SQL

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

1.ביטויים

פקודות SELECT ו –FROM

פקודת ה-SELECT הינה הפקודה הנפוצה ביותר ב-SQL

פקודה זו אינה פועלת לבד, אלא ביחד עם פקודת FROM (מטבלה)

SELECT <שם עמודה/ות> FROM<שם טבלה>

דוגמא: שליפת כל הרשומות מטבלת Employees:

```
SELECT * FROM Employees
```

כאשר הסימן * משמעו "הכל"

2.התניות

WHERE

התניה – קשר של אם-אז – משתמשים כאשר אנו רוצים למצוא פריט מסוים, או קבוצת פריטים, במסד הנתונים

התניה מכילה בדרך כלל:

משתנה, קבוע, אופרטור = דוגמא: LastName = 'Fuller'

דוגמא: על מנת להציג נתוני עובד מסוים:

```
SELECT *
FROM Employees
WHERE LastName = 'Fuller'
```

3.אופרטורים

גורמים הנרשמים בתוך ביטוי, על מנת להודיע ליישום כיצד לאחזר את הנתונים

קיימות 6 קבוצות של אופרטורים:

- חשבוניים
- השוואתיים
- תוויים
- לוגיים
- קבוצתיים
- שונים

3.1. אופרטורים חשבוניים

אופרטור +

שימוש בפלוס להוספת סכום מסוים לעמודה מסוימת

דוגמא: (הוספת 15 אגורות לכל מחיר)

```
SELECT productname, unitprice, unitprice+0.15 FROM products
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

הערה:
Alias: ניתן לשנות את שם העמודה החדשה לשם אחר כאשר משתמשים ב-AS. בדוגמא למעלה זה יופיע כך:

SELECT productname, unitprice, unitprice+0.15 AS NewPrice FROM Products

שים לב: אין חובה לרשום את המילה AS

אופרטור מינוס –

לאופרטור מינוס יש שני שימושים: שינוי הסימן של מספר והפחתה בין משתנים (הפחתת ערך של עמודה אחת מערך של עמודה אחרת)

שימוש 1 דוגמא:

SELECT productname, unitprice, -unitprice AS MinusPrice FROM products

וכל המספרים שהופיעו בעמודת UNITPRICE יקבלו סימון של מינוס

שימוש 2 דוגמא:

ברצוננו לראות כמה יחידות מכל מוצר ישארו במלאי לאחר כל הזמנה

SELECT productname, unitsonorder, unitsinstock, unitsinstock – unitsinorder AS RemainingInStock FROM products

ונקבל עמודה בה נראה את השארית של היחידות של כל מוצר לאחר ההזמנה

אופרטור חילוק /

חלוקת נתונים

לדוגמא: אם אנחנו רוצים לצאת במבצע של שניים במחיר:

SELECT productname, unitprice, unitprice/2 AS MechirMivtza
FROM products

ובעמודה החדשה יופיע לנו חצי מהמחיר המקורי של המוצר

אופרטור כפל *

כפל של נתונים

לדוגמא: הנחה של 10 אחוזים במחירי המוצרים שלנו מטבלת Products

SELECT productname, unitprice, unitprice*0.9 AS AfterDiscount FROM products

ובעמודה החדשה נקבל 90 אחוז מהמחיר של כל מוצר

אופרטור שארית

מחזיר את השארית השלמה של פעולת חילוק

לדוגמא:

5/2 = 1
6%2 = 0
7.3%3 = 1.3

טלפון: **077-4123211/2** Email: info@iitc.co.il אתר: www.iitc.co.il

דוגמא: לקבלת שארית החלוקה של היחידות בהזמנה ביחידות במלאי, מטבלת Products :

```
SELECT productname, unitsinorder, unitsinstock, unitsinorder % unitsinstock AS Sheerit
FROM products
WHERE unitsinstock <> 0 AND unitsinorder <> 0
```

3.2. אופרטורי השוואה

משווים בין ביטויים ומחזירים אחד משלושה ערכים: **TRUE, FALSE, UNKNOWN**

NULL

NULL – מצב בו אין שום ערך בשדה (גם לא אפס או רווח)

כאשר אנו מבצעים השוואה על שדה המכיל NULL – נקבל UNKNOWN בתור תוצאה

כדי למצוא את ה-NULL בטבלה מסוימת עלינו להשתמש באופרטור IS NULL

דוגמא:

```
SELECT CustomerID, CompanyName, Address, Fax FROM Customers WHERE fax IS NULL
```

ונקבל בתור תוצאה את העמודות שיש בהן ערך NULL בשדה FAX

הערה: אם נכתוב את השאילתה כך שהסוף שלה יהיה fax = NULL, לא נקבל שום תוצאה מאחר שהשוואה זו מחזירה ערך FALSE

שוויון =

נשתמש באופרטור זה כאשר נרצה לבחור ערך אחד מתוך ערכים רבים

דוגמא:

```
SELECT * FROM products WHERE productname = 'Ikura'
```

ונקבל את המוצר העונה לשם הזה

כאשר נרצה לבחור קבוצה מסוימת מתוך ערכים רבים:

```
SELECT * FROM products WHERE supplierID = 4
```

ונקבל את המוצרים אשר ה- SupplierID שלהם הוא 4

אופרטורי השוואה פשוטים:

=
<
>
=>
=<
<> (שונה מ)
=! (שונה מ)
<!
>!

אופרטורים פשוטים יכולים לעמוד מול מספרים, מחרוזות ותאריכים.

ערך מחרוזתי או תאריכי צריך להופיע בגרשים, ואילו ערך מספרי לא.

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

דמיון LIKE

- אם נרצה לשלוף את כל העובדים בשמם יש את האות R, נכתוב:

SELECT * FROM Employees WHERE firstname LIKE '%R%'

- אם נרצה לשלוף את כל העובדים שהאות השניה בשמם היא האות A, נכתוב:

SELECT * FROM Employees WHERE firstname Like '_A%'

אם נרצה לשלוף את כל העובדים ששמם מתחיל באות J, נכתוב:

SELECT * FROM Employees WHERE firstname LIKE 'J%'

שרשור מחרוזת

הסימן + משמש לשירשור שתי מחרוזות, אך אם אחת המחרוזות הינה NULL – תקבל כל המחרוזת ערך NULL

דוגמא:

אם נרצה לשרשר את השם הפרטי ושם המשפחה בטבלת Employees:

SELECT firstname + lastname AS FULLNAME FROM Employees

ונקבל את שמות העובדים כשהם דבוקים אחד לשני, לדוגמא NancyDavolio

ניתן גם להוסיף תווים/מילים בין מחרוזות:

SELECT firstname + ' ' + lastname AS FULLNAME FROM Employees

ונקבל את שמות העובדים עם רווח בין השם הפרטי לשם המשפחה: Nancy Davolio

3.3. אופרטורים לוגיים

מפרידים בין שתי התניות או יותר בהוראת WHERE של משפט SQL

ישנם שלושה אופרטורים לוגיים עיקריים: **AND, OR, NOT**

AND

דורש ששני הביטויים משני צידיו יחזירו TRUE

אם אחד מהם יחזיר ערך FALSE אז ה-AND יחזיר FALSE

דוגמא: מציאת סוכני המכירות הגברים מתוך טבלת העובדים

SELECT * FROM Employees WHERE Title = 'Sales Representative' AND TitleOfCourtesy = 'Mr.'

OR

מספיק שאחת ההתניות במשפט OR מתקיימת ונקבל בחזרה ערך TRUE

דוגמא: נכתוב את השאילתא הקודמת, עם OR במקום AND

SELECT * FROM Employees WHERE Title = 'Sales Representative' OR TitleOfCourtesy = 'Mr.'

עתה בתוצאה נקבל שמות נוספים, מאחר והם ענו על לפחות אחד מהתנאים בשאילתא: סוכן מכירות או גבר

NOT

כל מה שאינו עונה על ההתניה בשאילתא:

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

אם ההתניה שעליה מופעל ה-NOT מקבלת ערך TRUE, הרי שהיא תקבל ערך FALSE, ולהיפך
דוגמא: נרצה לקבל את שמות המשפחה של העובדים שאינם מתחילים באות D

```
SELECT lastname FROM Employees WHERE lastname NOT LIKE 'D%'
```

ניתן גם להשתמש ב-NOT יחד עם האופרטור IS (המצורף ל-NULL)
דוגמא: נרצה לקבל את כל הלקוחות שיש להם מספר פקס (שאינו NULL)

```
SELECT CustomerID, companyname, Phone, Fax FROM Customers WHERE Fax IS NOT NULL
```

3.4. אופרטורים שונים

IN

קובע האם ערך מסוים נמצא בקבוצה של ערכים (מתאים לקבוצת ערכים מספריים או תוויים)
דוגמא למציאת ערך מספרי: כדי למצוא את העובדים שמספרם 1, 6 או 8 בטבלת Employees :

```
SELECT FirstName FROM Employees WHERE EmployeeID IN (1, 6, 8)
```

דוגמא למציאת ערך תווי: כדי למצוא את העובדים ששם משפחתם הינו Fuller או King:

```
SELECT FirstName FROM Employees WHERE LastName IN ('Fuller', 'King')
```

BETWEEN

קובע האם ערך מסוים נמצא בתחום של ערכים
מתאים לקבוצת ערכים מספריים או תאריכים
הפרמטרים בשאילתא זו הינם כוללניים
דוגמא – כדי למצוא את העובדים בעלי שלוחה מספר 3000 עד 4000 (כולל)

```
SELECT firstname, Extension FROM Employees WHERE Extension BETWEEN 3000 AND 4000
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

4. פונקציות

מאפשרות לבצע פעולות שונות על נתונים

4.1. פונקציות צבירה

מכונות גם פונקציות קבוצתיות (Group Functions)

מחזירות ערך על סמך הערכים שבעמודה: COUNT, SUM, AVG, MAX, MIN, VAR, STDEV

COUNT

מחזירה את מספר הפריטים בקבוצה שעונים על ההתניה בהוראת WHERE

דוגמא – כדי לקבל את מספר הרשומות בטבלת Employees

SELECT COUNT (*) FROM Employees

כדי לדעת כמה סוכני מכירות יש בחברה:

SELECT COUNT (*) FROM Employees WHERE Title = 'Sales Representative'

כדי לדעת כמה עובדים עם ערך בעמודת Region ישנם בחברה:

SELECT COUNT (Region) FROM Employees

כדי לדעת כמה עובדים מאיזור שונה ישנם בחברה:

SELECT COUNT (DISTINCT Region) FROM Employees

SUM

מחזיר את סכום הערכים שבקבוצה

פועל על מספרים בלבד

דוגמא – כדי לחשב את סכום מחיר כל המוצרים (עמודת UNITPRICE) בטבלת Orders:

SELECT SUM(UnitPrice) AS TotalPrice FROM Products

כדי לחשב את סה"כ מחיר המוצרים, סה"כ כמות במלאי, סה"כ כמות בהזמנות, מטבלת Orders:

SELECT SUM(UnitPrice) AS TotalPrice, SUM(UnitsInStock) AS TotalInStock,
SUM(UnitsOnOrder) AS TotalOnOrders FROM Products

AVG

מחשבת את הממוצע של עמודה/קבוצה

מתעלמת מערכי NULL

פועלת רק על מספרים

דוגמא – כדי למצוא את ממוצע הפריטים בהזמנות:

SELECT AVG(UnitsOnOrder) AS 'Average Units On Order' FROM Products

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

MAX

מוצאת את הערך הגדול ביותר בעמודה או קבוצה
פועלת גם על מספרים וגם על תווים

דוגמא 1: כדי לגלות מהו מספר הפריטים הגדול ביותר בהזמנה נכתוב:
SELECT MAX (UnitsOnOrder) AS 'MAX Units In Order' FROM Products

דוגמא 2: כדי לגלות את שם המוצר בעל הערך הגבוה ביותר (הקרוב ביותר ל-Z) נכתוב:
SELECT MAX (ProductName) AS 'Max Name' FROM Products

MIN

אותו עיקרון כמו MAX
מוצאת את הערך הנמוך ביותר בעמודה או קבוצה
פועלת גם על מספרים וגם על תווים

דוגמא 1: כדי לגלות מהו מספר הפריטים הקטן ביותר בהזמנה נכתוב:
SELECT MIN (UnitsOnOrder) AS 'Min Units In Order' FROM Products

דוגמא 2: כדי לגלות את שם המוצר בעל הערך הנמוך ביותר (הקרוב ביותר ל-A) נכתוב:
SELECT MIN (ProductName) AS 'Min Name' FROM Products

MIN-ו MAX

ניתן לשלב את MIN ו-MAX כדי לקבל תחום
דוגמא: כדי לראות את מספר הפריטים הקטן ביותר והגדול ביותר בהזמנות נכתוב:
SELECT MIN (UnitsOnOrder) AS 'Min Units', MAX (UnitsOnOrder) AS 'Max Units' FROM Products

VAR

פונקציית השונות (Variance) – מפיקה את ריבוע סטיית התקן
חשובה מאד בחישובים סטטיסטיים רבים
פועלת על מספרים בלבד

דוגמא: כדי למצוא את השונות של עמודת UnitsOnOrder נכתוב:
SELECT VAR (UnitsOnOrder) AS 'Shonut units in order' FROM Products

STDEV

מחשבת את סטיית התקן (Standard Deviation) של עמודת מספרים
חשובה מאד בחישובים סטטיסטיים רבים
פועלת על מספרים בלבד

דוגמא: כדי לגלות את סטיית התקן של עמודת UnitsOnOrder נכתוב:
SELECT STDEV (UnitsOnOrder) AS 'Standard Deviation of units in order' FROM Products

