

S. q. 1

OSU1527427
OSU113736
OSU113736

בס"ד

כל הזכויות שמורות ר.ט.

03-6193806

ברסלב 9 בני ברק

שבט תשע"ז

Rivka37@gmail.com

תוכן עניינים

5	שאלות נחוניות – data query language DQL
5	שאילות פשוטות
8	ביטוי חישובי
11	פונקציות סקלריות (scalarfunction)
13	ערכים עליונים TOP
14	פונקציות מובנות
14	Group by – קיבוץ
16	צירוף טבלאות
17	צירוף טבלה אל עצמה – self join
19	שאילות מקוונות sub query
21	חת שאלות מתואמות
23	צירוף טבלה אל שאילתך
24	שאלות מורכבות
25	האופרטור union all
25	האופרטור intersect – חיתוך
25	האופרטור except
26	הפעול נתונים – Manipulating data
26	הכנסת נתונים אל תוך טבלה – Insert
27	עדכון נתונים קיימים – update
28	מחיקת נתונים מטבלאות delete
30	הגדרת נתונים – data definition language DDL
31	המשפט create table
32	שינוי הגדרות הטבלה alter table
33	биיצוע truncate לטבלה
33	מחיקת טבלאות – drop
34	הוספת אילוצים לטבלה constraint
36	אינדקסים
37	הרשות גישה לבסיס נתונים
37	טבלה זמנית
38	– טבלאות מדומות views
41	נספח א תרגילים
44	נספח ב
56	תשובות עמוד 43

מבוא
מהי שפת SQL?

שפה שאלותית מבנית – Structure query language-SQL

שפה SQL הינה השפה התקנית המשמשת כדי לתקשר ולהגדיר את צרכיך במסד נתונים יחסית-טבלאי.

מהו מסד נתונים?

במונחים פשוטים ביותר, מסד נתונים (database) הינו אוסף של נתונים. יש המעדיפים לחשב על מסד נתונים כעל מנגן מאורגן שיש לו את יכולת לאחסן מידע, אשר באמצעותו המשתמש יכול לאחזר את המידע בדרך ייעילה וחסכונית.

השימוש במסדי נתונים נפוץ ביותר, מבייל משים אנו נתקלות בשימוש מדי יום כגון: ספר טלפונים – הנתונים הכלולים בו שמורות כתובות ומספר טלפון וכן בנקים, ספריות ועוד...

רשימות הנתונים מסווגות לפי אינדקס אלפביתית או כל אינדקס אחר, המאפשר לאתר בקלות את הנתון הרצוי, בסופו של דבר, נתונים אלו מאוחסנים במסד הנתונים במקום כלשהו במחשב.

הנתונים במסד הנתונים עוברים מדי פעמי תהליך עדכון/מחיקה והוספה, אם השתמש בדוגמה ספר הטלפונים, אין מקלדים מדי שנה את כל נתונים הספר כגון שם ומספר טלפון כאשר מדפיסים מהדורה חדשה אלא מתחזקם את מסד הנתונים כשאנשיים עוברים למדינה אחרת יתכן ונctrax למחוק ערכיהם ממש הנתונים באותו מדה, יתכן שייצרכו לשנות ערכים של אנשים העוברים לכתובת אחרת במדינה זו, משנים שמורות או את מספרי הטלפון שלהם, וכך להלאה.

מהו מסד נתונים יחסית?

ניתן לבנות מסד נתונים באופנים שונים, השיטה הנוחה ביותר הינה מסד נתונים יחסית-טבלאי (relational database) זהו מסד נתונים המחולק לטבלאות (tables), המתיחסות – הקשורות (relate) זו לזו בתחום מסד הנתונים באמצעות מפתחות (keys) או שדות (fields) משותפים, ובאמצעות יחס – קשר זה ניתן לשולף נתונים ממספר טבלאות ולהציג בטבלה אחת. SQL מאפשרת הגדרת מסד נתונים יחסית ותחזוקתו. בSQL נבנה את הטבלאות ונבצע פעולות/שאלות לתחזוק הטבלאות שבנונו כגון: קריית נתונים, עדכון, מחיקה והוספה.

הבדלי יישום SQL בין היצרנים השונים:

החברות הבולטות ביותר ביחסם למסד נתונים יחסית הינן:

ANSI oracle Sybase, Informix, Microsoft, ibm ו-禺ת רבות כולן התאימו את היישום לתקן ANSI לצורך נידות ונוחות המשתמש. לדוגמה, אם חברה מהגרת משרת מסד נתונים אחר לאחר, יהיה זה מיאש מדי למשתמשי מסד נתונים אם יאלצו ללמידה שפה אחרת כדי לשמר על תפקודיות המערכת החדשה.

אך בכל אחד מהIMPLEMENTATIONS לשפת SQL קיימים שיפורים/הרחבות מעבר לתקן ANSI שהינן אפשריות בונוס לתקן הסטנדרטי.

בקורס זה נישם את שפת SQL על תוכנת **Microsoft SQL SERVER** של היצרנית Microsoft. שפת SQL אינה שפה מלאה אלא רק שפת נתונים המאפשרת גישה וטיפול נתונים בנסיבות הקיימים במסד נתונים.

כדי לבצע עיבוד מלא של הנתונים ולהפיק ממשק ידידותי למשתמש יש צורך להשתמש בשפה מאורחת ויחדיו ליצור תוכנה מושלמת.

סוגי פקודות בשפת SQL

להלן נפרט את הקטגוריות הבסיסיות כדי לבצע פונקציות שונות:

1. שפה לשאלתה על נתונים – data query language DQL
2. שפה להגדלת נתונים – data definition language DDL
3. שפה לתפעול נתונים – data manipulation language DML
4. שפה לבקרה על נתונים – data control language DCL
5. פקודות לניהול נתונים – data administration commands
6. פקודות לבקרה על טרנזקציות – transaction control commands

SQL-ל ACCESS

ההשוואה בין תוכנות של ACCESS לבין אלו של SQL SERVER דומות להשוואה בין מחשב במעמד שרת מול מחשב שולחני פשוט. ACCESS מוגבלת בתחום של מסד נתונים שולחני לעומת SQL SERVER העונה לדרישות מסד נתונים תאגידי המשרת/API משתמשים בו בזמן ובנפח תנועות גדול.

אין מדובר בשני מוצרים סותרים, אלא במוצרים שתוכלי לשלב ביניהם בצורה מושלמת.

להלן תוכנות SQL SERVER שאינן קיימות ב-ACCESS – מערכת המבוססת על ארכיטקטורת שרת/לקוח, השרת משוחרר מביצוע משימות השיכוך ללקוח.

-אפשרויות ביצוע גיבוי מסד נתונים מלא או גיבוי תוספות ושינויים ללא ניתוק המשתמשים בעת הגיבוי.

-תמכה באימונות וabweזה ממוקדים לגבי המשתמשים והאובייקטים שהם מנהלים.

-ניתול המשאים: מערכות הפעלה והמעבדים צדי להבטיח יציבות וביצועים.

-מציע שפע אפשרויות מובנות ומאפשר למשתמשים להתואש מסעון באמצעות תוכנות מובנות כמו יומני תנועות (transaction logs).

-יכולת לתמוך במסדי נתונים גדולים מאד, בטוחה של טרה בייטים (TB).

עם זאת, ניתן לשלב את היתרונות של מערכת ACCESS ולנצלם לשימוש נכון ויעיל.

ACCESS יכולות גרפיות מרשים, הଘותה בהרבה מהיכולות של שרת SQL. כך למשל, נוכל ליצור במהירות טפטים להזנת נתונים ע"י אשף הטפטים, ודווחות להדפסת נתונים ע"י אשף הדוחות

data query language DQL –

שאלות פשוטות

חלק זה מורכב מפקודה אחת בלבד, שהינה המוקד המרכזי ביותר בשפת SQL.
הפקודה הינה:

SELECT – בחר

פקודה זו, המלווה באפשרויות (options) ובפסוקים (clauses) ורבים, משמשת לשיליפה ממספר טבלאות לטבלה אחת, באמצעות שאלות פשוטות עד מורכבות.
מהי שאלתך?

שאלתך (query) היא שאלת למסד הנתונים באמצעות המשפט בשפת SQL המחזיר את התשובה לשאלת המבוקשת.

בשיעורים הבאים נפרט את הפקודות העיקריות בקטגוריות הנוספות שהזכרנו להלן.
לפנינו מסד נתונים לדוגמה:
המכיל את חמשת הטבלאות:

employee_tbl.1

(empid,lastname,firstname,phone,address,city,managerid,position,datehire,birthdate)

employee_pay_tbl(payid,empid,payrate,datelastraise,salary,bonus) .2

customers_tbl.3

(custid,custname,custaddress,custcity,custstate,custzip,custphone,custfax,empid)

orders_tbl(ordnum,custid,prodid,quantity,orddate) .4

products_tbl(prodid,proddesc,cost,price_buy,qty_in_stock) .5

באור הקשר בין הטבלאות:

העובדים-employees מקבלים משכורות employee_pay עבור לקוחות customers המבקשים

הזמנות-orders של מוצרים – products –

עובדים-Employee_tbl

Empid מספר עובד	Lastname משפחה	Firstname שם פרטי	birthdate תאריך לידה	position תפקיד	Phone טלפון	Address רחוב	City עיר	managerid	Datehire תאריך כניסה
1111111111	לי	דני	01/02/1975	1	3232222	ר' עקיבא	ירושלום	4444444444	01.01.1995
2222222222	כהן	צחי	30/05/1990	2	5454444	נרקיס	תל אביב	8888888888	01/01/2010
3333333333	ברגר	משה		2	8884544	שושנה	רמת גן	8888888888	
4444444444	אלוני	דוד		3	8999999	פרחים	גבעתיים		
5555555555	כהן	אהרון		1	8888885	צבי	בני ברק	4444444444	
6666666666	שורוני	רמי		1	6565656	אביים	תל אביב	8888888888	
7777777777	נחומי	אברהם		2	5455554	חכצלה	ירושלום	4444444444	
8888888888	שדעבי	יצחק		3	534989	שושנה	בני ברק		

משכורות עובדים-Employee_pay_tbl

payid קוד תשלום	Empid מספר עובד	Payrate שכר לשעה	Datelastraise תקופה	Salary משכורת	Bonus ボונוס
1	1111111111	60	01/01/2001	6500	120
2	2222222222	100	01/02/2006	5000	30
3	1111111111	25	01/01/1990	1500	300
4	4444444444	50	01/08/2002	6500	300
5	2222222222	30	06/10/2008		400

לקוחות-Customers_tbl

Custid מספר לקוח	Custname שם	Custaddress רחוב	Custcity עיר	Custstate מדינה	Custzip מיקוד	Custphone מספר טלפון	Custfax מספר פקס	Empid מספר עובד
1	דני	רשבם	ירושלום			2656555	564555	1111111111
20	דוד	רבינו תם	בני ברק			4534544		
39	נתיה	הנriskis	רמת גן			4545444	767676	2222222222
46	אברהם	החכצלה	בני ברק			8773222		
85	יצחק	הרकפת	חולון			6756756	564566	7777777777

הזמנות-Orders_tbl

מוצרים - Products_tbl				
Prodid קוד מוצר	Proddesc שם מוצר	Price_buy מחיר קניה	Cost מחיר	Qty stock
100	מחברת	0.5	0.80	500
101	דףת	1	1.50	100
102	קלסר קטן	5	7.60	230

Ordnun מספר הזמנה	Custid מספר לקוח	Prodid מספר מוצר	Qty כמות	Orddate תאריך הזמנה
1	20	100	5	01/01/2006
2	102	100	4	07/07/1985
3	999	105	9	03/05/1989

תחביר בסיסי של משפט select

Select [*:all:distinct] column1,column2 (רשימת עמודות)

From table1,table2 (רשימת טבלאות)

Where condition (תנאי לוגי)

- **Select** – המילה הפותחת כל שאלת אחזור נתונים ולאחריה תופענה רשימת העמודות אותן נבקש להציג בפלט השאלה: **column1,column2** (column1 וcolumn2).

– לאחר מילת מפתח זו תופיע שם הטבלה בה העמודות המפורטוות בשאלת זו, (ניתן לפרט עמודות מסוימות טבלאות ובחלק זה לציין את רשימת הטבלאות שבהם העמודות שהຕבקשו בשאלתך).

דוגמה לשאלתה פשוטה:

אחזור שמות המוצרים הקיימים בטבלת מוצרים **select proddesc from products_tbl**
* הכווכית משמשת לציין כי יש להציג את כל העמודות הקשורות בטבלה ואז אין צורך בפרוט שמותיהן לאחר ה-**select**, אך עדיף לרשום את שמות השדות ולא לשולף עמודות שאינן נצרכות.

לדוגמה: שיליפת כל העמודות בטבלת הזמנות: **select * from orders_tbl**
התוצאה: כל טבלת הזמנות תופיע

האפשרות **ALL** משמשת להציג את כל הערכים עבור העמודה לרבות ערכים כפויים – זהה פקודת ברירת מחדל ואין חובה לצייןנה.

דוגמה: שיליפת שמות הערים בהם מתגוררים לקוחות:

select all custcity from customers_tbl

האפשרות distinct מציגה שורות ללא כפליות.

לדוגמה: שיליפת עיר מגוריו לקוחות ללא כפליות:

select distinct custcity from customers_tbl

תנאי לוגי-Where

- **Where** – מאפשר הגדרת תנאי לוגי שיקבע איזה שורות ישלו מຕוך הטבלה, התנאי נרשם באמצעות אופרטור.

מהו אופרטור?

מילה שומרה אותו שמור המשמש בעיקר ב-**where** של משפט SQL על מנת לבצע פעולה כגון השוואת ופעולות חשבון, באופרטורים משתמשים כדי לציין תנאים במשפט SQL וכי לשמש מילوت חיבור לתנאים מרובים במשפט. האופרטורים האפשריים לשימוש במשפט **where**.

- אופרטורים השוואתיים לגודים כדוגמ: =,<,>.

דוגמה: הציג את שמות לקוחות הגרים בחולון:

Select custname From customers_tbl Where city='חולון'

- אופרטורים לוגיים לגודים כדוגמ:

In – משמש כדי להשוו ערך לרשימה של ערכים מילוליים שפורטו, כדי שיווזר true/false ייבוא.

הערך המושווה להתאים לפחות לאחד מהערכים שברשימה.

לדוגמה: הציגי את הזמנות של לקוחות שמספרם הוא: 20 או 102 או 85

Select * From orders_tbl Where custid in(20,85,102)

הערך המינימלי והערך המקסימלי.
- Not between/Between

דוגמיה: הציגי את פרטי הזמנות שכמותם בין 75 ל-87

Select * From orders_tbl Where qty between 75 and 87

– אופרטורים אלו משמשים בעיקר לצורכי מספר תנאים על שאלה אחת.

דוגמיה: הציגי את מספרי העובדים שהבונוס שלהם הוא 300 ומשכורת גדולה מ-30000

Select empid From employee_pay_tbl Where bonus=300 and salary>30000

תרגילים:

1. הציגי את נתוני טבלת העובדים (ע"י * או ע"י פרוט השדות)
2. הציגי קוד לקוחות, קוד מוצר ותאריך הזמנה מטבלת הזמנות בתנאי שהקוד מוצר = 105
3. הציגי את שמות הלקוחות שגרים בירושלים או חולון
4. הציגי את שמות המוצרים שמחירים בין 1 ל-4 ש"ח
5. הציגי את כל פרטי טבלת הזמנות שהתבצעו בשנת 2010 ע"י לקוח 20.
6. הציגי את כל פרטי טבלת הזמנות כאשר קוד מוצר = 10 או 103 והכמות קטנה מ-40.
7. הציגי את קודי העובדים שימושכורותם גדולה מ-15000 והבונוס שלהם מעל 170 לפאו
שהבונוס דיק, שימי לב להציג כל עובד פעם אחת בלבד.
8. הציגי הזמנות שלא נרשט עבורם ותאריךן לא התבצעו ב-2011.
9. הציגי שמות עובדים שתפקידם הוא 3 ונולדו אחרי 1980.
10. הציגי משכורות שחולקו משנה 2006 בלבד

ביטוי תיישובי

משתמשים באופרטוריים (, + , - , / , *)

לדוגמה: רשימת העובדים התקופה והמשכורת בתוספת הבונוס.
התוצאה: רשימת העובדים התקופה והמשכורת בתוספת הבונוס.

שרשור עמודות

ניתן לחבר עמודות נפרדות לעמודה אחת

לדוגמה: select custname + city from customers_tbl

התוצאה:

משה ירושלים
חיימטל - אביב

להוספה רוח אחד: select custname + ' ' + city

להוספה מילה: select custname + ' live in ' + city

שם נרדף לעמודה

ניתן לשנות את שם העמודה בתצוגה

לדוגמא: select empid, datelasraise, salary+bonus as allsalary

עמודות ב-select שאין עמודות מהטבלה

ניתן להציג select ערך כלשהו שאינו שם של עמודה

לדוגמא: select 'm-100', 50, getdate()

הפונקציה like

משתמשים ב- like כדי לחפש חלק ממחרוזת.

% אפס תווים או יותר

_ تو אחד בלבד

דוגמא: כל הרשומות המקוריות ב- A: 'A %'

Mastiyim B-A: '%A'

'%A %' : A מכיל

' _A %' A אות שלישי: '%_A'

יש רגשות לגביות האותיות

דוגמא: כל העובדים ששמות המקוריים באות 'Y' ו Mastiyim ב'ה' ובאמצע שני אותיות כלשהם.:

Select empid from employee_tbl where lastname like 'h _y'

הפונקציה not like

לדוגמ: כל העובדים שלא גרים בירושלים

Select empid from employee_tbl where city not like 'Jerusalem'

הפונקציה is not null/is null

משתמשים בתנאי null למציאת שדה ריק

דוגמא: select * from products_tbl where cost is null

תוצאה: כל המוצרים שאינם מחיר.

מיון order by

כאשר רוצים למיין את השורות בתוצאות משפט select

משתמשים בפקודה order by ואחריה את שם עמודה או מיקומה, פקודה זו היא תמיד אחרונה

. select משפט ה-

- סדר עולה (ברירת מחדל)

- סדר יורד

.select * from employee_tbl order by lastname

דוגמא נוספת: order by city desc, lastname - מיון ראשון לפי עיר בסדר יורד.

מיון שני לפי שם, בסדר עולה (ברירת מחדל ולכ"ן לא כתוב asc).

אם רוצים סדר יורד צרייך לכתוב desc אחרי כל אחד.

ניתן להשתמש במיקום (מספר) העמודה.

לדוגמא: select empid, position, salary from employee_pay_tbl order by 3;

תרגילים:

1. הציג את כל העובדים ששמות מתחילה באות מ או מסתיים באות נ או מכיל את האות ו
2. הציג את כל העובדים שאין להם תאריך לידה או שאין להם טלפון.
3. שלפי את שם המוצר מחיר קניה ומהירות מכירה בתוספת 17% וכן חישוב של הפרש בין מחיר מכירה למחיר קניה.
4. המשכורת של העובדים מיוצגת בדולרים, הציג אותה ב שקלים בהנחה שהدولר 3.777
5. הציג עיר עבד מטבלת עובדים. כל עירתוופיע פעמי אחת בלבד.
6. הציג קוד עובד, תאריך, ומשכורת+בונוס. מינני לפי קוד עובד, ומין יורד לפי משכורת.
7. חשבו לכל עובד את שנות הותק ע"י הפונקציה datediff הפרש בין תאריך כניסה לתאריך נוכחי. הציג רק עובדים שהותק שלהם יותר משנה.
8. הציג שם עובד בר"ת אות ראשונה שם משפחה ואות ראשונה שם פרטי
9. הציגי מספר הזמנה, קוד מוצר וכמות, ועמוזה נוספה תייצג ע"י התו * את כמות הזמנה לדוגמא: כמות 2 ** כמות 4 **** וכו' ע"י הפונקציה replicate
10. הציג את שמות העובדים, וחשבו את גילם. הציג רק עובדים שגילם גדול מ-40.
11. הציג שם עובד ומספר טלפון כל סימן '-' במספר הטלפון יוחלף לו רווח (פונ' replace)
12. הציגי מס הזמנה, החודש ושם
13. הציג את ההזדמנות שה הפרש בין תאריך הזמנה לתאריך נוכחי גדול משלושה חודשים.
14. הציגי תז, וחשבו ע"י פונקציה len את מספר התווים של התז.
15. הציגי שם משפחה+שם פרטי+כתובת בעמודה אחת. שימי לב שכאשך אחת מעמודות ריקה לא יוצג NULL העוזר בפונקציה isnull

פונקציות סקלריות (scalarfunction)

פונקציות מתמטיות-mathematical function		
ABS	ABS(numeric_expression)	הfonקציה מחזירה ערך מוחלט של גטור
CEILING	CEILING(numeric_expression)	הfonקציה מחזירה את הערך השלם הגדל הקרוב ביותר למספר שהתקבל כארוגומנט הפונקציה
FLOOR	FLOOR(numeric_expression)	הfonקציה מחזירה את הערך השלם הקטן הקרוב ביותר למספר שהתקבל כארוגומנט הפונקציה
PI	PI()	הfonקציה מחזירה את הערך קבוע PI
POWER	POWER(numeric_expression,y)	הfonקציה מבצעת העלאה בחזקה
RAND	RAND()	הfonקציה מחזירה מספר אקראי בין 0 ל-1
ROUND	ROUND(numeric_expression,length)	הfonקציה מעגלת מספר שהתקבל למספר המיקומות העשורי שצוין
SIGN	SIGN(numeric_expression)	הfonקציה מחזירה 0 אם המספר שווה 0, 1 אם המספר חיובי ומחזירה -1 אם המספר שלילי.
SQUARE	SQUARE(float_expression)	הfonקציה מחזירה ריבוע (העלאה בחזקת 2) של מספר שנייתן.
SQRT	SQRT(float_expression)	הfonקציה מחזירה שורש ריבועי של מספר שנייתן

string function		
LEFT	Left(str,number)	פונקציה המחזירה חלק שמאלית של מחזורות שהתקבל בקלט
LEN	Len(str)	פונקציה המחזירה את אורך המחרוזות
LOWER	Lower(str)	פונקציה ההופכת את הטקסט לאותיות קטנות
LTRIM	Ltrim(str)	פונקציה המקצתת תווים וריקים מצידה השמאלי של מחזורות
REPLICATE	Replicate(var,number)	פונקציה מקבלת تو ומספר ומPLICת את התו מספר פעמים כמספר שקיבלה
REVERSE	Reverse(str)	פונקציה ההופכת את האותיות במחזורות
RIGHT	Right(str,number)	פונקציה המחזירה חלק ימני של מחזורות שהתקבל
RTRIM	Rtrim(str)	פונקציה המנקה תווים וריקים מצידה הימני של מחזורות
Replace	replace (str, ' ', ' ')	פונקציית החלפה
SPACE	Space(number)	פונקציה המוסיפה רווח
SUBSTRING	Substring(str,starting,length)	פונקציה המחזירה חלק ממחרוזות מקום מסוים

UPPER	Upper(str)	פונקציה המחזירלה אותיות גדולות באנגלית
Isnull	Isnull(str,' '), isnull(salary,0)	אם str ריק מוחזר ערך הרשות בארגומנטן 2 לדוגמא: 0

		פונקציות התאריך והזמן
		<u>– חלק של תאריך</u>
Year	yyyy	כasher מופיעה בפונקציה המילה interval
Quarter	q	
Month	m	
Day of year	y	יש לרשום במקומה את אחת מהאפשרויות
Day	d	הרשומות בטבלה זו.
Weekday	w	
Week of year	ww	
Hour	h	
Minute	n	
Second	s	
DATEADD	dateadd(interval, 7, getdate())	הוספה ימים חדשים או שנים לתאריך נתון
DATEDIFF	Datediff(interval,date1,date2)	פונקציה המחזירה הפרש בין תאריכים
DATENAME	datename(interval, getdate())	פונקציה מקבלת תאריך וחלק ממנו (יום, חודש...) ומוחזירה את שמו
DATEPART	datepart(interval, getdate())	פונקציה המחזירה מספר המיצג חלק מתאריך (לדוגמה: יום,חודש)
DAY	Day(date)	פונקציה המחזירה יום מתאריך נתון
GETDATE	Getdate()	פונקציה המחזירה את התאריך הנוכחי
MONTH	Month(date)	פונקציה המחזירה את החודש מתוך תאריך
YEAR	Year(date)	פונקציה המחזירה את השנה מתוך תאריך
		פונקציות המרת
CAST	CAST(expression AS data_type)	המרה מטיפוס נתונים אחד לאחר
CONVERT	CONVERT(varchar(9),getdate())	המרה מטיפוס נתונים אחד לאחר

תאריך בפורמט

לדוגמא התאריך: 12/30/200

```
--1 select convert(varchar, getdate(), 1) 12/30/06
--2 select convert(varchar, getdate(), 2) 06.12.30
--3 select convert(varchar, getdate(), 3) 30/12/06
--4 select convert(varchar, getdate(), 4) 30.12.06
--5 select convert(varchar, getdate(), 5) 30-12-06
--6 select convert(varchar, getdate(), 6) 30 Dec 06
--7 select convert(varchar, getdate(), 7) Dec 30, 06
--10 select convert(varchar, getdate(), 10) 12-30-06
--11 select convert(varchar, getdate(), 11) 06/12/30
--101 select convert(varchar, getdate(), 101) 12/30/2006
--102 select convert(varchar, getdate(), 102) 2006.12.30
--103 select convert(varchar, getdate(), 103) 30/12/2006
--104 select convert(varchar, getdate(), 104) 30.12.2006
--105 select convert(varchar, getdate(), 105) 30-12-2006
--106 select convert(varchar, getdate(), 106) 30 Dec 2006
--107 select convert(varchar, getdate(), 107) Dec 30, 2006
--110 select convert(varchar, getdate(), 110) 12-30-2006
--111 select convert(varchar, getdate(), 111) 2006/12/30
פורט שיעות
--8 or 108 select convert(varchar, getdate(), 8) 00:38:54
```

```
--9 or 109 select convert(varchar, getdate(), 9) Dec 30 2006
12:38:54:840AM
--14 or 114 select convert(varchar, getdate(), 14) 00:38:54:840
```

עדכים עליונים TOP

האופרטור top מאפשר לשלוף מתוך הטבלה כ ערכים ראשוניים, או יכול ספרה כלשהי.

לדוגמא: select top 2 * from product order by cost

התוצאה: שני שורות שבןן המחיר נמוך ביותר. מפני שהטבלה ממוקנת ע"פ המחיר ובકשו שני שורות בלבד ע"י כתיבת הספרה 2 אחרי המילה top .

יש אפשרות לשלוף מתוך הטבלה כ אחוזים ראשוניים, ואז נרשם אחרי הספרה את המילה

percent

לדוגמא: select top 2 percent * from product order by cost

תרגילים

1. שלפי את 3 המשכורות הגבוהות ביותר

2. שלפי 30 % מהמושרים הזולים ביותר.

3. הציגי 2 הזרנות אחרונות.

4. הציגי עובד ותיק ביותר.

פונקציות מוגבלות

ספרה של מס' מופעים בשאלתה - count

חישוב ממוצע של ערכים Avg

חישוב סכום ערכים - Sum

מציאת הערך הגבוה ביותר - Max

מציאת הערך הנמוך ביותר - Min

Group by-קיבוץ

בשאלתה המכילה פונקציה צבירה במשפט select ובנוסף לכך עמודות רגילות. כל עמודה רגילה (שאינה פונקציה קבוצתית) אשר מופיעה ב-select חייב להופיע ב-group by.

משמעות פולת ה-group היא קיבוץ לפי. מקבץ שורות זהות אם כל הנתונים בכל העמודות זהים. וכך כל עמודה שנוטipa למשפט select תתווסף ל-group ותפחית את מספר השורות הקפולות.

לדוגמא: select custid, avg (qty) from orders_tbl group by custid

תוצאה: חישוב ממוצע כמהות לכל לקוח.

דוגמא נוספת: select custid, prodid, avg (qty)

From orders_tbl

Group by custid, prodid

תוצאה: רק כאשר ישנה שורה בהם הלקוח וה מוצר זהים יקצטם ויחשב ממוצע וכן מחשב ממוצע לכל לקוח בכל מוצר.

Having-תנאי על שורות מקובצות

המשפט Having מציב תנאי על עמודה מקובצת

ע"י הפונקציות (count, sum, max, min, avg).

שים לב! נשתמש ב-where כדי להגדיר תנאי על עמודה רגילה

נשתמש ב- having כדי להגדיר תנאי על עמודה שהיא תוצאה של פונקציה.

Select custid, avg (qty) from orders_tbl

Group by custid

having avg (qty)>9

תרגילים:

1. הציג את הסכום הכללי של המשכורות שוחולקו משנהות 2000

2. הציג את סכום המשכורות לכל העובדים שהמשכורת+בונוס הוא בין 1000 ל-3000..

3. הציג עיר וכמות העובדים הגרים בה.

4. הציג קוד עובד וסך כל משכורתו בשנה הנוכחית.

5. הציגי קוד ל��ח וכמה פעמים בצע הזמנה.
 6. הציגי 2 ל��חות שכמות הפעמים שביצעו הזמנה נמוּך ביותר.
 7. הציגי שנה וכמות העובדים שנולדו בה.
 8. הציגי את המשכורת המומוצעת לכל חודש ונה בתנאי שהמומוצע עולה על 3000 ש"ח, מיini ע"פ השנה והחודש.
 9. הציגי כמות הזמנה מקסימלית מכל מוצר שמספרו בין 100 ל-200 בתנאי שהכמות המקסימלית גדולה מ-5, מיini ע"פ הכמות המקסימלית בסדר יורד.
 10. ספרי כמות משכורות לכל עובד בשנת 2009 הציגי רק עובדים שכמות השירות היא: 12.
 11. הציגי לכל עובד משכורת מינימלית רק לעובדים שמשכורותם המינימלית נמוכה מ-1500 ש"ח.
 12. הציגי קוד עובד, שנת המשכורת וסכום משכורת כולל (salary+bonus). לכל עובד תופיע שורה אחת כוללת סך כל משכורתו בכל שנה.
 13. הציגי ל��ות שהזמין לפחות שני מוצרים שונים.
 14. הציגי לכל מוצר תאריך קניה ראשון, תאריך קניה אחרון, כאשר ההפרש בין הראשון לאחרון קטן מ-10.
 15. הציגי לכל שנה וחודש סכום משכורות כולל, כאשר הסכום הכלול גדול מ-5000 ש"ח והשנה גדולה מ-2001.
 16. הציגי 2 מוצרים שסך כל הכמות שהוזמנה (qty) sum מהם גבוה ביותר.
 17. הציגי עובד שסכום המשכורת שלו בשנת 2006 נמוּך ביותר.
 18. הציגי כמה אנשים נכנסו לעבוד בכל שנה, הציגי את השנה שבה נכנסו הכי הרבה עובדים.
 19. הציגי ממוצע גילאים של העובדים בכל עיר.
 20. הציגי כמה ל��ות בצעו הזמנות.
 21. הציגי כמה עובדים הביאו לפחות.
 22. הציגי לך, שנה וסך כל הכמות שהוזמנה.
 23. הציגי לך כמה שהזמין כמות כוללת גדולה מ-50 משנה 2005.
- תשובות בעמוד 44

```

select
from
where
group by
having
order by

```

<u>סדר ביצוע פעולות</u>		<u>select</u>	<u>סכום - תחביר משפט</u>
6		Select column, group- function (column)	רשימת עמודות*
1			רשימת טבלאות-From table
2		[Where condition]	תנאי על שורות לא מקובצות
3		[group by group- by- expression]	מקבץ שורות זהות לפי העמודה שנבקש
5		[having group- condition]	תנאי על שורות מקובצות
7		[order by column]	מיון לפי עמודה מסוימת
4			חשיבות פונקציות קבוצתיות
8			Top

צירוף טבלאות

- מילת מפתח שאחריה יופיעו שמות הטבלאות שברצונו לשלוף מהן.
 אם לא מצינים קישור בין הטבלאות, התוצאה תהיה מכפלה קרטזית.
 מספר השורות בטבלה אחת כפול מספר השורות בטבלה השנייה.
 לדוגמה: select lastname, position from employee_tbl, employee_pay_tbl

צירוף פנימי: inner join

שאילתה מסוג זה תשליף רק את השורות שיש להן נתונים בשתי הטבלאות.
 לדוגמה: הציגי את שמות העובדים ו^הsalary.

Select lastname, salary from employee_tbl inner join employee_pay_tbl

On employee_tbl.empid=employee_pay_tbl.empid

תחביר פקודות הקישור: קישור on שם טבלה inner join שם טבלה
 - ניתן לחתם שם נרדף לטבלה לצורכי קיצור ונוחיות הכתיבה.

לדוגמה: Select e.empid, lastname, salary

from employee_tbl e inner join employee_pay_tbl ep

On e.empid=ep.empid

כאשר משתמשים במשפט select בעמודה שמהו זהה בשני טבלאות כגון: empid. יש לרשום את
 שם הטבלה או הכינוי שלה לפני שם השדה כגון: employee.empid או pid

קישור ליותר מאשר שני טבלאות

לדוגמה: הציגי את שמות העובדים, שמות הלקוחות וכמות החזמנה.

select lastname,custname,qty

from customers_tbl c inner join orders_tbl o

on c.custid=o.custid inner join employee_tbl e

on e.emp_id=c.empid

צירוף חיצוני- outer join

שאילתת מסוג זה תשלוף גם שורות שאין להן נתונים בשני הטבלאות בנוספּע על שורות שבן נתונים.

נתונים

מ-2 טבלאות.

האפשרויות הן:

א. Full outer join – להראות כל השורות משני הטבלאות.

ב. left join - להראות כל השורות מטבלה ראשונה למורות שאין נתונים בשנית.

ג. right join - להראות כל השורות מטבלה שנייה למורות שאין נתונים בראשונה.

לדוגמא: הציגי שם עובד, שם לקוח כולל עובדים ללא לקוחות, וכן לקוחות ללא עובדים.

select lastname+' '+firstname as empname , custname

from employee_tbl e full outer join customers_tbl c

on e.emp_id=c.empid

right join

left join

full outer join

	empname	custname
1	דני כהן צחי	
2	NULL דוד	
3	נתיבי כהן צחי	
4	גדי ברגר משה	
5	יעזק ברגר משה	

	empname	custname
1	לוריא דבי	NULL
2	לורי דבי	NULL
3	ויל דוד	NULL
4	דני כהן צחי	
5	נתיבי כהן צחי	
6	NULL	דוד
7	יעזק ברגר משה	
8	גדי ברגר משה	

	empname	custname
1	לוריא דבי	NULL
2	לורי דבי	NULL
3	ויל דוד	NULL
4	דני כהן צחי	
5	נתיבי כהן צחי	
6	NULL	דוד
7	יעזק ברגר משה	
8	גדי ברגר משה	

תחביר פקודת קישור בין טבלאות ניתן לרשום באופן נוספּע.

ב-from תופיע רשימת הטבלאות מופרדות בפסיק, וב-where נרשום את הקישור בין הטבלאות.

לדוגמא: Select prodid, proddesc, orddate, cost

From products_tbl p ,orders_tbl o

Where p.prodid=o.prodid

צילוף טבלה אל עצמה- self join

במשמעות from ניתן לאוთה טבלה פעמיים, כדי שהמחשב יתייחס אל ההפניה כשני טבלאות שונות, יש לתת להן כינוי שונה.

לדוגמא: לכל עובד יש קוד, שם וקוד מנהל שלו. כדי להציג את שם העובד, שם המנהל שלו (אשר שמו מופיע ברשימה העובדים ולא בטבלה נפרדת). יש לפתח את טבלת העובדים פעמיים, ולאחר מכן קוד עובד לקוד מנהל. פעם אחת עבור רשימת העובדים ופעם אחת עבור רשימת המנהלים.

```
Select e.empname as 'worker', m.empname as 'manager'  
From employee e left join employee m  
On e.managerid=m.empid
```

תרגיל: בטבלה אנשים מופיעים השדות: קוד איש, שם, קוד משפחה,מין. הטבלה מכילה רשימת בני זוג. הציגו אותם זה לצד זה

תרגילים צירוף 2 טבלאות:

1. הציג שט מוצר ותאריך הזמנה אחרון כולל מוצרים שאין להם הזמנות.
2. הציג שט לקוח תאריך הזמנה ראשון, תאריך הזמנה אחרון, ספירת הזמנות שבצע, כולל לקוחות שלא ביצעו הזמנות.
3. הציג קוד לקוח, תאריך הזמנה, והסכום לתשלום (כמויות*מחיר).
4. הציג קוד לקוח סכום כולל לתשלום, הנחה 5% ומהירות לאחר הנחה בשאיילתיה יוצגו רק הזמנות בכמות כוללת גדולה מ-50.
5. חשבי משכורות ממוצעת וסכום משכורות לכל עיר רק לאלו שקידומת הטלפון שלהם היא: '03'.
6. הציג שט מוצר וסכום כולל של הכמות שהזמנה ממנו, בתנאי שמחירו מעל 1 ל.
7. הציג מוצר וטפוי את כמות הזמנתו רק מליקות הגרים בירושלים.
8. הציג את הרוחת הכללי עפ"י ההזמנות.
9. סכמי את הסכום לתשלום לכל לקוח עבור כל הזמנות.
10. הציג קוד לקוח וממוצע של מחירי המוצרים שקנה, בתנאי שהמוצע גובהה מ-75.
11. הציג קוד לקוח שהפרש השנים בין תאריך הזמנה ראשון לאחרון קטן מ-5 שנים.

תרגילים איזוף ליזוף מ-2 טבלאות, וצירוף טבלה אל עצמה.

1. הציג שט לקוח, עיר מגוריו, שט העובד שטפל בלקוח, ועיר מגוריו של העובד, כאשר עיר לקוח ועיר העובד זהים.
2. הציג שט עובד, שט לקוחות, תאריך הזמנה משנת 2005.
3. הציג שט לקוחות, שט מוצר ותאריך קניה, ומהירות ההזמנה, מיini ע"פ לקוח ותאריך.
4. הציג שט עובד ושם המנהל שלו.
5. הציג מוצרים שומנו בכמות זהה ללקוחות שונים.
6. הציג עובד שקבע משכורת זהה גוגאריכים שונים.
7. הציג לקוחות שקבע בהפרש של 5 ימים.
8. הציג מוצרים שכמות המכירות שלהם עלתה משנה לשנה.

