

בסיסי נתונים ושפת SQL

חוברת תרגול

קורס הסבה לבודקים

תוכן עניינים

SELECT	3
WHERE	5
SCALAR FUNCTIONS	7
JOIN	9
GROUP FUNCTIONS	11
SUBQUERIES	13
DML	14
פתרונות	16
SELECT	17
WHERE	20
SCALAR FUNCTIONS	24
JOIN	26
GROUP FUNCTIONS	29
SUBQUERIES	33
DML	36

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 (אליאס - ProNm) , UnitPrice (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 מחירים 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 מחירים אינו בין 50 ל- 100.
7. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים אשר מחירים נע בין 21.35 לבין 43.9, מיין 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. מיין 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 בשמן היא 'א'.
18. הצג מטבלת Products את שם המוצר ומספר הקטגוריה שלו, עבור המוצרים שבשמן האות 'א' 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 ההזמנה 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. 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.
3. הצג מתוך טבלת Products את מספר המוצר, שם המוצר, ובעמודה נוספת שוב את שם המוצר כאשר כל תו '?' יוחלף בתו '!'.
4. הצג את התאריך של היום.
5. 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).
6. הצג מתוך טבלת Employees את שמו הפרטי של העובד ואת גילו (ההפרש בשנים בין התאריך העכשווי ובין תאריך לידתו).
7. הצג מתוך טבלת Employees את שמו הפרטי של העובד, את היום בו התחיל לעבוד (לדוגמא, sunday, monday ...), ואת השנה בה הוא התחיל לעבוד.
8. 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.
9. בחר מטבלת Employees את מספר העובד ושם המשפחה משורשרים יחדיו כשביניהן רווח (תן כינוי מתאים לעמודה זו), ואת תאריך הלידה בעמודה נפרדת.
10. 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.

11. 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.

12. הצג מטבלת employees את העמודות הבאות:

שם המשפחה משורשר עם אורך שם המשפחה,

שם פרטי משורשר עם אורך השם הפרטי.

יש לתת לכל עמודה בתצוגה כינוי מתאים.

13. 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**.

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

שאלת אתגר

15. 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 מס' ההזמנה ומספר הלקוח מטבלת Orders ואת מספר המוצר , כמות, מחיר המוצר , הנחה וסה"כ שמורכב מכמות* מחיר כולל הנחה, תן שם מתאים לעמודה, עבור מספרי הזמנות שמספרן 10250-10260.
9. הצג את מספר ההזמנה וכמות מוזמנת מטבלת Order Details ואת שם המוצר מטבלת Products עבור המוצרים שכמותם המוזמנת גדולה מ-50.
10. הצג את מספר ההזמנה מטבלת Orders ואת מספר המשלוח ושם החברה המשלחת מטבלת Shippers עבור החברות ששמן מתחיל באותיות S או U.
11. הצג את מס' ההזמנה ושם העובד שביצע את ההזמנה מטבלת Orders ואת עיר העובד מטבלת Employees עבור העובדים שגרים ב London – Redmond . סדר את התוצאות לפי מס' הזמנה.
12. הצג את מס' ההזמנה, שנת הזמנה, עיר המשלוח ואזור המשלוח מטבלת Orders ואת שם חברת המשלוח מטבלת Shippers עבור המשלחים שהאזור שלהם אינו NULL ושנת המשלוח היא 1997.
13. הצג את מס' ההזמנה ומס' הלקוח מטבלת Orders ואת שמו המלא של העובד שביצע את ההזמנה, עבור עובד מס' 4.
14. הצג את מס' ההזמנה ועיר המשלוח מטבלת Orders ואת עיר הלקוח מטבלת לקוחות עבור ההזמנות שעיר המשלוח ועיר הלקוח זהות.

15. הצג את מספר ההזמנה, מספר הלקוח ושם העובד מטבלת הזמנות ואת עמודות TitleOfCourtesy, שם פרטי ושם משפחה משורשרות ביחד, תן שם מתאים לעמודה.
עבור ההזמנות שבוצעו ע"י נשים הגרות בארה"ב.

16. הצג את שם החברה של הלקוח מתוך טבלת customers ואת מספר ההזמנה מתוך טבלת orders עבור הלקוחות, גם אלו ללא הזמנות.

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 בלונדון.
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 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).
9. 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.
10. הצג מטבלת products את שמות כל המוצרים אשר **שם** הקטגוריה שלהם הוא Beverages או Condiments **וגם** אזור (region) הספק אינו ידוע.
11. הצג את שמות החברות (CompanyName מטבלת Suppliers) המספקות מוצרים מקטגוריית Beverages (עמודת CategoryName בטבלת Categories).

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 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 לפטר. 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.

