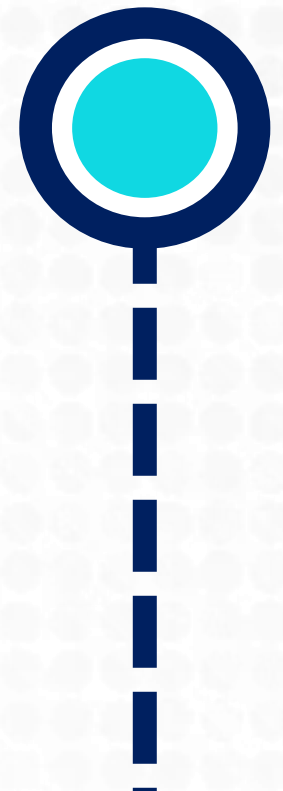
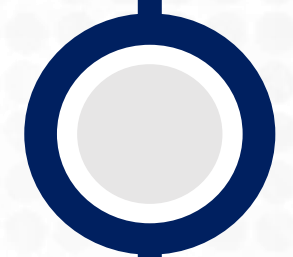


MySQL

במסגרת הנושא נכיר ונעמיק ידע ב-MYSQL ונלמד על אפשרויות מתקדמות בניהול מסדי נתונים, כולל שאילתות מורכבות ועוד.



מבוא והתקנות



MySQL – מבוא והתקנות

בחלק זה נכיר את MySQL אשר מקלה עלינו בניהול מסדי נתונים, בסיום הנושא תוכלו לענות על השאלות הבאות:



[MySQL :: Download MySQL Installer](#)

- מהי SQL ומה הקשר ל-MYSQL?
- במה היא מסייעת לנו?
- מהם השימושים המרכזיים של MySQL בעולם האמיתי?
- מהם הכלים השונים לגישה לניהול MySQL?
- כיצד אוכל להוריד ולהתקין את MySQL?
- מה עלינו להגדיר במהלך ההתקנה?
- כיצד מתחברים לשרת MySQL בפעם הראשונה לאחר ההתקנה?

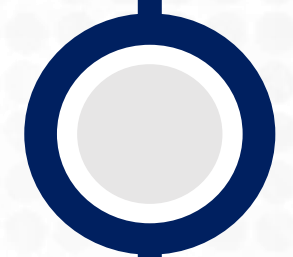
מבנה נתונים טבלאי

בחלק זה נכיר את היכולות הבסיסיות של MySQL אשר מקלה עלינו בניהול ושמירה של נתונים בצורה מאורגנת ויעילה. בסיום הנושא תוכלו לענות על השאלות הבאות:

	ProductId	ProductName	Description	Price	Category	Gender	StockQuantity	DateAdded
▶	1	Slim Fit Jeans	Classic blue slim fit jeans.	49.99	Jeans	Female	200	2023-01-03 00:00:00
	2	Striped T-Shirt	Cotton t-shirt with blue and white stripes.	19.99	T-Shirts	Female	150	2023-01-06 00:00:00
	3	Leather Jacket	Black leather jacket with zipper.	99.99	Jackets	Male	80	2023-01-09 00:00:00
	4	Ankle Boots	Black leather ankle boots for women.	79.99	Shoes	Female	120	2023-01-13 00:00:00
	5	Cotton Socks	Pack of 5 pairs of cotton socks.	9.99	Socks	Unisex	300	2023-01-15 00:00:00
	6	Wool Sweater	Warm wool sweater in light grey color.	59.99	Sweaters	Female	100	2023-01-20 00:00:00
	7	Silk Scarf	Elegant silk scarf with floral pattern.	29.99	Accessories	Female	200	2023-01-25 00:00:00
	8	Pleated Skirt	Black pleated skirt in midi length.	39.99	Skirts	Female	120	2023-01-30 00:00:00
	9	Tote Bag	Large tote bag in brown leather.	69.99	Bags	Female	90	2023-02-01 00:00:00
	10	Bomber Jacket	Green bomber jacket with side pockets.	89.99	Jackets	Unisex	100	2023-02-03 00:00:00
	11	Sports Bra	High support sports bra in black.	24.99	Sportswear	Female	150	2023-05-15 00:00:00
	12	Running Shoes	Men's running shoes in white.	59.99	Shoes	Unisex	100	2023-05-20 00:00:00
	13	Denim Shorts	High waisted denim shorts for women.	29.99	Shorts	Female	200	2023-05-22 00:00:00
	14	Men's Polo	Polo shirt in navy blue for men.	34.99	Polos	Male	180	2023-05-25 00:00:00
	15	Kids Sneakers	Colorful sneakers for kids.	29.99	Shoes	Female	250	2023-05-28 00:00:00
	16	Baby Romper	Cotton romper with cute prints for babies.	14.99	Baby	Unisex	300	2023-05-30 00:00:00
	17	Cycling Gloves	Comfortable cycling gloves in black.	19.99	Sportswear	Unisex	100	2023-06-01 00:00:00
	18	Rain Jacket	Waterproof rain jacket in yellow.	49.99	Jackets	Female	150	2023-06-04 00:00:00
	19	Sunglasses	UV protection sunglasses with round fra...	29.99	Accessories	Female	200	2023-06-06 00:00:00
	20	Beach Hat	Wide brim beach hat in straw.	19.99	Accessories	Female	220	2023-06-10 00:00:00
	21	Canvas Sneak...	Classic white canvas sneakers.	39.99	Shoes	Unisex	250	2023-02-15 00:00:00
	22	Leather Belt	Brown leather belt with brass buckle.	24.99	Accessories	Unisex	300	2023-02-20 00:00:00
	23	Wool Hat	Warm wool hat for winter.	19.99	Accessories	Female	150	2023-02-25 00:00:00
	24	Gym Shorts	Comfortable gym shorts for men.	29.99	Sportswear	Male	200	2023-03-01 00:00:00
	25	Yoga Mat	Non-slip yoga mat.	34.99	Sportswear	Unisex	100	2023-03-05 00:00:00
	26	Denim Jacket	Blue denim jacket with buttons.	59.99	Jackets	Female	150	2023-03-10 00:00:00
	27	Silk Dress	Elegant silk dress in black.	89.99	Dresses	Female	100	2023-03-15 00:00:00
	28	Leather Wallet	Compact leather wallet for men.	49.99	Accessories	Unisex	200	2023-03-20 00:00:00

- מהו מבנה נתונים טבלאי?
- כיצד מאורגנים הנתונים בטבלה?
- מה היא רשומה (Record) ומה היא עמודה (Column) בטבלה?
- מה הוא שדה בטבלה?
- מהם סוגי הנתונים (Data Types) הנפוצים בטבלאות SQL?
- במה מסייע לנו מבנה נתונים טבלאי לניהול מידע?
- כיצד מאגר נתונים טבלאי מסייע בשליפה וארגון מידע?