תשובות עמוד 45

שאילות מקווננות sub query

מתי משתמש בתת שאלה?

כאשר ברצינו ליצור שאלה עם תנאי מסוים שכדי לדעת את התנאי בלבד علينا ליצור שאלה

לדוגמא? מי מרוויח יותר מעובד מספר 7566

Select * from employee_tbl where ?

דוגמא לתת שאלה:

select * from employee_pay

where salary > (select salary from employee_pay where empid=7566)

כללים

(1) תת שאלה חייבת להיות תחומות. בסוגרים.

(2) ניתן לשלב תת שאלה אחריה (from) where,having:

(3) תת שאלה תחזיר עמודה אחת מאותו הסוג של העמודה בשאלתה הראשית

(4) משפט where של תת שאלה יכולה יכול להזכיר בעצמו תת שאלה נוספת.

(5) תת שאלה יכולה להכיל פניה לעמודות השיקוכות לטבלאות של השאלה הראשית, ולא רק לעמודות השיקוכות לטבלאות המופיעות במשפט from של תת השאלה.

(6) אי אפשר להשתמש בתת שאלה ב- order by בלבד במקרה שנשתמש ב-top

שאילתה המוחזרה ערך אחד

בשאילתה המוחזרה ערך אחד נستخدم אופרטורים =, <, >

לדוגמא, הציגי את כל העובדים שקבלו משכורת גבוהה ממשכורתו הממוצעת של

עובד: 123456789

Select empidFromemployee_pay_tbl

Where salary > (select avg(salary) from employee_pay_tbl e

where e.empid='123456789')

group by empid

שאילתה המוחזרה יותר מערך אחד

כאשר תת שאלה מוחזרה יותר מערך אחד נستخدم בין not,in,

In - שווה לאחד מן הערכים בתוצאה השאלה

Not in - לא שווה לכל הערכים בתוצאה השאלה

All - גדול מהמקסימום בתוצאה השאלה All < קטן מהמינימום בתוצאה השאלה

Any - גדול מהמינימום בתוצאה השאלה Any < קטן מהמקסימום בתוצאה השאלה

=Any - שווה לאחד מן הערכים בתוצאה השאלה (כמו in)

דוגמה: הציגי פרטי משכורות של עובדים שהביאו יותר משנה ללקוחות

```
Select * from employee_pay_tbl where empid in (select empid from customer_tbl
group by empid having count(1)>2)
```

דרך נוספת לפתרון תרגיל זה נראה בהמשך בנושא דתת שאילתות מדוימות

תרגילים תת שאילתות

1. הציגי את ההזמנות שבוצעו באחד מן התאריכים שהזמין לקוחות 20
2. הציגי את ההזמנות שהזמין בהם את אחד המוצרים שהזמין לקוחות 20
3. הציגי את שמות העובדים שקיבלו יותר מ加倍 מושכורת של עובד 011465986
4. הציגי עובד שקיבל משכורת קטנה ממשכורת ממוצעת כללית
5. הציגי עובד שקיבל יותר מכל משכורת ממוצעת של כל אחד מהמעדים.
6. סכמי לעובד את כל משכורותיו הציגי רק העובדים שסטטוסם קטן מהסכום הכללי של העובד משה.
7. הציגי לכל לקוח כמה ממוצעת שהזמין, וסכום כולל שהזמין הציגי רק לקוחות שהסכום גדול מ-20 והכמות הכוללת גדולה מ-100.
8. הציגי עובדים שתפקידם זהה לתפקיד של העובד יצחק
9. הציגי שם מוצר בתנאי שמחירו הוא הזול ביותר.
10. הציגי שם ללקוח ומחריר ממוצע למוצרים שקנה בתנאי שהמחיר הממוצע גבוהה מהמחיר הממוצע של הלוקו חיים לוי.
11. הציגי שם מוצר ומחריר עבור כל המוצרים שמחירים נוע בין 1 ל-6rup והיקף המכירות שלהם קטן מהיקף המכירות של מוצר בעל קוד 101 מינימית התוצאה מהמחיר הנמוך לגובה.

תשובות עמוד 47

הציגי שם עובד שמופיע יותר מפעם אחת בטבלה ניתן לפתור ללא תת שאילתת select lastNAME+' '+isnull(firstname, '') as fullname ,count(lastNAME+' '+isnull(firstname, '')) from dbo.EMPLOYEE_TBL group by lastNAME+' '+isnull(firstname, '') having count(lastNAME+' '+isnull(firstname, ''))>1

כאשר נרצה להציג בשאלתה גם תז' איז הקבוץ יהיה לפי שם ותו ואו יספר כפולים של שם ותו ולא שם בלבד, במקרה זה נצטרך להשתמש בנות שאילתת select lastNAME+' '+isnull(firstname, '') as fullname ,EMPID from dbo.EMPLOYEE_TBL where lastNAME+' '+isnull(firstname, '') in (select lastNAME+' '+isnull(firstname, '') from dbo.EMPLOYEE_TBL group by lastNAME+' '+isnull(firstname, '') having count(lastNAME+' '+isnull(firstname, ''))>1

חת שאלות מתואמות

חת שאלתה מתואמת היא חת שאלתה ה תליה במידע שבשאילתת הראשית.

חת השאלת מתבצעת כמספר הרשומות בשאלת החיזונית.

כיוון שלכל רשומה בשאלת החיזונית יתכן ותצא תוצאה אחרת בשאלת הפנימית...

דוגמא: הציגי פרטי משכורות של עובדים שהביאו יותר משני לקוחות

```
Select * from employee_pay_tbl e where  
(select count(1) from customer_tbl where empid=e.empid)>2
```

דוגמא: השאלת הבאה מ חוזרת לכל לקוח את ההזמנות שמחירם הסופי גדול מה ממוצע ללקוח זה.

```
Select c.custname, orders.*
```

```
From customers_tbl c inner join orders o on o.custid=c.custid inner join products  
on...
```

```
Where qty*cost>(select avg(qty*cost) from orders_tbl o1 inner join products on...
```

Where o.custid=o1.custid)

בדוגמא זו, חת השאלת הפנימית תלוי ב-custid שבשאילתת הראשית ולכל לקוח מתבצעת בדיקה האם עלות הזמנה גדול יותר מעלות הזמנה ממוצעת ללקוח זה...

תקמידה העיקרי של חת שאלת המשמשים בה על משפט select הוא להחזיר נתונים שבהם השאלת הראשית יכולה להשתמש לפטור את השאלת.

Exists

בדיקה האם בתוצאות השאלת קיימת לפחות רשומה אחת אם כן התוצאה היא true

אם תוצאות השאלת ריקה התוצאה היא false

exists-not לא קיים בדיקה הפוכה מ-

הרבה שאלות ניתן לפטור ע"י in או ע"י exists ולקבל את אותה תוצאה.

דוגמא: הציגי את רשימת לקוחות שיש להם הזמנות

```
select custname from customers where custid in  
(select custid from orders)
```

```
select custname from customers where exists  
(select 1 from orders where custid =customers.custid)
```

תרגילים:

1. הציגי פרטי הזמנות של המוצר היקר ביותר בלבד.
2. הציגי את המוצרים שלא הזמינו מהם בחודש ינואר 2008
3. הציגי עיר מגורים מטבלת עובדים רק כאשר עיר זו לא מופיעה בטבלת לקוחות.
4. הציגי את פרטי העובד שקבל משכורת מינימלית.
5. הציגי עובדים ששם זהה לשם אחד מן לקוחות
6. הציגי את שמות לקוחות שלא בצעו הזמנות בחודש האחרון.
7. הציגי את עיר המגורים ושם לקוח לכל לקוחות שהזמין לפחות אחד מה מוצרים שראובן או חיים הזמין. מין את התוצאה לפי שם עיר.
8. הציגי את רשימה המוצרים שהזמין מהם כמהות זהה לכמות שהזמן הל�� משא, או שהכמות זהה לאחד מהלפחות הגרים בבני ברק..
9. הציגי קוד לקוח מוצר וכמות רק להזמנות בכמות הגדולה מכמות ממוצעת מוצר זה.
10. הציגי שם לקוח, שם מוצר, תאריך הזמנה, כמהות ומהירות מוצר היקר ביותר שהזמן כל לקוח.
11. הציגי שם לקוח ומספר הזמנה לכמות הגדולה מ-5 ושם לקוח מופיע פעמי אחד בטבלת לקוחות, מין לפי הכמות בסדר יורד.
12. חשבי משכורת ממוצעת וסכום משכורות לכל עיר, אל תציג עיר שבה המוצע גובה ביותר או נמוך ביותר.

תשובות עמוד 48

13. הציגי עובד וסכום משכורתו הכללי לשנת 2006 בתנאי שהסכום גבוה מסכום משכורות של עובד זה לשנת 2007.
14. הציגי פרטי הזמנות של המוצרים שנמכרו יותר מ-3 פעמים.
15. הציגי שמות לקוחות שקבעו ב-2007 את אותו מוצר שקבעו בשנת 2008.
16. הציגי פרטי משכורת אחורונה של כל עובד
17. הציגי פרטי 2 המשכורות האחרונות לכל עובד

ציהוף טבלה אל שאלה

הציגי מוצר ושם הלקוח שקנה ממנו את הכמות הגדולה ביותר

ע"פ הנלמד עד עתה ניתן לפטור כך:

Select prodid,max(qty) from orders group by proid

באופן זה נקבל לכל מוצר את הכמות הגדולה ביותר ולא את שם הלקוח שבצע זאת

פתרון שגוי: Select prodid,custid,max(qty) from orders group by proid,custid

כיוון שהוא לא כולל רק את השם של המouser ולא את הלקוח

רף המקרים כיוון שהקיבוץ גם לפי לקוח

הפתרון הוא

```
select products.proddesc, customers.custname
from products left join orders
on orders.prodid= products.prodid
left join customers on customers.custid=orders.custid
inner join
(select prodid,max(qty)as maxqty from orders groupby proid)
as productsorders
on orders.prodid = productsorders.prodid and orders.qty=
productsorders.maxqty
```

תרגולים צירוף טבלה אל שאלה

1. הציגי עובד ממוצע משכורתו, משכורת נוכחת והפרש בין משכורת נוכחת למוצע.
2. הציגי לכל לקוח מחיר קניה ממוצע וההפרש בין מחיר קניה הנוכחי למוצע.
3. הציגי שם לקוחות ומוצרים למשתמש היKER ביותר לכל לקוח.
4. הציגי לכל שנה את המוצר שקנה ממנו בכמות הגדולה ביותר.

תשובות עמוד 50

שאלות מורכבות

מדוע ארצה אי פעם להשתמש בשאלתה מורכבת?

1. לשלב ולהגביל את התוצאות משנה משפט select.

2. להחזיר או להעלים את הפלט של רשומות כפולות.

3. להביא ייחדיו נתונים דומים המאוחסנים בשדות שונים.

לעתים קרובות קל יותר לכתוב שאלות מורכבות מאשר שאלתה בודדת עם תנאים מסוימים.

האופרטורים של שאלות מורכבות משתנים בין היצרנים של מסדי נתונים.

תקן ANSI כולל את האופרטורים הבאים: except, intersect, union, union all

האופרטור union

משתמשים באופרטור union union כדי לאחד תוצאות של כמה משפטי select לטבלה תוצאות אחת.
כללים:

1) השאלות צריכה להיות זהות מבחינת מספר העמודות. הכוורות נקבעות לפי משפט select הראשון.

2) השאלות צריכה להיות זהות מבחינת טיפוס הנתונים של העמודות.

3) השאלות צריכה להיות זהות מבחינת טיפוס הנתונים של העמודות.

4) ברירת מחדל של אופרטור האיחוד היא סילוק שורות כפולות. נשים לב שבירית המחדל זו היא הפוכה לברירת המחדל של הפקודה select, שאינה מסלקת שורות כפולות, אלא אם מבקשים זאת במפורש באמצעות ההוראה distinct.

5) אין מגבלה למספר השאלות שմבקשים לאחד.

6) Union all מציג את כל השורות כולל כפולים.

7) השימוש ב- Order by בשאלתה מורכבת יהיה רק במשפט select האחרון. במילון אפשר להשתמש בשם העמודה אם לעמודה שלפיה ממיינים יש את אותו שם בכל השאלות האינדיבידואליות של המשפט.

8) השימוש ב- group by בשאלתה מורכבת אפשרי בכל משפט select של שאלתה מורכבת, אבל אפשר להשתמש בו גם אחרי כל השאלות האינדיבידואליות. בנוסף, אפשר להשתמש ב- having (שימושים בו לעיתים עם group by) בכל אחד ממשפטים select במספט מורכב.

Select column1, [column2]

תחביר:

From table1,[table2] [where] [group by] [having]

Operator{union:except:intersect}

Select column1,[column2]

From table1,[table2] [where] [group by] [having]

[order by]

```
Select emp_id from
employee_tbl
Union
Select emp_id from employee_pay_tbl
```

דוגמא:

ניתוח: אוטם מספרי עובד הנמצאים בשתי הטבלאות גם יחד מופיעים רק פעם אחת בתוצאות.

האופרטור union all

משתמשים באופרטור union all כדי לשלב את התוצאות של שני משפטים select כולל רשומות כפולות. אוטם הכללים החלים על union all חלים גם על union.

האופרטור intersect - חיתוך

משתמשים באופרטור intersect כדי לשלב שני משפטי select, אבל הוא מחזיר רק שורה אחת select הראשון שהן זהות לשורה במשפט select השני. בדיקן כמו לגבי union, אוטם הכללים החלים כשמשתמשים באופרטור intersect

דוגמא: Select custid from customers_tbl

Intersect

Select custid from orders_tbl

ניתוח: משפט sql הקודם מחזיר את מספר הלקוח עבור אוטם הלוקחות שבוצעו הזמנה.

האופרטור except

משלב שני משפטי 'select' וממחזר שורות המשפט select הראשון שאינן מוחזרות ע"י המשפט השני. אוטם הכללים החלים על השימוש באופרטור union all חלים גם על האופרטור .except

תרגילים:

1. הציג עיר מטבלת עובדים שלא מופיעה בטלת לקוחות.
2. הציג ערים שיש בהם גם עובדים וגם לקוחות.
3. הציג רשימה ערים כוללת של עובדים ולקוחות ממונת בסדר עולה.
4. הציג רשימה ממונת בסדר עולה של שמות העובדים והלקוחות, עיר מגורים ומס' טלפון.
5. הציג עובדים שקיבלו משכורת בחודש ינואר 2008 וגם בחודש פברואר 2008
6. הציג מוצרים שהוזמנו ע"י הלוקחות בשנת 2008 ולא הוזמנו בשנת 2007
7. הציג רשימת תאריכי הזמנות, תאריכי לידה של עובדים, ותאריכי משכורות יחד בטבלה אחת.

תשובות עמוד 50

-Manipulating data תפעול נתונים-

הכנסת נתונים אל תוך טבלה - *Insert*

תחביר:
insert table-name

Values ('value1','value2',[null])

כאשר משתמשים בתחביר זהה של משפט *insert*, חייבים לכלול בראשימה *values* כל עמודה בטבלה המצוינה. שימי לב שכל ערך בראשימה מופרד באמצעות פסיק. הערכים המוכנסים אל תוך הטבלה חייבים להיות בתחוםים בתווים גרש יחיד עבור טיפוסי הנתוניםתו ותאריך. תווים גרש אינם נדרשים לטיפוסי נתונים מספריים או לערכים מסווג *null*.

לדוגמא הוספה מוצר חדש לטבלה מוצרים:

```
Insert products_tbl  
values('7725','computer',2224.99)
```

פלט:

בהוספה נתונים לכל העמודות בטבלה לפי הסדר, אין צורך לפרט את שמות העמודות אחרי שם הטבלה, אך בהוספה נתונים רק לחלק מהעמודות בטבלה, או בשינוי הסדר, יש צורך לפרט את שמות העמודות אליהן יכנסו הנתונים אחורי שם הטבלה, כפי הדוגמא הבאה:

תחביר:
insert table_name (column1,column2)

Values ('value1','value2')

insert orders_tbl(ordnum,custid,prodid,qty)

values(2316,109,'7725',2)

הערה: כאשר עמודה מוגדרת כמספר אוטומטי, מוגדרת עליה, לא רושמים אותה בראשימת העמודות, ולא מוסיפים לה ערך.
(בשאילתת הוספה אין היחסות לעמודה מספר אוטומטי, מעריכים ממנה Caino אינה קיימת)
לדוגמא: אם העמודה *ordnum* הוגדרה כמספר אוטומטי נרשם זאת כך:

insert orders_tbl(custid,prodid,qty)

values(109,'7725',2)

ג. הוספה מספר שורות לטבלה

פקודה זו מאפשרת הוספה מספר שורות לטבלה תוך שימוש בשאילתת שלילת השורות מטבלה אחרת.

- 1) השאילתת שלילת השורות אינה יכולה להכיל *order by*
- 2) חייבת להיות זהות בין מספר העמודות המופיעות במשפט *select* לבין מספר העמודות של טבלת היעד.

תחביר:

insert into table-name

select * from table-name

```
insert products_tmpselect * from products_tbl
```

דוגמא:

דוגמא של משפט insert עם תח שאלתה:

```
Insert rich_employees
```

```
Select e.emp_id,e.lastname,e.firstname,ep.salary
```

```
From employee_tbl e inner join employee_pay_tbl ep on e.emp_id=ep.empid
```

```
Where ep.salary>
```

```
(select salary from employee_pay_tbl
```

```
Where empid='313782439')
```

המשפט insert מכניס את העמודות emp_id,lastname,firstname,salary בטבלה בשם rich_employees עבור כל הרשומות של עובדים ששיעור השכר שלהם גבוה משיעור השכר של עובד שמספרו הוא: 313782439.

- השאלתה לשילוף השורות אינה יכולה להכיל order by

- חייבת להיות זהות בין מספר העמודות המופיעות במשפט select לבין מספר העמודות של טבלת היעד.

עדכון נתונים קיימים – update

```
Update table-name
```

תחביר:

```
Set column1='value',
```

```
[column2='value',]
```

```
[column3='value']
```

```
[where condition]
```

```
Update orders_tbl
```

קלט:

```
Set Custid=221 ,qty=1
```

```
Where ordnum=2316
```

```
1 rows updated
```

פלט:

פסיק משמש כדי להפריד בין שתי העמודות המעודכנות, Where הוא אופציוני אבל בדרך כלל נכון.

הערה: משתמשים במילת המפתח set רק פעם אחת עבור כל משפט update. אם יש צורך לעדכן יותר עמודה אחת, משתמשים בפסיק כדי להפריד את העמודות שיש לעדכן.

```
Update employee_pay_tbl
```

דוגמא לתח שאלתה :

```
Set salary=salary*1.1
```

```
Where empid in
```

```
(select empid from employee_tbl Where city='jerusalem')
```

מחיקת נתונים מטבלאות delete

Delete from

תחביר:

table name

[where condition]

Delete from orders_tbl

קלט:

Where ordnum=2316

פלט:

1 rows deleted

ازהרה: אם משמשים את where מהמשפט delete, אז כל שורות הנתונים תמחקנה מהטבלה!
כלל,

השתמשי תמיד ב- where עם המשפט delete.

delete from employee_pay_tbl

דוגמא לחת שאלתה עם המשפט delete:

Where empid not in

(select emp_id from customer_tbl)

תרגול שאלות פעולה

1. הוספי שורה לטבלה מוצרים הכליני ערכיהם לכל העמודות.
2. הוספי מוצר חדש לטבלה מוצרים, הכליני קוד מוצר, שם מוצר ומחיר בלבד.
3. עדכני את מחירי המוצרים והוספי להם 10%.
4. מחקי את כל ההזמנות שתאריך ההזמנה שלהם גדול מהיום.
5. מחקי את כל המשכורות לפני יותר מ-10 שנים וסכום קטן מ-5000.
6. הוספי שורה חדשה לטבלה הזמנות: קוד לקוח כלשהו, מוצר 109, השדה כמוות יתמלא בערך NULL, השדה תאריך יהיה תאריך הנוכחי. שמי לב שקוד הלוקה שבחרת מופיע בטבלה לקוחות.
7. צרי טבלה חדשה בשם oldproduct (חומרים ישנים) בטבלה יופיעו השדות קוד מוצר, שם מוצר, מחיר, כמוות כוללת שנמכרה ותאריך מכירה אחרון.
8. צרי שאלת הוספה שתוסיף לטבלה מוצרים ישנים את כל המוצרים שלא נמכרו השנה האחרונות.
9. הוספי את שמות העובדים מירושלים לטבלה לקוחות.
10. עדכני בטבלה עובדים את השם הפרטי לשם הפרטי + שם משפחה.
11. עדכני את השדה תפקיד בטבלה עובדים לערך 10 רק לעובדים שסכום ההכנסה הכוללת שלהם בשנה הקודמת הייתה מעל 10000 ₪.
12. עדכני בטבלה לקוחות את השדה קוד עובד המשוויק ללקוח. קוד עובד שגר בעיר מגוריו של הלקוח, מכיוון שיתכן שקיים כמה עובדים הגרים בעיר מגוריו של הלקוח. עדכני לעובד שמספרו הנמוך יותר.
13. מחקי מטבלת משכורות את כל השורות שהשדה משכורת ריק.

14. מחקי מטבלת עובדים את כל העובדים ללא משכורת.

15. מחקי מטבלת הזמנות את כל הזמנות של המוצרים שלא נמכרו שנה האחרונה..

תשובות עמוד 51

Update דוגמאות נפוצות

set

הת שאלתה ב-

לדוגמא: עדכן לכל לקוחות את עובד שמטפל בו על פי עיר מגורים זהה של הלוקו והעובד. מכיוון שיתכנו כמה עובדים
באותו עיר מגורים בוצע `top 1` כולם: הראשון מביניהם

```
update dbo.CUSTOMERS_TBL
    set EMPID=(select top 1 EMPID from dbo.EMPLOYEE_TBL where
        CITY=dbo.CUSTOMERS_TBL.CUSTCITY)
```

עדכן את מחיר המוצר למחיר בתוספת 1% מס' הכמות שנמכרה מאותו מוצר

```
update dbo.products_tbl
    set cost=cost+(select isnull(sum(QTY),0)*0.01 from dbo.ORDERS_TBL
        where prodid=dbo.products_tbl.prodid)
```

עדכן את הבונוס לעובד לפי הכמות של הזמנות שביצעו לקחוותיו

```
update dbo.EMPLOYEE_PAY_TBL
    set bonus=(select count(*)
        from dbo.ORDERS_TBL innerjoin dbo.CUSTOMERS_TBL
        on dbo.CUSTOMERS_TBL.custid=dbo.ORDERS_TBL.custid
            where EMPID=dbo.EMPLOYEE_PAY_TBL.EMPID)
```

join update

```
update dbo.EMPLOYEE_TBL
    set dbo.EMPLOYEE_TBL.ZIP=city.zip
        from dbo.EMPLOYEE_TBL innerjoin dbo.city
            on city.city=dbo.EMPLOYEE_TBL.city
```

– הגדרת נתונים – data definition language DDL**מספרים מדויקים**

טווח הערכים	סוג הנתונים
9,223,372,036,854,775,807 – עד 9,223,372,036,854,775,808	Bigint
2,147,483,647 – עד 2,147,483,648	Int
32,767 – עד 32,768	Smallint
0 עד 255	Tinyint
0 עד 1 (0 – 1 בוליאני)	Bit
10^38 – 1 עד 10^38 + 1	Decim
10^38 – 1 עד 10^38 + 1	Numeric
922,337,203,685,477.5807 – עד 922,337,203,685,477.5808	Money
214,748.3647 – עד 214,748.3648	Smallmoney

מספרים בקירוב

טווח הערכים	סוג הנתונים
E + 3081.79 – עד E + 3081.79	Float
38 + E3.40 – עד E + 383.40	Real

שעות ותאריךיכים

טווח הערכים	סוג הנתונים
Dec 31, 9999 עד Jan 1, 1753	Datetime
2079, Jun 6 עד Jan 1, 1900	Smalldatetime

מחרוזות תווים

תיאור הנתונים	סוג הנתונים
.non-Unicode עד 8000 תווים מסווג Char	Char
.non-Unicode עד 8000 תווים מסווג Varchar	Varchar
.non-Unicode עד 2^31 תווים מסווג non-unicode. רק ב-2005 varchar(max)	varchar(max)
.non-Unicode עד 2,147,483,647 תווים מסווג Text	Text

מחרוזות תווים - Unicode

תיאור הנתונים	סוג הנתונים
.Unicode עד 4000 תווים מסווג Nchar	Nchar
.Unicode עד 4000 תווים מסווג Nvarchar	Nvarchar
.unicode עד 2^30 תווים מסווג Unicode. רק ב-2005 nvarchar(max)	nvarchar(max)
.unicode עד 1,073,741,823 תווים מסווג Ntext	Ntext

create table

כאשר אתה עומד ליצור טבלה, عليك להגדיר:

- عمودה (או عمودות) שיהו מפתח הראשי. عمودות אלו לא יכולות להכיל ערך NULL
- לכל عمودה יש להגדיר טיפוס נתונים מסוימת, כמו מרזביים, תאריך.

Create table table_name

תחביר:

(

Field1 datatype [not null],
Field2 datatype [not null],
Field3 datatype [not null],
Field4 datatype [not null]

)

לדוגמה: create table employee_tbl:

(empid char(9) primary key,
Emp_name varchar(40) not null,
address varchar(20),
datehire datetime default getdate(),
position int default 1,
Emp_phone char(10) null)

המידע שמייחס לכל عمودה מופרד באמצעות פסיק, וסוגרים מקיפים את כל העמודות.
(סוגר שמאלית לפני העמודה הראשונה וסוגר ימני לאחר המידע שמייחס לעמודה האחרונה).

הערה: הערך null נקבע עבור عمودה כברירת מחדל. לכן, אין צורך להכניס אותו במשפט

create table.

default

ניתן לקבוע ערך בויירת מחדל לעמודה מסוימת, הערך ירשם בשורת ההגדירה של העמודה

לדוגמה: emp_city varchar(15) default 'jerusalem'.

מספר אווטומטי – identity

יש אפשרות להגדיר שדה מסוים כמספר אווטומטי.

ע"י שרשום: identity(seed,increment)

לדוגמה: payid int identity(1,1)

הערך ההתחלתי הוא: 1 וערך זה מתקדם בתוספת 1.

יצירת טבלה מטבלה קיימת.

תחביר:

Select [*/column1, column2] into שם טבלה חדשה

From table_name

[where]

Select * into products_tmp :from products_tbl

הערה: בעית יצירה טבלה מטבלה קיימת, הטבלה החדשה מקבלת את אותם מאפייני האחסון כמו אלו של הטבלה המקורי.

תרגילים:

צרי מסד נתונים לתוך דירות הכלול את הטבלאות הבאות:

1. לקוחות: ת.ז., שם ללקוח(קונה/utzer/שוכר/مشכיר), שם, כתובת, טלפון.
2. דירות: קוד דירה, רחוב, מס' בית, עיר, סוג(מכירה/שכרה), ת.ז בעל הדירה, מס' חדרים, מחיר
3. מכירות: קוד דירה, תאריך מכירה, ת.ז. קונה, מחיר.
4. השכירות: קוד דירה, תאריך התחלת השכירה, תאריך סיום השכירה, ת.ז שוכר, מחיר.
5. תשלום: קוד תשלום, ת.ז. לקוח, קוד דירה, תאריך, סכום

תרגיל חזקה שאלות מסד נתונים תיווך דירות:

1. הציגי את פרטי הדירות שאין מקורות ואינו מושכרות. כולל לא מופיעות בטבלה מכירות, ולא מופיעות בטבלה שכירות.
2. הציגי נתונים בטבלה השכירות, וכן עמודות חישוב ההפרש בין תאריך התחלת השכירה ותאריך סיום השכירה בחודשים.
3. הציגי לכל לקוח את ס"כ התשלומים שלו עבור כל דירה, ובנוסף הציגי מחיר דירה בטבלה דירות, וההפרש בין מחיר הדירה לתשלומים שהמוחה את היתרה.
4. הציגי מחיר ממוצע לפי עיר ומס' חדרים רק לדירות מסווג מכירה, ומהמוצע גדול מ- 1000000
5. הציגי רשימה ללקוחות שקבעו דירות (מופיעים במכירות), יחד עם רשימה הלקוחות ששכרו דירות
6. הציגי מחיר ממוצע לדירה בעיר X כאשר הוא זהה למחיר ממוצע לדירה בעיר Y.
7. מחקי את הלקוחות שלא בצעו מכירות ולא שכירות וגם אינם בעלי דירות.
8. הציגי דירות שנמכרו בשנתיים האחרונים ולא שולם עבורים כלל.
9. הציגי ערים שבהם כמהן הדירות להשכירה גדול מכמות הדירות למכירה
10. עדכני את הלקוחות לטטוטס קונה כאשר הם נמצאים בטבלה מכירות. וכן לשאר הסטטוסים

שינוי המדרת הטבלה alter table

הפקודה alter מאפשרת 3 שינויים בהגדרות הטבלה והם:

1. add – להוסיף עמודה/ות.
2. drop column – למחוק עמודה/ות.
3. modify – לשנות אופי של עמודה.

א. טיפוס הנתונים של העמודה. ב. האורך, הדיווק, ג. האם העמודה יכולה להכיל ערכים מטיפוס null.

תחביר: alter table table_name

[drop column:add:modify] [column_name] [datatype: null/not null]

לדוגמא: Alter table employee_tbl

add name_fathervarchar(10)

תוצאה: הוספה העמודה שם האב לatable עובדים.

דוגמא נוספת: Alter table employee_tbl

drop column state

תוצאה: מחיקת העמודה מדינה מatable עובדים.

תרגילים בסיס נתונים תוך דירות

1. הוספי לטבלה תשלומים שדה תאריך הפקדה.

2. הגדילי את האורך המרבי של השדה כתובת בטבלה לקוחות.

3. מחקי מטבלת לקוחות שדה מס' בית. לפני המחקה בצעי עדכן לרוחב שיכלול גם את מס' הבית.

ביצוע truncate לטבלה

פעולה זו מנקה את הטבלה מכל התוכן שלה ומאפשרת מספר אוטומטי, היא מהירה יותר מ delete וניתן להפעילה רק על כל הטבלה יחד.

לדוגמא" truncate table products

מחיקת טבלאות – drop

תחביר: Drop table table_name [restrict/cascade]

לדוגמא: Drop table products_tmp

תרגילים:

1. מחקי את כל הטבלאות שיצרת: דירות, לקוחות, מכירות, שכבות, תשלומים.

2. צרי אותם שוב ע"י הפעלת משפט create.

הוספת אילוצים לטבלה - constraint

(1) אילוצים על המפתח הראשי

א) לרשום ליד העמודה primary key

ב) בסוף משפט create table (צור טבלה), לאחר רשימת הגדרות לעמודות, לפני הסוגרים

האחרון נרשם: primary key(emp_id)

כמפתח לדוגמא: primary key(emp_id, date)

(2) אילוצי ייחודיות

נרשום את המילה unique: בתוך המשפט create table בסוף השורה של העמודה שתהייה ייחודית.

לדוגמה: emp_phone integer null unique,

אילוצי ייחודיות לעמודה בטבלה דומה למפתח ראשי פרט לכך שהמפתח ראשי משמש לייצירת סדר נתונים שבטבלה, ובאותו הקשור, לחיבור טבלאות הקשורות.

(3) אילוצי מפתח זר

מפתח זר-עמודה בטבלה צאצא (child table) שווה לעמודה בטבלה ההורה (parent table)

העמודה empid בדוגמה הבאה נקבעה כמפתח זר

נרשום בסוף משפט create table לפניו הסוגרים האחרונים, או בתוך משפט create table אחריו הפסיק של העמודה שהינה המפתח הזר.

constraint emp_id_fk foreign key (empid) references employee_tbl(emp_id),
כאשר קיימים כמה מפתחות זרים בטבלה נרשם לכל מפתח זר את ההגדרה הניל ונפריד ע"י פסיקיםעמן להכניס ערך בטבלה הצאצא צריך שהיה ערך בהורה
עמן לשלק ערך מההוראה יש לשלק קודם קודם מטבלת הצאצא.

(4) אילוצי בדיקה - check

בדיקות חוקיות נתונים המוכנסים לעמודות מסוימות בטבלה.

א) לדוגמה: טבלה employee_tbl

הגדרת מרחב ערכים, העמודה המוגדרת יכולה להכיל רק את הערכים a,b,c

Check (column_name in('a','b','c'))

ב) הגדרת טווח ערכים בין 20 ל-120

check (pay_rate between 20 and 120)

דוגמאות להוספה אילוצים לטבלה Constraint

1. הוספה אילוץ של בדיקת חד ערכיות לעמודה tz

Alter table employee_tbl add constraint unique(tz)

2. להוסיף ערך ברירת מחדל 'B' לעמודה position

Alter table employee_pay_tbl add constraint [df_position] default 'b' for position

3. הוספה אילוץ בדיקה לשדה תאריך לידה קטן מתאריך הנוכחי

**ALTER TABLE [dbo].[Employees] WITH NOCHECK ADD CONSTRAINT
[CK_Birthdate] CHECK (([BirthDate]<getdate()))**

4. הוספה אילוץ מפתח עיקרי לטבלת Products

Alter table products add primary key prodid

5. לבטל מפתח עיקרי של טבלת products

Alter table products drop primary key**תרגילים:**

(1) אילוצי מפתח ראשי:

(א) בטבלת דירות: קוד דירה,

(ב) בטבלת לקוחות: TZ,

(ג) בטבלת מכירות: קוד דירה ותאריך מכירה,

(ד) בטבלת השכירות: קוד דירה ותאריך התחלה השכירה,

(ה) בטבלת תשלוםיים קוד תשלום.

(2) אילוצי ייחודיות: בטבלת לקוחות טלפון

(3) אילוצי מפתח זר:

(א) בטבלת מכירות: TZ. קונה,

(ב) בטבלת דירות: TZ. בעל הדירה,

(ג) בטבלת תשלוםיים TZ. לקוחות,

(ד) בטבלת השכירות TZ. משכיר

(4) אילוצי בדיקה

1. בטבלת דירות הוסיף אילוץ בדיקה לשדה סוג שווייה שווה להשכרה או מכירה בלבד.

2. הוסיף אילוץ לטבלת הזמן הבזוק שערך השדה כמות גדול מ-0

3. הוסיף אילוץ לטבלת מוצרים לבדיקה שערך השדה מחיר גדול מ-0

4. הוסיף לטבלת מוצרים שדה חדש כמות בהזמנה

5. הוסיף אילוץ לטבלת מוצרים לשדה מחיר ברירת מחדל 0

6. הוסיף אילוץ ברירת מחדל 1 לשדה כמות בטבלת הזמן

תשובות עמוד 52

אינדקסים

מהו אינדקס?

מצביינו נתונים בטבלה, דומה מודגש לאינדקס בסוף ספר. אינדקס בספר מפנה מספר עמוד, אינדקס במסד נתונים מפנה למקום הפיזי המדויק של נתונים בטבלה. כשם שסדרת אינדקס של ספר מהירה יותר מדף דף אחריו דף, כך שימוש באינדקס מהיר יותר מסריקה טבלה מלאה. תפקידו העיקרי של אינדקס הוא שיפור הביצועים. חסרון: מגביל database בכך שמאוחסן באופן פיזי נפרד מהטבלה.

كيف אינדקסים פועלים?

כאשר אינדקס נוצר, הוא רושם את המיקום של ערכיהם בטבלה המשווים לעמודה עלייה נוצר האינדקס. נתונים חדשים מתווספים לטבלה, מתווספים גם לאינדקס. כאשר מתבצעה שאלתה ומצוין ב-where עמודה עם אינדקס. האינדקס נפרק תחילה בחיפוש אחר הערכים המצוינים בטבלה. הערך שנמצא באינדקס מחזיר את המיקום המדויק של הנתונים המבוקשים בטבלה.

	index		Table	
	data	location	Location	data
glass	6		1	Smith
jones	2		2	Jones
jones	9		3	Smith
plew	5		4	Wiliams
smith	1		5	Plew
smith	3		6	Glass
smith	7		7	Smith
smith	100000		8	Wallace
wallace	8		9	Jones
williams	4		100000	Smith

נניח שהופעלה השאלתה הבאה:

```
Select *
From table_name
Where name='smith'
```

הפקודה X

Create [unique] index index_name on table_name (column1,column2)

ביצועי השאלתה משתפרים כשהשניים את העמודה בעלת הערכים המגבילים ביותר ראשונה.

מחיקת index

Drop index table_name.index_name

אינדקס ייחודי – אינדקס אשר אינו מאפשר הכנסת ערכים כפולים לטבלה. האינדקס הייחודי פועל גם Caindex רגיל.

אינדקס מורכב – אינדקס על 2 عمודות או יותר של טבלה, רצוי לכתוב בהתחלה את העמודה שתגביל יותר את התוצאות.

מתי נמנע מאינדקסים ?

- 1) אין להשתמש באינדקסים על טבלאות קטנות.
- 2) אין להשתמש באינדקסים על عمודות המחזירות אחוז גבוה של שורות נתונים שהן שימושות בתנאי סינון בפקודה where של שאלתה.
- 3) אין להשתמש באינדקסים על عمודות המכילות מספר גבוה של ערכי NULL.
- 4) عمודות המתופעלות לעיתים תכופות אין ליצור עליהם אינדקסים.

הרשאות גישה לבסיס נתונים

Grant – הענקת הרשאות גישה

Revoke – ביטול הרשאות גישה

בדרכ' מנצל מסד הנתונים מפעיל פקדות אלו.

Select – הרשאה לשיליפה והציגנת נתונים

Update – הרשאה לעדכן בסיס הנתונים

Insert – הרשאה להוסיף לבסיס הנתונים

All – הושאה לכל הפעולות

Public – הרשאה לכל המשתמשים

לדוגמא:2
grant select,insert on employee_tbl to user1,user2
平淡：

grant succeeded

Grant option – הענקת זכות לנינת הרשאות לאחרים.

