

Querying Data with SQL Server

Exercise

תוכן עניינים

3SELECT
5 WHERE
7 SCALAR FUNCTIONS
9 JOIN
11GROUP FUNCTIONS
13SUBQUERIES
14 Set Operators
15Analytic Functions
16 DML
18 יצירת DATA BASE ופקודות DDL ליצירת טבלה
19 VIEWS
20 BASIC T- SQL
21 ERROR HANDLING
22STORED PROCEDURES
23 פתרונות
23SELECT
26 WHERE
30 SCALAR FUNCTIONS
33 JOIN
36GROUP FUNCTIONS
39 SUBQUERIES
42 Set Operators
43Analytic Functions
45 DML
47 יצירת DATA BASE ופקודות DDL ליצירת טבלה
48 VIEWS
51 BASIC T- SQL
55 ERROR HANDLING
57STORED PROCEDURES

SELECT

יש לבצע תרגול זה ואת הבאים אחריו בבסיס הנתונים Northwind

1. הצג את כל המידע מטבלת Orders.
2. הצג את כל העמודות מטבלת Employees
3. הצג מטבלת Employees את העמודות הבאות : Country , Region , HireDate , FirstName
4. הצג מטבלת Orders את העמודות הבאות : OrderDate , OrderID , CustomerID
5. הצג מטבלת Products את העמודות הבאות : ProductID (אליאס - ProId) , ProductName , UnitPrice (אליאס - ProNm) , (אליאס - UntPr)
6. הצג מטבלת Employees את העמודות הבאות : Address (אליאס - Add) , City (אליאס - Ct) , Region (אליאס - reg)
7. הצג מטבלת customers שתי עמודות: קוד הלקוח בעמודה אחת, הכתובת והעיר משורשרות יחדיו בעמודה שנייה. יש לתת את השם "full address" לעמודה השנייה.
8. הצג מטבלת Employees שלוש עמודות:
השמות המלאים של העובדים (שם פרטי משורשר עם שם משפחה) תחת הכותרת Full Name
את תאריך יום ההולדת שלהם פלוס 8 ימים תחת הכותרת Birth Date
ואת קוד המנהל (עמודת ReportsTo) תחת הכותרת Manager#
9. הצג מטבלת Employees את הערים (City) מהם מגיעים העובדים בצורה ייחודית
10. הצג מטבלת Employees את המדינות (Country) מהם מגיעים העובדים בצורה ייחודית
11. הצג מטבלת Employees את תיאור המשרה (Title) של העובדים בצורה ייחודית
12. א. הצג מטבלת Customers את המדינה (Country) ואת העיר (City)
ב. הצג מטבלת Customers את השילוב הייחודי של המדינה והעיר
13. הצג מטבלת Employees את שמו הפרטי של העובד , תאריך הולדתו , ותאריך הולדתו + 5 ימים
14. הצג מטבלת Products את שם המוצר , מחיר ליחידה , והמחיר ליחידה + 10

15. הצג מטבלת המוצרים את העמודות הבאות:

קוד מוצר
שם מוצר
מחיר ליחידה
מחיר לאחר העלאה של 16.5% (יש לתת לעמודה כינוי מתאים)
מספר מוצרים במלאי
מספר מוצרים מוזמנים.
ההפרש בין מספר המוצרים במלאי למספר המוצרים המוזמנים

שאלת אתגר

16. יש להציג מטבלת products את העמודות הבאות:

מספר המוצר,
שם מוצר,
ועלות המוצרים במלאי שלא הוזמנו (כלומר חישוב ההפרש בין מספר המוצרים במלאי
ומספר המוצרים שהוזמנו כפול מחיר ליחידה). יש לתת שם מתאים לעמודה.

WHERE

1. הצג מטבלת Employees את שמו הפרטי ושם המשפחה של עובד מספר 3
2. הצג מטבלת Products את שם המוצר ומחירו של מוצר מספר 4
3. הצג מטבלת Products את מספר המוצר, שם המוצר ומחירו עבור המוצרים אשר מחירם גבוה מ 20. סדר את התוצאות ע"פ מחיר (סדר עולה)
4. יש להציג מטבלת Employees את השם המלא בעמודה אחת, תאריך הלידה, ומספר העובד אליו הוא מדווח (ReportsTo) עבור עובד שמספרו 8.
5. הצג מטבלת Employees את קוד העובד, שמו המלא, ותאריך לידה עבור העובדים אשר גרים בעיר LONDON. יש לתת כינויים מתאים לעמודות.
6. הצג מטבלת Products את כל הפרטים עבור מוצרים אשר מחירם אינו בין 50 ל- 100.
7. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים אשר מחירם נע בין 21.35 לבין 43.9, מניין את התוצאות עפ"י המחיר (סדר יורד)
8. הצג מטבלת Employees את קוד העובד, שם המשפחה ותאריך תחילת עבודה עבור העובדים אשר גרים בערים LONDON או TACOMA.
9. הצג מטבלת Employees את מספר העובד, שמו הפרטי ושם משפחתו עבור העובדים אשר מספר העובד שלהם שווה ל 1, 2, 5
10. הצג מטבלת Employees את שמו הפרטי של העובד, שם משפחתו, ותאריך לידתו עבור העובדים אשר מספר העובד שלהם לא שווה לערכים 4, 5, 7
11. הצג מטבלת Products את מספר המוצר, שם המוצר ומספר הקטגוריה עבור המוצרים אשר מספר הקטגוריה שלהם שונה מ- 1, 2 או 7. מניין את התוצאות עפ"י מספר קטגוריה (סדר עולה)
12. הצג מטבלת Employees את שמם הפרטי של העובדים והאיזור שלהם, עבור העובדים אשר ערך האיזור שלהם (Region) הוא NULL
13. יש להציג מטבלת products את שמות המוצרים ומחירים עבור שלושת המוצרים היקרים ביותר.
14. יש להציג מטבלת Orders את מספרי ההזמנות, תאריך ההזמנה ותאריך הדרישה עבור כל ההזמנות שתאריך הדרישה שלהם אחרי חודש אוקטובר 1996.
15. יש להציג מטבלת employees את מספר העובדים, שם המשפחה שלהם ולמי הם מדווחים, רק עבור אותם עובדים שיש להם מנהל (כלומר יש להם למי לדווח). יש למיין את התוצאות ע"פ מספר עובד בסדר עולה

16. יש להציג מטבלת categories את כל הפרטים של קטגוריות שיש להן את האות 'ס' בשם.
17. הצג מטבלת Customers את שם החברה ומדינתה, עבור החברות שהאות האחרונה בשמם היא 'a'.
18. הצג מטבלת Products את שם המוצר ומספר הקטגוריה שלו, עבור המוצרים שבשמם האות 'a' היא אות אחת לפני הסוף.
19. יש להציג מטבלת Orders את מספר ההזמנה, מספר הלקוח ומספר העובד עבור ההזמנות שהתבצעו מחודש אפריל עד חודש מאי בשנת 1997.
יש לבצע מיון לפי תאריך הזמנה בסדר עולה ומיון משני לפי מספר לקוח בסדר יורד.
20. יש להציג מטבלת Customers את מספר הלקוח, שם החברה, ארץ, טלפון ואזור עבור הלקוחות שנמצאים במדינות ששמן מתחיל ב M,F או G, והאזור שלהם הוא NULL.
21. יש להציג מטבלת employees את מספר העובד, השם המלא, תאריך הלידה והמדינה עבור העובדים שבשם המשפחה שלהם מופיעה האות K או D או שנולדו בשנת 1963.
22. הצג מטבלת Products את שם המוצר, מחיר ליחידה ומס' הספק עבור המוצרים אשר מחירם גבוה מ- 30.
וגם מס' הספק שלהם הוא 1 או 3.

שאלות אתגר

23. יש להציג מטבלת Orders את מספר ההזמנה, מספר העובד, תאריך ההזמנה, תאריך הדרישה ושם המשלוח
עבור ההזמנות שעומדות בתנאים הבאים:
מספר העובד שלהם הוא 7
וגם שם המשלוח שלהם הוא אחד מהבאים:
QUICK-Stop
Du mond entire
Eastern Connection

וגם ההפרש בין תאריך הדרישה לתאריך ההזמנה גדול מ-20 יום.

24. יש להציג מטבלת products את מספר המוצר ושם המוצר עבור המוצרים ש:
שמספר הספק שלהם הוא 8,16 או 21.
או מחיר היחידה קטן מ 10.
בכל מקרה יש לכלול בתוצאות רק מוצרים שכמותם במלאי אינה בין 10 ל 100
יש למיין בסדר עולה לפי מחיר היחידה.

