**בסיס נתונים רלציוני – המודל הטבלאי**

בסיס נתונים (**DataBase**  או **DB**):

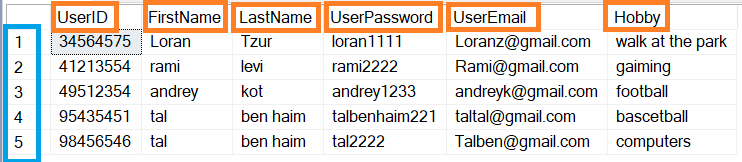
בסיס נתונים, הוא אמצעי (שרת על גבי מחשב), המשמש אותנו לאחסן נתונים, זאת על מנת שנוכל לאחזר ולעבד אותם. בדרך כלל נאחסן את הנתונים באמצעות מערכת לניהול בסיס נתונים (DBMS - Database Management System). היצירה, הניהול השוטף, והגישה לנתונים, יתבצעו באמצעות שפת סקריפט (script) הנקראת SQL (**S**tructured **Q**uery **L**anguage) – התרגום: שפת שאילתות מובנת. באמצעות השפה, אנו מסוגלים לשלוף נתונים מבסיס הנתונים, ולנתח אותם. ישנם מספר מודלים לניהול בסיס נתונים: רשתי, היררכי, **טבלאי**, מונחה עצמים, NOSQL.

מודל הטבלאי (**RDB - R**elational **D**ata**B**ase )

נקרא גם **המודל היחסי**, או **המודל הרלציוני**. נתחיל בהגדרות בסיסיות:

* במודל זה, הנתונים מאוכלסים **בטבלאות (Tables)**, כאשר הטבלה מייצגת **ישות** מסוימת.
* כל טבלה, מחולקת **לעמודות (Columns)**, אשר משמשות **מאפיינים** לישות.
* כל **שורה (Rows)** בטבלה הינה **רשומה**, כלומר **מקרה ספציפי** / ערך מתועד.

נשתמש בדוגמא על מנת להמחיש זאת. בדוגמא הבאה נוכל לראות ישות בשם לקוחות:



אפשר לראות בתמונה, את **העמודות מסומנות בכתום**. כל לקוח מאופיין על בסיס **השדות** הללו.

לכל לקוח מספר מזהה, שם פרטי, שם משפחה, סיסמא, מייל, ותחביב (משמאל לימין).

את **הרשומות** עצמן ניתן לראות **בכחול**. כל רשומה מייצגת לקוח ספציפי.

אם נפענח את השורה הראשונה בטבלה, נוכל להבין, כי הלקוח בשורה הראשונה הוא:

בעל תעודת זהות **34564575** ,

שמו הפרטי הינו **לורן**,

שם משפחתו הינו **צור**,

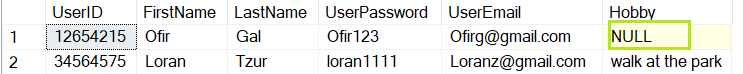
הסיסמא שלו היא **loran1111**,

המייל שלו [**loranz@gamil.com**](mailto:loranz@gamil.com),

והתחביב שלו הוא **ללכת בפארק**.

ערך ריק - NULL:

כל עמודה (שדה), מוגדרת באחת משתי הדרכים הבאות: כעמודה אשר מחייבת הזנת ערך (לדוגמא תעודת זהות), או **עמודה שהערך בה יכול להישאר ריק מתוכן (NULL).** שדה תחביב לדוגמא, לא חייב הזנת ערך. ראו מוקף בירוק ,ללקוחה אופיר לא הוזן תחביב, על כן הערך הינו ריק (NULL):



טיפוסי נתונים (TYPE):

כל עמודה (שדה), מוגדרת באמצעות סוג **טיפוס נתונים**, על מנת שהמערכת תדע כיצד לשמור את הנתונים המוזנים אליה, וכיצד לפנות ולערוך נתונים אלו. טיפוסים (**TYPE**) נפוצים לדוגמא:

* מספרי – רק ספרות (int),
* מחרוזת של תווים שונים (nvarchar),
* תאריך ושעה (datetime),
* בוליאני (BIT) – 2 אופציות בלבד כמו זכר/נקבה.

כמובן שישנם עוד טיפוסים רבים, אך לשם ההסבר נסתפק באלו.