דוגמא: grant select on employee_tbl to user1 with grant option

דוגמא ל-revoke
revoke insert on employee_tbl from user1:

דוגמא ל-public
grant select on employee_tbl to public :

טבלה זמנית

טבלה זמנית היא טבלה הנשמרת בזיכרון בזמן ריצה ולאחר מכן נמחקת.

נשתמש בטבלה זמנית כדי להעביר אליה נתונים מתוך שאילתת מסוימת בזמן ריצה לצורך ביצוע פעולות נוספות או מורכבות על השאלתה

לדוגמא:....
select * into #t from stores where

בשפה T-SQL הנלמד בהמשך החומר, נראה כי ניתן להגדיר משתנה מסווג table המוגדר כ@t ומומלץ יותר להשתמש בטבלה המוגדרת כמשתנה מאשר בטבלה זמנית המוגדרת ע"י #.

— טבלאות מדומות views

טבלה מדומה היא נראית כטבלה רגילה וניתן לשולףמנה נתונים ע"י שאלות כטבלה רגילה. אך בפועל היא אינה דורשת שטח אחסון פיזי, כיון שרק ההגדרה שלה נשמרת בסיס הנתונים. הטבלה המדומה نبנית רק כאשר שאלת sql מכילה התייחסות אליה במשפט from. כל פניה אל הטבלה המדומה יכולה ליצור טבלה מדומה שונה, בהתאם לתווך הרגעי של טבלאות הבסיס מהן נגורים הנתונים. בסיס הנתונים מאחסן את נתוני טבלאות הבסיס ורק את ההגדרות של הטבלאות המדומות. מנוקדת מבט של המשמש טבלה מדומה נראית בדיק כמו טבלה אמיתית והוא יכול לבצע עליה שאלות שונות, באופן דומה לצורת בניית שאלות על טבלאות הבסיס.

יתרונות הטבלה המדומה

אבטחת מידע

נניח שיש לך טבלה בשם employee_tbl הכוללת שמות עובדים, כתובות, מספרי טלפון, כתובות קשור ל مكانة חרום, محلקה, משרה ומשכורת או שייעור שכיר. יש לך כמה עובדים זמינים שבאו לכתוב כמה דוחות, את זוקה לדוח של רשות עובדים, כתובות ומספר טלפון. אם תיתני גישה לטבלה employee_tbl לעובדים הזמינים, הם יכולים לראות כמה מרווחם כל אחד מהעובדים שלו. את לא רוצה זהה יקרה כדי למנוע זאת צורת הצוגה המכילה רק את המידע הנוכחי: שם עובד, כתובות, טלפון וכתובת קשור ל مكانה חירותם.

נוחות ופשטות

נניח שיש לך טבלה המכילה מידע על אנשים, כגון: עיר המגורים שלהם, מינם משוכратם וגילם את יכולה ליצור צוגה המבוססת על הטבלה והציגה מספרים מסוימים לאנשים לכל עיר, כגון גיל מוצע, משכורת ממוצעת, סך כל הגברים וסך כל הנשים. אחרי שהצוגה נוצרה כדי לאחזר את המידע הזה מהטבלה את יכולה פשוט לسؤال את הצוגה במקום לחבר משפט select העשו להיות מסוובן.

שיפור אי תלות בנתונים

לעתים יש צורך לבצע שינויים בטבלאות הריגולות של בסיס הנתונים. משתמש הניגש אל טבלאות המקור דרך טבלאות מדומות יכול להמשיך ולבנות את השאלות שלו כרגיל, למروת שהטבלאות השתנו.

יעילות

הטבלה המדומה אינה תופסת מקום פיזי על הדיסק, כי רק ההגדרה שלה נשמרת במערכת.

חסרונות הטבלה המדומה

רמת עדכניות מוגבלת

עדכון דוד טבלה מדומה לא תמיד אפשרי, כדי שטבלה מדומה תהיה בת עדכון היא צריכה להיות ללא פונקציות, תת שאלתה, להכיל את כל העמודות מהם null not ועוד.

ביצועים

יש מצבים בהם הגדרות הטבלה המדומה מורכבת מאוד. וכיון שכל גישה אל הטבלה המדומה מחייבת הידור מחדש של השאלה יתכן שתהייה אטיות בرمת הביצועים לעומת השימוש בטבלאות המקור.

הגדרת טבלה מדומה

create view [שם הטבלה המדומה] as select...

אם לא נרשום את שמות העמודות בטבלה המדומה. אז הם יהיו זמינים לשימוש לעמודות בטבלת הבסיס.

דוגמא:

```
createview ord (custid,sum_qty)as
select custid,sum(QTY)
from dbo.ORDERS_TBL
groupby CUSTID
```

השאילתת בונה טבלה מדומה המכילה סכום כמות הזמנות לכל לקוח.

אם נרצה לדעת מהו הממוצע של הכמות נבצע השאילתת לחישוב הממוצע על הטבלה המדומה.

```
select avg(sum_qty)
from ord
```

דוגמא נוספת:

```
createview city_pay (city,avg_salary)as
select e.city,avg(p.salary) avg_pay
from employee_tbl e inner join employee_pay_tbl p on e.empid=p.empid
groupby e.city
```

עתה ניתן לבחור מתוך התצוגה המוטאמת:

```
select * from city_pay
```

באמצעות סיכון תצוגה, בחירות שעשויה להופיע בעtid אין מופשטות מול טבלת התשתית של התצוגה. ניתן להשתמש ב-view order by, רק כאשר משתמשים ב-select top. וכן אי אפשר לבצע איחוד בטבלאות המזומות שמי לב להגדיר שם לכל העמודות בשאילתת ליצירת טבלה מזוינה אחרת תיווצר שגיאה, שמי לב בעיקר לעמודות מחושבות או צבויות עלי פונקציה צבירה, ועל תשכחי להגדיר להם שם

מחיקת טבלה מדומה

Drop view [שם הטבלה מדומה]

הפרמטר cascade גורם לביטול כל האובייקטים המתיחסים לטבלה מדומה זו.

למשל, כל הטבלאות המדומות המתבססות על טבלה זו גם תתבטלנה

הפרמטר restrict הוא ברירת מחדל ומונע ביטול הגדרת טבלה מדומה אם מבוססת עליה טבלאות מדומות אחרות.

תרגילים:

1) צרי טבלה מדומה המכילה את כל נתוני הדירות המכירות והשכרות כולל דירות ללא מכירות וככל דירות ללא השכרות.

2) צרי טבלה מדומה המציגה לכל עיר את כמות הדירות שנמכרו בפועל ומהירות ממוצע לעיר, וכמות הדירות שהושכרו בפועל ומהירות ממוצע להשכרה. השתמשי בטבלה המדומה

בשאילתת המציג את כמות הדיירות למכירה במלאי לעומת כמות הדיירות שנמכרו וההפרש ביןיהם, וכן בדיירות להשכלה.

- (3) צרי טבלה מדומה על טבלת דיירות חדש, שנה, עיר ומחייר ממוצע לדירה.
- (4) צרי טבלה מדומה שתכלול את פרטי הדיירה מטבלת דיירות, ומהירות הדיירה מטבלת מכירות וההפרש ביןיהם, צרי שאלה על טבלה מדומה זו המציגת ממוצע של ההפרש בין המהירים.
- (5) צרי טבלה מדומה לדיירות שתאריך ההשכרה שלהם עומד להסתיים בחודש הקרוב, יחד עם דיירות שמוצעות להשכלה ועדין לא הושכרו.

	pupil
pupilid	קוד תלמיד
lastname	שם משפחה
firstname	שם פרטי
adress	כתובת
phone	טלפון
lastyear	שנת לימודים אחרונה
namefather	שם אב
namemother	שם אם
Class	כיתה
city	עיר
numsyster	מספר אחיוות
dateborn	תאריך לידה

נספח א תרגילים

payment

Tashhid	קוד תשלום
Pupilid	קוד תלמיד
Payment	סכום
Bank	בנק
Snif	סניף
Cheshbon	חשבון
numbershek	מספר שק
datepay	תאריך תשלום
Kind_pay	אופן תשלום
Status	סטטוס
datehaskada	תאריך הפקדה

bankim

Bankid	קוד בנק
Bankname	שם בנק

שאלות פשוטות ב-sql

1. הציגי את התשלומים שתאריך התשלום בין 20/08/09 ל-20/10/08.
2. הציגי קוד תלמיד מطالبת התשלומים שלו ליב כל קוד תלמיד יופיע פעם אחת בלבד.
3. הציגי את התשלומים שהשدة סוג תשלום הוא: מזומנים או שק והם מבנק 20.
4. הציגי את פרטי התשלומים, וחשבוב של הסכום בתוספת 5 טן הציגי רק התשלומים בין 150 ל-250.
5. הציגי את התאריכים שבהם התבצעו הפקודות, כל תאריך יופיע פעם אחת בלבד.
6. הציגי את התשלומים שהסטודנטים שלהם הוא 1 והופקו ב-20/11/2008.
7. הציגי הפרש בימים בין תאריך תשלום לתאריך הפקדה, שלפי שורות שהפרש גדול מ-30 ימים.
8. הציגי את פרטי התלמיד שם האב והאם יופיעו יחדיו בעמודה אחת.
9. הציגי את שמות התלמידים המתחילה באות ש ומכללים את האות ה.
10. הציגי שם תלמידה, תז' וחשיבי אורך השדה תז', הציגי תלמידות שאורך התז קטן מ-9.
11. הציגי שם פרטי ומשפחה של תלמידה בעמודה אחת רק לתלמידות שאין להם טלפון ושנת לימודים אחרונה היא ותשפט או ששנת לימודים ריקה מיini בסדר עולה לפי שם התלמידה.
12. הציגי את שם והזמידה (שם פרטי ומשפחה ייחדו) כתה ומס' האחים שיש לה מיini את הרשימה לפי כתה ומין נוסף לפי שם. שמי לב שכן אשר חסר שם פרטי או משפחה יופיע אחד מהם, העזרי בפונקציה nullif

שאילות group by

1. הציגי קוד תלמידה סכום כולל ששלמה וכמות הפעמים שלמה
2. הציגי סכום כסף כולל שהתקבל מכל בנק בכל תאריך הפקדה רק בתאריכים 08.09.2008 עד 08.12.31 מיני ע"פ תאריך הפקדה
3. הציג את סוג תשלום וכמות התשלומים לכל סוג תשלום.
4. חשבו בממוצע כמה כסף התקבל בכל תאריך תשלום.
5. ספרי כמה תלמידות לומדות בכל כיתה.
6. הציג את הפרטים: בנק סניף וחשבון עבור 2 תלמידים שששלמו בתאריך המאוחר ביותר.
7. הציג את מספר האחיזות המינימלי והמקסימלי לכל כיתה מיני לפי המקסימום בסדר יורד
8. הציג את כמות התלמידות שנולדו בכל תאריך לידה בשנים 1990 עד 2005
9. הציג את סכום הכספיimal שהתקבל מכל סניף של בנק 20.
10. הציג סכום כולל של התשלומים שהסתטוס שלהם הוא 1
11. הציג 10% מהתלמידות שמספר אחיזותיהן הינו המקסימלי.

שאילות – group by , having ,having תחת שאילות מתואמות

1. הציג שם תלמידה וסכום כולל ששלמה כולל תלמידות שלא שלמו
2. הציג שם תלמידה וכמות הפעמים שלמה הציג רק תלמידות שכמות הפעמים גדולה או שווה 2
3. חשבו סכום כולל שהתקבל מכל בנק הציג שם בנק כולל בנקים שלא בצעו תשלום
4. חשבו סכום כולל שהתקבל מכל בנק וסניף בין התאריכים 20/09/2008 עד 31/12/2008 הציג רק סכומים כוללים הגודלים מ-500 ש"ח, מיני מהסכום הגבוה ביותר לנמק ביותר
5. ספרי כמה תלמידים היו בכל תאריך תשלום בחודש 9/2008
6. הציג שם תלמידה ושם הבנק שלו כולל תלמידות לא בנק וכל בנק ללא תלמידות.
7. הציג שם בנק סניף ומספר חשבון וסכום כולל ששולם וכמות התשלומים שהתבצעו בשנת 2008 ורק מבנקים 09,20,12,52 ורק אם הסכום גדול מ-3000 וכמות התשלומים גדולה מ-3
8. חשבו בממוצע כמה כסף התקבל בכל תאריך, מיני לפי תאריך
9. ספרי כמה תלמידות בכיתה, הציג רק כיתות יותר מ-30, מיני לפי כיתה.
10. הציג את תאריך הפקדה הראשון ותאריך הפקדה האחרון.
11. ספרי כמה תלמידות בכל משפחה ע"פ שם המשפחה שם האב והאם.
12. הציג את רשימת המשפחות מטבלת תלמידות.
13. הציג את שמות התלמידות שששלמו בתאריך המאוחר ביותר.
14. הציג את שמות התלמידות וכמותן שששלמו בחודש ה-10 ולא שלמו בחודש ה-9
15. הציג את שמות הבנקים שלא בוצעו על ידם תשלום.
16. הציג את שם תלמידה,בנק,סניף,חשבון הציג רק תלמידות שששלמו יותר מ-2 פעמים, מיני את הרשימה מכמות הפעמים הגבוהה ביותר לנמקה ביותר.
17. הציג את תלמידות שהסכום הכללי שששלמו גבוהה מהסכום הכללי של תלמידה מס' 245

18. הציגי את התלמידות שלא משלמות דרך אותו בנק שמשלמת תלמידה מס' 105
19. הציגי את התלמידות מכיתה מס' 2 שאין להם כלל תלולמים בשנת 2009
20. הציגי שם תלמידה, שם בנק, סניף וחשבון אם משלמת בהוראת קבוע. כלומר: לא היו לה תלולמים נוספים מסווג תלולום אחר

21. הציגי תלמידה שלמה בשני סוגים תלולמים שונים
22. הציגי את התלמידות שם אמם זהה לשם אחית מבנות כיתתם.

תשובות בעמוד 53

שאליות צורף טבלה אל עצמה

23. הציגי את פרטי התלמידות שיש להם אחות בדק אם מספר הטלפון שלהם זהה
24. הציגי תלמידות שהאחות שלהם נמצאת כתה מתחתח או כיתה מעיליהם.

תשובות בעמוד 53

שאליות מורכבות

1. הציגי בנות שם משפחתן זהה לשם של אחד משמות האבות באחד המשפחות
2. הציגי שם תלמידה, שם האב ושם האם בשלוש שורות נפרדות בזיה אחר זה.
3. הציג שמות בנות שללו בחודש 9 ולא שלמו ב-10.

שאליות פעולה

1. צרי טבלה חדשה בשם בוגרות הטבלה תכיל את השדות: תז, שם התלמידה, שנת למוד אחרונה, טלפון, כתובות, עיר, תאריך תשלום אחרון, סכום כולל ששולם.
2. הוספי לטבלת בוגרות את כל התלמידות מכיתה ז.
3. מחקי את כל נתוני התலולמים של התלמידות מכיתה ז.
4. מחקי מטבלת תלמידות את כל התלמידות מכיתה ז.
5. צרי טבלה חדשה בשם כיתות ובה שדה: כתה ומבחן. הוספי לתוכה את רשימת הכתובות מטבלת תלמידות כל כתה פעמי אחית בלבד.
6. הוספי שורות לטבלת תלולמים עבור כל התלמידות שללו בחודש הקודם הקודם בהוראת קבוע. כל השדות יכלו את אותם נתונים אך תאריך התשלום ירשם בתאריך הקודם+חודש.
7. עדכני את השדה מספר אחיות ע"פ ספירת כמות הבנות שיש להן את אותו מספר טלפון.
8. מחקי את כל התלמידות שתו שלhn מופיעה יותר מפעם אחת בטבלה.

תשובות עמוד 56

נספח ב

תשובות לתרגילים עמוד 14

```

select sum(SALARY)
from dbo.EMPLOYEE_PAY_TBL
where datelastraise>'01/01/2000'
---2
select sum(SALARY)
from dbo.EMPLOYEE_PAY_TBL
where salary+bonus between 1000 and 3000
---3
select CITY, count(EMPID)
from dbo.EMPLOYEE_TBL
group by city
---4
select EMPID, sum(SALARY)
from dbo.EMPLOYEE_PAY_TBL
where year(datelastraise)=year(getdate())
group by EMPID
---5
select CUSTID, count(ORDNUM)
from dbo.ORDERS_TBL
group by CUSTID
order by count(ORDNUM)
---6
select top 2 CUSTID, count(ORDNUM)
from dbo.ORDERS_TBL
group by CUSTID
order by count(ORDNUM)
---7
select year(dateborn), count(EMPID)
from dbo.EMPLOYEE_TBL
group by year(dateborn),
---8
select avg(SALARY), year(datelastraise), month(datelastraise)
from dbo.EMPLOYEE_PAY_TBL
group by year(datelastraise), month(datelastraise)
having avg(SALARY)>3000
order by 2, 3
---9
select max(QTY), prodid
from dbo.ORDERS_TBL
where prodid between 100 and 200
group by prodid
having max(QTY)>5
order by max(qty) desc
---10
select count(SALARY), EMPID
from dbo.EMPLOYEE_PAY_TBL
where datelastraise between '01/01/2009' and '12/31/2009'
group by EMPID
having count(SALARY)=12
---11
select empid, min(salary)
from dbo.EMPLOYEE_PAY_TBL
group by empid
having min(salary)<1500
---12

```

```

select EMPID, sum(SALARY+bonus+(payrate*hours)),year(datelastraise)
from dbo.EMPLOYEE_PAY_TBL
groupby EMPID,year(datelastraise)
--13
select custid,count(distinct prodid)
from dbo.ORDERS_TBL
group by custid
having count(distinct prodid)>=2
--14
Select prodid,min(orddate),max(orddate)
From orders
Group by prodid
Having datediff(year, (orddate),max(orddate))<10
--15
Select year(datelastraise),month(datelastraise),sum(salary)
From employee_pay_tbl
Where year(datelastraise)>2001
Group by year(datelastraise),month(datelastraise)
Having sum(salary)>5000
--16
Select top 2 prodid,sum(qty)
From orders
Group by prodid
Order by sum(qty) desc
--17
Select empid ,sum(salary)
From employee_pay_tbl
where year(datelastraise)=2006
Group by empid
Order by sum(salary)
--18
Select top 1 year(datehire),count(empid)
From employee_tbl
Group by year(datehire)
Order by count(empid)
--19
Select city,avg(datediff(year,birthdate,getdate()))
From employee_tbl
Group by city
--20
Select count(distinct custid) from orders
--21
Select count(distinct empid) from customers
--22
Select custid,year(orddate),sum(qty)
From orders
Group by custid,year(orddate)
--23
Select custid, sum(qty)
From orders
Where year(orddate)>=2005
Group by custid
Having sum(qty)>50

```

תשויות בעמונן

```

---1
select proddesc,max(ORDDATE) 'תאריך אחרון'
from dbo.ORDERS_TBL rightjoin dbo.products_tbl
on dbo.products_tbl.prodid=dbo.ORDERS_TBL.prodid
groupby dbo.ORDERS_TBL.prodid,proddesc
---2
Select CUSTNAME,max(ORDDATE),min(orddate),count(ordnum)

```

```

from dbo.ORDERS_TBL right join dbo.CUSTOMERS_TBL
on dbo.CUSTOMERS_TBL.CUSTID=dbo.ORDERS_TBL.CUSTID
group by CUSTNAME, dbo.ORDERS_TBL.CUSTID
---3
select CUSTNAME, ORDDATE, QTY*cost 'מחיר',
from dbo.CUSTOMERS_TBL c inner join dbo.ORDERS_TBL o
on c.CUSTID=o.CUSTID innerjoin dbo.products_tbl
on o.prodid=dbo.products_tbl.prodid
---4
select CUSTID, QTY*cost 'מחיר', QTY*cost*0.05 'הנחה', QTY*cost*0.95 'סוף',
from dbo.ORDERS_TBL o inner join dbo.products_tbl p
on o.prodid=p.prodid
where QTY>50
---5
selectavg(SALARY), sum(SALARY), CITY
from dbo.EMPLOYEE_PAY_TBL ep inner join dbo.EMPLOYEE_TBL e
on ep.EMPID=e.EMPID
where PHONE like '03%'
group by city
---6
selectsum(QTY), proddesc
from dbo.products_tbl p innerjoin dbo.ORDERS_TBL o
on o.prodid=p.prodid
where cost>1
groupby proddesc, p.prodid
---7
selectcount(QTY), proddesc
from dbo.products_tbl p inner join dbo.ORDERS_TBL o
on o.prodid=p.prodid inner join dbo.CUSTOMERS_TBL c
on c.custid=o.custid
where CUSTCITY='ירושלים'
groupby proddesc, p.prodid
--8
selectsum(qty*cost)
from dbo.products_tbl p inner join dbo.ORDERS_TBL o
on o.prodid=p.prodid
--9
select CUSTID,sum(qty*cost)
from dbo.products_tbl p inner join dbo.ORDERS_TBL o
on o.prodid=p.prodid
groupby CUSTID
--10
select custid ,avg(cost)
from dbo.products_tbl p inner join dbo.ORDERS_TBL o
on o.prodid=p.prodid
group by CUSTID having avg(cost)>1.75
11-
select CUSTID from dbo.ORDERS_TBL groupby custid
havingdatediff(year,min(ORDDATE),max(ORDDATE))>2
--1
select custname,custcity,lastname+' '+firstname,city
from customers_tbl c inner join employee_tbl e
on c.empid=e.empid
where city=custcity
--2
Select lastname+' '+firstname,custname,orddate
from customers_tbl c inner join employee_tbl e
on c.empid=e.empid inner join orders_tbl o
on o.custid=c.custid
where year(orddate)>2005
--3
Select custname,proddesc,qty*cost

```

```

From from customers_tbl c inner join orders_tbl o
on c.custid=o.custid inner join products_tbl p
on o.prodid=p.prodid
order by custname,orddate
--4
select e.empid,e.lastname+' '+e.firstname,m.empid,m.lastname+
'+m.firstname
from dbo.EMPLOYEE_TBL e left join dbo.EMPLOYEE_TBL m
on e.managerid=m.empid
--5
select o.prodid
from orders_tbl o,orders_tbl o1
where o.prodid!=o1.prodid
group by o.prodid
having sum(o.qty)=sum(o1.qty)
--6
Select e.empid
From employee_pay_tbl e,employee_pay_tbl e1
Where e.empid=e1.empid and e.salary=e1.salary and
e.datelastraise!=e1.datelastraise
--7
Select custid
From orders_tbl o,orders_tbl o1
Where o.custid=o1.custid
and datediff(d,o.orddate,o1.orddate) between 1 and 5

```

תשובות עמוד 20-

```

--tar1
select ORDNUM
from dbo.ORDERS_TBL
where ORDDATE =any(select ORDDATE
from dbo.ORDERS_TBL
where CUSTID=20)
--tar2
select *
from dbo.ORDERS_TBL
where prodid =any(select prodid
from dbo.ORDERS_TBL
where CUSTID=20)
--tar3
select *
from dbo.EMPLOYEE_PAY_TBL
where SALARY>any(select SALARY
from dbo.EMPLOYEE_PAY_TBL
where EMPID =011465986)
----4
select SALARY,empid
from dbo.EMPLOYEE_PAY_TBL
where salary=(selectavg(salary)
from dbo.EMPLOYEE_PAY_TBL)
----5
select EMPID,SALARY
from dbo.EMPLOYEE_PAY_TBL
where salary>all(selectavg(salary)
from dbo.EMPLOYEE_PAY_TBL
groupby EMPID)
--6
selectsum(salary),empid
from dbo.EMPLOYEE_PAY_TBL
groupby empid
havingsum(salary)<(selectsum(salary)from dbo.EMPLOYEE_PAY_TBL ep
innerjoin dbo.EMPLOYEE_TBL e on e.empid=ep.empid

```

```

where firstname='משה')
--7
select CUSTID,avg(QTY),sum(QTY)
from dbo.ORDERS_TBL
groupby custid
having avg(QTY)>20 and sum(QTY)>100
--8
select empid
from dbo.EMPLOYEE_TBL
where POSITION in(select position from dbo.EMPLOYEE_TBL
where firstname='יצחק')
--9
select proddesc,cost
from dbo.products_tbl
where cost=(select min(cost) from dbo.products_tbl)
--10
select CUSTNAME,avg(cost*qty)
from dbo.CUSTOMERS_TBL c inner join dbo.ORDERS_TBL o
on o.custid=c.custid inner join dbo.products_tbl p
on p.prodid=o.prodid
groupby custname,c.custid
having avg(qty*cost)>
(select avg(qty*cost)
from dbo.CUSTOMERS_TBL c inner join dbo.ORDERS_TBL o
on o.custid=c.custid
inner join dbo.products_tbl p
on p.prodid=o.prodid
where custname='חימס לוי')
--11
select proddesc,cost,sum(qty*cost)
from dbo.ORDERS_TBL inner join dbo.products_tbl on p.prodid=o.prodid
where cost between 1 and 6
groupby prodid,proddesc,cost
having sum(qty*cost)<(select sum(qty*cost)
from dbo.ORDERS_TBL inner join dbo.products_tbl
on p.prodid=o.prodid where prodid=101) order by cost

```

22 תשובות عمود

```

--tar 1
select o.*
from dbo.ORDERS_TBL o inner join dbo.products_tbl p
on p.prodid=o.prodid
where cost =(select max(cost) from dbo.products_tbl)
--tar 2
select * from dbo.products_tbl
where prodid not in (select prodid from dbo.ORDERS_TBL
where month(ORDDATE)=1 and year(ORDDATE)=2008 )
--tar 2 נספף
select *
from dbo.ORDERS_TBL o
where not exists
(select 1 from dbo.ORDERS_TBL
where prodid=o.prodid and month(ORDDATE)=1 and year(ORDDATE)=2008 )
---3
select CITY from dbo.EMPLOYEE_TBL
where city not in (select CUSTCITY from dbo.CUSTOMERS_TBL )
---3 נספף
select CITY from dbo.EMPLOYEE_TBL
where not exists (select 1 from dbo.CUSTOMERS_TBL where custcity=city)
---4
select * from dbo.EMPLOYEE_TBL

```

```

where EMPID in(select EMPID from dbo.EMPLOYEE_PAY_TBL
               where salary=(select min(salary) from EMPLOYEE_PAY_TBL))

--5
select lastNAME+' '+firstname from dbo.EMPLOYEE_TBL
where lastNAME+' '+firstname
in(select CUSTNAME from dbo.CUSTOMERS_TBL)
--5-טורי ניסף
select lastNAME+' '+firstname from dbo.EMPLOYEE_TBL
where exists(select 1 from dbo.CUSTOMERS_TBL where custname=lastNAME+' '+firstname)

--6
select CUSTNAME from dbo.CUSTOMERS_TBL
where custid not in(select custid from dbo.ORDERS_TBL
                     where orddate>dateadd(month,-1,getdate()))

--7
select CUSTNAME,CUSTCITY from dbo.CUSTOMERS_TBL c innerjoin
dbo.ORDERS_TBL o on o.custid=c.custid
where prodid in (select prodid
                  from dbo.CUSTOMERS_TBL c innerjoin dbo.ORDERS_TBL o
                  on o.custid=c.custid
                  where custname in('רמי', 'רלאט', 'נירית'))
order by CUSTCITY

-
--8
select prodid from dbo.ORDERS_TBL
where qty in
(select QTY from dbo.ORDERS_TBL o innerjoin dbo.CUSTOMERS_TBL c on
c.custid=o.custid where custname='הוּא' or CUSTCITY='בנ' ברק')
--9
select CUSTID,prodid,QTY
from dbo.ORDERS_TBL o
where QTY>(select avg(Qty)
              from dbo.ORDERS_TBL
              where prodid=o.prodid)

---10
select c.CUSTNAME,p.proddesc,o.ORDDATE,o.QTY,p.cost
from dbo.CUSTOMERS_TBL c innerjoin dbo.ORDERS_TBL o on
c.CUSTID=o.CUSTID
innerjoin dbo.products_tbl p on o.prodid=p.prodid
where p.cost=(select max(cost)
              from dbo.ORDERS_TBL o1 innerjoin dbo.products_tbl
              p1 on o1.prodid=p1.prodid
              where o1.CUSTID=o.CUSTID)

---11
select c.CUSTNAME,o.ORDNUM,o.Qty
from dbo.CUSTOMERS_TBL c innerjoin dbo.ORDERS_TBL o on
c.CUSTID=o.custid
where Qty>5 and 1=(select count(CUSTID) from dbo.CUSTOMERS_TBL
                     where CUSTNAME=c.CUSTNAME) order by Qty desc

---12
Select avg(salary),sum(salary),city
from dbo.EMPLOYEE_PAY_TBL ep inner join dbo.EMPLOYEE_TBL e on
e.EMPID=ep.EMPID
group by e.CITY
having avg(salary)!=(select top 1 avg(salary)
from dbo.EMPLOYEE_PAY_TBL ep inner join dbo.EMPLOYEE_TBL e
on e.EMPID=ep.EMPID group by city order by 1)
and avg(salary)!=(select top 1 avg(salary) from dbo.EMPLOYEE_PAY_TBL
ep inner join dbo.EMPLOYEE_TBL e on e.EMPID=ep.EMPID
group by city
order by 1 desc)

```

תשובות עמוד 23

```
--1
select *,avg_salary,salary-avg_salary
from
(select empid,avg(salary)as avg_salary from dbo.EMPLOYEE_PAY_TBL
group by empid) as s
inner join dbo.EMPLOYEE_PAY_TBL ep on ep.empid=s.empid
--2
select ORDNUM,o.CUSTID,qty*cost,avg_cost,avg_cost-(qty*cost)
from dbo.ORDERS_TBL o inner join dbo.products_tbl p
on o.prodid=p.prodid inner join
(select custid,avg(qty*cost) as avg_cost
 from dbo.ORDERS_TBL o inner join dbo.products_tbl p
 on o.prodid=p.prodid group by custid )as s on s.custid=o.custid
order by custid

--3A
select CUSTNAME,proddesc,qty
from dbo.CUSTOMERS_TBL c inner join dbo.ORDERS_TBL o on
o.custid=c.custid
inner join dbo.products_tbl p on p.prodid=o.prodid
inner join (select max(cost) as maxi,CUSTID from dbo.ORDERS_TBL o
inner join dbo.products_tbl p on o.prodid=p.prodid group by custid)
as s
on s.custid=c.custid and p.cost=s.maxi
--3B
select CUSTNAME,proddesc,qty
from dbo.CUSTOMERS_TBL c inner join (select proddesc,qty,CUSTID from
dbo.ORDERS_TBL
inner join dbo.products_tbl p on p.prodid=dbo.ORDERS_TBL.prodid where
cost=(select max(cost) from dbo.ORDERS_TBL o inner join
dbo.products_tbl p on o.prodid=p.prodid where
custid=dbo.ORDERS_TBL.custid)) as s
on s.custid=c.custid
--4
select s1.y,max_qty,s2.prodid
from
(select year(ORDDATE) as y,prodid,sum(qty) as sum_qty
from dbo.ORDERS_TBL
group by year(ORDDATE),prodid) as s2
inner join
(select y,max(sum_qty) as max_qty
from (select year(ORDDATE) as y,prodid,sum(qty) as sum_qty
from dbo.ORDERS_TBL
group by year(ORDDATE),prodid) as s
group by y) as s1
on s2.y=s1.y and s2.sum_qty=s1.max_qty
```

תשובות עמוד 25

```
--1
Select city from [dbo].[EMPLOYEE_TBL]
except
select [CUSTCITY] from [dbo].[CUSTOMERS_TBL]
--2
Select city from [dbo].[EMPLOYEE_TBL]
intersect
select [CUSTCITY] from [dbo].[CUSTOMERS_TBL]
--3
Select city from [dbo].[EMPLOYEE_TBL]
```

```

from dbo.payment
where datehafkada between '09/20/2008' and '12/31/2008'
groupby bank,datehafkada
orderby datehafkada
--targil 3
select kindpay,count(payment)
from dbo.payment
groupby kindpay
--targil 4
select datepay,avg(payment)
from dbo.payment
groupby datepay
--targil 5
select grade,number,count(pupilid)
from dbo.pupils
groupby grade,number
--targil 6
selecttop 2 bank,snif,cheshbon,datehafkada
from dbo.payment
orderby datehafkada desc
--targil 7
select grade,number,min(numsister),max(numsister)
from dbo.pupils
group by grade,number
order by max(numsister)desc
--targil 8
select dateborn,count(pupilid)
from dbo.pupils
where dateborn between '01/01/1990' and '01/01/2005'
group by dateborn
--targil 9
Select max(payment),snif
from payment
where bank=20
group by snif
--targil 10
Select sum(payment)
from dbo.payment
wherestatus='3'
--targil 11
Select top 10 percent*
from dbo.pupils
order by numsister desc
--targil 12
selectmin(dateborn),max(dateborn),lastyear
from dbo.pupils
group by lastyear
--targil 13
Select avg(numsister)
from dbo.pupils
where numsister>1

```

4. תשובות עמוד

```

---1
select lastname,sum(payment)
from dbo.payment right join dbo.pupils
on dbo.pupils.pupilid=dbo.payment.pupilid
group by lastname, dbo.pupils.pupilid

```

```

--- 2
select pupilid,count(tashid)
from dbo.payment
group by pupilid
having count(tashid)>=2
---3
select namebank,sum(payment)
from dbo.payment right join dbo.bankim
on dbo.payment.bank=dbo.bankim.bankid
group by namebank
---4
select sum(payment),bank,snif
from dbo.payment
where datepay between '09/20/2008' and '12/31/2008'
group by bank,snif
having sum(payment)>500
order by 1 desc
---5
select count(tashid),datehafkada
from dbo.payment
where year(datehafkada)=2008 and month(datehafkada)=9
group by datehafkada
---6
select lastname,namebank
from dbo.bankim full outer join dbo.payment
on dbo.bankim.bankid=dbo.payment.bank full outer join dbo.pupils
on dbo.payment.pupilid=dbo.pupils.pupilid
---7
select namebank,snif,cheshbon,sum(payment),count(tashid)
from dbo.payment inner join bankim
on bankim.bankid=payment.bank
where year(datehafkada)=2008 and bank in(09,20,12,52)
group by namebank,snif,cheshbon
having sum(payment)>300 and count(tashid)>3
---8
select datehafkada,avg(payment)
from dbo.payment
group by datehafkada
order by datehafkada
---9
select count(pupilid),class
from dbo.pupils
group by class
having count(pupilid)>30
order by class
---10
select min(datehafkada),max(datehafkada)
from dbo.payment
---11
Select count(pupilid),lastname, namefather,namemother
from dbo.pupils
group by lastname, namefather,namemother
---12
Select distinct lastname
from dbo.pupils
group by lastname
---13
select lastname
from dbo.payment inner join dbo.pupils p on
p.pupilid=dbo.payment.pupilid
where datehafkada=(select max(datehafkada)
                  from dbo.payment)

```

```

---14
select lastname,adress
from dbo.pupils inner join dbo.payment on
dbo.pupils.pupilid=dbo.payment.pupilid
where month(datehafkada)=10 and pupilid not in(select pupilid
from dbo.payment
where month(datehafkada)=9)

---15
select bankid,namebank
from dbo.bankim
where dbo.bankim.bankid not in(select dbo.payment.bank
from dbo.payment where bankid is not null)

---16
select lastname,bank,snif,cheshbon ,count(tashid)
from dbo.payment inner join dbo.pupils on
dbo.payment.pupilid=dbo.pupils.pupilid
group by dbo.payment.pupilid,lastname,bank,snif,cheshbon
having count(tashid)>2
order by count(tashid)desc

---17
select pupilid,sum(payment)
from dbo.payment
groupby pupilid
having sum(payment)>(select sum(payment)
from dbo.payment
where pupilid=245)

---18
select pupilid,bank
from dbo.payment
where bank not in(select bank
from dbo.payment
where pupilid=105)

---19
select pupilid
from dbo.pupils
where class='2r' and pupilid not in(select pupilid
from dbo.payment
wher eyear(datepay)='2009')

---20
select namebank,snif,cheshbon,firstname
from dbo.pupils p inner join dbo.payment pay on p.pupilid=pay.pupilid
inner join dbo.bankim b on b.bankid=pay.bank
where kindpay='הוראת קבע' and pay.pupilid not in(select pupilid
from dbo.payment
where kindpay !='הוראת קבע')
groupby p.pupilid,namebank,snif,cheshbon,firstname

---21
select pupilid
from dbo.payment p
where exists (select kindpay
from dbo.payment
where pupilid=p.pupilid and
dbo.payment.kindpay!=p.kindpay)

---21
Select pupilid,count(distinct kindpay)
From payment
Group by pupilid
Having count(distinct kindpay)>1

---22
select pupilid
from dbo.pupils p

```

```

where namemother in(select firstname from dbo.pupils
                     where p.class=dbo.pupils.class)
---23
Select o.pupilid
from dbo.pupils p inner join pupils p1
on p.phone =p1.phone and p.lastname=p1.lastname
and p.adress=p1.adress and p.pupilid!=p1.pupilid
)
---24
Select o.pupilid,p.grade
from dbo.pupils p inner join pupils p1
on p.phone =p1.phone and p.lastname=p1.lastname
and p.adress=p1.adress and p.pupilid!=p1.pupilid
and (p.class+1=p1.class or p.class-1=p1.class)

```

תשובות 43

```

select lastname from pupils
intersect
select namefather from pupils
--2
select pupilid,lastname+' '+firstname from pupils
union
select pupilid,lastname+' '+namefather from pupils
union
select pupilid,lastname+' '+namemother from pupils
order by 1
---3
select pupilid
from dbo.payment
where month(datehafkada)=9
except
select pupilid
from dbo.payment
where month(datehafkada)=10

ת"ז ינואר 2013
--2
insert bogrot
select
[tz],[lastname]+'\''+[firstname],[lastyear],[phone],[adress],[city],max([datepay])
),sum([payment])
from [dbo].[payment] p join [dbo].[pupils] pu
on pu.pupilid=p.pupilid
where [class]='i' and [number]=5
group by [tz],[lastname]+'\''+[firstname],[lastyear],[phone],[adress],[city]
--3
delete payment
where pupilid in(select pupilid from [dbo].[pupils] where [class]='i' and
[number]=5)
--4
delete [dbo].[pupils]
where [class]='i' and [number]=5
--5-
insert class
select distinct [class],[number]
from [dbo].[pupils]
group by [class],[number]
--6
insert [dbo].[payment]
select getdate(), dateadd(month,1,datehafkada) , status, pupilid, kindpay,
payment, bank, snif, cheshbon, numbershek
from [dbo].[payment]

```

SQL SERVER

```
where [kindpay]='ת>New Card' and [datehafkada ]='01/20/2017'  
--7  
update [dbo].[pupils]  
set [numsisiter]=(select count(*) from [dbo].[pupils] p where  
[phone]=[dbo].[pupils].phone)  
--8  
delete from [dbo].[pupils]  
where tz in (select p1.tz from [dbo].[pupils] p1,[dbo].[pupils]  
where [dbo].[pupils].[pupilid]>p1.[pupilid] and [dbo].[pupils].tz=p1.tz  
)
```



סמינר בית יעקב
ע"ש הרוב וולף זצ"ל

הו ٢٠١٣

transact-sql

T-SQL

תשע"ט

0	SQL-T
2	A.
2	תכנות בסיסי.....
3	הגדרת משתנים:
4	הדפסת משתנים.....
4	תנאים Conditions
4	שימוש ב-BETWEEN.....
5	שימוש ב-Case.....
6	לולאות – While
7	לולאות – GOTO
8	תרגילים:.....
10	B. טבלה מדומה View
12	תרגילים.....
13	C. – פ्रוצדורה מאוחסנת Stored-Procedure
13	עבודה.....
14	D. פרמטרים בפרוצדורות
15	E. עבודה.....
16	בוחן ב-SQL על תרגיל מסמכים אחרונים
17	F. פונקציות
17	פונקציות סקלריות (מחזירות ערך בודד)
18	G. עבודה.....
19	פונקציות שמחזירות טבלה

T-SQL

21	דוגמאות לפונקציות	.
23	בוחן ב SQL – וו ופונקציות	.
24	מבחן ב SQL - T-SQL - SQL - פרצדרות, וו, פונקציות	.
29	סמן - Cursor	.1.
29	עבודה	.
31	- מזניק Trigger	.2.
32	שיעורי בית	.
33	בוחן טריגרים	.
34	בוחן ב SQL – trigger, cursor – SQL	.
35	Sql Server – טיפול בשגיאות בסביבת Try catch	.ח.
37	טיפול בשגיאות – עבודה	.
38	Transaction	.ט.
38	עבודה	.
39	Sql – Dynamic Sql – דינמי	.ו.
40	פרויקטים	.יא.
40	1. פרויקט בנק.	.
43	2. פרויקט מעבדה	.
49	מבחנים	.יב.
49	מבחן - 4 תרגילים מהברת אירוקס	.1.
54	מבחן אמריקאי	.2.
58	מבחן ב SQL	.3.
61	מבחן ב SQL	.4.