MySQL Workbench



CREATE/create-schema.sql

```
CREATE DATABASE IF NOT EXISTS `moviesdb`;
```

```
USE `moviesdb`;
```

```
DROP TABLE IF EXISTS `actor`;
```

```
CREATE TABLE
```

```
moviesdb.actor (
  `Name` NVARCHAR (50) NOT NULL,
  `Age` INT NOT NULL,
  `Gender` NVARCHAR (50) NOT NULL,
  `Country` NVARCHAR (50) NOT NULL
);
```

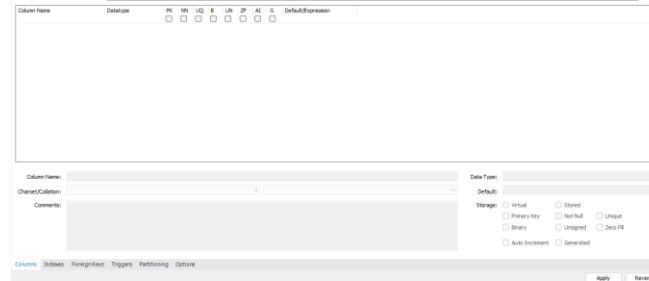
```
DROP TABLE IF EXISTS `movie`;
```

```
CREATE TABLE
```

```
moviesdb.movie (
  `Name` NVARCHAR (50) NOT NULL,
  `Year` INT NOT NULL,
  `Genre` NVARCHAR (50) NOT NULL,
  `TimeInMinutes` INT NOT NULL
);
```

בחלק זה נכיר את MySQL Workbench, כלי עזר רב-עוצמה לניהול, עיצוב וביצוע שאילתות במאגרי נתונים. בסיום הנושא תוכלו לענות על השאלות הבאות:

- מהו MySQL Workbench ובמה הוא מסייע לנו?
- כיצד מתקינים ומגדירים MySQL Workbench?
- כיצד יוצרים ומנהלים חיבורים לשרתים ב-Workbench?
- מה היא סכמה ומדוע היא חשובה לנו?
- כיצד נוכל ליצור סכמה חדש באמצעות Workbench?
- כיצד ניתן לעצב וליצור טבלאות באמצעות הכלי?
- כיצד ניתן לשלוח שאילתות SQL בעזרת MySQL Workbench?



תרגול – MySQL Workbench

הפעילו את **MySQL Workbench** ופתרו את התרגילים לפי הסדר.

Movie	
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

תרגיל	תיאור המשימה
Ex-1	צרו סכמה חדשה בשם moviesdb .
Ex-2	צרו טבלה בשם movie והכניסו לה עמודות בהתאם לדוגמה.
Ex-4	צרו טבלה בשם actor והכניסו לה עמודות בהתאם לדוגמא.

הוספת נתונים באמצעות שאלות



הוספת נתונים באמצעות שאילתות

בחלק זה נלמד כיצד ניתן להוסיף נתונים חדשים למסד הנתונים באמצעות שאילתות SQL. בסיום הנושא תוכלו לענות על השאלות הבאות:



- כיצד נראית פקודת **INSERT** בסיסית?
- כיצד ניתן להוסיף הערה למסמך **SQL**?
- מהם השלבים להוספת רשומה חדשה לטבלה?
- כיצד ניתן להוסיף רשומות מרובות בפקודת **INSERT** אחת?
- מהם השדות שעלינו לכלול בשאילתת **INSERT**?

INSERT INTO

```
moviesdb.movie (`Name`, `Year`, `Genre`, `TimeInMinutes`)
```

VALUES

```
("The Zlit", 2014, "Sci-Fi", 90);
```

הוספת נתונים באמצעות שאילתות – תרגול

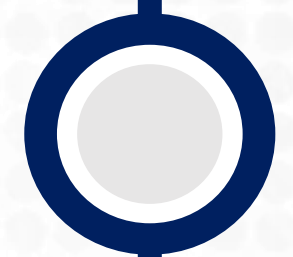
הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.

Movie	
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

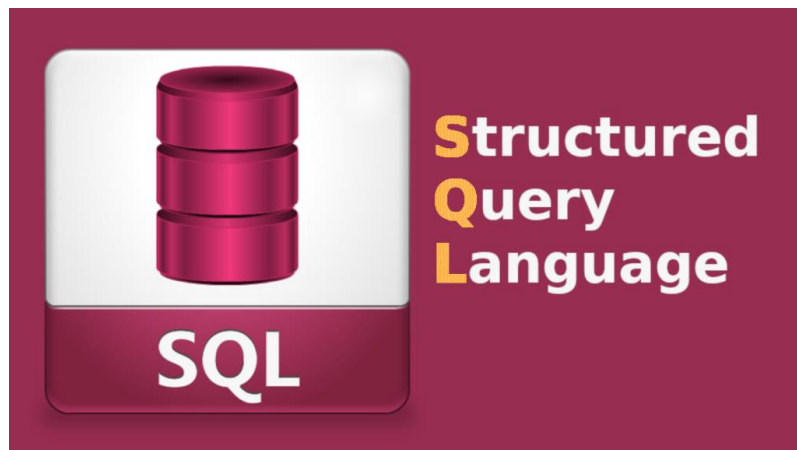
תרגיל	תיאור המשימה
Ex-1	הוסיפו לטבלה movie רשומה חדשה על ידי שימוש בשאילתה.
Ex-2	הוסיפו לטבלה movie 4 רשומות חדשות על ידי שימוש בשאילתה אחת.
Ex-4	הוסיפו לטבלה actor 5 רשומות חדשות על ידי שימוש בשאילתה אחת.

שליפת נתונים באמצעות שאילתות



שליפת נתונים באמצעות שאילתות – תרגול

בחלק זה נכיר את התחביר הבסיסי של שאילתות ב-MYSQL ונלמד כיצד לשלוק נתונים בצורה יעילה. בסיום הנושא תוכלו לענות על השאלות הבאות:



```
SELECT * FROM moviesdb.movie  
WHERE movie.TimeInMinutes > 100  
ORDER BY movie.name DESC;
```

Name	Year	Genre	TimeInMinutes
Zinbad The Mailman	2005	Drama	108
Zaratusta	1967	Drama	125
The Frog	2019	Action	120

- מהו התחביר הבסיסי של שאילתת SQL?
- כיצד נראית שאילתה פשוטה לשליפת נתונים ומה המשמעות של "כוכבית" (*)?
- מהו המבנה של פקודת SELECT וכיצד משתמשים בה?
- כיצד מסננים נתונים באמצעות WHERE?
- מה השימוש באופרטורים AND, OR ו-NOT?
- כיצד ניתן למיין תוצאות עם ORDER BY?
- מה ההבדל בין ASC ל-DESC?
- מהו LIMIT וכיצד מגבילים את כמות התוצאות?
- כיצד ניגשים לשדות ספציפיים בטבלה?