מפתח ראשי (primary key):

**לכל ישות (טבלה), מפתח ראשי אחד בלבד**, אשר מורכב משדה אחד, או שילוב של מספר שדות.

שדה / שדות אלו, יוצרים זיהוי ייחודי לכל רשומה. בדוגמא הראשונה שהצגתי, ניתן לראות מספר לקוחות עם מאפיינים שונים. שדה המפתח שנקבע לישות זו היה מס תעודת זהות. מדוע? כיוון ושדה מפתח ראשי צריך לעמוד בכמה דרישות:

* שדה **ייחודי** (חד-חד ערכי), שלא יכול לשמש 2 רשומות שונות. לדוגמא, יכולים להיות 2 אנשים שונים עם אותו שם, על כן שדה שם לא יכול להיות שדה מפתח.
* שדה שערכיו נקבעים פעם אחת בלבד, **ולא מסוגלים להשתנות**. לדוגמא, כתובת מייל יכולה להיות ייחודית לכל משתמש, אך היא יכולה להשתנות. על כן מייל לא תשמש כשדה מפתח.
* מחייב ערך (**לא מאפשר NULL**) . חייב להיות שדה המחייב אותנו להזין אליו ערך ממשי. לדוגמא, תחביב לא יכול להיות NULL, כיוון ומתאפשרת אי הזנת ערך.

קרדינליות הקשר - **Cardinality** (קשרי גומלין בין טבלאות):

התקשורת בין הישויות השונות מתבצעת באמצעות הגדרת **קשרים בין הטבלאות**. קשרים אלו מוגדרים על פי צורכי המערכת ובהתאם להיגיון ניהול הנתונים. הם באים לידי ביטוי באמצעות הגדרת יחס בכל צד של ישות (**יחיד 1** או **רבים N**). במקרים מסוימים, טבלה יכולה אף להיות בקשר עם עצמה. איך נגדיר נכון את הקשר?

1. **נרכיב משפט המאפיין את היחס בין 2 הישויות (טבלאות).**
2. **נשאל את עצמנו מה מספר המופעים שכל טבלה מייחסת לטבלה השנייה.**

נמחיש זאת באמצעות דוגמאות. נתחיל עם קשר של יחיד לרבים 1: N

יחיד לרבים, 1:N

ישנם 2 ישויות (טבלאות):

* לקוחות
* הזמנות

המשפט המאפיין את הקשר: **לקוחות --** מבצעים **-- הזמנות**.

כעת, נברר מהו סוג הקשר:

* כל לקוח יכול לבצע **מספר רב** של הזמנות.
* כל הזמנה שייכת ללקוח **אחד בלבד**.

על כן, נוכל להגדיר את הקשר כך:

1

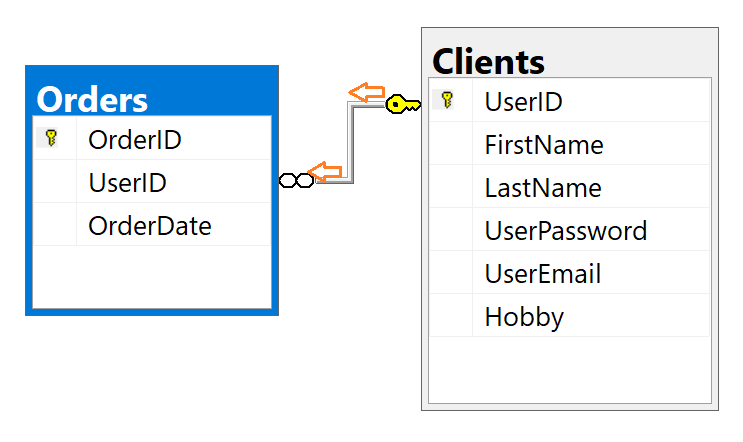
N

לקוחות

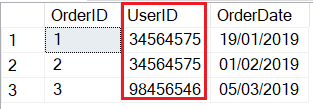
הזמנות

מפתח זר (Foreign key)

לאחר שהגדרנו את הקשר בין 2 הטבלאות, שדה המפתח הראשי ישוכפל מהיחיד לרבים. במקרה הזה, שדה המפתח של טבלת לקוחות יופיע כשדה גם בטבלת הזמנות (ראו כיוון החץ). משמעות הדבר – לטבלת הזמנות קיימת תלות מסוימת בטבלת לקוחות. בואו נראה זאת בדוגמא:



כעת ניתן לראות את שדה תעודת זהות בטבלת הזמנות:



כך בעצם ניתן להגיד, שכל הזמנה משויכת ללקוח אחד בלבד (במידה ושדה USERID לא יאפשר NULL ).

