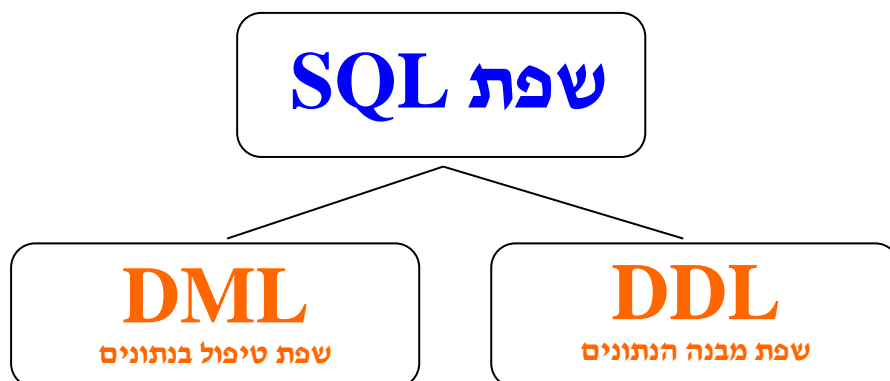


תרגול מס' 2

שפת SQL והמודל הטבלאי

בתרגול הקודם הראנו את היתרונות של המודל הטבלאי למול המודלים האחרים, כעת נלמד את השפה שאיתה נוכל לכתוב שאילתות לשליפת נתונים מבסיס הנתונים. השפה איתה נעבוד נקראת שפת SQL המתחלקת ל 2 תתי שפות.



• **DML - שפת טיפול בנתונים** (Data Manipulation Language) - שפה המשמשת לטיפול בנתונים.

- פקודת SELECT - המשפט המוכר ביותר בקטגוריה זו, מחזיר נתונים מתוך טבלה/טבלאות
- פקודת INSERT מוסיפה שורות חדשות לטבלה קיימת
- פקודת UPDATE מעדכנת נתונים בשורות הקיימות בטבלה
- פקודת DELETE מוחקת שורות נתונים מטבלה

• **DDL - שפת הגדרת נתונים** (Data Definition Language) - שפה המשמשת לטיפול במבנה הנתונים (הוספת ומחיקת טבלאות, עמודות ואינדקסים), ולקביעת הרשאות גישה לרכיבי בסיס הנתונים.

- פקודת CREATE יוצרת פריט מבנה חדש - טבלה, עמודה, אינדקס וכד'
- פקודת ALTER משנה את תכונותיו של פריט קיים
- פקודת DROP מוחקת פריט קיים
- פקודת GRANT מעניקה זכות גישה למשתמש או לקבוצת משתמשים
- פקודת REVOKE מונעת זכות גישה ממשתמש או מקבוצה

מבנה פקודת שליפה בסיסית

מבנה של פקודה לשליפת נתונים שבה נשתמש לאורך כל הקורס:

SELECT *
FROM Students

Students		
ID	Name	Age
1	Roei	10
2	Omer	20
3	Gili	30

בפקודת שליפה זו אנו מגדירים את 2 פקודות החובה שכל שאילתת שליפת נתונים ב SQL חייבת להכיל.

לאור העובדה שלפעמים נרצה להוסיף תנאים לשליפה נוכל להשתמש בפקודה הבאה:

SELECT *
FROM Students
WHERE Students.id = 3344

שמות כל העמודות שנרצה לשלוף מכל הטבלאות המשתתפות בפעולת השליפה

שמות כל הטבלאות אשר משתתפות בפעולת השליפה

כל התנאים שנרצה להתנות על המידע מהטבלאות המשתתפות בשליפה

הסבר על מבנה הפקודה:

- פקודת ה **SELECT** - ביצוע פעולת בחירת עמודות מסוימות מתוך טבלה (מופרדים בפסיקים ביניהם).
- פקודת ה **FROM** – אוסף הטבלאות שישתתפו בשליפה הנוכחית.
- פקודת ה **WHERE** - ביצוע פעולה של בחירת (סינון) השורות המקיימות את התנאי או התנאים המוגדרים בה – פקודה זו **הינה אופציונאלית** בביצוע השליפה. ללא פקודה זו תתבצע שליפה של כל השורות בטבלה (שקול לאי מתן תנאי לסינון תוצאת השליפה).