T-SQL

A. T-SQL BASIC

תכנות בסיסי

- שפה זו היא הרחבה של SQL המאפשרת ביצוע פעולות חכניות על מסד הנתונים (משתנים, פונקציות, לולאות, תנאים ועוד).

בפרק זה נעסוק בפקודות הבסיסיות של T-SQL עמן נעבד וنعוזר בהמשך הדרכ באובייקטים נוספים שילמדו בהמשך:

- מושגי יסוד
- הגדרת משתנים מקומיים
- השימוש ערך למשנה מקומי (אוחול משתנה)
- השימוש ערך לתוך משתנה מתוך משפט select
- הדפסת משתנה
- תנאים
- CASE
- WHILE
- לולאת GOTO
- טיפול בשגיאות

מושגי יסוד:

- - Batch – זה רצף של פעולה/ שאלתה או מספר פעולות הנשלחות לשרת לשם ביצוע יחדיו.(בפקודות DDL כל פקודה היא Batch נפרדת).
- [] – למלים שמורות או רוחחים בשמות טבלאות או עמודות.
- @- סימן למשנה.
- @@ROWCOUNT – משתנה גלובלי שמחזיר את מספר הרשומות שהושפטו מפקודה מסוימת (select).

הערות: ניתן לנתח הערות בתוך הקוד בשתי צורות:

- 1. הערה על פניו מספר שורות:

```
/* this is a comment
```

```
And this is also a comment */
```

2. הערה עד סוף השורה:

```
Select title,isbn --this is a comment
```

```
From books
```

T-SQL

הגדרת משתנים:

משתנים מקומיים מוגדרים בתחום משפט SQL או בתחום פרוצדורה באמצעות מילת המפתח DECLARE. לאחר שם המשתנה מצינים את הטיפוס, טיפוס הנתונים יכול להיות כל אחד מהטיפוסים המוכרים ב-sql, ניתן להגיד ברצף כמה משתנים עם סימן , ביניהם.

מבנה:

```
DECLARE @local_variable data_type [, @local_variable data_type]
```

דוגמאות להגדרת משתנים:

```
DECLARE @CustName varchar(30)  
DECLARE @type int,  
@price money
```

השנת ערך למשתבה אקורסוי

ניתן להשים ערכים למשתנים על ידי:

- שימוש ב-SET ובכך לאתחל את המשתנה. (אתחול כל המשתנה מתבצע על ידי שימוש במשפט SET בנפרד)

```
SET @local_variable = value
```

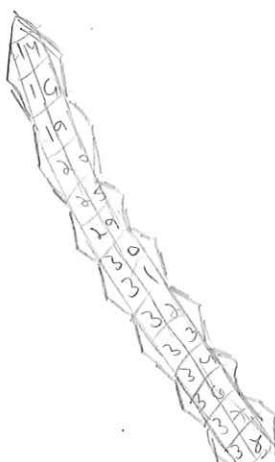
- השנת ערך מתחום בסיס הנתונים על ידי שימוש במשפט SELECT. כאשר המשתנים מקומיים מוגדרים ב-select משפט ה-select חייב להחזיר שורה אחת בלבד.

```
SELECT @local_variable = column_name  
[, @local_variable = column_name....]  
FROM table_name  
WHERE condition
```

לדוגמה:

```
Declare @CustomerID char(9)  
SET @Customerid='301958745'  
Select * from customers_tbl where custID= @CustomerID
```

```
Declare @g int  
Select @g=max(price)  
From products_tbl
```



T-SQL

הדף משתנים

ניתן להדפיס משתנים מקווצמים:

- בוחץ Message על ידי שימוש ב-PRINT PRINT מקבל ערכים מהרשותם בלבד, במידה הצורך יש לבצע המרה – שימוש ב-CAST או (CONVERT

מבנה:

PRINT @local_variable

- בוחץ RESULT על ידי שימוש ב-.SELECT

מבנה:

Select @local_variable

תנאים Conditions

שימוש ב-IF מאפשר לבצע בדיקה במידה ותנאי מתקיים או לא ומהן הפקודות שיתבצעו בהתאם.

מבנה:

If Boolean_expression

....

Else

....

במקרה שנדרצה לציין יותר מפקודה אחת יש להשתמש בבלוק:

מבנה בלוק פקודות:

IF Boolean_expression

Begin

...

...

End

Else

Begin

...

...

End

לדוגמה:

If(select avg(price) from books)>20

Print "more than 20"

Else

Print "less than 20"

SQL היא הרחבה לשפת SQL ניתן להשתמש בה בפונקציות ואופרטורים המוכרים מ-

.CASE ו-EXISTS לمثال

שימוש ב-Exists בתנאי

T-SQL

מבנה:

EXISTS subquery

Returns True if a subquery contains any rows.

דוגמה:

```
If exists (select * from orders  
Where customerID = 'vinet')  
BEGIN  
PRINT 'The customer has order'  
END  
ELSE  
BEGIN  
Delete customers where customerID='vinet',  
PRINT 'Customer Deleted'  
END
```

שימוש בביוטי CASE

הפקציה CASE פועלת באופן דומה לתחן IF, ואפשר להשתמש בה בקוד T-SQL.
לזכור, CASE הוא ביוטי המספק אמצעי להחזרת תנאים שונים המבוסס על בדיקת הערך המוחזר.
ב-SQL קיימות שתי וריאציות שונות של case והן:

- Simple CASE expression
- Search CASE expression

דוגמה:

בדוגמה הנ"ל המשתנה @case_result מקבל מחרוזת בהתאם לערכו של x, וזאת תוך שימוש ב-

.case

```
DECLARE  
@x int,  
@case_result varchar(20)  
Set @x=1  
Set @case_result=case @x  
When 1 then 'you are number one!'  
When 2 then 'Almost...'  
Else 'Loser'  
End
```

ביוטי Case הוא ביוטי שמעירק תנאי אחד או יותר, ומחזיר ערך בוודוד מתוך מספר אפשרויות בהתאם למתואמת של הערךת הביוטי.

T-SQL

שימוש ב Case חוקי בכל מקום שבו-Amor להופיע ביטוי שמחזיר ערך, כשלה במשפט select, כחלק מתנאי במשפט Group by, Order by, Where בביטוי Case השימוש ב Else הוא אופציונלי, ובמקרה שלא צוין else יוחזר null. קיימות שתי צורות תחביר לביטוי Case • פשטוט – הביטוי משווה שדה / ביטוי מול ערך/ערכים אפשריים.

לדוגמה:

```
SELECT ProductId, Name,  
CASE Price  
    WHEN 0 THEN 'not for resale'  
    ELSE 'for resale'  
END AS PriceRange  
FROM Product
```

• מורכב – הביטוי מעריך תוצאה של תנאי או מספר תנאים

לדוגמה:

```
SELECT ProductId, Name,  
CASE  
    WHEN Price = 0 THEN 'not for resale'  
    WHEN Price < 50 THEN 'Under $50'  
    WHEN Price >= 50 and Price < 250 THEN 'Under $250'  
    WHEN Price >= 250 and Price < 1000 THEN 'Under $1000'  
    ELSE 'Over $1000'  
END AS PriceRange  
FROM Product
```

לולאות While

לולאת While מאפשרת ביצוע מחזורי של משפטים SQL. משפטים ה-SQL מופעלים באופן מחזורי כל עוד תנאי שהוגדר מתקיים. מבנה הלולאה:

WHILE Boolean_expression

(sql_statement | statement_block)

begin end

ניתןukan שתי לולאות While או יותר אחת בתוך השניה.

שימוש ב-break

שימוש במילה break מאפשר להפסיק את הלולאה.

שימוש ב-continue

שימוש ב-continue מאפשר להפסיק את האיתרציה הנוכחית של הלולאה ומעבירנו לאיתרציה הבאה.

T-SQL

לדוגמה:

```
DECLARE @n int
Set @n=1
While (select count(*) from employees)>=@n
Begin
If (select lastName from employees
Where employeeID =@n)= 'King'
Begin
Print 'King EmployeeID=' + CAST (@n as varchar(2))
Break
End
Else
Set @n=@n+1
End
```

לולאה – GOTO

לולאת GOTO מאפשרת הריצה מוחזרת של משפטים SQL כל עוד תנאי שהוגדר מתקיים.
אך בኒיגוד לlolאת while, הלולאה תחbezע פעם אחת לפחות והתנאי ייבדק רק לאחר הביצוע הראשון.
לולאה מסוג זה משמשת גם לביצוע פעולות אשר חוזרות על עצמו בקורס.

מבנה הלולאה:

Label:

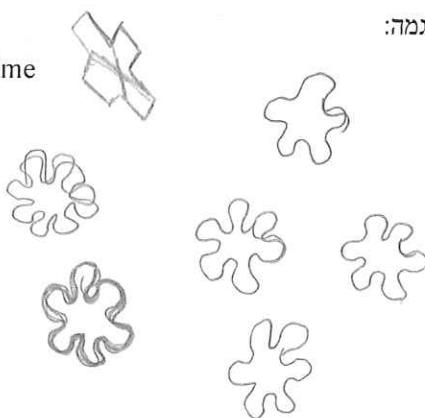
Sql_statement | statement_block

IF condition GOTO Label

משמעותו של המילה Label אינה מילה שמורה. ניתן להשתמש בכל רצף אותיות אחר.

לדוגמה:

```
DECLARE @emp_id int, @name varchar(20), @date datetime
Set @emp_id=4
Label:
Select @name=lastname, @date=birthdate
From employees
Where employeeID = @Empid
PRINT @name + cast(@date as varchar)
Set @emp_id = @emp_id+1
If @name like %a%' goto Label
```



T-SQL

תרגילים:

א. כתבי מה מבצע הקוד ה T-Sql הבא:

1. מה מבצע הקוד הבא:

```
Declare @d datetime  
Declare @text nvarchar(20)  
Set @d = '2010-1-1'  
If @d < getdate()  
    Print 'before'  
Else  
    if @d > getdate()  
        Begin  
            Set @text = 'the date in the variable is '  
            Select @text + @d  
        End
```

2. מה מבצעת הпроцедורה הבאה:

```
Create procedure proc1  
As  
Declare @x int, @y int  
Select @x = min(computerNumber), @y = max(computerNumber)  
From tbl_computers  
While @x < @y  
    Begin  
        set @x = @x + 1  
        Print @x  
    End
```

ב. בצעי את המשימות הבאות ב T-SQL. רשמי ליד כל שאלתה את הפלט שלה.

1. הגדרי שני משתנים מסוג Int אתחלי את שניהם והדפיסי את סכומם.
2. אתחלי רק אחד מהמעות והדפיסי את סכומם.
3. הגדרי משתנה מסוג מחרוזת, אתחלי אותו והדפיסי אותו באותיות גדולות.
4. הגדרי משתנה נוסף מסוג מחרוזת שאינו מאותחל ובדק כי אם הוא שווה ל'קדום', אם כן הדפיסי "שווים". בדק כי If נפרד אם אינם שווים, הדפיסי "שונים".

7-8Q

5. הגדרי משתנה נוסף מסוג Date, והוסיפו לו בוללאת while יום אחד בכל שלב עד שהיה שווה לתאריך של היום.
- בסיום הלולאה הדפסי בכמה ימים קטן הערך במשתנה מהתאריך של היום.
6. כתבי פרוצדורה השולפת את הסטאטוס השכיח ביותר בטבלת המחשבים.
- אם הסטאטוס הינו "תיקין" הפרוצדורה תדפיס "מצוין", אם הסטאטוס הינו "דרוש תיקון" הפרוצדורה תדפיס "מצב דרוש שיפור", אם הסטאטוס הינו "יצא מכלל שימוש", הפרוצדורה תדפיס "יש לבצע בדיק בית יסודי".
- השתמשי ב.case.
7. כתבי פרוצדורה המקבלת את הפרמטרים שם תוכנה, מספר סידורי ומהירות. ומוסיפה אותם לטבלת התוכנות.
- לפני ההוספה לטבלה בצעי בדיקה אם המספר הסידורי תקין. מספר תיקין הוא מספר שאורכו 14 תווים המחולקים ל3 קבוצות של 4 ספרים וביניהם חתו'ן.
- אם המספר אינו תיקין לא תתווסף הרשומה לטבלה.
8. כתבי פרוצדורה המקבלת פרמטר מסוג ARRAY, ופרמטר Out מסוג int.
- הפרוצדורה בודקת אם קיימות תוכנות שהתווקנו לאחר תאריך זה, אם כן הפרוצדורה תחזיר את שמותיהן. (שים לי לב לא להחזיר את אותה תוכנה פעמיים), אחריה הפרוצדורה תחזיר את כל התוכנות. עדכני בפרמטר Out ערך המסמך האם הרשימה שחזרה מכילה את כל התוכנות או תוכנות לפי התאריך סופוק.

case - עבודה

1. כתבי ויו שמציג את נתוני הפציינטים. יהיו יכול את השדות מזהה פציגנט, שם מלא, ת.ז.. השדה שם מלא יכול באופן רגיל שרשור של שם משפחה + שם פרטי, למעט במקרה שם המשפחה הוא מהשמות הנפוצים: 'כהן' או 'לוין', במקרה כזה ישורשר אחריו השם המשפחה והפרט המיליה 'בן' ואחריה יופיע שם האב.
2. כתבי פרוצדורה המקבלת שנה ומחזירה טבלה במבנה הבא: שם מלא (מהו), ת.ז., האם ביצוע בדיקות בשנה זו? - שדה זה יכול V במקרה שהפצינט ביצע בדיקות בשנה זו, ו X במקרה שלא ביצע. בצעי תרגיל זה גם בעוזרת union, וגם בעוזרת case.
3. כתבי פרוצדורה המקבלת ת.ז. של פציגנט ומחזירה עבורו תוצאות בדיקות במבנה הבא:
- שם פציגנט, תאריך בדיקה, שם בדיקה, תוצאה, טווח תוצאות (ירכב משרשור של מינימום קו מפheid ומקסימום, לדוגמה 50-100), ושרותות גרפי. שדה זה יכול את הסימן (*) במקרה שההתוצאה בטווח, (...) * במקרה שההתוצאה קטנה מהטווח, ו * (...) במקרה שההתוצאה גדולה מהטווח.
- בצעי תרגיל זה גם בעוזרת union, וגם בעוזרת case.
4. כתבי ויו ובו שדות עברו כל בדיקה. יהיו יכול רשומות עברו כל שנה ובכל רשומה יצוין עברו כל בדיקה כמה פעמים בוצעה באותה שנה.

ב. טבלה מדומה View

ה-VIEW הוא אובייקט אשר אחסון שאילתת בסיס הנתונים. צורת שילוף הנתונים ממנו והגישה אליו כטבלה לכל דבר ועל כן שמו: "טבלה מדומה".

בפועל הview אינו חופס מקום פיזי, כיון שרק ההגדרה שלו נשמרת בסיס הנתונים ואילו הערכים נשלפים בכל פניה ל-VIEW במשפט `from`. מכיוון שבכל פניה ל-view, הview מורץ מחדש, כל פניה יכולה ליצור עין טבלה שונה לפי הנתון הרגעי של TABLES הבסיס מהן נגזרים הנתונים.

יתרונות השימוש בVIEW:

- אבטחת מידע

ניתן לשולף או להסתיר עבור כל משתמש את המידע הרלוונטי אליו, ניתן לפחות כל המידע המוצג למשתמשים וכו'. לדוגמה: גנich מזKir מתחיל צריך לתקשר לכל המורים כדי להודיעו הוועדה כלשהי, מספר הטלפון נמצא בטבלה של נתוני העובדים המכילה גם את החוב המשכורת הממוצעת של אותו עובד. איןנו מעוניינים שהוא מזKir יראה את כל המידע הזה. לפיכך ביכולתנו ליצור view שיציג לאותו מזKir רק את שם העובד, כתובות וטלפון.

כמו"כ טבלת ציונים ותלמידים ביכולתנו להציג עבור כל מורה רק את הציונים במקצועות שלו וכו'.

- נוחות ובהירות

לעתים קיימים קשרים מסוימים בין TABLES, קשרים אלו רלוונטיים להרבה דוחות, לצורך כך ניתן לבנות view שיכיל מאחוריו הקלים את כל הלוגיקה, ואנחנו נוכל לפנות אל view ככל שנצרך.

כמו כן לעתים יש צורך לבצע שינויים על הטבלאות הרגילות, משתמש הניגש אל הטבלאות המדומות יכול להמשיך ולבנות את השאלות הללו כרגע, למטרות שהטבלאות השתנו. כמו"כ הדבר מקל מאד על כתיבת שאלות מסובכות.

- שיפור ויעילות- הטבלה עצמה לא תופסת מקום בזיכרון כי רק ההגדרה שלה נשמרת.

- מאorgan מידע עבור ייצוא לאפליקציות אחרות, ועוד.

כמובן שיש גם חסרונות לview, כמו לדוגמה:

- קשה לעדכנה- כדי שיוכלו לעדכן את ה-view צריך שיוכלו תנאים שיורחבו בהמשך.

- ביצועים- מכיוון שבכל פניה לview השאלתא שמאחוריו טבלה מדומה זו רצה מחדש, ויש לעתים שהשאלה

שעומדת מאחוריו ה-view מורכבת מאד, חייב להיות שהדבר ישפייע על ביצועי המערכת.

מבנה VIEW

`CREATE VIEW view_name`

`AS`

שאילתת

[with check option]

[with read only]

דוגמאות:

יצירת view המציג את פרטי העובדים שיש להם פקס

`Create view emp_fax_v`

T-SQL

As

```
Select employeeID, lastName, firstName, birthdate  
From employees_tbl  
Where fax is not null
```

הצגת הנתונים מה view תבוצע על ידי הרצת השאלתה:

```
SELECT * FROM emp_fax_v
```

מכיוון שלא רשمنו את שמות העמודות שברצוננו שהview יכיל, שמות העמודות יהיו אוטומטית כשמות העמודות בטבלאות הבסיס.

דוגמא נוספת:

```
Create view qty_prod_v(prodName, sum_qty) as  
Select p.name, sum(o.qty)  
From product_tbl p inner join order_tbl o  
On p.prodID= o.prodID  
Group by prodName
```

בשאלתא זו שמות העמודות בטבלה המדומה יהיו: sum_qty ו- prodName

פקודות על VIEW

פקודות DML על VIEW כמו עדכון, הוספה ומיקעה אפשריות אך ורק במידה ו-

- ה view מורכב מנתונים של טבלה אחת.
- ה-view לא מכיל את המילה DISTINCT.
- ה-view לא מכיל GROUP BY.
- למשתמש ניתנה הרשאה גם לכך.
- ה-view מכיל את כל השדות המחייבים בהכנסת ערך (השדות שמוגדרים כ- not null) בטבלה המקורית או שהוגדר ערך בירית מחדל עבורם.

שינוי VIEW

```
ALTER VIEW view_name
```

שאלתה

```
drop view view_name
```

:View

CHECK OPTION

פעולות DML המפרות את תנאי ה - where של השאלה שהגירה את ה- view לא יאפשרו כאשר ב-view יהיה מוגדר . with check options

דוגמיה:

יצירת view של משכורות הגבוהות מ-10000.

```
Create view high_salary_v
```

As

```
Select *
```

T-SQL

From salary_tbl

Where salary > 10000

With check option

היות שה- view הוגדר **with check option** לא מפרות את תנאי
המשמעות כלומר:

א. לא ניתן להכנס משכורת נוספת נספחת חדש שגובחה פחות מ-10000.

ב. לא ניתן לעדכן משכורת קיימת לפחות מ-10000.

כך למשל אפשר לבצע את העדכון הזה:

UPDATE high_salary_v

Set empid = '301989897'

Where datehire = '01/03/2011'

UPDATE high_salary_v

Set salary = 4000

Where datehire = '01/03/2011'

אך לא ניתן לבצע את העדכון הבא:

תרגילים

לפניך בסיס נתונים המשמש לרישום שירותי טלפון שבוצעו ע"י לקוחות. לכל שירות נרשם: שעת התחלת השירות, שעת סיום השירות ומהיר עלות לדקה. יתכונו הנחות בהתאם לمسلسل ההנחה שנבחר ע"י הלקוחה. לכל לקוח מסלול ההנחה אחד בלבד.

לקוחות (קוד לקוח, שם לקוח, כתובת, עיר, טלפון, קוד הנחה)

Customers (custid, custname, address, city, phone, discountid)

שירות (קוד שירות, קוד לקוח, טלפון, תאריך ועתת התחלת, מהיר ועתת סיום, עלות לדקה)

Usage (usageId, custId, fromPhone, toPhone, timeStart, timeEnd, cost)

הנחות (קוד הנחה, שם מלא, אחוזה הנחה)

Discount (discountId, discountName, part)

צרי בכל תרגיל טבלה מדומה לפי המבנה המתבקש, התביסטי על הטבלאות של רישום השירות טלפון: צרי view בשם **AllCustomers** הציג שמות לקוחות, כתובות + עיר, טלפון ושם ההנחה שבחר הציגי גם

ללקוחות ללא הנחות. (בין 19:00-7:00): קוד לקוח.

(ב) צרי view בשם **NightUsages** הציג את פרטי השירות שבוצעו בלילה (בין 7:00-19:00): קוד לקוח.

שם לקוחות, טלפון, טלפון, תאריך, זמן התחלת, משך שירות.

(ג) כתבי **SumCost** שתציג לכל לקוח את משך הזמן של כל השירות שלו בכל חודש ואת מהירות לפני ההנחה.

ולאחריה. שמייל במחיר שירות ע"פ הדקות כפול 0.72agi. אל תשכח להציג גם לקוחות ללא הנחות.

(ד) **MaxNumUsage** הציג שמות 5 לקוחות שביצעו את מספר השירות הגבוה ביותר.

(ה) **MaxSumLengthUsage** הציג שמות 5 לקוחות שדיברו במשך זמן הארוך ביותר.

(ו) **MaxLengthUsage** הציג שמות 5 לקוחות שהichen את השירות הארוכות ביותר.

T-SQL

ג. – פְּרָוֹצְדּוֹרָה מְאוֹחֶסֶנֶת – Stored-Procedure

פְּרָוֹצְדּוֹרָה מְאוֹחֶסֶנֶת (Procedures Stored) הינה שאלת SQL המאווחסת בשורת של בסיס הנתונים. כמו בפְּרָוֹצְדּוֹרָה תכניתית רגילה, לפְּרָוֹצְדּוֹרָה המאווחסת יש שם ייחודי המזהה אותה, היא יכולה להגדיר פרמטרים, ועיקרא הוא גוף הפְּרָוֹצְדּוֹרָה – שהינו קוד המבצע פעולה אחת או מספר פעולות. הקוד ייכתב כМОבן בשפת SQL.

♦ דוגמה לייצירת פְּרָוֹצְדּוֹרָה

```
Create procedure select_HighBills @abovePrice  
As
```

```
Select billId, PhoneId, totalPrice  
From tbl_Bills  
Where totalPrice > @abovePrice
```

הפְּרָוֹצְדּוֹרָה מקבלת כפרמטר מחיר מקסימלי, ושולפת את כל החשבונות שמחירן גבוה מהמחיר שהתקבל כפרמטר

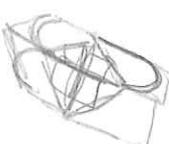
• מטרות השימוש בפְּרָוֹצְדּוֹרָות

1. פְּרָוֹצְדּוֹרָה (SQL) פְּרָוֹצְדּוֹרָה (SQL) פְּרָוֹצְדּוֹרָה (SQL) פְּרָוֹצְדּוֹרָה (SQL) פְּרָוֹצְדּוֹרָה (SQL)
2. פְּרָוֹצְדּוֹרָה יעליה פְּרָוֹצְדּוֹרָה יעליה פְּרָוֹצְדּוֹרָה יעליה פְּרָוֹצְדּוֹרָה יעליה פְּרָוֹצְדּוֹרָה יעליה
3. פְּרָוֹצְדּוֹרָה אוסף פְּרָוֹצְדּוֹרָה אוסף פְּרָוֹצְדּוֹרָה אוסף פְּרָוֹצְדּוֹרָה אוסף
- 4.
- 5.

עבודה

1. כתבי פְּרָוֹצְדּוֹרָה שתחזיר את שמות הלוקחות, ואת מספר החשבונות, ואת סכום החשבון המקסימלי של כל אחד מהם.

כתבו פְּרָוֹצְדּוֹרָה נוספת שתקבל כפרמטר קוד ללקוח, ותחזיר את מספר החשבונות שלו, ואת סכום החשבון המקסימלי.



כתבו את הפְּרָוֹצְדּוֹרָות ביעילות מירבית.

2. כתבו פְּרָוֹצְדּוֹרָות לתחזוקת הנתונים בטבלה טלפונים

- a. פְּרָוֹצְדּוֹרָה המשמשת להוספה רשומה
- b. פְּרָוֹצְדּוֹרָה המעדכנת רשומה בטבלה
- c. פְּרָוֹצְדּוֹרָה המוחקת רשומה מהטבלה

חשי אליהם פרמטרים יש להגדיר בכל פְּרָוֹצְדּוֹרָה.

3. כתבו קוד המפעיל את הפְּרָוֹצְדּוֹרָות הנ"ל.

T-SQL

7. פרטרים בפrozדורות

❖ ערכי בירית מחדל לפרטרים

ניתן להגדיר ערכי בירית מחדל לפרטרים, ואז אין חובה לשלוח ערך עבור הפרטר. הפרוזדורה תשתמש בערך שהוגדר כערך בירית מחדל לדוגמה:

```
create procedure EmployeesByCity  
@cityId int = 10,  
@departmentId int = 2  
as  
select *  
from tbl_Employees  
where cityId = @cityId
```

בהפעלת הפרוזדורה כך exec EmployeesByCity 5, 3 יחזיר רשימת העובדים הגרים בעיר 5 ועובדים במחילה 3.
בהתאם באה הפעלת הפרוזדורה כך exec EmployeesByCity 10, 2 יחזיר רשימת העובדים שגורים בעיר 10 ועובדים במחילה 2.

❖ דרכי לשילוח פרטיים

1. **לפי מקום** – כאשר שולחים ערכים בלבד, ה-SQL יכנס את הערכים לפרטרים לפי הסדר המוגדר בפרוזדורה בהתאם.

לדוגמה: exec EmployeesByCity 5, 3 – המספר הראשון, 5, יכנס לפרטר הראשוני, והמספר 3 יכנס לפרטר השני.

2. **לפי שם** – כאשר שולחים שם פרטר + ערך, הפרוזדורה תכנס את הערך לפרטר לפי השם שצוין, ולא תתחשב בסדר שבו נשלחו הפרטרים.

לדוגמה: exec EmployeesByCity @departmentId =3, @cityId = 5 – המספר 3 יכנס לפרטר המחלקה והמספר 5 ייכנס לפרטר עיר, בשונה מהסדר שבו הם מוגדרים בפונקציה.

יתרונות לשימוש בשילוח לפי מקום:

.1

.2

יתרונות לשימוש בשילוח לפי שם:

.1

.2

❖ פרטר output – זיהום און אוף

כאשר יש צורך שפרוזדורה תחזיר ערך בוודד בצורה שיהיה נוח להשתמש בזהות זאת ע"י פרטר מסוג output.

לדוגמה:

```
create procedure countEmployees  
@empCount int output  
as  
select @empCount = count(*)  
from tbl_employees
```

T-SQL קריאה לפורוצדורה:

```
declare @c int  
exec countEmployees @c output  
print @c
```

עבודה

1. כתבי פורוצדורה מקבלת ת.ז. של לקוחות ומוחירה את כל הפעולות שביצעו לקוחות זה. כמו"כ מוחיר הפורוצדורה בפרמטר `output` את היתר הנסיבות בחשבונו. אם לא נשלחה ת.ז. מוחיר הפורוצדורה את כל הפעולות.
2. כתבי פורוצדורה מקבלת תאריך המחללה ותאריך סיום. האידי רצוי בירית מחדל לפרמטרים אלו. הפורוצדורה מוחיר טבלה המסכמת כמה פעולות בוצעו בכל תאריך שנמצא בטוחה התאריכים שהתקבלו ומהו סכומן הכלול.
כתב 4 צורות שלicha שונות לפורוצדורה.
3. כתבי פורוצדורה מהזירה נתונים אודות טבלה מסוימת. הפורוצדורה מקבל שם טבלה ומחייר נתונים נתונים אודות השודות בטבלה ואודות המפתחות בטבלה.

T-SQL

בוחן בSQL על תרגיל מסמכים אחורוניים

הקיים את התשובה/ות הנכונה/ות:

(a) בטבלת מסמכים אחורוניים שהו בשימוש - איזו עמודה/ות תבחרי כמפתח ראשי:

1. מזהה משתמש
2. שם קבץ
3. תאריך ושעת גישה

(b) בטבלת העדפות המשתמשים - איזו עמודה/ות תבחרי כמפתח ראשי:

1. מזהה משתמש
2. טיפוס העדפה
3. ערך

(c) באלו מקרה/ים - יהו למשתמש רשותות פחות מהערך שנותן בהעדפותיו (X):

1. כאשר הוא משתמש תמיד באותו הקובץ.
2. כאשר הוא משתמש במספר קבצים קבועים - שקטן מ-X.
3. בכל X פחות אחד הקבצים הראשונים שיפתח
4. כל התשובות נכונות

(d) מי מהפקודות הבאות מתאימה למשפט DML

1. create
2. insert
3. delete
4. drop

(e) באלו צורה/ צורות ניתן לשЛОח פרמטרים ל프로그램ה:

1. בסדר אקראי - כאשר מספר הפרמטרים הנשלחים זהה למספר הפרמטרים שהפואציגורה מקבלת
2. לפי שם - בציון שם פרמטר = ערך , גם כאשר שלוחים רק חלק מהפרמטרים
3. לפי מקום - ללא ציון שם הפרמטר, גם כאשר שלוחים רק חלק מהפרמטרים

(f) איזו פסוקית/ות אינה/ חוקית/ות בויה - view:

1. top
2. order by
3. having
4. group by

ה. פונקציות

ב Sql-Server קיימת אפשרות ליצור פונקציות המאוחסנות בסיס הנתונים. קיימים מספר סוגים פונקציות שנבדלים זה מזה בסוג הערך המוחזר מהן.

פונקציות סקלריות (מחזירות ערך אחד)

דוגמא ליצירת פונקציה:

הפונקציה מקבלת תאריך ותו מפריד ומזהירה מחזרות המכילה את התאריך כשלקוי מופרדים ע"י התו שנסלה.

Create function funcdateformat (@mydate datetime,@seperator char(1))

returns nvarchar(20)

as

begin

return

convert(nvarchar(20),datepart(dd,@mydate)) + @seperator +
 convert(nvarchar(20),datepart(mm,@mydate)) + @seperator +
 convert(nvarchar(20),datepart(yy,@mydate))

end

קריאה לפונקציה שליחת פרמטר והצבת הערך המוחזר לתוך משתנה :

Declare @k nvarchar(20)

set @k=dbo.funcdateformat(getdate(),'*')

print @k

קיימים הבדלים בין האובייקטים השונים ש מספק SQL Server

Function	View	Stored-Procedure	
funcdateformat	תפקידם של פונקציות המוחזרות (לעומת פונקציות המוחזרות ערך אחד)	ערך מוחזר out	ערך מוחזר
select	פ. ר. ו. ו. select	לעומת פונקציות המוחזרות ערך אחד	DML Data Manipulation Language
select	פ. ר. ו. ו. select	לעומת פונקציות המוחזרות ערך אחד	שימוש ב T-Sql (גזרה SQL)
func	פונקציית	לעומת פונקציות המוחזרות ערך אחד	שימוש ב פרמטרים
func		לעומת פונקציות המוחזרות ערך אחד	פניה לאובייקטים אחרים

T-8Q7

עבודה

4. כתבי פונקציה המקבלת מספר טלפון ומחזירה האם הוא תקין.
5. כתבי פונקציה המקבלת מזהה פציינט ומחזירה את שמות הבדיקות שביצע כשהן משורשנות. כתבי פרוצדורה המחזירה את רשימת הפציינטים שביצעו יותר מבדיקה אחת. הפרוצדורה תחזיר טבלה שמכילת את השדות: קוד פציינט, ת.ז., תאריך ביקור אחרון ושמות הבדיקות שביצע הפציינט משורשנות.
6. כתבי פונקציה המקבלת מזהה תוצאה ומחזירה 0 אם התוצאה תקינה, 1 אם התוצאה מעל הנורמות, ו -1 אם התוצאה מתחת%nורמות. כתבי שאלתה המציגת את נמנוי כל התוצאות שאינן תקינות. העזרי בפונקציה.

T-SQL

פונקציות שמחזירות טבלה

לפניך 2 דוגמאות לפונקציות, שמודגימות שני סוגי פונקציות. (הפונ' נמצאות בקובץ בתיキית המורה וניתנות להרצאה).

• פונקציית inline

Create function visitsCount(@year int, @patientId int)

As

Return *table*

Select count(*)

From visits

Where patientId = @patientId and datepart(Year, visitDate) = @year)

מה מבצעת הפונקציה?

כתבו דוגמא לקריאה לפונקציה

• פונקציה שמחזירה ערך טבלה

Create function invalidResults (@patintId int)

Returns @t table(
 testId int,
 testName nvarchar(20),
 result int)

As

Begin

 Declare @lastTestDate datetime

 Select @lastTestDate = max(testDate) from tests where patientId = @patientId

 If @lastTestDate is not null

 Insert into @t

 Select testId, testName, result from tests inner join ~~x~~ on tests.testType =

 x.TestType

 Where tests.result not between x.min and x.max and testDate=

 @lastTestDate

 end

מה מבצעת הפונקציה?

כתבו דוגמא לקריאה לפונקציה

כתבו עבור כל משפט לאיזה סוג פונקציה מתאים:

1. תומך בפרמטרים

2. מחייב הגדרת משתנה מסווג טבלה במקטע הערך המוחזר

3. מכיל רק משפט Select אחד.
4. מבנה טיפוס הנתונים המוחזר מרומז על ידי המשפט return
5. יכול להכיל משפטי If-Sql
6. מחויר נתונים על ידי יצירת טבלה, והוספת הרשומות המתאימות לתוך הטבלה שתוחזר.
7. אינו מכיל begin ו end

שיעור בית:

1. כתבי פונקציה שמקבלת מזהה פציינט ומחזירה אותו ואת ילדיו, ממיינים לפי גילים. בוצי ב 2 צורות!
2. כתבי פונקציה שמקבלת מזהה פциינט ומחזירה את מספר הביקורים שביצעה כל משפחה בשנה האחרון.
3. כתבי פונקציה שהוחזירה טבלה ובה מספר הביקורים שביצעה כל משפחה בשנה האחרון. דאגי שמשפחה לא תופיע פעמיים.
4. כתבי פונקציה שמקבלת טווח חירגה מתאריך ועד תאריך ומחזירה את הנתונים הבאים:
שם פציינט, שם בדיקה, ערך, ערך תקין בתנאי שערך הבדיקה שלו חרג מהערך התקין בטווח החירגה.
אם ערך החירגה הנשלח שלילי תוחזרנה תוצאות שחרגו כלפי מטה, אם הוא חיובי תוחזרנה תוצאות שחרגו כלפי מעלה,
אם הוא 0 תוחזרנה כל התוצאות.
בכל המקרים תוחזרנה רק תוצאות שבין התאריכים שסופקו לפונקציה.

יש להדפיס את שיעורי הבית ולהגיש בתחילת השיעור הבא.

T-SQL

דוגמאות לפונקציות

לפניך 2 דוגמאות לפונקציות, שمدגימות שני סוגי פונקציות:

פונקציית inline

Create function visitsCount(@year int, @patientId int)

As

Returns

Select count(*)

From visits

Where patientId = @patientId and datepart(Year, visitDate) = @year)

מה מבצעת הפונקציה?

