מטלת בונוס

מערכת לניהול פסטיבלים והופעות מוזיקה:

המערכת נועדה לנהל פסטיבלים והופעות מוזיקה, ותומכת במגוון פעילויות ניהול.
במסגרת ניהול האומנים והלהקות, המערכת מאפשרת להזין ולשמור את פרטי המשתתפים, כולל שמות, סגנונות מוזיקליים, פרטי אנשי קשר ודרישות טכניות להופעות. בנוסף, ניתן לנהל את הבמות השונות באירוע, תוך תיעוד שם הבמה, מיקומה, קיבולת הקהל והציוד הקיים בה.

בתחום הכרטיסים, המערכת תומכת בהגדרת סוגי כרטיסים שונים (כגון רגיל, VIP או יומי), במעקב אחר מכירות ובניהול מקומות בהתאם לסוג האירוע - ישיבה או עמידה. ניהול ההופעות כולל קביעת מועדים, שיבוץ אומנים, שיוך לבמות רלוונטיות והגדרת משך ההופעה.

המערכת מאפשרת גם ניהול של פרטי הקהל, הכוללים את פרטי רוכשי הכרטיסים, מעקב אחרי כניסות לאירוע והנפקת כרטיסים דיגיטליים. לצד זה ניתן לנהל מידע על ספונסרים, כולל פרטי התקשרות, חבילות חסות ותיעוד של הסכמים.

כמו כן, המערכת מספקת כלים לניהול עובדים ומתנדבים, עם אפשרות לשבץ אותם לתפקידים שונים כגון סדרנים, אבטחה, דוכנים ותפעול. לבסוף, המערכת מאפשרת הפקת דוחות ניהוליים מגוונים - כגון דוחות מכירת כרטיסים, תפוסה של הופעות והכנסות כוללות - לטובת קבלת החלטות וניהול יעיל של האירוע.

המערכת צריכה לתמוך בתרחישים הבאים:

1. הצגת רשימת האומנים המשתתפים בפסטיבל, כולל סגנון מוזיקה ודרישות טכניות.
2. הצגת ההופעות המתוכננות – תאריך, שעה, אומן, במה ומשך הופעה.
3. הצגת סוגי הכרטיסים לאירוע (רגיל, VIP, יומי) ומספר הכרטיסים שנמכרו מכל סוג.
4. הצגת פרטי רוכשי כרטיסים ובדיקת כניסה לאירוע באמצעות כרטיס דיגיטלי.
5. הפקת דוח הכנסות כולל ממכירת כרטיסים לכל ההופעות בפסטיבל.

1. כיתבו קוד ליצירת הטבלאות הנדרשות. הקפידו על נרמול.
2. פרטו נמקו את כל ההחלטות העיצוביות שקיבלתם:
   a. ציינו מה כל הישויות שאתם מייצגים, מה התרחיש בשבילו הם נדרשים, ומה היצוג שנבחר. אם יש ישויות רלוונטיות שאתם לא מייצגים ציינו והסבירו למה.
   b. הסבירו את השדות שבחרתם, התרחיש לו הם נדרשים, והטיפוס שבחרתם להם.
   c. נמקו את המפתחות שבחרתם, האם הם טבעיים (קיימים בעולם האמיתי) או פנימיים (לדוגמה auto increment)
   d. ציינו מה היחסים בין הישויות (1:1, n:1, או n:n) בעולם האמיתי. אם בחרתם ליצג יחס אחר, ציינו ונמקו. מספיק לציין יחסים רק בין ישויות שכנות, שיש בניהן קשר ישיר.
   e. פרטו מה ה constraints (לדוגמה: unique, FK, not null, check) שבחרתם ומה ההגנה שהם מספקים. אם יש נקודות שלא מוגנות על ידי בסיס הנתונים ציינו אותן ונמקו איך להגן.