מבנה פקודה מתקדמת

מבנה פקודת SQL מתקדמת (בסוף הקורס תדעו לכתוב פקודות מסוג זה ואפילו מסובכות יותר):

```
SELECT      City, Count(*)
FROM        Students AS S
LEFT JOIN   Courses AS C
ON          S.courseID = C. courseID
WHERE       S.id BETWEEN 3344 AND 3355
GROUP BY   City
HAVING     Count(*) >2
ORDER BY   City
```

השליבים השונים בבניית טבלה חדשה בבסיס הנתונים



נתחיל ביצירת טבלה חדשה ב SQL

על מנת ליצור טבלה חדשה ב SQL נצטרך לכתוב פקודה במבנה הבא:

CREATE TABLE **table_name**

```
(  
  column_name1 data_type,  
  column_name2 data_type,  
  column_name3 data_type,  
  ....  
)
```

שם העמודה

סוג העמודה

שלושת השלבים בבניית טבלה חדשה במסד נתונים:

1) הגדרת הסכמה של הטבלה: נגדיר את העמודות שאנו רוצים שיהיו בטבלה החדשה שניצור וליד כל עמודה נגדיר את ה TYPE שלה שיגדיר איזה ערכים היא יכולה לקבל:

```
CREATE TABLE Students
```

1

```
(  
    Id int,  
    First_Name text,  
    Last_Name text,  
    Birth_Date date,  
    Hour_Salary float  
);
```

2) הכנסת ערכים לטבלה שנוצרה בהתאם לסכמה (סוגי משתנים והסדר שלהם) שהוגדרה בשלב הקודם, יש לשים לב שיהיו סוגריים בתחילת ובסוף הרשומה :

```
INSERT INTO Students  
VALUES( 123456789, 'Israel', 'Israeli', '1980-03-20', 120.40 );
```

2

הערה: כשנרצה להכניס מספר רשומות בפקודה אחת נוכל לרשום את הפקודה הבאה:

```
Values( , , , , ), ( , , , , ), ( , , , , ), ( , , , , )  
רשומה 1 רשומה 2 רשומה 3 רשומה 4
```

3) הרצת שאילתא במטרה לשלוף את כל נתוני הטבלה במטרה לראות שהטבלה נבנתה נכונה ושהערכים שהוכנסו אכן קיימים בה:

```
SELECT Id, First_Name, Last_Name, Birth_Date, Hour_Salary  
FROM Students;
```

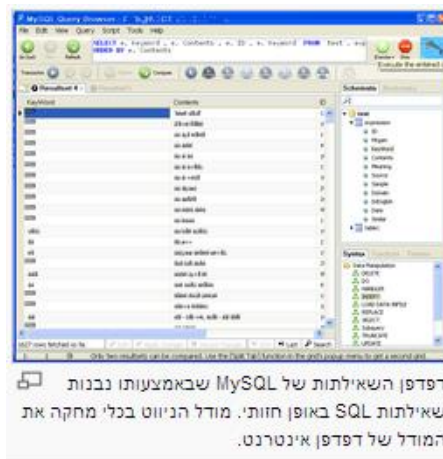
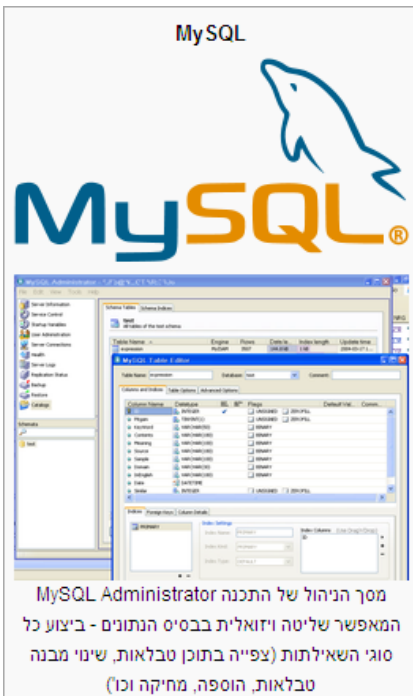
3