כתבו דוגמא לкриיאת פונקציה

פונקציה שמחזירה ערך טבלה

Create function invalidResults (@patintId int)

Returns @t table(

testId int,

testName nvarchar(20),

result int)

As

Begin

Declare @lastTestDate datetime

Select @lastTestDate = max(testDate) from tests where patientId = @patientId

If @lastTestDate is not null

Insert into @t

Select testId, testName, result from tests inner join x on tests.testType = x.TestType

Where tests.result not between x.min and x.max and testDate= @lastTestDate

end

מה מבצעת הפונקציה?

כתבו דוגמא לкриיאת פונקציה

כתבו עבור כל משפט לאיזה סוג פונקציה מתאים:

.1. תומך בפרמטרים

.2. מחייב הגדרת משתנה מסווג טבלה במקטע הערך המוחזר

.3. מכיל רק משפט Select אחד.

.4. מבנה טיפוס הנתונים המוחזר מרמז על ידי המשפט return

.5. יכול להכיל משפטי T-Sql

.6. מוחזר נתונים על ידי יצירת טבלה, והוספת הרשומות המתאימות לתוך הטבלה שותהזה.

.7. אין מכיל begin ו end

שיעור בית:

1. כתבי פונקציה שמקבלת מזהה פציינט ומחזירה אותו ואת ילדו, ממוינים לפי גילם. בוצי ב 2 צורות!
2. כתבי פונקציה שמקבלת מזהה פציינט ומחזירה את מספר הביקורים שביצעה כל משפחתו לשנה השנה.
3. כתבי פונקציה שהוחירה טבלה ובה מספר הביקורים שביצעה כל משפחה לשנה השנה. דאגי משפחה לא תופיע פעמים.

4. כתבי פונקציה שמקבלת טווח חירגה מתאריך ועד תאריך ומחזירה את הנתונים הבאים:
שם פציינט, שם בדיקה, ערך, ערך תקין בתנאי שערכ הבדיקה שלו חרג מהערך התקין בטווח החריגת.
אם ערך הבדיקה הנשלח שלילי תוחזרנה תוצאות שחרגו כלפי מטה, אם הוא חיובי תוחזרנה תוצאות שחרגו כלפי מעלה,
אם הוא 0 תוחזרנה כל התוצאות.
- בכל המקרים תוחזרנה רק תוצאות שבין התאריכים שסופקו לפונקציה.

יש להדפיס את שיעורי הבית ולהגיש בתחילת השיעור הבא.

T-SQL

בוחן ב SQL – ויו ופונקציות

1. אלו מילים חיבוט להופיע בפונקציה שמחזירה ערך סקלרי? סמני ליד כל מילה V או X.

returns .ג.

declare .א.

select .ז.

return .ב.

2. מה אי אפשר לבצע בפונקציה? סמני ליד כל אפשרות V או X.

כ. להפעיל פרצדורה

א. קירוץ

ד. להפעיל פונקציה אחרת

T-Sql .ב.

3. מה המעלות של פונקציה לעומת ויו? סמני ליד כל אפשרות V או X.

כ. ניתן לקבץ

א. ניתן להגדיר עליה אינדקס

ד. ניתן להשתמש ב T-Sql

ב. ניתן להשתמש בפרמטרים

4. מה תהיה התוצאה של הריצת הקוד הבא? בחרי בתשובה הנכונה

Create view V

As

Select Id, LastName + FirstName

From tbl_Students

Order by LastName

.א. יוצר ויו

.ב. תישלפנה כל הרשומות של התלמידים עם שם משורשר ממויינים בסדר עולה.

.ג. הרדעת שגיאה.

5. נחונה הפונקציה הבאה:

create function f (@word1 nvarchar(3), @word2 int, @num int)

returns nvarchar(200)

as

begin

declare @s nvarchar(200)

while @num > 0

set @s=@s+@word1+cast (@word2 as nvarchar(3))

return @s

end

מה יהיה פלט הפונקציה עבור ערכי הפרמטרים:

_____ @word1='abc', @word2=null, @num=3 .a

_____ @word1='12', @word2=3, @num=2 .b

_____ @word1='Hello', @word2=10, @num=1 .c

6. נחונה שאלתה. שכחבי אותה לפונקציה בשתי צורות התchapir

Select patientId, count(*) as visitsCount, max(visitDate) as lastVisit

T-SQL

```
From tbl_visits  
Where year(visitDay) = @year  
Group by patientId  
from tbl_visits
```

מבחן ב T-SQL - SQL מושגים, פונקציות וו'ו, פונקציות

* חלק א'

בחרי בתשובה הנכונה (8 מתוך 10)

1. מה מוחזר הביטוי null=null?

- א. true
- ב. false
- ג. לא true ולא false
- ד. שגיאה

2. מי יכול להפעיל פרגזורה

- א. פונקציה ופראזורה
- ב. רק פרגזורה
- ג. אף אובייקט
- ד. וו'ו, פונקציה ופראזורה

3. מה חוקי לבצע בו'ו?

- א. לשולף שדה חישובי ללא כינוי
- ב. לכתוב יותר משאללה אחת
- ג. למיין
- ד. לקבץ

4. איזה מההצבות הבאות אינה חוקית?

- א. declare @x int = 5
- ב. set @x = 5
- ג. @x = 5
- ד. select @x = result from tbl_results

5. מה הבדל בין פונקציה בתחביר inline לפונקציה בתחביר ארכ'?

- א. בInline א"א לכתחום פקודות T-SQL
- ב. בinline א"א לבצע מיוון
- ג. Inline מוחירה טבלה והאחרת מוחירה ערך סקלרי
- ד. רק בתחביר הארכ' ניתן להפעיל פקודות DML

T-SQL

6. במאי אפשר להשתמש בחולק ה From של שאלתה?

א. פרוצדורה, ויו ופונקציה

ב. ויו ופונקציה סקלרית

ג. פונקציה טבלאית ומשתנה טבלה

ד. ויו, פונקציה טבלאית ומשתנה טבלה

7. מהי הצורה הנכונה לכתיבת Case?

Case result when < 10 then 'low' else 'high' end

Case when result < 10 then 'low' else 'high' end

Case result when result < 10 then 'low' else 'high' end

Case when result < 10 then 'low' else when result > 10 'high' end

8. מתי לא נשימוש בוו?

א. כדי לפשט שאלתה מורכבה עם תת שאלתה, נוציא את תת השאלתה לוו

ב. כשנרצה להציג אינדקס על תוצאות של שאלתה

ג. כדי למנוע כפל קוד במקרה שאוותה שאלתה מבוצעת יותר מפעם אחת

ד. כדי שלא לכתוב קוד SQL בתחום תוכנית (כמו בC#), אלא לשמר את כל ה SQL בסיס הנתונים

9. מה הפלט של קוד ה SqI הבא?

```
Declare @s nvarchar(5)  
Set @s = 'abc'  
Set @s += 'def'
```

abcde ג.

א. שגיאה

def ז.

ב. abcdef

10. נתונה הפרוצדורה proc1 שמוגדרת כך:

```
create procedure proc1 @x int = 5, @y int  
as...
```

איזה מהקריאות הבאות לפרוצדורה אינה חוקית?

Exec proc1 10, 7 א.

Exec proc1 7 ב.

Exec proc1 @x=10, @y=7 ג.

Exec proc1 @y=7 ד.

❖ חלק ב'

כתבו נכון או לא נכון (16 מתוך 18)

1. פרוצדורה יכולה להחזיר ערך ע"י _____ return

2. פרוצדורה יכולה להחזיר ערך ע"י _____ output

3. פונקציה חייבת לקבל פרמטר _____

T-SQL

4. פונקציה חיובית להחזיר ערך _____
5. חובה להעביר ערך לכל פרמטר המוגדר בפראז'ורה _____
6. א"א לצרף טבלה שחזרת מפונקציה ע"י join _____
7. אפשר לצרף טבלה שחזרת מפונקציה ע"י join אך יש מתחת כינוי לטבלה החזרת מהפונקציה _____
8. ניתן לצרף טבלה החזרת מפונקציה ע"י join כרגע _____
9. כאשר משרשים ערך ל null התוצאה תהיה null _____
10. באותו select א"א לשולח חלקழותות לתוך משתנים, וחלק להציג בתוצאות השאלה _____
11. View יכול לשפר ביצועים אם מדירים עליו אינדקס _____
12. כאשר if מכיל יותר מפקודה אחת חובה לתחום את הפקודות בין begin ל end _____
13. כאשר פרז'ורה מכילה יותר מפקודה אחת חובה לתחום את הפקודות בין begin ל end _____
14. ב case חובה תמיד לציין else _____
15. ב case אם לא כתבנו else יוחזר null _____
16. ב case אם לא כתבנו else יוחזר 0 _____
17. הפונקציה cast משמשת להמרה בין נתונים _____
18. הפונקציה cast מקבלת שני פרמטרים נתון וטיפוס המופרדים ביניהם בפסיקים, לדוגמה: (cast('1234', int), _____

❖ חלק ג'

- כתבו את האובייקטים הבאים, וכן שורת קוד המפעילה אותם (4 מטרק 5)
1. פרז'ורה המקבלת קוד מסלול ושולפת טבלה עם השדות שנה וסכום ממוצע לחשבונית. כמו"כ מוחירה את מספר הלקוחות במסלול זה. אם לא נשלח מספר מסלול, תבצע השילפה למסלול מס' 10.
- _____

הפעלה:

2. כתבו פרז'ורת עדכון לטבלת חשבונות. הפרז'ורה מקבלת כל הפרמטרים הדרושים ותבצע את העדכון לפיהם.
לפני העדכון הפרז'ורה תבודוק שלא מעדכנים את הערך בשדה האם שולם מ"כן" ל"לא", מכיוון שפעולה זו אינה חוקית, במקרה כזו יודפס "פעולה לא חוקית" והעדכון לא יתבצע.
- _____

T-SQL

הפעלה:

3. כתבי ויו המציג לכל מסלול את סכום החשבונית המקורי לאותו מסלול לפי שנים, כל מסלול יישלח בעמודה נפרדת וכל שנה חוצג בשורה נפרדת.

לדוגמה:

Maslul_30	Maslul_20	Maslul_10	bill_year
720	NULL	250	2010
660	525	178	2011
338	480	330	2012

הפעלה:

4. כתבי פונקציה סקלרית המשמשת ליצור מספר טלפון חדש ללקוח חדש. הפונקציה תקבל קידומת ותיקון מספר רנדומלי ע"י הפונקציה Rand (פונקציה זו מחזירה מספר רנדומלי בין 0 ל 1 – שבר). יש לדאוג שהמספר שיוחזר לא יהיה קיים בטבלה, וכן שלא יתחל ב 0. הפונקציה תחזיר שרשור של הקידומת, מקף והמספר.

הפעלה:

T-SQL

5. כתבי פונקציה טבלאית השולפת נתונים ללקוחות: מזהה, שם מלא, תאריך ה策טרופת לחברה (תאריך החשבונית הראשונה) וכן שדה שיכיל את הערך "לקוח ותיק" במקרה שהלקוח ה策טרף לחברה לפני מעלה מ 5 שנים, "לקוח חדש" אם ה策טרף בשנה לאחרונה, ו Null בכל מקרה אחר.

הפעלה:

* חלק ד'

בחורי בהסביר המתחאים לשגיאות SQL הבאות (4 מתוך 6)

השגיאות:

- _____ Invalid object name 'all_visit' .1
 _____ Ambiguous column name 'patientId' .2
 _____ There is already an object named 'calc_age' in the database. .3
 _____ Create View or Function failed because no column name was specified for column .4
Procedure or function 'Delete_Patient' expects parameter '@patientId', which was not
_____ supplied. .5
Column 'last_name' is invalid in the select list because it is not contained in either an .6
_____ aggregate function or the GROUP BY clause.

ההסברים:

- א. לא ניתן שם לעמודה במשתנה מסווג טבלה
ב. שם העמודה מופיע יותר מפעם אחת בטבלאות השאלה, יש לציין במפורש מייזו טבלה הוא נלקח.
ג. שם הטבלה/פרוצדרה/אובייקט לא נמצא בסיס הנתונים.
ד. חסר return בפונקציה
ה. לא נשלח ערך לפרטור חובה
ו. לא ניתן לרשום ב select של שאלה שיש בה קיבוץ שם של שדה שלא מופיע ב Group ולא מופעלת עליו פונקציה צבירה
ז. לפרוצדרה ניתן שם לא תקין, לדוגמה השם מתחילה במספר.
ח. לא ניתן שם לעמודה חישובים בויו או בפונקציה טבלאית
ט. אובייקט בעל שם זהה כבר קיים בסיס הנתונים, א"א להריץ את משפט `Create`
י. בפרוצדרה לא מוגדר מה הטיפוס של הפרמטר

כמה!

T-SQL

1. Cursor - סמן

הינו טיפוס נתונים שמצויב על רשומה, ובכך מאפשר גישה לשורות ייחודיות של נתונים, וחוסף פעולה על סט של שורות (כמו שמתבצע בעת הפעלה משפטי Select, Update וכו'), באמצעות Cursor ניתן לטפל בכל שורה באופן ייחודי ולבצע פעולה שורה אחריו שורה. ניתן לראות בcursor תחילה של SQL למערכות.

בעת שימוש ב cursor יש לבצע את השלבים הבאים:

1. הגדרת-משתנה מסוג Cursor, והגדרת הרשות שעלייהן הוא יפעל.
2. פתיחת ה cursor, בשלב זה מתבצעת השליפה ואתחול ה cursor בפועל.
3. שליפת השורה הראשונה ע"י הפקודה Fetch Next
4. ריצה בולולאת While על השורות שעלייהן ה Cursor מצביע כל עוד קיימות שורות, וכיום ה cursor לשורה הבאה.
5. סגירת cursor ע"י Close Cursor
6. שחרור המשאבים של cursor ע"י Deallocate Cursor

לדוגמה:

```
DECLARE @name nvarchar(20)
```

```
DECLARE Employee_Cursor CURSOR FOR
SELECT LastName FROM tbl_Employees
```

```
OPEN Employee_Cursor
```

```
FETCH NEXT FROM Employee_Cursor INTO @name
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT @name
    FETCH NEXT FROM Employee_Cursor
END
```

```
CLOSE Employee_Cursor
DEALLOCATE Employee_Cursor
```

משתני מערכת המתפללים ב Cursor

- מחזיר ערך 0 כל עוד קיימות שורות ב cursor .cursor
- מחזיר את מספר השורות ב cursor .cursor - @@CURSOR_ROWS

עבודה

7. כתבי פרודורת עדכון לטבלת תוצאות, הפרודורה תבודוק שהתוצאה אינה מתחת ל 0, אחרת תכנס ערך 0 לשדה תוצאה.

T-827

כתב פרוצדורה נוספת המחייבת מזהה בדיקה, וערך מינימלי חדש. הפרוצדורה تعدכן את כל התוצאות הקיימות בטבלת התוצאות עבור הבדיקה זו בהפרש שבין הערך המינימלי הישן לחידש. לצורך העדכון הפרוצדורה משתמש בפרוצדורות העדכון של טבלת התוצאות.

8. כתבי פרוצדורה ששולפת עבור כל הפציינטים טבלה ובה שמות בני המשפחה שלהם. הפרוצדורה תציג ראשית את שם הפציינט, ואת נתוניו ואח"כ את רישימת בני משפחתו, העורי בפונקציה הקיימת
9. כתבי פרוצדורה שמדפיסה לכל פציינט את שמות כל הבדיקות שביצע כשהן משורשות.

ז. Trigger - מזניק

מזניק (Trigger) הינו פרוצדורה המופעלת אוטומטית בעת שינוי בטבלה מסוימת. מזניקים הם כלים רבי עזרה, עם זאת הם פועלים לפחות יותר מפרוצדורות, שכן נועד להשתמש בפרוצדורה כאשר UPDATE מזניק מופעל כש-SQL מבצע שאלחת פעולה בטבלה אליה המזניק מקושר. לדוגמה, אם מופעל שאלחת UPDATE בטבלה Employees, מזניק העדכון (אם קיים כזה) מופעל.

המזניקים גורמים ליצירת הטבלאות INSERTED או DELETED בהתאם לסוגם.

קיימות שלוש פעולות המעוררות את המזניק:

- הוספה - גורמת ליצירת הטבלה INSERTED המכילה את כל השורות שנוספו לטבלה.
- מחיקה - גורמת ליצירת הטבלה DELETED המכילה את כל השורות שנמחקו.
- עדכון - גורמת ליצירת הטבלאות INSERTED ו-DELETED המכילה את מבנה השורות לפני עדכון ו-INSERTED המכילה את מבנה השורות לאחר עדכון.

ניתן ליצור מזניק שמתפלט בכמה סוגים פעולה. כגון: מזניק המופעל בעת הכנסה או מחיקת רשומה מהטבלה. מזניק מופעל פעם אחת בכל שאלה גם אם היא מטפלת במספר שורות. לדוגמה: שאלחת המעדכנת 10 שורות תפעיל את מזניק העדכון פעם אחת בלבד.

- **הפונקציה update** – פונקציה זו מקבלת שם של שדה ובודקת האם הפעולה שהזינקה את הטריגר ביצעה שינוי בעמודה זו.

♦ דוגמה

מזניק המעדכן בטבלה מוצרים את כמות המלאי בהתאם לכמויות שנרשמה בהזמנה

```
Create trigger amount on orders
  For update,insert
  As
  Begin
    If update(quantity)
      Begin
        Update products
        Set
        products.unitinstock=products.unitinstock-
          (inserted.quantity-deleted.quantity)
        from inserted inner join products on
        inserted.prodid=products.prodid full outer join deleted on
        deleted.prodid= inserted.prodid
      end
    end
```

♦ שימושי המזניק

1. עדכון טבלאות או שדות התלויים ברשומה המעודכנת.

T-SQ1

2. תיעוד שינויים בסיסי הנטוניים.
3. בדיקות תקינות וחסימות שינויים בלתי חוקיים.

שיעור בית

1. הוסיף טבלת התרעות. ובה השדות סוג התרעעה (בדיקה כפולת/מצויה חריגה), מזהה פציגנט, מזהה בדיקה, תאריך, תוכן התרעעה.
לagi שבסכל מקרה שימושיים או מעכניים תוצאה בדיקה בלתי תקינה כתוצאה מתווסף התרעעה בטבלת התרעות, וכן בכל מקרה שפגיזנט מבצע את אותה בדיקה פעמיים באותו יום תווסף התרעעה מתחילה.
2. כתבי טריגר שיחסום את האפשרות להזין ת.ז. שאינה תקינה.
3. הוסיף טבלת ארכיון שתשמור את נתוני הבדיקות שנמחקו וכן את תאריך המחיקה.
כל בדיקה שנמחקה מטבלת הבדיקות תישמר בארכיון לפחות 3 חודשים.
לכן כל פעם שימושיים רשותם לטלטלת הארכיון יש לבדוק לאלו רשומות עברו כבר שלושה חודשים מהתאריך מחיקתן ויש להסיר מטבלת הארכיון.

T-8Q1

בוחן טרייגרים

סמי נכון או לא נכון: (1 רשות - 6 * (בסיס הנתונים - בנק)

1. הטריגרים הם אובייקטים בסיס הנתונים שמאוחסנים בעבור טבלאות ובעבור יוסט.
2. יצירת טריגר היא דרך טובה לביצוע שימרת היסטורית של נתונים בטבלה.
3. טריגר אחד יכול להגיב רק לפעולה אחת מתחום: הוספה עדכון ומהיקה- של הטבלה.
4. טריגר עדכון אחד יהיה מזנק בעבור כל עדכון על כל שדה ברשומה שהתעדנה.
5. בדיקת העمرדה שהתעדנה תישאלה בעזרת הפונקציה update
6. ניתן לבטל את הפעולה שעומדת לקרות בעזרת הפקודה rollback
7. הטבלה הזמנית inserted - יכולה להכיל רק רשומה בודדת.
8. ניתן לפנות לטבלה הזמנית לאחר שהושלמה פעולה הטריגר.
9. בעת עדכון מתמלאות 2 הטבלאות הזמניות: גם טבלת inserted וגם טבלת deleted.
10. כדי לשמר על תקינות שדות מחושבים - טריגר היא דרך טובה יותר מאשר פרוצדורה.
11. כדי לשמר על שלמות הקשרים בין הטבלאות - ועל החלטה מדורגת של שינויים - פרוצדורה מתאימה יותר מאשר טריגר.
12. בטריגר הראשון שכתבנו בתרגול שיעור - מהקנו בעבור כל לקוח את כל התנויות שקיימות לו - זה דוגמא לשימרה על החלטה מדורגת של השינויים ואכיפה שלמות הקשרים בין הטבלאות.
13. בטריגר האחרון שכתבנו בתרגול שיעור - בדקנו בעבור כל לקוח האם קיימות לו תנויות - ומונענו את המהיקה - זה דוגמא לשמירת התקינות של שדות מחושבים.
14. כדי שהטריגרים שכתבנו יעבדו גם במצב של ריבוי רשומות שנמחקו - علينا להימנע משימוש בהשחתת המזהה של הלוקוח לחוק משותה.
15. כדי שהטריגרים שכתבנו יעבדו גם במצב של ריבוי רשומות שנמחקו - علينا להימנע משימוש באופרטור IN.
16. אפשר לכתוב את כל פרוצדורות ההוספה ומהיקה שנכתבו בש"ב בטריגר - אבל את העדכון א"א מפני שהוא טבלת updated.

10. נפקח: סמי ו/או במקומם/ות המתאים:

- אם ניצור טריגר למנוע עדכון רשומה תנואה שאינה באותו יום עסקים -

נצטרך להשתמש ב moveDate(`moveDate`) update(`update`).
 נצטרך להשתמש ב datePart(`weekday`...)

בהצלחה!!

trigger, cursor – SQL בוחן

1. כאשר נשתמש ב cursor נבצע:

- הקצת ערכות רשומות ע"י כתיבת For select לאחר הגדרת הסמן
- פתיחה וסגירה של הסמן
- הרצת פרוצדורה עבור כל אחת מהרשומות.
- lolאה עם תנאי של כל עוד $@@fetch_status <> 0$
- ביטול הקצת הסמן בעזרת deallocate

2. כיצד נדע אם התעדכן השדה סכום תנועה בתוך טריגר?

a. If update(sum_move)

If not exists (select * from inserted I inner join deleted d on i.move_id =
d.move_id and i.move_sum = d.move_sum)

g. If update(select move_sum from deleted)

d. אין דרך לדעת

3. מתי יופעל הטריגר המוגדר כז:

Create trigger trg1 on table1 for update. Delete as ...

a. כאשר מפעילים אותו בצורה מפורשת ע"י קרייה

b. בכל ביצוע של פעולה נוספת עדכון או מחיקה בטבלה

g. בעת ביצוע פעולה נוספת בטבלה

d. בעת ביצוע פעולה עדכון בטבלה

h. בעת ביצוע פעולה מחיקה בטבלה

Sql Server - טיפול בשגיאות בסביבה Try catch .ח.

- TRY ○
- CATCH ○
- לכידת נתוני שגיאות ע"י ErrorNumber וכו'.
- Throw – מתיק הטריגר ○
- טבלת הודעות המערכת ○
- הוספה ומחיקה ○
- שימוש בהודעות ○

Try & catch הם דרך לטיפול בשגיאות, כל המשפטים שיימצאו בתחום TRY ייבדקו וברגע שתעלה שגיאה, השגיאה תטופל בתחום CATCH.

מבנה:

```
BEGIN TRY
    Sql statements
END TRY
BEGIN CATCH
    Sql statements
END CATCH
```

דוגמאות:

1. יש לציין את בלוק ה CATCH ישר אחרי ה END TRY (לא ניתן להכניס משפטי SELECT באמצע).
2. יש לבצע שימוש בטרנזקציות – במידה ופקודה אחת נכשלה, כל קודמותיה ייכשלו. אחרת, חלק מהפעולות מתבצעות ורק הפעולה עם השגיאה תטופל בהתאם ותכשל.
3. ניתן להפוך שגיאות שהתקבלו על ידי שימוש ב:
 - (-)ERROR_MESSAGE() – הודעה שהתקבלה
 - (-)ERROR_NUMBER() – מספר השגיאה
 - (-)ERROR_LINE() – אינדיקציה למספר השורה בה התקבלה השגיאה
 - (-)ERROR_PROCEDURE() – הפרוצדורה בה קרויה השגיאה
 - (-)ERROR_SEVERITY() – חומרת השגיאה
 - (-)ERROR_STATE() – במידה ויש DISTREBUTED DATE BASES

T-SQL

RAISERROR

דרך נוספת לטיפול בשגיאות, השגיאה עולה במצבה יזומה. ניתן לשלב את משפט ה – RAISERROR בבלוקים של TRY ו CATCH.

מבחן: RAISERROR יוצא מהבלוק SQL SERVER לא יבצע את שאר הפעולות בבלוק. ומציג הודעה שגיאה בהתאם. RAISERROR לא סגור טרנזקצייה ולא מבצע !ROLLBACK

RAISERROR (message_string | message_id | variable , severity , state[,])
ההודעה אשר תתקבל בחוץ ה-MESSAGE_STRING
MESSAGE_ID -> מספר הודעה אשר נוספה בצורה ידנית על ידי sp_addmessage
-> VARIABLE -> משתנה אשר יכול להיות מחרוזתי או יכול הודעה, אוו מספרי ויכול מספר שגיאה.
-> SEVERITY -> 10 - מידע בלבד, ללא צבע אדום
-> SHIGIOT -> 11-16
-> connection, dba, סוגר 19-25-
STATE -> מספר לזיהוי השגיאה במקומות שונים בקוד.
ניחס להוסף שגיאה ידנית בשימוש ב system procedure addmessage אשר נקראת :
מבנה:

EXEC sp_addmessage

```
@msgnum = value over 50000,  
@severity = value between 1 and 25,  
@msgtext = 'The message'
```

צפייה בהודעות הידניות תחכזע על ידי שליפה מטבלת sys.message:

SELECT *

FROM sys.messages

מחיקה של הودעה ידנית:

```
EXEC sp_dropmessage message_number
```

@@ERROR

משתנה גלובלי אשר מוחזיר את מספר השגיאה האחורה שהתרחשה.

0-> מעיד על הצלחה. כל מספר אחר -> מעיד על שגיאה.

ניתן לתחקור ביחסם כל פעולה האם שוניה מ-0. במידה שכנו – יש לבצע טיפול בהתאם.

בגירסאות 2005 מומלץ להשתמש ב@@ERROR.NUMBER@@@@ERROR.ERROR.

T-8Q7

טיפול בשגיאות - עבודה

◀ הودעות מערכת

1. הוספי לטבלת ההודעות את שתי ההודעות הבאות:

a. "לא ניתן להוסיף פעולה עבורה חשבון שאינו רשום במערכת."

b. "לא ניתן לרשום את הפעולה, מכיוון שהיא גורמת לחריגה שאינה מורשית בחשבון".

2. חשי וכתבו היכן משתמשי בהודעות אלו.

_____ .a

_____ .b

3. ממשי את השימוש בהודעות.

Try-Catch ◀

4. שני את המימוש של סעיף a, כך שבמקרה להחבטה על בדיקת נתונים בטבלה כדי להציג את ההודעה,

תחבטשי על נתונים שגיאת קשרי גומלין שיילכדו במקטע Catch.

T-SQL

ט. Transaction

דוגמה שבה יש לומר מה ייכנס לטבלה בסוף ביצוע הסקריפט.

שיעור בית

1. הוסיף לטבלת הוודאות הזעעה היכן תשמשי? הוסיף.
2. שני את 1 כך שבמוקם לבצע בדיקה שתמשי ב Try-Catch .
3. אחרי 2 מקומות במערכת בהם תרצה להשתמש בטרנווקציה.מתי יכול להיות מצב של טרנווקיות מקוננות? כיצד תפתרי.

ממשי.

עבודה

טנווקציות ↗

5. מצאי שני מקומות שבהם חשוב יותר להשתמש בטרנווקציה, שימוש לב Ci מדבר בקטע קוד שהיביך לבצע יותר פעולה אחת כדי שתידרש טנווקציה.

.a

.b

6. חשבו על מקרה שבו תבוצענה טנווקיות מקוננות

מה עלולה להיות הביעיתות בכך?

כיצד תפתרי את הבעיה?

7. ממשי את הטרנווקציות.

T-SQL

Sql – Dynamic Sql .

ב Sql Server קיימת אפשרות להריץ שאלתה המשמורה כמחרוזת ע"י הפקודה Exec. שאלתה זו מכונה שאלתה דינמית.

♦ דוגמה:

```
declare @s nvarchar(MAX) = 'select * from tbl_patients'  
exec (@s)
```

♦ שימושי לב!

1. אין אפשרות להתייחס למשתנים המוגדרים מחוץ למחרוזת בתחום המחרוזת!
2. אין אפשרות להריץ שאלתה שחלקה דינמי וחלקה קבוע. במקרה זה יש לשרש את החלק הקבוע חלק הדינמי כמחרוזת.

♦ היתרון בשימוש ב Sql דינامي

4. יש אפשרות שחלק מהשאלתה ישתנה לפי פרמטרים מהמשתמש.
5. חיסכון בקוד וקלות תחזקה – ניתן לבצע פעולות שטפלות במספר רב של טבלאות או אובייקטים אחרים בשורת קוד קצרה.

♦ החסרונות בשימוש ב Sql דינامي

1. שגיאות בקוד תגלינה רק בזמן הריצה ולא בזמן קומpileציה.
2. קוד פחות יעיל, מכיוון שמקומפל ועובד אופטימיזציה רק בזמן הריצה, ולא מראש.



טבלאות מערכת ופrozצדורות מערכת

שם	סקמה	סוג	עריך מוחזר	שדות חשובים
Tables	Information_Schema	טבלה		
Columns	Information_Schema	טבלה		
sp_pkeys	sys	פרוצדורה		
sp_fkeys	sys	פרוצדורה		
Routines	Informatio_Schem.	טבלה		

עבודה

1. חפשי בכל הטבלאות את התאריך '4-5-2011'. שימושי לב לבצע את החיפוש רק בשדות מסוג תאריך.

T-SQL

2. שמה של טבלת התוצאות השתנה מ `results` ל `examinationResults`. בצעי חיפוש בכל הפרוצדורות, אם אין משתמש בטבלה, אם כן עדכני בהן את שם הטבלה לשם החדש.
3. צרי פרוצדורה המחוללת פרוצדורות `Insert` פשוטה לטבלה מסוימת. הפרוצדורה מקבלת כפרמטר את שם הטבלה, ואת השם שרצוים להעניק לפרוצדורה החדשה, ובונה בעבורה את הפרוצדורה לפי שדות הטבלה ולפי המפתחות הראשיים בה.

יא. פרויקטים

I. פרויקט בנק

ניהולحسابות בנק

א. בני את הטבלאות הבאות:

1.حسابות

שם שדה	טיפוס	אפיקון מיוחד: מפתח ראשי/חויבה/איולץ
מוזהה		
שם משפחה		
שם פרטי		
שם מלא (שדה חישובי)		
ת.ז.		
טלפון		
מספר חשבון		
מספר סניף		
סוג חשבון (פרט/עסק)		
גובה מסגרת		

נתונים לדוגמה:

מוזהה	שם משפחה	שם פרטי	ת.ז.	טלפון	מספר חשבון	סוג חשבון	גוובה מסגרת	גוובה
1	כהן	אברהם		52626255 03-2344555	35422	פרט	0	1
2	קלין	יוסף		65765787 02-8876752	9876	פרט	8000	2
3	קלין	יוסף		65765787 02-8876752	90722	פרט	3000	3

T-SQL

10000	עסק	3	8724	077- 9862455	09998882	יש	חיים	4
-------	-----	---	------	-----------------	----------	----	------	---

.2. תנועות

שם שדה	טיפול	אפיון מיוחד: מפתח ראשי/חובב/אליז'
מזהה		
תאריך		
מזהה חשבון		
תאור מילולי		
סכום		
יתרה		

נתונים לדוגמה:

מזהה	תאריך	סכום	תאור מילולי	מזהה חשבון	מספר	יתרה
1	01/01/2011	200	הפקדת מזומנים	1	200	
2	01/01/2011	1000-	משיכה מכספומט	1	800-	
3	03/01/2011	75000	הפקדת שקים	4	75000	
4	03/01/2011	1700	העברה מהחשבון אחר	1	900	
5	04/01/2011	60000-	העברה לחשבון אחר	4	15000	
6	10/01/2011	100004.50	הפקדת שקים	2	100004.50	
7	10/02/2011	60900	הפקדת שקים	3	60900	
8	10/03/2011	12000	הפקדת שקים	2	112004.50	
9	10/03/2011	40000	הפקדת שקים	2	152004.50	

.3. סניפים

שם שדה	טיפול	אפיון מיוחד: מפתח ראשי/חובב/אליז'
מזהה		
מספר		
שם		

נתונים לדוגמה:

T-SQL

שם	מספר	مزזה
בני-ברק	20	1
ירושלים – מרכז	12	2
ירושלים – צפון	13	3

ב. הוסיף את האובייקטים הבאים לבסיס הנתונים:

1. כתבי View סיכום יומי המציג את הנתונים הבאים:
מזהה חשבון, ת.ז. לקוחות, של לקוחות, תאריך, סכום זיכויים, סכום חיובים, יתרה, חריגה מהמסגרת?
2. כתבי פרוצדורה שליפה המציג את הנתונים מטבלת פעולות. השדות: מזהה פעולה, שם מלא, מספר חשבון, מספר סניף, סכום, תיאור, יתרה.
הפרוצדורה מקבל כפרמטר מזהה פעולה (ב"מ [int]). אם לא נשלח מזהה תוחזרנה כל הרשומות.
3. כתבי פרוצדורה נוספת לטבלת תנועות. הפרוצדורה מקבל את נתוני כל השדות למעט שדה תאריך שיתעדכן באופן אוטומטי לתאריך הנוכחי. הפרוצדורה תוסיף את הנתונים, אם מספר חשבון אין קיים תזורך שגיאה.
4. כתבי פרוצדורת עדכון. הפרוצדורה מאפשר עדכון רק באותו יום עסקים (שבת וראשון – נחשיים يوم שני), לאחר מכן אין אפשרות לשנות נתונים.
כתבו פרוצדורה מחיקה, הממחק את הרשומה מטבלת הפעולות ותאכسن אותה בטבלת ההיסטוריה.
5. כתבי פונקציה המקבלת תאריך חריגה מהמסגרת, ומהזירה תוך כמה ימים כוסתה החריגה.
6. כתבי פונקציה הבודקת עבור תנועה שלילית האם היא מורשת.
7. כתבי פונקציה המקבילה לתאריך חריגה של תנועה מעוררת יותר מהסכום המקורי.
8. תנועה הינה תנועה העונה על אחד מהכללים הבאים:
 - a. החרגה בחשבון לא תהיה נמוכה יותר מהמסגרת המוגדרת בחשבון.
 - b. אם מדובר בסכום של עד 1000 ש"ם מעבר למסגרת, ואשר לקוחות זה אם היו חריגות הן כוסו תוך שבועיים.
9. כתבי טריגר המעדכן את יתרת החשבון בפעולות של הוספה, עדכון ומחיקה. חשבו כמה רשומות עלייך לעדכן. (השתמשי ב cursor).
10. כתבי פונקציה המקבלת תאריך ומספר ימים ומהזירה את סיכומי הפעולות מתוך ה View, אשר בוצעו במשך מספר הימים שהתקבל כפרמטר מאו התאריך הנשלחה.
11. כתבי פרוצדורה המשמשת לאיתור הלבנות הון.
הפרוצדורה צריכה חיפוש באופנים הבאים:
 - a. תזהה לקוחות אשר להם יותר מחשבון אחד, ואשר בוצעו בהם באותו יום עסקים פעולות בסכומים של מעל 50000 ש"ם שהתרחשו לפחות פעם אחת בחודש במשך שלושה חודשים רצופים. השתמשי ב cursor .
 - b. תזהה חשבון אשר בהם פעולות של זיכוי בסכומים של מעל 50000 ש"ם, ובאותו שבוע פעולות חיוב בסכומים של מעל 50000 ש"ם.

7807

2. פרויקט מעבדה

A. בניית DB

1. בני DB חדש בנושא תוכאות מעבדה, הטלألوות:
 - a. פציינטימ – מזהה, מספרי זהות שם פרטי ושם משפחה.
 - b. סוגי בדיקות – קוד, תאור וטוווח תוכאות.
 - c. ביקורתים – מזהה, מספר זהות ותאריך.
 - d. تוצאות – מזהה, ביקור, סוג בדיקה ותוצאה.
2. הוסיף לטבלת פציינטימים עמודה מחושבת שתציג את שם הפרטוי והמשפחה משורשים.
3. הוסיף view שיציג את ממוצע התוצאות עבור כל בדיקה וכן עבור כל ממוצע את הערכים: High, Low, Normal
4. הוסיף טבלת היסטוריה - עברו:
 - a. טבלת פציינטימ – לשדה שם משפחה
 - b. ולטבלת סוג בדיקות לשדה של טוווח תוצאות, מבנה הטבלה: שם טבלה, שם שדה, תאריך, ערך ישן.