SCALAR FUNCTIONS

1. הצג מתוך טבלת Employees את שמו הפרטי של העובד באותיות קטנות , שם משפחתו באותיות גדולות עבור העובדים אשר מספר העובד שלהם בין 3 ל 5.
2. א. הצג מתוך טבלת Employees את שמו הפרטי של העובד , ובעמודה נוספת את מיקום האות 'a' בתוך שמו
ב. בצע סינון לשאילתה והצג רק את העובדים אשר אין להם את האות 'a' בשם
3. הצג מתוך טבלת Categories את שם הקטגוריה , את תיאור הקטגוריה ואת מיקום התו 'i' בתוך תיאור הקטגוריה 4 ספרות מתחילת המילה.
4. הצג מתוך טבלת Employees את שמו הפרטי של העובד, שם משפחתו ושם משתמש. שם המשתמש יורכב מ 3 אותיות ראשונות של שמו הפרטי יחד (שרשור) עם האות הראשונה של שם משפחתו
5. הצג מתוך טבלת Products את מספר המוצר, שם המוצר , ובעמודה נוספת שוב את שם המוצר כאשר כל תו '?' יוחלף בתו '-'
6. הצג את התאריך של היום.
7. ע"מ להציג את התאריך האחרון להוצאת כל הזמנה , הצג מתוך טבלת Orders את מספר הלקוח , מספר ההזמנה, תאריך ההזמנה , ואת תאריך ההזמנה פלוס 45 יום (השתמש בפונקצית תאריך על מנת לבצע את החישוב) .
8. הצג מתוך טבלת Employees את שמו הפרטי של העובד ואת גילו (ההפרש בשנים בין התאריך העכשווי ובין תאריך לידתו)
9. הצג מתוך טבלת Employees את שמו הפרטי של העובד, את היום בו התחיל לעבוד (לדוגמא ... sunday, monday), ואת השנה בה הוא התחיל לעבוד.
10. הצג מטבלת Products את העמודות ProductID ו- UnitPrice כפול 0.12 יש לעגל את התוצאה למספר שלם (בעיגול משתנה לפי המספר). תן שם מתאים לעמודה.
11. בחר מטבלת Employees את מספר העובד ושם המשפחה משורשרים יחדיו כשביניהן רווח (תן כינוי מתאים לעמודה זו), ואת תאריך הלידה בעמודה נפרדת.
12. הצג מטבלת Employees את שמות המשפחה באותיות גדולות בעמודה אחת ואת תאריך הלידה בפורמט של DD/MM/YY בעמודה שנייה, עבור העובדים אשר שם משפחתם מתחיל באות K או D. יש להשתמש בפונקציה SUBSTRING בתנאי הWHERE במקום האופרטור LIKE.

13. הצג מטבלת products את שתי העמודות הבאות:

מספר המוצר ומספר הספק באותה העמודה כאשר המילה 'AND' מופיעה ביניהן. תן את השם "PRODUCT" לעמודה (שים לב להתאמת ה data types),

מחיר ליחידה (unitprice) כפול 1.165 כאשר יש לעגל את התוצאה למספר שלם כלפי מטה. תן את השם "FULL PRICE" לעמודה המחושבת.

יש להציג רק את המוצרים שמחירם החדש גדול מ- 40.

14. הצג מטבלת employees את העמודות הבאות:
שם המשפחה משורשר עם אורכ שם המשפחה,
שם פרטי משורשר עם אורכ השם הפרטי.
יש לתת לכל עמודה בתצוגה כינוי מתאים.

15. הצג מטבלת employees את העמודות הבאות:
שם המשפחה,
שם המשפחה מהסוף להתחלה (תן כינוי מתאים לעמודה).
הצג את הנתונים רק עבור העובדים אשר יש להם מנהל (על פי עמודת ReportsTo).

16. הצג מטבלת Orders את מספר ההזמנה, תאריך ההזמנה ותאריך הדרישה עבור כל ההזמנות אשר מספר הרבעונים בין תאריך ההזמנה (OrderDate) לתאריך הדרישה (RequiredDate) שווה ל - 1.

17. הצג מטבלת הלקוחות את ארבעת האותיות הראשונות של שם החברה (CompanyName) עבור כל הלקוחות שה CompanyName שלהם מתחיל באות 'a'.

שאלת אתגר

18. הצג מטבלת Employees את העמודות הבאות:

- שם משפחה משורשר עם תאריך הלידה
- תאריך תחילת עבודה בפורמט 104
- קוד המנהל (ReportsTo), במידה ובעמודה זו מופיע NULL להציג את המחרוזת 'No Manager'

יש לכלול בתוצאות רק עובדים ששם המשפחה שלהם ארוך או שווה לשם הפרטי

JOIN

1. הצג את שם המוצר מתוך טבלת products , ואת שם הקטגוריה שלו מתוך טבלת categories
2. הצג את שם המוצר מתוך טבלת products ואת שם החברה של הספק שלו מתוך טבלת suppliers
3. הצג את מספר ההזמנה מתוך טבלת orders ואת שם החברה של הלקוח מתוך טבלת customers , עבור החברות ששמן מתחיל באות 'a'
4. הצג מתוך טבלת region את תיאור התחום (RegionDescription) ומתוך טבלת Territories את תיאור האיזור (TerritoryDescription)
5. הצג את שם המוצר ומחיר המוצר מתוך טבלת products , ואת שם הקטגוריה שלו מתוך טבלת categories , עבור המוצרים שמחירים גבוה מ – 50
6. הצג את מספר המוצר , מחיר המוצר , ואת מספר הספק מתוך טבלת products ואת שם הקטגוריה שלו מתוך טבלת categories , עבור המוצרים שמספר הספק שלהם הוא 3.
7. הצג את מספר המוצר , מחיר המוצר , ומספר הספק מתוך טבלת products ואת שם הקטגוריה שלו מתוך טבלת categories , עבור המוצרים שיש להם בשם הקטגוריה את האות a
8. הצג את שם המוצר מתוך טבלת products , את שם הקטגוריה מתוך טבלת categories , ואת שם החברה של הספק מתוך טבלת suppliers
9. הצג את שם המוצר מתוך טבלת products , את תיאור הקטגוריה מתוך טבלת categories (Description) , ואת עיר הספק מתוך טבלת suppliers , כאשר שם העיר של הספק היא London או Tokyo
10. הצג את מספר המוצר מתוך טבלת products , את תיאור הקטגוריה מתוך טבלת categories , ואת הארץ מתוך טבלת suppliers , כאשר האות הראשונה של הארץ מתחילה באות a .
11. הצג את שם החברה של הלקוח מתוך טבלת customers ואת מספר ההזמנה מתוך טבלת orders עבור הלקוחות , גם אלו ללא הזמנות.
12. הצג מספר הזמנה (OrderID) , תאריך הזמנה (OrderDate) , וכתובת למשלוח (ShipAddress) מטבלת ORDERS , ובנוסף את קוד הלקוח (CustomerID) , שם הלקוח (CompanyName) ומספר הטלפון שלו (Phone) מטבלת Customers . יש לכלול בתוצאה רק הזמנות שבוצעו בשנת 1996 , ורק לקוחות שקוד הלקוח שלהם (CustomerID) מתחיל באותיות A או C

13. חזור על השאלה הקודמת,
הוסף לה גם את עמודות שם פרטי (FirstName) ושם משפחה (LastName)
מטבלת employees
תן שמות מתאימים לעמודות.
סדר את התוצאות עם תאריך ההזמנה בסדר יורד.

14. א. הצג את שם משפחתו של העובד ושם משפחתו של מנהלו, בסס את תשובתך
על סמך הקשר הפנימי בתוך טבלת Employees בין מספר המנהל
ומספר העובד
ב. הצג גם את העובדים ללא המנהלים.

שאלת אתגר

15. הצג מטבלת המוצרים, בעזרת join, את מספר המוצר (ProductID), שם המוצר (ProductName),
ומחירו (UnitPrice) עבור מוצרים שעולים יותר ממוצר בשם Alice Mutton.

GROUP FUNCTIONS

1. הצג מטבלת Employees את שם המשפחה הקטן ביותר מבחינה אלפאבטית
2. הצג מטבלת Employees את השם הפרטי הגדול ביותר מבחינה אלפאבטית
3. הצג מטבלת Employees את מספר הרשומות שיש בטבלה
4. הצג מטבלת Employees את מספר הרשומות שיש בעמודת Region (לא כולל NULL)
5. הצג מטבלת Products את ממוצע המחיר ליחידה
6. הצג מטבלת Products את המחיר (UnitPrice) הגבוה ביותר, ואת המחיר הממוצע. תן שמות מתאימים לעמודות.
7. הצג מטבלת Employees את תאריך יום ההולדת הנמוך ביותר ותאריך יום ההולדת הגבוה ביותר. יש להציג את התאריכים בפורמט 113 תן שמות מתאימים לעמודות.
8. הצג את מספר הלקוחות השונים הקיימים בטבלת Customers, תן שם מתאים לעמודה.
9. הצג את מספר הלקוחות השונים הקיימים בטבלת Orders, תן שם מתאים לעמודה. שים לב, הפעם יתכן שלקוח מסויים ביצע יותר מהזמנה אחת.
10. הצג מטבלת Products את המחיר (UnitPrice) המקסימאלי, המינימאלי, והממוצע למוצר עבור כל קטגוריה (CategoryID). תן שמות מתאימים לעמודות.
11. הצג מטבלת Products את המחיר הגבוה ביותר של המוצר לפי כל מספר ספק, מיין לפי מספר הספק (סדר יורד)
12. הצג מטבלת Products את ממוצע היחידות במלאי לפי כל מספר ספק מיין לפי ממוצע היחידות (סדר יורד)
13. הצג מטבלת Customers את מספר הלקוחות לפי המדינה, ולפי העיר
14. הצג מטבלת Products את המחיר הממוצע של המוצרים לפי כל קטגוריה, עבור המוצרים שהמחיר שלהם גדול מ 40
15. הצג מטבלת Customers את מספר הלקוחות לפי עיר, עבור הלקוחות אשר גרים ב: London, Berlin, Paris או Rio de Janeiro.

16. הצג מתוך טבלת Products את המחיר הגבוה ביותר, המחיר הנמוך ביותר, המחיר הממוצע, וכמות המוצרים, לפי כל מספר קטגוריה ומספר ספק

17. הצג מטבלת Products את המחיר המקסימלי לפי כל קטגוריה, עבור המוצרים אשר המחיר המקסימלי שלהם גדול מ 40

18. הצג מטבלת Products את המחיר הממוצע לפי כל ספק, עבור המוצרים אשר המחיר הממוצע שלהם גדול מ 40

19. הצג מטבלת Products את סך כל הפריטים שהוזמנו (UnitsOnOrder), ואת סך כל היחידות במלאי (UnitsInStock) עבור כל קטגוריה. בנוסף, הצג את שם הקטגוריה (מטבלת Categories). יש לכלול בתוצאות רק קטגוריות שבשמן מופיעה האות C, ורק מוצרים שסך כל היחידות המוזמנות מהם גדול מ 100. יש למיין את התוצאות ע"פ שם הקטגוריה (סדר עולה).

