

תרגיל בית 1 בקורס מסדי נתונים:

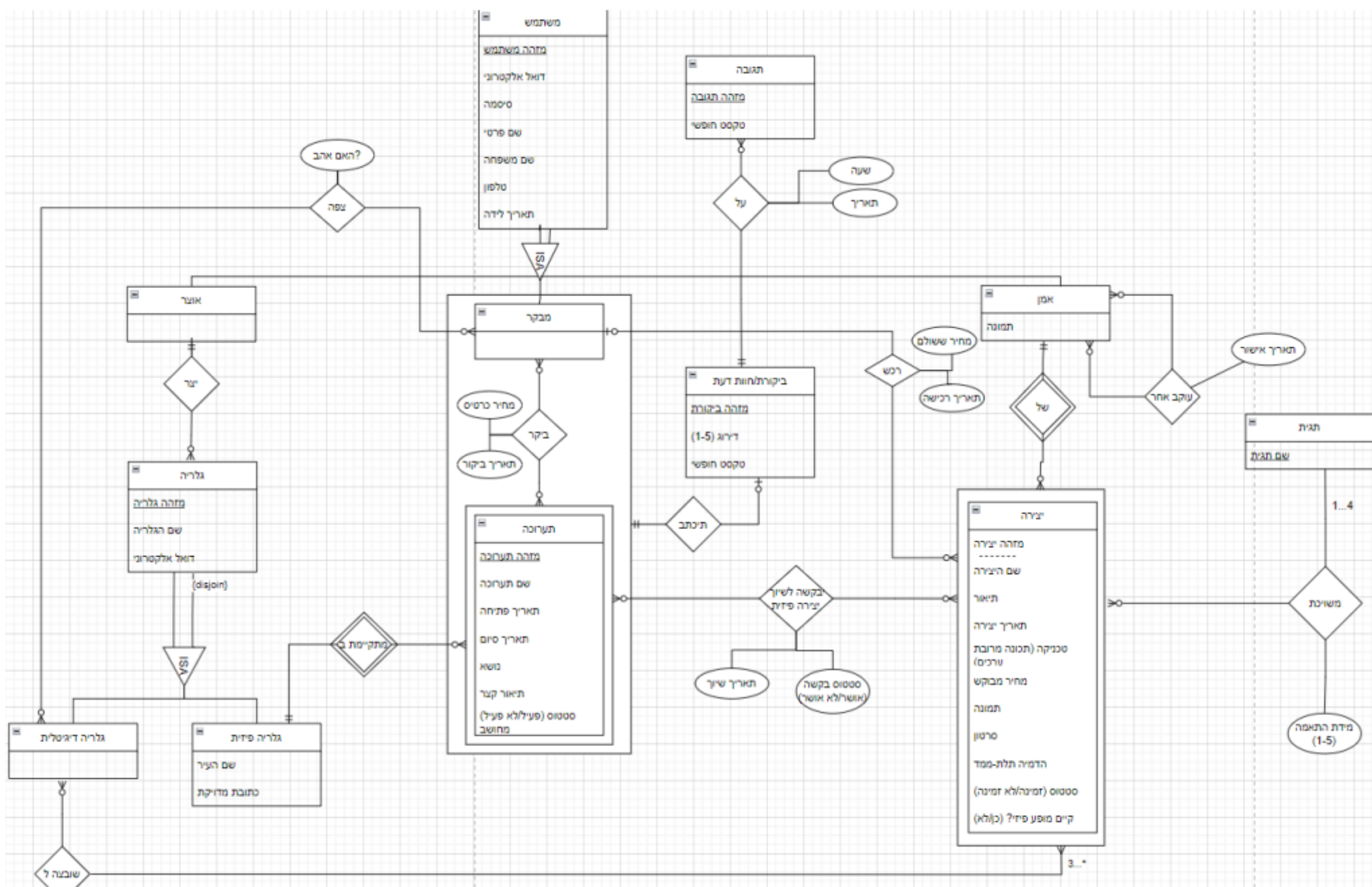
הגדרת סכמת נתונים –

ERD וטבלאות ב- SQL Server

שם המערכת: "ArtConnect+"