**שימו לב! השדה המקשר יכול להיות לשמש שדה מפתח בטבלה החדשה, אך הדבר לא מחייב. כמו בדוגמא זו, שדה תעודת זהות איננו משמש מפתח ראשי בטבלת הזמנות, אלא רק שדה זר.**

רבים לרבים, N:N

כאשר נעבוד על פי אותה סכמה של בדיקת הקשר, נמצא מקרים בהם היחס הינו רבים לרבים. משמעות הדבר היא, שלקשר בין 2 הישויות יש תכונות משלו.

לדוגמא, היחס בין

* מוצרים
* הזמנות

המשפט המאפיין את הקשר: **מוצרים –** מופיעים ב **-- הזמנות**.

כעת, נברר מהו סוג הקשר:

* כל מוצר יכול להופיע ב**מספר רב** של הזמנות.
* בכל הזמנה יכולים להופיע **מספר רב** של מוצרים.

על כן, נוכל להגדיר את הקשר כך:

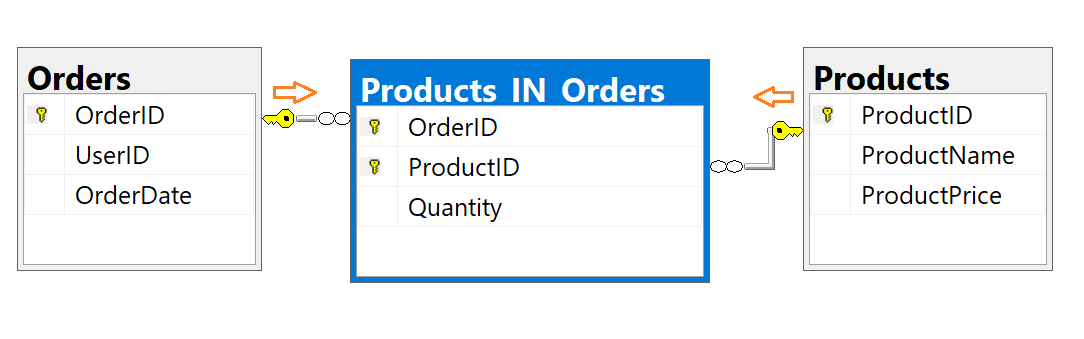
N

N

מוצרים

הזמנות

במקרה שכזה, ניצור טבלה מקשרת בין הטבלאות, אליה יעברו השדות אשר שימשו כשדות מפתח:

****

שימו לב, במקרה זה נוצרה טבלת "**מוצרים בהזמנות**", ואלו ההגדרות שיצרנו:

* **מפתח ראשי בטבלת מוצרים שוכפל לטבלה המקשרת**
* **מפתח ראשי מטבלת הזמנות שוכפל לטבלה המקשרת**
* **2 המפתחות הזרים, הוגדרו ביחד כמפתח ראשי אחד בטבלה המקשרת**

**תזכורת: השדה המקשר יכול להיות לשמש שדה מפתח בטבלה החדשה, אך הדבר לא מחייב. בדוגמא, השדות הזרים (מספר הזמנה ומוצר) אכן משמשים מפתח ראשי בטבלה המקשרת.**

שימו לב, נוצר שדה חדש – כמות. שדה זה יתאר לנו כמה מוצרים יש בהזמנה ספציפית.

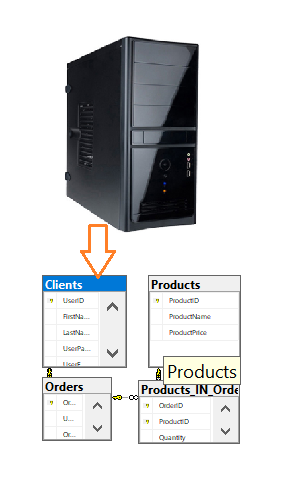
**שפת SQL**

שפת מחשב, אשר פותחה בשנות ה 70 על ידי IBM לשם שליפה ותשאול נתונים מ- DB. עיקר עבודתה עם DB טבלאי. אנו נלמד את 3 הפקודות הבסיסיות לתשאול על פי SQL (Microsoft). פקודות אלו מופעלות בבת אחת כמקשה אחת, על מנת לספק שליפת נתונים מטבלה קיימת:

