITY) VALUES ({theater_id}, '{theater_name}', {capacity});\n")
        file.write("END;\n")
        file.write("/\n")

if __name__ == "__main__":
    flowers = read_flower_names('flower_names.txt')
    num_records = 400
    output_filename = 'insert_theaters.sql'
    generate_plsql_script(flowers, num_records, output_filename)
    print(f"PL/SQL script '{output_filename}' generated successfully.")
```

```python
from datetime import datetime, timedelta

# Function to generate sequential dates
def generate_sequential_dates(start_date, num_days):
    return [start_date + timedelta(days=i) for i in range(num_days)]

# Define hours and days
hours = ["14:00", "17:00", "20:00", "23:00"]

# Start date for the records
start_date = datetime(2024, 6, 1)
num_days = 120  # Number of days to generate
total_records = 400  # Total number of unique records to generate

# Generate sequential dates
sequential_dates = generate_sequential_dates(start_date, num_days)

# Calculate how many records per day are needed
records_per_day = total_records // num_days

# Generate 400 unique records
records = []
ur = set()
date_id = 100  # Starting DateID

for day_date in sequential_dates:
    day = day_date.strftime('%a').upper()
    date_d = day_date.strftime('%d-%m-%Y')
    hour_count = 0

```

```python
    for hour in hours:
        if len(records) >= total_records:
            break
        # Use a tuple to ensure uniqueness
        record = (date_id, hour, day, date_d)
        unique_key = (hour, day, date_d)

        # Add record to the set if the unique combination does not exist
        if unique_key not in ur:
            ur.add(unique_key)
            records.append(record)
            date_id += 1
            hour_count += 1

# Write the insert statements to a SQL file
with open('Date_insertion.sql', 'w') as file:
    for record in records:
        date_id, hour, day, date_d = record
        insert_statement = f"INSERT INTO DATES (DATEID, HOURH, DAYD, DATED) VALUES ({date_id}, '{hour}', '{day}', TO_DATE('{date_d}', 'DD-MM-YYYY'));"
        file.write(insert_statement + '\n')
```

3. mockaroo: (Subscribers, Movies)





עבור שדה מטיפוס תאריך זה הקוד שכתבנו:

```
code(concat(code("to_date('"),this, code("', 'dd-mm-yyyy')")))
```

הצגת Create tables ופקודות Desc:

1. Movies:

```
CREATE TABLE Movies (
    MovieID NUMERIC(3) PRIMARY KEY,
    Title VARCHAR2(100) NOT NULL,
    Genre VARCHAR2(100),
    Director VARCHAR2(100),
    ReleaseDate DATE,
    Duration INT NOT NULL,
    Rating INT
);
```

```
SQL> desc Movies;
Name         Type           Nullable Default Comments
----------- ------------- -------- ------- --------
MOVIEID      NUMBER(3)
TITLE        VARCHAR2(100)
GENRE        VARCHAR2(100) Y
DIRECTOR     VARCHAR2(100) Y
RELEASEDATE  DATE          Y
DURATION     INTEGER
RATING       INTEGER       Y
```

:Dates .2

```sql
CREATE TABLE Dates (
    DateID NUMERIC(3) PRIMARY KEY,
    HourH VARCHAR2(100) NOT NULL,
    DayD VARCHAR2(100) NOT NULL,
    DateD DATE NOT NULL
);
```

```
SQL> desc Schedules
Name            Type       Nullable Default Comments
-------------- --------- -------- ------- --------
SCHEDULEID      NUMBER(3)
THEATERID       NUMBER(3)
AVAILABLESEATS  INTEGER
MOVIEID         NUMBER(3)
DATEID          NUMBER(3)
```

:Theaters .3

```sql
CREATE TABLE Theaters (
    TheaterID NUMERIC(3) PRIMARY KEY,
    TheaterName VARCHAR2(30) NOT NULL,
    Capacity INT NOT NULL
);
```

```
SQL> desc theaters
Name        Type          Nullable Default Comments
----------- ------------- -------- ------- --------
THEATERID   NUMBER(3)
THEATERNAME VARCHAR2(100)
CAPACITY    INTEGER
```

:Schedules .4