שאלות – שליפת נתונים – תרגול

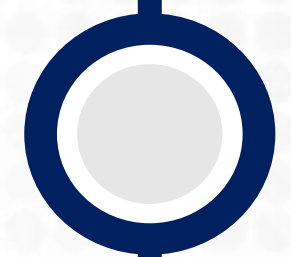
המשיכו את התרגיל הקודם ופתרו את התרגילים לפי הסדר.

Movie	
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

תרגיל	תיאור המשימה
Ex-1	כתבו שאלתה לטבלת הסרטים שתחזיר רק את הסרטים שיצאו בשנת 2024.
Ex-2	כתבו שאלתה לטבלת הסרטים שתחזיר רק את סרטי הקומדיה.
Ex-3	כתבו שאלתה לטבלת הסרטים שתחזיר את כל סרטי האקשן שיצאו בשנת 2024 ואורכם מעל ל-120 דקות בסדר עולה לפי שם הסרט.
Ex-4	כתבו שאלתה לטבלת השחקנים שתחזיר רק את השחקנים שעברו את גיל 36.
Ex-5	כתבו שאלתה לטבלת השחקנים שתחזיר רק את השחקנים שגרים בארה"ב.
Ex-6	כתבו שאלתה לטבלת השחקנים שתחזיר את כל את השחקניות שגרות בארה"ב וגילן מעל ל-20 בסדר יורד לפי גיל השחקנית.

עדכון נתונים באמצעות שאלות



עדכון נתונים באמצעות שאילתות

בחלק זה נלמד כיצד לעדכן נתונים קיימים במסד הנתונים בעזרת שאילתת UPDATE בסיום הנושא תוכלו לענות על השאלות הבאות:

```
-- *** Simple example of an UPDATE statement in SQL ***
```

```
-- Safe mode is enabled by default in MySQL.
-- This means that you cannot update a table without a WHERE clause on
-- a column that is not a Primary Key.
-- This is also used to prevent you from accidentally updating all rows in a table.
```

SET

```
-- You can disable safe mode by running the following command:
SQL_SAFE_UPDATES = 0;
```

UPDATE moviesdb.actor

SET

```
actor.Name = "Yen Wu Solomon"
```

WHERE

```
actor.Name = "Bob Smoon";
```

```
-- You can enable safe mode by running the following command:
```

SET

```
SQL_SAFE_UPDATES = 1;
```

```
-- Please note that if we had Primary Key in the table,
-- we could have used it in the WHERE clause instead of the Name column
-- without disabling safe mode, which would have been more efficient .
```



- כיצד נראית פקודת UPDATE בסיסית?
- לשם מה משמש האופרטור SET?
- מה זה SQL_SAFE_UPDATES?
- מדוע שנרצה לבטל אותו כל עוד אין מזהה ייחודי לטבלה?
- מהם השלבים לעדכון רשומות בטבלה?
- כיצד ניתן לסנן את הנתונים שרוצים לעדכן באמצעות WHERE?
- לשם מה נשתמש באופרטור LIKE?
- מה תפקידו של "אחוז" (%) בהקשר זה?
- כיצד ניתן לשלב תנאים מורכבים בעזרת פקודת UPDATE?
- מה חשוב לזכור כאשר מבצעים עדכון על מספר שדות?

עדכון נתונים באמצעות שאילתות – תרגול

הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.

Movie	
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

תרגיל	תיאור המשימה
Ex-1	עדכנו את השם של אחד השחקנים בטבלה actor .
Ex-2	עדכנו את מדינת המוצא של כל השחקנים בטבלה actor שגילם מעל ל-35 לישראל.
Ex-3	עדכנו את האורך של כל סרטי האקשן בטבלה movie ל-120 דקות.
Ex-4	עדכנו את שנת ההפקה של כל סרטי הקומדיה בטבלה movie שמתחילים באות Z ששוויים או מעל ל-120 דקות, לשנת ההפקה - 1998.

מחיקת נתונים באמצעות שאלות



DELETE/example-truncate.sql
DELETE/example-delete.sql

מחיקת נתונים באמצעות שאילתות

בחלק זה נלמד כיצד למחוק נתונים קיימים במסד הנתונים בעזרת שאילתת DELETE. בסיום הנושא תוכלו לענות על השאלות הבאות:

```
-- *** Simple DELETE statement example ***
SET
    SQL_SAFE_UPDATES = 0;

DELETE FROM moviesdb.actor
WHERE
    actor.Name = "Yuli Boily";

SET
    SQL_SAFE_UPDATES = 1;
```



- כיצד נראית פקודת DELETE בסיסית?
- מהם השלבים למחיקת רשומות מטבלה?
- כיצד ניתן לסנן את הנתונים שרוצים למחוק באמצעות WHERE?
- מה הסיכונים במחיקת נתונים וכיצד להימנע מטעויות?
- כיצד ניתן למחוק את כל הרשומות בטבלה בעזרת DELETE?
- מה ההבדל בין DELETE לבין TRUNCATE?

מחיקת נתונים באמצעות שאילתות – תרגול

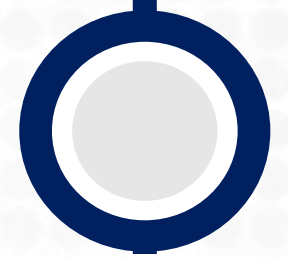
הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.

Movie	
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

תרגיל	תיאור המשימה
Ex-1	מחקו את אחד השחקנים בטבלה actor .
Ex-2	מחקו את כל השחקנים בטבלה actor שגילם מעל ל-40 במידה ומינם הוא זכר.
Ex-3	מחקו את כל סרטי האקשן בטבלה movie שאורכם 120 דקות וקיימת בשם שלהם האות A.
Ex-4	מחקו באמצעות שאילתה אחת את כל הנתונים משני הטבלאות.

מפתחות – Primary Key



מפתחות – Primary Key

בחלק זה נכיר את המפתח הראשי (**Primary Key**) במבנה טבלאי ונבין את חשיבותו בניהול נתונים. בסיום הנושא תוכלו לענות על השאלות הבאות:

```
-- Create a primary key on the Id column of the actor table
ALTER TABLE moviesdb.actor
ADD COLUMN `Id` INT AUTO_INCREMENT PRIMARY KEY FIRST;

CREATE INDEX idx_actor_id ON moviesdb.actor (Id);
```



- מהו **Primary Key** וכיצד הוא מוגדר בטבלה?
- מדוע חשוב להשתמש ב-**Primary Key**?
- כיצד **Primary Key** מבטיח ייחודיות של רשומות?
- מה ההבדל בין **Primary Key** לשדה רגיל בטבלה?
- לשם מה נשתמש ב-**ALTER**?
- כיצד ניתן להגדיר **Primary Key** בעת יצירת טבלה דרך הדיזיינר?
- כיצד ניתן להגדיר **Primary Key** בעת יצירת טבלה על ידי שימוש בשאילתה?

מפתחות – Primary Key – תרגול

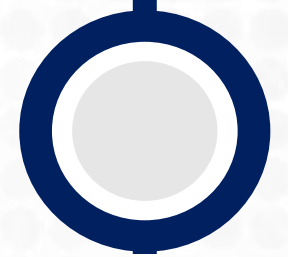
הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.

