דף נוסחאות בסיסי נתונים

מבנה שאילתא:

Select *, Column,Column

From TableName

Where condition

Group by Column Name

Having Condition on Group

Order by Sort Asc Or Desc

שירשור עמודות של טקסט

Select FirstName + LastName

From Employees

Select FirstName + ' ' +LastName

From Employees

אליאס שם זר

את השם החדש לעמודה ניתן לתת בכמה דרכים שלפניכם.

Select FirstName + LastName as **FullName**

From Employees

Select FirstName + LastName **as [Full Name]**

From Employees

Select FirstName + LastName as **"FullName"**

From Employees

אופרטורים פשוטים של WHERE

= שווה

!= לא שווה

< > שונה

< גדול

> קטן

<! לא גדול

>! לא קטן

=< גדול שווה

=> קטן שווה

משמשים בעיקר לתנאים של WHERE

דוגמאות:

Select * from Employees where employeeid > 34

Select * from Employees where FirstName='Alex' and LastName = 'Stive'

אופרטורים מורכבים

Between - משמש למציאת תחום ערכים בין מספר טקסט או תאריך כאשר הקצוות נמצאים בתוך התחום משני קצותיו ונכללים בתחום.

Select * from Employees where Employeeid between 1 and 45

Select * from Employees where Birthdate between '1990-02-02' and '1996-12-01'

IN משמש כדי לכלול הרבה ערכים ברשימה שתחליף את OR. IN(1,2,3)

פירושו שיש לכלול את הערכים 1 או 2 או 3

Select * from Employees where City IN('London','Takoma')

Select * from Employees where Employeeid in (1,4,5)

LIKE האופרטור הזה תפקידו לחפש אותיות או תת מחרוזת בתוך טקסט נתון.

אפשר לחפש הופעה של תת מחרוזת בתחילת המחרוזת, בסוף המחרוזת או באמצעה

באמצעות סימון עם אחוזים.

Select * from Employees where firstName like '%a%'

בדוגמה הזו מצאנו את כל מי שיש לו את האות a במקום כלשהו במחרוזת.

Select * from Employees where FirstName like 'a%'

בדוגמה הזו מצאנו את כל מי שהשם שלו מתחיל ב a

Select * from employees where FirstName like '%a'

אופרטור **IS NULL** משמש למצוא עמודות שיש או אין בהם NULL.

Select * from Employees where City is null

Select * from Employeeswhere City is not null

האופרטור **NOT** משמש לשלילה של אופרטור אחר.

לדוגמה: select * from Employees where Employeeid not Between 1 and 5

Select * from Employees where Employeeid not IN(1,2,5)

פקודת **DISTINCT** משמשת לצמצם הופעות של ערכים בתוצאה. למשל, אם יש הרבה פעמים את העיר לונדון בעמודת ערים נוכל לקבל כל ערך פעם אחת

Select Distinct City From Employees

Select Distinct Country,City from employees

<u>מיון של ערכים ORDER BY</u>

מבצעים מיון של ערכים לפי עמודה אחת או יותר בשורה האחרונה של השאילתה.

מיון יורד מסומן בעזרת **DESC**

מיון עולה מסומן באמצעות **ASC** או שלא מסומן והוא ברירת מחדל.

Select * from Employees order by Employeeid asc

Select * from Employees order by Employeeid desc

TOP - הבאת רשומות מראש הרשימה על פי בחירה שלנו. זאת אומרת מהתוצאות של השאילתא אנחנו מבקשים מספר מסויים של רשומות מראש הרשימה.

Select top 10 * from Employees

Select top 3 FirstName,LastName from Employees

Select top 3 * from Products order by UnitPrice Desc

בשאילתה האחרונה בעצם הצגנו את שלושת המוצרים היקרים ביותר.

Top N Percent - הבא רשומות עליונות באחוז יחסי למספר הרשומות.

לדוגמא אם יש 1000 רשומות ואנו נבקש 10 אחוז נקבל 100 רשומות עליונות.

Select top 10 percent from Employees

ISNULL הפונקציה הזו מסוגלת להחליף ערכים בשדות שיש בהם NULL ולשתול ערך אחר.

לדוגמא אנחנו מחפשים שם עיר ומי שאין לו שם עיר נשתול שם מחרוזת not Found

Select IsNull(City,'Not Found') from Employees

Select isnull(UnitPrice,23) from products

CAST - פונקציות המרה מטיפוס נתונים אחד לטיפוס נתונים שני. לדוגמא יש לנו את הגיל ואנו רוצים להדפיס אותו ביחד עם השם, לכן נמיר את המספר לטקסט.