Students				
Id	First_Name	Last_name	Birth_date	Hour_Salary
0123456789	Israel	Israeli	20/03/1980	120.40

עבודה למול MySQL

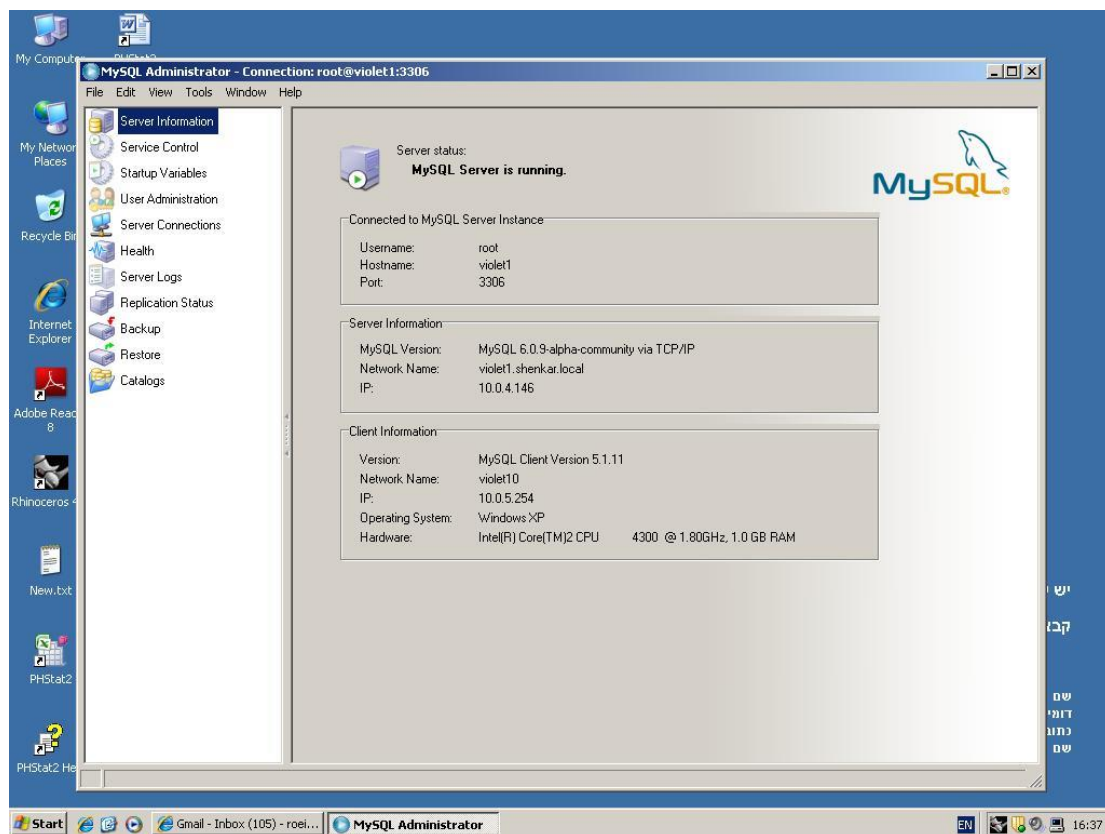
MySQL - הינו שרת בסיס נתונים רב נימי ורב משתמשים מבוסס שפת SQL, האפליקציה הינה חנימית וניתנת להורדה חופשית.

בשנים האחרונות מסד נתונים זה זוכה לפופולאריות רבה, ומאומץ גם על ידי ארגונים גדולים כדוגמת Yahoo, NASA, Facebook ועוד.