3. כיצד תמנעו שאמן יופיע בשעות חופפות או שיהיו הופעות בשעות חופפות באותה הבמה?
4. נמקו את היצוג של היחס (אמן, סגנון מוזיקלי). פרטו מה היתרונות והחסרונות של בחירתכם.
5. יש אמנים בעלי יכולת שונה למשוך קהל, מה שצריך להשפיע את הבמות המוקצות להם. כיצד אתם מציעים לנהל זאת? אחד השיקולים שעליכם לקחת בחשבון הוא פרימדונות ואגו.
6. ננסה למדוד אגו כהופעה בבמות בגודל גבוה מהביקוש להופעה. שילפו את האמנים שיחס הכרטיסים לגודל במה שלהם גרוע מהממוצע.
7. מה ההשלכה של כרטיסים יומיים על השאילתא הקודמת? כיצד ראוי להתייחס לכך? נמקו

פתרון:

1. קוד ליצירת הטבלאות הנדרשות:

```
DROP DATABASE IF EXISTS FestivalDB;

CREATE DATABASE FestivalDB;

USE FestivalDB;


CREATE TABLE Genres (

    genre_id INT PRIMARY KEY AUTO_INCREMENT,

    genre_name VARCHAR(100) UNIQUE NOT NULL

);


CREATE TABLE Artists (

    artist_id INT PRIMARY KEY AUTO_INCREMENT,

    name VARCHAR(100) NOT NULL,

    genre_id INT NOT NULL,

    requirements VARCHAR(100) NOT NULL,

    popularity_level INT CHECK (popularity_level BETWEEN 1 AND 10),

    CONSTRAINT fk_genre FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)

);
```

```
CREATE TABLE Stages (
    stage_id INT PRIMARY KEY AUTO_INCREMENT,
    stage_name VARCHAR(100) NOT NULL UNIQUE,
    location VARCHAR(100),
    capacity INT CHECK (capacity > 0),
    equipment TEXT
);

CREATE TABLE Performances (
    performance_id INT PRIMARY KEY AUTO_INCREMENT,
    artist_id INT NOT NULL,
    stage_id INT NOT NULL,
    start_time DATETIME NOT NULL,
    end_time DATETIME NOT NULL,
    CONSTRAINT fk_artist FOREIGN KEY (artist_id) REFERENCES Artists(artist_id),
    CONSTRAINT fk_stage FOREIGN KEY (stage_id) REFERENCES Stages(stage_id),
    CONSTRAINT chk_time CHECK (end_time > start_time)
);

CREATE TABLE Audience (
    audience_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
);
```

```
CREATE TABLE Tickets (

    ticket_id INT PRIMARY KEY AUTO_INCREMENT,

    ticket_type ENUM('REGULAR','VIP','DAILY') NOT NULL,

    price DECIMAL(10,2) NOT NULL,

    audience_id INT NOT NULL,

    performance_id INT NOT NULL,

    status ENUM('VALID','USED') DEFAULT 'VALID',

    CONSTRAINT fk_audience FOREIGN KEY (audience_id) REFERENCES
Audience(audience_id),

    CONSTRAINT fk_performance FOREIGN KEY (performance_id) REFERENCES
Performances(performance_id)

);
```

2. הישויות והתרחישים:

- **Genres** (סגנונות מוזיקה)
  נדרשים לניהול קטגוריות מוזיקליות (לדוגמה: רוק, ג'אז).
  מימוש בטבלה נפרדת מונע כפילויות ומאפשר נרמול מלא (טבלת Artists מצביעה ל־Genres).

- **Artists** (אומנים/להקות)
  נדרשים לניהול אומנים המשתתפים בפסטיבל, כולל שם, דרישות טכניות ורמת פופולריות (יכולת משיכת קהל).
  מיוצגים בטבלה עם FK ל־Genres.

- **Stages** (במות)
  נדרשים לניהול הבמות בהן מתקיימות ההופעות: שם במה, מיקום, קיבולת וציוד.
  טבלה עצמאית, ללא תלות.

- **Performances** (הופעות)
  נדרשים לניהול הופעות בפסטיבל: אומן, במה, זמן התחלה וזמן סיום.
  מיוצגים בטבלה המקשרת בין Artists ל־Stages.

- **Audience** (קהל)
  נדרשים לניהול פרטי רוכשי כרטיסים (שם, אימייל).
  מיוצגים בטבלה עם אימייל ייחודי.