Select Cast(Age as char(2)) + firstName From Employees

Select Cast('1999-02-03' as Datetime) from Employees

Select Cast(BirthDate as Varchar(20)) from Employees

<u>פונקציית המרה CONVERT</u>

הפונקציה הזו דומה מאוד ל CAST רק שהיא יכולה לקבל סטייל או פורמט הדפסה עבור תאריכים כשהפורמט הוא בין 100 ל 130 על פי לוח שנמצא באינטרנט.

Select Convert(varchar(20), BirthDate,113) from Employees

Select Convert(Varchar(3), Age) from Employees

פעולות אריתמטיות על עמודות נעשות גם ב SELECT וגם ב WHERE

Select UnitPrice +100 from products

Select UnitPrice / Quantity from OrderDetails

חיבור +

חיסור –

כפל *

חילוק /

פונקציות קבוצתיות - נועדו לסכם נתונים של עמודות עבור טבלה שלמה או קבוצה גדולה.

AVG - פונקציה שמביאה ממוצע מספרי של עמודה.

Select Avg(UnitPrice) from products

SUM - פונקציה שסוכמת מספרים של עמודה

Select Sum(UnitPrice) from Products

COUNT – פונקציה שסופרת את מספר הרשומות. בשיטה אחת סופרים עם Count(*)

ואז סופרים את כל הרשומות, ו בשיטת ספירת עמודה הפונקציה סופרת הכל חוץ מ NULL

Select count(*) from Products סופרת את כל הרשומות.

Select count(Region) from employees סופרת רק היכן שיש ערך שונה מ NULL

MAX – מביא את הערך המקסימלי מהעמודה גם במספר וגם בטקסט.

Select max(UnitPrice) from products

MIN – מביאה את הערך המינימלי מהעמודה גם מספר וגם טקסט.

Select Min(UnitPrice) from products

<u>פונקציות סקלריות</u>

מחרוזות:

ASCII – מקבלת תו ומחזירה מספר אסקי

Select Ascii('A')

התוצאה: 65

CHAR – מקבלת מספר אסקי ומחזירה תו

Select Char(65)

התוצאה: ('A')

SUBSTRING– פונקציה שחותכת תת מחרוזת מתוך מחרוזת. מקבלת מחרוזת מקורית, מיקום התחלה, אורך תת מחרוזת.

Select Substring('abcd',3,2)

מחזירה :' cd'

CHARINDEX – פונקציה שמחזירה מיקום של תו במחרוזת. מקבלת תו לחיפוש ומחרוזת, יכולה לקבל גם מיקום לתחילת חיפוש ומחזירה את מיקום התו במחרוזת.

select charindex('a','abdc',1)

הפונקציה מחזירה מספר 1

RIGHT – הפונקציה מחזירה תת מחרוזת מצד ימין של המחרוזת המקורית לפי מספר תוים שמתבקש.

select right('abcd',2)

הפונקציה מחזירה 'cd'

LEFT – הפונקציה מחזירה תת מחרוזת מצד שמאל של המחרוזת המקורית לפי מספר תוים שמתבקש.

select left('abcd',2)

הפונקציה מחזירה 'ab'

RTRIM - הפונקציה מנקה רווחים מימין למחרוזת

LTRIM – הפונקציה מנקה רווחים משמאל למחרוזת

select rtrim('abcd ')

מחזירה 'abcd'

select ltrim(‘abcd ')

מחזירה 'abcd'

LEN - מקבלת מחרוזת כפרמטר ומחזירה את האורך שלה במספר תוים.

select len('abcd')

מחזירה 4

REPLACE – מחליפה תת מחרוזת בתוך מחרוזת.