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

פונקציות צבירה – צירופים

ניתן להשתמש בפונקציות הצבירה בצירופים שונים

דוגמא:

```
SELECT COUNT (*) AS 'Count',
 AVG (UnitsInStock) AS 'Average Units',
MIN (UnitsOnOrder) AS 'Min Units in order',
MAX (UnitsOnOrder) AS 'Max Units in order',
STDEV (UnitsOnOrder) AS 'STDevation Units in order',
VAR (UnitsOnOrder) AS 'Variance units in order',
SUM (UnitPrice) AS 'Sum Units Price',
FROM Products
```

4.2. פונקציות חשבוניות

מיועדות לשימושים מתמטיים (ABS, CEILING/FLOOR, ROUND, EXP, POWER, SQRT, SIGN)

ABS

מחזירה את הערך המוחלט של מספר עליו מצביעים

משנה את הערכים השליליים לחיוביים, ואינה משנה את הערכים החיוביים

דוגמא: לקבלת הערך המוחלט של עמודה A מטבלת Numbers:

```
SELECT ABS(A) AS Absolute_Value FROM Numbers
```

CEILING / FLOOR

CEILING – מחזירה את המספר השלם, הקרוב והגדול מהארגומנט שלה או שווה לו

FLOOR – מחזירה את המספר השלם, הקרוב והקטן מהארגומנט שלה או שווה לו

דוגמא: נרצה קודם כל לשלוף מספר מחירי פריטים מטבלת Products :

```
SELECT ProductID, ProductName, UnitPrice FROM Products WHERE ProductID IN
(5,14,15,18)
```

כעת נכתוב את השאילתא הבאה:

```
SELECT UnitPrice, CEILING (UnitPrice) AS 'Ceiling', FLOOR (UnitPrice) AS 'Floor'
 FROM Products
WHERE ProductID IN (5,14,15,18)
```

התוצאה: נקבל בטבלה את שני הערכים (הגדול והקטן הקרובים לערך)

נניח נתון מחיר הפריט 21.35, נקבל בשדה ה-Ceiling 22, ובשדה ה-Floor נקבל 21

ROUND

מעגלת מספר שהתקבל למספר הקרוב ביותר

דוגמא:

```
SELECT UnitPrice, ROUND (UnitPrice, 0) AS 'ROUND'
FROM Products
WHERE ProductID IN (5,14,15,18)
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

EXP

מאפשר להעלות בחזקה

דוגמא:

SELECT UnitPrice, EXP (UnitPrice) AS 'Exp UnitPrice' FROM Products

SQRT

מחזיר את השורש הריבועי של מספר (המספר חייב להיות חיובי)

דוגמא:

SELECT UnitPrice, SQRT (UnitPrice) AS 'SQRT' FROM Products

SIGN

מחזירה:

1- אם המספר קטן מאפס (שלילי)

0 אם המספר שווה לאפס

1 אם המספר גדול מאפס (חיובי)

ניתן להשתמש ב-SIGN גם בהוראת WHERE

דוגמא: ברצוננו להציג את הסימן של הערך בעמודת UnitsOnOrder בטבלת Products

SELECT UnitsOnOrder, SIGN (UnitsOnOrder) AS 'Sign' FROM Products

דוגמא לשימוש ב-SIGN בהוראת WHERE:

ברצוננו להחזיר את שם המוצר וכמה מוצרים בהזמנה רק ממוצרים שיש להם הזמנות (UnitsOnOrder>0), לכן נכתוב:

SELECT ProductName, UnitsOnOrder FROM Products WHERE SIGN (UnitsOnOrder) = 1

4.3. פונקציות תווים

ישנן פונקציות תווים רבות, המטפלות בתווים ובמחרוזות

הפונקציות השכיחות ביותר הן:

CHAR/ASCII,UPPER/LOWER,REPLICATE,SPACE,LEN,LTRIM,RTRIM,REPLACE,SUBSTRING, PATINDEX,CHARINDEX

CHAR

מקבלת מספר שלם ומחזירה את התו המיוצג על ידו בטבלת ASCII

דוגמא:

SELECT CHAR (72)

ונקבל את האות H

ASCII

מקבלת תו ומחזירה את ערכו המספרי לפי טבלת ASCII

דוגמא:

SELECT ASCII ('H')

ונקבל את הערך 72

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

UPPER/LOWER

UPPER – מחזירה את כל התווים כאותיות גדולות

LOWER – מחזירה את כל התווים כאותיות קטנות

דוגמא: כדי לראות את כל השמות הפרטיים כתובים באותיות גדולות ובאותיות קטנות נכתוב:

```
SELECT firstname, UPPER (FIRSTNAME) AS Capital,
LOWER (FIRSTNAME) AS Small
FROM Employees
```

REPLICATE

חוזר על מחרוזת מסוימת מספר מוגדר של פעמים

דוגמא:

```
SELECT FIRSTNAME, REPLICATE (FIRSTNAME,2) AS DOUBLENAME FROM Employees
```

SPACE

מחזיר את המחרוזת עם מספר תווי רווח שהגדרנו לפני/אחרי המחרוזת

דוגמא: כדי לראות את כל השמות הפרטיים כתובים עם רווח של 4 תווים לפני השם ואחרי השם:

```
SELECT FirstName,
SPACE (4) + FirstName AS SPACESFIRST,
FIRSTNAME + SPACE (4) AS SPACESLAST
FROM Employees
```

LEN

מחזירה את אורך הארגומנט

עבור מחרוזת: מחזירה את מספר התווים במחרוזת

עבור משתנה: את מספר הבתים כדי לאחסן משתנה זה

לא סופר רווחים בסוף המחרוזת

דוגמא: כדי לראות את מספר התווים עבור כל שם פרטי:

```
SELECR FIRSTNAME, LEN (FIRSTNAME) AS NUMCHAR FROM Employees
```

והתוצאה שנקבל היא (למשל עבור השם Nancy נקבל את הספרה 5)

LTRIM / RTRIM

LTRIM – מקצץ תאי רווח מצד שמאל של המחרוזת

RTRIM – מקצץ תאי רווח מצד ימין של המחרוזת

REPLACE

מחזירה מחרוזת שבה מוחלף חלק מהמחרוזת בחלק אחר, מספר מוגדר של פעמים

מקבלת 3 ארגומנטים: המחרוזת המקורית, תת-המחרוזת אותה רוצים להחליף ותת-המחרוזת המחליפה

דוגמא: להחלפת כל המופעים של תת המחרוזת 'ha' ב-'**' בשם המשפחה:

```
SELECT LASTNAME, REPLACE (LASTNAME,'ha','**') AS REPLACEMENT FROM Employees
```

נקבל את התוצאה: למשל עבור שם המשפחה Buchanan נקבל Buc**nan

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

SUBSTRING

מחלצת קטע ממחרוזת היעד

דוגמא: להצגת 3 התווים מהתו השני בשם הפרטי:

SELECT FIRSTNAME, SUBSTRING (FIRSTNAME, 2, 3) AS Tat_Machrozet FROM Employees

התוצאה שנקבל: למשל מהשם Nancy נקבל 'anc'

PATINDEX

מחזירה מיקום במחרוזת היעד בו נמצאת תבנית מסוימת

דוגמא: למציאת המיקום הראשון של האות 'O' בשם המשפחה:

SELECT LASTNAME, PATINDEX ('%O%', LASTNAME) AS O_POSITION

התוצאה שנקבל: למשל עבור השם Davolio נקבל את הערך 4

CHARINDEX

פונקציה המקבלת תו לחיפוש, מחרוזת וממיקום התו ממנו תתחיל לחפש במחרוזת, ומחזירה את מיקום התו במחרוזת

דוגמא:

SELECT CHARINDEX ('l', 'MichalTal',7)

נקבל את התוצאה 9 – כלומר המיקום של התו L במחרוזת, כאשר הוא מתחיל לחפש מהתו השביעי הינו 9

4.4. סוגי נתונים תאריכים וזמנים

ב-SQL ישנם שלושה טיפוסי נתונים סטנדרטיים לאחסון תאריך ושעה (DATETIME): DATE, TIME, DATETIME

בנוסף להם ישנם טיפוסי נתונים לתאריכים וזמנים, ייחודיים לכל יישום SQL

DATE

מאחסן תאריכים ככתבם

מבנה: YYYY-MM-DD

TIME

מאחסן שעות ככתבן

מבנה: HH:MI:SS.nn (HH – שתי הספרות של השעה, MI – שתי הספרות של הדקה, SS – שתי הספרות של השניות, nnnnnnn – 0 עד 7 ספרות המייצגות חלקי השניה

DATETIME

מאחסן תאריכים ושעות ככתבם

מבנה: YYYY-MM-DD HH:MI:SS.nn

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

<u>טיפוסי תאריכים וזמנים T-SQL</u>

טיפוסי נתונים של תאריכים וזמנים, ייחודים ל-T-SQL (SQL Server):

SMALLDATETIME, DATETIMEOFFSET

SMALLDATETIME

מאחסן את פרטי התאריך והשעה, אך מכיל תחום ערכים קטן יותר מאשר DATETIME

מבנה: YYYY-MM-DD HH:MI:SS

תחום תאריך: 1900-01-01 עד 2079-06-06

תחום זמן: 00:00:00 עד 23:59:59.999

DATETIMEOFFSET

מאחסן את פרטי התאריך והשעה, תוך התחשבות באזורי זמן שונים

תחביר: YYYY-MM-DD HH:MM:SS[.nnnnnnn]

תחום התחשבות באיזור הזמן: 14:00- עד 14:00+

<u>אזורי זמן</u>

IST - Israel Standard Time (UTC+2)

4.5. פונקציות תאריך ושעה

פונקציות לטיפול במושגי שעות ותאריכים:

GETDATE,DATEADD,DATEDIFF,DATEPART,DATENAME,DAY,MONTH,YEAR

GETDATE

מחזירה את התאריך והשעה של המחשב

פועלת גם מחוץ להוראת SELECT (למשל בהוראת WHERE)

<u>דוגמא:</u> כדי לראות את התאריך של היום, נכתוב:

```
SELECT GETDATE() AS CurrentDate
```

DATEADD

פונקציה המוסיפה מספר ימים/חודשים/שנים לתאריך

תחביר הפונקציה: (שדה תאריך, מספר להוספה, חלק התאריך) DATEADD

פועלת גם מחוץ להוראת ה-SELECT (למשל בהוראת WHERE)

<u>דוגמא:</u> כדי להוסיף שנתיים לשנת הלידה של העובד נכתוב:

```
SELECT BirthDate,
 DATEADD (yy,2,BirthDate) AS 'New Birthdate'
 FROM Employees
```

DATEDIFF

מחזיר את ההפרש בין תאריך X ל-Y

ניתן לבצע פקודה זו כדי לראות את ההפרש בימים, שבועות, חודשים, שנים, רבעון, שעות, דקות, שניות, מילישניות, מיקרו שניות, ננו שניות

תחביר הפונקציה: (תאריך סיום, תאריך התחלה, חלק התאריך) DATEDIFF

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

דוגמא: כדי לראות באיזה גיל היה כל עובד בזמן קבלתו לעבודה (ההפרש בשנים בין תאריך ההולדת לתאריך קבלתו לעבודה):

```
SELECT birthdate, hiredate, DATEDIFF (yy,birthdate,hiredate) AS 'Hired at age' FROM
Employees
```

DATEPART

מחזיר ערך שלם של חלק מתאריך

תחביר: DATEPART (datepart, date)

- דוגמא: לקבלת חלק השנה מהתאריך 27/7/09 נכתוב:

```
SELECT DATEPART (yyyy, '2009-07-27')
```

ונקבל את התוצאה 2009

- לקבלת חלק החודש של אותו תאריך :

```
SELECT DATEPART (month,'2009-07-27')
```

ונקבל את התוצאה 7

- לקבלת השנה של התאריך הנוכחי:

```
SELECT DATEPART (yy,getdate())
```

DATENAME

מחזיר ערך טקסט של חלק מהתאריך (את שמו)

תחביר: DATEPART (datepart, date)

דוגמא: לקבל שם החודש של תאריך 27/7/09:

```
SELECT DATENAME (month,'2009-07-27')
```

ונקבל את התוצאה: July

DAY

DAY (Date) – מחזירה מספר שלם בין 1 ל-31 המייצג את היום בחודש

דוגמא: הצגת היום בו נולד כל עובד:

```
SELECT birthdate, day (birthdate) AS 'Born in day' FROM Employees
```

MONTH

MONTH (Date) – מחזירה מספר שלם בין 1 ל-12 המייצג את החודש בשנה

דוגמא: הצגת החודש בו נולד כל עובד:

```
SELECT birthdate, month (birthdate) AS 'Born in month' FROM Employees
```

YEAR

YEAR (Date) – מחזירה מספר שלם המייצג את השנה

דוגמא: הצגת השנה בה נולד כל עובד:

```
SELECT birthdate, year (birthdate) AS 'Born in year' FROM Employees
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

4.6. פונקציות המרה

מספקות דרך נוחה להמרת נתונים מסוג אחד לסוג אחר
ההמרה מתבצעת בשליפת הנתונים בלבד, אינה משנה את סוג הנתונים בעמודה בטבלה עצמה
נשתמש בהמרה כאשר נרצה להשוות שני שדות (או ערכים) מסוגי נתונים שונים
סוגי נתונים: מחרוזת, מספר (ארוך, עשרוני, יחיד, כפול), בינארי, מונה, תאריך/זמן, מטבע

STR

ממיר נתון מסוג מספר למחרוזת
דוגמא: להמרת נתוני ה-EmployeeID (נתון מסוג מספר) לנתונים מסוג מחרוזת, בטבלת Employees

```
SELECT EmployeeID, STR (EmployeeID) AS EmpString FROM Employees
```