פתרונות

העתק זה הורד על ידי אנטון מייאסניקוב - anton-r@live.com

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 "Prold" , 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
```

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

.8

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

.9

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

.10

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

.11

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

.12

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

.13

.14
 SELECT OrderID, OrderDate, RequiredDate
 FROM orders
 WHERE RequiredDate > '1996-10-01'
 -- 781 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 '1997-05-30'
 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 LIKE '%1963%'
-- 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 mond entire',' Eastern Connection')
AND orderDate + 20 < requireddate
--1 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
-- 20 rows

SCALAR FUNCTIONS

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

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

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

SELECT GetDate()

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

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

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

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

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

10.
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

11.
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

12.
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

13.
SELECT OrderID, OrderDate, RequiredDate
FROM Orders
WHERE DATEDIFF(q, OrderDate , RequiredDate) =1
-- 249 rows

14.
SELECT SUBSTRING(companyName , 1 , 4)
FROM customers
WHERE companyName LIKE 'a%'
-- 4 rows

15.
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
-- 7 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 o.OrderID,o.CustomerID,od.ProductID,od.OrderID,od.Quantity,
od.Discount,od.Quantity*od.UnitPrice*(1-od.Discount) as "fix price"
FROM Orders o join [Order Details] od
on o.OrderID=od.OrderID
WHERE o.OrderID between 10250 and 10260
--33 rows
```

.9

```
SELECT OD.OrderID,P.ProductName,OD.Quantity
FROM Products P JOIN [Order Details] OD
ON P.ProductID=OD.ProductID
WHERE OD.Quantity>50
--159 rows
```

.10

```
SELECT O.OrderID,S.ShipperID,S.CompanyName
FROM Orders O JOIN Shippers S
ON O.ShipVia=S.ShipperID
WHERE S.CompanyName LIKE '[S,U]%'
```

.11

```
SELECT O.OrderID,O.EmployeeID,E.City
FROM Orders O JOIN Employees E
ON O.EmployeeID=E.EmployeeID
WHERE E.City IN('LONDON','REDMOND')
ORDER BY O.OrderID
```

12.
SELECT O.OrderID, YEAR(OrderDate), S.CompanyName, O.ShipCity, O.ShipRegion
FROM Orders O JOIN Shippers S
ON O.ShipVia=S.ShipperID
WHERE O.ShipRegion IS NOT NULL
AND YEAR(O.OrderDate)=1997

13.
SELECT o.OrderID, o.CustomerID, e.FirstName+ ' '+e.LastName as "full name"
FROM Employees e join Orders o
on e.EmployeeID=o.EmployeeID
WHERE o.EmployeeID=4
--156 rows

14.
SELECT O.OrderID, O.ShipCity, C.City
FROM Orders O JOIN Customers C
ON O.CustomerID=C.CustomerID
WHERE O.ShipCity=C.City

15.
SELECT O.OrderID, O.CustomerID, O.EmployeeID,
E.TitleOfCourtesy+ ' '+E.FirstName+ ' '+E.LastName AS "EMPLOYEE'S FULL NAME", E.Country
FROM Employees E JOIN Orders O
ON E.EmployeeID=O.EmployeeID
WHERE E.Country='USA'
AND E.TitleOfCourtesy IN ('MS.', 'MRS.')

16.
SELECT cust.companyName , ord.orderID
FROM customers cust LEFT OUTER JOIN orders ord
ON cust.customerID = ord.customerID
--832 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

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

12.
SELECT avg(UnitsInStock) , SupplierID
FROM Products
GROUP BY SupplierID
ORDER BY avg(UnitsInStock) DESC

13.
SELECT count(CompanyName) , Country , City
FROM Customers
group by Country , City

14.
SELECT AVG(unitPrice) , CategoryID
FROM Products
WHERE unitPrice > 40
GROUP BY CategoryID

15.
SELECT count(CompanyName) , City
FROM Customers
WHERE City = 'London'
GROUP BY City

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
-- 49 rows
```

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

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

```
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
```

```
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
```

```
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
```

.21

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 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 , UnitPrice
FROM products
WHERE unitPrice < (SELECT min(UnitPrice)
                   FROM Products
                   WHERE categoryID = 5 )

-- 3 rows
```

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

.6

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

.7

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

.8

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

.9

```
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
```

.10

11.
SELECT CompanyName
FROM Suppliers
WHERE SupplierID IN (SELECT SupplierID
FROM Products
WHERE CategoryID =
(SELECT CategoryID
FROM Categories
WHERE CategoryName = 'beverages'))

-- 8 rows

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)

בדיקה:

SELECT * FROM my_employees

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

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

.17

```
SELECT * FROM my_employees
```

.18

```
COMMIT
```

.19