מקבלת מחרוזת מקורית שלמה, תת מחרוזת לחיפוש, ועם איזו תת מחרוזת להחליף.

select replace('abcd','cd','ss')

הפונקציה מחזירה 'abss'

REVERSE - פונקציה שהופכת את סדר האותיות המחרוזת מהסוף להתחלה.

select reverse('abcd')

מחזירה 'dcba'

UPPER – מקבלת מחרוזת והופכת לאותיות גדולות

select upper('abcd')

מחזירה 'ABCD'

LOWER – מקבלת מחרוזת והופכת לאותיות קטנות

select lower(' ABCD ')

מחזירה 'abcd'

פונקציות תאריך וזמן

טבלת ייצוג חלקי זמן. בכל פונקציה של זמן חייבים לספק פרמטר שיאמר באיזה חלק זמן אנחנו מטפלים באותה עמודה.

abbreviation	datepart
yy, yyyy	year
q, qq	quarter
m, mm	month
dy, y	dayofyear
dd, d	day
wk, ww	week
hh	hour
mi, n	minute
ss, s	second

הפונקציה **GETDATE**() מחזירה תאריך וזמן עכשוי

Select getdate()

הפונקציה **DATEADD** מקבלת סוג זמן, כמה להוסיף במספר, ותאריך נתון.

select dateadd(dd,12,'2016-11-27')

מחזירה : '09-12-2016'

הפונקציה **DATEDIFF** מחזירה הפרש בתאריך בין שני תאריכים נתונים.

select datediff(d,getdate(),'2016-11-27')

הפונקציה מחזירה יום 1

הפונקציה **DATENAME** מקבלת תאריך וסוג של מה נדרש ומחזירה את שמו.

select datename(M,'2016-11-27')

הפונקציה מחזירה 'november'

הפונקציה **DAY** מקבלת תאריך ומחזירה את היום.

select day(getdate())

הפונקציה מחזירה את היום

הפונקציה **MONTH** מקבלת תאריך ומחזירה את החודש במספר.

select month(getdate())

הפונקציה מחזירה את החודש

הפונקציה **YEAR** מקבלת תאריך ומחזירה את החודש במספר.

select year(getdate())

הפונקציה מחזירה את השנה

הפונקציה **EOMONTH** מקבלת תאריך ואינטרוול ומחזירה את סוף החודש הרצוי

select EOMONTH('01-01-2017',4)

הפונקציה תחזיר '31-05-2017'

פונקציות מתימטיות

הפונקציה **CEILING** מקבלת מספר עשרוני ומחזירה את השלם הגבוה הקרוב ביותר.

select ceiling(8.56)

הפונקציה תחזיר 9

הפונקציה **FLOOR** מקבלת מספר עשרוני ומחזירה את השלם הקרוב הנמוך ביותר

select floor(8.78)

הפונקציה תחזיר 8

הפונקציה **ROUND** מקבלת מספר עשרוני ובאיזה מקום לעגל אותו ומחזירה מספר מעוגל.

select round(8.798,2)

הפונקציה תחזיר 8.800

SIMPLE CASE

שיטת ההחלפה של נתונים מתבצעת גם באמצעות CASE פשוט שלא מבצע חיפושים אלא רק מחליף ערך ידוע מראש ולא על סמך ביטוי. בקטע הקוד הבא מתבצעת החלפה של ערכים על פי ערכי קוד עובד. הפלט לכל אורך הכתיבה חייב להיות מאותו טיפוס נתונים.

```
Select Case Employeeid when 1 then 'Old Worker'

When 2 then 'Mid Worker'

Else FirstName

End as [New Title]
```

SEARCH CASE

שיטת ההחלפה של נתונים מתבצעת שונה בשיטת החיפוש כי אפשר לשלב ביטויים של חיפוש ולא חייבים על אותה עמודה. ולכן לא חייבים לדעת מהו הערך הנתון אלא מחפשים עם אופרטורים. לדוגמא...

```
Select case when Employeeid between 1 and 3 then 'Old Worker'

When (Employeeid > 3) and (employeeid  between 6 and 7) then 'New Worker'

Else FirstName

End as [New Worker double]
```

קיבוץ נתונים לפי קבוצות והפעלת פונקציות סיכום.

קיבוץ הנתונים נעשה על ידי GROUP BY ובעצם מקבץ סיכומים על הקבוצה.

בדוגמה שלפנינו אנחנו מקבצים סיכומים לפי מספר ספק בטבלת Products

```
Select SupplierID, Max(unitprice),Min(unitprice)

From Products

Group by SupplierID
```

התוצאה לשאילתא תהיה המינימום והמקסימום לכל קבוצה שמקובצת על פי ספק.

אסור לעמודה להופיע ב SELECT ללא פונקציה אלא אם כן הקיבוץ נעשה לפי העמודה הזו.

פקודות מתקדמות ב-sql

T-SQL כתיבת קוד.