Movie	
Id	INT (PK)
Name	NVarchar(50)
Year	INT
Genre	NVarchar(50)
TimeInMinutes	INT

Actor	
Id	INT (PK)
Name	NVarchar(50)
Age	INT
Gender	NVarchar(50)
Country	NVarchar(50)

תרגיל	תיאור המשימה
Ex-1	וודאו ששני הטבלאות ריקות מנתונים בהתאם לתרגיל הקודם.
Ex-2	הוסיפו מפתח ייחודי (Primary Key) בשם Id לשני הטבלאות.
Ex-3	הוסיפו 5 רשומות לפחות לכל אחת מהטבלאות.
Ex-4	שלפו מטבלת הסרטים את הרשומה שיש לה את המזהה הייחודי 4.
Ex-5	עדכנו בטבלת הסרטים את שנת ההפקה של כל הסרטים שיש להם את המזהים הייחודיים 4 ו-5 לשנת ההפקה - 2005 (מבלי להשתמש ב-Safe Updates).
Ex-6	אתגר: מחקו מטבלת השחקנים מקסימום 2 רשומות שיש להן מזהה ייחודי זוגי (מבלי להשתמש ב-Safe Updates) בשאילתה אחת.

Foreign Key – מפתחות



מפתחות – Foreign Key

בחלק זה נכיר את המפתח הזר (**Foreign Key**) ונלמד כיצד הוא משמש לקישור בין טבלאות. בסיום הנושא תוכלו לענות על השאלות הבאות:

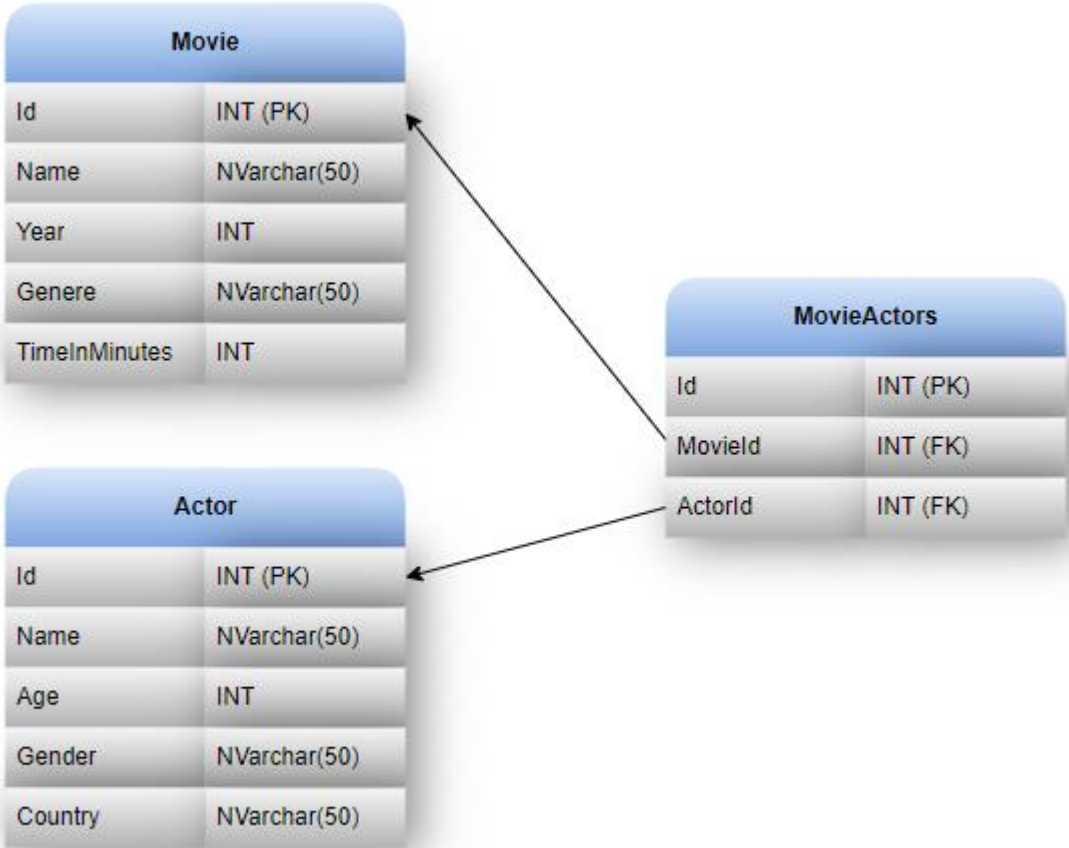
```
CREATE TABLE
  moviesdb.`movie-actors` (
    Id INT PRIMARY KEY AUTO_INCREMENT,
    MovieId INT,
    ActorId INT,
    FOREIGN KEY (MovieId) REFERENCES movie (Id),
    FOREIGN KEY (ActorId) REFERENCES actor (Id)
  );
```



- מהו **Foreign Key** וכיצד הוא מוגדר בטבלה?
- כיצד המפתח הזר מאפשר קישור בין טבלאות?
- מדוע חשוב להשתמש ב-**Primary Key**?
- כיצד **Foreign Key** תורם לשלמות הנתונים (**Data Integrity**)?
- מהו הקשר בין **Primary Key** ל-**Foreign Key**?
- כיצד ניתן להגדיר **Foreign Key** בעת יצירת טבלה דרך הדיזיינר?
- כיצד ניתן להגדיר **Foreign Key** בעת יצירת טבלה על ידי שימוש בשאילתה?

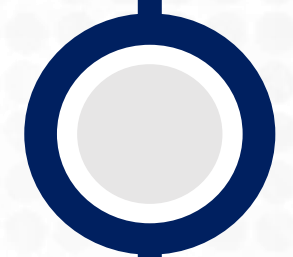
מפתחות – Foreign Key – תרגול

הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.