שאלות אתגר

20. הצג מטבלת Customers את האזור, העיר ומספר הלקוחות השייכים לאותו האזור עבור אותן הערים שקיימות בשמן האותיות 'M' או 'L' וגם האזור אינו NULL. יש לכלול בתוצאות רק אזורים שמספר הלקוחות בהם גדול או שווה ל 2.

21. הצג את העמודות הבאות:

- שם עובד (LastName) מטבלת Employees

- סך כל ההזמנות שהעובד ביצע (מטבלת Orders)

- התאריך האחרון שהעובד ביצע הזמנה (OrderDate מטבלת Orders)

תן שמות מתאימים לעמודות

יש לכלול בתוצאות רק עובדים שביצעו מעל 100 הזמנות

SUBQURIES

1. הצג מטבלת Products את שמות המוצרים אשר מחירם **נמוך** מהמחיר של מוצר מספר 8.
2. הצג מטבלת Products את שמות המוצרים ומחירם, עבור המוצרים אשר מחירם גבוה יותר ממוצר ששמו Tofu
3. הצג מטבלת employees את שמות העובדים ותאריך גיוסם עבור העובדים שגויסו **לאחר** עובד שמספרו 6.
4. הצג מטבלת Products את מספר המוצר, שם המוצר ומחיר יחידה עבור המוצרים אשר מחירם **גבוה** מהמחיר הממוצע ליחידה.
5. הצג מטבלת Products את שמות המוצרים וכמותם במלאי, עבור המוצרים אשר כמותם במלאי **נמוכה** יותר מהכמות המינימלית בקטגוריה מספר 5
6. הצג מטבלת Products את כל פרטי המוצרים שנמצאים **באותה קטגוריה** כמו מוצר ששמו Chai. אין להציג בתוצאה הסופית את המוצר chai עצמו.
7. הצג מטבלת Products את שם המוצר, מחירו ומספר הקטגוריה שלו, עבור המוצרים שמחירם **שווה** לאלו של קטגוריה מס' 5
8. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים אשר מחירם גבוה **מלפחות** אחד המוצרים בקטגוריה מספר 5
9. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים אשר מחירם גבוה **מכל** המוצרים של קטגוריה מספר 5
10. הצג מטבלת Orders את מספרי ההזמנות ותאריך ביצוע ההזמנות עבור כל ההזמנות שהלקוחות שלהן מ-Germany, France או מ-Sweden **וגם** תאריך ההזמנה היה בשנת 1997 (יש לשים לב לכמה שורות מחזירה התת - שאילתא).
11. הצג מטבלת products את שמות המוצרים ואת הקוד שלהם (productID) יש לכלול בתוצאה רק המוצרים שמחירם **גדול** מהמחיר הממוצע של המוצרים אשר מספר היחידות במלאי (UnitsInStock) **גדול** מ-50.
12. הצג מטבלת products את שמות כל המוצרים אשר **שם** הקטגוריה שלהם הוא Beverages או Condiments **וגם** אזור (region) הספק אינו ידוע.
13. הצג את שמות החברות (CompanyName מטבלת Suppliers) המספקות מוצרים מקטגוריית Beverages (עמודת CategoryName בטבלת Categories)

Set Operators

1. הצג את רשימת הערים מתוך טבלת עובדים
2. הצג את רשימת הערים מתוך טבלת לקוחות
3. הצג את השילוב הייחודי של הערים בטבלת לקוחות ועובדים
4. הצג את כל הערים בטבלת לקוחות ועובדים כולל ערים אשר חוזרות על עצמן
5. הצג את הערים המשותפות לטבלת לקוחות ועובדים
6. הצג את הערים אשר מופיעות בטבלת עובדים ולא מופיעות בטבלת לקוחות

Analytic Functions

1. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מס' הקטגוריה שלו וסך כל המחירים עבור כל המוצרים בתוך טבלה זו.
2. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו וסך כל המחירים עבור מס' הקטגוריה של אותו מוצר.
3. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו וסך כל המחירים עבור מס' הקטגוריה של אותו מוצר.
הצג את המוצרים בכל קטגוריה ממויינים לפי מחירם - מהגבוה לנמוך
4. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו וסך כל המחירים עבור מס' הקטגוריה של אותו מוצר.
הצג את המוצרים בכל קטגוריה ממויינים לפי מחירם - מהגבוה לנמוך
כאשר סכימת המחירים תתבצע מהשורה הראשונה של אותה קטגוריה עד השורה האחרונה.
5. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו וסך כל המחירים עבור מס' הקטגוריה של אותו מוצר.
הצג את המוצרים בכל קטגוריה ממויינים לפי מחירם - מהגבוה לנמוך
כאשר סכימת המחירים תתבצע מהשורה הראשונה של אותה קטגוריה עד השורה הנוכחית.
6. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו, דירוגו לפי מחיר באמצעות פונקציית Rank, דירוגו לפי מחירו באמצעות פונקציית Dense_Rank, מספר השורה, ומספר קבוצתו בהנחה שהשורות בטבלה חולקו ל 10 קבוצות.
7. מתוך טבלת מוצרים הצג את שם המוצר, מחירו, מספר הקטגוריה שלו, דירוגו בתוך הקטגוריה בה הוא נמצא לפי מחירו באמצעות פונקציית Rank, דירוגו בקטגוריה בה הוא נמצא לפי מחירו באמצעות פונקציית Dense_Rank, מספר השורה, ומספר קבוצתו בהנחה שהשורות בטבלה חולקו ל 10 קבוצות.
8. מתוך טבלת מוצרים הצג את מספר המוצר, שם המוצר, מחירו, מחיר המוצר הקודם לו (מיון לפי מחיר המוצר), מחיר המוצר הבא אחריו (מיון לפי מחיר המוצר) עבור המוצרים אשר מספר הקטגוריה שלהם היא 1. מייין את תוצאות השאילתה לפי מחיר המוצר.

DML

1. מהי המשמעות של @@IDENTITY ?

2. יש ליצור טבלה על ידי העתקת הקוד הבא :

```
CREATE TABLE my_employees (
  id INT PRIMARY KEY ,
  name VARCHAR (50),
  title VARCHAR(50),
  deptid INT,
  salary MONEY DEFAULT 3500)
```

* במידה והתקבלה הודעה כי הטבלה כבר קיימת יש לבצע DROP לטבלה:

```
DROP TABLE my_employees
```

ואז ליצור אותה מחדש.

3. שלוף את תיאור הטבלה

4. להלן נתונים עליהם בהם יש להיעזר בתרגילים הבאים.

ID	NAME	TITLE	DEPTID	SALARY
1	Aviv Cohen	Clerk	10	4000
2	Miriam levi	Sales Manager	20	3750
3	AION Romano	OperatiON Manager	30	NULL
4	Baruch Nave	NULL	30	3500
5	Danny SalomON	Sales Representative	20	7000

לפני תחילת ההזנה יש לפתוח טראנסאקציה.

5. הכנס את הרשומה הראשונה בלבד מהנתונים לדוגמא, ללא ציון רשימת העמודות בפקודת ה INSERT.

6. הכנס את הרשומה השנייה מהנתונים לדוגמא, הפעם, יש לציין במפורש את רשימת העמודות בפקודת ה INSERT.

7. הכנס את הרשומה השלישית תוך שימוש במילה השמורה NULL.

8. הכנס את מהרשומה הרביעית ערכים רק לעמודות ID, Name ו-DeptID. בדוק מה הרשומה שנכנסה, האם היא תואמת את הרשומה בנתונים לדוגמא? למה?

9. יש להזין את הרשומה האחרונה.

10. בדוק כי הרשומות נוספו.

11. עובד מספר 2 תפקד מצוין החודש והוחלט להעלות את משכורתו ל- 4500 יש לבצע את השינוי בטבלה.

12. יש לשנות את שמו של עובד מספר 4 לשמך ואת מספר המחלקה ל- 20.

13. יש לוודא כי השינויים בוצעו.

14. הוחלט על קיצוץ מחלקה 30, על כן יש להעביר את כל העובדים במחלקה 30 למחלקה 10.

15. העובד AION Romano אינו מתפקד כראוי, לכן הוחלט לפטרו. יש למחוק את הרשומה מהטבלה.

16. בדוק כי השינויים בוצעו.

17. הוחלט על איחוד חברות. יש להזין את העובדים מטבלת Employees לטבלת my_employees. יש להזין רק עובדים שמספר העובד שלהם (EmployeeID) גדול מ 5. יש להתאים את העמודות כאשר בעמודת DEPTID יוזנו NULL ובעמודת SALARY יוזנו ערכי DEFAULT

18. ודא כי כל השינויים בוצעו בהצלחה

19. שמור את השינויים באופן סופי ב - Data Base

יצירת DATA BASE ופקודות DDL ליצירת טבלה

1. יש ליצור DATA BASE חדש בשם: human_resource_db באמצעות כלי ה - SQLSERVER MANAGEMENT STUDIO.
2. יש ליצור 2 טבלאות – טבלת עובדים בשם employees וטבלה נוספת של מחלקות בשם departments.
את טבלת ה - departments יש ליצור דרך הכלי כאשר הטבלה תכלול את השדות והאילוצים הבאים:
 - department_id – מסוג INT, אילוץ PRIMARY KEY כולל IDENTITY המתחיל ב 10 וגדל ב 10.
 - department_name – מסוג VARCHAR(20) אילוץ NOT NULL.
 - location – מסוג VARCHAR(30) אילוץ NOT NULL.
- את טבלת ה - employees יש ליצור באמצעות קוד כאשר הטבלה תכלול את השדות והאילוצים הבאים:
 - Employee_id – מסוג INT, אילוץ PRIMARY KEY כולל IDENTITY המתחיל ב 1 וגדל ב 1.
 - Last_name – מסוג VARCHAR(20), אילוץ NOT NULL.
 - Hire_date – מסוג DATETIME ערך DEFAULT – התאריך הנוכחי, אילוץ NOT NULL.
 - Salary – מסוג MONEY ערך DEFAULT 5000.
 - Email – מסוג VARCHAR(30), אילוץ NULL + UNIQUE.
3. לטבלת ה - employees יש להוסיף את העמודה המייצגת את הקשר בין העובדים למחלקות:
 - Department_id – מסוג INT, FOREIGN KEY לעמודת department_id בטבלת departments.
4. יש ליצור טבלת emp_history המבוססת על מבנה טבלת ה - employees.
5. יש למחוק את טבלת emp_history (נתונים ומבנה).