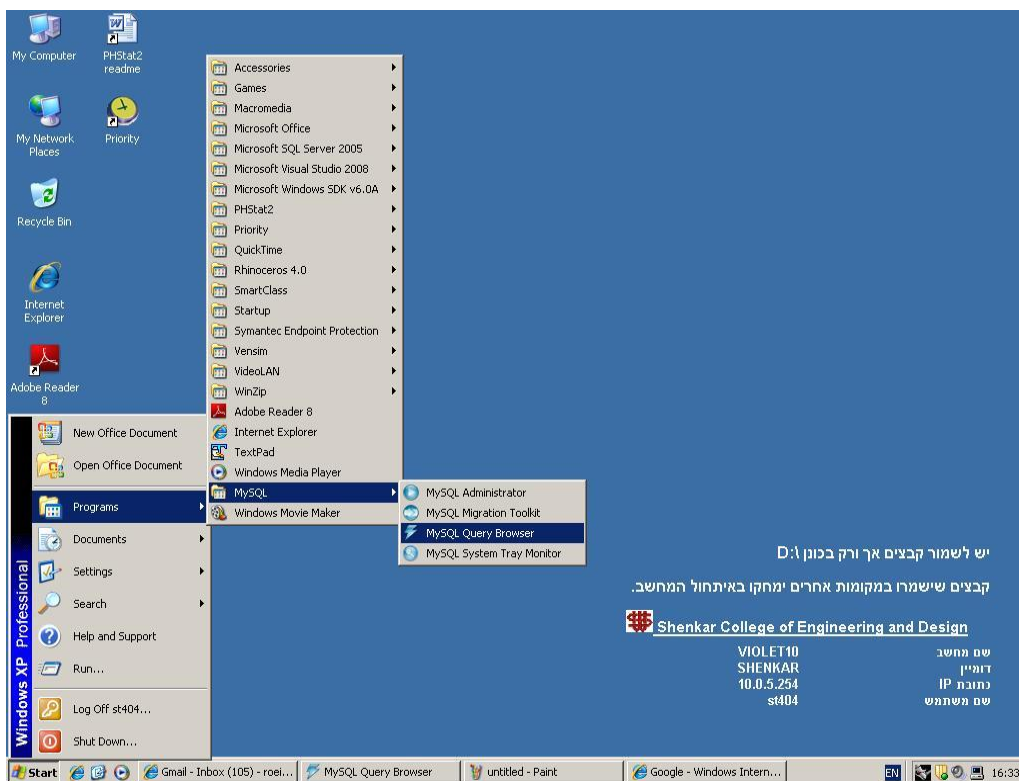


בית הספר לתעשייה וניהול
בסיס נתונים 31-028-24

MySQL - הגדרות כלליות - א