B. פונקציות ופרוצדורות

1. הוסיף טבלת התראות, השדות: קוד, פציינט, סוג בדיקה, תוצאה, מס' טלפון
 - a. צרי פרוצדורה שתקבל פציינט, סוג בדיקה, ותוצאה, הפרוצדורה תבודוק האם צרי להכנס התראה (במידה שהතוצאה נמוכה או גבוהה מהטוווח). במידה שיש צורך להכניס התראה הפרוצדורה תחזיר את מספר הטלפון של הפציינט, אחרת הפרוצדורה תחזיר מחזורת ריקה.
 - b. הפעיל את הפרוצדורה הנ"ל מטריגר על הכנסת רשומה לטבלת תוצאות ובמידה שהזרם מספר טלפון, הכנסי רשומה מתאימה לטבלת התראות.
2. כתבי פונקציה שמקבלת: סוג בדיקה ותוצאה ומחזירה בהתאם את הערכים high, low, normal.
פתחי את view שיצרת בחלק הקודם ושני אותו שיישתמש בפונקציה.
 3. צרי שליפה לדוח במבנה הבא:
עמודות: קוד פציינט, שם פציינט – וכן עבור כל סוג בדיקה עמודה בוליאנית. הדוח יציג עבור כל לקוח את הערך שnez עבור בדיקות שהלקוח ביצע אי פעם עם את הערך false עבור בדיקות שהלקוח אף פעם לא ביצע.
4. כתבי פונקציה שתתקבל כפרמטר מספר בדיקה ותחזיר טבלה במבנה הבא:
שם פциינט, סוג בדיקה, ממוצע תוכאות לבדיקה, כמה פעמים שהבדיקה בוצעה, ומספר הביקור האחרון בו בוצעה הבדיקה מטבלת תוכאות עבור סוג הבדיקה שהתקבל כפרמטר,

T-SQL

- במידה שהתקבל כפרמטר הערך 1- יוחזרו רשומות עבור כל סוג הבדיקות.
בטבלה יופיעו רק פצינטים + בדיקות שקיימות להם התראות בטבלת התראות.
5. כתבי פרוצדורה השולפת מן הטבלה הנ"ל רשומות עבור כל הבדיקות ומציגו רק את הפסיכינטים שהמוצע שלהם בכל בדיקה גבוהה מן הטווח.
6. כתבי פרוצדורה נוספת, המקבלת כפרמטר מספר ימים, הפרוצדורה תשלוף מן הטבלה הנ"ל רשומות עבור כל סוג הבדיקות ותציג רק רשומות שבן הביקור היה במהלך הימים הקודמים לתאריך הנוכחי, כאשר מספר הימים הקודמים הוא הפרמטר שהתקבל.
7. כתבי פונקציה שתחזיר טבלה מבנה הבא:
מספר ביקור, קוד בדיקה, ממוצע תוצאות ועומדת תואר מסווג (50) nvarchar
הfonkziaה תקבל 2 פרמטרים בוליאניים: קבץ פצינטים, קבץ בדיקות.
- א. במידה שיש לקבץ פצינטים:
הfonkziaה תחזיר מטבלת תוצאות את ממוצע התוצאות לכל פצינט בכל סוג הבדיקות, ואת מספר הביקור הראשון בו ביקר הפסיכינט במעבדה, בעמודת התיאור יופיע שם הפסיכינט.
- ב. במידה שיש לקבץ בדיקות:
הfonkziaה תחזיר מטבלת תוצאות את ממוצע התוצאות לכל סוג בדיקה עבור כל הפסיכינטים, ואת מספר הביקור בו בצעה הבדיקה לראשונה, בעמודת התיאור יופיע תאור הבדיקה.
8. כתבי פרוצדורה שתשתמש בfonkziaה הנ"ל, הפרוצדורה תקבל את אותם פרמטרים שהfonkziaה מקבלת + פרמטר נוסף מספר שנים, הפרוצדורה תציג מתוך תוצאות הfonkziaה רשומות רק עבור ביקורים במספר השנים האחרונות, כאשר מספר השנים הוא הפרמטר שהתקבל.
9. כתבי פרוצדורה שתחזיר את המבנה הבא: עמודת קוד בדיקה, עמודה בוליאנית שתכיל האם כל התוצאות של הבדיקה בטווח, עמודה שתכיל את מספר התוצאות של הבדיקה שגובהן מהוות.
10. הוסיף לטבלת פצינטים עמודה parented שתכיל את קוד הפסיכינט של האימה של הפסיכינט.
א. שכותבי פרוצדורה שתחזיר מטבלת תוצאות את כל הפסיכינטים שאם ביצה בדיקת דם והතוצאה הייתה בטווח (המבנה: קוד הפסיכינט, שם הפסיכינט ועומדה שתכיל את הערך 'אימו של + [שם הפסיכינט] + ביצה בדיקת דם') וגם את כל האמהות שאחד מילדיהן ביצע בדיקת המוגלובין שגובהה מהטוווח (המבנה: קוד הפסיכינט, שם הפסיכינט, ועומדה שתכיל את הערך 'לאחד מילדיה של + [שם הפסיכינט] + ביצע בדיקת המוגלובין והתוצאה הייתה גבוהה מהטוווח')

T-SQ/

- o. כתבי פרוצדורה נוספת שתבצע את שלב 2 אבל תחזיר רק את קוד ושם הפעינת ותחזיר רק את הפעינות מ 2 הקבוצות שהו להם לפחות 2 תוצאות שగבוות מהטווות.
11. הוסיף לפרוצדורה מהתרגיל הקודם שמחזירה עבור כל פצינט את תאריך הביקור הראשון שלו, שתחזיר עבור כל פצינט גם את מספר הילדים שלו.
12. כתבי פרוצדורה שתחזיר את כל הבדיקות שיש עבורן תוצאות מהשבוע הראשון בו בוצעו במעבדה בדיקות לראשונה, וגם את כל הבדיקות שיש עבורן רק תוצאות שגבוות מהטווות המבנה: קוד בדיקה, שם בדיקה, תאריך: 'מהשבוע הראשון \ רק תוצאות גבוהות'
13. עבור הנ"ל החזיר רק את הבדיקות שעונთ לשתי התנאים, וכן רק בדיקות שלא עונת לאף אחד מן התנאים.

T-SQL

ג. שרשוי

```
declare @checkName nvarchar(max)
set @checkName = ''
select @checkName = @checkName + Name + ', ' from dbo.ChecksTypes
select @checkName

(select list from
(SELECT name + ','
--AS [text()]
from dbo.ChecksTypes
FOR XML PATH
('')) l(list))

--pivot העמודות ידועות מראן
select * from
(select ct.name, cv.checkValue from dbo.ChecksValues cv inner join
dbo.ChecksTypes ct
on cv.CheckTypeID = ct.Code ) q
pivot
( max(CheckValue) for q.name in
([Hemoglobin],[Luctoz],[Loycocites],[trombosites])
) as pvt

--צירה דינמית של העמודות
declare @checktypes nvarchar(max), @str nvarchar(max)

(SELECT '['+name + '],'
AS [text()]
from dbo.ChecksTypes
FOR XML PATH
('')) l(list)

select @checktypes
--select substring(@checktypes, 0, len(@checktypes))
set @str = 'select * from (select ct.name, cv.checkValue from
dbo.ChecksValues cv inner join dbo.ChecksTypes ct
on cv.CheckTypeID = ct.Code ) q
pivot
( max(CheckValue) for q.name in
(' + substring(@checktypes, 0, len(@checktypes)) +'')
) as pvt'
```

T-SQL

```
print @str  
exec (@str)
```

7. שרשור ומספור ניסיון לרקורסיה

1. כתבי פרוצדורה ששולפת עברו כל ביקור את כל הבדיקות שנעשו בבדיקה זה. המבנה: שם פצ'ינט,
תאריך ב'יקור, ושמות כל הבדיקות משורשות.
2. כתבי פרוצדורה שתקבל קוד בדיקה ותחזיר עבורה את כל התוצאות הנמכות מהטוווח שקיימות
מסוג הבדיקה שהתקבל, הציגי בשליפה עמודה נוספת נוספת שתציג את התוצאות כאשר הנמכה
bijuter = 1 והגבוהה מבין כל התוצאות שנמכות מהטוווח = מספר התוצאות שנמכות מהטוווח
3. כתבי פונקציה שתקבל קוד פצ'ינט ותחזיר את כל צאצאיו(בניים, נכדים , נינים וכו')

ה. רקורסיה ו pivot

1. לפני הקוד של הרקורסיה שהראינו בצעי את תרגיל 4 בעזרת הקוד הזה.
 2. לפני הקוד של `htosonic`, כתבי פרוצדורה שתקבל קוד פצ'ינט, ותחזיר עבורו את כל מספרי הביקורים בעמודות, ועבור כל ביקור את מספר הבדיקות שנעשו באותו ביקור
 3. כתבי פרוצדורה שמקבלת מחרוזת של קוד פצ'ינטים מופרדים בפסיקים ומחזירה את תאריך הביקור האחרון שקיים במערכת עבור אחד מן הקבוצות
 4. כתבי פרוצדורה נוספת שתקבל קוד פצ'ינט, ותשלה לפרקודה הנ"ל את כל המשפחה של הפצ'ינט מופרדים בפסיקים, ותחזיר עבור המשפחה את תאריך הביקור האחרון שקיים לאחד מהמשפחה.
 5. הוסיף לטבלת תוצאות שדה דרגה, כתבי פרוצדורה שתקבל קוד פצ'ינט וקוד בדיקה ותעדכן עבורו את השדה דרגה כאשר התוצאה הגבוהה ביותר = 1 והנמוכה ביותר = מספר התוצאות של הפצ'ינט מהבדיקה.
- רשות: נסוי לדרג בקרה הבהא: כאשר כל הנמוכים מהטוויה מתחילה מ-1- ועד (מספר הנמוכים-)
אלו שבטווח = 0 וכל הגבויים מ-1 ועד מספר הגבויים
6. כתבי פרוצדורה שתקבל קוד פצ'ינט ותחזיר את שמות ילדיו כשביניהם המילה `and`, ז"א עבור פצ'ינט מסוים שיש לו 2 ילדים הפרוצדורה תחזיר:
`'yosi kohen and esti kohen'` (השתמשי בקוד שהציגנו)
 7. כתבי פרוצדורה שתקבל קוד בדיקה ותחזיר את כל התוצאות מקוד הבדיקה שהתקבל שגדלות מהטוויה + שדה דירוג, שידרג את התוצאות בסדר יורד (ז"א התוצאה הגבוהה ביותר תמוספר ב-1, והאחרונה תהיה מספר התוצאות הגבוהות מהטוויה שקיימות לסוג הבדיקה)

T-SQL

יב. מבחנים

1. מבחן - 4 תרגילים מהברת אירוקס

תרגיל 1 - ג'ינרט קורסים עפ"י תקופות

נתונה טבלה המגדירה סוגים קורסים (CoursesTypes). עבור כל סוג קורס (CourseType) היא מגדרה באיזה תאריך יתקיים הקורס הראשון בסדרה (FirstCourseDate) ועד איזה תאריך יתקיימו קורסים מן הסוג זהה (MaxCourseDate).

בנוסף, ישנה טבלה (CoursesDays) המגדירה באילו ימים בשבועו יתנהל כל סוג קורס (מכילה את השדות: CourseTypeID ו-Day (סוג int)).

לדוגמא:

בטבלת סוג קורסים:

ID	CourseType	FirstCourseDate	MaxCourseDate
101	גרפיקה	1.7.2010	1.10.2030
102	מחשבים	20.5.2011	1.5.2015

ובטבלת ימי הקורסים:

CourseTypeID	Day
101	2
101	3
102	2

כלומר: קורס גרפיקה מתרחש בכל יום שני ושלישי בשבועו וקורס מחשבים מתרחש בכל יום שני בשבועו.

א. יש לכתוב פרוצדורה שמנήה לטבלה שלישית (Courses) את כל רשימת הקורסים שיתקיימו עבור כל

סוג קורס ובאיזה תאריך הם יתקיימו. הטבלה מכילה את השדות הבאים:

1. CourseID - מזהה הקורס (PK)

2. CourseTypeID - מזהה סוג קורס (מקשר לשדה ID בטבלת סוג קורסים)

3. CourseDate - התאריך שבו יתרחש הקורס

לדוגמא: עבור הדוגמא הקודמת, הפרוצדורה תכניס:

1.7.2010 ,101

2.7.2010 ,101

8.7.2010 ,101

9.7.2010 ,101

.....

T-SQL

.....
20.5.2011 ,102.
27.5.2011 ,102
..... ,102

פתרון אופטימאלי לתרגיל של ייצור טבלת תאריכים

--להביא לטבלה זמןית עבור כל تاريخ תחילת קורס את تاريخ השבת שלפני הקורס

```
SELECT CourseTypeID, CourseDay, CourseFirstDate, CourseLastDate,  
startDay = Datepart(weekday, CourseFirstDate)  
, startDate = dateadd(day, Datepart(weekday, CourseFirstDate) * -1  
,CourseFirstDate)  
    into #CoursesDaysStatrtDays  
    from CoursesDays  
    inner join CoursesTypes on CoursesDays.CourseTypeID =  
CoursesTypes.ID  
    declare @weeksCounter int, @maxWeeks int  
    set @weeksCounter = 0  
    מציאת מספר שבועות מקסימלי שהיו בקורסים--  
    select @maxWeeks = max(datediff(week, CourseFirstDate,  
CourseLastDate)) from dbo.CoursesTypes  
    delete from Courses  
    while(@weeksCounter <= @maxWeeks)  
    begin  
        insert into dbo.Courses  
        select CourseTypeID, dateadd(day, (7*@weeksCounter) +  
CourseDay,startDate) from #CoursesDaysStatrtDays  
            where dateadd(day, (7*@weeksCounter) + CourseDay,startDate)  
>= CourseFirstDate and dateadd(day, (7*@weeksCounter) +  
CourseDay,startDate) <= CourseLastDate  
            set @weeksCounter = @weeksCounter + 1  
    end
```

T-SQL

תרגיל 2 – דוח אובייקטים ותפקידים למשתמש

נתונות הטבלאות הבאות:

MeetingID, Subject, Manager, Status Meetings .1

ReminderID,Subject,To,From,Status Reminders .2

TaskID,Subject,Description,Responsible,Owner, Status Tasks .3

(סטטוסים והतואר שלהם) Statuses .4

MeetingID, UserID (משתתפים בפגישה) Participant .5

UserID, UserName (משתמשים במערכת) Users .6

א. יש ליצור את הטבלאותocr (ключи, indexes וכו')

ב. כתבי SP המקבלת שם משתמש ומחזירה את רשימת כל האובייקטים בהם אותו אדם משתתף באיזה שהוא תפקיד, הנושא, הסטוס והתפקיד בו הוא משתתף.

לדוגמא:

סוג אובייקט	נושא (Subject)	סטטוס (Status)	תפקיד (Role)	משימה (Owner)
משימה	xxx	חדש	Owner	מטייל המשימה
תזכורת	yyy	נקראה	To	מקבל התזכורת
משימה	yyy	מאושרת	Responsible	אחראי
פגישה	zzz	נערכת	Participant	משתתף (مطلوب)

סוגי האובייקטים הקיימים:

פגישה - אובייקט מטבלת פגישות

תזכורת - אובייקט מטבלת תזכורות

משימה - אובייקט מטבלת משימות

סוגי תפקידים:

בפגישה: משתתף (مطلوب), מנהל (Manager)

במשימה: אחראי (Responsible), מטייל המשימה (Owner)

בתזכורת: השולח (From), הנמען (To)

ג. כתבי SP זהה לקודם אלא שיחזר את הרשימה עבור כל האנשים בארגון שהשתתפו באיזה שהוא אובייקט.

ד. שכתבו את ה-SP הקודם כך שאם בעtid תתווסף עוד טבלה אובייקטים מסווג נוסף או תפקיד נוסף באחד מן הטבלאות, לא יצטרכו לשנות את הפרוצדורה (ניתן להווסף טבלאות או כל רעיון אחר...).
לדוגמא: תתווסף טבלת הרצאות ובה מרצה ומשתתפים או יתווסף שדה בודק לטבלת משימות.

תרגיל 3 – רשומות מוכלות ורשומות כפולות

נתונה טבלה עם 3 שדות:

PK -ID

T-SQL

int - From

int - To

השדות From ו-To מהווים טווח של מספרים כאשר תמיד-h-From יהיה קטן מה-h-To. כתבי SP שמוצא עבור כל רשומה האם טווח המספרים שלה מוכל בטווח המספרים של רשומה אחרת. ה-SP יוסיף עמודה נוספת לטבלה (אם היא עדין לא קיימת) ויכתוב בה "מוכלת" עבור כל רשומה שטוווח המספרים שלה מוכל בטווח של רשומה אחרת, ו"אינה מוכלת" אם הטווח אינו מוכל באף רשומה. לדוגמה, נתונים המספרים הבאים:

ID	From	To
100	50	150
101	50	160
102	60	150
103	25	70

במקרה זה:

102 מוכל ב-100 וב-101

100 מוכל ב-101

101 ו-103 אינם מוכלים באף רשומה

את המילים "מוכלת" ו"אינה מוכלת" יש לנקות מטבלת תרגומים (Contain) שמכילה:

ID	Description
1	מוכלת
2	אינה מוכלת

ב. מחקי את כל הטווחים הכפולים שמופיעים בטבלה כך שיישאר מכל טווח רק רשומה בודדת.

ג. שמרי בטבלה נפרדת (טבלת סטטיסטייה) עבור כל טווח מספרים שהייתה בעבר רשומות כפולות, מהן כמהן הרשומות הכפולות שנמחקו עברו בסעיף הקודם. בצעז זאת ע"י טריגר כך שהמידע יישמר גם אם המחיקה לא תבוצע ע"י h-SP הנ"ל. יש להקפיד שבטבלת הסטטיסטייה לא יהיה שום טווח מספרים המוכל בטווח אחר.

טבלת הסטטיסטייה תכיל את השדות הבאים: From, To, DeletedCnt.

תרגיל 4 – ממוצע ציונים לתלמיד

נתונות שלוש טבלאות:

1. Students - תלמידים (StudentID, Name)

2. Courses – קורסים (CourseID, Name)

T-SQL

2. CoursesGrades – ציונים בקורסים (StudentID, CourseID, Grade)

כתב שאלת שתרצה עבור כל תלמיד מה ממוצע הציונים שלו ובאיזה קורס קיבל ציון הקרוב ביותר לממוצע ומה היה הציון שלו בקורס זה. אם אין שם קורס שבו ציון הקורס קרוב לממוצע (פלוא מינוס 10), לא יוחזר קורס וציון.

לדוגמא:

בטבלת תלמידים:

StudentID	Name
1	משה
2	דוד

בטבלת קורסים:

CourseID	Name
1	קורס גרפיקה
2	קורס מחשבים
3	קורס פילוסופיה
4	קורס מתמטיקה

בטבלת ציונים:

StudentID	CourseID	Grade
1	1	100
1	2	50
2	1	100
2	2	85
2	3	90
2	4	90

עבור הטבלאות הנ"ל תחזיר התוצאה הבאה:

StudentName	AvvargeGrade	NearestCourse	GradeInNearestCourse
משה	75		
דוד	91	קורס פילוסופיה	90

T-SQL

2. מבחן אמריקאי

מבחן סיכון ב sql server

סמיינר ב

בawsview מאוחסנים הנתונים בנוסף להיוותם שמורים בטבלה.

1. המשפט הבא:

```
create view v_Test as select top 100 * from courses order by  
CourseTypeId
```

i. התchapיר אינו תקין – והוא לא יקומפל כלל.

ii. הוא יתקמפל בהרצתו – תישלפנה 100 רשומות מטבלת קורסים שהCourseTypeId הנמור ביותר.

iii. הוא יתקמפל אך בהרצתו – תישלפנה 100 רשומות אקרואיות מטבלת קורסים.

2. מעילות שגורת מאוחסנות:

i. שיתוף קוד עבור תוכניות שונות.

ii. ניתן לצרף טבלה לטבלה שחזרה מה SP בחזון פשוט.

iii. משתמש יכול להריץ פרוצדורה גם אם אין לו הרשות גישה לטבלאות.

iv. משפרות ביצועים, בכך שפונקציית שרת פעם אחת ומחזירות תשובה אחת במקום כמה.

v. פרוצדורות יכולות לבצע הן משפטי DDL והן משפטי DML

drop Procedure EmployeesSelect as select * from employees 3. המשפט הבא:

i. יגרום למחיקת פרוצדורת EmployeesSelect

ii. יגרום למחיקת הנתונים בטבלת employees

iii. המשפט אינו תקין ולא יבוצע כלל.

4. הפלט של המשפטים הבאים:

```
declare @s nvarchar(max)  
declare @str nvarchar(max)=' Good Luck'  
set @str =@s+ @str  
print @str
```

i. התchapיר אינו תקין – ולא יהיה פלט כלל.

ii. התchapיר תקין – ולא יהיה פלט כלל.

iii. הפלט יהיה: ' Good Luck ' (בל' הגרש שבצדדים)

5. הפלט של המשפטים הבאים:

```
create procedure sp(@i int output)  
select @i=count(*) from Courses  
go  
declare @i int
```

T-SQL

```
exec sp @i  
print @i
```

- i. קיימת שגיאה תחבירית ביצירת הפרוצדורה
- ii. קיימת שגיאה תחבירית בהרצת הפרוצדורה
- iii. לאחר תיקון התחריר (הין שצרי - אם צרי), הפלט (או חלק ממנו) הוא הרשומות

בטבלת Courses

6. כאשר הוי כבר קיים, הפלט של המשפטים הבאים יהיה:

```
create view v_Test as  
    select * from courses  
go  
select @@error  
if @@error = 0  
    print 'succeeded'  
else  
    print 'failed'
```

- i. ב>Show results – יוצג במספר השגיאה הערך אפס.
- ii. ב>Show results תוצג תחילת הודעת המערכת המתריעה שכבר קיים אובייקט בשם זה.
- iii. ב>Show results תוצג גם כן המילה failed.

7. מעילות הפונקציות:

- i. פונקציות יעילות מוגו בכר שמקבלות פרמטרים.
- ii. ניתן לצרף טבלה לטבלה שחזרת מהפונקציה בחזון פשוט.
- iii. פונקציה יכולה לקרוא לprocדורה.
- iv. פונקציות יעילות מוגו בכר שיוכלו להפעיל משפטי DML.
- v. פונקציות יעילות מוגו בנווגע לשילוב קוד (משפטי sql-t).

8. מה מתבצע ע"י הטריגר הבא:

```
create trigger Trag2 on Emolyees  
for update,insert as  
    if update(lastdate)  
        rollback Transaction
```

- i. מבטל פעולות הכנסת לטבלה Emolyees
- ii. מבטל פעולות עדכון לטבלה Emolyees
- iii. מבטל פעולות עדכון על عمودת lastdate אך מאפשר פעולה הכנסתה.

T-SQL

- .iv. מבטל פעולות עדכון בעת שינוי בעמודת lastdate.
- .v. מבטל פעולות הכנסה בעת שינוי בעמודת lastdate.
- .vi. אינם מזנק בעת מחיקה מטבלת Emolyees

9. עברו אלו פרמטרים תחזיר הפרוצדורה הבאה תוצאות:

```
alter Procedure ProcExam  
@sql nvarchar(max)=''  
as  
set @sql='select * from Courses where ' + @sql  
exec (@sql)
```

- i. כל תנאי רשומות קורסים עונות עליון
- ii. מחוזצת ריקה
- iii. null
- iv. המחרוזת '1=1'
- v. כאשר לא ישלח פרמטר כלל

10. אופרטורים שפועלים בין 2 טבלאות:

- i. Join
- ii. For
- iii. Pivot
- iv. unPivot

pivot \ unPivot .11

- i. תמיד ניתן לשחזר פלט של pivot בעזרת הפוקודה unPivot
- ii. ניתן לשחזר פלט של pivot בעזרת הפוקודה Pivot ובתנאי שהאגרציה היא:

Max/Count

- iii. כל העמודות שנמננות בתת השאלה הראשונה ואין נמנות בתת השאלה השנייה

תהפוך לכותרות שורה לאחר ביצוע ההצלה.

12. פונקציות ופרוצדורות מערכתי:

- i. ניתן לפרמטר את עיצוב התאריך בעזרת הפונקציה cast.
- ii. ניתן לפרמטר את עיצוב התאריך בעזרת הפונקציה convert – אך הערך שיוחזר יהיה מחוזצת.
- iii. הפונקציה @nestLevel מציגה את רמת הקינון – בפרוצדורת רקורייביות בלבד.
- iv. הפונקציה @rowCount מציגה את מספר הרשומות שעודכנו או נשלפו בפעם الأخيرة
- v. המירה בעזרת cast עדיפה על המרה בעזרת convert

T-SQL

13. הקוד הבא יתריע על אובייקט 'EmployeesSelect' לא קיים' כאשר:

```
if object_id('EmployeesSelect')=null  
    print 'לא קיים'
```

i. אין אובייקט בשם EmployeesSelect

ii. יש אובייקט בשם EmployeesSelect

iii. אין פרוצדורה בשם EmployeesSelect

14. בהגדרת סמן:

i. חובה להגדיר בהצהרה לאחר הטיפוס cursor את משפט השליפה בעזרת for

select

ii. ניתן להריץ פרוצדורה עבור כל אחת מהרשומות.

iii. כשתהטיימה ערכות הרשומות המשטנה @@fetch_status=0

iv. ביטול הקצאת הסמן בעזרת deallocate פוטר מסגירת הסמן בעזרת close

בהצלחה רבה!

3. מבחן ב SQLחלק א'

(40%+10%) (6 מתוך 5) כתבי את האובייקטים הבאים, והוסify שורה המפעילה אותם.

- כתבי פרוצדורה המוחקת מטבלת הודעות המערכת (sys.messages) את כל ההודעות שהוכנסו ע"י המשתמשים. הפרוצדורה תחזיר כמה הודעות נמחקו.
- כתבי פרוצדורה המוסיפה לכל הלוקוחות שההכנסות שלהם בשנת 2014 היו מתחת 5000 נט רשות עם חיבור בסך של 20 נט. לעומת זאת לקוחות אשר הכנסותם היו מעל 5000 נט יצכו במבנה של 50 נט. הוסify להם רשות זיכוי מתאימה.
- כתבי פונקציה מקבלת שנה ומחזירה טבלה המציג מספר חשבון, סך הכנסות בחשבון ועומדה נוספת לסק ההוצאות בחשבון אותה שנה
- כתבי פונקציה המקבלת תאריך ומחזירה אותו בפורמט מילולי ארוך. לדוגמה: בעבור התאריך 1-1-2015 14:15 תחזיר הפונקציה: יום חמישי ה 1 לינואר 2015 שעה 15:14 בצהרים.
- כתבי ויו המציג את רשימת התנויות: מס' חשבון, תאריך, סכום, תאור ויתרה, כמו"כ יופיע שדה המציג אם הייתה חריגה מהמסגרת המוגדרת בחשבון. אם לא הייתה חריגה ייכתב "לא הייתה חריגה", אם הייתה חריגה של עד 100 ש"ן יוצג "חריגה קלה", אחרית יותג "חריגה חמורה".
- כתבי טריגר הפועל בעת הוספה או עדכון סוג בדיקה בטבלת examinations. בעת הוספה יבדוק הטריגר שערך המינימום קטן מערך המקסימום, ובעת עדכון יבחן שלא בוצע שינוי בשזרות ערך מינימום או מקסימום. אם בוצע אחד מהן"ל תבוטל פעולה הוספה/עדכון

חלק ב'

(24%)

(10 מתוך 8) בחרי בתשובה הנכונה

- היכן אפשר לשלב פקודות T-Sql ?
 - פרוצדורה ופונקציה multi-statement
 - פרוצדורה וכל צורות הפונקציה
 - פרוצדורה ווו
 - פרוצדורה, פונקציה multi-statement וטריגר
- מה איננו שימוש של טריגר?
 - לבצע בדיקת תקינות במקרה שהוא ממשמעותית מאוד
 - לתעד שינויים שנעשו בטבלה
 - למנוע ממי שאין לו הרשאה לבצע שינויים בטבלה
 - לעדכן שדות חישובים

3. מה איננו שימוש של ?cursor

- להפעיל פרוצדורה בעבור כל שורה בשאלתה
- להפעיל פונקציה סקלרית על כל שורה בשאלתה
- לבצע פעולה עדכון נפרד על כל שורה ושורה בשאלתה באופן שאם תהייה שגיאה בשורה אחת, זה לא יבטל את כל העדכון של כל השורות.
- לפשט שאלות מורכבות מאוד כאשר אין משמעות למשך זמן הביצוע

4. כיצד נקבל את מספר השגיאה האחרונה שהתרחשה?

- @@error
- Error_number()
- או () – error_number(), לא משנה מה נכתב
- Error_message()

5. מה יקרה אם לא נכתב ב case את ה ?else

- הודעת שגיאה
- 이חוור 0
- 이חוור null
- מצב זה חוקי רק לצורך התחבר הקצירה כאשר מפעילים פונ' צבירה על הערך החזר, שאז אפשר להתעלם מקרה שבו לא חזר ערך ב else. (לדו' count(case when... then... end))

6. כיצד נוסיף שגיאה מותאמת אישית ל ?sql

- נפעיל את הפרוצדורה sp_addmessage ונוסיף הודעה עם מזהה מעל 50000
- נפעיל את הפרוצדורה sp_addmessage ונוסיף הודעה עם מזהה בלבד.
- מוסיף רשומה לטבלה sys.messages ע"י פקודת insert
- מוסיף רשומה לטבלה sys.messages ע"י פקודת insert שהמזהה שלה מעל 50000

7. מה אפשר לשלב בחור משפט ?select

- פקודת if...else
- יצירת משתנה
- קריאה לפרוצדורה
- קריאה לפונקציה

8. כיצד ניתן לשלב משתנים בחלק ה select של שאלתה?

- ניתן לבצע הצבה של ערך של שדה אחד במשתנה אחד בלבד
- ניתן לבצע הצבה של ערכי מספר שדות במשתנים רבים
- ניתן לשולח חלק מהשדות לתוך משתנים וחלק לא
- ניתן להציב ערך במשתנים רק כמשמעותו בשאלתה שמחזירה רשומה אחת.

9. متى יופעל טריגר?

- כאשר נקרא לו בצורה מפורשת או באופן אוטומטי בפעולת הוספה/עדכון/מחיקה

T-SQL

- ב. באופן אוטומטי בכל פעולה של הוספה, עדכון או מחיקה של כל טבלה ב DB
ג. באופן אוטומטי בכל פעולה של הוספה, עדכון או מחיקה של טבלה אחת מסוימת שהוגדרה בטריגר.
ד. באופן אוטומטי כאשר תבצעו אחת הפעולות שהוגדרו מראש בטבלה מסוימת.

10. מה לא נכון לגבי משתנים ב ?sql

- א. כל משתנה מאותחל ביצירתו בערך 0.
ב. שמו של כל משתנה רגיל יתחל ב @
ג. ניתן להגדיר משתנה מסווג טבלה, במשתנה זה יש לפרט בסוגרים את שמות שדות הטבלה וטיפוסיהם
ד. המשתנים הינם המשתנים מקומיים הקיימים רק בתחום הסקריפט שבו הוגדרו ומשתחררים מהזיכרון עם סיום

ביצוע הקוד

חלק ג'

(26%)

כתבי נכון או לא (13 מתוך 15)

1. במצב של עדכון גם פעולה `inserted` ו `deleted` של הטריגר תהיה מלאות _____
2. הפקציה `update` מקבלת שורה ומוחירה האם היא עודכנת או לא _____
3. המשתנה `fetch_status` מכיל 0 כאשר מספר השורות שנתרו בcursor הגיע ל 0. _____
4. המשתנה `row_number` מכיל את מספר השורות שהושפטו (נשלפו או העדכנו) בפקודה الأخيرة. _____
5. המשתנה `error` שומר כל פעם את מספר השגיאה الأخيرة שהתרחשה _____
6. כדי לסיים טרנסקציה נכתב `end transaction`. _____
7. אפשר לבצע טרנסקציה רק בתחום `catch` _____
8. הפקציה `raiserror` מוחירה את נתוני השגיאה الأخيرة שהתרחשה _____
9. הפקציה `error_severity` מוחירה את רמת החומרה של השגיאה الأخيرة שהתרחשה _____
10. פרוצדורה יכולה להחזיר ערך ע"י פרמטר `output` _____
11. פרוצדורה יכולה להחזיר ערך ע"י `return` _____
12. פונקציה יכולה להחזיר ערך ע"י `return` _____
13. פונקציה יכולה להחזיר ערך ע"י פרמטר `output` _____
14. הביטוי `10 + null` יהיה שווה 10. _____
15. חובה להשתמש ב `cast` כאשר משרשים מחרוזות עם ערך מספרי או עם תאריך. _____

גאומטריה כהה!

SQL מבחן ב .4

a. ניתן לחלק את השפה למספר קטגוריות של פקודות (צנ"י בימין 2*21 : (a,b,c,d))

.1. **DQL - (data query language)**

.2. **DML - (data manipulation language)**

.3. **DDL - (data definition language)**

a. אוסף פקודות המאפשרות לבצע מניפולציות על נתונים. למשל להזין רשומה חדשה לטבלה או לעדכן רשומה קיימת. כולל את הפקודות: Delete, Insert, Update.

b. אוסף פקודות להגדלת היישות הנחוצה בסיס הנתונים והקשרים ביניהם. למשל יצירת טבלה.

כולל את הפקודות: Drop, Alter, Create, Truncate ו –

c. פקודות שטרכתן שליפת הנתונים מתוך הטבלאות. כוללת את הפקודה Select.

ב. סמני V או X בימין השורה:

1. פקודות SQL אינן Case Sensitive, כלומר, אין רגשות לגודל האות, אין הבחנה בין אותיות גדולות לקטנות.

2. ניתן לכתחוב משפט SQL עם שימוש ברוחחים והזחות.

3. אסור לרדת שורה באמצעות פקודה.

4. ניתן לקבץ לפי המיקום של עמודה בפסוקית ה Select. לדוגמה: Group By 2.

5. המילה percent היא אפשרות בעת שימושו ב TOP

SELECT max(avg(unitprice)).group function SQL Server לבצע קניון של

6. ניתן ב – SELECT ISNULL(Phone, 'Unknown Phone') from... במקומ הטל המקורי

7. 1-7 DATEPART(weekday, date) מוחירה ערך מספרי בין

8. – h Simple Case – 가능 בבדיקה ערך של משתנה בווד (שווה או !=, <, >) והציג ערך אחר במקומו.

9. CASE WHEN – ELSE – END בביטוי CASE הוא רשות. אם ממשיטים אותו, כל הערכאים שלא פורטו יקבלו NULL.

10. union – איחוד הכותרות נקבעות לפי משפט select – האחרון.

11. משמשים all union union all כדי לשלב את התוצאות של מספר משפטי select כולל רשומות כפולות.

12. בפקודת הכנסה לשדות המוגדים כת-identity השלח null

13. בהוספה של מספר שורות לטבלה תוך שימוש בשאלתה לשיליפת השורות מטבלה אחרת חובה לשמור על

14. מספר עמודות בשיליפה = מספר העמודות בטבלת היעד.

15. כאשר עליינו לעדכן כמה דברים בו זמנית, אבל לא נרצה שם יוצר בעיה באחת מן הטבלאות, אז העדכן בשאר הטבלאות יבוצע נוכל להיעזר בטרנסקציה.

16. כאשר נרצה לרוקן טבלה וגם לאפס את המיספור האוטו' משתמש ב delete

17. מחסرونויות view: כמעט איןנו ניתן לעדכן ומורייד ביצועים.

18. השמת ערך מטבלה על ידי שימוש במספט SELECT מהיבר שמשפט select – יחויר שורה אחת בלבד.

19. ב – t-sql כמו ב DQL ניתן להשתמש ב CASE אך לא ניתן ב – EXISTS

20. טיפול בשגיאות try-catch פוטר שימוש בטרנסקציה.

21. כאשר יש שימוש בטרנסקציה בתחוםblk try חלק מהפעולות מתבצעות ורק הפעולה עם השגיאה תטופל בהתאם ותכשל.

22. ניתן להציג ERROR_MESSAGE() ההודעה שהתקבלה רק בבלוק catch.
 23. RAISERROR לא סגור טרנסקציה ולא מבצע ROLLBACK.
24. הפקודה הבאה מתאימה להיות חלק מאית הפקודות בפרויקט שמכיל זריית שגיאה:
`EXEC sp_addmessage @msgnum = 50002,@severity = 1,@msgtext = 'The message'`
25. משנה גלובלי אשר מוחזר את מספר השגיאה האחרונה שהתרחשה לאו דווקא בפקודה[@@ERROR](#) לאחרונה.
26. פרוצדורות יכולות להכיל משתנים, פונקציות, תנאים, לולאות ועוד.
27. ניתן לקרוא לפרוצדורה מפרוצדורות אחרות.
 28. יעיל יותר לקרוא לפרוצדורה כך: EXEC AddCustomer '12345', 'hhh'
`Exec AddCustomer @Customerid = '12345', @CompanyName = 'hhh'`
29. פרמטר מוחזר יוגדר כמו פרמטר ה INPUT רק שלאחר ציון הסוג שלו תזיהו המילה OUTPUT.
 בחני את הבדלים בין 2 קטעי הקוד הבאים:

DECLARE @answer smallint EXEC MathMult 5,6,@answer OUTPUT Select @answer	Declare @ answer int EXEC @ answer = MathMult 5,6 SELECT @ answer
--	--

30. כתוצאה מההרצה בחוץ התוצאות יודפסו 30 בשני המקרים.
 31. הערך המוחזר מהפרוצדורות הנ"ל הוגדר בעזרת הפuktודה return בשתייה.
 32. ניתן שהפרוצדורה MathMult לגמרי זהה בשני המקרים.
 33. ניתן להחזיר גם null בעזרת הפuktודה return.
 34. EXEC Sp_help '[SalesLT].[Address]'
 האובייקט שנשלח כפרמטר.
 t-sql – יכולות להכיל t-sql Multi statement table valued functions ו Scalar functions .35
 .36 In line table valued function – אינה יכולה להכיל מין כמו שאר ביטויי הטבלה אא"כ יש טופ.
 .37 Multi statement table valued functions – מכילה משפט select בו וודד בו ניתן לשלב פרמטרים.
 .38 Scalar functions – תוכן הפונקציה- צ"ל חום בין begin ו end .
 .39 trigger rollback הוא יבטל גם את הפעולה שהפעילה את ה-trigger .
 .40 . deleted inserted – כל פעולה DML נוצרת בטריגר 2 הפעולות: .