VIEWS

1. יש ליצור VIEW בשם emp_vw, המציג מטבלת Employees את העמודות הבאות: EmployeeID, LastName, FirstName ו- address עבור העובדים מהעיר Tacoma או London.
2. יש ליצור VIEW בשם pro_vw, המציג מטבלת Products עבור קטגוריות 1,2 ו-4 את מחיר המוצרים הממוצע (UnitPrice).
3. יש לשנות את ה- VIEW כך שיוצגו שמות קטגוריות (CategoryName מטבלת Categories) במקום מספרי קטגוריות, וכן לכלול ב- view ורק הקטגוריות שהמחיר הממוצע שלהן גדול מ-25.
4. צור VIEW בשם cust_orders_vw המציג את סך ההזמנות שביצע לקוח (הצג את השדות CompanyName וחישוב מספר ההזמנות).
5. כתוב שאילתה (select) המציגה את שמות הלקוחות שביצעו יותר מ-10 הזמנות, כשהשאלתה תשלוף נתונים מ-View שיצרת בסעיף הקודם.
6. צור View בשם Prods המציג את העמודות הבאות:
 - ProductID
 - ProductName
 - UnitPrice
 - ההפרש בין סך כל הפריטים במלאי (UnitsInStock) ו- סך כל הפריטים המוזמנים (UnitsOnOrder), קרא לעמודה זו "UnitsLeft"יש לכלול רק מוצרים שעולים פחות מ-30.
7. פתח טראנסאקציה ובצע את הפעולות הבאות (על ה- View)
 - א. עדכן את שם מוצר מספר 52 ל- 'Jiffa'.
 - ב. עדכן את מחירו של מוצר זה ל-31.
 - ג. נסה לשלוף את מוצר מספר 52 מה- view. מה קרה? למה?
 - ד. נסה לעדכן את עמודת UnitsLeft של מוצר מספר 1 ל-41. האם הצלחת? מדוע?בטל את כל השינויים.
8. שנה את ה- view מהסעיף הקודם כך שיכלול את המשפט With check option.
9. פתח טראנסאקציה חזור על סעיפים א' ו- ב' של שאלה 7- מה קרה ולמה?
* בטל את השינויים לסיום.

BASIC T- SQL

1. הגדר שני משתנים מסוג מחרוזת, יש להשים להם את הערך 'MY FIRST' לאחד מהם ו-
'VARIABLE' לשני. יש להדפיס את שני המשתנים משורשרים יחדיו לחלונת ה - MESSAGES.
2. יש להגדיר משתנה אחד מסוג מחרוזת אשר יכיל את המחרוזת הבאה:
'THE DATE TODAY IS:' ומשתנה נוסף מסוג תאריך אשר יכיל את התאריך הנוכחי.
יש לשרשר בין שני המשתנים ולהדפיסם לחלונת ה- MESSAGES.
3. קלוט לתוך משתנה מטיפוס מתאים את שם משפחתו של עובד מספר 4 והדפס אותו לחלונת ה-
MESSAGES.
4. קלוט לתוך משתנים מטיפוסים מתאימים את שם מוצר מספר 5 ואת שם הקטגוריה אליה הוא שייך.
הדפס את שני המשתנים לחלונת ה - MESSAGES.
5. הגדר משתנה אשר יכיל את מחירו של מוצר מספר 9, אם מחיר המוצר גדול מ 50 יש להדפיס הודעה
בחוץ ה MESSAGE :
' I LIKE THIS PRODUCT'
אחרת, יש להדפיס הודעה בחלון ה RESULT :
' I DO NOT LIKE THIS PRODUCT'
- * הרץ את הקוד שוב, הפעם עם מוצר מספר 17.
6. צור קוד המכיל שני משתנים: שם מוצר , ומחיר מוצר. אתחל את משתנה המחיר ע"פ הערך במשתנה
השם, בעזרת שאילתה על טבלאת Products.
בדוק את המחיר שהתקבל והוסף בנוס בהתאם לתנאים הבאים:
- אם המחיר קטן מ - 20 יש להוסיף למשתנה 10% ולהדפיס הודעה מתאימה עם המחיר החדש.
- אם המחיר גדול או שווה ל - 20 וקטן מ - 40 יש להוסיף למשתנה 20% ולהדפיס הודעה מתאימה
עם המחיר החדש.
אם מחיר המוצר שווה או גדול מ - 40 יש להוסיף למשתנה 50% ולהדפיס הודעה מתאימה עם
המחיר החדש.
- בחן את הקוד עם המוצרים הבאים: 'CHAI', 'TOFU' ו- 'Tarte au sucre'
7. קלוט לתוך משתנה מטיפוס מתאים לשם משפחה של עובד.
אם לעובד יש את האות E בשם הדפס את שם המשפחה (אם אין האות אי- אל כלום).
רוץ בעזרת לולאת While על שמות כל העובדים (מעובד שה- EmployeeID שלו הוא 1 ועד עובד
שה- EmployeeID שלו הוא 9)
8. הדפס את פרטי קטגוריה מספר 4 על ידי שימוש בלולאת GOTO. אם בשם הקטגוריה מופיעה האות
A המשך בלולאה לקטגוריה הבאה. אם לא צא מהלולאה והדפס הודעה מתאימה.

ERROR HANDLING

1. בצע מחיקה של עובד מספר 2 מטבלת EMPLOYEES.
 - א. מדוע לא התבצעה המחיקה?
 - ב. בצע טיפול בשגיאה על ידי שימוש ב TRY ו CATCH כך שתופיע הודעה מתאימה למשתמש.
2. בצע עדכון של מוצר מספר 5 מטבלת PRODUCTS לקטגוריה מספר 11.
טפל בשגיאה באמצעות TRY ו CATCH כך שבחלון ה - RESULT תופיע מספר השגיאה והודעת השגיאה.
3. עדכן בתוך בלוק TRY ו- CATCH את עירו של עובד מספר 17 ל- ASHKELON. מכיוון שלא קיים עובד מספר 17, יש ליזום הודעת שגיאה אשר תוציא הודעה בהתאם (ניתן להשתמש ב @@ROWCOUNT - שמחזיר את מספר הרשומות שעודכנו).
4. הוסף שגיאה ידנית לניסיון מחיקה של רשומה אשר לא ניתן למחוק כשאר ההודעה אשר תופיע תהיה :
' AN ERROR ACCORD, CAN NOT DELETE RECORD'
5. תחת בלוק של TRY ו- CATCH בצע מחיקה של קוד לקוח 'ALFKI' ובמידה ותתבצע שגיאה יש להעלות את השגיאה שיצרת בסעיף 4.

STORED PROCEDURES

1. צור פרוצדורה בשם prod_det המציגה עבור מוצר מספר 6 את שמו, ושם הקטגוריה אליו הוא שייך.
2. שנה את הפרוצדורה מסעיף 1 כך שהפרוצדורה תקבל את מספר מוצר שונה בכל הרצה.
3. כתוב פרוצדורה בשם top_price_prod שתציג את פרטי עשרת המוצרים היקרים ביותר (יש להציג את קוד המוצר, שמו ומחירו).
4. שנה את הפרוצדורה מתרגיל 3 כך שהפרוצדורה תקבל פרמטר מספרי שונה בכל הרצה ותציג את המספר המבוקש של מוצרים יקרים ביותר. כך למשל אם הפרוצדורה תקבל את המספר 5 היא תציג את חמשת המוצרים היקרים ביותר.
5. צור פרוצדורה בשם emp_det, אשר תקבל את מספר עובד ותציג את הפרטים הבאים עבור העובד:
שם המשפחה, תאריך הלידה והעיר.
במידה ובשם המשפחה של העובד קיימת האות a.
אם לא קיימת האות a הפרוצדורה תציג את ההודעה:
'EMPLOYEE NUMBER: ' ומספר העובד 'HAS NO LETTER "A" ' ושם העובד.
יש להריץ את הפרוצדורה עבור עובדים מספר 4 ו-7.
6. צור פרוצדורה בשם Cust_det המקבלת קוד לקוח (שים לב עמודת CustomerID היא מחרוזתית) ומציגה את הפרטים הבאים:
Address, Country, CompanyName – לחלון ה-Results.
7. שנה את הפרוצדורה מסעיף 6 ולהוסיף לה את הבדיקה הבאה:
אם הלקוח מ-Mexico, יש להדפיס הודעה "הלקוח צרפתי",
אם הלקוח מ-Germany יש להדפיס הודעה "הלקוח גרמני", אחרת הדפס את שם הלקוח ופרטיו
לחלון ה results.
הרץ את הפרוצדורה עבר הלקוחות הבאים:
'ALFKI' 'ANTON' 'BERGS'.
8. א. צור פרוצדורה אשר תקבל קוד לקוח כפרמטר נכנס ותחזיר את סך כל ההזמנות שהלקוח ביצע כפרמטר יוצא. הפרוצדורה תבדוק את סך ההזמנות המתקבל. אם סך ההזמנות גדול מ-10 תודפס הודעה בהתאם (לחלון ה messages), אחרת יוצגו פרטי ההזמנות של אותו לקוח (בחלון ה results).
ב. קרא לפרוצדורה מתוכנית בעלת פרמטרים מתאימים עם קוד לקוח 'ALFKI' ו-'BERGS'.