CONVERT / CAST

ממירות נתון מסוג אחד לנתון מסוג אחר
שתי הפעולות בעלות אותו תפקוד, אך עם תחביר שונה
CAST מוגדרת כסטנדרט ב-ANSI
CONVERT אינה סטנדרטית, אך מאפשרת להוסיף סגנון
כאשר ממירים סוגי נתונים מספריים, הנבדלים במאפייניהם (למשל המרה מנתון Decimal לנתון Integer), לעיתים התוצאה נחתכת או נעגלת, על מנת להתאימה לסוג הנתונים הרצוי
דוגמאות:

- ההמרה הבאה תניב תוצאה 10:

```
SELECT CAST (10.6496 AS int)
```

אנו רואים כי הספרות לאחר הנקודה נחתכו

- ההמרה הבאה תניב תוצאה 10.3497:

```
SELECT CAST (10.3496847 AS money)
```

התוצאה עוגלה כדי להתאים לסוג הנתונים money, הכולל רק 4 ספרות לאחר הנקודה

- ההמרה הבאה תניב מספר שלם 3:

```
SELECT CONVERT (int, 3.14765)
```

התוצאה נחתכה כדי להתאים לסוג הנתונים Integer, שאינו כולל נקודה עשרונית

המרות של תאריכים

בהמרות מסוג זה משתמשים בעיקר לשינוי טיפוס הנתונים של התאריך, על מנת: 1) לבצע השוואה בין ערכי תאריך מטיפוסי נתונים שונים. 2) לעיצוב ערכי תאריך כמחרוזת תווים. 3) להמרת מחרוזת תווים למבנה של תאריך

CAST

דוגמא להמרת תאריך (המוגדר כסוג DateTime) למחרוזת:

```
SELECT GETDATE () AS CurrentDate,
CAST (GETDATE () AS char(12)) AS String
```

התוצאה: 2009-07-16 10:46:31.737 יהפוך ל-Jul 16 2009

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

CONVERT

כאשר הביטוי שאנו ממירים הינו תאריך או זמן, ה-Style מייצג את הפורמט אליו אנו רוצים להמיר
דוגמא: להצגת תאריך התחלת עבודה עבור העובדים, בפורמט אירופאי, עם 2 ספרות בחלק של השנה (Style 3), כסוג נתונים: תווים, עלינו לכתוב:

```
SELECT HireDate,
 CONVERT (char,Hiredate, 3)
 AS New_Date FROM Employees
```

תוצאה: מ-HireDate שמופיע כ- 1992-05-01 נקבל: 01/05/92

דוגמא להמרת תאריך (המוגדר כ-DateTime) למחרוזת בעיצוב אמריקאי (mm/dd/yyyy) ולמחרוזת בעיצוב אירופאי (dd/mm/yyyy) :

```
SELECT GETDATE () AS CurrentDate,
CONVERT (char(12), GETDATE (), 101) AS StringAmerican,
CONVERT (char(12), GETDATE (), 103) AS StringEurope,
```

נקבל את התוצאה: 2009-07-16 10:46:31.737 בערך CurrentDate יקבל את הערכים הבאים:
07/16/2009 באמריקאי, ו-16/07/2009 באירופאי

4.7. פונקציות כלליות

SUSER_NAME

מחזירה את שם המשתמש הנוכחי במסד הנתונים

```
SELECT suser_name ()
```

ISNULL

מחליפה NULL בערך מסוים (בתצוגה בלבד, לא בטבלה עצמה)
דוגמא: הצגת שמות העובדים ושם האיזור שלהם, ועבור אלו שאין להם איזור – הצג Unknown

```
SELECT LastName,
 ISNULL (Region, 'Unknown') AS Region
 FROM Employees
```

TOP

מחזירה רק את מספר השורות העליונות (ראשונות) שצוין
מספר השורות המוחזר יכול להיות במספרים או באחוזים
ניתן להשמש ב-TOP בפקודות UPDATE, DELETE, SELECT, INSERT
דוגמאות:

- בטבלת Orders ישנם 830 רשומות
- ברצוננו לשלוף את 7 הלקוחות הראשונים מטבלה זו

```
SELECT TOP (7) CustomerID FROM Orders
```

- שליפת 7% הלקוחות הראשונים מטבלת Orders

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

SELECT TOP (7) PERCENT CustomerID FROM Orders

תוצאה: נקבל את 59 השורות הראשונות

5. הוראות בשאילתות

5.1 ORDER BY

בשאילתת SELECT FROM רגילה, סדר הצגת הנתונים הינו לפי סדר הזנתם (בטבלה ללא מפתח ראשי), ללא מיון (הערה: במידה ולטבלה יש מפתח ראשי – הנתונים יוצגו ממוינים לפי המפתח הראשי).

ORDER BY מציגה את תוצאות השאילתה בסדר כלשהו

<u>דוגמא 1:</u> להצגת נתוני העובדים ממוינים לפי שדה FirstName בסדר עולה:

```
SELECT *
FROM Employees
ORDER BY FirstName
```

<u>דוגמא 2:</u> להצגת הנתונים ממוינים לפי שדה LastName בסדר יורד (מהסוף להתחלה, מ-Z ל-A)

```
SELECT *
FROM Employees
ORDER BY LastName DESC
```

ניתן להפעיל את ORDER BY על יותר משדה אחד

יש משמעות לסדר הופעת העמודות בהוראות ה-ORDER BY:

מיון ראשי – לפי השדה הראשון בהוראה

מיון משני – לפי השדה המשני בהוראה

ניתן להפעיל את ORDER BY גם על עמודה שאינה מופיעה במשפט ה-SELECT

<u>דוגמא:</u> להצגת הנתונים ממוינים לפי שדה Title ו-LastName:

```
SELECT EmployeeID, Title, LastName, FirstName, FROM Employees
ORDER BY Title, LastName
```

5.2 GROUP BY

מקבצת ארגומנטים בקבוצות, לפי המוגדר לה

כל קבוצה מופיעה כשורה אחת בשליפה

פועלת בשילוב פונקציות צבירה המתוארות במשפט ה- SELECT (COUNT, SUM, AVG, MIN, MAX)

ניתן להשתמש ביותר מפונקציית צבירה אחת

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

דוגמא: על מנת לראות סה"כ מחיר עבור כל הזמנה בטבלת Order Details:

```
SELECT OrderID, SUM (UnitPrice) AS Total
FROM [Order Details]
GROUP BY OrderID
```

דוגמא לשילוב של כמה פונקציות צבירה עם הוראת GROUP BY:
כדי לראות כמה פריטים ישנם בכל הזמנה וסה"כ מחיר עבור כל הזמנה:

```
SELECT OrderID, SUM (UnitPrice) AS Total,
COUNT (OrderID) AS NumOfItems
FROM [Order Details]
GROUP BY OrderID
```

שים לב: כל עמודה בחלק ה-SELECT, שלא מבצעים עליה פונקציית צבירה – חייבת להופיע בחלק ה-GROUP BY

5.3. HAVING

מגדירה תנאי חיפוש לקבוצות או פונקציות צבירה, בדומה להוראת WHERE
הוראות WHERE אינה יכולה לעבוד עם פונקציות צבירה
לשם כך יש הוראת HAVING, הפועלת עם פונקציות צבירה וקבוצות (GROUP BY)
דוגמא 1: ניתן להשתמש בפונקציית צבירה בתוך הוראת HAVING: כדי לראות את סה"כ המחיר של כל ההזמנות שה-TOTAL שלהן גדול מ-60:

```
SELECT OrderID, SUM (UnitPrice) AS Total
FROM [Order Details]
GROUP BY OrderID
HAVING (SUM (UnitPrice) > 60)
```

דוגמא 2: כדי לראות סה"כ מחיר עבור כל הזמנה, רק עבור הזמנות שמספרן גדול מ-10260:

```
SELECT OrderID, SUM (UnitPrice) AS Total
FROM [Order Details]
GROUP BY OrderID
HAVING (OrderID > 10260)
```

ניתן להשתמש גם באופרטורים הבאים בתוך הוראת HAVING:

- אופרטורי השוואה: >, =>, <, <=, <>, =
- אופרטורים לוגיים: AND, OR, NOT
- אופרטור IN

לא ניתן להשתמש בסוגי הנתונים הבאים בתוך הוראת HAVING: text, ntext, image

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

5.4. שילוב הוראות

SQL מאפשר לכתוב שאילתות המורכבות משילוב הוראות
בהוראות בהן נרצה להשתמש גם ב-GROUP BY וגם ב-ORDER BY: על הוראת ה-GROUP BY להופיע לפני הוראת ה-ORDER BY

<u>דוגמא:</u> עבור כל הזמנה מטבלת Order Details הצג:

- סה"כ לתשלום
- מספר הפריטים בהזמנה
- רק עבור הזמנות בהן יש יותר מ-3 פריטים
- הנתונים ימויינו לפי סה"כ לתשלום להזמנה, בסדר יורד

```
SELECT OrderID, SUM (UnitPrice) AS Total,
COUNT OrderID AS 'Num of Itmes'
FROM [Order Details]
GROUP BY OrderID
HAVING COUNT (OrderID) > 3
ORDER BY Total desc
```

<u>סיכום</u>

WHERE פועלת בדרך כלל בשאילתות הפועלות על שורות
GROUP BY ו- HAVING פועלות בשילוב עם פונקציות צבירה
ORDER BY פועלת עם: GROUP BY, WHERE, HAVING

6. צירוף טבלאות

SQL מאפשר לשלוף נתונים מכמה טבלאות בו זמנית בשאילתה אחת
ישנן מספר דרכים לצירוף טבלאות: בעזרת פסיקים, בעזרת פקודת JOIN

6.1. בעזרת פסיקים

תחביר:
SELECT שם טבלה. שם עמודה, שם טבלה. שם עמודה...
FROM TABLE 1, TABLE 2
WHERE קריטריון השליפה AND תנאי
כאשר (תנאי AND) הינו אופציונלי

<u>מציאת העמודה הנכונה</u>

במשפט SQL בו יש צירוף טבלאות, יש לזהות בצורה חד ערכית את שמות העמודות:

- עבור כל עמודה שיש לשלוף – מאיזו טבלה היא
- ניתן לקבוע ולהשתמש בשם חלופי לשם הטבלה (Alias) לצורך נוחות בכתיבת השאילתא ובקריאתה
-

<u>דוגמא</u>: שליפת שמות המוצרים ושמות הקטגוריות שלהם, משתי טבלאות: Categories ו-Products תוך שימוש ב-Aliases:

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

```
SELECT P.ProductName, C.CategoryName
FROM Products P, Categories C
```

מכפלה קרטזית

חיבור כל השורות מכל הטבלאות המוזכרות בהוראת FROM

דוגמא: טבלאות 1 ו-2 מכילות 200 שורות כל אחת

יתקבלו 40,000 שורות בתוצאה (200*200): עבור כל שורה בטבלה 1 יוצמדו כל השורות מטבלה 2

על מנת למנוע מכפלה קרטזית – יש להשתמש בקריטריון שליפה

קריטריון שליפה

קריטריון השליפה – שימוש בשדה משותף לשתי טבלאות על מנת לשלוף רק את הנתונים העונים על התנאי המוגדר

קריטריון השליפה מבוסס בדרך כלל על הקשר בין PK-FK:

Table 1.PrimaryKey = Table2.ForeignKey

דוגמא לקריטריון שליפה:

```
WHERE P.CategoryID=C.CategoryID
```

דוגמא: נתונות שתי טבלאות:

- טבלת Products מכילה נתונים על כל מוצר
- טבלת Categories מכילה נתונים על כל קטגוריה של מוצרים

שליפת שמות המוצרים ושמות הקטגוריות שלהם, משתי הטבלאות:

```
SELECT P.ProductName, C.CategoryName
FROM Products P, Categories C
WHERE P.CategoryID = C.CategoryID
```

סוגי תנאים

ניתן להוסיף אופרטורים לוגיים להוראת WHERE של השאילתה

כגון: <, =<, >, =>, <>, =, AND, OR, NOT וכו'

דוגמא: נתונות שתי טבלאות

- Orders (כוללת נתונים כלליים על כל הזמנה)
- Order Details (כוללת פירוט המוצרים בכל הזמנה)

ברצוננו לשלוף את הנתונים עבור כל הזמנה, רק עבור הזמנות שבהן כמות המוצר גדולה מ-100:

```
SELECT O.OrderID, O.CustomerID, OD.ProductID, OD.Quantity, OD.UnitPrice
FROM Orders AS O, [Order Details] AS OD
WHERE O.OrderID = OD.OrderID
AND OD.Quantity > 100
```

בצירוף שוויוני Equi-Join (=) ישנה התאמה בין ערכי העמודות בטבלה אחת לערכים המתקיימים בטבלה אחרת – רק הנתונים הזהים בשתי העמודות נשלפים

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

בצירוף בלתי שוויוני נשתמש בכל סימן פרט ל-(=) – רק הנתונים הלא זהים בשתי העמודות נשלפים
הצירוף השוויוני נפוץ הרבה יותר מהצירוף הבלתי שוויוני

6.2. צירוף – JOIN

מאפשר לאחזר במהירות מידע מטבלאות שונות ולהראותם למשתמש כאוסף נתונים אחד
יראה את הנתונים הרלוונטים משתי הטבלאות
תחביר:
...שם עמודה. שם טבלה, שם עמודה. שם טבלה SELECT

