



קריית החינוך
השש שנתית
פארק המדע
עמיד של חדשנות



תכנות באינטרנט

SQL in Visual Studio

גלעד מרקמן

קריית החינוך פארק המדע,
נס ציונה

מסד נתונים הוא אוסף של מאגרי מידע בנושא מסויים.

בכדי לנהל מסד נתונים משתמשים במערכת לניהול מסד נתונים
(שרת מסד נתונים). Data Base Management System.

יש מערכות לניהול בסיסי נתונים מסחריות רבות, כגון: MySQL, MS-
SQL, Oracle ועוד'.

המודל הטבלאי/היחסי (Relational Model)

- טבלה - ניהול הנתונים נעשה באמצעות טבלאות.
- תכונה – כל עמודה בטבלה מייצגת תכונה (field).
- יישות (רשומה) – כל שורה בטבלה מייצגת יישות (record).
- מפתח – תכונה שמשתתף לזיהוי חח"ע של יישות (רשומה) בטבלה (key).

	תכונה	תכונה	תכונה	תכונה	תכונה
	עיר	רחוב	כיתה	שכבה	שם
ישות	תל-אביב	נורדאו 5	1	יב	יוסי כהן
ישות	רמת-גן	הרצל 14	1	יב	זהבה לוי
ישות	תל-אביב	ויצמן 94	2	יא	חיים מצליח
ישות	תל-אביב	ז'בוטינסקי 9	1	י	זהר אוריין
ישות	גבעתיים	חרל"פ 19	2	י	חיים ביטון
ישות	תל-אביב	זנד 9	1	ט	יונה קדוש


ערך

- SQL - Structured Query Language
- שפה שנועדה לנהל בסיסי נתונים, לאחזר נתונים, לשנות ולשמור נתונים.
- שפת SQL הינה שפה סטנדרטית המשמשת שרתי בסיסי נתונים רבים, והינה השפה הנפוצה ביותר לניהול בסיסי נתונים.
- לצורך שימוש בבסיסי נתונים אנו זקוקים לשרת Data Base אשר אנו נפנה אליו בשאלות SQL והוא יבצע אותם על בסיס הנתונים הנמצא אצלו.
- ישנם שרתים רבים, שמרביתם יודעים לעבוד עם שפת SQL, כדוגמת: MySQL, Microsoft SQL, Oracle, PostgreSQL ועוד'.

- בהרצאה זו נלמד כיצד להשתמש ב SQL Server Data Tools for Visual Studio.
- כיצד ליצור בסיס נתונים בפרויקט שלנו.
- כיצד ליצור טבלה ראשונה ולקבוע את המאפיינים של הטבלה.
- כיצד להזין נתונים לטבלה באמצעות שאילתה של SQL.
- כיצד לשלוף (לאחזר) נתונים מהטבלה באמצעות שאילתות.
- כיצד למחוק נתונים, להוסיף נתונים ולשנות נתונים באמצעות שאילתות.
- התלמידים נדרשים לטפל בבסיס נתונים הכולל טבלה אחת בלבד.

- להפעיל את Visual Studio Installer , וללחוץ על Modify.
- לסמן "V" על Data storage and processing , וללחוץ על Install.


Installed Available




Visual Studio Professional 2019
16.8.4
Professional IDE best suited to small teams
[Release notes](#)

Modify
Launch
More ▾

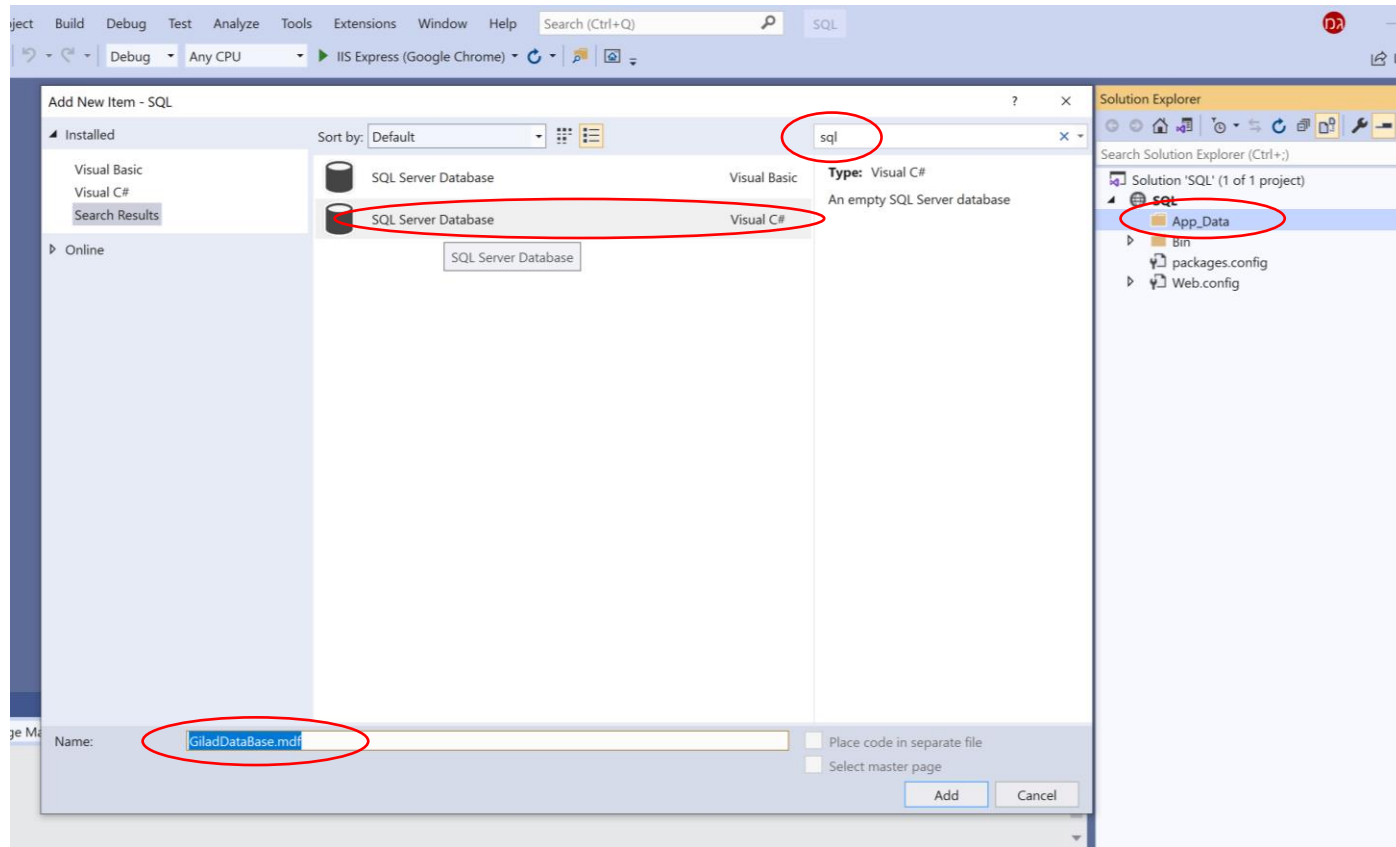
Other Toolsets (6)