T-SQL

גניך שקיימת הטבלה:

Chag	Cost
פסח	1000
שבועות	200
סוכות	900

שאילתת pivot מתאימה תראה:

```
SELECT * FROM ( SELECT chag , cost FROM Sales
) p PIVOT ( SUM(cost )
FOR chagim , [ שבועות ] , [ פסח ] IN ( )
) AS pvt
```

41. לא ניתן יהיה להחזיר את התוצאה לטבלת מקור בעזרת unpivot

42. בתוצאה תהיה 2 עמודות

43. בתוצאה תהיה שורה אחת

44. במאים יופיעו הערכיים: 1000,200,900

45. הפקודה הבאה חוקית: select * from dbo.Employees where @sqlWhere

46. SQL דינامي לא יוכל לעבוד כאשר יצטרך להכיל מחרוזת שמכילה גרש.

בחני את הקוד הבא:

```
declare @PivotHeader nvarchar(max)
set @PivotHeader = ''
select @PivotHeader=@PivotHeader+ ' [' + City+ '], '
from dbo.Employee_tbl
group by City
set @PivotHeader = substring(@PivotHeader,0,len(@PivotHeader))
```

declare @pivotQuery nvarchar(max)

set @pivotQuery = '

SELECT * FROM (select

```
        EmpID,
        city
        from Employee_tbl) piv
```

pivot(count(EmpID) for city in(' +@PivotHeader +')) P'

execute (@pivotQuery)

47. ניתן היה לבצע זאת גם בעזרת השאילתת פיווט רגילה, שאינה דינמית.

48. לא ניתן יהיה להחזיר את התוצאה לטבלת מקור בעזרת unpivot

49. בשילפה הבאה חמוספר כל רשומה במספר ייחודי:

```
select row_number() over(order by Id), * from dbo.dbo.Employee_tbl
```

T-SQL

ג. תרגמי ותקני %: 3*3=9%

1. The function 'Row_Number' must have an OVER clause with ORDER BY.
2. There is already an object named 'GetDesc' in the database.
3. Recursive common table expression 'temp_patients' does not contain a top-level UNION ALL operator.
4. The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified.

ד. בוחני את הפרויקצידורות של פנוייד: (2*17)

```
ALTER procedure [dbo].[P_ObjectDefinition]
(@nvProcedureName nvarchar(100)=null)
as
begin

if (@nvProcedureName)is not null
    SELECT OBJECT_DEFINITION(OBJECT_ID(@nvProcedureName))
end
end
```

הפרויקצידורה הנ"ל שולפת את תוכן הפרויקצידורה שבפרמטר מטבלאות קטלוג המערכת, הערכים בה לדו':

tbl_proceduresToBeCopied

ObjType	objectDefenition	ObjName
Procedure	Create procedure abc as select...	abc
Procedure	Create procedure XXX as select...	xxx

סמנוי ב V וב X ותקני:

1. אין חובה לשלוח לפרויקטידורה זו שם של פרויקצידורה.
2. אם ישלח שם של פרויקצידורה תשלף ההגדירה שלה ואם לא- תיזרק שגיאה.
3. אם לא ישלח פרמטר תשלפנה ההגדירות של כל הפרויקצידורות.
4. אם ישלח שם של פרויקצידורה לא קיימת - ID_OBJECT=null

טבלת tbl_proceduresToBeCopied משתמשת בפרויקטידורה קודמת ומכללה תכני פרויקצידורות של

בסיס נתונים אחר (מקור). הפרויקטידורה הבאה משתמשת בה ומשרתת העברת פרויקצידורות מבוסיס נתונים

אחד לאחר, סמנוי ב V וב X ותקני:

5. ניתן להפעיל את הפרויקטידורה בלי לשלוח פרמטרים כלל.

T-SQL

6. תיזרק שגיית מערכת אם מעティקים מאותו מקום לאותו מקום
7. תיזרק שגייה יזומה אם מעティקים מאותו מקום לאותו מקום והתוכנית חיעץ.
8. תוצג הודעה שגייה והפרוצדורה ממשין.
9. התוכנית הנ"ל יכולה להיות לא סמן ומילא הייתה עיליה בהרבה.
10. ה `try` המופיע שבתוכה הפרוצדורה גורמת שיתכן מצב שהקל מהפרוצדורות שבטבלה תיבנינה וחלק לא.
11. כל הגדרת פרוצדורה שהודפסה ע"י ה `Print` בהצלחה תציג 'succeeded'
12. כל פרוצדורה שנבנתה בהצלחה תציג 'succeeded'
13. האדרת פרוצדורה שלא נשלפה מטבלת `tbl_proceduresToBeCopied` תציג 'failed'
14. אם פרוצדורה מהטבלה כבר קיימת - הפרוצדורה תזרוק פירוט שגייה ולא תבצע העתקת אף פרוצדורה
15. אם פרוצדורה מהטבלה כבר קיימת - הפרוצדורה תזרוק פירוט של שם הפרוצדורה שלא הוועתקה
16. אם פרוצדורה מהטבלה כבר קיימת - הפרוצדורה תציג את הודעה השגיאה של המערכת
17. בסיום התוכנית בחוץן ההודעות תוצג המילה 'succeeded' כמספר הרשומות שבטבלת `tbl_proceduresToBeCopied`
18. בסיום התוכנית בחוץן ההודעות תוצגנה ההגדירות של כל הפרוצדורות שבטבלת `tbl_proceduresToBeCopied`
19. בחוץן ההודעות תוצג הגדרה וציון הצלחה/כשלון לכל רשותה מטבלת ההגדירות.

```
ALTER procedure [dbo].[P_CopyProcedures]
    @sourceServername nvarchar(50)='s2008',
    @sourceDBname nvarchar(50)='bank1'
as
begin
begin try
if @@Servername=@SourceServername and DB_NAME()=@SourceDBname
    RAISERROR(' The source DB is as same as the target DB',16,1 )
else
begin
    DECLARE CSR_PROC CURSOR FOR
    SELECT ObjectDefention,objType,objName
    FROM     .tbl_proceduresToBeCopied

    OPEN CSR_PROC
    FETCH NEXT FROM CSR_PROC INTO @routineDesc,@objType,@objName
    WHILE @@FETCH_STATUS = 0
    BEGIN
        begin try
            print @routineDesc
            exec (@routineDesc)
            print 'succeeded'
        end try
    END
end
end try
end
```

T-SQL

```
end try
begin catch
    select @@ERROR,ERROR_MESSAGE() ,ERROR_PROCEDURE ()
    print 'failed'
end catch
FETCH NEXT FROM CSR_PROC INTO @routineDesc,@objType,@objName
END

CLOSE CSR_PROC
DEALLOCATE CSR_PROC
end

end try

begin catch
SELECT
    ERROR_NUMBER() AS ErrorNumber
    ,ERROR_SEVERITY() AS ErrorSeverity
    ,ERROR_STATE() AS ErrorState
    ,ERROR_PROCEDURE() AS ErrorProcedure
    ,ERROR_LINE() AS ErrorLine
    ,ERROR_MESSAGE() AS ErrorMessage

end catch
end
```

SQL

SSMS - my work now and then

1st my 1st
mfc

תרגיל:

1. פותח את SSMS . מהו השם המלא של התוכנה?
2. בדקי אילו שירותים מופעלים ברשת. כיצד בדקת?
3. התחברי לשרת של הסמינר. מהו שמו?
4. הוסיף והתחברי לשרת של המחשב המקומי שלך. כיצד עשית זאת בקיצור?
5. צרי מסד נתונים חדש בשרת שלך. היכן גרשמו קבצים חדשים?
6. צרי מסד נתונים חדש ריק בשרת של הסמינר. היכן גרשמו קבצים חדשים?
7. צרי מסד נתונים חדש שיוצר לתחילת בקורס C של המחשב שלך.
8. באיזה שירות ניתן לעשות זאת?
9. באיזה שירות לא ניתן לעשות זאת, ומדוע?
10. צרי באחד ממשדי הנתונים שיצרת 2 טבלאות חדשות. אחת בעדר ה- designer וחתמת בעזרת סקריפט.
וזאי שהן נמצאות.
11. הוסיף 2 שורות נתונים לכל טבלה. לטבלה אחת בעזרת ה- designer , לטבלה השנייה בעזרת סקריפט שהבסיס שלו יוצר באופן אוטומטי.
12. צרי סקריפט להעתיקת מסד הנתונים למחשב אחר.
13. שני את הסקריפט כך שמסד הנתונים יוכל להווצר במיקום בירית המandal של שירות אחר.
14. הריצוי את הסקריפט על שירות אחר מהשרת שבו נכתב.
15. צרי גיבוי לאחד ממשדי הנתונים. פרסיו אותו בחזרה.

sql server

SQL Server Manager Studio (1)

Server name → <Browse for servers> → Network Servers (2)
Database Engine

sql server

connect → Database Engine → Local server (4)

Program files → Microsoft SQL Server → MSSQL13.MSSQLSERVER (5)
→ DATA

D:\SQL-DATA

(3)

(4)

(5)

(6)

(7)

(8)

(9)

← Advanced ← Next ← Next ← Generate script ← Tasks ← 'Run Job' (10)
← Next ← 3 step ← Next ← ok ← Schema and data ← Types of data to script

-> sql p.3 n.6 view script

Tools → Options → Designers → Table... → Prevent...; v-n vt
-nysm

Create View View, p.6
as

Select... from

Insert into... values (,) - my own

Drop View-

New p.6

SQL - Dictionary

exists - $\exists p$

Batch - B

Clipboard - CL

T-SQL

יחידה 1: הקדמה

ראשי פרקי:

TSQ , SQL והבדלים בינהם.

תוכנות שונות למסדי נתונים

Sql-server

Sql local-db

oracle

sql-express

PostgreSQL

My-sql

Db2

Ssms

קבץ מסד נתונים

MDF

LDF

SQL

Connect – disconnect

Connect to local

Attach-detach

פקודות סקריפט ופקודות ייזואליות

יצירת מסד נתונים חדש

יצירת סקריפט לכל מסד נתונים.

יצירת סקריפט להוספה/מחיקה/עדכון נתונים

גביי מסד נתונים, פרישה

טבלה מדומה View

ה-VIEW הוא אובייקט אשר אחסון שאילתת בסיס הנתונים. צורת שילוף הנתונים ממנו והגישה אליו כטבלה לכל דבר ועל כן
שםו: "טבלה מדומה". *טבלה מדומה היא טבלה חדשה שמשמשת ל视 הנטנוים*

בפועל הview אינו תופס מקום פיזי, כיון שרק ההגדרה שלו נשמרת בסיס הנתונים ואילו הערכים נשלפים בכל פניה ל-VIEW
 במשפט הview מופיע `from`, מכוון שבעל פניה ל view, הview מושך מחדש, כל פניה יכולה ליצור כען טבלה שונה לפי הנתון הרגעי של
 טבלאות הבסיס מהן נגזרים הנתונים.

יתרונות השימוש בVIEW:

- **אבטחת מידע**
 ניתן לשולוף או להסתיר עבור כל משתמש את המידע הרלוונטי אליו, ניתן לפקח אחר כל המידע המוצג למשתמשים וכו'.
 לדוגמה: נניח מזכיר מתחילה צריך להתקשר לכל המורים כדי להודיע הודעה כלשהי, מספרי הטלפון נמצאים בטבלה של
 נתונים העובדים המכילה גם את חשב המשכורת הממוצעת של אותו עובד. איןנו מעוניינים שהוא מזכיר יראה את כל
 המידע הזה. לפיכך יוכל ביכולתו ליצור view שיציג לו אותו מזכיר רק את שם העובד, כתובות וטלפון.
 כמו"כ טבלת ציונים ותלמידים ביכולתנו להציג עבור כל מורה רק את הציונים במקצועות שלו וכו'.
- **נוחות ובהירות**
 לעיתים קיימים קשרים מסוימים בין טבלאות, קשרים אלו רלוונטיים להרבה דוחות; לצורך כך ניתן לבנות view שיכיל
 לאחריו הקלעים את כל הלוגיקה, ואנו נוכל לפנות אל view ככל שנרצה.
- **כמו כן** לעיתים יש צורך לבצע שינויים על הטבלאות הרגילות, משתמש הניגש אל הטבלאות המדומות יכול להמשיך
 ולבנות את השאלות שלו כרגע, למרות שהטבלאות השתנו. כמו"כ הדבר מקל מאד על כתיבת שאלות מסוימות.
- **שיעור ויעילות**- הטבלה עצמה לא תופסת מקום בזיכרון כי רק ההגדרה שלה נשמרת.
- **מארגן מידע** עבור ייצוא לאפליקציות אחרות, ועוד.

כמובן שיש גם חסרונות לview, כמו לדוגמה:

- **קשה לעדכנה**- כדי שיוכלו לעדכן את ה-view צריך שיוכלו תנאים שיורחו בהמשך.
- **ביצועים**- מכיוון שבעל פניה לview השאלה שמאחורי טבלה מדומה זו רצה מחדש מוחדר, ויש לעתים שהשאלה שעומדת
 מאחוריו ה-view מורכבת מאד, חייב להיות שהדבר ישפיע על ביצועי המערכת.

מבנה VIEW

CREATE VIEW view_name

AS

שאילתת

[with check option]

[with read only]

ל גזען דע - Select
DQL - SQL select
ל גזען דע - create Alter
DML - Insert, DELETE, UPDATE

דוגמה:

יצירת view המציג את פרטי העובדים שיש להם פקס

Create view emp_fax_v

As

Select employeeID, lastName, firstName, birthdate

From employees_tbl

Where fax is not null

הצגת הנתונים מה view תתבצע על ידי הרצת השאילתה:

SELECT * FROM emp_fax_v

מכיוון שלא רשםנו את שמות העמודות שברצוננו שהview יכיל, שמות העמודות יהיו אוטומטית כשמות העמודות בטבלאות הבסיס.

דוגמא נוספת:

Create view qty_prod_v(prodName, sum_qty) as

Select p.name, sum(o.qty)

From product_tbl p inner join order_tbl o

On p.prodID= o.prodID

Group by prodName

בשאילתא זו שמות העמודות בטבלה המדומה יהיו: sum_qty -I prodName

פקודות DML על VIEW

פקודות DML על VIEW כמו עדכון, הוספה ומחיקה אפשריות אך ורק במידה ו-

- ה view מורכב מנתונים של טבלה אחת.
- ה-view לא מכיל את המילה DISTINCT.
- ה-view לא מכיל GROUP BY.
- למשתמש ניתנה הרשות גם לכך.
- ה-view מכיל את כל השדות המחייבים בהכנסת ערך (השdots שמוגדרים כ-not null) בטבלה המקורית או שהוגדר ערך ברירת מחדל עבורם.

שינוי VIEW

ALTER VIEW view_name

שאילתתה As

drop view view_name

מחיקת View:

CHECK OPTION

פעולות DML המפרות את תנאי ה – where של השאילתת שהגדירה את ה- view לא יאפשרו כאשר ב-view יהיה מוגדר . with check options

דוגמא:

יצירת view של משכורות הגבוהות מ-10000.

Create view high_salary_v

As

Select *

From salary_tbl

Where salary>10000

With check option

היות שה- view הוגדר with check option אפשר לבצע בו פקודות DML אך בתנאי שהן לא מפרות את תנאי המשכורת
כלומר:

א. לא ניתן להכניס משכורת נוספת חדש שגובהה פחות מ-10000.

ב. לא ניתן לעדכן משכורת קיימת לפחות מ-10000.

כך למשל אפשר לבצע את העדכון הזה:

```
UPDATE high_salary_v
```

```
Set empid= '301989897'
```

```
Where datehire='01/03/2011'
```

אך לא ניתן לבצע את העדכון הבא:

```
UPDATE high_salary_v
```

```
Set salary=4000
```

```
Where datehire='01/03/2011'
```

לפניך בסיס נתונים המשמש לרישום שיחות טלפון שbowtun ע"י לקוחות. לכל שיחה נרשם: שעת התחלת השיחה, שעת סיום השיחה ומחריר עלות לדקה. יתכונו הנחות בהתאם למסלול ההנחה שbowtun ע"י הלוקו. לכל לקוח מסלול הנחה אחד בלבד.

לפחות (קוד לקוח, שם לקוח, כתובת, עיר, טלפון, קוד הנחה)

Customers (custid, custname, address, city, phone, discountid)

שיחות (קוד שיחה, קוד לקוח, מטלפון, טלפון, תאריך ושעת התחלת, תאריך ושעת סיום, עלות לדקה)

Usage(usageld, custId, fromPhone, toPhone, timeStart, timeEnd , cost)

הנחות (קוד הנחה, שם מלא, אחוז הנחה)

Discount (discountId , discountName,part)

צרי בכל תרגיל טבלה מדומה לפי המבנה המתבקש, התבוסס על הטבלאות של רישום שיחות טלפון:

א) צרי view בשם AllCustomers V הציגי שם לקוח, כתובת + עיר, טלפון ושם ההנחה שבחר הציגי גם לקוחות לא הנחות.

ב) צרי view בשם NightUsages V הציגי את פרטי השיחה שבוצעו בלילה (בין 00:00-07:00): קוד לקוח, שם לקוח, מטלפון, טלפון, תאריך, זמן התחלת, משך שיחה.

ג) כתבי Cost_SumCost V שתציג לכל לקוח את משך הזמן של כל השיחות שלו בכל חודש ואת מחירן לפני ההנחה ולאחריה. שמי לב מחיר שיחה מחושב ע"פ הדקות כפול 0.72 אג'. אל תשכח לחייב גם לקוחות לא הנחות.

ד) MaxNumUsage V הציגי שמות 5 לקוחות שביצעו את מספר השיחות הגבוה ביותר.

ה) MaxSumLengthUsage V הציגי שמות 5 לקוחות שדיברו במשך זמן הארוך ביותר.

ו) MaxLengthUsage V הציגי שמות 5 הלקוחות שדיברו את השיחות הארוכות ביותר.

drop view viewer - view pair

Select * from view_pe - view in not run

batch - 121 per b 13

and 13, $\sin \nu = g_0$ remains the

select @? be nmp

set @ } .nn381

Work Select is available (in Spanish)

...nigen per cogitabilis

AIMP /MM if-> nsgs le nhsd B.

end → next begin

新嘉坡 1960年 1月 1日

778 1-800-381-1130

7.18 ~~12381~~ - stored procedures

Playfair cipher is a type of transposition cipher.

- LEP $\tilde{q} \bar{q}$ \rightarrow $l^+ l^- \nu \bar{\nu}$ \rightarrow $e^+ e^- \nu \bar{\nu}$

mult freq \rightarrow more fit

the original results list - now,

create procedure FollowStudents

$$I_{MN} / f_{\text{exp}} = \alpha s$$

declare @app1 as nvarchar(50)

wholly across me

begin print 'Hello' & convert(varchar(2),@a)

set $\omega_a = \omega_{a-1}$

235

பிள்ளை கீழ் பாடுகளை விடுவது

exec privilege on every user's right side.

execute instruction -> $\boxed{\text{init memory register stack frame}}$ - $\boxed{\text{load memory}}$ \rightarrow $\boxed{\text{execute}}$

create procedure FollowStudent (@x int)

A1 F50

MR - DROP

high level programming

programmability

Stored Procedures

re-use modify your code

Generic User ID - Guid always same

user-defined UDF user define function

reusable code

create function

(parameters)

returns table

as

return

(result)

Syntax - tip where to write the code

function name

T-SQL

Case Expression ב.

ביטוי Case הוא ביטוי שמעיריך תנאי אחד או יותר, ומהזיר ערך בווד מתח מספר אפשרויות בהתאם ל贤אזה של הערצת הביטוי.

שימוש ב Case הוקי בכל מקום שבו-Amor להופיע ביטוי שמחזיר ערך, כשהה במשפט select, חלק מתנאי המשפט Group by, Order by, Where.

בביטוי Case השימוש ב Else הוא אופציונלי, ובמקרה שלא צוין else יוחזר null. קיימות שתי צורות תחביר לביטוי Case

❖ פשות – הביטוי משווה שדה / ביטוי מול ערך/ערכיהם אפשריים.

לדוגמה:

```
SELECT ProductId, Name,  
CASE Price  
    WHEN 0 THEN 'not for resale'  
    ELSE 'for resale'  
END AS PriceRange  
FROM Product
```

❖ מרכיב – הביטוי מעיריך תוצאה של תנאי או מספר תנאים Case

לדוגמה:

```
SELECT ProductId, Name,  
CASE  
    WHEN Price = 0 THEN 'not for resale'  
    WHEN Price < 50 THEN 'Under $50'  
    WHEN Price >= 50 and Price < 250 THEN 'Under $250'  
    WHEN Price >= 250 and Price < 1000 THEN 'Under $1000'  
    ELSE 'Over $1000'  
END AS PriceRange  
FROM Product
```

❖ בחוק שאילתת Case

```
Select cityId,  
CASE columnName
```

When value1 then result1

When value2 then result2

When value3 then result3

End

As columnNewName

From tableName

לדוגמה

```
select [Order ID],
```

```
caseIsPaid
```

T-SQL

```
Begin  
set @x = @x + 1  
Print @x  
End
```

ב. בצעי את המשימות הבאות ב [sq]-t. רשמי לצד כל שאלתה את הפלט שלו.

- .1 הגדרוי שני משתנים מסוג Int אתחלי את שניהם והדפיסי את סכומם.
- .2 אתחלי רק אחד מהמשתנים והדפיסי את סכומם.
- .3 הגדרוי משתנה מסוג מחרוזת, אתחלי אותו והדפיסי אותו באותיות גדולות.
- .4 הגדרוי משתנה נוסף מסוג מחרוזת שאינו מאותחל ובדקי אם הוא שווה לקודם, אם כן הדפיסי "שווים". בדק ב If נפרד אם אינם שווים, הדפיסי "שונים".
- .5 הגדרוי משתנה נוסף מסוג Date, והוספי לו בלולאת while יומ אחד בכל שלב עד שהירה שווה לתאריך של היום.
- .6 כתבי פרוצדורה השולפת את הסטטוס השכיה ביוור בטבלת המחשבים.
- .7 אם הסטטוס הינו "תיקין" הפרוצדורה תדפיס "מצוין", אם הסטטוס הינו "דרוש תיקון" הפרוצדורה תדפיס "מצב דרוש שיפור", אם הסטטוס הינו "יצא מכלל שימוש", הפרוצדורה תדפיס "יש לבצע בדיק בית יסודי".
השתמשי ב .case
- .8 כתבי פרוצדורה מקבלת פרמטר מסוג תאריך, ופרמטר Out מסוג int. הפרוצדורה בודקת אם קיימות תוכנות שהתווקנו לאחר תאריך זה, אם כן הפרוצדורה תחזיר את שמותיהן. (שים לב לא להחזיר את אותה תוכנה פעמיים), לאחרת הפרוצדורה תחזיר את כל התוכנות. עדכני בפרמטר Out ערך המטען האם הרשימה שהזורה מכילה את כל התוכנות או תוכנות לפי התאריך שטופק.

```
--גוזן 1 מסוכן inline
alter function kavim3(@kodn int)
returns table
return
select max([kav]) 'max kav' from [dbo].[bus] where @kodn=[driverid]
go
select * from dbo.kavim3(205)

go
--גוזן 3 מסוכן multi
create function kavim4(@kodb int)
returns @t table (kav int,tmanim time)
begin
declare @a time
declare @b time
declare @f int
select @a=[time_start],@b=[time_end],@f=[frequncy] from [dbo].[bus] where @kodb=
[kav]
select @a=[time_start],@b=[time_end],@f=[frequncy] from [dbo].[bus] where @kodb=
[kav]
while @a<=@b
begin
insert @t values(@kodb,@a)
set @a=dateadd(minute,@f,@a)
end
return
end
go
select * from dbo.kavim4(2)
```

מסד נתונים אוטובוסים – פונקציות

מסד הנתונים אוטובוסים מכיל את הטבלאות הבאות:

station_bus	station	drivers	bus
kav – מספר הקו	stationid – קוד תחנה	driverid – קוד נהג	– מספר הקו
stationid – תחנה	station_n – שם תחנה	n_driver – שם נהג	– קוד הנהג
		tel – טלפון	– שעת התחלת
			– שעת סיום
			– תדירות
			– תחנת מוצא
			– תחנת סיום

- כתבי פונקציה המתקבלת מספר תחנה ומחזירה שם התחנה.
 קראי לפונקציה מתוך משפט select מטבלת אוטובוסים :
 ה/cgi קו, נהג, תחנת יציאה ויעד.
 תחנות היציאה והיעד שמורות בטבלה כמספרים, ע"י קרייה לפונקציה בתוך המשפט select ניתן לראות את שם תחנת היציאה ושם תחנת יעד.
- כתבי פונקציה המתקבלת מספר תחנה וקו אוטובוס, ומחזירה משתנה בוליאני bit 0/1 כאשר הקו לא עוצר בתחנה זו הפונקציה מחזירה 0, אחרת הפונקציה מחזירה 1. העזרי ב- if exists .
 קראי לפונקציה ובדק את התוצאות. עבור קו ומספר תחנה כלשהו.
 קרייה לפונקציה בתוך משפט select עבור קו כלשהו וכל התחנות מטבלת תחנות.
 קרייה לפונקציה בתוך משפט select עבור תחנה כלשהו וכל הקווים מטבלת אוטובוסים.
 קרייה לפונקציה מתוך משפט select עבור כל הקווים וכל התחנות.
- צרי פרוצדורה שמקבלת מספר תחנה ומציג את הקווים שעוצרים בתחנה זו בתבנית הבא "קו 2 עוצר בתחנת עוזרא" (השתמשי בפונקציה שבתרגיל הקודם).
- כתבי פונקציה המתקבלת שעה ומחזירה ערך : תקין/לא תקין. שעה תקינה היא : בין 00:59:23 ל-23:00 .
 כתבי שאלתה על טבלת אוטובוסים ומציג את רשימת הקווים שהשעה שלהם אינה תקינה.
- כתבי פונקציה המתקבל קו ומספר תחנה, ומחזירה את השעה שבה הגיע הקו לתחנה זו. ע"פ החישוב הבא :
 יש לבצע לולאת while שמצוות את שעת היציאה הקרובה ביותר לשעה הנוכחית. הלולאה רצה כל עוד שעת היציאה קטנה יותר than $\text{while } @datestart < \text{convert}(\text{time}, \text{getdate}(), 108)$
 בתוך הלולאה ישקדם את שעת היציאה לשעת היבא ע"פ התדריות (העזרי בפונקציה dateadd())
- כתבי פונקציה המתקבל שני קווים ומספר תחנה ומחזירה את קו האוטובוס שיגיע יותר מהר. העזרי בקרייה לפונקציה מסעיף קודם (תרגיל 4).
- כתבי פונקציה המתקבלת תחנה ומחזירה תשובה האם זו תחנת יציאה או תחנת סיום או תחנת ביניהם.
- כתבי פונקציה המתקבל קו אוטובוס, ומחזירה את רשימת התחנות משורשרות עם פסיק ביןיהם.

מסד הנותנים לעובדים

תרגילי פונקציות סקלריות המחזירות ערך בודד

1. כתבי פונקציה המקבלת קוד עובד ומחזירה את סך כל משכורותיו.
2. כתבי פונקציה המקבלת חודש שנה, עובד ועובד ב, ומחזירה את ההפרש במשכורת בין עובד א לעובד ב בחודש ובשנה שנשלחו.
3. כתבי פונקציה המקבלת לקוח ומחזירה טקסט. הפונקציה תספר את כמות הזמןות ללקוח אם הכמות גדולה מ-10 ותחזיר 'לקוח קבוע' אחרת 'לקוח מזדמן'.
4. כתבי פונקציה המקבלת קוד עובד חדש ושנה ומחזירה ערך בוליани האם בחודש ושנה זו קיימת לעובד זה במשכורת.

```
--Lesson#6
--תנאי תקינות
CREATE FUNCTION FuncLenDesc ( @start time ,@end time)
RETURNS nvarchar(10)
AS
BEGIN
    -- Declare the return variable here
    DECLARE @Result nvarchar(10)

    declare @len int=datediff(minute,@start,@end)
    if(@len<10)
        set @Result='short'
    else
        if(@len>100)
            set @Result='ארוך'
        else
            set @result='normal'

    RETURN @Result
END
GO

print dbo.FuncLenDesc('8:00','9:00')

select *,dbo.FuncLenDesc(timeStart,timeEnd) 'תאזריך משך השימוש' from usage
צדי פונקציה סקלרית לפירמות נארוך
create function func_FormatDate(@date date,@char nvarchar)
returns nvarchar(10)
as
begin
    return cast(day(@date) as nvarchar) + @char
    + cast(month(@date) as nvarchar) + @char
    + cast(year(@date) as nvarchar)
end
select *,dbo.Func_FormatDate(dateusage,'#') from Usage

alter FUNCTION Func_tableInLine
(
    @CustomerID int
)
RETURNS TABLE
AS
RETURN
(
    select *,dbo.funcLenDesc(timestart,timeend)'Len' from usage where
        custId=@CustomerID
)
GO--הפעלה
select * from Func_tableInLine(1)


השימוש במשתנה מסווג טבלה--
יש להגדיר שמות למזהות וטיפוסים--
declare @t table( id int,name nvarchar(20))
insert into @t--הכנסת מרובה למשתנה הטבלה
select custId,Custname from Customer
--שליפה מהמשתנה טבלה
select * from @t t join Customer c on t.id=c.custid
```

```
פונקציה שמחזירה טבלה בריבוי פקודות--  
CREATE FUNCTION Func_Table_MultiStatement--  
(  
    @custId int  
)  
RETURNS  
@userManMin TABLE  
(  
    fromPhone varchar(50),  
    toPhone varchar(50),  
    minOrMax varchar(4)  
)  
AS  
BEGIN  
    insert into @userManMin  
        select top 1 fromphone,toPhone, 'min' from usage where custid=@custId order by ?  
            datediff(minute,timestart,timeend)  
    insert into @userManMin  
        select top 1 fromphone,toPhone, 'max' from usage where custid=@custId order by ?  
            datediff(minute,timestart,timeend)desc  
  
    RETURN  
END  
GO  
select * from Func_Table_MultiStatement(1)
```

פונקציות המחזירות טבלה - Multi-statement או Inline

בסיס הנתונים אוטובוסים - bus

1. כתבי פונקציה שתקבל קוד נהג ותחזיר את רשימת הכווים שהוא נוהג בהם (בצדישתי הצורות).
2. כתבי פונקציה שתקבל קו אוטובוס ותחזיר את כל התחנות שהוא עובר בהם, ממויינים על פי שם התחנה.
3. כתבי פונקציה שתקבל קו אוטובוס ותחזיר טבלה של כל זמני הנסיעה שלו.
4. כתבי פונקציה שתקבל קו אוטובוס ותחזיר רשימת תחנות שלא שייכות לקו זה.
5. כתבי פונקציה שתקבל שעה ותחזיר רשימת אוטובוסים שפועלים אחרי השעה זו ואת שמות התחנות משורשות (ע"י קריאה לפונקציה סקלרית מס' 8).
6. כתבי פונקציה המתקבלת מספר (n) ומחזיר רשימת נהגים (קוד ושם) שכמויות האוטובוסים שלהם גדולה מהמספר (n).
7. הוספי שדה salary לטבלה נהגים. הכנייש משכורת בסיסית לכל הנהגים בסך 6500 ש"ח. עדכני לכל הנהגים המוחזרים ע"י הפונקציה שבתרגיל הקודם תוספת של 5% למשכורת.

תרגיל סיום פונקציות ופrozידורות

1. הוספי טבלה חדשה שרות לקוחות. ובה השדות: מס תלונה, מס קו, תאריך, תז לקוח, נושא.
2. הוספי לטבלה נהגים את השדה nikud.
3. צרי פרוזידורה שמרתה להקל על הוספה ועדכון נתונים בטבלה זו.
נתן לשומר בטבלה לכל קו אוטובוס עד 10 תלונות. כאשר מתקבלת תלונה מ' 11 לקו, צוברים ללקוח נקופה אחת לחובתו, ומוחקם את התלונה החדשה ביחס.
לפניך פרוטו שלבים לייצור הproxידורה.

- הפרוצידורה מקבלת קו אוטובוס, תאריך, וטז ללקוח&בודקת האם קיימת תלונה זהה לפרטים אלו.
- ✓ אם כן אין לבצע כלום.
 - ✓ אם לא

יש לבדוק האםקיימים לקו זה יותר מ-10 תלונות. (העזרי בcount select וקריטריון על הקו החיבי התוצאה לשנתנה)

כאשר כמות התלונות קטנה מ-10 יש להוסיף את הנתונים שנשלחו לproxידורה כשורה חדשה בטבלה תלונות.

כאשר כמות התלונות גדולה מ-10 יש למחוק את התלונה החדשה ביותר לקו זה,
עדכן בטבלה נהגים בשדה nikud לתוספת 1 נקופה.
ולהוסיף את התלונה החדשה לטבלה שרות לקוחות.

4. כתבי פונקציה המתקבלת מספר (n) ומציג אותה (n) התלונות האחרונות, העזרי ב-(n) select top .

5. הציגי את נושא התלונה הפופולרי ביותר בטבלה שנוצרה ע"י הפונקציה בתרגיל קודם. (משמעותה כי הרבה פעמים ב(n) של התלונות האחרונות).

景德 הנתונים עובדים - emp

1. כתבי פונקציה המקבלת קוד עובד ונתן משכורת ומחזירה טבלה עם כל המשכורות של עובד זה בשנה שנסלחה.
2. כתבי פונקציה שמקבלת שני תאריכים ומחזירה טבלה של שמות הלקוחות שמוט המוציאים והכמות שהוזמנו בין 2 התאריכים ממוינוים לפי התאריך מהחישב לישן.
3. כתבי פונקציה המקבלת קוד עובד. הפונקציה משתמשת בולאה העוברת על כל החודשים בשנת 2012, ובודקת האם העובד קיבל משכורת בחודש זה (העוזר בקריאה לפונקציה סקלרית מס' 4), הפונקציה תחזיר טבלה של שם החודש והמשכורת.

declare

cursor cursor cursor
לכון נסמן כוונת
לכון נסמן כוונת

declare cursor re cursor for select ...
open cursor re

fetch next from cursor into int num re num

while begin @fetchstatus =
if no exit ~

end
close
deallocate

1. צרי טבלת נתונים_שגויים, הטבלה תכיל שני עמודות קו אוטובוס וסוג שגיאה.
כתבו סמן העובר על טבלת אוטובוסים
הסמן בודק לכל קו האם קיימות לו תחנות אם לא מוסיפה לטבלת השגויים את מספר הקו ובעמודה
שגיאה ירשם: לא קיימות תחנות, אם השדה נהג ריק יש לרשום שגיאה: אוטובוס ללא נהג. אם
האוטובוס מתחילה ומטיים באותה תחנה יש לרשום שגיאה: תחנת יציאה וסיום זהות.

2. צרי סמן העובר על טבלת אוטובוסים, ובודק את סוג הקו, אם הקו עירוני מדפסה את רשימת התחנות
משורשיות (קריאה לפונקציה סקלרית 7), אם הקו ביןעירוני מדפסה שם תחנת מוצא + שם תחנת יעד.

סמנים - Cursor

מד הנתונים אוטובוסים - bus

מד הנתונים עובדים - emp

1. מועדון לקוחות: הוסיף בטבלת לקוחות 2 עמודות חדשות,
מועדון לקוחות מסוג bit, מספר נקודות מסוג מסטר
צרי סמן העובר על טבלת לקוחות, ובודק עבור כל לקוח כמה פעמים בצע הזמנה, אם בצע יותר מ-2
הزمנות, יש לעדכן בטבלת לקוחות את הליקות הנ"ל כדי למועדון לקוחות.
ערך העמודה מועדון לקוחות = 1,
והשדה מסטר נקודות יעדכן ב-1 נקודה עבור כל 100 ש ממחיר הקניה.
יש לחשב את סך כל מחיר הקניות של הליקות, כדי לעדכן את מסטר הנקודות.

2. בדיקת סיסמא: הוסיף בטבלת עובדים 3 עמודות חדשות,
סיסמא מסוג (6) varchar, ותאריך עדכון אחריך, נסיכון פריצת סיסמא מסוג בוליאני
צרי סמן העובר על בטבלת עובדים ובודק את הסיסמאות של כל עובד.
אם הסיסמא ריקה יש לעדכן לערך 00000,
אם הסיסמא מלאה ו עבר חודש מתאריך עדכון, יש להוסיף את שם העובד ומספר הטלפון לטבלה זמנית
שתכיל את שמות העובדים שפג תוקף הסיסמא שלהם.
אם נסיכון פריצת סיסמא = true יש להדפיס על המסך הודעה "עדכנו קו X בדחיפות".

3. בונוס עובדים:
יש ליצור בטבלת בונוסים לעובדים ובה השדות: עובד, תאריך, בונוס
צרי סמן העובר על בטבלת הזמנות של השנה האחורונה,
עבור כל הזמנה של קו יש להוסיף שורה בטבלת: בונוס לעובד שהביא אותו, בטבלת בונוס הוסיף:
קוד עובד שהביא את הליקות, תאריך הזמנה, ובונוס 1 ש לכל 200 ש קניה בפעם הראשונה, כאשר
מתבצעות קניות נוספות לאותו קו מקבלים בונוס של 5 ש"ח בלבד ללא תלות במחיר הקניה.

שימוש בביוטי Case

מסד הנתונים אוטובוסים -案

1. הosiPI טבלה חדשה: שרות לקוחות (מס תלונה, קו, תאריך, מס תחנה, נושא, סטטוס,הערה)
2. הציג קו ומספר תלונות בדק ע"י case כאשר מספר התלונות=0 רשמי 'מעולה', כאשר כמות התלונות=1 רשמי 'מעט' אחרית רשמי 'רבה'.
3. הosiPI לittel אוטובוסים 2 עמודות: סוג אוטובוס (עירוני, ביןעירוני), ומחר נסעה. עדכני את מחר הנסעה ע"י case אוטובוס עירוני 5.9 ש, ביןעירוני 10.5 ש.
4. הציג קו אוטובוס, כמות תלונות בשנת 2017, וכמות תלונות בשנים קודמות (הערך בcase בתוך count).
5. הציג שם נהג ומספר טלפון, הציג ע"י case את AUTHORISATION המגורים של הנהג לפי התווים הראשונים של מס' הטלפון שלו.