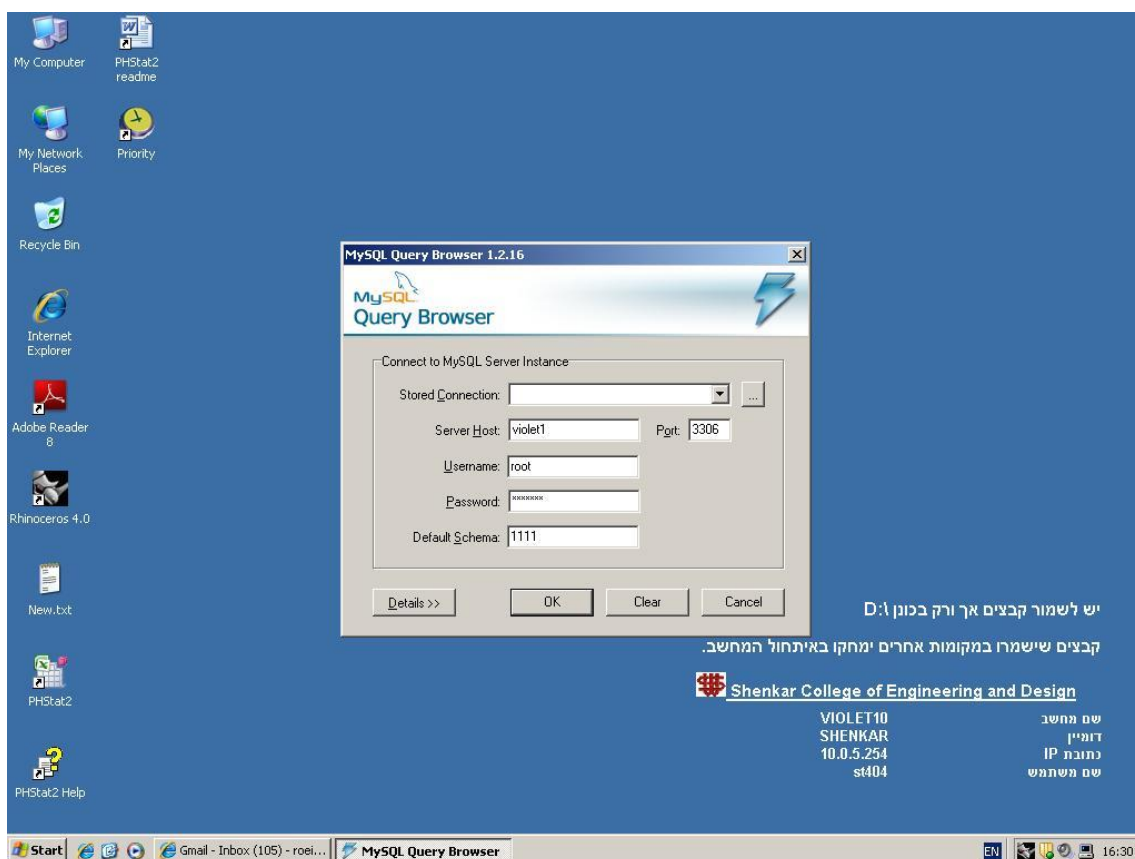


ב) כניסה למערכת



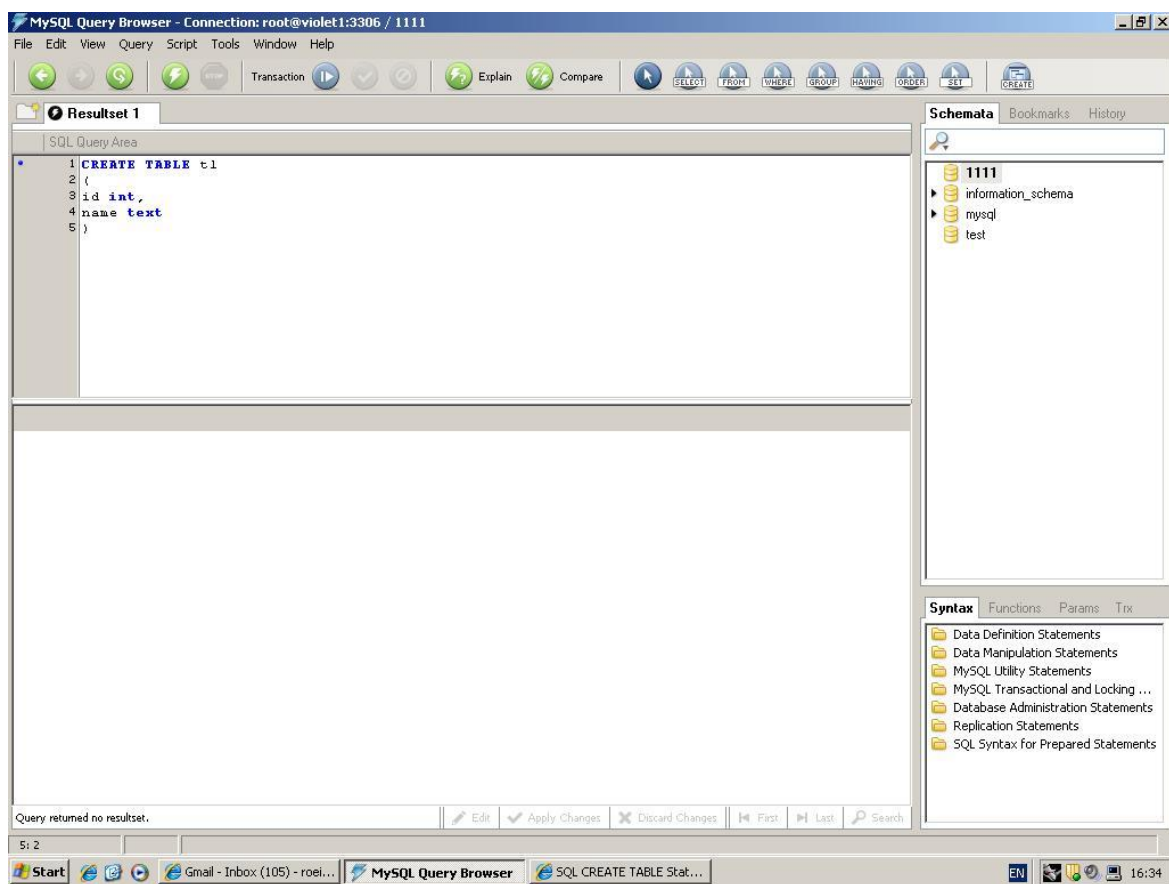
Start → Programs → MySQL → MySQL Query Browser