משתנים לוקליים: נכתבים תמיד עם כרוכית לפניהם וטיפוס נתונים אחריהם.

```
Declare @Num as Int

Declare @Name as nvarchar(50)
```

BEGIN END מסמנים בלוק כתיבה שבסופו יכול להופיע GO כמו בדוגמה הבאה.

```
Begin

Declare @x as int

Declare @y as int

Set @x=23

Select @x+@y

Print  convert(varchar(10), @x+@y)

End
```

ההבדל בין PRINT ן SELECT הוא שה PRINT מדפיס את התוצאה בתור טקסט בטאב המודעות ואילו ה SELECT מדפיס את התוצאה בתור טבלה ב RESULTS

איך נבצע תנאים של IF ELSE

במידה וצריך שורת פקודה אחת אחרי התנאי אין צורך ב BEGIN ו END אלא אפשר להסתפק ב IF ושורת פקודה.

לדוגמה:

```
If (@x>56 and @y<78)

        Select @x+@y

Else

        Print ('All Is Good')
```

איך נבצע תנאי IF כשיש בלוק של פקודות? נפתח ב BEGIN ו END בלוקים של פקודות

לדוגמה:

```
If (@x>@y)
Begin
        Set @x=@x + @y
        Print @X
End
Else
Begin
        Set @y=@y + 6
        Select @y
End
```

לולאת WHILE משמשת לאיטרציות על הטבלה. זאת אומרת סיבובים על הטבלה והדפסת ערכים או מניפולציה של ערכים.

לדוגמה:

```
Declare @Count as Int
Declare @Mone as In
Set @Mone= (select count(*) from Employeeswhere Employeeid > 4)
Set @Count=1
While (@Count>@Mone)
Begin
         Select FirstName where Employeeid = @Count
          Set @Count++
End
GO
```

פרוצדורות שמורות.

פרוצדורות מקלות על הגישה לבסיסי הנתונים ומאפשרות שמירת קוד T-SQL בתוך בסיס הנתונים והן עוברות קומפילציה פעם אחת בלבד אחרי השינוי וההטמעה.

יצירת פרוצדורה:

```
Create Procedure P_Test

      @x as int=0

AS

Select * from employees where Employeeid > @x

GO
```

הפרמטר שמתקבל מבחוץ נקרא בפרוצדורה x@ והוא מקבל ברירת מחדל 0

אם לא נשלח מבחוץ.

איך מריצים פרוצדורה?

```
Execute P_Test 3
```

איך מוחקים פרוצדורה?

```
Drop procedure P_Test
```

איך משנים פרוצדורה? בעזרת פקודת ALTER

```
Alter Procedure P_Test

    @x int =9

AS

    Select * from Employees where Employeeid between @x and 90

GO
```

איך מייצרים VIEW?

VIEW הוא שאילתא שמורה שמונעת מהמשתמש להתקרב לטבלאות של בסיס הנתונים ומאפשרת למנהל בסיס הנתונים להחליט איזה מידע לחשוף בפני המשתמש.

ה VIEW לא מקבל פרמטרים ומסוגל לבצע שאילתות מכל הסוגים וגם הוא נשמר בבסיס הנתונים לתמיד עד שיימחק.

```
Create View V_Test

AS

   Select CategoryName , Max(UnitPrice) as [MaxPrice]

From Products as P join Categories as C

On P.CategoryID=C.CategoryID

Group by Categoryname


GO
```

איך משתמשים ב VIEW - מתייחסים אליו כמו טבלה ומבצעים עליו שאילתות.

למשל:

```
Select * from V_Test where MaxPrice> 45
```

טריגרים

טריגר הוא אובייקט של בסיס הנתונים והוא מופעל כשאנחנו עושים הכנסת נתונים או עדכון או מחיקה ולא רואים את הפעולה שלו שהוכנה מראש. למשל אם הכנסתי נתונים לטבלת פרטי הזמנות ואני רוצה להקים רשומה גם בטבלת הזמנות באותו זמן הכנסת נתונים.

יצירת טריגר

```
Create Trigger T_TestIsert

 On Employees

For Insert

AS

Declare  @x int

Set @x= Employeeid from Inserted.Employeeid

Insert into TestTable (id,Description)

Values(@x,'this is the test')

GO
```

במקום FOR אפשר גם לכתוב AFTER או INSTEAD

יש צורך לשאוב את הערכים מטבלת ה INSERTED או מטבלת DELETED

ואם רוצים לדעת אם עמודה התעדכנה יש לשאול If Update(column name)

שינוי טריגר אף הוא נעשה בעזרת ALTER