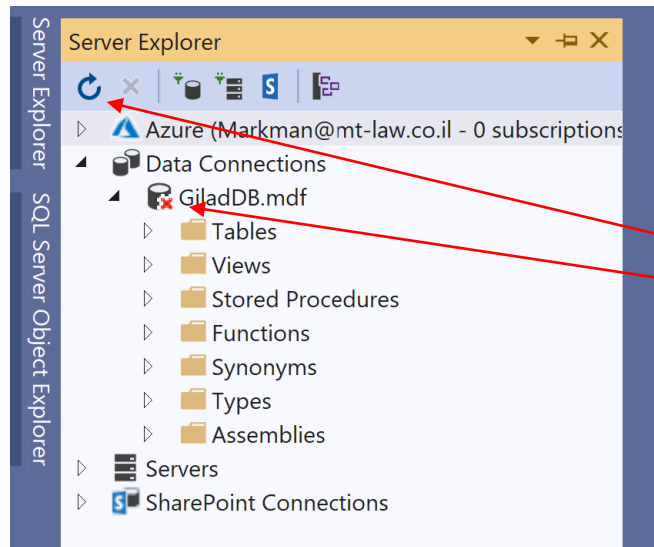
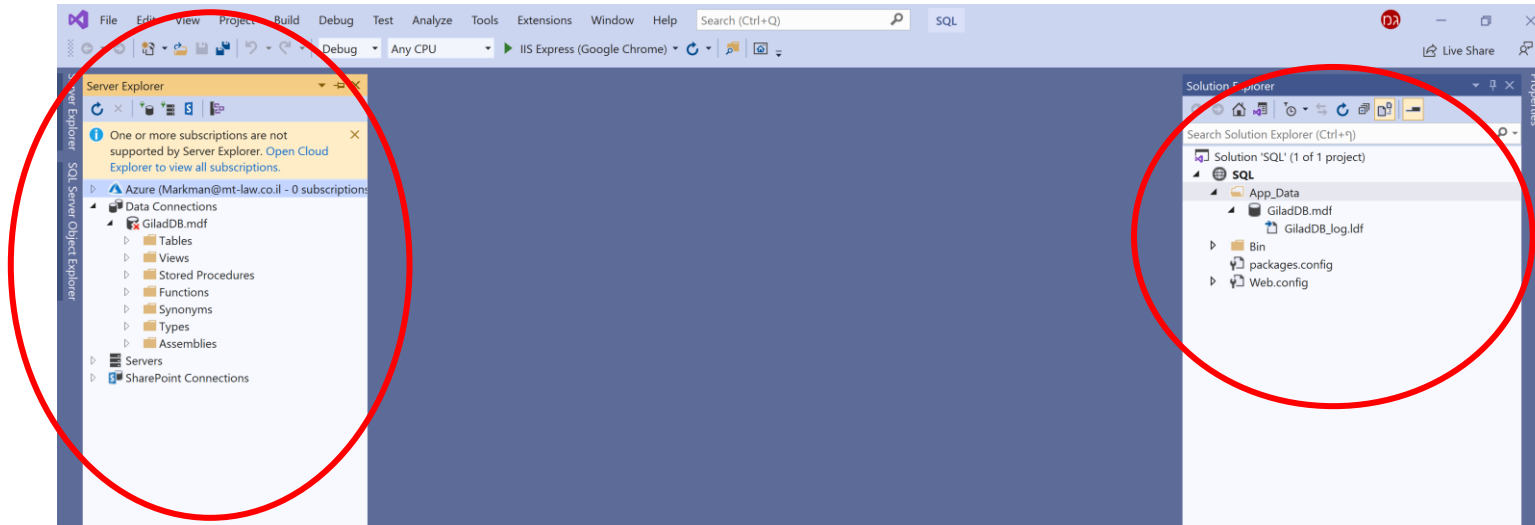
Data storage and processing
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.



Data science and analytical applications
Languages and tooling for creating data science applications, including Python and F#.



- יצירת ספרייה App_Data
- הוספת פריט חדש לספרייה.
- בחירת SQL Server Database C#
- מתן שם לבסיס הנתונים יש לשמור על סיומת .mdf



- אם התפריט לא מופיע יש לפעול באחת משני הדרכים:

- בתפריט ללחוץ View/Server Explorer

- קיצור מקשים Ctrl + Alt + s

- גם ב Solution Explorer ניתן לראות את בסיס הנתונים.

- אם בסיס הנתונים מסומן באיקס אדום יש ללחוץ refresh ב server Explorer.