- **Tickets** (כרטיסים)
  נדרשים לניהול כרטיסים שנמכרו: סוג כרטיס, מחיר, שיוך לרוכש ולהופעה, סטטוס (שומש/בתוקף).
  מיוצגים בטבלה המקשרת בין Audience ל־Performances.

השדות שנבחרו

- **מזהים**: INT AUTO_INCREMENT - מפתחות פנימיים רציפים, שאינם תלויים בנתונים אמיתיים.
- **VARCHAR** - לשמות, מיקומים ואימיילים (גמיש ומתאים לטקסט קצר).
- **DECIMAL** - למחירים, כדי לשמור על דיוק כספי
- **DATETIME** - לזמני הופעות (תחילה וסיום).
- **ENUM** - לערכים מוגדרים מראש (סוג כרטיס, סטטוס).
- **CHECK** - לחוקים פשוטים כמו טווח רמות פופולריות או קיבולת במה.

המפתחות

- **Primary Key** - מפתח מלאכותי פנימי (AUTO_INCREMENT).
- אבחר במקום מפתח טבעי, כי שמות או אימיילים עלולים להשתנות או להיות כפולים.
- **Foreign Key** - שמירה על שלמות ייחוסית (למשל אומן חייב להשתייך לסגנון קיים).
- **UNIQUE** - מניעת כפילות באימיילי קהל, שמות במות ושמות סגנונות.

היחסים בין הישויות

- **Genres ו- Artists** : יחס N:1 (סגנון יכול להשתייך להרבה אומנים, לאומן סגנון אחד בלבד).
- **Artists ו- Performances** : יחס N:1 (אומן מופיע בכמה הופעות, הופעה קשורה לאומן יחיד).
- **Stages ו- Performances** : יחס N:1 (במה מארחת הופעות שונות, הופעה מתקיימת על במה אחת).
- **Performances ו- Tickets** : יחס N:1 (להופעה נמכרים מספר כרטיסים).

- **Audience ו- Tickets** : יחס N:1 (לקוח יכול לרכוש מספר כרטיסים).

Constraints והגנות

- **NOT NULL** - על שדות חובה (שם אומן, זמנים, מחיר כרטיס).
- **UNIQUE** - על אימייל בקהל, שמות במות וסגנונות מוזיקה למניעת כפילויות.
- **CHECK** -
  - רמת פופולריות בין 1–10.
  - קיבולת במה > 0.
  - שעת סיום הופעה אחרי שעת התחלה.
- **FK** - שמירת שלמות ייחוסית בין טבלאות.

3. מניעת הופעות חופפות

לא ניתן להבטיח מניעת חפיפה בעזרת מפתחות או CHECK, כיוון שיש צורך להשוות בין שורות שונות.
הפתרון הוא ניהולי: בעת יצירת הופעה חדשה או עדכון הופעה קיימת, המערכת תבצע שאילתת בדיקה לוודא שאין חפיפה בזמן עבור אותו אומן או אותה במה.

כדי לייעל את התהליך, נגדיר אינדקסים על (artist_id, start_time, end_time) ועל (stage_id, start_time, end_time), כך שהבדיקה תהיה מהירה גם כשיש הרבה הופעות.
אם מתגלת חפיפה – הפעולה תיחסם.

כך נשמרת שלמות הנתונים ונמנעות טעויות שיבוץ גם ברמת האמן וגם ברמת הבמה.

4. ייצוג היחס אומן–סגנון מוזיקלי

לרוב האמנים יש סגנון עיקרי אחד (`Artists.genre_id → Genres.genre_id`), מה שמקל על סיווג ושליפה סטטיסטית. עם זאת, לא כל האמנים מתאימים לז'אנר יחיד: יש אמנים רב-ז'אנריים, ויש אמנים שנשמעים כמו אחרים אך רוצים להגדיר לעצמם ז'אנר ייחודי שהמציאו.

כדי להתמודד עם מקרים אלו ניתן להרחיב את המודל בשתי דרכים:

1. **טבלת קשר `ArtistGenres` : Many-to-Many -** מאפשרת שיוך למספר ז'אנרים רשמיים במקביל.