```
CREATE TABLE Schedules (
    ScheduleID NUMERIC(3) PRIMARY KEY,
    AvailableSeats INT NOT NULL,
    MovieID NUMERIC(3) NOT NULL,
    TheaterID NUMERIC(3) NOT NULL,
    DateID NUMERIC(3) NOT NULL,
    FOREIGN KEY (MovieID) REFERENCES Movies(MovieID),
    FOREIGN KEY (TheaterID) REFERENCES Theaters(TheaterID),
    FOREIGN KEY (DateID) REFERENCES Dates(DateID)
);
```

```
SQL> desc Schedules
Name            Type      Nullable Default Comments
--------------- --------- -------- ------- --------
SCHEDULEID      NUMBER(3)
THEATERID       NUMBER(3)
AVAILABLESEATS  INTEGER
MOVIEID         NUMBER(3)
DATEID          NUMBER(3)
```

:Subscribers .5

```
CREATE TABLE Subscribers (
    SubscriberID NUMERIC(3) PRIMARY KEY,
    DiscountPercents INT NOT NULL,
    ExpirationDate DATE
);
```

```
SQL> desc Subscribers
Name             Type      Nullable Default Comments
---------------- --------- -------- ------- --------
SUBSCRIBERID     NUMBER(3)
DISCOUNTPERCENTS INTEGER
EXPIRATIONDATE   DATE      Y
```

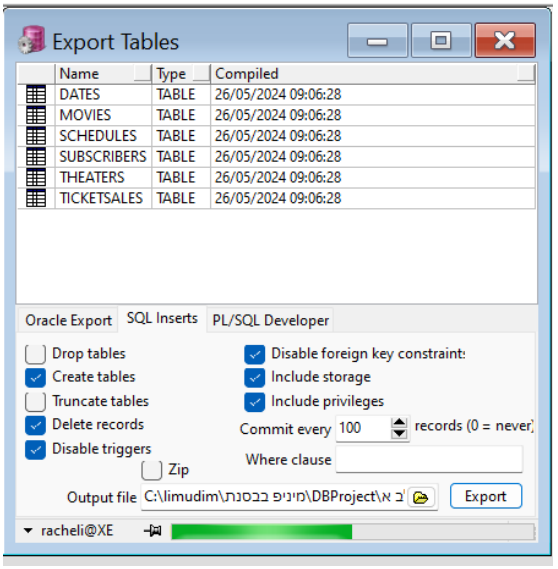
:TicketSales .6