Create Table

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

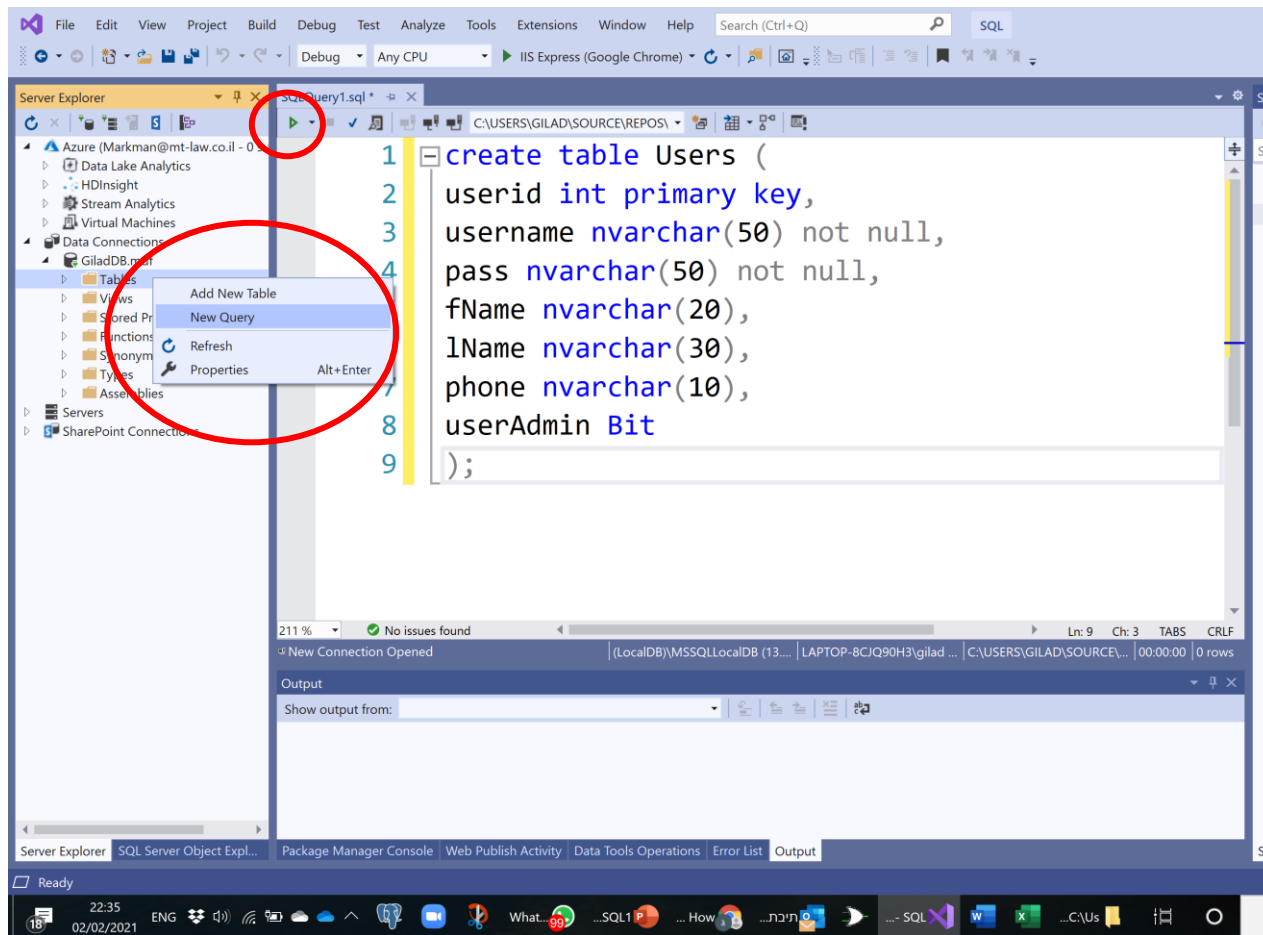
- בסיס הנתונים מבוסס על טבלאות – בדומה לטבלאות אקסל.
- הטבלה בנויה מעמודות Columns. לכל עמודה יש להגדיר את שם העמודה וסוג טיפוס הנתונים.
 - INT, NVARCHAR(50), BIT, FLOAT, DATE, DATETIME
- בנוסף, יש להגדיר לכל טבלה מפתח ראשי – primary Key.

יצירת טבלת משתמשים

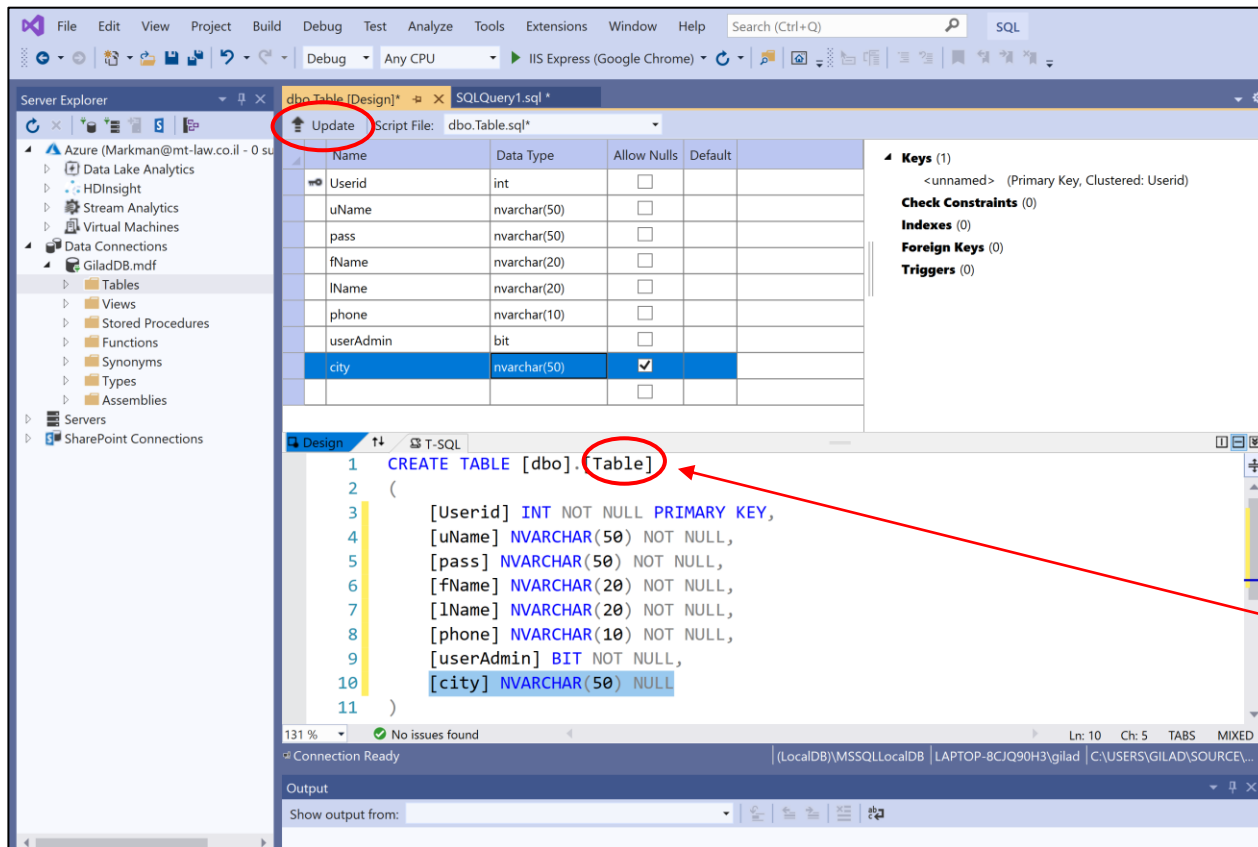
```
CREATE TABLE tblUsers  
(  
    UserId INT NOT NULL IDENTITY(1, 1) ,  
    firstName NVARCHAR(50) NOT NULL,  
    lastName NVARCHAR(50) NOT NULL,  
    admin BIT NOT NULL DEFAULT 0,  
    country NVARCHAR(50) NULL,  
    tel NCHAR(10) NULL,  
    username NVARCHAR(50) NOT NULL UNIQUE,  
    password NVARCHAR(50) NOT NULL,  
    PRIMARY KEY ([UserId]),  
    address NVARCHAR(50)  
)
```