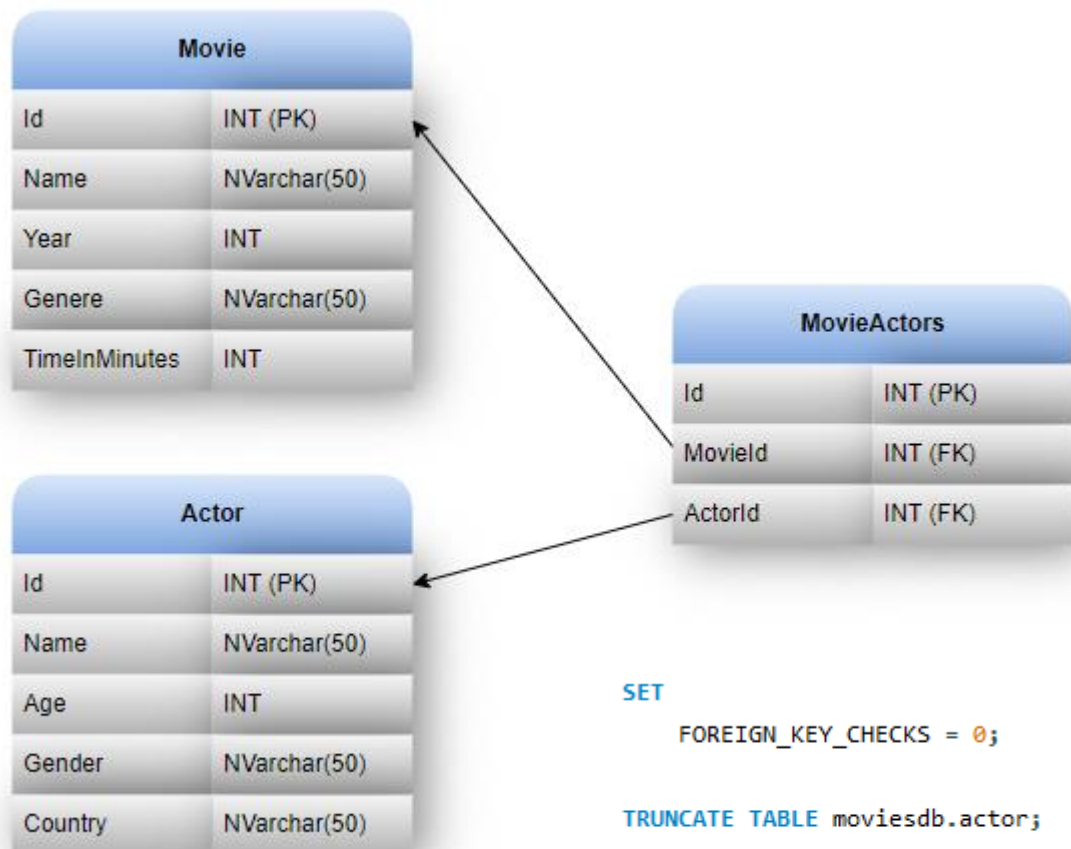
תרגיל	תיאור המשימה
Ex-1	טפלו בטבלת הסרטים כך שיהיו לכם 5 רשומות, ובטבלת השחקנים כך שיהיו לכם 5 רשומות.
Ex-2	צרו טבלה חדשה בשם 'movie-actors' בהתאם לדוגמה.
Ex-3	הוסיפו 5 רשומות לטבלה 'movie-actors' כך שכל שחקן יופיע בסרט אחר. שימו לב! אם לא תזינו id רלוונטי של שחקן ולאז סרט תתקבל שגיאה.

יחסים בין טבלאות ו-*JOIN*



יחסים בין טבלאות

בחלק זה נכיר את סוגי היחסים בין טבלאות במאגר נתונים טבלאי ונלמד כיצד הם מוגדרים ומשפיעים על שלמות המידע:



```

SET
    FOREIGN_KEY_CHECKS = 0;

TRUNCATE TABLE moviesdb.actor;

SET
    FOREIGN_KEY_CHECKS = 1;
  
```

- מה הם יחסים (Relations) בין טבלאות?
- איזה סוגי יחסים קיימים?
- מה המשמעות של One-To-One?
- מה המשמעות של One-To-Many?
- מה המשמעות של Many-To-Many?
- איך Foreign Key קשור לנושא?
- כיצד ניתן להגדיר יחסים בין טבלאות?
- כיצד יחסים תורמים לשלמות הנתונים ולשימוש יעיל במסד הנתונים?
- מה עלינו לעשות על מנת שנוכל למחוק נתונים מטבלה שמקושרת על ידי Foreign Key?
- לשם מה נשתמש ב-FOREIGN_KEY_CHECKS?

צירוף טבלאות - JOIN

בחלק זה נכיר את מנגנון ה-JOIN ב-SQL ונלמד כיצד לצרף טבלאות על מנת לשלוף נתונים בצורה משולבת. בסיום הנושא תוכלו לענות על השאלות הבאות:

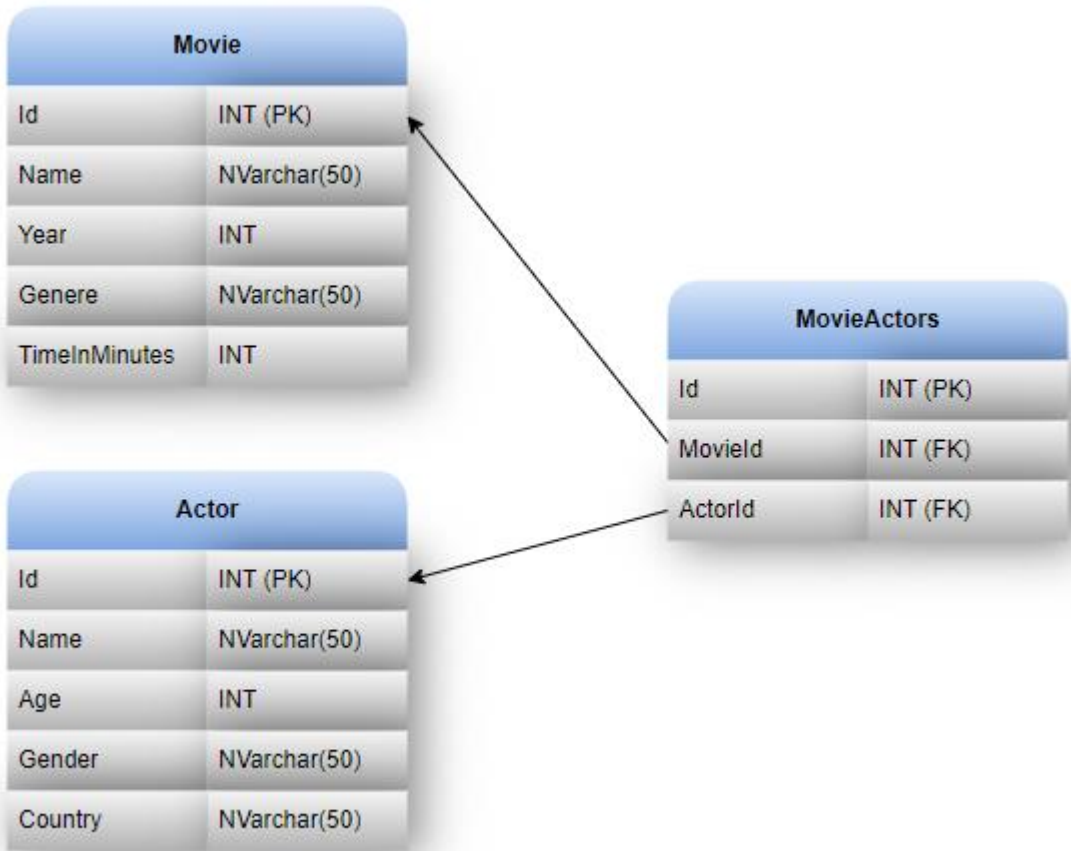
```
SELECT
    `movie-actors`.Id,
    actor.Name
FROM
    moviesdb.`movie-actors`
JOIN moviesdb.actor ON `movie-actors`.ActorId = actor.Id;
```

- מהו JOIN וכיצד הוא משמש לחיבור בין טבלאות?
- איזה סוגי JOIN קיימים?
- מה ההבדל בין סוגי ה-JOIN השונים?
- כיצד נראית שאילתה פשוטה עם JOIN?
- מתי נשתמש ב-INNER JOIN לעומת LEFT JOIN?
- כיצד המפתחות (Primary Key ל-Foreign Key) משחקים תפקיד בשימוש ב-JOIN?
- כיצד ניתן לשלב מספר טבלאות באמצעות שימוש ב-JOIN?
- לשם מה נשתמש ב-AS?