SELECT

1.
SELECT *
FROM orders
-- 830 rows
2.
SELECT *
FROM employees
-- 9 rows
3.
SELECT FirstName , HireDate , Region , Country
FROM employees
-- 9 rows
4.
SELECT CustomerID , OrderID , OrderDate
FROM orders
-- 830 rows
5.
SELECT ProductID AS 'Proid' , ProductName AS 'ProNm' , unitPrice AS 'Untpr'
FROM products
-- 77 rows
6.
SELECT Address AS 'add' , City AS 'ct' , Region AS 'reg'
FROM Employees
-- 9 rows
7.
SELECT customerid, address + ' ' + city AS 'full address'
FROM customers
-- 91 rows

.8

```
SELECT LastName+' '+ FirstName AS 'Full name', BirthDate +8 as 'Birth date', ReportsTo
As 'Manager#'
FROM Employees
-- 9 rows
```

.9

```
SELECT distinct city
FROM employees
-- 5 rows
```

.10

```
SELECT distinct country
FROM employees
--2 rows
```

.11

```
SELECT distinct title
FROM employees
-- 4 rows
```

.12

```
SELECT Country , City
FROM Customers
-- 91 rows
```

```
SELECT Distinct Country , City
FROM Customers
-- 69 rows
```

.13

```
SELECT firstName , birthDate , birthDate + 5
FROM employees
-- 9 rows
```

.14

```
SELECT productName , unitPrice , unitPrice + 10
FROM products
-- 77 rows
```


.15

```
SELECT    ProductID,  
          ProductName,  
          UnitPrice,  
          UnitPrice*1.165 AS 'After Raise',  
          UnitsInStock,  
          UnitsOnOrder,  
          UnitsInStock - UnitsOnOrder AS 'Units Left'  
FROM Products  
-- 77 rows
```

.16

```
SELECT    productID,  
          ProductName,  
          (unitsINstock - unitsONOrder)* unitprice AS 'Price for UnOrderd Units'  
FROM products  
-- 77 rows
```

WHERE

- .1
SELECT lastName , FirstName
FROM employees
WHERE employeeID = 3
-- 1 row
- .2
SELECT productName , UnitPrice
FROM products
WHERE productID = 4
-- 1 row
- .3
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice > 20
ORDER BY UnitPrice ASC
-- 37 rows
- .4
SELECT firstname + ' ' +lastname AS 'full Name', birthdate, reportsto
FROM employees
WHERE employeeid = 8
-- 1 row
- .5
SELECT EmployeeID, Lastname+' '+FirstName As 'Full Name', BirthDate
FROM Employees
WHERE City = 'LONDON'
-- 4 rows
- .6
SELECT *
FROM Products
WHERE UnitPrice NOT BETWEEN 50 AND 100
- .7
SELECT productName , UnitPrice
FROM products
WHERE UnitPrice BETWEEN 21.35 AND 43.9
ORDER BY UnitPrice DESC
-- 24 rows

.8

```
SELECT EmployeeID, Lastname, HireDate
FROM Employees
WHERE City IN ('London', 'Tacoma')
-- 5 rows
```

.9

```
SELECT employeeID , lastName , FirstName
FROM employees
WHERE employeeID IN (1,2,5)
-- 3 rows
```

.10

```
SELECT lastName , FirstName , BirthDate
FROM employees
WHERE employeeID NOT IN (4,5,7)
-- 6 rows
```

.11

```
SELECT ProductID, ProductName, CategoryID
FROM Products
WHERE CategoryID NOT IN (1, 2, 7)
ORDER BY CategoryID
-- 48 rows
```

.12

```
SELECT firstName , region
FROM employees
WHERE region IS NULL
-- 4 rows
```

.13

```
SELECT TOP 3 ProductName, UnitPrice
FROM products
ORDER BY UnitPrice DESC
-- 3 rows
```

.14

```
SELECT OrderID, OrderDate, RequiredDate
FROM orders
WHERE RequiredDate > '1996-10-31'
-- 755 rows
```

.15

```
SELECT employeeid, lastname, ReportsTo
FROM employees
WHERE ReportsTo IS NOT NULL
ORDER BY employeeID
-- 8 rows
```

.16

```
SELECT *
FROM categories
WHERE categoryname LIKE '%o%'
-- 6 rows
```

.17

```
SELECT companyName , Country
FROM customers
WHERE companyName LIKE '%a'
-- 7 rows
```

.18

```
SELECT productName , categoryId
FROM products
WHERE productName LIKE '%a_'
-- 7 rows
```

.19

```
SELECT OrderID, CustomerID, EmployeeID
FROM orders
WHERE OrderDate BETWEEN '1997-04-01' AND '1997-05-31'
ORDER BY OrderDate ASC, CustomerID DESC
-- 63 rows
```

.20

```
SELECT CustomerID, CompanyName, Country, Phone , region
FROM Customers
WHERE (Country LIKE 'm%'
      OR Country LIKE 'f%'
      OR Country LIKE 'g%')
AND
      Region IS NULL
-- 29 rows
```

.21

```
SELECT EmployeeID, FirstName+' '+LastName AS 'Full Name', BirthDate, Country
FROM employees
WHERE lastname LIKE '%k%'
      OR lastName LIKE '%d%'
      OR birthdate between '1963-01-01' and '1963-12-31'
-- 6 rows
```

.22

```
SELECT productName , UnitPrice , SupplierID
FROM products
WHERE UnitPrice > 30
AND
    (SupplierID = 1 OR SupplierID = 3)
-- 1 row
```

.23

```
SELECT orderid, employeeID , orderdate, requireddate , shipName
FROM orders
WHERE employeeid = 7
AND shipname IN ('QUICK-Stop', 'Du monde entier','Eastern Connection')
AND orderDate + 20 < requireddate
--5 row
```

.24

```
SELECT ProductID, ProductName
FROM products
WHERE (SupplierID IN (8,16,21 ) OR UnitPrice < 10)
AND UnitsInStock NOT BETWEEN 10 AND 100
ORDER BY UnitPrice ASC
-- 6 rows
```

SCALAR FUNCTIONS

1.

```
SELECT LOWER(firstName) , UPPER(lastName)
FROM employees
WHERE employeeID BETWEEN 3 AND 5
-- 3 rows
```
2.

```
SELECT firstName , CHARINDEX('a' , firstName)
FROM employees
-- 9 rows
```



```
SELECT firstName , CHARINDEX('a' , firstName)
FROM employees
WHERE CHARINDEX('a' , firstName) = 0
-- 2 rows
```
3.

```
SELECT categoryName , Description , CHARINDEX('i' , Description , 4)
FROM categories
-- 8 rows
```
4.

```
SELECT firstName , LastName,
SUBSTRING(firstName, 1 , 3) + SUBSTRING(lastName ,1 ,1) AS 'userName'
FROM employees
-- 9 rows
```
5.

```
SELECT productID , ProductName , REPLACE(productName , '?' , '-')
FROM products
-- 77 rows
```
6.

```
SELECT GetDate()
```
7.

```
SELECT CustomerID , OrderID , orderDate , DATEADD(dd , 45 , orderDate)
FROM orders
-- 830 rows
```

.8

```
SELECT FirstName , DATEDIFF(yyyy, birthDate , getdate()) AS 'age'
FROM employees
-- 9 rows
```

.9

```
SELECT FirstName , DATENAME(dw, hireDate) , YEAR(hireDate)
FROM employees
-- 9 rows
```

.10

```
SELECT ProductID, ROUND(UnitPrice*0.12,0) AS 'New price'
FROM Products
-- 77 rows
```

.11

```
SELECT
    LastName+ ' '+CAST(employeeID AS VARCHAR) AS 'name and num' ,
    birthDate
FROM Employees
-- 9 rows
```

.12

```
SELECT UPPER(LastName) AS 'Last name',
CONVERT(CHAR(12), BirthDate, 103) AS 'Birth date'
FROM Employees
WHERE SUBSTRING(LastName, 1, 1) IN ('K', 'D')
-- 3 rows
```

.13

```
SELECT CAST(ProductID AS varchar(12))+ ' AND ' + CAST(SupplierID AS char(12))AS
'product',
FLOOR(UnitPrice*1.165) "full price"
FROM products
WHERE FLOOR(UnitPrice*1.165) > 40
-- 17 rows
```

.14

```
SELECT
    LastName + CAST(len(LastName) AS varchar(2)) AS 'last name AND length',
    FirstName + CAST(len(FirstName) AS varchar(2)) AS 'first name AND length'
FROM employees
-- 9 rows
```

.15

```
SELECT LastName name, REVERSE(LastName) AS 'reverse name'
FROM employees
WHERE ReportsTo IS NOT NULL
-- 8 rows
```

.16

```
SELECT OrderID, OrderDate, RequiredDate
FROM Orders
WHERE DATEDIFF(q, OrderDate, RequiredDate) = 1
-- 249 rows
```

.17

```
SELECT SUBSTRING(companyName, 1, 4)
FROM customers
WHERE companyName LIKE 'a%'
-- 4 rows
```

.18

```
SELECT
    LastName + ' ' + CONVERT(char(25), BirthDate) AS 'name+birthdate',
    CONVERT(char(25), HireDate, 104) AS 'hire date',
    ISNULL(CONVERT(char(25), ReportsTo), 'no manager') AS 'manager?'
FROM employees
WHERE LEN(LastName) >= LEN(FirstName)

-- 6 rows
```