FROM TABLE1 JOIN TABLE2
ONקריטריון השליפה
WHEREתנאי

<u>דוגמא:</u>

```
SELECT P.ProductName, C.CategoryName
FROM Products P INNER JOIN Categories C
ON P.CategoryID = C.CategoryID
WHERE CategoryName < 'd'
```

ישנם שלושה צירופים JOIN אפשריים:

- צירוף פנימי (INNER JOIN)
- צירוף חיצוני (OUTER JOIN)
- צירוף מוצלב (CROSS JOIN)

הערה: בצירוף החיצוני נכללים גם RIGHT JOIN, LEFT JOIN, FULL JOIN

<u>צירוף פנימי – INNER JOIN</u>

צירוף בו ישלפו רק השורות שלהן יש ערך תואם בשתי הטבלאות
הצירוף הפנימי משתמש בתנאי השליפה כדי לקבוע התאמה
צירוף זה הינו ברירת המחדל בשאילתות JOIN
<u>דוגמא:</u> כדי לשלוף את שמות המוצרים ושמות הקטגוריות שלהם, עבור קטגוריות ששמם קטן מ-d, מסודר לפי שם הקטגוריה:

```
SELECT P.ProductName, C.CategoryName
FROM Products P INNER JOIN Categories C
ON P.CategoryID = C.CategoryID
WHERE CategoryName < 'd'
ORDER BY CategoryName
```

<u>צירוף חיצוני – OUTER JOIN</u>

מקבץ את הנתונים בדרך שונה מהצירוף הפנימי
מציג את כל השורות של טבלה אחת, גם אילו שאינן כוללות נתונים תואמים לתנאי השליפה מהטבלה השניה

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

סוגי צירופים חיצוניים:

- צירוף חיצוני ימני (RIGHT OUTER JOIN)
- צירוף חיצוני שמאלי (LEFT OUTER JOIN)
- צירוף חיצוני מלא (FULL OUTER JOIN)

אין הכרח לכתוב את המילה OUTER בשאילתא

צירוף חיצוני ימני – RIGHT OUTER JOIN

מחזיר את כל הנתונים מהטבלה הימנית בהוראת ה-FROM
מציב ערכים ריקים (NULL) בשדות הטבלה השמאלית שאינם עונים על תנאי ההתאמה בין הטבלאות

דוגמא:

בטבלת Orders ישנן 830 הזמנות
הצגת כל ההזמנות מטבלת Orders, וציון שם הלקוח (גם הזמנות שלהן אין מספר לקוח) מטבלת Customers:

```
SELECT C.CompanyName, O.OrderID
FROM Customers C RIGHT JOIN Orders O
ON C.CustomerID = O.CustomerID
```

צירוף חיצוני שמאלי – LEFT OUTER JOIN

מחזיר את כל מערכת הנתונים מהטבלה השמאלית בהוראת ה-FROM
מציב ערכים ריקים (NULL) בשדות הטבלה השמאלית שאינם עונים על תנאי ההתאמה בין הטבלאות

דוגמא:

הצגת כל הלקוחות מטבלת Customers, וציון מספר ההזמנות עבור אלו שביצעו הזמנות (מטבלת Orders) :

```
SELECT C.CustomerID, O.OrderID
FROM Customers C LEFT JOIN Orders O
ON C.CustomerID = O.CustomerID
```

צירוף חיצוני מלא – FULL OUTER JOIN

מחזיר את כל התנאים הרצויים משתי הטבלאות בהוראת FROM, כולל הנתונים מכל טבלה שאינם עונים על תנאי ההתאמה בין הטבלאות
מציב ערכים ריקים (NULL) בשדות הטבלאות, שאינם עונים על תנאי ההתאמה בין הטבלאות

דוגמא:

הצגת כל הלקוחות מטבלת Customers, וכל ההזמנות מטבלת Orders:

```
SELECT C.CustomerID, O.OrderID
FROM Customers C FULL JOIN Orders O
ON C.CustomerID = O.CustomerID
ORDER BY O.OrderID
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

צירוף מוצלב – CROSS JOIN

צירוף מוצלב = מכפלה קרטזית: הכפלת מספר השורות בטבלה 1 במספר השורות בטבלה 2
צירוף טבלאות ללא הוראת ON
במידה ומוסיפים הוראת ON – צירוף זה מתנהג כמו צירוף פנימי (INNER JOIN)

דוגמא: שליפת כל הלקוחות מטבלת Customers, ועבור כל לקוח – כל ההזמנות מטבלת Orders:

```
SELECT C.CustomerID, O.OrderID
FROM Customers C CROSS JOIN Orders O
ORDER BY O.OrderID
```

תוצאה: בטבלת Customers ישנן 91 רשומות ובטבלת Orders ישנן 830 רשומות
סה"כ נשלפו 75,530 שורות

אם ננסה לבצע שליפה לפי מכפלה קרטזית, בעזרת INNER JOIN ללא חלק ה-ON – נקבל הודעת שגיאה

צירוף עצמי

צירוף טבלה לעצמה
נשתמש במקרים בהם נרצה לצרף רשומות מטבלה לרשומות אחרות באותה טבלה
יש להשתמש ב- Aliases עבור שמות הטבלה (שם הטבלה יופיע פעמיים)

דוגמא: נתונה טבלה עובדים EMP
על מנת למצוא את שם המנהל של כל אחד מהעובדים:

```
SELECT E.FirstName EmpName, M.FirstName MngName
FROM Employees E, Employees M
WHERE E.Reports To = M.EmployeeID
```

צירוף של יותר מ-2 טבלאות

ניתן לבצע צירוף של יותר מ-2 טבלאות: על-ידי פסיקים ובאמצעות הוראת JOIN

על ידי פסיקים:

דוגמא: שליפת שם חברת הלקוח, מספר הזמנה ושם העובד שביצע ההזמנה
המידע נמצא בשלוש טבלאות: Customers, Orders, Employees

```
SELECT C.CompanyName, O.OrderID, E.FirstName
FROM Customers C, Orders O, Employees E
WHERE C.CustomerID = O.CustomerID AND O.EmployeeID = E.EmployeeID
```

הסבר: בטבלת Orders ישנם שני שדות FK: CustomerID מטבלת Customers, ו-EmployeeID מטבלת Employees
בעזרת שדות אלו ניתן לבצע את שליפת הנתונים התואמים משלושת הטבלאות

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

על ידי הוראת JOIN:

אותה דוגמא: שליפת שם חברת הלקוח, מספר ההזמנה ושם העובד שביצע את ההזמנה:

```
SELECT C.CompanyName, O.OrderID, E.FirstName
FROM Customers C JOIN Orders O
ON C.CustomerID = O.CustomerID
JOIN Employees E
ON O.EmployeeID = E.EmployeeID
```

6.3. אופרטור UNION

אופרטור זה יוצר שאילתת איחוד, אשר מצרפת את תוצאותיהן של שתי שאילתות בלתי-תלויות או יותר. משלבת תוצאות שחזרו מ-2 או יותר שאילתות לתוצאה אחת הכוללת את כל השורות השייכות לכל השאילתות

UNION מסיר מהתוצאה שורות כפולות.

חוקים לשימוש ב-UNION:

- מספר וסדר העמודות צריך להיות זהה בכל השאילתות
- סוגי הנתונים חייבים להיות תואמים

שונה מפקודת JOIN, המשלבת עמודות משתי טבלאות: JOIN אינה מחייבת זהות במספר וסדר העמודות ובסוגי הנתונים בשתי הטבלאות

דוגמא 1: נתונות 2 טבלאות, Customers ו- Suppliers

על מנת להציג את נתוני העיר, שם החברה ואיש הקשר משתי הטבלאות כסט תוצאות אחד (ללא כפילויות):

```
SELECT City, CompanyName, ContactName
FROM Customers
UNION
SELECT City, CompanyName, ContactName
FROM Suppliers
ORDER BY City, CompanyName
```

דוגמא 2: הצג את כל העובדים ששם משפחתם גדול מ-D או מ-L (ללא כפילויות):

```
SELECT EmployeeID, LastName
FROM Employees
WHERE LastName > 'D'
UNION
SELECT EmployeeID, LastName
FROM Employees
WHERE LastName > 'L'
```

UNION ALL

מחזירה רשימה משולבת משתי השאילתות, כולל שמות כפולים (אינה משמיטה כפילויות)

דוגמא: הצג את כל העובדים ששם משפחתם גדול מ-D או מ-L (עם כפילויות):

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

```
SELECT EmployeeID, LastName
FROM Employees
WHERE LastName > 'D'
UNION ALL
SELECT EmployeeID, LastName
FROM Employees
WHERE LastName > 'L'
```

7.שאילתות משנה (Subquery)

נשתמש בשאילתת משנה כאשר התנאי ב-WHERE מכיל ביטוי שלא ידוע מראש, אלא תלוי בתוכן טבלה מסוימת
שאילתת משנה היא שאילתה שתוצאותיה מועברות כארגומנט לשאילתה אחרת
שאילתה ב' (המשנית) מקוננת בתוך שאילתה א' (הראשית) ומופיעה בתוך סוגריים
תת השאילתה תחושב קודם לשאילתה הראשית
שאילתות משנה מאפשרות לקשר בין מספר שאילתות
ניתן לבצע קינון (nesting) שאילתות משנה עד 32 רמות קינון
שאילתת המשנה יכולה להופיע בהוראות ה-SELECT, WHERE, HAVING של השאילתה החיצונית
שאילתת משנה יכולה לקנן בתוך שאילתות מסוגים: SELECT, INSERT, UPDATE, DELETE

חוקים:

- רשימת ה-SELECT בתוך שאילתת משנה, הכוללת אופרטור השוואה, יכולה לכלול רק ביטוי אחד או שם עמודה אחד
- אם הוראת ה-WHERE של השאילתה הראשית כוללת שם עמודה – הוא חייב להיות תואם לשם העמודה בשאילתת המשנה

סוגי תת שאילתות

ישנם כמה סוגי תת-שאילתות:

- תת שאילתה המחזירה ערך בודד
- תת שאילתה המחזירה מספר ערכים בעמודה אחת
- תת שאילתה המחזירה מספר עמודות
-

7.1 תת שאילתה המחזירה ערך בודד

אופרטורים בסוג שאילתה זה הינם: >, =>, <, <=, <>, =,
ניתן להשתמש בפונקציות צבירה בשאילתה זו (SUM, COUNT, MIN,MAX,AVG)
שאילתות משנה המחזירות ערך בודד – אינן יכולות לכלול הוראות GROUP BY ו-HAVING

<u>דוגמא</u> לתת שאילתה המחזירה ערך בודד:
הצגת שם החברות מאותו איזור של עובד מספר 4:

```
SELECT CompanyName
FROM Customers
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

```
WHERE Region =
(SELECT Region
FROM Employees
WHERE EmployeeID = 4)
```

הסבר:

- אנו רוצים להציג נתונים מטבלה א' בלבד, אך התנאי פועל על שדה מטבלה ב' (EmployeeID=4)
- השדה המשותף לשתי הטבלאות: Region
- שאילתה ב' שולפת את הערך בשדה Region התואם לעובד מס' 4 מטבלה ב', והוא מהווה את הארגומנט לשאילתה א'

<u>דוגמא</u> לשימוש בפונקציית צבירה בתת שאילתה:
הצגת מספר המוצר, שם המוצר ומחיר יחידה מטבלת Products, עבור מוצרים שמחירם גבוה מהמחיר הממוצע ליחידה

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

הסבר:

- תת השאילתה מחזירה את הממוצע של עמודת UnitPrice מטבלת Products: 28.8663
- השאילתה הראשית מחזירה את נתוני המוצרים שמחירם גבוה ממספר זה

7.2. תת שאילתה המחזירה מספר ערכים

כאשר תת השאילתה מחזירה עמודה אחת עם מספר ערכים, נשתמש באופרטורים הבאים בשאילתה הראשית: ALL, ANY/SOME ,IN/NOT IN

<u>ALL</u>

משווה בין הביטוי בשאילתה הראשית לכל אחד מהערכים שחזרו מהתת שאילתה
מחזיר TRUE אם הביטוי בשאילתה הראשית שווה לכל אחד מהערכים שחזרו מהתת שאילתה
אם הביטוי בשאילתה הראשית אינו שווה לאחד הערכים שחזרו מהתת שאילתה – מחזיר FALSE
יכול לפעול עם אופרטורי השוואה (<, =>, >, =<, <>, =)
ייכתב לאחר סימן ההשוואה

<u>דוגמא:</u> הצגת כל המוצרים שהספקים שלהם נמצאים בלונדון:

```
SELECT S.CompanyName, P.ProductName, S.City
FROM Products P, Suppliers S
WHERE S.CompanyName = ALL
(SELECT CompanyName
FROM Suppliers
WHERE City = 'London')
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

הסבר:

- שאילתת המשנה: שולפת את שם חברות הספקים הנמצאים בלונדון מטבלת Suppliers
- האופרטור ALL = בשאילתה הראשית בודק התאמה מלאה של הספקים העונים על התנאי ואם ישנה התאמה מלאה – הוא מחזיר TRUE לשאילתה הראשית
- השאילתה הראשית מציגה רק את נתוני הספק והמוצר עבור הספקים שהוחזרו משאילתת המשנה

הרחבה:
אם הביטוי בתנאי של השאילתה הראשית הינו:
(Subquery) ALL => ביטוי WHERE
ותת השאילתה החזירה 2 ערכים: 3, 2
האופרטור ALL => יחזיר TRUE לערך 2 מהשאילתה הראשית

אם הביטוי בתנאי של השאילתה הראשית הינו:
(Subquery) ALL = ביטוי WHERE
האופרטור ALL= יחזיר FALSE, כיוון שחלק מהערכים שחזרו מתת השאילתה אינם עומדים בהשוואה (הערך 3)

- ALL> (גדול מ-ALL): גדול מהערך המקסימלי שהחזירה התת השאילתה (כלומר, מעל לטווח המוחזר מהתת שאילתה)
- ALL< (קטן מ-ALL): קטן מהערך המינימלי שהחזירה התת שאילתה (כלומר, מתחת לטווח המוחזר מהתת שאילתה)

ANY / SOME

בודקים אם ערך העמודה מהשאילתה הראשית נמצא בנתונים שהוחזרו על-ידי שאילתת המשנה
מחזיר TRUE אם הערך בשאילתה הראשית שווה לערך אחד לפחות מהערכים שהוחזרו על-ידי התת-שאילתה
מחזיר FALSE אם הערך בשאילתה הראשית אינו שווה לשום ערך מהערכים שהוחזרו על-ידי התת-שאילתה
יכול לפעול עם אופרטורי השוואה (<, =>, >, =<, <>, =)
ייכתב לאחר סימן ההשוואה

<u>דוגמא</u>: הצגת כל המוצרים שהספקים שלהם נמצאים בלונדון, פריז או תל אביב:

```
SELECT S.CompanyName, P.ProductName, S.City
FROM Products P, Suppliers S
WHERE S.CompanyName = ANY
(SELECT CompanyName
FROM Suppliers
WHERE City IN ('London', 'Paris', 'Tel Aviv'))
```

הרחבה:

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

- ANY> (גדול מ-ANY): גדול מהערך המינימלי שהחזירה התת-שאילתה (כלומר בתוך הטווח המוחזר מהתת-שאילתה)
- ANY< (קטן מ-ANY): קטן מהערך המקסימלי שהחזירה התת-שאילתה (כלומר, בתוך הטווח המוחזר מהתת-שאילתה)

IN / NOT IN

IN: בודק אם ערך העמודה מהשאילתה הראשית נמצא בנתונים שהוחזרו על-ידי שאילתת המשנה
מחזיר TRUE אם הערך בשאילתה הראשית שווה לערך שחזר על-ידי התת-שאילתה
מחזיר FALSE אם הערך בשאילתה הראשית אינו שווה לשום ערך מהערכים שהוחזרו על-ידי התת-שאילתה
NOT IN: בודק אם ערך העמודה מהשאילתה הראשית אינו נמצא ברשימת הערכים שהוחזרו על-ידי שאילתת המשנה

דוגמא לשימוש ב-IN:
הצגת כל המוצרים שהספקים שלהם נמצאים באחת מהערים לונדון, פריז, ברלין:

```
SELECT S.CompanyName, P.ProductName
FROM Products P, Suppliers S
WHERE S.CompanyName IN
(SELECT CompanyName
FROM Suppliers
WHERE City IN ('London', 'Paris', 'Berlin'))
```

הסבר:
- שאילתת המשנה: שולפת את שם חברות הספקים הנמצאים באחת מהערים המבוקשות מטבלת Suppliers
- האופרטור IN בשאילתה הראשית מעביר את שמות הספקים העונים על התנאי לשאילתה הראשית
- השאילתה הראשית מציגה רק את נתוני הספק והמוצר עבור הספקים שהוחזרו משאילתת המשנה

הערות:
- תת שאילתה שאינה מחזירה רשומות – תגרום לכך שגם השאילתה הראשית לא תחזיר רשומות, **אלא אם התנאי בחלק ה-Where של השאילתה הראשית מצפה לקבל שום רשומה מתת השאילתה**.
- תת שאילתה המחזירה ערך NULL – תגרום לכך שהשאילתה הראשית לא תחזיר שום רשומה.

7.3. תת שאילתה המחזירה מספר עמודות

כאשר תת השאילתה מחזירה ערכים ממספר עמודות, נשתמש באופרטור EXISTS בשאילתה הראשית
EXISTS בודק האם תת השאילתה החזירה שורות:
- אם כן, מחזיר TRUE

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

- אם לא, מחזיר FALSE

השאילתה הפנימית תבדק מחדש עבור כל שורה מהשאילתה החיצונית

EXISTS / NOT EXISTS

כאשר משתמשים באופרטור EXISTS - תת השאילתה לא באמת מחזירה שורות נתונים, אלא מחזירה אינדיקציה (True/False) האם ישנן שורות העונות על התנאי

אם השאילתה הפנימית החזירה TRUE – השורה הנבדקת מהשאילתה הראשית תהפוך תישלף

NOT EXISTS: פועל הפוך מ-EXISTS

דוגמא ל-EXISTS:

שליפת שמות הלקוחות מטבלת Customers בתנאי שקיימות להם הזמנות בטבלת Orders

```
SELECT C.CustomerId
FROM Customers C
WHERE EXISTS (SELECT O.CustomerID, O.OrderID
FROM Orders O
WHERE C.CustomerID = O.CusotmerID)
```

הסבר:

- השאילתה המשנית מבצעת שליפת שמות הלקוחות ומספרי ההזמנה שלהם מטבלת Orders כאשר שם הלקוח שווה בשתי הטבלאות (מהווה את קריטריון השליפה)
- השאילתה הראשית מבצעת שליפת שמות הלקוחות מטבלת Customers - רק אם חזרו מתת-השאילתה (הבדיקה מתבצעת עבור כל שורה מטבלת Customers)

NOT EXISTS

מחזיר רק את השורות בשאילתה הראשונה, אשר התוצאה עבורן ב-EXISTS היתה FALSE (כלומר שלא החזירו נתונים בתת שאילתה)

דוגמא: המשך הדוגמא הקודמת: שליפת שמות הלקוחות מטבלת Customers שלא ביצעו שום הזמנה

```
SELECT C.CustomerId
FROM Customers C
WHERE NOT EXISTS (SELECT O.CustomerID, O.OrderID
FROM Orders O
WHERE C.CustomerID = O.CusotmerID)
```

סיכום

- שאילתת משנה הינה שיטה להפעלת התניות נוספות על הנתונים המוחזרים משאילתה
- מאפשר לנו גמישות רבה בהגדרת התניות, בעיקר כאשר הערך המדויק אינו ידוע לנו

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

8. טיפול בנתונים

8.1. סוגי נתונים (Data Types)

כאשר יוצרים את הטבלאות בפעם הראשונה, צריך להקצות סוגי נתונים לכל עמודה בטבלה
סוג הנתונים מתאר את אופי הנתונים שיאוחסן בשדה

שלמות הנתונים (Data Integrity)

וידוא דיוק ועקביות בהזנת הנתונים ובאחסונם במסד הנתונים
שלמות הנתונים חשובה לצורך אחזור הנתונים וביצוע השוואות בין נתונים שונים
שמירת הדיוק של הנתונים כרוכה בהחלת כללים (אילוצים) על עמודות הטבלה, כדי שאפשר יהיה להזין סוגים מסוימים של נתונים בלבד
דוגמא: בעמודה המכילה שכר עובד יהיו נתונים מספריים בלבד, בעמודת NAME יהיו אותיות וספרות בלבד

8.2. אילוצים (Constraints)

אילוץ: אובייקט במסד נתונים טבלאי, האוכף כללים על הנתונים המתווספים לעמודות שבטבלה
ב-SQL ישנם מספר סוגי אילוצים:

- NOT NULL
- מפתח ראשי (Primary Key)
- ייחודיות (Unique)
- מפתח זר (Foreign Key)
- בדיקה (Check)

אילוץ NOT NULL

NULL - ערך חסר או בלתי ידוע. ללא ערך.
NOT NULL – איסור על העמודה להכיל ערך NULL (חייב להיות ערך בעמודה)
אם העמודה אינה מוגדרת כ-NOT NULL, יכולים להיות בה ערכי NULL

אילוץ מפתח ראשי (Primary Key)

מפתח ראשי – לזיהוי עמודה אחת או יותר, אשר עושה את השורה לייחודית
המטרה – מפתח ראשי ייחודי לכל רשומה
בדרך כלל המפתח הראשי מורכב מעמודה אחת בטבלה (כגון EmployeeID, CustomerID)
ניתן ליצור מפתח ראשי גם מכמה עמודות בטבלה
הקצאת המפתח הראשי נעשית בזמן יצירת הטבלה
המפתח הראשי הוא העמודה אליה פונים בדרך כלל בשאילתות ופעולות צירוף הטבלאות
המפתח הראשי הוא העמודה לפיה ממוינים הנתונים בטבלה

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

אילוץ ייחודיות (Unique)

קובע שכל ערך בעמודה יהיה ייחודי
דומה מאד למפתח הראשי
ניתן להגדיר עמודה אחת כמפתח ראשי, ולהחיל אילוץ ייחודיות על עמודה אחרת, שאינה מהווה מפתח ראשי
דוגמא: בטבלת Employees מוגדר שדה EmpID כמפתח ראשי ושדה EmpPhone כשדה ייחודי
כלומר, לא ייתכן מצב שלשני עובדים יהיה מספר טלפון זהה

אילוץ מפתח זר (Foreign Key)

מפתח זר: העמודה בטבלת הבן אשר מתייחסת לעמודה בטבלת האב
עמודה המוגדרת כמפתח זר משמשת כהפניה לעמודה המוגדרת כמפתח ראשי בטבלה אחרת
טבלת הבן תלויה בטבלת האב:

- על מנת להוסיף רשומה בטבלת הבן יש להוסיף תחילה רשומה תואמת בטבלת האב
- על מנת למחוק רשומה מטבלת האב יש למחוק תחילה את כל הרשומות התואמות מטבלת הבן

אילוצי בדיקה (Check)

ניתן להשתמש באילוצי בדיקה על מנת לבדוק את תקינות הנתונים המוזנים לעמודות מסוימות בטבלה
ישנם יישומי SQL בהם חלון היישום מגביל את הערכים אותם ניתן להזין לעמודות או לאובייקטים
אילוצי הבדיקה מהווים דרך נוספת לאספקת שכבת הגנה על הנתונים
דוגמא:
בטבלת שכר_עובדים, הוגדר שדה שכר_לשעה כשדה עם אילוץ בדיקה, המגביל את הערכים המוזנים בו לערך מינימום של 24 ₪ לשעה

8.3. טיפול בנתונים

טיפול בנתונים הנמצאים בתוך טבלה, באמצעות משפטי SQL
ישנם יישומי SQL המספקים ממשק גרפי לטיפול בנתונים
טיפול בנתונים כולל:

- משפט INSERT – הוספת נתונים לטבלה
- משפט UPDATE – עדכון נתונים בטבלה
- משפט DELETE – מחיקת נתונים מטבלה

INSERT

המשפט INSERT מאפשר לנו להוסיף נתונים לטבלה במסד הנתונים
מורכב משתי הוראות:

1. INSERT ..... VALUES
2. INSERT ..... SELECT

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

INSERT ..... VALUES

להוספת רשומה אחת לטבלה

שימושי לפעולות קטנות, אשר מטפלות במספר רשומות קטן בלבד

תחביר:

```
INSERT INTO table_name (column1, column 2…)
VALUES (value1, value2…)
```

משפט זה מוסיף רשומה לטבלה על פי העמודות שהגדרנו והערכים אותם אנו מורים לו להוסיף

יש לשמור על שלושה כללים:

1. טיפוס הנתונים של הערכים המוכנסים צריך להתאים לטיפוס הנתונים של שדות היעד
2. גודל הנתונים המוכנסים צריך להיות בתחום של גודל עמודת היעד
3. מיקום הנתונים המוכנסים ברשימת ה-Values חייב להתאים למיקום של עמודות היעד (שבחלק ה-Insert)

משפט ה-INSERT הזה אינו מחייב ציון שמות של עמודות

כאשר אין מציינים את שמות העמודות, SQL מוסיפה את הנתונים לפי סדר העמודות בטבלה

יש לזכור כי בפקודת ה-INSERT נתונים תוויים צריכים להופיע בין גרשיים, לעומת נתונים מספריים – שאינם צריכים גרשיים

<u>דוגמא:</u> כדי להכניס רשומה לטבלת Territories נרשום את המשפט הבא:

```
INSERT INTO Territories
VALUES (02345, 'Gush Dan', 2)
```

הוספת ערכי NULL – אפשרית רק כאשר אין אילוץ NOT NULL על העמודה

- במידה וקיים אילוץ NOT NULL על העמודה, ואין אנו יודעים את ערך הנתון, נוכל להכניס רווחים בעמודה במקום הנתון החסר
- SQL יתייחס אל הרווחים כאל ערך

IF NOT EXISTS

הוספת ערכים ייחודיים – במידה וישנו אילוץ ייחודיות (Unique) על העמודה אליה רוצים להכניס נתונים – נחויב להכניס רק נתונים ייחודיים לעמודה זו (לא ייתכנו שני נתונים זהים בעמודה זו)

כדי לבדוק האם הנתון אותו אנו מעוניינים להכניס קיים כבר בעמודה, נשתמש בהוראת:

```
>table_name <SELECT * FROM IF NOT EXISTS (
)>condition <WHERE
>table_name <INSERT INTO
VALUES (…..)
```


www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

דוגמא: הוספת רשומה לטבלת Employees רק בתנאי שאין רשומה עם אותו מיקוד בטבלה:

```
IF NOT EXISTS
(SELECT * FROM Employees
WHERE PostalCode = '9777'
INSERT Employees
VALUES
('Cohen', 'Moshe', 'Tester', 'Mr.', '1969-11-25', '2009-03-20','Haazmaut 40','Bat
Yam',1,'9777','Israel',03223344,null,null,null,2,null)
```

IDENTITY

בהקמת הטבלה, ניתן להגדיר עמודה מסוימת עם IDENTITY. SQL ייתן מספור ייחודי אוטומטי לעמודה זו. מתאים בעיקר לעמודת מפתח ראשי. בהגדרת עמודה עם IDENTITY מגדירים גם את הערך ההתחלתי והקפיצות. כאשר מכניסים ערכים לטבלה הכוללת שדה עם IDENTITY, אין להכניס ערך לשדה זה.

דוגמא: בטבלת Employees מוגדר שדה EmployeeID כ-IDENTITY
על מנת להכניס רשומה חדשה לטבלת Employees, אין להכניס ערך לעמודת EmployeeID:

```
INSERT INTO Employees (LastName, FirstName, Title)
VALUES ('Levi','Eshkol','President')
```

@@IDENTITY and IDENT_CURRENT ()

ל-SQL ישנם משתנים פנימים העוזרים לו "לזכור" את מספר ה-IDENTITY האחרון בכל טבלה
כדי לראות את ערך ה-IDENTITY של הטבלה האחרונה לה הכנסנו נתונים :

```
SELECT @@IDENTITY
```

(רק לאחר פקודת INSERT)

ניתן להשתמש בפונקציית IDENT_CURRENT המחזירה את ערך ה-IDENTITY של טבלה מסוימת:

```
SELECT IDENT_CURRENT ('Table_Name')
```

דוגמא 1:
לאחר הכנסת שורה לטבלת Employees:

```
SELECT @@IDENTITY
```

נקבל את ערך ה-IDENTITY של טבלת Employees: 11

דוגמא 2:
כדי לראות את ערך ה-IDENTITY של טבלת Orders:

```
SELECT IDENT_CURRENT ('Orders')
```

נקבל את ערך ה-IDENTITY של טבלת Orders: 11077

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

INSERT…..SELECT

משמש להוספת רשומות לטבלה
מאפשר להעתיק נתונים מטבלה אחת לטבלה אחרת
תחביר:

```
INSERT INTO <table_name>  (column1, column2…)
SELECT column1, column2…
FROM <table_name>
WHERE <search_condition>
```

כללים עליהם יש לשמור בהוראה זו:

- משפט ה-SELECT אינו יכול לבחור נתונים מהטבלה שאליה מתבצעת ההוספה
- מספר העמודות במשפט INSERT INTO חייב להיות זהה למספר העמודות שהוחזרו על-ידי משפט ה-SELECT
- טיפוסי הנתונים של העמודות במשפט INSERT INTO חייבים להיות זהים לאלה של העמודות המוחזרות על-ידי משפט ה-SELECT

<u>דוגמא:</u> העתקת כל תכולת טבלה א' (Region), לטבלה ב' (Region_Backup), הזהה לה במבנה, בסוגי השדות ובאילוצים: על מנת להעתיק רק את הנתונים עבור איזור 2 מטבלת Region לטבלת Region_Backup נכתוב:

```
INSERT INTO Region_Backup (RegionID, RegionDescription)
SELECT RegionID, RegionDescription FROM Region
WHERE RegionID = 2
```

על מנת להעתיק את כל הנתונים מטבלת Region לטבלת Region_Backup נכתוב:

```
INSERT INTO Region_Backup (RegionID, RegionDescription)
SELECT RegionID, RegionDescription FROM Region
```

UPDATE

משמש לעדכון / שינוי ערכים ברשומות קיימות
תחביר:

```
UPDATE  <table_name>
SET columnname1 = value1, columnname2 = value2…
WHERE <search_condition >
```

העדכון יתבצע עבור כל הרשומות בטבלה הנתונה, העומדות בתנאי החיפוש

<u>דוגמא:</u>
עדכון שם עובד "משה כהן" ל"עדי בן-סימון" בטבלת Employees, כאשר שם המשפחה = כהן והמיקוד = 9777:

```
UPDATE  Employees
SET LastName = 'Ben-Simon', FirstName = 'Adi'
WHERE LastName = 'Cohen' and PostalCode = 9777
```


www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

כאשר משמיטים את הוראת ה-WHERE, מתבצע עדכון של כל אחת מהרשומות בעמודת הרלוונטיות שבטבלה בערך הנקוב

<u>דוגמא</u>: עדכון מספר השלוחה של כל העובדים לשלוחה מספר 555:

```
UPDATE Employees
SET Extension = 555
```

משפט ה-UPDATE יכול גם לעדכן עמודות על סמך תוצאות של ביטוי חשבוני

במקרה כזה יש לזכור:

- טיפוס הנתונים של התוצאה חייב להיות זהה לטיפוס הנתונים של השדה אותו רוצים לעדכן
- גודל הערך חייב להתאים לגודל השדה אותו רוצים לעדכן

<u>דוגמא:</u>

לאור עליות המחירים במשק, הוחלט להעלות את מחיר כל המוצרים בטבלת Products ב-4 ₪:

```
UPDATE Products
SET UnitPrice = UnitPrice + 4
```

DELETE

משמש למחיקת נתונים מטבלה

תחביר:

```
DELETE FROM <table_name>
WHERE <condition>
```

בהתאם להוראת ה-WHERE, SQL יכולה לבצע את הפעולות הבאות:

- מחיקת שורה בודדת
- מחיקת שורות רבות
- מחיקת כל השורות
- אי מחיקה של שורות

משפט DELETE אינו יכול למחוק ערכים בשדות בודדים (לשם כך נשתמש בפקודת UPDATE)

מחיקת רשומות מטבלה אחת יכולה לגרום לבעיות של שלמות הנתונים בטבלאות אחרות

על מנת למחוק רשומה מטבלת האב – יש למחוק תחילה את כל הרשומות התואמות מטבלת הבן

באמצעות DELETE ניתן למחוק רשומות בלבד, אך לא את הטבלה עצמה

השמטת ה-WHERE גורמת למחיקת כל הרשומות הנמצאות באותה הטבלה

<u>דוגמא:</u>

למחיקת כל השורות מטבלת Employees בהן המיקוד = 97777:

```
DELETE FROM Employees
WHERE PostalCode = '97777'
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

8.4. בקרת תנועה (Transactions Control)

תנועות (Transactions) – יחידות עבודה אותן צריך לבצע בסדר לוגי מסוים ולסיימן בהצלחה כקבוצה, או לא לבצען כלל
לכל תנועה יש התחלה וסוף
אם דבר מה משתבש במהלך התנועה – ניתן לבטל את כל יחידת העבודה
אם הכל נראה תקין – ניתן לשמור את כל יחידת העבודה במסד הנתונים
שימושי בעיקר כאשר היישום מבצע חישובים ושינויים בנתונים שבבסיס הנתונים, ואנו רוצים לוודא שישמור את השינויים רק במידה והתהליך כולו הסתיים ללא שגיאות

<u>דוגמא</u>: חישוב חשבון לכל הלקוחות של חברת טלפון, ביום מסוים בחודש
- תהליך זה יכול לקחת זמן ממושך (8 שעות ואף יותר)
- אם, מסיבה כלשהי, התהליך לא יסתיים כמצופה – בסיס הנתונים יכיל נתונים ששונו ונתונים שלא שונו, ואנו לא נוכל להבחין ביניהם או לתקנם

ניתן להתחיל תנועה בצורה מפורשת, על-ידי הוראת:

```
BEGIN TRANSACTION <transaction name>
```

ניתן לסגור תנועה על-ידי הוראות:
- Rollback – ביטול כל התנועה במידה ודבר מה משתבש במהלך ביצועה (=UNDO)
- Commit – שמירת כל התנועה במסד הנתונים, במידה והתנועה הסתיימה בהצלחה

ניתן להגדיר תנועות מקוננות (תנועה בתוך תנועה)
ניתן לשמור את התנועות לשימוש עתידי חוזר

ב-SQL SERVER ברירת המחדל היא שכל הוראת טיפול בנתונים הינה תנועה בפני עצמה, הנפתחת ונסגרת באופן אוטומטי
ב-SQL SERVER:
- הוראת COMMIT מתבצעת אוטומטית בסיום מוצלח של כל הוראת טיפול בנתונים בודדת (לא בתוך Transaction)
- הוראת ROLLBACK מתבצעת אוטומטית במקרה של בעיה במהלך ביצוע טיפול בנתונים

<u>דוגמא</u> לתנועה שלא תישמר:

```
BEGIN TRANSACTION
IF NOT EXISTS
(SELECT * FROM Employees
WHERE PostalCode = '9778'
INSERT INTO Employees
VALUES
('Cohen', 'Moshe', 'Tester', 'Mr.', '1969-11-25', '2009-03-20','Haazmaut 40','Bat
Yam',1,'9778','Israel',03223344,null,null,null,2,null)
 ROLLBACK
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

דוגמא לתנועה שתישמר:

```
BEGIN TRANSACTION
IF NOT EXISTS
(SELECT * FROM Employees
WHERE PostalCode = '9778'
INSERT INTO Employees
VALUES
('Cohen', 'Moshe', 'Tester', 'Mr.', '1969-11-25', '2009-03-20','Haazmaut 40','Bat
Yam',1,'9778','Israel',03223344,null,null,null,2,null)
COMMIT
```

סיכום

ישנם טיפוסי נתונים שונים ב-T-SQL

ישנם אילוצים שונים האוכפים כללים שיחולו על נתונים בעמודת בטבלאות על מנת להבטיח דיוק ושלמות הנתונים

למדנו 3 משפטים לטיפול בנתונים:

- INSERT
- UPDATE
- DELETE

הגדרת תנועות ופקודות COMMIT, ROLLBACK – לבקרת תנועות במסד הנתונים

טלפון: **077-4123211/2** Email: info@iitc.co.il אתר: www.iitc.co.il

חוברת תרגילים

SQL

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

הוראות הוספת Northwind DB ל-databases ב-SQL Server Management Studio

****** עבור win7 והלאה: יש להפעיל את תוכנת ה- SQL Management Studio כמנהל מערכת (כ- Administrator) ע"י סימון ה-Icon של ה-Management Studio -> כפתור ימני בעכבר -> הרץ כמנהל מערכת (או Run As Administrator).**

1. שים את הספרייה של ה-SQL Sample Databases אצלך במחשב (למשל ב: C)
2. פתח את ה-SQL Server Management Studio
3. עמוד עם העכבר על תיקיית Databases ולחץ כפתור ימני -> Attach
4. בחלון שיפתח לחץ Add ומצא את התיקייה של ה- Sample Databases על המחשב שלך
5. סמן את Northwind.MDF (כמדומני) תחת הסיפרייה הנ"ל ולחץ OK ושוב OK
6. בדוק שאתה רואה את ה-Northwind database תחת תיקיית ה-Databases

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

1. שאילתות - תרגיל

ענה על השאלות הבאות:

1. האם ההוראות הבאות מחזירות פלט שונה?
   SELECT * FROM CHECKS;
   Select * from checks;

2. אף אחת מהשאילתות הבאות לא תפעל. מדוע?
   a. Select *
   b. Select amount name payee FROM checks;

3. אילו הוראות, מתוך ההוראות הבאות, יפעלו?
   a. Select *
      From checks;
   b. Select * from checks
   c. Select * from checks
      /

4. כתוב שאילתה אשר מחזירה רק את השם הפרטי והתפקיד של העובדים מתוך טבלת EMPLOYEES בבסיס הנתונים Northwind (בתוכנת SQL Server).
5. שכתב את השאילתה מהסעיף הקודם, כך שהתפקיד יופיע בעמודה הראשונה של תוצאות השאילתה.
6. כתוב שאילתה המחזירה את כל התפקידים מטבלת EMPLOYEES, כך שכל שם תפקיד יופיע רק פעם אחת בתוצאות השאילתה.
7. הצג את כל המידע מטבלת Orders
8. הצג מטבלת Employees את העמודות הבאות:
   FirstName, HireDate, Region, Country

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

2. אופרטורים – תרגיל

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

1. כתוב שאילתה אשר מחזירה את כל האנשים ששמם הפרטי מתחיל באות M, מתוך טבלת EMPLOYEES.
2. כתוב שאילתה המחזירה את כל העובדים המתגוררים בעיר LONDON, אשר שמם הפרטי הוא Robert, מתוך טבלת Employees.
3. כתוב שאילתה אשר מחזירה את שם המוצר, מחיר מחירון (UNITPRICE), כמות במלאי (UNITSINSTOCK) ושארית החלוקה של הכמות במלאי במחיר המחירון, כאשר שם העמודה המחושבת יהיה SHEERIT, מתוך טבלת PRODUCTS.
4. באיזה קיצור ניתן להשתמש במקום:

   WHERE a > = 10 AND a < = 30

5. הצג מטבלת Products את העמודות הבאות: ProductID (שתיקרא: ProdID), ProductName (שתיקרא OrodName), UnitPrice (שתיקרא UntPrc)
6. הצג מטבלת Customers שתי עמודות: קוד הלקוח בעמודה אחת, הכתובת והעיר משורשרות יחד עם רווח ביניהן – בעמודה השניה. תן שם לעמודה השנייה: Full Address
7. הצג את כל הנתונים עבור כל הלקוחות מאנגליה (Country: UK)
8. הצג את המוצרים הפעילים (Discontinued = 0) שנגמרו במלאי (unitsinstock = 0) מטבלת Products

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

9. נתונה טבלת FRIENDS הבאה:

LASTNAME	FIRSTNAME	AREACODE	PHONE	STATE
BUNDY	AL	100	555-1111	IL
MEZA	AL	200	555-2222	US
MERRICK	BUD	300	555-6666	UK
MAST	JD	381	555-6767	NZ
BULHER	FERRIS	345	555-3223	IL
PARKINS	ALTON	911	555-3116	US
BOSS	SIR	204	555-2345	FR

a. מה תחזיר השאילתה הבאה?