מסד הנתונים עובדים -emp

1. עדכני את הבונוס של העובדים בטבלת משכורות ע"פ גובה המשכורת, משכורת מתחת 5000 הבונוס הוא 100 משכורת עד 10000 הבונוס 50 אחרית בונוס 0
2. הציג שם עובד ועיר מגורו. בדק ע"י case האם העיר = בני ברק. אם כן רשמי 'מקומי', אחרת רשמי 'חיצוני'
3. הציג שם מוצר ומחר. בדק ע"י case כאשר המחיר קטן מ-10 הוא 2%, כאשר המחיר בין 10 ל-20 הוא 1%, אחרת יש להציג את המחיר כרגע.
4. הציג שם לקוחות וכמות הקניות שלו. בדק ע"י case האם הכמות גבוהה מ-3, אם כן כתבי 'כמות גבוהה', אחרת 'כמות נמוכה'.
5. הציג לכל עובד סכום משכורת ב-2006 וכךם משכורת ב-2007 בצעי ע"י שימוש ב-case בתוך סוגרים של פונקציית צבירה.

sum (case when year(datelastraise)=2006 then salary end)

6. הגדרי משתנה מסוג תאריך, הציג לתוךו את תאריך הלידה שלו. הגדרי משתנה נוסף והציבי לתוךו את טווח גילך במילאים ע"י case: בין '20-0' או בין '50-20' או בין '120-50'.
7. הציג שם עובד, וכמות המשכורות שלו, כאשר הכמות היא 1 רשמי עובד מתחילה אחרת רשמי עובד ותיק.

8. סכמי לכל לקוחות סך הקניות ב-2010 וכן הקניות ב-2011.

מזהיקים - Trigger

מסד הנתונים אוטובוסים - bus

1. הוסיף בטבלת תחנות עמודה חדשה מסוג varchar(20) בשם oldstation. צרי טריגר על טבלת תחנות בעת עדכון: בכל שינוי של שם התחנה המקורי את השם הישן לעמודה oldstation. ע"י המשפט:

```
update station set old station = deleted.station_n from station join deleted on
station.stationid=deleted.stationid
```

2. צרי טבלה בשם tpeulot בת 2 עמודות: מס' קו ועמודה מחוזית שבה ישמור סוג הפעולה שהתבצעה בזקן.

צרי טריגר על טבלת אוטובוסים בעת עדכון או הוספה.

הגדיר 2 משתני עזר: קו אוטובוס, ומשתנה מסוג varchar(50) לשימרת סוג הפעולה. בדק אם עודכנו שורה קיימת.

if exists (select * from inserted) and exists (select * from deleted)
הציב ערך למשנה קו ולמשנה המחזוזתי: עודכנו שורה חדשה בתאריך ... (נוכחי).
בדקי ע"י הפונקציה () update האם בוצע עדכון בשעת יציאה או בשעת סיום, אם כן, הוסיף
למשנה המחזוזתי, שגם בוצע עדכון בשעות.

אם לא, בדק אם נוספה שורה חדשה לטבלה:
if exists (select * from inserted)
הציב ערך למשנה המחזוזתי: נוספה שורה חדשה בתאריך ... (נוכחי).

הוסיף שורה לטבלה tpeulot ושמרי בה את מס' הקו ואת הפעולה,

3. צרי טריגר על טבלת אוטובוסים בעת עדכון והוספה.

טריגר לא מאפשר להכניס לטבלה ערך זהה לעמודת תחנת יציאה ותחנת יעד.
כאשר מכניסים ערך זהה הטריגר יבצע עדכון אוטומטי לערך NULL לעמודה שאותה עדכנו.

4. הוסיף בטבלת אוטובוסים עמודה חדשה מסוג תאריך.
צרי טריגר על טבלת אוטובוסים, בעת עדכון והוספה: יש לעדכן את השדה תאריך לנוכחי,
מעבר שורה נוספת או עודכנה.
ע"י המשפט:

```
update bus set date=getdate() where kav in(select kav from inserted)
```

5. צרי טבלת ארכון נהגים.

צרי טריגר על טבלת נהגים בעת מחיקה:

א. יש לעדכן בטבלת אוטובוסים את השדה נהג ל-NULL.

ב. יש להוסיף את הנהג שנמחק לטבלת ארכון נהגים.

תרגילים טיפול בשגיאות – Try Catch

מד הנתונים אוטובוסים - bus

- צרי פרוצדורה המבצעת הוספת שורה חדשה לטבלת תחנות לאוטובוסים. הפרוצדורה מקבלת 3 פרמטרים: קוו אוטובוס, מספר תחנה ומקום במסלול. לפני ההוספה יש לבדוק האם האוטובוס קיים, אם לא יש לחתה הודעה שגיאה.

2. הוסיף את שלושת הטבלאות שללן:

Traveler_in_travel	travels	travelers
Id – קוד נוסע	Travelid – קוד נסעה	Id – קוד נוסע
Travelid	Bus – מס' אוטובוס	Balance – יתרה בכרטיס
	Price – מחיר נסעה	

- צרי פרוצדורה המקבלת קוד נוסע וקוד נסעה. הפרוצדורה בודקת האם יתרה בכרטיס גבוהה מהמחיר לנסעה זו. אם כן – מוסיפה רשותה. ומעדכנת יתרה נסעה בכרטיס, לאחר הפחתת מחיר הנסעה. אם יתרה נמוכה_Zורקת שגיאה.
- שכללי את הפרוצדורה שבמקרה שמנסים להכניס נתונים כפולים למפתח (קוד נוסע וקוד נסעה) השגיאה תטופל ב-catch.
- שכללי את הפרוצדורה, שתטפל במקרה שקוד נוסע או קוד נסעה לא קיימים בטבלה הראשית, ואז א"א להכניס נתונים לטבלה המשנית.
- צרי פרוצדורה להטעת הרוב קו. הפרוצדורה תקבל קוד נוסע וסכום להטענה, הפרוצדורה תבודוק האם הסכום הוא 30 או כפולות של 50. אם כן, עדכני את יתרה, אם לא זורק שגיאה.

מד הנתונים עובדים - emp

- צרי טריגר על טבלת משכורות עובדים, הטריגר יבדוק אם בוצע עדכון לשכר העובד, אם עודכן השכר הנמוך מהשכר הקודם יש לחתה הודעה שגיאה.
- צרי פרוצדורה המוסיפה שורה לטבלת הזמן. הפרוצדורה מקבלת מס' מוצר, מס' לקוחות וכמות. לפני ההוספה יש לבדוק שהכמות המוזמנת אינה גדולה מהכמות המלאי. אם הכמות קטנה או שווה לכמות המלאי תתווסף הרשמה ויעודכן המלאי, ואם הכמות גדולה מהמלאי תזרק השגיאה "חסרה כמות של ? יחידות מלאי".

הפונקציה ROW_NUMBER

את הפונקציה ניתן לשלב בשאלתה כך שתציג בשאלתה עמודת מספר, כל שורה תקבל מספר לפי הקבוצות והמיון שבתוך הסוגרים של הפונקציה.

row_number() over ([partition_by_column] [order_by_column]) תחביר:

- ❖ לדוג' נוכל להציג את ההזמנות עם מספור עבור כל לקוח ממויין לפי תאריך ההזמנה.

```
select row_number() over(partition by [CUSTID] order by [ORDDATE])
as rounum, * from [dbo].[ORDERS_TBL]
```

בדוג' הנ"ל השתמשנו גם בקבוצות (partition_by) וגם במאיון (order_by), אך אפשר להשתמש גם רק באחד מהם.

- ❖ כדי להציג לדוג' את הקינה הראשונה של כל אחד מה לקוחות, נצטרך להשתמש ב-`rownum=1`:
אך לשם כך יש להשתמש בתת שאלה כנ"ל:

```
select * from (select row_number() over(partition by [CUSTID] order
by [ORDDATE]) as rounum, * from [dbo].[ORDERS_TBL]) as rownums where
rounum=1
```

תרגילים מסד הנתונים אוטובוסים – bus

1. הציגי מספור לטבלת אוטובוסים בקבוצות לפי נהג כך שהמספר יתחילה מחדש בכל נהג.
2. הציגי את האוטובוס הראשון (שתחיל בשעה הci מוקדמת) של כל נהג.
3. הוסיף בטבלת תחנות לאוטובוסים עמודה חדשה: מרחק בק"מ מתחנת המוצא. הציגי לכל קו אוטובוס את רשימת תחנות שלו ממספרות ע"פ המרחק מתחנת המוצא.
4. מופיעי את השורות בטבלת אוטובוסים, בקבוצות לפי תחנת מוצא ומניין ע"פ התדירות, הציגי לכל תחנת מוצא את 3 הקווים התדריים ביותר.
5. הציגי לכל נהג את פרטי הקו האוטובוס שאיתו הוא מס'ם.
6. הציגי לכל פועע את פרטי הנסיעה הראשונה והאחרת לפי-האחרונה שלו. (העזרי ב-חסוען).
7. הוסיף בטבלת נסיעות עמודה חדשה: תאריך נסיעה. (מלאי בתאריכים מחודשים שונים מ-1 עד 3 לכל חודש).
הציגי מספור לנסיעות, בכל חודש יתחיל המספר מחדש. ממויין לפי התאריך.

טבלת הצלבות PIVOT / UNPIVOT

האופרטור PIVOT מאפשר לך לסובב על ציר נתוניים, ממצב של קבוצות מרובות שורות לממצב של רבים טורים בשורה יחידה עבור כל קבוצה, תוך כדי חישובי צבירה כחלק מהתהליך.

טבלה א'	
<u>Month</u>	<u>qty</u>
Month1	100
Month2	2000
Month1	600
Month3	500
Month4	6000
Month5	1500
Month4	500

טבלה ב'				
Month1	Month2	Month3	Month4	Month5
700	2000	500	6500	1500

כדי לדמות PIVOT יש ללקחת את טבלה א' ולהביא אותה לממצב של טבלה ב' ככלומר להפוך את תוכן העמודות שבשדה הראשון כתורים, כאשר שמות העמודות בתוצאת select יהיה תוכן שבעמודה הראשונה.

מבנה בסיסי:

Select * from(select(select * from

([שם העמודה שתופיע בתוך הטבלה] pivot([פונקציית צבירה]))

for [שם העמודה שתופיע ככותרת] לעמודות q

in as ([כותרת עמודה שלישית],[כותרת עמודה שנייה],[כותרת עמודה ראשונה]) pvt

לדוגמא:

Select * from(select s.station_n, b.kav, orbit from dbo.bus b join dbo.station_bus sb on
b.kav=sb.kav join dbo.station s on s.stationid=sb.stationid) as q

Pivot

(max(station_n) for q.orbit in ([1],[2],[3]) as pvt

תרגילים PIVOT מסד הנתונים עובדים – emp

1. הציגי שם הלקוח, שנה וכמות ההזמנות בשנה זו.
2. הציגי שם עובד, חודש וסכום משכורת בחודש זה.
3. הציגי עיר, תפקיד וכמות העובדים.
4. הציגי שנה, חודש וסכום משכורות כולל.
5. הציגי מוצר, שנה, וסכום הקניות ממוצר זה.
6. הציגי מוצר, לפקח, ומאריך קניה האחרון.

UNPIVOT

דוג' ל- UNPIVOT

```
Select * from (select custid, min(orddate) as mini, max(orddate) as maxi from dbo.order  
Group by custid) as q  
Unpivot (value for property in (mini, maxi)) as pvt
```

תרגילים PIVOT מסד הנתונים עובדים – emp

1. הציגי כל לקוחות ב-2 שורות אחת עבור שם המוצר הראשון שכנה והשנייה עבור שם המוצר האחרון שכנה.
2. הציגי לכל עובד 3 שורות: אחת עבור שם המנכלה שלו, השנייה עבור תאריך התחלת עבודתה והשלישי עבור ממוצע המשכורות שלו. שימי לב: עבור שם המנכלה ותאריך התחלת השימוש בפוקצית הסכומים MAX.

DYNAMIC SQL

תרגילים מסד הנתונים אוטובוסים – bus

1. צרי פרוצדורה המקבלת שם של טבלה ושם של שדה מהטבלה ומציגה את תוכן השדה ממויין בסדר עולה.
2. צרי פרוצדורה המקבלת מחרוזת תנאי שלפיו יוצג תוכן הטבלה bus.
3. צרי פרוצדורה שמקבלת מחרוזת של סדר המיון (ערך ברירת מחדל=מחרוזת ריקה) ומציגה את שמות התחנות ממויינות בהתאם לאפשרות המיון שהתקבלה.
4. צרי פרוצדורה המקבלת 3 משתני מחרוזת:
 - א. אחד מהאופרטורים <,>,=,<>
 - ב. שם עמודה שמודדרת כמחרוזת.
 - ג. ערך מחרוזתי כלשהו.הפרוצדורה תשלוף מטבלה אוטובוסים את כל העמודות לפי תנאי על העמודה שנשלחה כפרמטר המתאים לערך שנשלח כפרמטר.
5. הפci את הלקוחות בשאלתה שיצרנו כתובן ללקוחות ע"י משתנה דינמי.

תרגילים לשעורי בית

צירוף חיצוני, צירוף הUBLE אל עצמה, תחת שאילתת מתואמת:

1. הציני את פרטיו כל המוצרים ואת הנסיבות שהזמנת פאלו שהזמננו.
2. הציני את כל הלקוחות מבני ברק ואת תאריך האירוע הציני גם את הלקוחות מבני ברק שלא הזמן.
3. הציני את שמות המאכלים שלא הזמננו.
4. הציני זוגות מוצרים שמחירים שווה.
5. הציני את שמות המוצרים וסה"כ הכמות שהזמננו רק לאלו שהה"כ הכמות שהזמנת מהם גדולה או שווה למוצע הכמות שהזמננה בכל ההזדמנות.
6. הציני את שם המוצר הזול ביותר.
7. הציני את פרטי המוצרים שהזמננו בשנת 2018 וסוגם הוא כמו הסוג של מוצר מס' 10.
8. הציני פרטי הלקוחות שהזמינו את המוצר היקר ביותר.
9. הציני את ההזדמנות שבוצעו באחד החודשים שהזמין לךו שת"ז שלו 8-36555988.
10. הציני את שמות המוצרים שהזמננו לאחד מהאירועים בשנת 2018.
11. הציני מוצר שהזמן יותר פעמים מכל מוצר אחר.
12. הציני את המוצר שמחירו זול יותר מאשר כל אחד מה מוצרים מסווג ממנו אחרונה.
13. סכמי לכל לך את הסכם לתשלום עבור כל ההזדמנות שביצע והציני רק לкупות שהסכם לתשלום קטן מהסכום לתשלום אירועים של יאיר ברקוביץ.
14. הציני את המוצר שהכמות שהזמנת ממנו קענה ממוצע הכמות הכללי שהזמין מאותו מוצר.
15. הציני שם לךו ודבר מאכל שהזמין אותו לךו ב-2 ההזדמנות הראשונות.

```

--1
--select [pupilid],sum([payment])"pay",count([datepay])"count"
--from [dbo].[payment]
--group by[pupilid]
--2
--select[bank],[datehafkada ],sum([payment])
--from[dbo].[payment]
--where [datehafkada ]between '09/20/08' and '12/31/08'
--group by[bank],[datehafkada ] ויל נון
--3
--select[kindpay],count([tashid])"count"
--from[dbo].[payment]
--group by[kindpay]
--4
--select[datepay],avg([payment])"avg"
--from[dbo].[payment]
--group by[datepay]
--5
--select [class],count([pupilid])"count"
--from[dbo].[pupils]
--group by[class]
--6
--select top 2[bank],[snif],[cheshbon],[datehafkada ]
--from[dbo].[payment]
--order by[datehafkada ]desc
--7
--select [class],min([numsister])"min",max([numsister])"max"
--from[dbo].[pupils]
--group by[class]
--order by max([numsister])desc בז אטלי גיאת בז
--8
--select count([pupilid])"count",year([datebon])"year"
--from[dbo].[pupils]
--where year([datebon]) between '1990' and '2005'
--group by year([datebon])
--9
--select [bank],[snif],max([payment])"max"
--from[dbo].[payment]
--where [bank]='20'
--group by[snif],[bank]
--10
--select sum([payment])"sum"
--from[dbo].[payment]
--where [status]=1
--11
--select top 10 percent[pupilid],max([numsister])"max"
--from[dbo].[pupils]
--group by[pupilid]
--order by max([numsister])desc

```

*Having - וו זכרן
 וו זכרן, 16/12 - c entering*

Q1

Select last name, first name,
from customers c, order o

Where c.cust_id = o.cust_id and address = "spurc"
order by last name, first name

Select food name, price, type name
from foods f, type t

Where f.type = t.type_id and type_name in ("Co", "Sal", "Pp")

Select ^{top 3 year order date} cust_id, first name, last name, phone, address

From customers c, order o

Where c.cust_id = o.cust_id

Order by order date

- Select first name, phone, max(event_date - order_date)
From customers c, order o

Where c.cust_id = o.cust_id

Group by cust_id

Select food name, quantity

From foods f, order foods o

Where f.food_id = o.food_id and price > 20

Select food name, type name

From foods f, type t

Where f.type = t.type and type name like "%pz%" and type_id = "Co"

Select count(pay-finish)

From

(1)

(2)

(3)

(4)

(5)

(6)

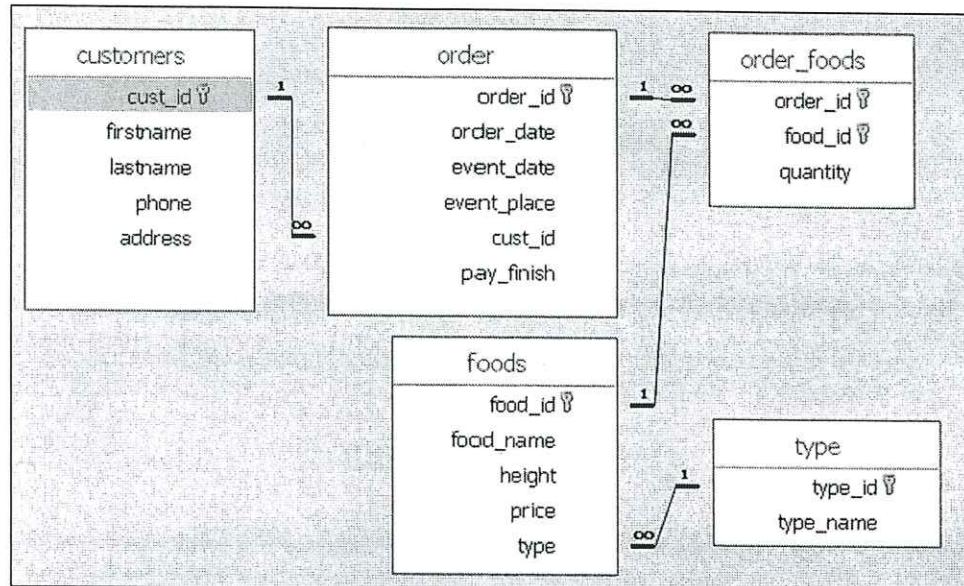
(7)

```
create table teachers
--הגדרת מפתח רצוי מסוג מסוף אוטומטי המתחילה ב-100 ומתקדם ב-1
(teachid int identity(100,1) primary key,
--הגדרת שם מורה מסוג מחרוזת עד 12 תווים , נדרש למלא--
teachname varchar(12) not null,
--הגדרת תאריך לידיה שבברירת מחדל הוא התאריך הנוכחי---
birthdate date default getdate(),
--הגדרת שכר לשעה מסוג מטבע-
wage money,
--הגדרת גובה מסוג מספר עשרוני בעל 5 ספרות ש2 מתחום מימין לנקודה-
height decimal(5,2),
--הגדרת נסיעות מסוג כן/לא-
expirience bit)

create table lessons
(lessonid int identity(100,1),
--הגדרת מפתח דר קוד מורה מטבלה מורות-
teachid int constraint teach_fk foreign key (teachid) references teachers(teachid),
--הגדרת יותר משדה אחד למפתח-
primary key(lessonid,teachid))
```

קייטרינג – Catering

צירוף טבלאות



תרגילים צירוף פנימי - 2 טבלאות:

1. הציגי את שם משפחה ושם פרטי של הלקוחות מהעיר אופקים שהזמיןו בקייטרינג, ממויינים לפי שם משפחה ומיון שני לפי שם פרטי.
2. הציגי את שמות המאכלים ומחרם מהסוגים הבאים: סלטים,תוספות,קינוח.
3. הציגי את פרטי שלושת הלקוחות שהזמיןו אחרוניים.
4. הציגי את שם הלקוח ומספר הטלפון שההפרש בין תאריך ההזמנה לתאריך האירוע הוא הנגדל ביותר.
5. הציגי שם מאכל וכמות הפעמים שהוזמן בתנאי שמחירו מעל 30 נט.
6. הציגי את שמות המאכלים שמתחילה במילה כרוב מסוג סלטים.
7. הציגי את מס' הלקוחות ששסיימו לשלם עבור הזמנה.
8. הציגי את סוג המנה שמס' המאכלים שלא הגבה ביותר.
9. הציגי את מס' המאכלים שהוזמנו בכל תאריך (שימי לב למספר כל מאכל פעם אחת).
10. הציגי את שמות הלקוחות (משורשים) שהזמיןו מהקייטרינג בתאריכים שלפני החודש הנוכחי.
11. הציגי את הכמות הפקסימלית מכל אחד מהמאכלים שהוזמנו.

תרגילים צירוף פנימי של יותר מ-2 טבלאות:

12. הציגי את הסכום הכלול של הזמנה לאירוע שיתקיים באולם השלים בתאריך 10/4/17.
13. הציגי שם לקו, תאריך הזמנה והסכום לתשלום (כמות*מחיר).
14. הציגי את שמות המאכלים שהוזמנו לשמה האחורה.
15. הציגי את שמות הלקוחות במס' הטלפון שלהם מכילה את הספרה 2 והזמין יותר מ-15 מאכלים.
16. הציגי את שמות המאכלים שהזמין הלקוח דוד דרורי וסה"כ המחיר לכל אחד מהם.
17. הציגי את שם המשפחה של הלקוחות מבר שבע ואת מחיר המאכלים שהזמין כל אחד בתנאי שהמחיר בין 20 ל-50 נט.

תרגיל חזרה ב- SQL

לפניך בסיס נתונים לניהול שיעורים פרטיים לתלמידות ע"י מורות, כל מורה יכולה ללמד כמה מקצועות כגון: אנגלית, מתמטיקה ועוד.

שיעורים (קוד שיעור, קוד מורה, שם תלמידה, מקצוע, תאריך, משך זמן בשעות)
Lessons (lessonid, teachid, pupilname, sub, lessondate, hours)

מורות (קוד מורה, שם מורה, קוד עיר, טלפון, תעריף לשעה)
Teachers (teachid, teachname, city, phone, cost)

מורות ומקצועות (קוד מורה, מקצוע)
Teach_sub (teachid, sub)

ערים (קוד עיר, שם עיר)
citys (cityid, cityname)

1. הגדרי את הטבלאות כולל מפתחות וקשרי גומלין בין הטבלאות.
2. עדכני לכל המורות שמלאות יותר מ-2 מקצועות את התעריף בתוספת 15%.
3. הציג שיעורים שנמסרו ע"י מורה ותלמידה בשם זהה.
4. הציג את פרטי המורה שהתעריף שלה גבוה ביותר.
5. הציג לכל מורהסה"כ תשלום לכל חדש על פי השעות שעבדה*תעריף לשעה.
6. הציג לכל מורה את התלמידה שזמן השיעור שלה ארוך ביותר.
7. הציג רשימה כוללת ממינת של שמות המורות והתלמידות ייחדי.
8. הציג מורות שנמסרו שיעורים בשנים קודמות ולא מסרו השנה הנוכחית.
9. מחקי את המורות שלא מסרו שיעורים כלל.
10. צרי טבלה חדשה בשם zipupk ובבה השדות: קוד ושם תלמיד. הוסיף לטבלה זו את רשימת התלמידות מטבלת שיעורים. שיימי לבן כל תלמידה תתווסף רק פעם אחת.
11. הציג שם מורה, תאריך שיעור אחרון, וחישוב הפרש זמן בימים בין תאריך נוכחי לתאריך שיעור אחרון, כולל מורות שלא מסרו שיעורים כלל.
12. הציג 2 מורות שכמות השיעורים שנמסרו הוא הגבוה ביותר.
13. הציג מורות שנמסרו שיעור במקצוע שאינו מופיע כאחד המקצועות שלהן.
14. הציג מורות שנמסרו שיעור לאוותה תלמידה בהפרש של פחות מ-7 ימים.
15. הציג מורות שמשר הזמן בשעות בכל השיעורים שנמסרו לא עולה על 3 שעות.
16. הציג תלמידות שלמדו את אותו מקצוע אצל שני מורים שונים.
17. הציג מורות שמקבלות תעריף נמוך יותר מהתעריף הממוצע בעיר מגורייהם.
18. הציג מורות פופולריות שנמסרו יותר מ-10 שיעורים, או שנמסרו לאוותה תלמידה יותר מ-3 שיעורים או שהן מתחמות ביתר מ-2 מקצועות.
19. הציג מורות שאין פופולריות – שלא נמצאות בשאלתה 18 (ע"י in not או ע"י except).

--
--
--2
update [dbo].[teachers]
set [cost]=[cost]*1.15
where [teachid] in (select teachid
from [dbo].[teach_sub]
group by teachid
having count(sub)>2)
--
--
--3
update [dbo].[teachers]
set cost=cost*1.15
where (select count(*) from [dbo].[teach_sub] where [teachid]=[dbo].[teachers].
teachid)>2

--
--4
select [lessonid]
from [dbo].[lessons] a, [dbo].[teachers] b
where a.[teach]=b.[teachid] and [pupilname]=[teachname]

--
--5
select [pupilname],teachid,[hours]
from [dbo].[lessons] a
where [hours] in
(select max([hours])
from [dbo].[lessons] b
where a.[teachid]=b.[teachid])

--
--6
select a.teachid,sum([hours]*[cost])
from[dbo].[teachers] a join [dbo].[lessons] b
on a.teachid=b.teachid
group by a.teachid,month([lessondate]),year([lessondate])

--
--7
select [teachname]
from[dbo].[teachers]
union
select [pupilname]
from[dbo].[lessons]

--
--8
select [teachname]
from[dbo].[teachers] a, [dbo].[lessons] b
where a.teachid=b.teachid and year(getdate())>year([date_lesson])
and a.teachid not in
(select teachid
from[dbo].[lessons]
where year(getdate())=year([date_lesson]))

--
--9
delete [dbo].[teachers]
where teachid not in
(select teachid
from [dbo].[lessons])

--
--11
select [teachname],max([lessondate]),DATEDIFF(d,max([lessondate]),GETDATE())
from [dbo].[teachers] a left join [dbo].[lessons] b on a.teachid=b.teachid
group by [teachname]

--12

```

select top 2 a.* ,COUNT([lessonid])
from [dbo].[teachers] a, [dbo].[lessons] b where a.teachid=b.teachid
group by [teachid],[teachname], [city] ,[phone], [cost]
order by COUNT([lessonid])

--13--
--distinct לאו וgroup by ניתן להשוויה בלבד ניתן
select [teachid]
from [dbo].[lessons] le
where [sub] not in (select [sub]
                     from [dbo].[teach_sub]
                     where le.[teachid]=[dbo].[teach_sub].[teachid])
group by teachid

--14--
select [teachname]
from [dbo].[teachers] a, [dbo].[lessons] b where a.teachid=b.teachid
and a.[teachid] in(select [teachid]
                     from[dbo].[lessons] c
                     where b.studentname=c.studentname
                     and a.teachid=c.teachid
                     and DATEDIFF(d,b.date_lesson,date_lesson)<7)

--15--
select [teachid]
from [dbo].[lessons] le
where [teachid] not in (select [teachid]
                     from [dbo].[lessons]
                     where [hours]>3)

--16--
select [studentname]
from [dbo].[lessons] le
where [sub] in (select [sub]
                     from [dbo].[lessons]
                     where le.studentname=studentname
                     and le.teachid!=teachid)
--אפשר גם CL
select [studentname]
from [dbo].[lessons] le
where exist (select [sub]
                     from [dbo].[lessons]
                     where le.studentname=studentname
                     and le.teachid!=teachid
                     and le.sub=sub)

--17--
select t.*
from [dbo].[teachers] t ,[dbo].[city] c
where t.city=c.cityid
and [cost]<(select AVG([cost])
                     from [dbo].[teachers], [dbo].[city]
                     where city=cityid
                     and city=c.cityid)

--18--
select distinct t.*
from [dbo].[lessons] le join [dbo].[teachers] t
on le.teachid=t.teachid
where
teachid in(select [teachid]
                     from [dbo].[lessons]
                     group by teachid
                     having COUNT(1)>10)
or teachid in (select [teachid]
                     from [dbo].[lessons]
                     where le.studentname=studentname
                     group by teachid,studentname
                     having COUNT(1)>3)
or teachid in (select [teachid]

```

```
from [dbo].[teach_sub]
group by teachid
having COUNT(1)>2

--ללא--
select distinct t.*
from [dbo].[teachers] t join [dbo].[lesson] le
on t.teacherid=le.teachid
where (select count([lessonid]) from [dbo].[lesson] l where l.[teachid]=t.teachid)>
      10
or (select count([student_name]) from [dbo].[lesson] l where l.[teachid]=t.teachid
     and [student_name]=le.student_name)>3
or (select count([sub]) from [dbo].[teach_sub] where [teachid]=t.teachid)>2

--19--
select * from [dbo].[teachers]
except
select distinct t.*
from [dbo].[teachers] t join [dbo].[lesson] le
on t.teacherid=le.teachid
where (select count([lessonid]) from [dbo].[lesson] l where l.[teachid]=t.teachid)>
      10
or (select count([student_name]) from [dbo].[lesson] l where l.[teachid]=t.teachid
     and [student_name]=le.student_name)>3
or (select count([sub]) from [dbo].[teach_sub] where [teachid]=t.teachid)>2
```

and one exists - 21
and alterable - 32
33 now 910 34
1'3 14 - 412 - 34 2112
1'3 15 - 35 2112

המוצרים שהאמנו ושלא האמנו--

```
select [proddesc]
from [dbo].[products_tbl] p left outer join [dbo].[ORDERS_TBL] o on p.prodid=o.prodid
```

המוצרים שלא האמנו--

```
select [proddesc]
from [dbo].[products_tbl] p left outer join [dbo].[ORDERS_TBL] o on p.prodid=o.prodid
where [ORDNUM] is null
```

המוצרים שהאמינו לקוחות גם מוצרים שלא האמנו וגם לקוחות שלא האמינו--

```
select [proddesc],[CUSTNAME]
from [dbo].[products_tbl]p full outer join [dbo].[ORDERS_TBL]o on p.prodid=o.prodid full outer join [dbo].[CUSTOMERS_TBL] c on o.CUSTID=c.custid
```

צירוף הטבלה אל עצמה מציגה עובד לצד מנהל שהוא גם עובד--

```
select e.[lastNAME],m.[lastNAME] "מנהל"
from [dbo].[EMPLOYEE_TBL] e ,[dbo].[EMPLOYEE_TBL] m
where e.managerid=m.empid
```

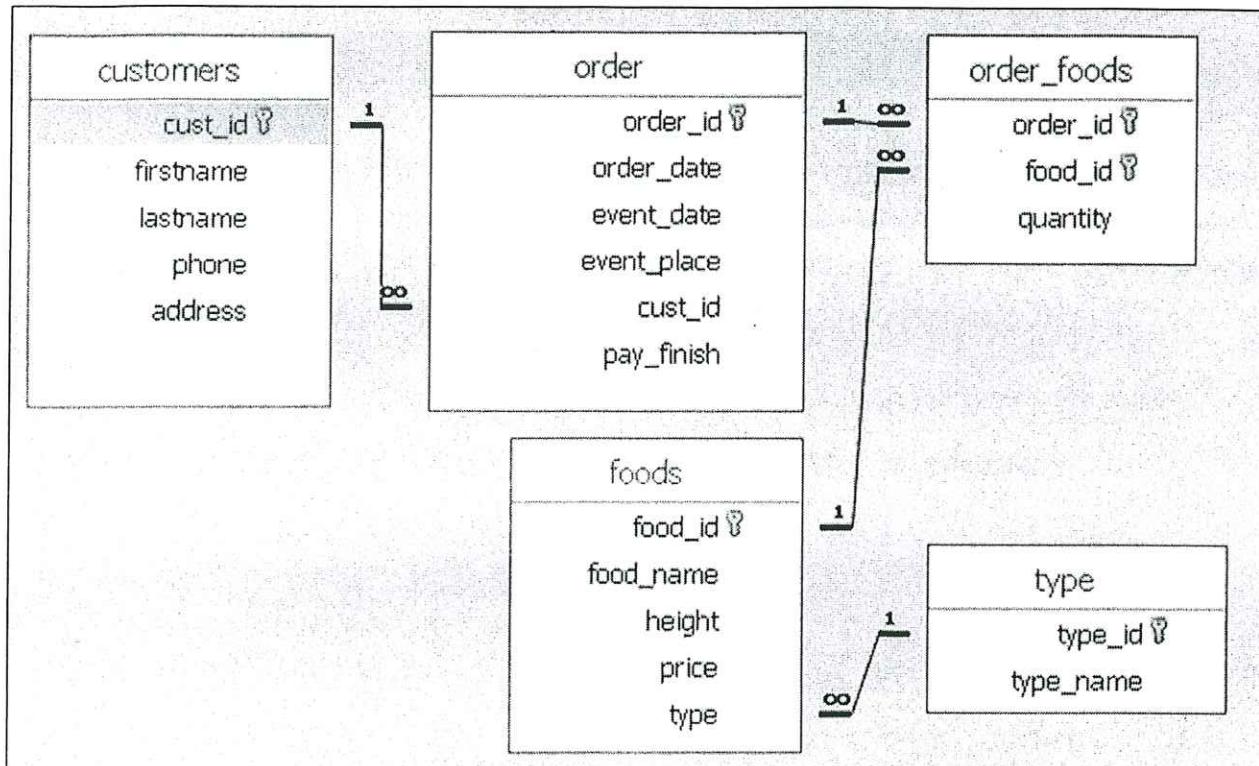
צירוף זוגות אפשריים ללא חזרה הגרים בירושלים--

```
select e.[lastNAME],m.[lastNAME]
from [dbo].[EMPLOYEE_TBL] e ,[dbo].[EMPLOYEE_TBL] m
where e.empid<m.empid and e.city='ירושלים' and m.city=''
```

```
--2
update [dbo].[teachers]
set[cost]*=1.15
where [teacher_id]in(select [teacher_id]from[dbo].[teachers_subs]group by
[teacher_id] having count( [sub])>2)
--3
select *
from[dbo].[lesson]l,[dbo].[teachers]t
where t.[teacher_id]=l.[teacher_id] and teacher_name like '%'+'%'+[pupil_name]+'%'
--4
select top 1 *
from[dbo].[teachers]
order by[cost] desc
--5
select sum([lesson_hours])*t.[cost]'pay',t.[teacher_id]
from[dbo].[lesson]l,[dbo].[teachers]t
where t.[teacher_id]=l.[teacher_id]
group by t.[teacher_id],MONTH([lesson_date]),[cost]
--6
select[pupil_name],[teacher_id],[lesson_hours]
from[dbo].[lesson]
where[lesson_hours]in(select max([lesson_hours])'max'from[dbo].[lesson]group by
[teacher_id])
--7
select [pupil_name]
from[dbo].[lesson]
union
select[teacher_name]
from[dbo].[teachers]
--8
select [teacher_id]
from[dbo].[lesson]
where datepart(YEAR,getdate())not in()
```

חזרה - קייטריניג

פונקציות, פרוצדורות, סמנטים, טירגרים, טיפול בשגיאות



1. צרי פונקציה המקבלת שם ל��וח בודקת אם הליקוּן הזמן מתקיינֵג בחודש האחרון אם כן מחזירה את מס' ההזמנות של הליקוּן אחרית תחרת תחזיר 0.
2. צרי פונקציה המקבלת מס' הזמנה ומחזירה את שמות המאכלים שהוזמנו בה משורשרים במחוזות אחת, כשביניהם פסיק.
3. צרי פונקציה המקבלת שם מאכל ומחזירה את רשימת ההזמנות שהוא מופיע בהן.
4. צרי פונקציה המקבלת שם מאכל בודקת את הכמות המוזמנת ממנו, אם הכמות בין 100 ל-200 יוחזר 'מאכל שכיח', אם הכמות בין 50 ל-100 יוחזר 'מאכל רגיל', אחרת יוחזר 'מאכל נדיר'. (השתמשי במקרה).
5. הוסיף שדה לטבלה ההזמנות בשם difference. כתבי פונקציה שמקבלת מס' הזמנה וمعدכנת לו את השדה difference להפרש שבין תאריך ההזמנה לתאריך האירוע. ותחזיר טבלה עם השדות מס' הזמנה, שם המזמין וההפרש.

6. כתבי פרוצדורה המקבלת תאריך וסוג מאכל ומוחקתו את כל המאכלים מהסוג שהתקבל
שהוזמנו מאז התאריך שהתקבל.
7. כתבי פרוצדורה שתקבל שם לקוח ותאריך אירוע ותוסיף שורה להזמנת הלוקו שקיבלה
לטבלת הזמנות, לפני ההוספה יש לוודא שתאריך האירוע בשנה הנוכחית. אם לא יש
להפיק הודעה שגיאה יש להזמין את האירוע בתחילת השנה הבאה.
8. צרי Cursor העובר על טבלת המאכלים, אם מחיר המאכל קטן מ-10 ₪, מוסיף אותו
להזמנה מס' 5 בכמות 3.
9. צרי Cursor שייעבור על טבלת ליקוחות וימחק את הליקוחות שאין להם הזמנות.
10. צרי טריגר בעת עדכון של השدة מחיר אם המחיר גדול מ-30 ₪ יעדכן השدة מחיר
להנחה של 5% על המחיר הנוכחי.
11. צרי טריגר בעת הוספת מאכל, הוא יתווסף לאירוע שהזמן אחרון.
12. הציגי את הזמנה הראשונה בכל אלומ.
13. הציגי את המוצר היקר ביותר והזול ביותר בכל הזמן.
14. צרי פרוצדורה שמקבלת קוד לקוח, תאריך הזמנה ותאריך אירוע ומוסיפה הזמנה עם
 הפרטים הנ"ל. לפני ההוספה יש לבדוק שתאריך הזמנה קטן מתאריך האירוע אם לא
תופיע הודעה שגיאה.
15. צרי טריגר בעת מחיקת לקוח אם קיימות הזמנות ללוקו זה יש לבטל את הפעולה.
16. צרי טבלת הצלבות שתציג לכל לקוח את סה"כ מחיר שצרכיר לשלם על כל מאכל שהזמן.