סמסטר: חורף, תשפ"ו

פתרון מוצע של צוות הקורס – ישנם פתרונות נכונים נוספים



טבלאות מנורמלות 3NF:

שם הטבלה	שמות השדות
משתמש	מזהה משתמש, כתובת דואל, סיסמה, שם פרטי, שם משפחה, מספר טלפון, תאריך לידה <u>UserID</u> , <u>UserName</u> , <u>UserPassword</u> , <u>UserFirstName</u> , <u>UserLastName</u> , <u>UserPhoneNumber</u> , <u>UserBirthDate</u>
משתמש אוצר	מזהה אוצר (מ.ז. למשתמש) <u>CuratorUserID</u> (F.K <u>tblUser</u>)
משתמש אמן	מזהה אמן (מ.ז. למשתמש), תמונה <u>ArtistUserID</u> (F.K <u>tblUser</u>), <u>ArtistPicture</u>
משתמש מבקר	מזהה מבקר (מ.ז. למשתמש) <u>CriticUserID</u> (F.K <u>tblUser</u>)
טכניקה	שם טכניקה <u>TechnicName</u>
יצירה	מזהה אמן (מ.ז. לאמן), מזהה יצירה, שם יצירה, תיאור, תאריך יצירה, מזהה טכניקה (מ.ז. לטכניקה), מחיר, תמונה, סרטון, הדמיה תלת ממד, סטטוס (זמינה/לא זמינה), קיים מופע פיזי?, מזהה מבקר שרכש יצירה (מ.ז. למבקר), מחיר רכישה, תאריך רכישה. <u>ArtistUserID</u> (F.K. <u>tblArtistUser</u>), <u>CreationID</u> , <u>CreationName</u> , <u>CreationDescription</u> , <u>CreationDate</u> , <u>TechnicName</u> (F.K <u>tblTechnic</u>), <u>CreationPrice</u> , <u>CreationPicture</u> , <u>CreationVedio</u> , <u>Creation3Dvisualization</u> , <u>CreationStatus</u> (Available/Not Available), <u>CreationIsPhysicalShow?</u> (yes/no), <u>CriticUserID</u> (F.K <u>tblCriticUser</u>), <u>PurchasePrice</u> , <u>purchaseDate</u>
משתמש אמן עוקב אחר	מזהה אמן עוקב (מ.ז. אמן), מזהה אמן נעקב (מ.ז. אמן), תאריך אישור <u>FollowerID</u> (F.K. <u>tblArtistUser</u>), <u>FollowedID</u> (F.K. <u>tblArtistUser</u>), <u>ApprovalDate</u>
תגית	מזהה תגית, שם תגית <u>TagName</u>
תגית משויכת ליצירה	שם תגית (מ.ז. לתגית), מזהה יצירה + מזהה אומן (מ.ז. ליצירה), מידת התאמה (1-5) <u>TagName</u> (F.K <u>tblTag</u>), <u>CreationID</u> + <u>ArtistUserID</u> (F.K <u>tblCreation</u>), <u>MatchToCreation</u> (1-5)
גלריה	מזהה גלריה, מזהה אוצר (מ.ז. לאוצר), שם גלריה, דוא"ל <u>GalleryID</u> , <u>CuratorUserID</u> (F.K <u>tblCuratorUser</u>), <u>GalleryName</u> , <u>GalleryEmail</u>
עיר	שם עיר <u>CityName</u>
גלריה פיזית	מזהה גלריה פיזית (מ.ז. לגלריה), מזהה עיר (מ.ז. לעיר), כתובת <u>PhysicalGalleryID</u> (F.K <u>tblGallery</u>), <u>CityName</u> (F.K <u>tblCity</u>), <u>PhysicalGalleryAddress</u>
גלריה דיגיטלית	מזהה גלריה דיגיטלית (מ.ז. לגלריה), מזהה יצירה + מזהה אומן שיבוץ 1 (מ.ז. ליצירה), מזהה יצירה + מזהה אמן שיבוץ 2 (מ.ז. ליצירה), מזהה יצירה + מזהה אמן שיבוץ 3 (מ.ז. ליצירה) <u>DigitalGalleryID</u> (F.K <u>tblGallery</u>), <u>CreationID1</u> + <u>ArtistUserID1</u> (F.K <u>tblCreation</u>), <u>CreationID2</u> + <u>ArtistUserID2</u> (F.K <u>tblCreation</u>), <u>CreationID3</u> + <u>ArtistUserID3</u> (F.K <u>tblCreation</u>)
מבקר צפה בגלריה דיגיטלית	מזהה מבקר (מ.ז. למבקר), מזהה גלריה דיגיטלית (מ.ז. לגלריה דיגיטלית), האם אהב? (כן/לא) <u>CriticUserID</u> (F.K <u>tblCriticUser</u>), <u>DigitalGalleryID</u> (F.K <u>tblDigitalGallery</u>), <u>Like?</u> (yes/no)
תערוכה	מזהה תערוכה, מזהה גלריה פיזית (מ.ז. לגלריה פיזית), שם תערוכה, תאריך פתיחה, תאריך סיום, נושא, תיאור קצר, סטטוס (פעיל/לא פעיל) <u>ExhibitionID</u> , <u>PhysicalGalleryID</u> (F.K <u>tblPhysicalGallery</u>), <u>ExhibitionName</u> ,

