1.

SELECT TOP 30

c.CustomerId AS CustomerId,

COUNT (pd.ID) AS PolicyNumCount,

SUM(pd.InsuranceAmt) AS InsuranceAmtSum

From CRM.Customer c

JOIN CRM.PolicyDetails pd ON pd.CustomerId = c.CustomerId

GROUP BY c.CustomerId

HAVING COUNT (pd.ID) > 4 AND SUM(pd.InsuranceAmt) > 2000

2.

SELECT c. ID AS ID,

c.IdentityNo AS IdentityNo,

c.FullName AS FullName,

COUNT (pd.ID) AS PolicyNumCount,

SUM(pd.InsuranceAmt) AS InsuranceAmtSum

FROM CRM.Customer c

JOIN CRM.PolicyDetails pd ON pd.CustomerId = c.CustomerId

WHERE pd.StateCode = 4

GROUP BY c.ID, c.IdentityNo, c.FullName

4.

השימוש בבלוק משמעו שהאוביקט שהצהרנו עליו בתוך הסוגריים קיים בבלוק בלבד, לא ניתן יהיה להשתמש בו מחוץ לבלוק (נמחק מהזיכרון לאחר היציאה מהבלוק)

בדרך כלל משתמשים בבלוק עבור אוביקטים מסוג IDisposable, לדוגמא עבור db, file וכו'

כאשר הבלוק מסתיים הנתונים של האוביקט נמחקים מהזיכרון שהוקצה עבורו

5.

טבלאות זמניות לוקאליות (per session) וגלובאליות

לוקאליות – כאשר הטבלה נצרכת למשתמש מסוים בלבד או למידע ספציפי לדוגמא: צריך להחזיר את פרטי הביטוח ללקוח מסוים, אם ישנו עיבוד משמעותי ומורכב של הנתונים,select אחד עלול להיות בעיתי מבחינת מורכבות וביצועים ולכן משתמשים בטבלאות זמניות לאותה פרוצדורה/פונקציה בלבד

גלובאליות – כאשר יש צורך ביצירת טבלה זמנית שתהיה זמינה להרבה משתמשים ובהרבה , תהליכים, לדוג': דוח חודשי של לקוחות, כאשר מדובר בחודש עבר. אם מספר משתמשים במערכת צריכים פחות או יותר את אותם נתונים אפשר לעבד להם את המידע המרכזי בטבלה הזמנית ועליו לעשות את השליפות הרצויות

6.

Promise – מקשיב לפעולה אחת בלבד, ברגע שמתקבלת תשובה הוא מפסיק להקשיב, לדוג': נשלחה קריאה לשרת לקבל נתונים – ברגע בו יתקבלו הנתונים המערכת תמשיך לפעולה הבאה

Observable – פעיל כל הזמן, מרגע שהגדירו אותו כמקשיב לפעולה מסוימת, הוא ימשיך להקשיב, לדוג': צ'אט, המערכת מקשיבה לשרת לקבלת עדכוני צ'אטים תוך כדי שהמשתמש יכול להמשיך בפעולות ההתכתבות האחרות

7.

Dependency Injection – משמעות המילים היא הזרקת תלויות, הזרקת תלויות בעצם מאפשרת למחלקה מסוימת להשתמש במחלקה חיצונית.

.NetCore - זוהי פלטפורמה של ווינדוס שבעצם מהווה שדרוג של .netFramework בשונה ממנה היא מאפשרת לפתח על מחשבים שאינם דווקא עובדים על ווינדוס וכן להעלות את המוצר המוגמר לשרתים שאינם ווינדוס. ישנו שיפור בביצועים האסינכרוניים שלה ונתווספה לה הזרקת תלויות

האופן בו משתמשים בהזרקת תלויות –

נניח ויש לי קונרטרולר שאני רוצה להשתמש בservice חיצוני

אני מוסיפה בstartup > configureServices את הservice וה interface שלו

בקונטרולר אני יוצרת משתנה מסוג הinterface, מוסיפה בconstructor משתנה גם מאותו סוג ומשימה את הערך שלו במשתנה הראשון שיצרתי ואז בעצם אני אוכל להשתמש במתודות ובנתונים השונים של הservice