ג) התחברות לשרת בסיס הנתונים (בשנקר)



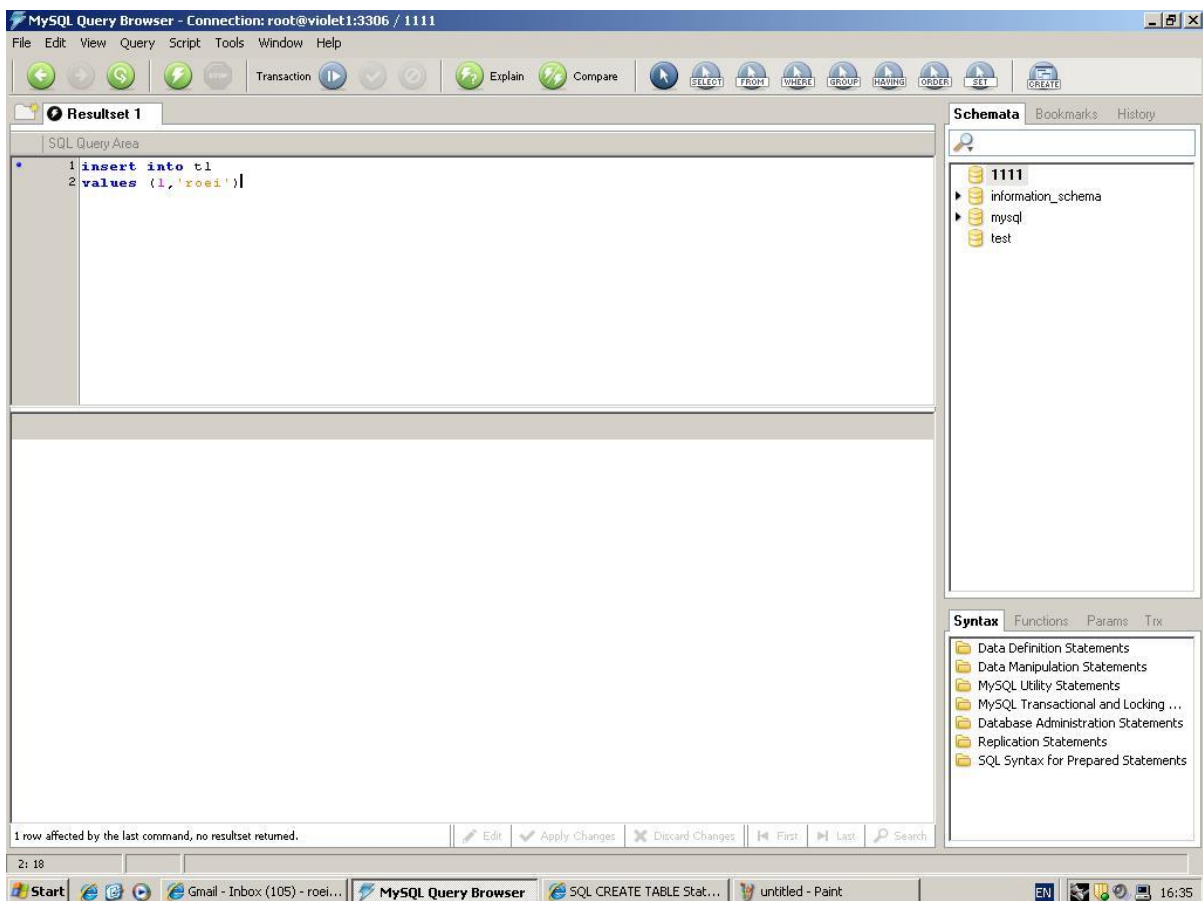
Server Host: Violet1 (localhost)
Username: root
Password: shenkar
Scheme: 1234 → (students will enter their ID number)