```
SELECT FIRSTNAME
FROM FRIENDS
WHERE FIRSTNAME = 'AL'
AND LASTNAME = 'BULHER';
```

b. מה תחזיר השאילתה הבאה?

```
SELECT FIRSTNAME
FROM FRIENDS
WHERE FIRSTNAME = 'AL'
OR  LASTNAME = 'BULHER';
```

10. השתמש בטבלת FRIENDS, וכתוב שאילתה אשר מחזירה את התוצאה הבאה:

NAME	ST
AL FROM	IL

11. השתמש בטבלת FRIENDS, וכתוב שאילתה אשר מחזירה את התוצאה הבאה:

NAME	PHONE
MERRICK, BUD	300-555-666
MAST, JD	381-555-6767
BULHER, FERRIS	345-555-3223

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

3. פונקציות – תרגיל

ענו על השאלות הבאות:

1. אילו פונקציות מוכרות גם בשם פונקציות קבוצתיות (Group functions)?
2. שדה LASTNAEME הינו מסוג text. האם השאילתה הבאה תפעל?

SELECT COUNT (LASTNAME) FROM CHARACTERS

3. שדה LASTNAEME הינו מסוג text. האם השאילתה הבאה תפעל?

SELECT SUM (LASTNAME) FROM CHARACTERS

4. מה פירוש התשובה 6 לשאילתה הבאה:

SELECT COUNT (*) FROM TEAMSTATS

5. ציין את הגורמים הפנימיים של ערך DATETIME
6. מה עשוי להיות גורם חיוני הנוגע לייצוג ערכי תאריך ושעה, והשוואה ביניהם, כאשר מדובר בארגון בינלאומי?
7. מדוע לא רצוי לאחסן במסד הנתונים את גילו של אדם?

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

8. השתמש בטבלת ORDER DETAILS וכתוב שאילתה המציגה את מחיר היחידה (UNITPRICE) כשדה Integer.
9. השתמש בטבלת EMPLOYEES וכתוב שאילתה המציגה את השורש הריבועי של שדה EmployeeID.
10. השתמש בטבלת ORDERS וכתוב שאילתה המציגה את הפרש הזמן בימים בין ביצוע הזמנה (orderdate) למשלוח (shippeddate). קרא לעמודת הפרש הזמן בשם: DURATION.
11. כתוב שאילתה הממירה את המחרוזת '2009-07-27' לסוג נתונים DATETIME.
12. הצג מטבלת Employees את שמו הפרטי של העובד באותיות קטנות, שם משפחתו באותיות גדולות, עבור העובדים שמספר העובד שלהם בין 3 ל-5.
13. הצג מטבלת Categories את שם הקטגוריה, תיאור הקטגוריה, ואת מיקום התו 'i' בתיאור הקטגוריה 4 תווים מתחילת המילה.
14. הצג מטבלת Products את מספר המוצר, שם המוצר, ובעמודה נוספת שוב את שם המוצר כאשר כל תו 'e' יוחלף בתו '-'
15. הצג מטבלת Employees את שם המשפחה הקטן ביותר מבחינה אלפאבתית והשם הפרטי הגדול ביותר מבחינה אלפאבתית
16. הצג מטבלת Employees את מספר הרשומות שיש בטבלה
17. הצג מטבלת Employees את מספר הרשומות שיש בעמודת region (לא כולל NULL).
18. הצג מטבלת Products את ממוצע המחיר ליחידה

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

19. הצג את מספר הלקוחות השונים הקיימים בטבלת Orders, תן שם מתאים לעמודה. שים לב, יתכן שלקוח מסוים ביצע יותר מהזמנה אחת.
20. הצג את הנתונים של 10 הלקוחות הראשונים מטבלת Customers
21. הצג את הנתונים של 10% מהלקוחות הראשונים מטבלת Customers

טלפון: **077-4123211/2** Email: info@iitc.co.il אתר: www.iitc.co.il

4. הוראות בשאילתות GROUP BY & ORDERBY – תרגיל

1. מהו תפקידה של ההוראה GROUP BY?
2. האם משפט ה-SELECT הבא יפעל?

```
SELECT NAME, AVG (SALARY), DEPARTMENT
FROM PAY_TBL
WHERE DEPARTMENT =  'ACCOUNTING'
ORDER BY NAME
GROUP BY DEPARTMENT, SALARY;
```

3. כאשר משתמשים בהוראת HAVING, האם צריך תמיד להשתמש גם ב-GROUP BY?
4. האם ניתן להפעיל את ORDER BY גם על עמודה שאינה חלק מהעמודות המוזכרות במשפט ה-SELECT, או ה- Group By, במקרים הבאים:
   a. כאשר השאילתה כוללת הוראת Group By?
   b. כאשר השאילתה אינה כוללת הוראת Group By?

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

5. השתמש בטבלת PRODUCTS וכתוב שאילתה המציגה את שם המוצר ומספרו (PRODUCT ID), ממוינים לפי שם המוצר מהסוף להתחלה.
6. השתמש בטבלת PRODUCTS וכתוב שאילתה המציגה את סה"כ המחיר (UNITPRICE) עבור כל קטגוריה (CATEGORYID). השתמש בהוראת GROUP BY.
7. השתמש בטבלת PRODUCTS וכתוב שאילתה המציגה את המחיר המקסימלי והמחיר המינימלי (UNITPRICE) עבור כל קטגוריה (CATEGORYID).
8. השתמש בטבלת CUSTOMERS ומצא את כל הלקוחות (CompanyName) הגרים בלונדון או בעיר המתחילה באותיות 'MA'. הרשימה צריכה להיות ממוינת לפי שם החברה
9. השתמש בטבלת CUSTOMERS ומצא את כל הלקוחות (CompanyName) שה-REGION שלהם אינו NULL. הרשימה צריכה להיות ממוינת לפי ה-Region
10. השתמש בטבלת PRODUCTS וכתוב שאילתה שתראה עבור כל ספק (SupplierID):
    1. סה"כ מחיר היחידות
    2. מספר UnitsInStock

    עבור סה"כ מחיר הגדול מ-50 ועבור מספר קטגוריה גדול מ 2. התוצאות יוצגו לפי מספר היחידות במלאי בסדר יורד. השתמש ב-Group By, Having, Order By
11. הצג את הנתונים של 10% מהלקוחות **האחרונים** מטבלת Customers

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

5. צירוף טבלאות JOIN – תרגיל

1. כמה שורות ייצור צירוף (ע"י פסיקים) של שתי הטבלאות ללא התניה בהוראה WHERE, אם בטבלה אחת יש 50,000 שורות ובשנייה 100?
2. איזה סוג צירוף מופיע בהוראת ה-SELECT הבאה:

```
SELECT E.NAME, E.EMPLOYEE_ID, EP.SALARY
FROM EMPLOYEE_TBL E JOIN  EMPLOYEE_PAY_TBL EP
ON  E.EMPLOYEE_ID = EP.EMPLOYEE_ID;
```

3. האם משפטי ה-SELECT הבאים יפעלו?

a.
```
SELECT NAME, EMPLOYEE_ID, SALARY
FROM EMPLOYEE_TBL  E JOIN  EMPLOYEE_APY_TBL  EP,
ON EMPLOYEE_ID = EMPLOYEE_ID
AND NAME LIKE '%MITH';
```

b.
```
SELECT E.NAME, E.EMPLOYEE_ID, EP.SALARY
FROM EMPLOYEE_TBL E, EMPLOYEE_PAY_TBL EP
WHERE E.NAME LIKE '%MITH';
```

c.
```
SELECT E.NAME, E.EMPLOYEE_ID, EP.SALARY
FROM EMPLOYEE_TBL E  JOIN  EMPLOYEE_PAY_TBL EP
ON  E.EMPLOYEE_ID = EP.EMPLOYEE_ID
AND E.NAME LIKE '%MITH';
```

d.
```
SELECT E.NAME, E.EMPLOYEE_ID, EP.SALARY
FROM EMPLOYEE_TBL E  JOIN  EMPLOYEE_PAY_TBL EP
ON  E.EMPLOYEE_ID = EP.EMPLOYEE_ID
WHERE  E.NAME LIKE '%MITH';
```

e.
```
SELECT E.NAME, E.EMPLOYEE_ID, EP.SALARY
FROM EMPLOYEE_TBL E, EMPLOYEE_PAY_TBL EP
WHERE E.EMPLOYEE_ID = EP.EMPLOYEE_ID
AND E.NAME LIKE '%MITH';
```

4. בעת צירוף טבלאות, האם אתה מוגבל לצירוף של עמודה אחת, או ניתן להשתמש בעמודות רבות?

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

5. הצג את שם המוצר מתוך טבלת Products ואת שם הקטגוריה שלו מתוך טבלת Categories
6. הצג את שם המוצר ומחירו מתוך טבלת Products, ואת שם הקטגוריה שלו מתוך טבלת Categories, עבור המוצרים שמחירם גבוה מ-50.
7. השתמש בשלושת הטבלאות הבאות:
   a. Employees
   b. EmployeeTerritories
   c. Territories

   כתוב שאילתה אחת המחזירה את הנתונים הבאים:
   עבור כל עובד – שם פרטי, שם משפחה, מספר הטריטוריה שלו ותיאור של כל טריטוריה.

8. השתמש בשתי הטבלאות הבאות:
   a. Orders
   b. Customers

   כתוב שאילתת **Left Join** המחזירה את הנתונים הבאים:
   מספר לקוח, מספרי ההזמנה, תאריך משלוח, איש קשר, עיר וטלפון עבור הזמנות שמספרן גדול מ-10700. התוצאה צריכה להיות מסודרת בסדר יורד לפי מספר לקוח (CustomerID).

9. כתוב שאילתה המציגה את קוד הלקוח, ועירו מטבלת CUSTOMERS, ומחיר המוצרים (unitprice) מטבלת [ORDER DETAILES], שהזמינו לקוחות מלונדון, ומחיר אותם מוצרים בתוספת של 5 ש"ח. השתמש בטבלה מקשרת.
10. הצג את מספר המוצר, מחירו ומספר הספק מתוך טבלת Products ואת שם הקטגוריה שלו מתוך טבלת Categories, עבור המוצרים שבשמם הקטגוריה שלהם ישנה האות a.
11. הצג את שם המוצר מטבלת Products, את תיאור הקטגוריה מתוך טבלת Categories (Description), ואת עיר הספק מתוך טבלת Suppliers, כאשר עיר הספק היא London או Tokyo
12. הצג את שם החברה של הלקוח מתוך טבלת Customers ואת מספר ההזמנה שלו מתוך טבלת Orders, עבור כל הלקוחות (גם עבור לקוחות שלא ביצעו הזמנות).
13. הצג את שמו הפרטי ושם משפחתו של העובד, ואת שם המנהל שלו (שמו הפרטי ושם משפחתו), מתוך טבלת Employees. בסס את הקשר בין מספר המנהל (ReportsTo) ובין מספר העובד (EmployeeID). הצג גם עובדים ללא מנהלים.
14. הצג את כל המוצרים, מחיריהם, הספק שלהם, ושם איש הקשר (בחברת הספק). הנתונים נמצאים בשתי טבלאות: Suppliers, Products

טלפון: **077-4123211/2** Email: info@iitc.co.il אתר: www.iitc.co.il

6. שאילתות משנה – תרגיל

1. האם המשפטים הבאים נכונים או לא נכונים?

a. כל פונקציות הצבירה הבאות: MAX, MIN, COUNT,SUM , ו-AVG מחזירות ערכים רבים

b. המספר המרבי של השאילתות המקוננות הוא 2

2. האם שאילתות המשנה הבאות, אשר משתמשות בטבלאות PART,ORDERS יפעלו?

טבלת ORDERS:

ORDERDON	NAME	PARTNUM	QUANTITY	REMARKS
15-MAY-96	TRUE WHEEL	23	6	PAID
19-MAY-96	TRUE WHEEL	76	3	PAID
2-SEP-96	TRUE WHEEL	10	1	PAID
30-JUN-96	BIKE SPEC	54	10	PAID
30-MAY-96	BIKE SPEC	10	2	PAID
30-MAY-96	BIKE SPEC	23	8	PAID
17-JAN-96	BIKE SPEC	76	11	PAID
17-JAN-96	LE SHOPPE	76	5	PAID
1-JUN-96	LE SHOPPE	10	3	PAID
1-JUN-96	AAA BIKE	10	1	PAID
1-JUN-96	AAA BIKE	76	4	PAID
1-JUN-96	AAA BIKE	46	14	PAID
11-JUL-96	JACKS BIKE	76	14	PAID

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

טבלת PART:

PARTNUM	DESCRIPTION	PRICE
54	PEDALS	54.25
42	SEATS	24.50
46	TIRES	15.25
23	MOUNTAIN BIKE	350.45
76	ROAD BIKE	530.00
10	TANDEM	1200.00

א.
```
SELECT * FROM ORDERS
WHERE PARTNUM =
SELECT PARTNUM FROM PART
WHERE DESCRIPTION = 'TRUE WHEEL';
```

ב.
```
SELECT PARTNUM
FROM ORDERS
WHERE PARTNUM = (SELECT * FROM PART
                 WHERE DESCRIPTION = 'LE SHOPPE');
```

ג.
```
SELECT NAME, PARTNUM
FROM ORDERS
WHERE EXISTS (SELECT * FROM ORDERS
              WHERE NAME = 'TRUE WHEEL');
```


www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

3. הצג מטבלת Products את שמות המוצרים אשר מחירם נמוך מהמחיר של מוצר מספר 4
4. הצג מטבלת Products את שמות המוצרים ומחירם, עבור המוצרים שמחירם גבוה ממוצר ששמו CHAI
5. הצג מטבלת Employees את שמות העובדים ותאריך גיוסם, עבור העובדים שגויסו לאחר עובד שמספרו 6
6. הצג מטבלת Products את מספר המוצר, שם המוצר ומחיר יחידה, עבור מוצרים שמחירם גבוה מהמחיר הממוצע ליחידה.
7. הצג את מספר המוצרים שכמותם במלאי (unitsInstock) נמוך מהכמות הממוצעת במלאי
8. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם שווה לפחות לאחד המוצרים בקטגוריה מספר 7
9. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם אינו שווה לאף אחד מהמוצרים בקטגוריה מספר 7
10. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם יקר מלפחות אחד המוצרים בקטגוריה מספר 7
11. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם זול מלפחות אחד מהמוצרים בקטגוריה מספר 7
12. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם יקר יותר מכל המוצרים בקטגוריה מספר 7
13. הצג מטבלת Products את שם המוצר ומחירו, עבור המוצרים שמחירם זול יותר מכל המוצרים בקטגוריה מספר 7
14. כתוב שאילתה המציגה את הספק עם המוצר היקר ביותר
15. כתוב שאילתה המציגה את הספק עם הכי הרבה הזמנות
16. הצג מטבלת Products את מספר המוצר (ProductID), שם המוצר ומחירו, עבור מוצרים שעולים יותר ממוצר "Alice Mutton".
17. כתוב שאילתה המציגה את מספר ההזמנות של העובד המבוגר ביותר.
18. מהו המחיר הממוצע להזמנה ללקוחות הגרים באזור (Region): WA
19. השתמש בשתי הטבלאות הבאות:
    a. EmployeeTerritories
    b. Employees

    כתוב שאילתה המציגה את מספר העובד ושמו עבור עובד מספר 7 (שם פרטי ושם משפחה) ומספר הטריטוריות שלו. השתמש בתת שאילתה
20. השתמש בשתי הטבלאות:
    a. EmployeeTerritories
    b. Employees

    כתוב שאילתה המציגה את מספר העובד ומספר הטריטוריות שלו עבור כל העובדים. השתמש בתת שאילתה.
21. נתונות שתי טבלאות: EMPLOYEES ו- CUSTOMERS, שכל אחת מהן מכילה עמודה בשם: REGION. כיצד תוכל להציג את פרטי העובדים המשתייכים לאזורים בהם יש

טלפון: 077-4123211/2 Email: info@iitc.co.il אתר: www.iitc.co.il

לקוחות (כלומר, אנו לא מעוניינים להציג עובדים המשתייכים לאזורים שאין בהם לקוחות)? כתוב שאילתה זו.

22. הצג את שם החברה, שם איש הקשר ושם המוצר עבור כל המוצרים שספק Leka Trading מספק לנו. הנתונים נמצאים בשתי טבלאות: Products, Suppliers. השתמש בתת שאילתה.

23. הצג מטבלת Products את שמות כל המוצרים אשר שם הקטגוריה שלהם הוא Beverages או Condiments וגם אזור (Region) הספק אינו ידוע.

24. כתוב שאילתה המציגה את כל העמודות מטבלת TERRITORIES עבור RegionID=1 ו- RegionID=2. השתמש באופרטור Union.

25. הצג את שם העיר, שם החברה ושם איש הקשר עבור כל הלקוחות (מטבלת Customers) וכנ"ל עבור כל הספקים (מטבלת Suppliers), ללא כפילויות. השתמש באופרטור Union.

26. כתוב שאילתה המציגה את פרטי המוצרים שקיימות עבורם הזמנות של לקוחות שגרים בצרפת או במקסיקו. השתמש בטבלאות הבאות:

    a. Customers
    b. orders
    c. [Order Details]
    d. products

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

7. טיפול בנתונים – תרגיל

1. האם אילוצים יכולים ליצור בעיות בייבוא נתונים למסד הנתונים?
2. האם חלה מגבלה כלשהי על מספר העמודות שאותם ניתן להגדיר כייחודיות?
3. מה מבצעים אילוצי בדיקה?
4. בעת הוספת נתונים למסד נתונים, איזו רשומה יש להוסיף קודם – את האב או את הבן?
5. בעת מחיקת נתונים ממסד הנתונים, איזו רשומה יש למחוק קודם - את האב או את הבן?
6. מה שגוי במשפטים הבאים?
   a. DELETE EMPLOYEES;
   b. INSERT INTO EMPLOYEES
      SET SELECT * FROM TABLE_2
   c. UPDATE TERRITORIES (01288, 'Tel Aviv', 4);

7. מה יקרה אם תפעיל את ההוראות הבאות?
   a. DELETE * FROM TERRITORIES;
   b. DELETE FROM TERRITORIES;
   c. UPDATE SUPPLIERS
      SET PHONE = 4444-3322
      SET FAX = 3322-1155

8. האם המשפטים הבאים יפעלו?
   a. INSERT INTO SUPPLIERS
      SET PHONE = 123-4567
      WHERE SUPPLIERID = 23;
   b. UPDATE SUPPLIERS
      SET PHONE = 123-4567
      WHERE SUPPLIERID = 23;

ענה על השאלות הבאות בעזרת בבסיס הנתונים Northwind (בתוכנת SQL Server):

9. הוסף רשומה לטבלת TERRITORIES עם הנתונים הבאים:

   TerritoryID: 23234
   TerritoryDescription: TEL AVIV
   RegionID: 5

   a. מדוע השאילתה לא הצליחה?
   b. רשום את הודעת השגיאה שהוחזרה ותקן את השאילתה בהתאם.

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

10. הכנס רשומה חדשה עבורך (עם הפרטים שלך) לטבלת Employees, ללא ציון שם העמודות בהוראת ה-INSERT
11. הכנס רשומה נוספת לטבלת Employees עבור עובד שיום הולדתו גדול מהתאריך הנוכחי. האם הצלחת? מה הסיבה?
12. הכנס רשומה נוספת לטבלת Employees, עם ציון העמודות (לפי הסדר שלמטה), עבור העובד:
    a. שם פרטי: Larry
    b. שם משפחה: King
    c. City: Los Angeles
    d. Title: announcer
    e. TitleOfCourtesy: NULL

13. העובד Larry King עבר דירה. יש לעדכן את השדות הבאים בטבלת Employees עבור עובד זה:
    a. Address: 56 Hatikva Str.
    b. City: Ramat Hasharon
    c. Region: TA
    d. PostalCode: 34533
    e. Country: IL

14. לאור הקיצוצים בחברה הוחלט לפטר את Larry King. יש למחוק את הרשומה מהטבלה.

15. בצע את שאלות 3,4,5 שוב, הפעם תוך שימוש בטרנזאקציה:
    a. פעם אחת כאשר בסופה לא ישמרו הנתונים (roll Back)
    b. פעם שניה כאשר בסופה ישמרו הנתונים (commit)

16. מס חדש נחקק בלונדון, על כן כל המחירים עבור תושבי לונדון מתייקרים ב 5 דולר:
    a. בצע פעולת עדכון UPDATE רק לרשומות הרלוונטיות בטבלת orders (שים לב לא לעדכן את כל ההזמנות)
    b. על מנת לגלות אילו מחירים יש לעדכן, יש צורך לעבוד עם 3 טבלאות: Customers, Orders, [Order Details]
    c. ניתן לפתור באמצעות SUB QUEIRY
17. 2 עובדים חדשים התקבלו לחברה. הכנס רשומה חדשה עבור כל אחד מהם. כתוב שאילה המציגה את פרטי העובדים **שאין** להם הזמנות.

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

8. פקודות DDL ו-DML - תרגיל

1. ב Northwind DB - צור טבלה בשם US_CUSTOMERS בעלת מבנה זהה לטבלת Customers הקיימת. השתמש בפקודת CREATE
2. אכלס את הטבלה שיצרת עם הנתונים מטבלת Customers הקיימת. כלול רק את הרשומות בהם שדה Country מכיל את הערך USA
3. הוסף עמודה בשם Customer_Type לטבלה שיצרת, העמודה תכיל נתונים של תו יחיד
4. עדכן את הערך בעמודת Customer_Type כך שיכיל את הערך "W" עבור מדינות בחוף המערבי של ארה"ב (California, Oregon, or Washington)
5. הוסף עמודה בשם Company_Start_Date עם נתונים מסוג תאריך, היכולה לקבל NULL
6. הוסף אילוץ Check לשדה Company_Start_Date הבודק שהתאריך המוכנס לטבלה קטן מהיום
7. הוסף נתונים לעמודת Company_Start_Date עבור כל הלקוחות בטבלה, ובדוק האם ניתן להכניס תאריך שהינו היום או גדול מהיום.
8. מחק את עמודת הפקס מטבלת US_CUSTOMERS
9. מחק את אילוץ PK מטבלת US_CUSTOMERS
10. הוסף את אילוץ PK בחזרה לטבלת US_CUSTOMERS (לעמודת CustomerID)
11. נקה את טבלת US_CUSTOMERS מכל הנתונים
12. מחק את טבלת US_CUSTOMERS מה-DB

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון

9. SQL Server Management Studio Express

9.1 יצירת DB חדש:

סמן את הספרייה של ה-Databases -> כפתור ימני -> New DB

9.2 התחברות ל-DB החדש:

סמן את ה-DB הרצוי -> כפתור ימני -> New Query

9.3 הגדרת טבלה חדשה:

1. עמוד על ספריית Tables -> כפתור ימני -> New Table
2. מגדירים כל שדה + סוג הנתונים + האם ניתן להכניס בו NULL
3. כאשר עושים Save הוא שואל לשם הטבלה.

9.4 סימון שדה כ-Primary Key:

סמן את השדה הרצוי -> לחץ על כפתור PK (מפתח)

9.5 הגדרת שדה כ-Foreign Key:

1. סמן את ספריית Keys תחת שם הטבלה -> כפתור ימני -> New Foreign Key
2. בחלון שיפתח לחץ על הכפתור בשורה: Tables & Columns Specification
3. הגדר את טבלת ה-PK ושדה ה-PK, ואת שדה ה-FK בטבלה הנוכחית (השדה המקשר בין הטבלאות)
4. סגור את החלון ושמור את השינויים
5. לחץ על Refresh כדי לראות את השינויים בספריית Keys של הטבלה הנוכחית

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

10. נספח 1: סוגי נתונים – Data Types

All the data values of a column must be of the same data type.

Datatype	Min	Max	Type	Notes
Bigint	-2^63	2^63-1	Exact numeric	
Int	-2,147,483,648	2,147,483,647	Exact numeric	
Smallint	-32,768	32,767	Exact numeric	
Tinyint	0	255	Exact numeric	
Bit	0	1	Exact numeric	
Decimal	-10^38+1	10^38−1	Exact numeric	Decimal and numeric data type is exactly the same. Precision is the total number of digits. Scale is the number of decimals. For booth the minimum is 1 and the maximum is 38.
Numeric	no			
Money	-2^63 / 10000	2^63-1 / 10000	Exact numeric	
Smallmoney	-214,748.3648	214,748.3647	Exact numeric	
Float	-1.79E + 308	1.79E + 308	Approximate numerics	Precision is specified from 1 to 53.
Real	-3.40E + 38	3.40E + 38	Approximate numerics	Precision is fixed to 7.
Datetime	1753-01-01 00:00:00.000	9999-12-31 23:59:59.997	Date and time	If you are running SQL Server 2008 and need milliseconds precision, use datetime2(3) instead to save 1 byte.
Smalldatetime	1900-01-01 00:00	2079-06-06 23:59	Date and time	
Date	0001-01-01	9999-12-31	Date and time	
Time	00:00:00.0000000	23:59:59.9999999	Date and time	Specifying the precision is possible. TIME(3) will have milliseconds precision. TIME(7) is the highest and the default precision. Casting values to a lower precision will round the value.
Datetime2	0001-01-01 00:00:00.0000000	9999-12-31 23:59:59.9999999	Date and time	Combines the date datatype and the time datatype into one. The precision logic is the same as for the time datatype.
Datetimeoffset	0001-01-01 00:00:00.0000000 -14:00	9999-12-31 23:59:59.9999999 +14:00	Date and time	Is a datetime2 datatype with the UTC offset appended.
Char	0 chars	8000 chars	Character string	Fixed width
Varchar	0 chars	8000 chars	Character string	Variable width
Varchar(max)	0 chars	2^31 chars	Character string	Variable width
Text	0 chars	2,147,483,647 chars	Character string	Variable width
Nchar	0 chars	4000 chars	Unicode character string	Fixed width

www.iitc.co.il :אתר info@iitc.co.il :Email 077-4123211/2 :טלפון

Nvarchar	0 chars	4000 chars	Unicode character string	Variable width
Nvarchar(max)	0 chars	2^30 chars	Unicode character string	Variable width
Ntext	0 chars	1,073,741,823 chars	Unicode character string	Variable width
Binary	0 bytes	8000 bytes	Binary string	Fixed width
Varbinary	0 bytes	8000 bytes	Binary string	Variable width
Varbinary(max)	0 bytes	2^31 bytes	Binary string	Variable width
Image	0 bytes	2,147,483,647 bytes	Binary string	Variable width
Sql_variant			Other	Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
Timestamp			Other	Stores a database-wide unique number that gets updated every time a row gets updated.
Uniqueidentifier			Other	Stores a globally unique identifier (GUID).
Xml			Other	Stores XML data. You can store xml instances in a column or a variable.
Cursor			Other	A reference to a cursor.
Table			Other	Stores a result set for later processing.

www.iitc.co.il :אתר info@iitc.co.il :Email **077-4123211/2** :טלפון