JOIN

.1

```
SELECT pro.productName , cat.CategoryName  
FROM products pro JOIN categories cat  
ON pro.categoryID = cat.categoryID  
--77 rows
```

.2

```
SELECT pro.productName , sup.companyName  
FROM products pro JOIN suppliers sup  
ON pro.supplierID = sup.supplierID  
--77 rows
```

.3

```
SELECT ord.orderID , cust.companyName  
FROM orders ord JOIN customers cust  
ON ord.customerID = cust.customerID  
WHERE cust.companyName LIKE 'a%'  
-- 30 rows
```

.4

```
SELECT reg.RegionDescription , ter.TerritoryDescription  
FROM Territories ter JOIN Region reg  
ON reg.RegionID = ter.RegionID  
--53 rows
```

.5

```
SELECT pro.productName , pro.UnitPrice , cat.CategoryName  
FROM products pro JOIN categories cat  
ON pro.categoryID = cat.categoryID  
WHERE pro.UnitPrice > 50  
-- 7 rows
```

.6

```
SELECT pro.ProductID , pro.UnitPrice , pro.supplierID, cat.CategoryName  
FROM products pro JOIN categories cat  
ON pro.categoryID = cat.categoryID  
WHERE pro.supplierID = 3  
-- 3 rows
```

.7
SELECT pro.ProductID , pro.UnitPrice , pro.supplierID, cat.CategoryName
FROM products pro JOIN categories cat
ON pro.categoryID = cat.categoryID
WHERE cat.categoryName LIKE '%a%'
-- 47 rows

.8
SELECT pro.productName , cat.categoryName , sup.companyName
FROM products pro
JOIN categories cat
ON pro.categoryID = cat.categoryID
JOIN Suppliers sup
ON sup.supplierID = pro.supplierID
-- 77 rows

.9
SELECT pro.productName , cat.Description , sup.city
FROM products pro
JOIN categories cat
ON pro.categoryID = cat.categoryID
JOIN Suppliers sup
ON sup.supplierID = pro.supplierID
WHERE
sup.city IN ('london' , 'tokyo')
-- 6 rows

.10
SELECT pro.productID , cat.Description , sup.country
FROM products pro
JOIN categories cat
ON pro.categoryID = cat.categoryID
JOIN Suppliers sup
ON sup.supplierID = pro.supplierID
WHERE
sup.country LIKE 'a%'
-- 8 rows

.11
SELECT cust.companyName , ord.orderID
FROM customers cust LEFT OUTER JOIN orders ord
ON cust.customerID = ord.customerID
--832 rows

.12

```
SELECT      o.OrderID,
            o.OrderDate,
            o.ShipAddress,
            c.customerID,
            c.CompanyName,
            c.Phone
FROM Customers c JOIN ORDERS o
ON c.CustomerID = o.CustomerID
WHERE YEAR(OrderDate) = 1996
AND (c.CustomerID LIKE 'a%' OR c.CustomerID LIKE 'c%')
-- 8 rows
```

.13

```
SELECT      o.OrderID AS 'Order#',
            o.OrderDate AS 'Date of Order',
            o.ShipAddress AS 'Where to send it',
            c.customerID AS 'customer Number',
            c.CompanyName AS 'who ordered',
            c.Phone AS 'phone number',
            e.LastName AS 'Emp last name',
            e.FirstName AS 'Emp first name'
FROM Customers c
      JOIN ORDERS o
      ON c.CustomerID = o.CustomerID
      JOIN employees e
      ON e.employeeID = o.EmployeeID
WHERE YEAR(OrderDate) = 1996
AND (c.CustomerID LIKE 'a%' OR c.CustomerID LIKE 'c%')
ORDER BY o.OrderDate DESC
-- 8 rows
```

.14

```
SELECT emp.lastName AS 'employee' , mng.lastName AS 'manager'
FROM employees emp JOIN employees mng
ON emp.reportsto = mng.employeeID
-- 8 rows
```

```
SELECT emp.lastName AS 'employee' , mng.lastName AS 'manager'
FROM employees emp LEFT OUTER JOIN employees mng
ON emp.reportsto = mng.employeeID
-- 9 rows
```

.15

```
SELECT p.ProductID, p.ProductName, p.unitPrice
FROM products p JOIN products am
ON p.UnitPrice > am.UnitPrice
   AND am.ProductName = 'Alice Mutton'
-- 13 rows
```

GROUP FUNCTIONS

- .1
SELECT min(lastName)
FROM employees
-- 1 row
- .2
SELECT max(firstName)
FROM employees
-- 1 row
- .3
SELECT count(*)
FROM employees
-- 1 row
- .4
SELECT count(region)
FROM employees ;
-- 1 row
- .5
SELECT avg(UnitPrice)
FROM Products ;
-- 1 row
- .6
SELECT MAX(UnitPrice) AS 'max price' , AVG(UnitPrice) AS 'average price'
FROM Products
-- 1 row
- .7
SELECT CONVERT(VARCHAR, MIN(BirthDate), 113) AS 'Min Birth-date' ,
CONVERT(VARCHAR, MAX(BirthDate), 113) AS 'Max Birth-date'
FROM Employees
-- 1 row
- .8
SELECT COUNT(CustomerID) AS 'Number of customers'
FROM Customers
-- 1 row
- .9
SELECT COUNT(DISTINCT CustomerID) AS 'Number of customers'
FROM Orders

-- 1 row

.10

```
SELECT      MAX(UnitPrice) AS 'max price',
            MIN (UnitPrice) AS 'min Price',
            AVG(UnitPrice) AS 'Avg Price',
            CategoryID
FROM products
GROUP BY CategoryID
-- 8 rows
```

.11

```
SELECT max(unitPrice) AS 'highest price' , SupplierID
FROM Products
GROUP BY SupplierID
ORDER BY SupplierID DESC
-- 29 rows
```

.12

```
SELECT avg(UnitsInStock) , SupplierID
FROM Products
GROUP BY SupplierID
ORDER BY avg(UnitsInStock) DESC
-- 29 rows
```

.13

```
SELECT count(CompanyName) , Country , City
FROM Customers
group by Country , City
-- 69 rows
```

.14

```
SELECT AVG(unitPrice) , CategoryID
FROM Products
WHERE unitPrice > 40
GROUP BY CategoryID
-- 7 rows
```

.15

```
SELECT count(CompanyName) , City
FROM Customers
WHERE City in ( 'London','Berlin','Paris','Rio de Janeiro')
GROUP BY City
-- 4 rows
```

.16

```
SELECT      MAX(UnitPrice) as 'max price', MIN (UnitPrice) AS 'min Price',
            AVG(UnitPrice) as 'Avg Price',
            COUNT(*) AS 'Num Of Products',
            CategoryID,
            SupplierID
FROM products
GROUP BY CategoryID, SupplierID
```

.17

```
SELECT MAX(unitPrice) , CategoryID
FROM products
GROUP BY CategoryID
HAVING MAX(unitPrice) > 40
-- 7 rows
```

.18

```
SELECT AVG(unitPrice) , SupplierID
FROM products
GROUP BY SupplierID
HAVING AVG(unitPrice) > 40
-- 4 rows
```

.19

```
SELECT      sum(p. UnitsOnOrder) AS ' UnitsOnOrder ',
            sum(p.UnitsInStock) AS ' UnitsInStock',
            c.CategoryName
FROM products p JOIN categories c
      ON c.CategoryID = p.CategoryID
WHERE c. categoryname LIKE '%c%'
GROUP BY c.CategoryName
HAVING sum(p. UnitsOnOrder) > 100
ORDER BY c.categoryName
-- 3 rows
```

.20

```
SELECT Region, city, count(*) AS 'how many customers'
FROM customers
WHERE (city LIKE '%m%' OR city LIKE '%l%')
AND Region is not null
GROUP BY Region, city
HAVING count(*) >= 2
-- 2 rows
```

.21

```
SELECT      e.LastName AS 'employee',
            COUNT(o.OrderID) AS 'total orders',
            MAX(o.OrderDate) AS 'last Order'
FROM Orders o JOIN Employees e
      ON e.EmployeeID = o.EmployeeID
GROUP BY e.LastName
HAVING COUNT(o.OrderID) > 100
-- 4 rows
```

SUBQUERIES

.1

```
SELECT ProductName, UnitPrice
FROM Products
WHERE UnitPrice < (SELECT UnitPrice
                   FROM Products
                   WHERE ProductID =8)

-- 64 rows
```

.2

```
SELECT ProductName, UnitPrice
FROM Products
WHERE UnitPrice > (SELECT UnitPrice
                   FROM Products
                   WHERE ProductName = 'Tofu')

-- 30 rows
```

.3

```
SELECT LastName, HireDate
FROM employees
WHERE HireDate > (SELECT HireDate
                  FROM employees
                  WHERE employeeid = 6)

-- 3 rows
```

```
.4
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE UnitPrice > (SELECT AVG(UnitPrice)
                  FROM Products)

-- 25 rows
```

.5

```
SELECT productName , unitsInStock
FROM products
WHERE unitsInStock < (SELECT min(unitsInStock)
                     FROM Products
                     WHERE categoryID = 5 )

-- 29 rows
```

.6

```
SELECT *
FROM Products
WHERE CategoryID = (SELECT CategoryID
                    FROM Products
                    WHERE ProductName = 'Chai')
AND ProductName <> 'Chai'
-- 11 rows
```

.7

```
SELECT productName , UnitPrice , CategoryID
FROM products
WHERE unitPrice IN (SELECT unitPrice
                   FROM products
                   WHERE categoryID = 5)
-- 12 rows
```