(ד) יצירת טבלה חדשה (תחת סכמה 1111) - (DDL)



```
CREATE TABLE t1
(
  Id int ,
  Name text
)
```

(ה) הכנסת ערכים לתוך הטבלה החדשה (DML)



```
INSERT INTO    t1
VALUES        (1 , 'roei')
```

כפתור להרצת השאילתות

1) שליפת נתונים מטבלה קיימת

The screenshot shows the MySQL Query Browser interface. The SQL Query Area contains the following query:

```
1 select *
2 from t1
```

The Schemata panel on the right shows the database structure:

- 1111
 - information_schema
 - mysql
 - test
 - roei
 - t1

The Resultset 1 panel displays the following data:

id	name
1	roei

Annotations in the image include:

- A box pointing to the 'Execute' button (a green play icon) with the text 'כפתור להרצת השאילתות'.
- A box pointing to the 'SELECT * FROM t1' query with the text 'אזור כתיבת השאילתות'.
- A box pointing to the first row of the result set (1, roei) with the text 'אזור התוצאות/השגיאות'.
- A box pointing to the Schemata panel with the text 'משתמשים מחוברים'.

**SELECT *
FROM t1**

כתיבת שאילתות פשוטות

נגדיר טבלת מידע שעליה נבצע את הדוגמאות עבור הפעולות לשיעור זה:

Students			
FirstName	LastName	ID	Grade
Avi	Cohen	3333	92
Ben	Levi	3456	80
Chen	Levin	4321	95
Debi	Dvir	3344	92

(1) פקודת ה SELECT (חובה)

[דוגמאות:](#)

(1) הצגת כל המידע מטבלת הסטודנטים:

```
SELECT *  
FROM Students;
```

FirstName	LastName	ID	Grade
Avi	Cohen	3333	92
Ben	Levi	3456	80
Chen	Levin	4321	95
Debi	Dvir	3344	92

(2) שליפת שמותיהם המלאים של כל הסטודנטים:

```
SELECT LastName , FirstName
FROM Students;
```

LastName	FirstName
Cohen	Avi
Levi	Ben
Levin	Chen
Dvir	Debi

(3) שליפת רשימת הציונים של כל הסטודנטים:

```
SELECT Grade
FROM Students;
```

Grade
92
80
95
92

נוכל לראות שהציון '92' מופיע פעמיים, חשוב להבין שב SQL המצב תקין ולכן כאשר נרצה לבטל הופעת שורות כפולות נוסיף את פקודת ה **DISTINCT** מיד לאחר פקודת ה **SELECT** ולפני רשימת העמודות שנרצה לשלוף:

```
SELECT DISTINCT Grade
FROM Students;
```

Grade
92
80
95

הערה: אם אחרי פקודת ה **DISTINCT** היינו רושמים מספר עמודות, אזי הייתה מתבצעת פעולה של ביטול כפילויות ברמת רשומה ולא ברמת השדה הבודד כמו במקרה לעיל.

פעולות נוספות שנוכל לבצע במהלך שליפת הנתונים:

דוגמאות:

1) נניח שמרצה הקורס היה רוצה לתת פקטור של 3 נקודות לכל הסטודנטים בכיתה, נוכל לבצע זאת בפקודה אחת תוך כדי ביצוע השליפה:

נשלוף את כל ת.ז. של הסטודנטים והציון המעודכן שלהם:

```
SELECT id, grade + 3  
FROM Students;
```

ID	Grade +3
3333	95
3456	83
4321	98
3344	95

2) נניח שנרצה להוסיף עמודה קבועה לטבלת התוצאה הנ"ל המסבירה שלסטודנט עם ת.ז. X יש ציון Y, נוכל לרשום זאת כך:

```
SELECT id, 'with grade', grade  
FROM Students;
```

ID	With Grade	Grade
3333	With Grade	92
3456	With Grade	80
4321	With Grade	95
3344	With Grade	92

נקבל את המחרוזת שביקשנו גם בכותרת וגם בכל אחד מהשורות בטבלה.