2. **שדה טקסט חופשי (`custom_genre`)** - מאפשר לאמן להגדיר ז'אנר ייחודי או חדש שהמציא. השדה אופציונלי כדי לא לפגוע בנרמול או ליצור כפילויות.

השילוב מאפשר לשמור על סיווג ברור לרוב האמנים, ובמקביל נותן מענה לאמנים יוצאי דופן או רב-ז'אנריים, מבלי לפגוע ביכולת ניתוח הנתונים או בניהול הפסטיבל.

.5

כדי לנהל את התאמת האמנים לבמות, לכל אמן נשמר שדה `popularity_level`: 1–10 ובטבלת הבמות נשמרת `capacity`. השיבוץ נעשה כך שאמנים פופולריים יקבלו במות גדולות יותר, ואמנים פחות מוכרים יקבלו במות קטנות יותר.

לצורך ניתוח "פרימדונות ואגו", מחשבים את **יחס הביקוש להופעה (RATIO)**:

$$\frac{\text{Tickets Sold} + \text{Waitlist Count}}{\text{Stage Capacity}} = \text{Ratio}$$

- RATIO ≈ 1 → הקצאה מתאימה.
- RATIO ≪ 1 → הבמה גדולה מדי ביחס לביקוש (אגו).
- RATIO > 1 → הבמה קטנה ביחס לביקוש האמיתי (מספר האנשים שמעוניינים להגיע גבוה מהקיבולת).

שיטה זו מאפשרת הפקת דוחות ניהוליים, זיהוי מקרים של אגו או חוסר מקום, ותכנון שיבוצים טובים יותר בפסטיבלים עתידיים, מבלי למכור יותר כרטיסים מהקיבולת בפועל.

.6

שלב 1: יצירת VIEW עם סך הכרטיסים והיחס לכל הופעה

```
USE FestivalDB;

CREATE OR REPLACE VIEW PerformanceTickets AS

SELECT

    p.performance_id,

    a.artist_id,

    s.stage_id,

    COUNT(t.ticket_id) AS tickets_sold,

    s.capacity,

    COUNT(t.ticket_id) / s.capacity AS ratio

FROM Performances p

JOIN Artists a ON p.artist_id = a.artist_id

JOIN Stages s ON p.stage_id = s.stage_id

LEFT JOIN Tickets t ON p.performance_id = t.performance_id

GROUP BY p.performance_id, a.artist_id, s.stage_id, s.capacity;
```

שלב 2: VIEW עם ממוצע יחס לכל אמן

```
CREATE OR REPLACE VIEW ArtistAverageRatio AS

SELECT

    artist_id,

    AVG(ratio) AS avg_ratio

FROM PerformanceTickets

GROUP BY artist_id;
```

שלב 3: שאילתת האמנים עם "אגו" (יחס נמוך מהממוצע של אותו אמן)

```
SELECT
    a.name AS artist_name,
    s.stage_name,
    pt.tickets_sold,
    pt.capacity,
    pt.ratio
FROM PerformanceTickets pt
JOIN Artists a ON pt.artist_id = a.artist_id
JOIN Stages s ON pt.stage_id = s.stage_id
JOIN ArtistAverageRatio aar ON pt.artist_id = aar.artist_id
WHERE pt.ratio < aar.avg_ratio;
```

.7

כרטיסים יומיים והשפעתם על חישוב יחס תפוסה:
כרטיסים יומיים מקנים כניסה לכל ההופעות ביום מסוים, אך לא כל רוכש יגיע לכל ההופעות. אם נכליל אותם ישירות בחישוב `tickets_sold / capacity`, היחס יהיה **מוטה כלפי מעלה** והערכת "אגו" או ביקוש אמיתי תהיה לא מדויקת.

כיצד להתייחס לכך:

- לא לכלול את הכרטיסים היומיים ישירות בחישוב יחס תפוסה להופעה בודדת.
- ניתן לנהל אותם בנפרד או לחלק את סך הכרטיסים היומיים בין ההופעות באותו יום לפי כלל מוסכם (למשל לפי קיבולת הבמה או פופולריות האמן), עם ציון שמדובר בהערכה בלבד.
- לחלופה מדויקת יותר, להשתמש בלוג כניסה בפועל להופעות.