צירוף טבלאות – One To Many - תרגול

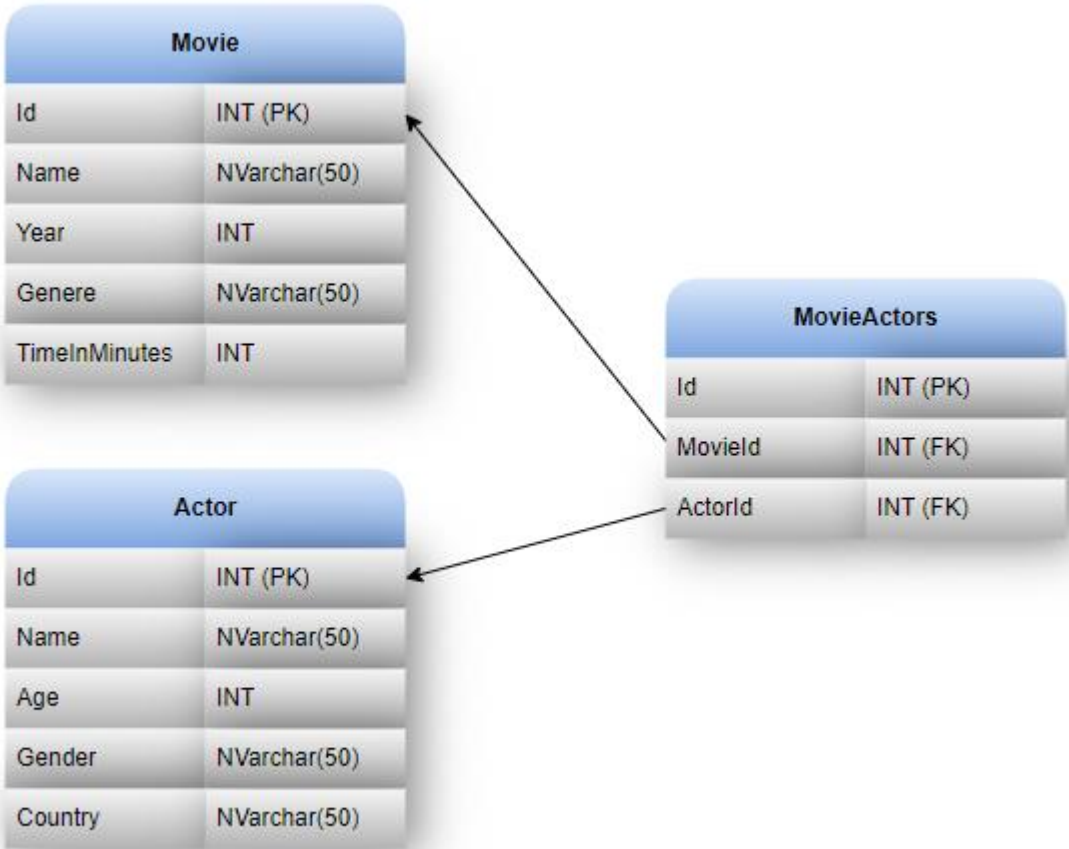
הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.



תרגיל	תיאור המשימה
Ex-1	מחקו את כל הרשומות מהטבלה <code>`movie-actors`</code> .
Ex-2	טפלו בטבלת הסרטים כך שיהיו לכם 5 רשומות, ובטבלת השחקנים כך שיהיו לכם 15 רשומות.
Ex-3	הוסיפו 5 רשומות לטבלה <code>`movie-actors`</code> כך שבסרט מסוים יוכלו להופיע מספר שחקנים, אך כל שחקן יופיע בסרט תנא בלבד (One-To-Many).
Ex-4	שלפו מהטבלה <code>`movie-actors`</code> באמצעות שימוש ב- JOIN את שמות כל השחקנים ואת שם הסרט שהם משחקים בו במידה ושנת ההפקה של הסרט מתחת ל-2012.
Ex-5	שלפו מהטבלה <code>`movie-actors`</code> באמצעות שימוש ב- JOIN את השמות והגילאים של כל השחקנים האמריקאים שמתחת לגיל 40 ומשחקים בסרט דרמה שאורכו מעל ל-100 דקות.

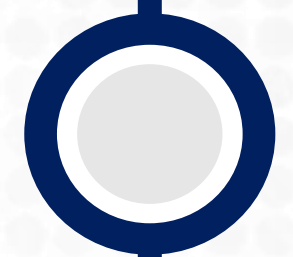
צירוף טבלאות – Many To Many - תרגול

הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.



תרגיל	תיאור המשימה
Ex-1	מחקו את כל הרשומות מהטבלה <code>movie-actors</code> .
Ex-2	טפלו בטבלת הסרטים כך שיהיו לכם 5 רשומות, ובטבלת השחקנים כך שיהיו לכם 15 רשומות.
Ex-3	הוסיפו 10 רשומות לטבלה <code>movie-actors</code> כך שבסרט מסוים יוכלו להופיע מספר שחקנים, ושחקנים מסוימים יופיעו במספר סרטים (Many-To-Many).
Ex-4	שלפו מהטבלה <code>movie-actors</code> באמצעות שימוש ב- JOIN את שמות כל השחקנים ואת שם הסרט שהם משחקים בו במידה ושנת ההפקה של הסרט מתחת ל-2020, ואורך הסרט שווה או מעל ל-120 דקות.
Ex-5	שלפו מהטבלה <code>movie-actors</code> באמצעות שימוש ב- JOIN את השמות והגילאים של כל השחקנים האמריקאים שמתחת לגיל 40 ומשחקים בסרט דרמה שאורכו מעל ל-100 דקות, שם הסרט שהם משחקים בו ואורך הסרט.