1. **Select** – בחירה של שדות להצגה
2. **From** – מתוך טבלת
3. **Where** – על פי תנאי לסינון נתונים

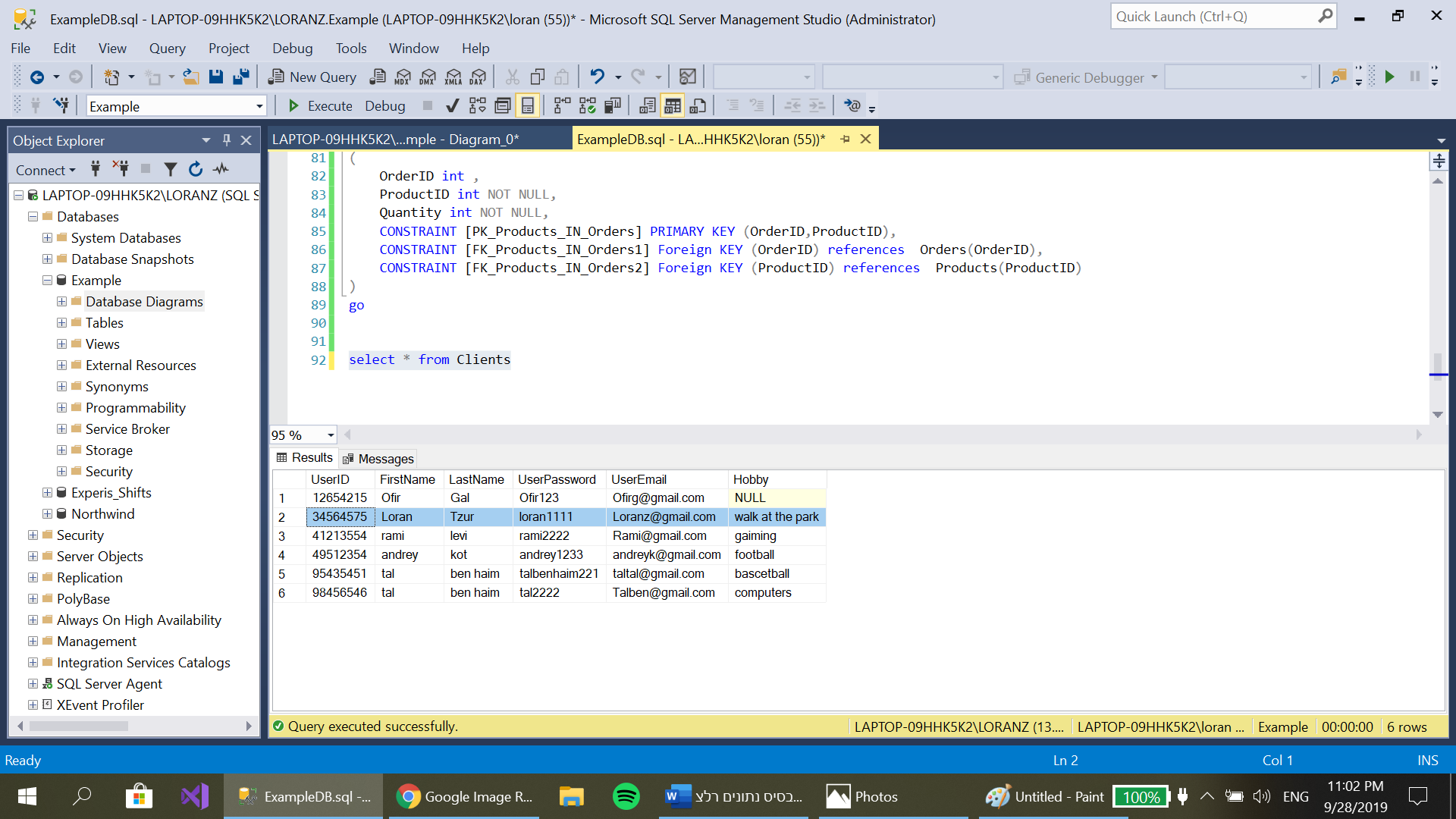
זהו הסדר בו מתבצעת כתיבת השאילתה בקוד. תחילה Select , לאחר מכן from, ולסוף where. חשוב שנבין את סדר הפעולות מאחורי הקלעים:

1. כאשר הפקודה רצה, המחשב קודם רוצה לדעת לאיזו טבלה לגשת. על כן (FROM) היא הפעולה הראשונה שמתבצעת. לדוגמא, שלוף את טבלת לקוחות:

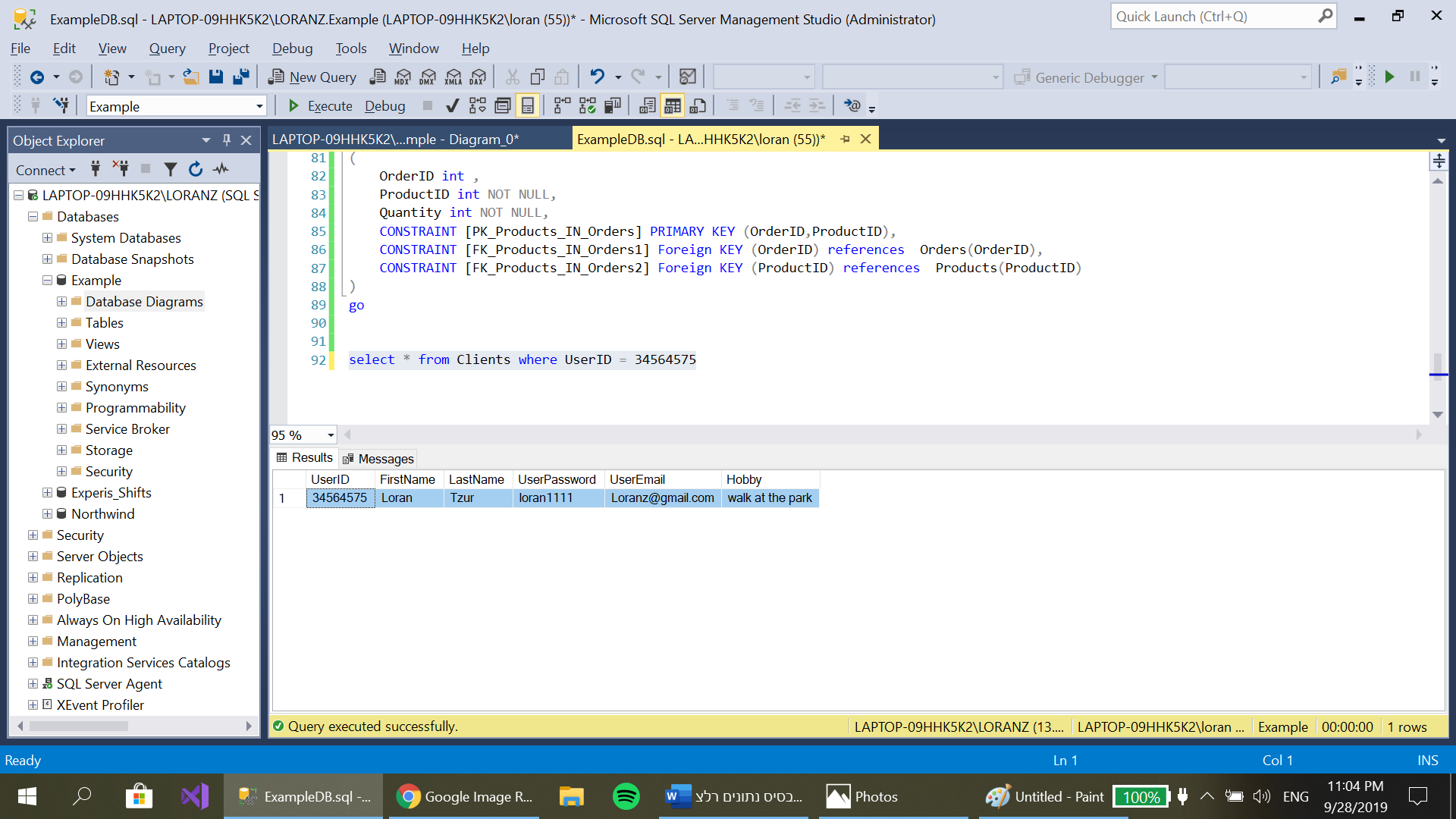