```
CREATE TABLE TicketSales (
    TicketID NUMERIC(3) PRIMARY KEY,
    Price NUMERIC(5, 2) NOT NULL,
    IsSold CHAR(1) DEFAULT 'N' CHECK (IsSold IN ('Y', 'N')) NOT NULL,
    ScheduleID NUMERIC(3) NOT NULL,
    SubscriberID NUMERIC(3),
    FOREIGN KEY (ScheduleID) REFERENCES Schedules(ScheduleID),
    FOREIGN KEY (SubscriberID) REFERENCES Subscribers(SubscriberID)
);
```

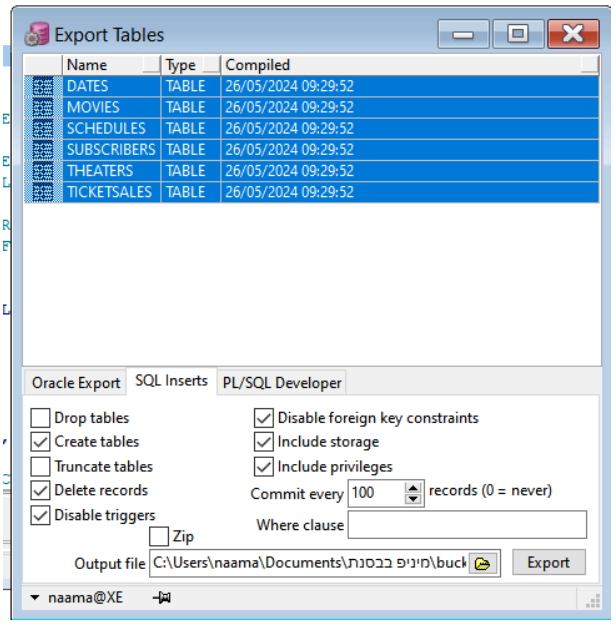
```
SQL> desc TicketSales
Name          Type         Nullable Default Comments
------------- ------------ -------- ------- --------
TICKETID      NUMBER(3)
PRICE         NUMBER(5,2)
ISSOLD        CHAR(1)               'N'
SCHEDULEID    NUMBER(3)
SUBSCRIBERID  NUMBER(3)    Y
```

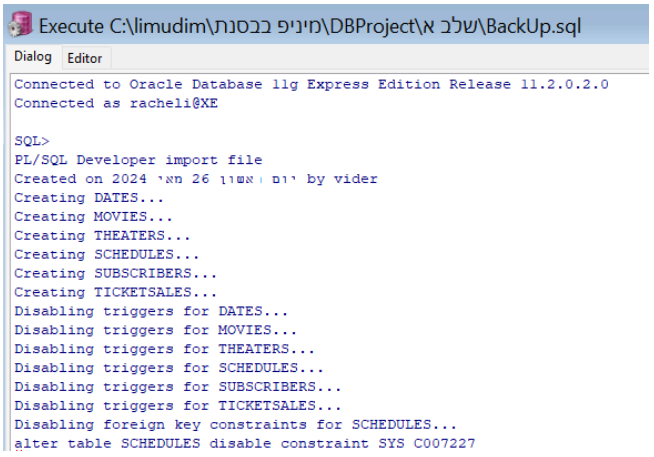
גיבוי ושחזור:

גיבוי:


Export Tables
Name Type Compiled
DATES TABLE 26/05/2024 09:06:28
MOVIES TABLE 26/05/2024 09:06:28
SCHEDULES TABLE 26/05/2024 09:06:28
SUBSCRIBERS TABLE 26/05/2024 09:06:28
THEATERS TABLE 26/05/2024 09:06:28
TICKETSALES TABLE 26/05/2024 09:06:28
Oracle Export
SQL Inserts
PL/SQL Developer
Drop tables
Create tables
Truncate tables
Delete records
Disable triggers
Zip
Disable foreign key constraints
Include storage
Include privileges
Commit every 100 records (0 = never)
Where clause
Output file C:\limudim\מיניפ בבסנת\DBProject\ב א
Export
racheli@XE

שחזור:


Export Tables
Name Type Compiled
DATES TABLE 26/05/2024 09:29:52
MOVIES TABLE 26/05/2024 09:29:52
SCHEDULES TABLE 26/05/2024 09:29:52
SUBSCRIBERS TABLE 26/05/2024 09:29:52
THEATERS TABLE 26/05/2024 09:29:52
TICKETSALES TABLE 26/05/2024 09:29:52
Oracle Export
SQL Inserts
PL/SQL Developer
Drop tables
Create tables
Truncate tables
Delete records
Disable triggers
Zip
Disable foreign key constraints
Include storage
Include privileges
Commit every 100 records (0 = never)
Where clause
Output file C:\Users\naama\Documents\מיניפ בבסנת\buck
Export
naama@XE


Execute C:\limudim\מיניפ בבסנת\DBProject\שלב א\BackUp.sql
Dialog
Editor
Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
Connected as racheli@XE
SQL>
PL/SQL Developer import file
Created on 2024 מאי 26 יום שישי by vider
Creating DATES...
Creating MOVIES...
Creating THEATERS...
Creating SCHEDULES...
Creating SUBSCRIBERS...
Creating TICKETSALES...
Disabling triggers for DATES...
Disabling triggers for MOVIES...
Disabling triggers for THEATERS...
Disabling triggers for SCHEDULES...
Disabling triggers for SUBSCRIBERS...
Disabling triggers for TICKETSALES...
Disabling foreign key constraints for SCHEDULES...
alter table SCHEDULES disable constraint SYS_C007227

Execute C:\limudim\מיניפ בבסנת\DBProject\שלב א\BackUp.sql

Dialog Editor

```
100 records committed...
200 records committed...
300 records committed...
400 records loaded
Loading MOVIES...
100 records committed...
200 records committed...
300 records committed...
400 records committed...
401 records loaded
Loading THEATERS...
100 records committed...
200 records committed...
300 records committed...
400 records loaded
Loading SCHEDULES...
100 records committed...
200 records committed...
300 records committed...
400 records loaded
Loading SUBSCRIBERS...
100 records committed...
200 records committed...
300 records committed...
400 records committed...
401 records loaded
Loading TICKETSALES...
100 records committed...
200 records committed...
300 records committed...
400 records loaded
Enabling foreign key constraints for SCHEDULES
```