.8

```
SELECT productName , UnitPrice
FROM products
WHERE unitPrice > ANY (SELECT unitPrice
                     FROM products
                     WHERE categoryID = 5)
-- 73 rows
```

.9

```
SELECT productName , UnitPrice
FROM products
WHERE unitPrice > ALL (SELECT unitPrice
                     FROM products
                     WHERE categoryID = 5)
-- 14 rows
```

.10

```
SELECT OrderID, OrderDate
FROM Orders
WHERE CustomerID IN (SELECT CustomerID
                   FROM customers
                   WHERE country IN('germany','France','sweden'))
AND YEAR(OrderDate) = 1997
-- 120 rows
```


.11

```
SELECT ProductName,ProductID
FROM products
WHERE UnitPrice > (SELECT AVG(UnitPrice)
                   FROM products
                   WHERE UnitsInStock>50)

-- 42 rows
```

.12

```
SELECT ProductName
FROM products
WHERE CategoryID IN (SELECT CategoryID
                    FROM categories
                    WHERE CategoryName IN('Beverages','Condiments'))
AND SupplierID IN (SELECT SupplierID
                  FROM suppliers
                  WHERE region IS NULL)

-- 12 rows
```

.13

```
SELECT CompanyName
FROM Suppliers
WHERE SupplierID IN (SELECT SupplierID
                    FROM Products
                    WHERE CategoryID =
                        (SELECT CategoryID
                         FROM Categories
                         WHERE CategoryName = 'beverages'))

-- 8 rows
```

Set Operators

.1

SELECT city FROM employees ;

.2

SELECT city FROM customers ;

.3

SELECT city FROM employees
UNION
SELECT city FROM customers ;

.4

SELECT city FROM employees
UNION ALL
SELECT city FROM customers ;

.5

SELECT city FROM employees
INTERSECT
SELECT city FROM customers ;

.6

SELECT city FROM employees
EXCEPT
SELECT city FROM customers ;

Analytic Functions

.1

```
SELECT productName , UnitPrice , categoryID ,  
       SUM(UnitPrice) OVER () AS 'Avg_Price'  
FROM products
```

.2

```
SELECT productName , UnitPrice , categoryID ,  
       SUM(UnitPrice) OVER (PARTITION BY categoryID) AS 'Avg_Price_Per_Category'  
FROM products
```

.3

```
SELECT productName , UnitPrice , categoryID ,  
       SUM(UnitPrice) OVER (PARTITION BY categoryID ORDER BY unitPrice DESC) AS  
'Avg_Price_Per_Category'  
FROM products
```

.4

```
SELECT productName , UnitPrice , categoryID ,  
       SUM(UnitPrice) OVER (PARTITION BY categoryID  
                           ORDER BY unitPrice DESC  
                           ROWS BETWEEN UNBOUNDED PRECEDING AND  
UNBOUNDED FOLLOWING ) AS 'Avg_Price_Per_Category'  
FROM products
```

.5

```
SELECT productName , UnitPrice , categoryID ,  
       SUM(UnitPrice) OVER (PARTITION BY categoryID  
                           ORDER BY unitPrice DESC  
                           ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT  
ROW ) AS 'Avg_Price_Per_Category'  
FROM products
```

.6

```
SELECT productName , UnitPrice , categoryID ,
       RANK() OVER ( ORDER BY unitPrice) AS 'Rank' ,
       DENSE_RANK() OVER ( ORDER BY unitPrice) AS 'Dense Rank',
       ROW_NUMBER() OVER ( ORDER BY unitPrice) AS 'Row Number' ,
       NTILE (10) OVER ( ORDER BY unitPrice) AS '10 Groups'
FROM products
```

.7

```
SELECT productName , UnitPrice , categoryID ,
       RANK() OVER ( PARTITION BY categoryID ORDER BY unitPrice) AS 'Rank' ,
       DENSE_RANK() OVER ( PARTITION BY categoryID ORDER BY unitPrice) AS 'Dense Rank',
       ROW_NUMBER() OVER ( PARTITION BY categoryID ORDER BY unitPrice) AS 'Row
Number' ,
       NTILE (10) OVER ( PARTITION BY categoryID ORDER BY unitPrice) AS '10 Groups'
FROM products
```

.8

```
SELECT productID ,
       ProductName,
       unitPrice ,
       LAG(unitPrice, 1) OVER (ORDER BY unitPrice) as lag,
       LEAD(unitPrice,1) OVER (ORDER BY unitPrice) as lead
FROM products
WHERE categoryID = 1
ORDER BY unitPrice;
```

DML

1. @@IDENTITY מחזיר את הערך האחרון אשר הוכנס בפקודת INSERT לעמודה אשר מוגדר בה מספור אוטומטי.

2.

```
CREATE TABLE my_employees
(id INT PRIMARY KEY ,
name VARCHAR (50),
title VARCHAR(50),
deptid INT,
salary MONEY DEFAULT 3500)
```

3.

Sp_help my_employees

4. נתונים לדוגמא.

ID	NAME	TITLE	DEPTID	SALARY
1	Aviv Cohen	Clerk	10	4000
2	Miriam Levi	Sales Manager	20	3750
3	AION Romano	OperatiON Manager	30	NULL
4	Baruch Nave	NULL	30	3500
5	Danny SalomON	Sales Representative	20	7000

BEGIN TRAN

5.

```
INSERT INTO my_employees
Values (1, 'Aviv Cohen', 'Clerk', 30, 4000)
```

6.

```
INSERT INTO my_employees (id, name, title, deptid, salary)
Values (2, 'Miriam Levi', 'Sales Manager', 20, 3750)
```

7.

```
INSERT INTO my_employees
Values (3, 'AION Romano', 'OperatiON Manager', 30, null)
```

8.

```
INSERT INTO my_employees (id, name, deptid)
Values (4, 'Baruch Nave', 30)
```

SELECT * FROM my_employees

הרשומה תואמת את הרשומה בנתונים לדוגמא, שכן אם לא מציינים את עמודת salary במשפט ה-INSERT היא מקבלת את ה DEFAULT שלה, ואם לא מציינים את עמודת title היא מקבלת ה NULL.

.9

INSERT INTO my_employees
 Values (5, 'Danny SalomON', 'Sales Representative', 20, 7000)

.10

SELECT *
 FROM my_employees

.11

UPDATE my_employees
 SET salary = 4500
 WHERE id = 2

.12

UPDATE my_employees
 SET name = 'your name',
 Deptid = 20
 WHERE id = 4

.13

SELECT *
 FROM my_employees

.14

UPDATE my_employees
 SET deptid = 10
 WHERE dept id = 30

.15

DELETE FROM my_employees
 WHERE name = 'Baruch Nava'

.16

SELECT *
 FROM my_employees

.17

INSERT INTO my_employees (id, name, title)
 SELECT employeeid, lastname, title
 FROM employees
 WHERE employeeid >5

.18

SELECT * FROM my_employees

.19

COMMIT

יצירת DATA BASE ופקודות DDL ליצירת טבלה

1.

שימוש בכלי ה- MICROSOFT MANAGEMENT STUDIO :

לחצן ימני על DATABASES ... ← NEW DATA BASE

לציין את שם ה DATA BASE – human_resource_db

2.

```
CREATE TABLE departments
(department_id INT IDENTITY(10,10)
CONSTRAINT dept_dept_id_pk PRIMARY KEY,
department_name VARCHAR(20)
CONSTRAINT dept_dept_name_nn NOT NULL,
location VARCHAR(30) CONSTRAINT dept_loc_nn NOT NULL)
```

```
CREATE TABLE employees
(employee_id INT IDENTITY(1,1)
CONSTRAINT emp_emp_id_pk PRIMARY KEY,
last_name VARCHAR(20) CONSTRAINT emp_lname_nn NOT NULL,
hire_date DATETIME DEFAULT GETDATE()
CONSTRAINT emp_hdate_nn NOT NULL,
salary MONEY DEFAULT 5000,
email VARCHAR(30) CONSTRAINT emp_email_nn NOT NULL,
CONSTRAINT emp_email_uk UNIQUE (email))
```

3.

```
ALTER TABLE employees
ADD department_id INT CONSTRAINT emp_dept_id_dept_id_fk REFERENCES
departments(department_id)
```

4.

```
SELECT * INTO emp_history
FROM employees
WHERE 1=2
```

5.

```
DROP TABLE emp_history
```

VIEWS

.1

```
CREATE VIEW emp_vw
AS
SELECT EmployeeID, LastName, FirstName, address
FROM employees
WHERE city IN ('london', 'tacoma')
```

.2

```
CREATE VIEW pro_vw
AS
SELECT CategoryID, AVG(UnitPrice) as "avg_price"
FROM products
WHERE CategoryID IN (1,2,4)
GROUP BY CategoryID
```

.3

```
ALTER VIEW pro_vw
AS
SELECT c.CategoryName, AVG(p.UnitPrice)
FROM products p JOIN categories c
ON c. CategoryID=p. CategoryID
WHERE c.CategoryID IN (1,2,4)
GROUP BY c.CategoryName
HAVING AVG(p.UnitPrice) > 25
```

.4

```
CREATE VIEW cust_orders_vw
AS
SELECT c.CompanyName, COUNT(o.OrderID) NUMBER_OF_ORDERS
FROM customers c JOIN orders o
ON c.CustomerID = o. CustomerID
GROUP BY c. CompanyName
```


5.

```
SELECT *
FROM cust_orders_vw
WHERE NUMBER_OF_ORDERS > 10
```

6.

```
CREATE VIEW Prods
AS
SELECT      ProductID,
            ProductName,
            UnitPrice,
            UnitsInStock-UnitsOnOrder as "UnitsLeft"
FROM Products
WHERE UnitPrice > 30
```

7.

```
BEGIN TRAN
```

א.

```
UPDATE prods
SET ProductName = 'Jiffa'
WHERE ProductID=52
```

ב.

```
UPDATE prods
SET UnitPrice = 31
WHERE ProductID = 52
```

ג.

```
SELECT * FROM Prods WHERE ProductID = 52
```

לא נשלפות רשומות, מוצר 52 כבר לא מופיע כחלק מה-view, כי מחירו כעת הוא מעל 30.