ExhibitionDateStart, ExhibitionDateEnd, ExhibitionSubject, ExhibitionDescription, ExhibitionStatus(Active/Inactive)	
<u>מזהה מבקר (מ.ז למבקר), מזהה תערוכה + מזהה גלריה פיזית (מ.ז לתערוכה), מחיר כרטיס ביקור, תאריך ביקור</u> CriticUserID (F.K tblCriticUser), ExhibitionID + PhysicalGalleryID (F.K tblExhibition), TicketPrice, VisitDate	מבקר ביקר בתערוכה tblCVisitedE
<u>מזהה ביקורת, מזהה מבקר + מזהה תערוכה + מזהה גלריה פיזית (מ.ז לביקר), דירוג (1-5), טקסט חופשי</u> ReviewID, CriticUserID + ExhibitionID + PhysicalGalleryID (F.K tblCVisitedE), ReviewDegree(1-5), ReviewText	ביקורת/חוות דעת tblReview
<u>מזהה תגובה, מזהה ביקורת (מ.ז לביקורת), טקסט חופשי, שעה ותאריך</u> ResponseID, ReviewID (F.K Review), ResponseText, ResponseHour, ResponseDate	תגובה tblResponse

פקודות ה-SQL DDL היוצרות את טבלאות ב-SQL Server:

-- משתמש

```
CREATE TABLE tblUser (  
  UserID INT PRIMARY KEY IDENTITY(10000,1),  
  UserEmail NVARCHAR(100) NOT NULL UNIQUE,  
  UserPassword CHAR(8) NOT NULL,  
  UserFirstName NVARCHAR(50) NOT NULL,  
  UserLastName NVARCHAR(50) NOT NULL,  
  UserPhoneNumber CHAR(10) NOT NULL,  
  UserBirthDate DATE NOT NULL,  
  CONSTRAINT CK_User_Email CHECK (UserEmail LIKE '%@%.%'),  
  CONSTRAINT CK_User_Password CHECK (UserPassword LIKE '[A-Z][a-z][a-z][a-z][a-z][a-z][0-9][0-9]'),  
  CONSTRAINT CK_User_Phone CHECK (UserPhoneNumber LIKE '0[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  CONSTRAINT CK_User_Age CHECK (DATEDIFF(YEAR, UserBirthDate, GETDATE()) >= 16)  
);
```

-- משתמש אומן

```
CREATE TABLE tblArtistUser (  
  ArtistUserID INT PRIMARY KEY,  
  ArtistPicture NVARCHAR(255),  
  FOREIGN KEY (ArtistUserID) REFERENCES tblUser(UserID) ON DELETE CASCADE  
);
```

-- משתמש מבקר

```
CREATE TABLE tblCriticUser (  
  CriticUserID INT PRIMARY KEY,  
  FOREIGN KEY (CriticUserID) REFERENCES tblUser(UserID) ON DELETE CASCADE  
);
```