FROM Clients

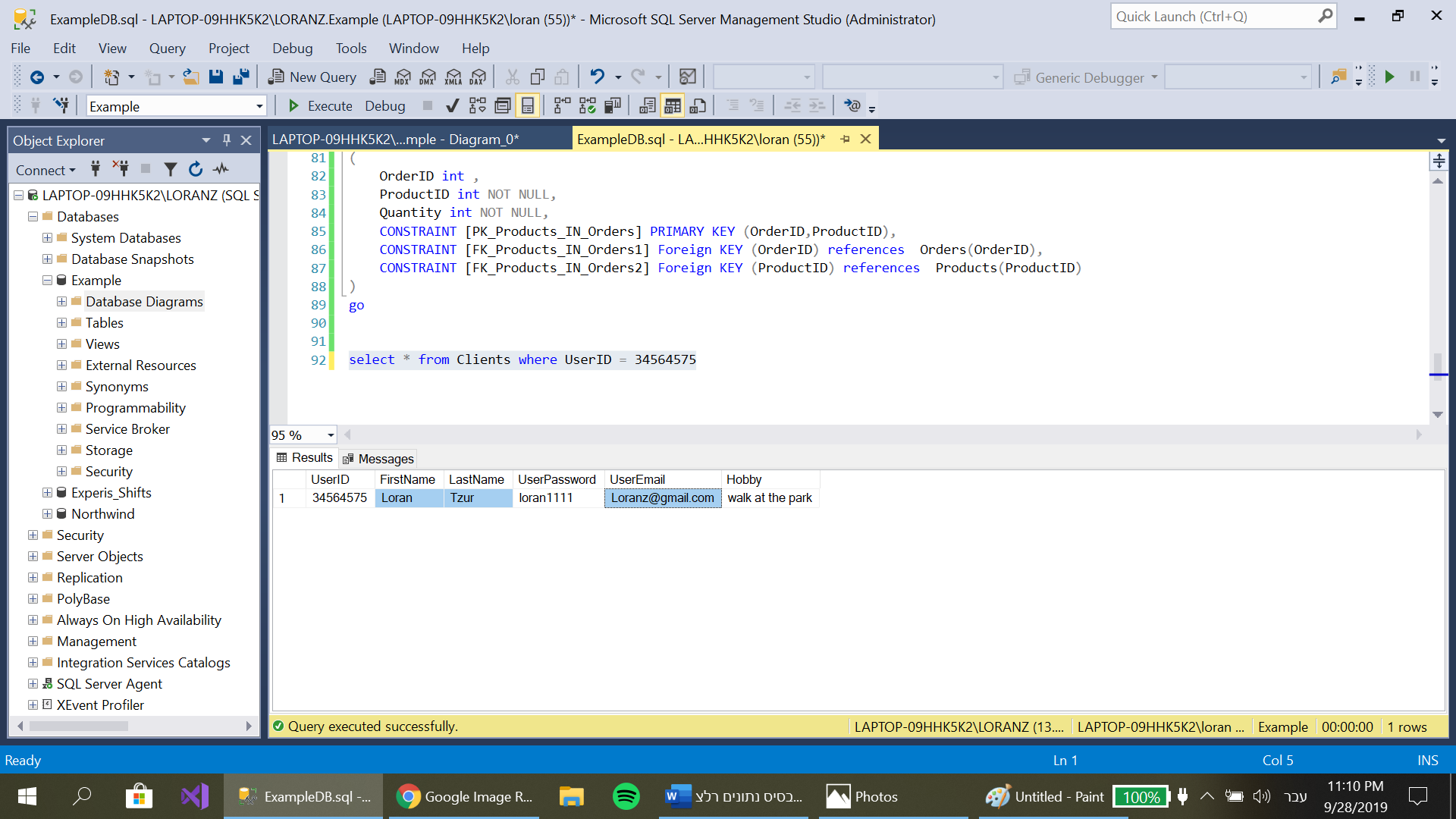
1. לאחר שהמחשב ניגש לטבלה הרצויה, הוא ירצה לדעת מה לסנן (WHERE). לדוגמא, שלוף את הלקוח עם תעודת זהות 34564575:



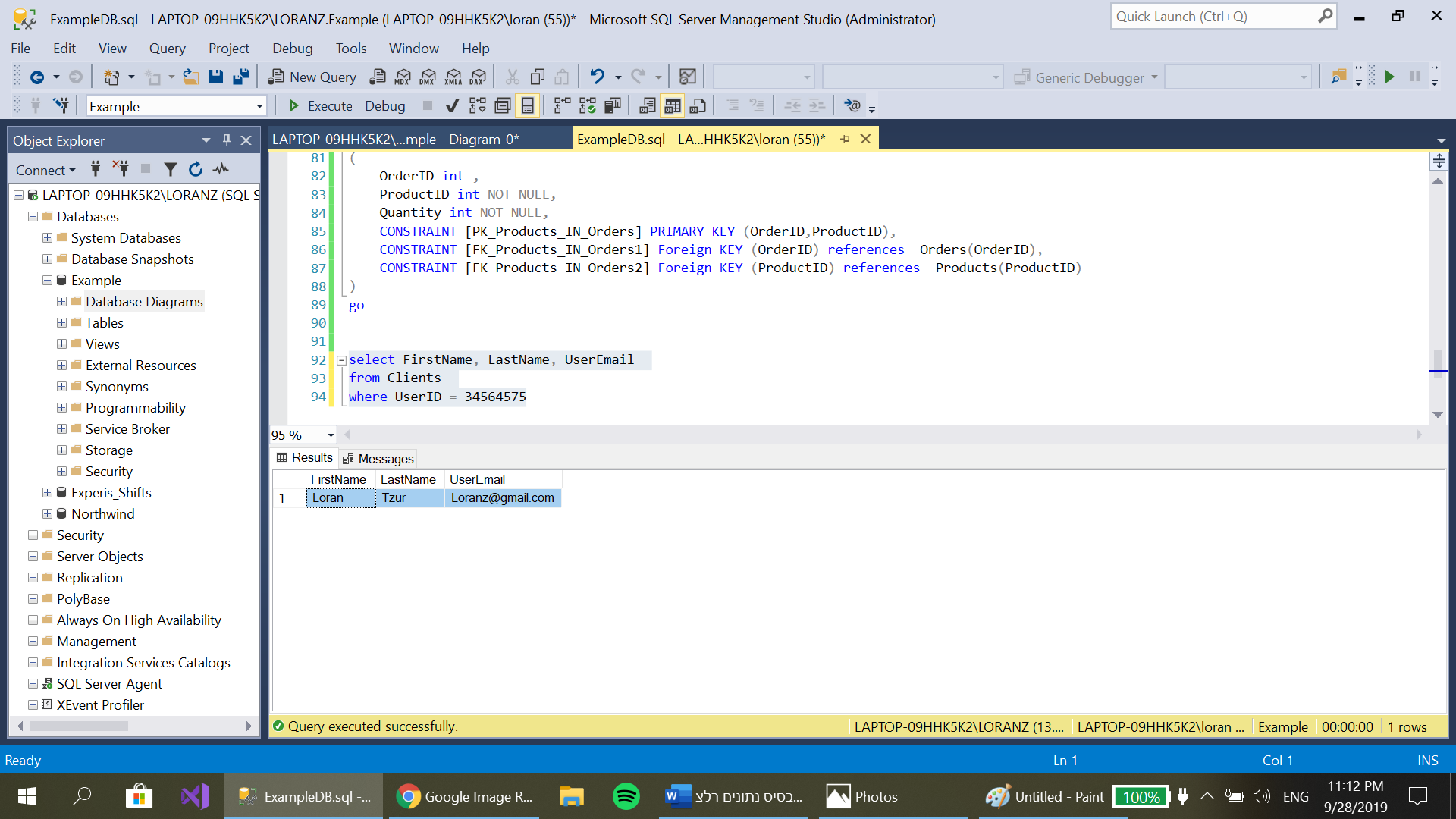
WHERE UserID = 34564575



1. הפעולה השלישית שתתבצע, הינה פעולת ה- SELECT. בפקודה זו אני בוחר אילו שדות אני מעוניין לשלוף. לדוגמא, שלוף את שדות שם פרטי, שם משפחה, ומייל של הלקוח:



SELECT FirstName, LastName, UserEmail



הפקודה בשלמותה תיכתב בקוד בצורה הבאה:

Select FirstName, LastName, UserEmail

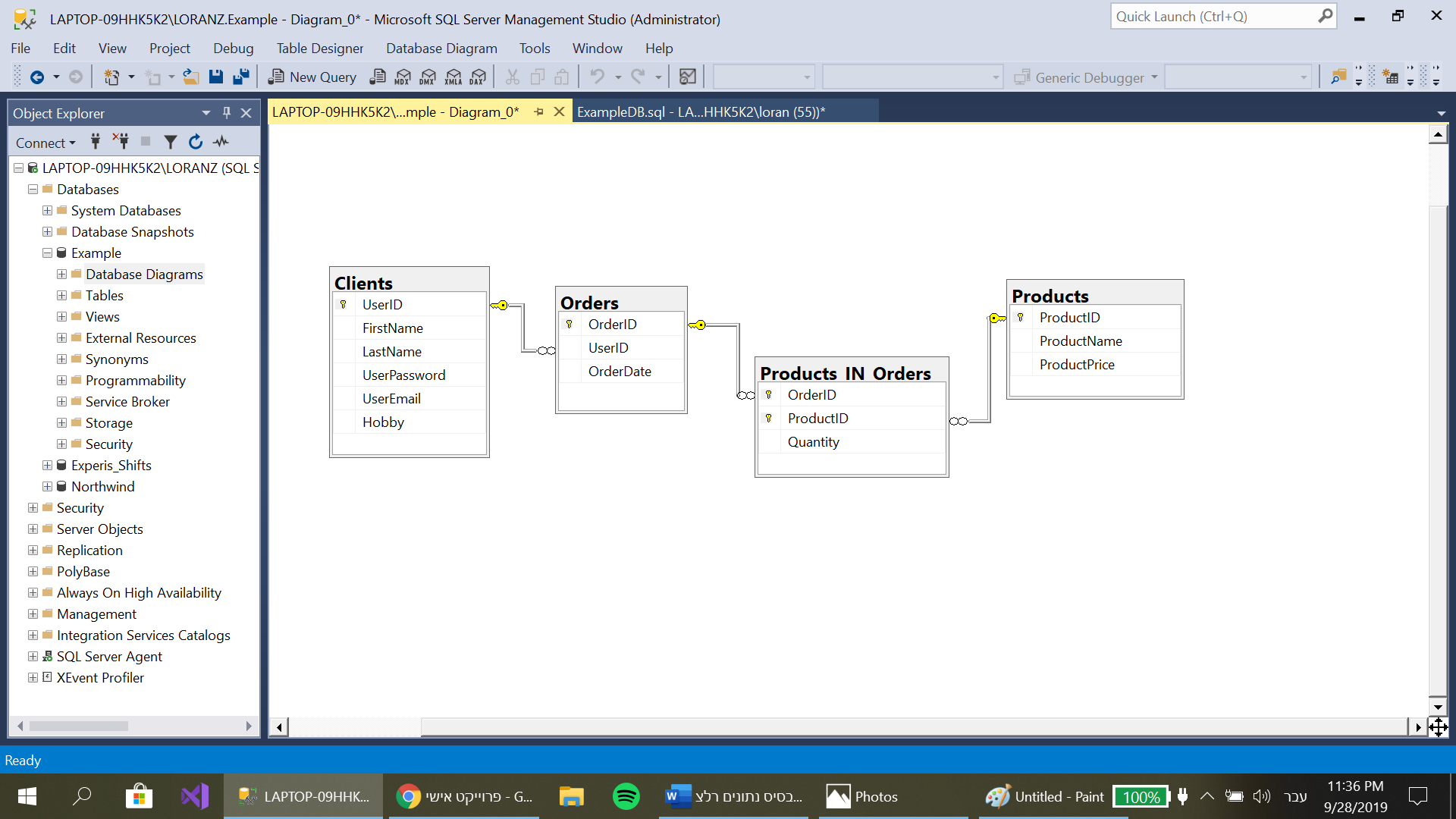
From Clients

Where UserID = 34564575

לסיכום

עולם הדטה הוא עולם שלם של אפשרויות, ובמסמך זה הכרנו חלק מזערי ממסד הנתונים הרלציוני. המידע במסמך זה הוא על חוד המזלג, אך מהווה אבן יסוד חשובה בהבנת המודל הטבלאי.

יכולים לראות את הדיאגרמה של מסד הנתונים שבניתי לדוגמאות שביצעתי כאן שימוש בהם:



ועוד כמה פקודות לשליפה עם הסבר:

Select ProductName from Products where ProductPrice = 5

שלוף את שמות המוצרים, אשר מחירם 5

Select ProductID, Quantity from Products\_IN\_orders where OrderID = 1

שלוף את מספר המוצר ואת הכמות שלו בהזמנה מספר 1

Select \* from orders

שלוף את כל השדות מטבלת הזמנות