

מסדי נתונים – 89-281

תרגול 9 - SQL - 3 DDL

עמיעד רוזנברג

A SQL query goes into a bar,
walks up to two tables and asks, "Can I join you?"

- עד כה התעסקנו עם שליפת מידע מטבלאות.
- בשיעור זה נראה כיצד ליצור ולעדכן את הטבלאות עצמן.

בסיס הנתונים - DDL

- בשלב הראשוני, עדיין אין לנו שום בסיס נתונים לעבוד עליו.
- יצירת בסיס נתונים חדש תתבצע ע"י הפקודה:
 - `CREATE DATABASE databade_name;`
 - מחיקת בסיס נתונים תתבצע ע"י הפקודה:
 - `DROP DATABASE database_name;`
 - הצגת בסיסי הנתונים הקיימים:
 - `SHOW DATABASES;`
 - שימוש ב \ כניסה אל בסיס נתונים ספציפי:
 - `USE database_name;`

טבלאות - DDL

- לאחר שנכנסנו אל בסיס נתונים מסויים – ניצור טבלאות ע"י :
- `CREATE TABLE table_name (
col1_name col1_type, col2_name col2_type,
col3_name col3_type,...);`
- לסקירה של טיפוס הנתונים :
- <http://dev.mysql.com/doc/refman/5.0/en/data-types.html>

טבלאות - DDL

- בנוסף, ניתן לתת הגדרות שונות לעמודות. לדוגמה:
- NOT NULL : הערך בעמודה יהיה שונה מ NULL.
- AUTO INCREMENT : ערך שעולה אוטומטית.
- לדוגמא:
- ```
CREATE TABLE employees (e_number INT
 AUTO_INCREMENT, name CHAR(30), salary INT,
 PRIMARY KEY(e_number));
```
- ל(הרבה) פרטים נוספים:
- <http://dev.mysql.com/doc/refman/5.0/en/create-table.html>

# טבלאות - DDL

- מחיקת טבלה :

- DROP TABLE **table\_name**;

- מחיקת נתוני הטבלה (מבלי למחוק את הטבלה עצמה) :

- TRUNCATE TABLE **table\_name**;

- הצגת רשימת הטבלאות בבסיס הנתונים :

- SHOW TABLES;

- הצגת התיאור של הטבלה :

- DESCRIBE **table\_name**;

# טבלאות - DDL

- שינוי מבנה טבלה :
- ALTER TABLE **table\_name** **requested\_change**;
  - השינויים האפשריים (requested\_change) הם :
    - הוספת עמודה :
    - ADD **col\_name** **col\_type** [FIRST / AFTER col] ;  
ברירת המחדל היא הוספת העמודה כעמודה אחרונה.
    - מחיקת עמודה :
      - DROP **col\_name**;
      - שינוי סוג עמודה :
    - MODIFY **col\_name** **col\_type**;
    - שינוי שם עמודה :
  - CHANGE **old\_name** **new\_name** **col\_type**;
  - פרטים נוספים :
- <http://dev.mysql.com/doc/refman/5.0/en/alter-table.html>

# עדכוני מידע - DML

- הכנסת ערכים לטבלה :
- INSERT INTO **table\_name** (**col1**, **col2**,...) VALUES (**val1.1**, **val2.1**,...), (**val1.2**, **val2.2**,...), ... ;
  - אם נרצה להכניס ערכים לכל העמודות, אין צורך לרשום את שמות כל העמודות :
- INSERT INTO **table\_name** VALUES (**val1.1**, **val2.1**,...), (**val1.2**, **val2.2**,...), ... ;
- פרטים נוספים :
- <http://dev.mysql.com/doc/refman/5.0/en/insert.html>



# עדכוני מידע - DML

- עדכון נתונים :

- UPDATE **table\_name**  
SET **col\_name1** = **value1**, **col\_name2** = **value2**, ...  
WHERE **condition** ;

- מחיקת רשומות בטבלה :

- DELETE FROM **table\_name**  
WHERE **condition** ;

- פרטים נוספים :

- <http://dev.mysql.com/doc/refman/5.0/en/update.html>

# DDL – View

- View מאפשר לנו ליצור "טבלה מדומה".
  - בטיחות - מאפשר לנו לתת הרשאה לנתוני ה-View בלבד.
  - נוחות - מאפשר לנו לצפות בנתונים ישירות ללא צורך לכתוב את כל השאילתה.
- 
- `CREATE VIEW view_name AS (SELECT ... );`
  - `CREATE VIEW class_view AS (SELECT *  
FROM class  
WHERE a=1);`
  - `SHOW TABLES;`
  - `SELECT * FROM class_view;`

# DDL – View

- נבצע ניסוי קטן :
- נוסיף ערך לטבלה class
- INSERT INTO class VALUES (1,8);
- נבדוק מה ייתן ה-view עכשיו
- SELECT \* FROM class\_view;
- מסקנה – שינוי נתונים בטבלה המקורית משנה את נתוני ה-view.
- בצורה זו ניתן למשל ליצור עבור כל מחלקה view למנהל המחלקה שיראה רק את העובדים תחת אותו המנהל ולתת לו הרשאת גישה ל-view במקום לטבלת העובדים כולה.