-- משתמש אוצר

```
CREATE TABLE tblCuratorUser(  
  CuratorUserID INT PRIMARY KEY,  
  FOREIGN KEY (CuratorUserID) REFERENCES tblUser(UserID) ON DELETE CASCADE  
);
```

-- טכניקה

```
CREATE TABLE tblTechnic (  
  TechnicName NVARCHAR(50) PRIMARY KEY  
);
```

-- יצירה

```
CREATE TABLE tblCreation (  
  ArtistUserID INT NOT NULL,  
  CreationID INT NOT NULL,  
  CreationName NVARCHAR(100) NOT NULL,  
  CreationDescription NVARCHAR(MAX),  
  CreationDate DATE NOT NULL,  
  TechnicName NVARCHAR(50) NOT NULL,  
  CreationPrice DECIMAL(10,2) NOT NULL DEFAULT 0 CHECK (CreationPrice >= 0),  
  CreationPicture NVARCHAR(255),  
  CreationVideo NVARCHAR(255),  
  Creation3DVisualization NVARCHAR(255),  
  CreationStatus VARCHAR(12) NOT NULL CHECK (CreationStatus IN ('Available', 'Not Available')),  
  CreationIsPhysicalShow BIT NOT NULL DEFAULT 0,  
  CriticUserID INT NULL,  
  PurchasePrice DECIMAL(10,2) CHECK (PurchasePrice >= 0),  
  PurchaseDate DATE,  
  PRIMARY KEY (ArtistUserID, CreationID),
```

```

FOREIGN KEY (ArtistUserID)
REFERENCES tblArtistUser(ArtistUserID) ON DELETE CASCADE,
FOREIGN KEY (TechnicName) REFERENCES tblTechnic(TechnicName),
FOREIGN KEY (CriticUserID) REFERENCES tblCriticUser(CriticUserID)
);

-- משתמש אמן עוקב אחר
CREATE TABLE tblArtistUserFollow (
    FollowerID INT NOT NULL,
    FollowedID INT NOT NULL,
    ApprovalDate DATE,
    PRIMARY KEY (FollowerID, FollowedID),
    FOREIGN KEY (FollowerID) REFERENCES tblArtistUser(ArtistUserID) ON DELETE
    CASCADE,
    -- FOREIGN KEY (FollowedID) REFERENCES tblArtistUser(ArtistUserID) ON
    DELETE
    --CASCADE,
    CONSTRAINT CK_Follow_NotSelf CHECK (FollowerID <> FollowedID)
);

-- תגית
CREATE TABLE tblTag (
    TagName NVARCHAR(50) PRIMARY KEY
);

-- תגית משויכת ליצירה
CREATE TABLE tblAssociatedC(
    TagName NVARCHAR(50) NOT NULL,
    ArtistUserID INT NOT NULL,
    CreationID INT NOT NULL,
    MatchToCreation INT NOT NULL CHECK (MatchToCreation BETWEEN 1 AND 5),
    PRIMARY KEY (TagName, ArtistUserID, CreationID),
    FOREIGN KEY (TagName) REFERENCES tblTag(TagName),
    FOREIGN KEY (ArtistUserID, CreationID) REFERENCES tblCreation(ArtistUserID,
    CreationID) ON DELETE CASCADE
);

-- גלריה
CREATE TABLE tblGallery (
    GalleryID INT PRIMARY KEY IDENTITY(1,1),
    CuratorUserID INT NOT NULL,
    GalleryName NVARCHAR(100) NOT NULL,
    GalleryEmail NVARCHAR(100) NOT NULL,
    FOREIGN KEY (CuratorUserID) REFERENCES tblCuratorUser(CuratorUserID) ON
    DELETE CASCADE,
    CONSTRAINT CK_Gallery_Email CHECK (GalleryEmail LIKE '%@%.%.%')
);

-- עיר
CREATE TABLE tblCity (
    CityName NVARCHAR(50) PRIMARY KEY
);

-- גלריה פיזית
CREATE TABLE tblPhysicalGallery (
    PhysicalGalleryID INT PRIMARY KEY,
    CityName NVARCHAR(50) NOT NULL,
    PhysicalGalleryAddress NVARCHAR(255) NOT NULL,
    FOREIGN KEY (PhysicalGalleryID) REFERENCES tblGallery(GalleryID) ON DELETE
    CASCADE,
    FOREIGN KEY (CityName) REFERENCES tblCity(CityName)
);

-- גלריה דיגיטלית
CREATE TABLE tblDigitalGallery (
    DigitalGalleryID INT PRIMARY KEY,
    ArtistUserID1 INT NOT NULL,