פונקציות אגרגציה ו-Group By



פונקציות אגרגציה ו-Group By

בחלק זה נכיר את פונקציות האגרגציה ב-SQL ונלמד כיצד להשתמש ב-Group By על מנת לארגן ולסכם נתונים. בסיום הנושא תוכלו לענות על השאלות הבאות:

```
SELECT
  year,
  COUNT(*) AS MovieCount,
  SUM(TimeInMinutes) AS TotalTime
FROM
  Movie
GROUP BY
  year;
```

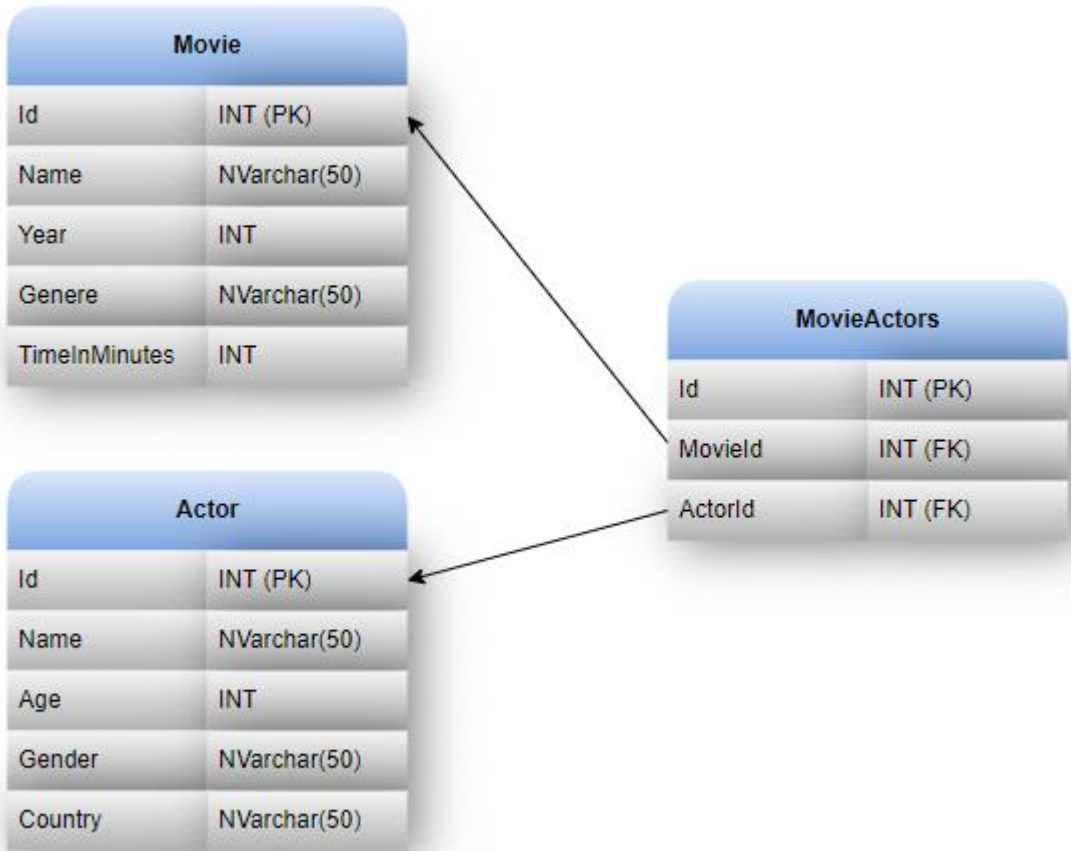


	year	MovieCount	TotalTime
▶	1967	7	875
	2005	9	1001
	2024	15	1583
	1980	1	125
	2019	6	720

- מהן פונקציות אגרגציה ב-SQL?
- כיצד משתמשים בפונקציות כמו COUNT, SUM, AVG, MIN, MAX?
- מהו Group By וכיצד הוא משמש לקיבוץ נתונים?
- כיצד משלבים Group By עם פונקציות אגרגציה?
- כיצד ניתן לקבץ נתונים לפי ערך מסוים בעזרת Group By?
- מה ההבדל בין שימוש ב-Group By לבין SELECT רגיל?
- כיצד ניתן לשלב בין Group By לבין תנאים עם HAVING?
- לשם מה משמש האופרטור DISTINCT?

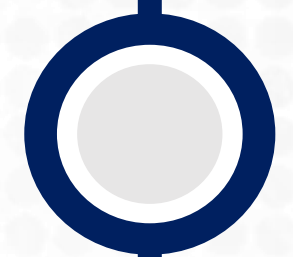
פונקציות אגרגציה ו-Group By

הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.



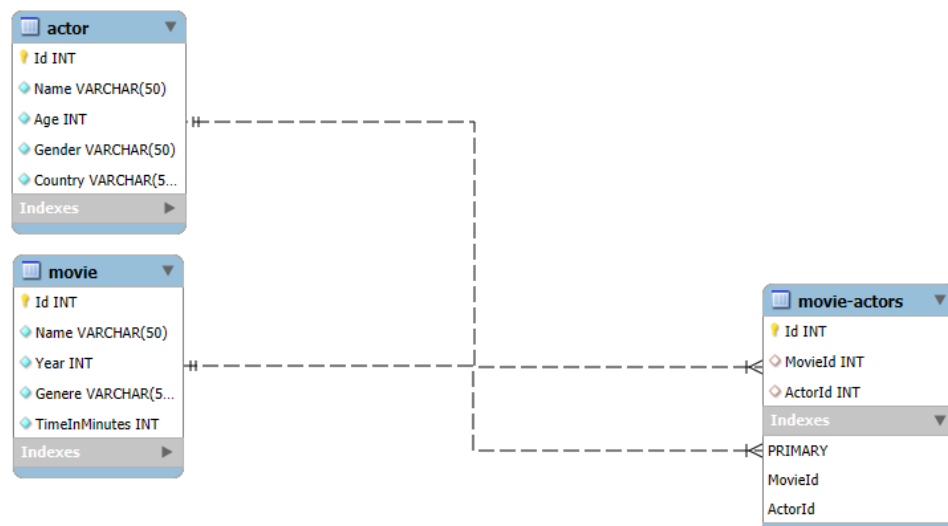
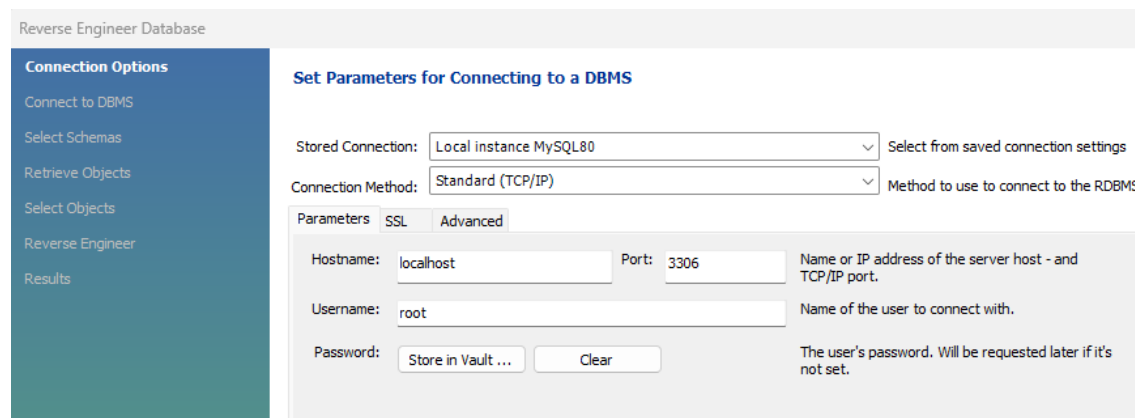
תרגיל	תיאור המשימה
Ex-1	כתבו שאילתה שתציג את מספר הסרטים שיצאו בכל שנה ואת הזמן הכולל שלהם בדקות.
Ex-2	כתבו שאילתה שתחשב את הגיל הממוצע של שחקנים לפי המדינה שלהם.
Ex-3	כתבו שאילתה שתציג את מספר הסרטים הקיימים בכל ז'אנר.
Ex-4	כתבו שאילתה שתציג את כל הסרטים – כל שם סרט יוצג עם מספר השחקנים שמופיעים בו.
Ex-5	כתבו שאילתה שתציג את מספר הסרטים ואת מספר השחקנים בכל ז'אנר כל עוד קיים יותר מסרט אחד באותו הז'אנר.