- דוגמה ליצירת טבלת משתמשים.
- יש לשים לב למאפיינים הבאים:
- NOT NULL – חייבים להכניס נתונים לשדה.
- UNIQUE – ייחודי. לא ניתן שיהיו שתי שורות עם ערך זהה בשדה זה.
- Primary Key הגדרת שדה מפתח ראשי. לא מאפשר כפילויות בשדה זה.
- ניתן להגדיר שדה ראשי גם בשורת ההגדרה של השדה.

יצירת טבלה באמצעות פקודת Create Table



- קליק ימני על Tables ובחירה של New Query – שאילתה חדשה.
- כתיבת פקודת Create Table
- לחיצה על לחצן בצע (חץ ירוק).



- קליק ימני על Tables ובחירת Add New Table.
- הזנת הנתונים בטבלה תיצור אוטמטית את פקודת ה Create Table.
- לחיצה על update בסיום כדי ליצור או לעדכן את הטבלה.
- יש לשנות את שם הטבלה.

- אם חלון מאפיינים סגור – ללחוץ F4. למשל מספור אוטומטי של מפתח ראשי.

The screenshot shows the SQL Server Enterprise Designer interface. The 'Table Designer' window is open, displaying the 'Keys' tab. The 'Userid' column is highlighted as the primary key and has a unique constraint. The 'Columns' tab shows the following table structure:

Name	Data Type	Allow Nulls	Default
Userid	int	<input type="checkbox"/>	
uName	nvarchar(50)	<input type="checkbox"/>	
pass	nvarchar(50)	<input type="checkbox"/>	
fName	nvarchar(20)	<input type="checkbox"/>	
lName	nvarchar(20)	<input type="checkbox"/>	
phone	nvarchar(10)	<input type="checkbox"/>	
userAdmin	bit	<input type="checkbox"/>	
city	nvarchar(50)	<input checked="" type="checkbox"/>	

The 'Columns' tab also shows the following table structure:

Name	Data Type	Allow Nulls	Default
[Userid]	INT NOT NULL		
[uName]	NVARCHAR(50) NOT NULL		
[pass]	NVARCHAR(50) NOT NULL		
[fName]	NVARCHAR(20) NOT NULL		
[lName]	NVARCHAR(20) NOT NULL		
[phone]	NVARCHAR(10) NOT NULL		
[userAdmin]	BIT NOT NULL		
[city]	NVARCHAR(50) NULL		

The 'Columns' tab also shows the following table structure:

Name	Data Type	Allow Nulls	Default
[Userid]	INT NOT NULL		
[uName]	NVARCHAR(50) NOT NULL		
[pass]	NVARCHAR(50) NOT NULL		
[fName]	NVARCHAR(20) NOT NULL		
[lName]	NVARCHAR(20) NOT NULL		
[phone]	NVARCHAR(10) NOT NULL		
[userAdmin]	BIT NOT NULL		
[city]	NVARCHAR(50) NULL		

The 'Columns' tab also shows the following table structure:

Name	Data Type	Allow Nulls	Default
[Userid]	INT NOT NULL		
[uName]	NVARCHAR(50) NOT NULL		
[pass]	NVARCHAR(50) NOT NULL		
[fName]	NVARCHAR(20) NOT NULL		
[lName]	NVARCHAR(20) NOT NULL		
[phone]	NVARCHAR(10) NOT NULL		
[userAdmin]	BIT NOT NULL		
[city]	NVARCHAR(50) NULL		

The 'Columns' tab also shows the following table structure:

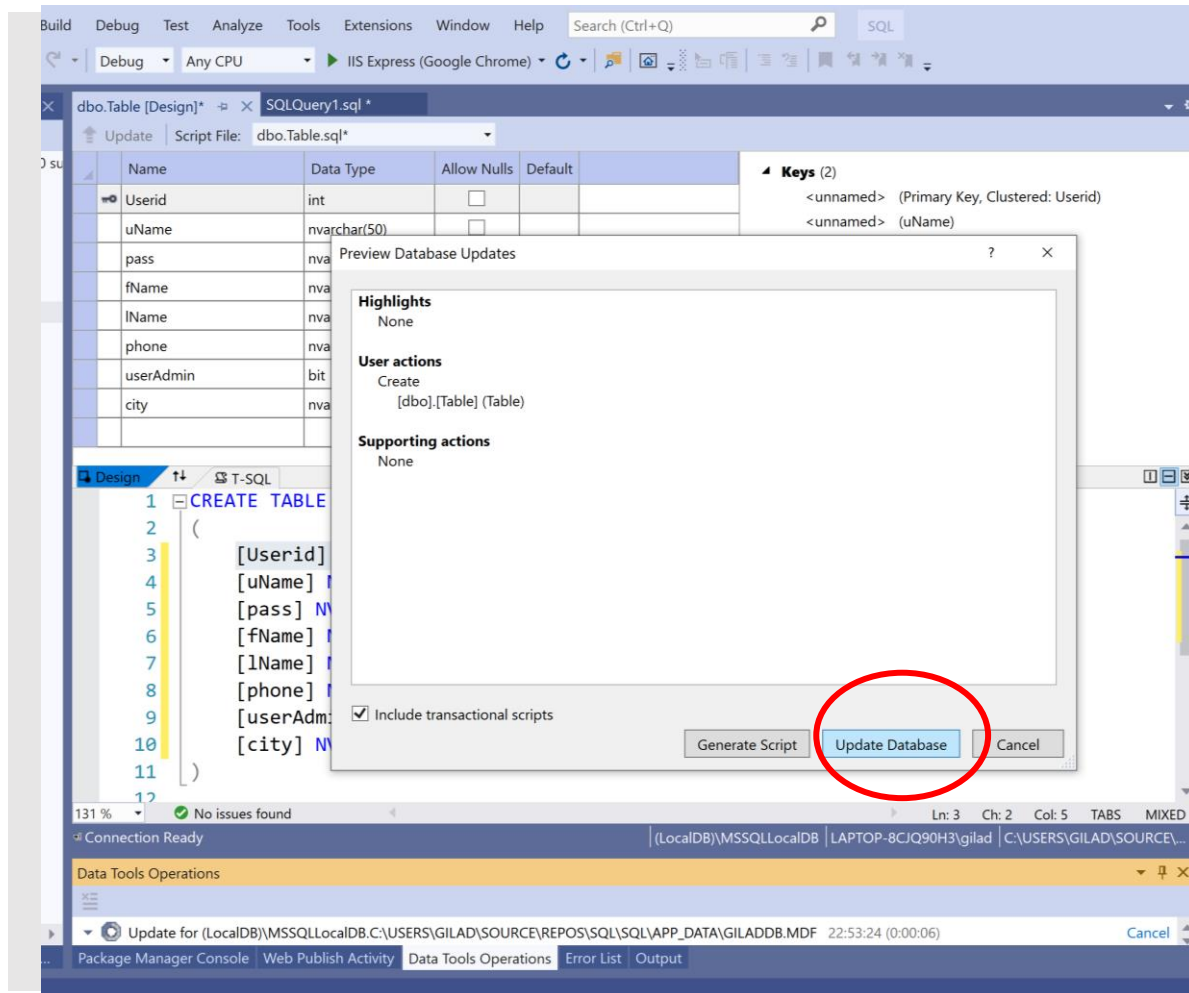
Name	Data Type	Allow Nulls	Default
[Userid]	INT NOT NULL		
[uName]	NVARCHAR(50) NOT NULL		
[pass]	NVARCHAR(50) NOT NULL		
[fName]	NVARCHAR(20) NOT NULL		
[lName]	NVARCHAR(20) NOT NULL		
[phone]	NVARCHAR(10) NOT NULL		
[userAdmin]	BIT NOT NULL		
[city]	NVARCHAR(50) NULL		

The screenshot shows the 'Properties' window for the 'Userid' column. The 'Identity Specification' section is expanded, showing the following properties:

Property	Value
Identity Specification	True
(Is Identity)	True
Identity Increment	1
Identity Seed	1
Is Column Set	False
Is File Stream	False
Is ROWGUID Column	False
Is Sparse	False

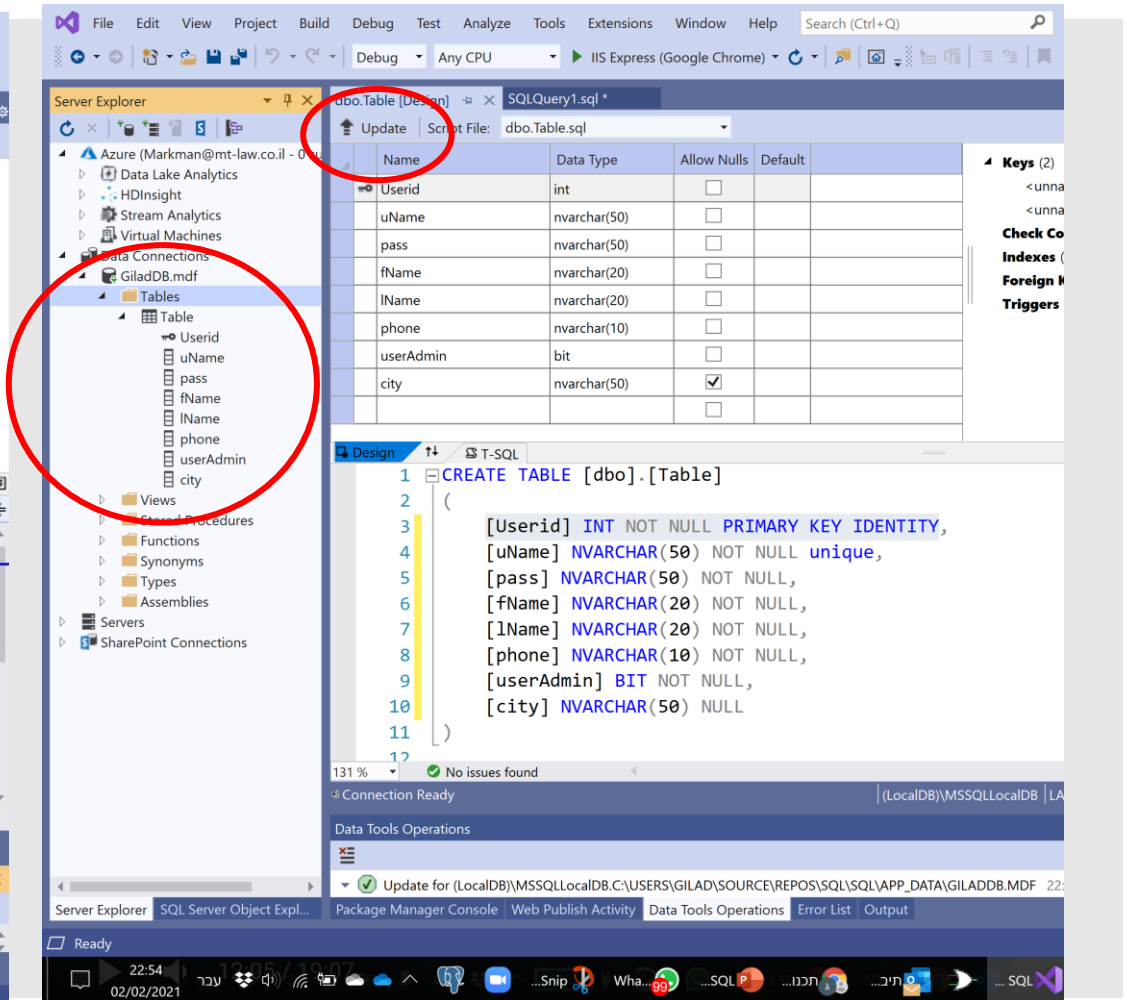
The 'Identity Specification' section is expanded to show properties specific to identity columns.