```

```

CreationID1 INT NOT NULL,
ArtistUserID2 INT NOT NULL,
CreationID2 INT NOT NULL,
ArtistUserID3 INT NOT NULL,
CreationID3 INT NOT NULL,
FOREIGN KEY (DigitalGalleryID) REFERENCES tblGallery(GalleryID) ON DELETE
CASCADE,
FOREIGN KEY (ArtistUserID1, CreationID1) REFERENCES
tblCreation(ArtistUserID,
CreationID),
FOREIGN KEY (ArtistUserID2, CreationID2) REFERENCES
tblCreation(ArtistUserID,
CreationID),
FOREIGN KEY (ArtistUserID3, CreationID3) REFERENCES
tblCreation(ArtistUserID,
CreationID),
CONSTRAINT CH_Creative_Differences CHECK(CreationID1<>CreationID2 and
CreationID1<>CreationID3)
);

```

-- מבקר צפה בגלריה דיגיטלית

```

CREATE TABLE tblCViewInDG (
CriticUserID INT NOT NULL,
DigitalGalleryID INT NOT NULL,
Liked BIT NOT NULL DEFAULT 0,
PRIMARY KEY (CriticUserID, DigitalGalleryID),
FOREIGN KEY (CriticUserID) REFERENCES tblCriticUser(CriticUserID),
FOREIGN KEY (DigitalGalleryID) REFERENCES
tblDigitalGallery(DigitalGalleryID)
);

```

-- תערוכה

```

CREATE TABLE tblExhibition (
ExhibitionID INT NOT NULL,
PhysicalGalleryID INT NOT NULL,
ExhibitionName NVARCHAR(100) NOT NULL,
ExhibitionDateStart DATE NOT NULL,
ExhibitionDateEnd DATE NOT NULL,
ExhibitionSubject NVARCHAR(100),
ExhibitionDescription NVARCHAR(MAX),
ExhibitionStatus NVARCHAR(10) CHECK (ExhibitionStatus IN
('Active', 'Inactive')),
PRIMARY KEY (ExhibitionID, PhysicalGalleryID),
FOREIGN KEY (PhysicalGalleryID) REFERENCES
tblPhysicalGallery(PhysicalGalleryID) ON DELETE CASCADE,
CONSTRAINT CK_Exhibition_Dates CHECK (ExhibitionDateEnd >=
ExhibitionDateStart)
);

```

-- מבקר ביקר בתערוכה

```

CREATE TABLE tblCVisitedE (
CriticUserID INT NOT NULL,
ExhibitionID INT NOT NULL,
PhysicalGalleryID INT NOT NULL,
TicketPrice DECIMAL(10,2) NOT NULL CHECK (TicketPrice >= 0),
VisitDate DATE NOT NULL,
PRIMARY KEY (CriticUserID, ExhibitionID, PhysicalGalleryID),
FOREIGN KEY (CriticUserID) REFERENCES tblCriticUser(CriticUserID),
FOREIGN KEY (ExhibitionID, PhysicalGalleryID) REFERENCES
tblExhibition(ExhibitionID, PhysicalGalleryID)
);

```

-- ביקורת/חוות דעת

```

CREATE TABLE tblReview (
ReviewID INT PRIMARY KEY IDENTITY(1,1),
CriticUserID INT NOT NULL,
ExhibitionID INT NOT NULL,

