1. בנספח  
  
3. זה הזרקת שאילתה SQL, בדרך כלל דרך Input כל שהוא באפלקציה שלנו,  
 שאילתה שיוכלה לקחת מידע, לשנות אותו.  
 הדרך למנוע: לסנן את היכולת להכניס תווים מיוחדים  
 להשתמש בשפת QUERY כדי לעשות בקשות לDB   
4. בקטע קוד הבא המשמעות שלו שאובייקט Font יהיה מנוקה לאחר השימוש בו.(dispose)  
 נשתמש בUSING שנרצה שאובייקט מסויים למשל, שאין לנו צורך בו בהמשך התוכנית ינוקה מיד אחרי השימוש בו.

5. א.table Local temp  
 Global Temp Table  
 ב. נשתמש table Local temp – כאשר נרצה לשמור נתונים על session אחד ספציפי  
 נשתמש Global Temp Table – כאשר נרצה להשתמש בה בכמה session שמחוברים כרגע  
 ג. ב table Local temp כאשר נרצה לבצע פרוצדרוה גדולה   
 ב Global Temp Table כאשר נרצה מידע משותף בין כמה session שמחוברים כרגע  
 אבל ממש חשוב לשים לב שעושים את זה בדרך הנכונה, כי אחרת זה סתם הורס ביצועים

6. **Promises**  – 1. מייצג Value אחד שיוחזר, אם תשובה של successful or error

2. אי אפשר לבטלו

עדיף להשתמש: לדוגמא במקרה שנרצה תשובה מAPI ומחכים לתשובה  
 אסינכרונית

**Observables** - 1. אפשר לספק איתו כמה Value במקביל בבקשה אחת

2. ניתן לבטלו עם unsubscribe

עדיף להשתמש: לדוגמא במקרה שנצטרך מידע ב real time

7. שימוש בעקרון IOC, במקום שבכל פעם שנצטרך לבצע new instance לאובייקט שמשתמשים בו דיי הרבה, פשוט אותו אובייקט בתחילת הריצה ייוצר ויוכנס ל factory ומשם בכל פעם שנרצה להשתמש בו, נקרא לו מה construcator של ה class  
וחדש ל .NET CORE: 1. היכולת שאחרי כל בקשת לקוח נעשה dispsed  
 2. האפשרות להשתמש ב IServiceProviderFactor שמאפשר למפתחים ליצור  
 DI containers משל עצמם בצורה נכונה וקלה.

8. 1. public string Name {get;set;}; - יש להוריד את ה;   
 2. \_clientList.FirstOrDefault – בעדיפות לא להשתמש בזה כי יכול לחזור לי ערך שלא נרצה  
 במקרה שלא יהיה כזה לקוח, נרצה לטפל המקרה זה ולנהל  
 את זה במקרה ויהיה null