יצירת הטבלה בבסיס הנתונים - Update



Update Database dialog box:

- Generate Script
- Update Database
- Cancel



T-SQL Script:

```

1 CREATE TABLE [dbo].[Table]
2 (
3     [Userid] INT NOT NULL PRIMARY KEY IDENTITY,
4     [uName] NVARCHAR(50) NOT NULL unique,
5     [pass] NVARCHAR(50) NOT NULL,
6     [fName] NVARCHAR(20) NOT NULL,
7     [lName] NVARCHAR(20) NOT NULL,
8     [phone] NVARCHAR(10) NOT NULL,
9     [userAdmin] BIT NOT NULL,
10    [city] NVARCHAR(50) NULL
11 )
12
13

```

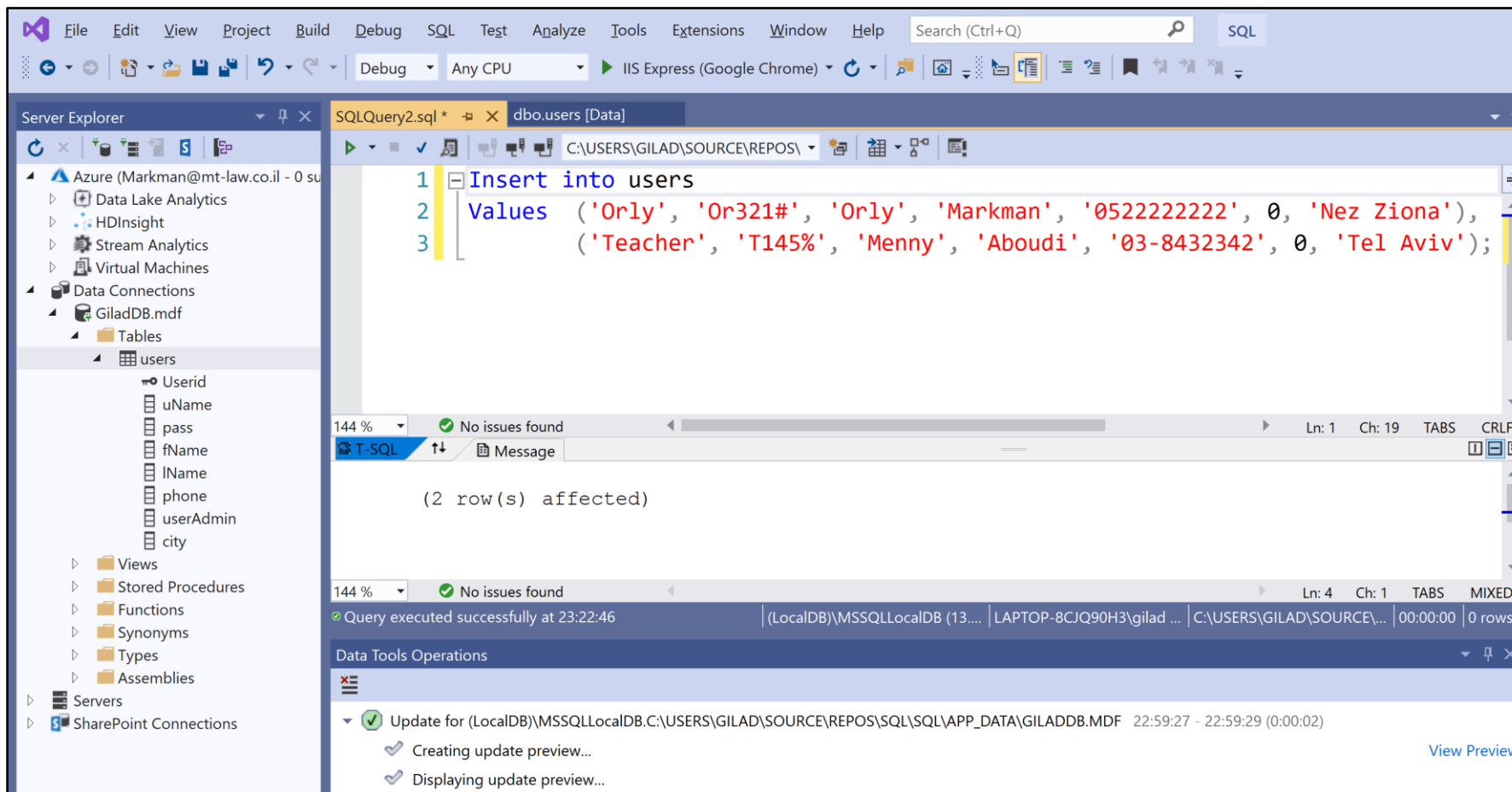
```
INSERT INTO table_name (column1, column2, ...)  
VALUES (value1, value2, ...);
```

אם מכניסים את כל השדות

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO tblUsers (firstName, lastName, username, password, admin,  
birthday)  
VALUES ('Haim', 'Cohen', 'HaCo', 'Haim1', 'false', '12/31/2020')
```

הוספת נתונים על ידי שאילתה



The screenshot displays the SQL Server Enterprise Edition interface. The main window shows a T-SQL query being executed in the 'dbo.users [Data]' context. The query is as follows:

```

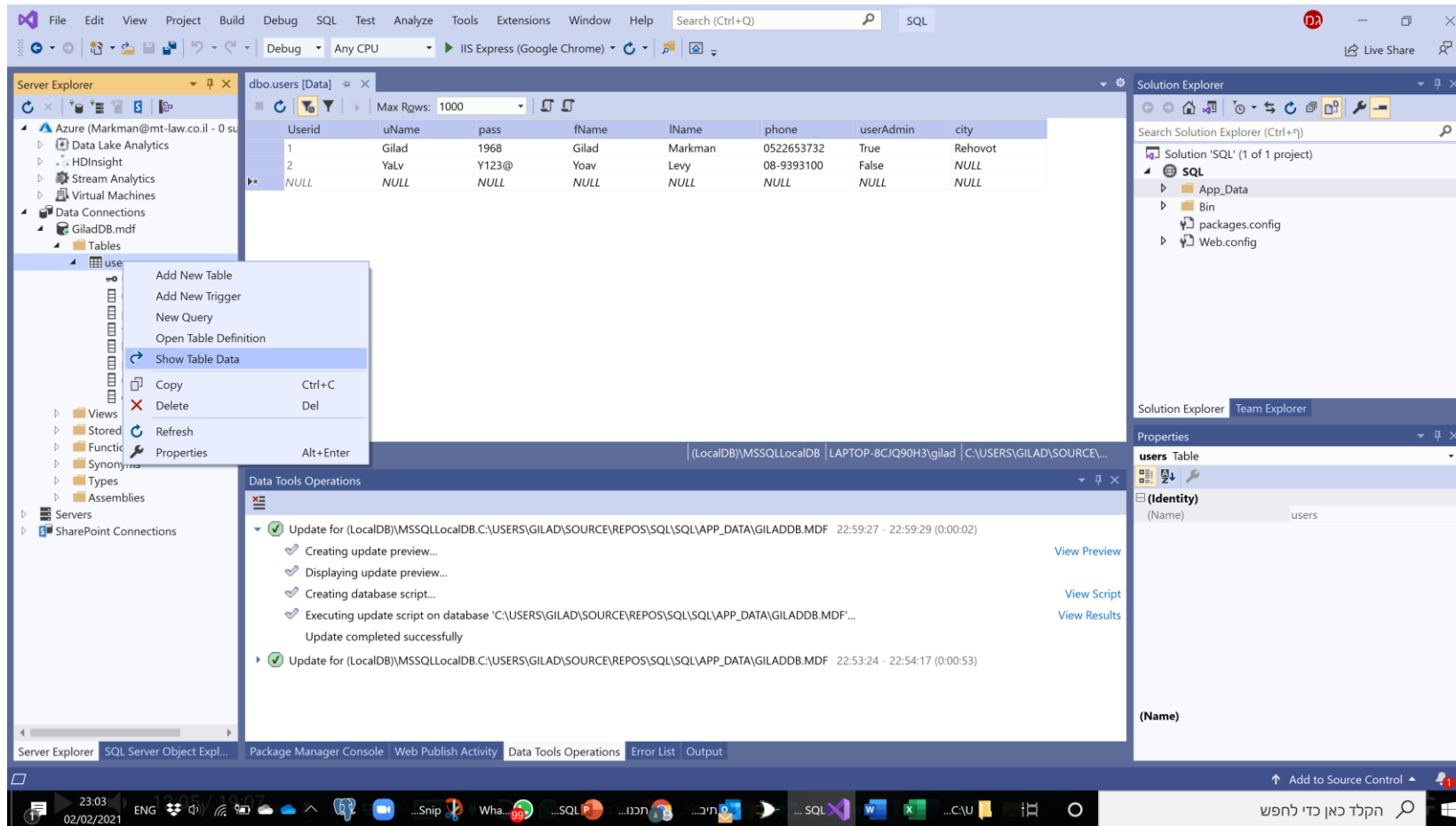
1 Insert into users
2 Values ('Orly', 'Or321#', 'Orly', 'Markman', '052222222', 0, 'Nez Ziona'),
3        ('Teacher', 'T145%', 'Menny', 'Aboudi', '03-8432342', 0, 'Tel Aviv');

```

The execution results pane shows the message: "(2 row(s) affected)". Below this, the Data Tools Operations pane indicates that the update was successful and shows the following operations:

- Update for (LocalDB)\MSSQLLocalDB.C:\USERS\GILAD\SOURCE\REPOS\SQL\SQL_APP_DATA\GILADDB.MDF 22:59:27 - 22:59:29 (0:00:02)
 - Creating update preview...
 - Displaying update preview...

The Server Explorer on the left shows the database structure for 'GiladDB.mdf', including a 'users' table with columns: Userid, uName, pass, fName, lName, phone, userAdmin, and city.



The screenshot displays the Visual Studio IDE with a SQL query executed in the SQL Server Enterprise Edition. The query results show a table with columns: Userid, uName, pass, fName, lName, phone, userAdmin, and city. The data rows are:

Userid	uName	pass	fName	lName	phone	userAdmin	city
1	Gilad	1968	Gilad	Markman	0522653732	True	Rehovot
2	YaLv	Y123@	Yoav	Levy	08-9393100	False	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The Data Tools Operations window shows the following update process:

- Update for (LocalDB)\MSSQLLocalDB\C:\USERS\GILAD\SOURCE\REPOS\SQL\SQL\APP_DATA\GILADDB.MDF 22:59:27 - 22:59:29 (0:00:02)
 - Creating update preview...
 - Displaying update preview...
 - Creating database script...
 - Executing update script on database 'C:\USERS\GILAD\SOURCE\REPOS\SQL\SQL\APP_DATA\GILADDB.MDF'...
 - Update completed successfully
- Update for (LocalDB)\MSSQLLocalDB\C:\USERS\GILAD\SOURCE\REPOS\SQL\SQL\APP_DATA\GILADDB.MDF 22:53:24 - 22:54:17 (0:00:53)

- קליק ימני על הטבלה.
- Show Table Data
- הזנת נתונים בתוך הטבלה.
- במעבר שורה הנתונים מתעדכנים ישירות ב Database

SELECT

```
SELECT column1, column2 / *  
FROM tableName  
WHERE condition  
ORDER BY column1 ASC, column2  
DESC;
```

```
SELECT *  
FROM tblUsers  
WHERE firstName LIKE 'g%'  
ORDER BY firstName ASC, lastName  
ASC;
```

- שאילתה לשליפת נתונים מהטבלה.
- הפקודות SELECT, FORM הכרחיות. הפקודות לאחר מכן אינן חובה.
- WHERE – תנאי. השרת עובר על הטבלה שורה שורה ומחזיר רק את השורות שעומדות בתנאי.
- ORDER – השורות מוחזרות בסדר שנקבע. ASC – ascending סדר עולה. DESC – decending סדר יורד.