```

```

PhysicalGalleryID INT NOT NULL,
ReviewDegree INT CHECK (ReviewDegree BETWEEN 1 AND 5),
ReviewText NVARCHAR(MAX),
UNIQUE (CriticUserID, ExhibitionID, PhysicalGalleryID),
FOREIGN KEY (CriticUserID, ExhibitionID, PhysicalGalleryID)
REFERENCES tblCVisitedE(CriticUserID, ExhibitionID, PhysicalGalleryID)
ON DELETE CASCADE
);

-- תגובה
CREATE TABLE tblResponse (
ResponseID INT PRIMARY KEY IDENTITY(1,1),
ReviewID INT NOT NULL,
ResponseText NVARCHAR(MAX) NOT NULL,
ResponseDateTime DATETIME NOT NULL DEFAULT GETDATE(),
FOREIGN KEY (ReviewID) REFERENCES tblReview(ReviewID) ON DELETE CASCADE
);

```


אילוצים שלא ניתן לפתור ברמת קוד הטבלאות/ברמת התרשים ERD

1. גלריה פיזית יכולה להפעיל כמה תערוכות במקביל – בתנאי שהן אינן בעלות אותו שם. בעת הכנסת רשומה לטבלת "מתקיימת ב" אין דרך לאלץ את השדה שם התערוכה להיות שונה מכל שמות התערוכה שכבר קיימים בטבלה. (ניתן לפתור ברמת האפליקציה).
2. אמנים רשאים להגיש בקשת שיוך של יצירותיהם לתערוכות פעילות, אך אותה יצירה לא יכולה להשתתף בשתי תערוכות שתאריכיהן חופפים. בעת הכנסת רשומה לטבלת "בקשת שיוך" אין דרך לאלץ לבדוק האם יצירה זו כבר נמצאת בתערוכה פעילה שתאריכיה חופפים עם תאריך הבקשת שיוך. (ניתן לפתור ברמת האפליקציה).
3. אמן לא רשאי לרכוש יצירה של עצמו. בעת הכנסת רשומה לטבלת "רכש" לא ניתן לוודא כי ערכו של המזהה של המשתמש מבקר לא נמצא גם בטבלת האמנים במערכת. (ניתן לפתור ברמת האפליקציה). בנוסף לא הוגדר במודל disjoin ERD על ישות משתמש.
4. תאריך רכישה של יצירה צריך להיות גדול או שווה לתאריך הביקור של המבקר בתערוכה ובין תאריכי התערוכה בפועל. בעת הכנסת רשומה לטבלת "רכש" לא ניתן לבדוק זאת כיוון שנתונים אלו נמצאים בטבלאות "ביקר" ו"תערוכה". (ניתן לפתור ברמת האפליקציה).
5. מבקר אינו יכול לרכוש יותר מכרטיס אחד לאותה תערוכה. בעת הכנסת רשומה לטבלת "ביקר" לא ניתן לבדוק האם מזהה המבקר X בתערוכה של גלריה פיזית Y לא קיים כבר כרשומה בטבלה. (ניתן לפתור ברמת האפליקציה).
6. בהנחה שיצרנו טבלת קשר משויכת בין תגית ליצירה ונדרש לשייך ליצירה של אומן לכל היותר 1 או 4 תגיות שונות. בעת הכנסת רשומה לטבלת "תגית משויכת ליצירה" לא ניתן לבדוק האם ליצירה X של אמן Y שוייכו כבר 4 תגיות שונות. (ניתן לפתור ברמת האפליקציה).
7. יוקם מזהה יצירה הממוספר בסדר רץ (החל מ-1 עבור אמן זה). לא ניתן להגדיר IDENTITY לצורך הגדרת מספר רץ עבור כל אמן מכיוון שהוא כללי לכלל הרשומות בטבלה. (ניתן לפתור ברמת האפליקציה)
8. לכל תערוכה מוגדרים, מזהה ייחודי (מספר רץ החל מ-1 עבור כל גלריה פיזית). לא ניתן להגדיר IDENTITY לצורך הגדרת מספר רץ עבור כל אמן מכיוון שהוא כללי לכלל הרשומות בטבלה. (ניתן לפתור ברמת האפליקציה)