ד.

```
UPDATE prods
SET UnitsLeft = 41
WHERE ProductID = 1
```

התקבלה שגיאה. העמודה UnitsLeft לא באמת קיימת בטבלה, היא עמודה וירטואלית של ה-view.

```
ROLLBACK
```

8.

```
ALTER VIEW Prods
AS
SELECT      ProductID,
            ProductName,
            UnitPrice,
            UnitsInStock-UnitsOnOrder as "UnitsLeft"
FROM Products
WHERE UnitPrice > 30
WITH CHECK OPTION
```

9.

```
BEGIN TRAN
```

א.

```
UPDATE prods
SET ProductName = 'Jiffa'
WHERE ProductID=52
```

העדכון התבצע בהצלחה.

ב.

```
UPDATE prods
SET UnitPrice = 31
WHERE ProductID = 52
```

לא ניתן לבצע את העידכון משום שבview עם check option לא ניתן לבצע עדכונים המפרים את תנאי ה where. במקרה שלנו לא ניתן לעדכן מחירים למעל 30.

```
ROLLBACK
```

BASIC T- SQL

.1

```
DECLARE    @var1 VARCHAR(25),  
           @var2 VARCHAR(25)
```

```
SET @var1 = 'MY FIRST'  
SET @var2 = 'VARIABLE'
```

```
PRINT @var1 + ' ' + @var2
```

.2

```
DECLARE    @var1 VARCHAR(25),  
           @var2 DATETIME
```

```
SET @var1 = 'THE DATE TODAY IS:'  
SET @var2 = GETDATE()
```

```
PRINT @var1 + ' ' + CAST(@var2 AS VARCHAR(30))
```

.3

```
DECLARE @name NVARCHAR(20)
```

```
SELECT @name = LastName  
FROM employees  
WHERE EmployeeID = 4
```

```
PRINT @name
```

.4

```
DECLARE    @prod_name NVARCHAR(40),
            @cat_name NVARCHAR(15)

SELECT @prod_name = p.ProductName, @cat_name = c.CategoryName
FROM products p JOIN categories c
      ON p.CategoryID = c. CategoryID
WHERE p.productid = 5

PRINT @prod_name+' ' +@cat_name
```

.5

```
DECLARE @prod_price MONEY

SELECT @prod_price = UnitPrice
FROM products
WHERE ProductID = 9

IF @prod_price > 50
    PRINT 'I LIKE THIS PRODUCT'
ELSE
    SELECT 'I DO NOT LIKE THIS PRODUCT'
```

```
DECLARE    @prod_name NVARCHAR(40),  
           @prod_price MONEY
```

```
SET @prod_name = 'CHAI'      -- CHANGE THE VALUE OF PROD_NAME  
                             ACCORDING TO THE QUESTION
```

```
SELECT @prod_price = UnitPrice  
FROM products  
WHERE ProductName = @prod_name
```

```
IF @prod_price < 20  
    BEGIN  
        SET @prod_price = @prod_price*1.1  
        PRINT  
            'THE NEW PRICE OF PRODUCT: ' + @prod_name + ' IS: ' +  
            CAST(@prod_price AS VARCHAR(8))  
    END  
ELSE  
    IF @prod_price >= 20 AND @prod_price < 40  
        BEGIN  
            SET @prod_price = @prod_price*1.2  
            PRINT 'THE NEW PRICE OF PRODUCT: ' +  
                @prod_name + ' IS: ' + CAST(@prod_price AS  
                VARCHAR(8))  
        END  
    ELSE  
        IF @prod_price >= 40  
            BEGIN  
                SET @prod_price = @prod_price*1.5  
                PRINT 'THE NEW PRICE OF PRODUCT: ' +  
                    @prod_name + ' IS: ' + CAST(@prod_price AS  
                    VARCHAR(8))  
            END  
        END
```

.7

```
DECLARE
    @EmpID INT,
    @EmpName VARCHAR(20)
SET @EmpID = 1
WHILE @EmpID <= 9
    BEGIN
        SELECT @EmpName = LastName
        FROM employees
        WHERE EmployeeID = @EmpID

        IF @EmpName LIKE '%e%'
            PRINT @EmpName

        SET @EmpID = @EmpID + 1
    END
```

.8

```
DECLARE @cat_num INT,
        @cat_name NVARCHAR(15)

SET @cat_num = 4

LABEL:
    SELECT *
    FROM categories
    WHERE Categoryid = @cat_num

    SELECT @cat_name = categoryname
    FROM categories
    WHERE categoryid = @cat_num

    SET @cat_num = @cat_num + 1

    IF @cat_name LIKE '%a%'
        GOTO LABEL
    ELSE
        SELECT 'NO MORE CATEGORIES
        WITH THE LETTER A!'
```

ERROR HANDLING

.1

```
BEGIN TRY

DELETE FROM employees
WHERE EmployeeID = 2

END TRY
BEGIN CATCH

PRINT 'can''t delete this employee, this employee is a manager!'

END CATCH
```

.2

```
BEGIN TRY

UPDATE products
SET CategoryID = 11
WHERE ProductID = 5

END TRY
BEGIN CATCH

SELECT error_number() as "Err#", error_message() as "Message"

END CATCH
```

.3

```
DECLARE @rows INT

UPDATE employees
SET city = 'ASHKELON'
WHERE employeeid = 17
--
SET @rows = @@ROWCOUNT
IF @@ROWCOUNT = 0
    RAISERROR ('NO SUCH EMPLOYEE!', 16, 1)
```

.4

```
EXEC sp_addmessage  
    @msgnum = 50004,  
    @severity = 16,  
    @msgtext = 'AN ERROR ACCORD, CAN NOT DELETE RECORD'
```

.5

```
BEGIN TRY  
  
DELETE FROM customers  
WHERE customerid = 'ALFKI'  
  
END TRY  
BEGIN CATCH  
  
    RAISERROR (50004, 16,1)  
  
END CATCH
```


STORED PROCEDURES

.1

```
CREATE PROCEDURE prod_det  
AS  
SELECT p.productname, c.categoryname  
FROM products p JOIN categories c  
ON p.categoryid = c.categoryid  
WHERE p.productid = 6
```

-- הרצה של הפרוצדורה

```
EXEC prod_det
```

.2

```
ALTER PROCEDURE prod_det  
(@prod_num INT)  
AS  
SELECT p.ProductName, c.CategoryName  
FROM products p JOIN categories c  
ON p.CategoryID = c. CategoryID  
WHERE p.ProductID = @prod_num
```

-- הרצה של הפרוצדורה

```
EXEC prod_det_in_parameter 17
```

.3

```
CREATE PROCEDURE top_price_prod  
AS  
SELECT TOP 10 ProductID, ProductName, UnitPrice  
FROM products  
ORDER BY UnitPrice DESC
```

-- הרצה של הפרוצדורה

```
EXEC top_price_prod
```

.4

```
ALTER PROCEDURE top_n_prod
(@top_n INT)
AS
SELECT TOP (@top_n) productid, productname, unitprice
FROM products
ORDER BY unitprice DESC
```

-- הרצה של הפרוצדורה

```
EXEC top_n_prod 5
```

.5

```
CREATE PROCEDURE emp_det
(@emp_num INT)
AS
```

```
DECLARE @emp_name NVARCHAR(20)
```

```
SELECT @emp_name = lastname
FROM employees
WHERE employeeid = @emp_num
```

```
IF @emp_name LIKE '%a%'
    SELECT LastName, BirthDate, City
    FROM employees
    WHERE EmployeeID = @emp_num
ELSE
    PRINT ' Employee Number: ' + CAST(@emp_num AS VARCHAR(4))+ '
    Has no letter "A" - '+@emp_name
```

-- הרצה של הפרוצדורה

```
EXEC emp_det 7
```

```
CREATE PROCEDURE cust_det
(@cust_id NCHAR(5))
AS
SELECT CompanyName, Country, Address
FROM customers
WHERE CustomerID = @cust_id
```

-- הרצה של הפרוצדורה

```
EXEC cust_det 'ALFKI'
```

```
ALTER PROCEDURE cust_det
(@cust_id NCHAR(5))
AS
DECLARE @country NVARCHAR(15)
```

```
SELECT @country = country
FROM customers
WHERE CustomerID = @cust_id
```

```
IF @country = ' MEXICO '
    PRINT 'The Customer is from MEXICO'
ELSE
    IF @country = 'GERMANY'
        PRINT ' The Customer is from GERMANY'
    ELSE
        SELECT CompanyName, Country, Address
        FROM customers
        WHERE customerid = @cust_id
```

--הרצה של הפרוצדורה--

```
EXEC cust_det 'ALFKI'
```

```
EXEC cust_det 'ANTON'
```

```
EXEC cust_det 'BERGS'
```

```
CREATE PROCEDURE cust_ord_num
(@cust_id  NCHAR(5),
 @ord_num  INT OUTPUT)
AS
SELECT @ord_num = COUNT(OrderID)
FROM orders
WHERE CustomerID = @cust_id

IF @ord_num > 10
    PRINT 'NUMBER OF ORDERS IS GREATER THAN 10'
ELSE
    SELECT *
    FROM orders
    WHERE CustomerID = @cust_id
```

הרצה של הפרוצדורה--

```
DECLARE @orders_amount INT

EXEC cust_ord_num 'ALFKI' ,@orders_amount OUTPUT

PRINT @orders_amount
```