```
SELECT *  
FROM tblUsers  
WHERE firstName = 'Gilad' COLLATE SQL_Latin1_General_CP1_CS_AS  
ORDER BY firstName ASC, lastName ASC;
```

Case sensitive •

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

מחזירה מספר רשומות (ללא הרשומות עצמן).

SELECT DISTINCT

```
SELECT DISTINCT column1, column2 / *  
FROM tableName  
WHERE condition  
ORDER BY column1 ASC, column2 DESC;
```

אם יש ערכים כפולים תחזיר רק ערכים ייחודיים.

INSERT INTO

```
INSERT INTO table_name (column1, column2, ...)  
VALUES (value1, value2, ...);
```

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

אם מכניסים את כל השדות

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

```
UPDATE tblUsers  
SET firstName = 'gilad'  
WHERE firstName='GILAD' AND lastName= 'Markman';
```

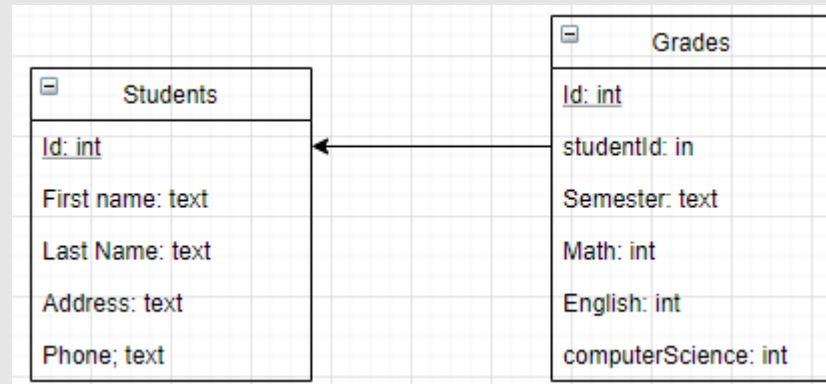
```
UPDATE tblUsers  
SET firstName = firstName + '_admin'  
WHERE admin='true';
```

DELETE

```
DELETE FROM table_name  
WHERE condition;
```

```
DELETE From tblUsers  
WHERE USERID=10;
```

ניתן לבצע שאילתות על מספר טבלאות בהתאם "ליחס" ביניהם, שיוגדר באמצעות השאילתה.
נניח, לדוגמה, שיש לנו טבלת תלמידים וטבלת ציונים המוגדרות כך:



• הנתונים של שתי הטבלאות הם הנתונים הבאים:

• Students

	Id	FirstName	LastName	Address	Phone
1	1	Gilad	Markman	Rehovot	0522222222
2	5	Lior	Cohen	Tel Aviv	0523333333

• Grades

	Id	Math	English	Computers_Science	Behavior	Semester	userId
1	1	90	100	95	B	2020 A	1
2	2	80	75	80	A	2020 B	1
3	3	90	100	100	A	2020 A	5
4	4	100	100	100	B	2020 B	5

• הקשר ("היחס") בין הטבלאות נעשה באמצעות השדות Students.ID ו Grades.userId.

```
SELECT *  
FROM Students, Grades;
```

• הפקודה הבאה יוצרת מכפלה קרטזית בין הטבלאות:

	Id	FirstName	LastName	Address	Phone	Id	Math	English	Computers_Science	Behavior	Semester	userId
1	1	Gilad	Markman	Rehovot	0522222222	1	90	100	95	B	2020 A	1
2	1	Gilad	Markman	Rehovot	0522222222	2	80	75	80	A	2020 B	1
3	1	Gilad	Markman	Rehovot	0522222222	3	90	100	100	A	2020 A	5
4	1	Gilad	Markman	Rehovot	0522222222	4	100	100	100	B	2020 B	5
5	5	Lior	Cohen	Tel Aviv	0523333333	1	90	100	95	B	2020 A	1
6	5	Lior	Cohen	Tel Aviv	0523333333	2	80	75	80	A	2020 B	1
7	5	Lior	Cohen	Tel Aviv	0523333333	3	90	100	100	A	2020 A	5
8	5	Lior	Cohen	Tel Aviv	0523333333	4	100	100	100	B	2020 B	5

• התוצאה היא:

- כל שורה של טבלת הסטודנטים מחוברת לכל השורות של טבלת הציונים.
- סך הכל מספר השורות הוא המכפלה של מספר השורות בטבלה אחת X מספר השורות בטבלה השניה.

- קישור בין טבלאות על בסיס יחס מסויים יעשה באמצעות תנאי:

```
SELECT *  
FROM Students, Grades  
WHERE Students.Id = Grades.UserId;
```

- התוצאה:

Id	FirstName	LastName	Address	Phone	Id	Math	English	Computers_Science	Behavior	Semester	userId
1	Gilad	Markman	Rehovot	0522222222	1	90	100	95	B	2020 A	1
1	Gilad	Markman	Rehovot	0522222222	2	80	75	80	A	2020 B	1
5	Lior	Cohen	Tel Aviv	0523333333	3	90	100	100	A	2020 A	5
5	Lior	Cohen	Tel Aviv	0523333333	4	100	100	100	B	2020 B	5

- חישובים על טבלאות מקושרות:

```
SELECT LastName, Avg(Math) AS Math, Avg (English) AS English
FROM Students, Grades
WHERE Students.Id = Grades.UserId
GROUP BY LastName;
```

Id	FirstName	LastName	Address	Phone	Id	Math	English	Computers_Science	Behavior	Semester	userId
1	Gilad	Markman	Rehovot	0522222222	1	90	100	95	B	2020 A	1
1	Gilad	Markman	Rehovot	0522222222	2	80	75	80	A	2020 B	1
5	Lior	Cohen	Tel Aviv	0523333333	3	90	100	100	A	2020 A	5
5	Lior	Cohen	Tel Aviv	0523333333	4	100	100	100	B	2020 B	5

LastName	Math	English
Cohen	95	100
Markman	85	87

- התוצאה:

- למדנו בשיעור זה כיצד ליצור בסיס נתונים בספריה Data_App בפרויקט שלנו.
- למדנו כיצד ליצור טבלה ולהגדיר את העמודות (המאפיינים) שלה.
- הדגמנו כיצד להזין נתונים לתוך הטבלה.
- ולמדנו את השאילתות הבסיסיות בשפת SQL.