יצירת דיאגרמת **ER** ממסד נתונים



יצירת דיאגרמת ER ממסד נתונים

בחלק זה נלמד כיצד ניתן ליצור דיאגרמת ER (Entity-Relationship) ממסד נתונים על מנת להבין את המבנה והקשרים בין הטבלאות. בסיום הנושא תוכלו לענות על השאלות הבאות:



מהי דיאגרמת ER ומה מטרתה?

כיצד יוצרים דיאגרמת ER ב-MYSQL Workbench?

כיצד דיאגרמה זו משקפת את הקשרים (Relations) בין הטבלאות?

כיצד מוודאים שהמפתחות הראשיים והזרים משולבים נכון

בדיאגרמה?

כיצד ניתן לייצא את הדיאגרמה לתמונה או לקובץ PDF?

כיצד דיאגרמת ER עוזרת בתכנון ושיפור מסד הנתונים?

ייצוא מסד נתונים ל-SQL Script



ייצוא מסד נתונים ל-SQL Script

בחלק זה נלמד כיצד לייצא את מבנה הנתונים והמידע ממסד הנתונים שלנו לקובץ SQL Script לצורך גיבוי או העברה לשרת אחר. בסיום הנושא תוכלו לענות על השאלות הבאות:



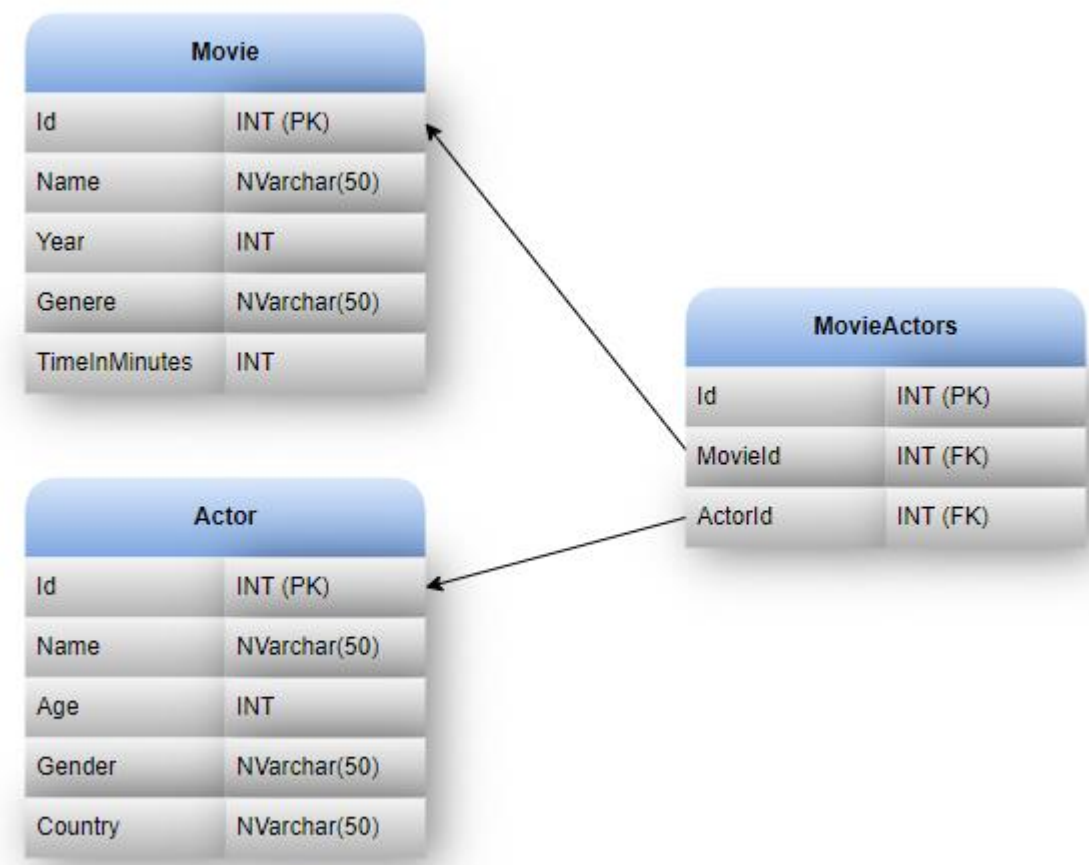
- כיצד ניתן לייצא מסד נתונים ל-SQL Script ב-MYSQL Workbench?
- מה ההבדל בין ייצוא מבנה הטבלאות בלבד לבין ייצוא אשר כולל נתונים?
- כיצד מוודאים ש-SQL Script מכיל את כל השאילתות הנדרשות לשחזור מסד הנתונים?
- מה חשוב לבדוק לפני ייצוא של מסד נתונים גדול?
- כיצד ניתן לייבא מסד נתונים מקובץ SQL Script חזרה ל-MYSQL?
- כיצד פקודת `mysqldump` משמשת לייצוא מסד נתונים משורת הפקודה?

```
mysqldump -u root -p --databases moviesdb > moviesdb-dump.sql
```

```
EXPORT/moviesdb.sql
EXPORT/moviesdb-dump.sql
```

ייצוא מסד נתונים ל-SQL Script

הפעילו את **MySQL Workbench**, פיתחו את הסכמה **moviesdb** ופתרו את התרגילים לפי הסדר.



תרגיל	תיאור המשימה
Ex-1	ייצאו את מסד הנתונים שלכם כולל הנתונים למסמך SQL .
Ex-2	וודאו שהמסמך תקין ומכיל את כל השאילתות הדרושות ליצירת מסד הנתונים.
Ex-3	הריצו את המסמך ובידקו שאין שגיאות.
Ex-4	ייצאו שוב את מסד הנתונים אך הפעם באמצעות הפקודה mysqldump .



זמן שאלות

בדיקת רמת ידע



סוף

