

אקסס 97

מתכנתים חלק ג'



כתיבה ועריכה : אפי ברק

גרסה 2.00

תוכן עניינים

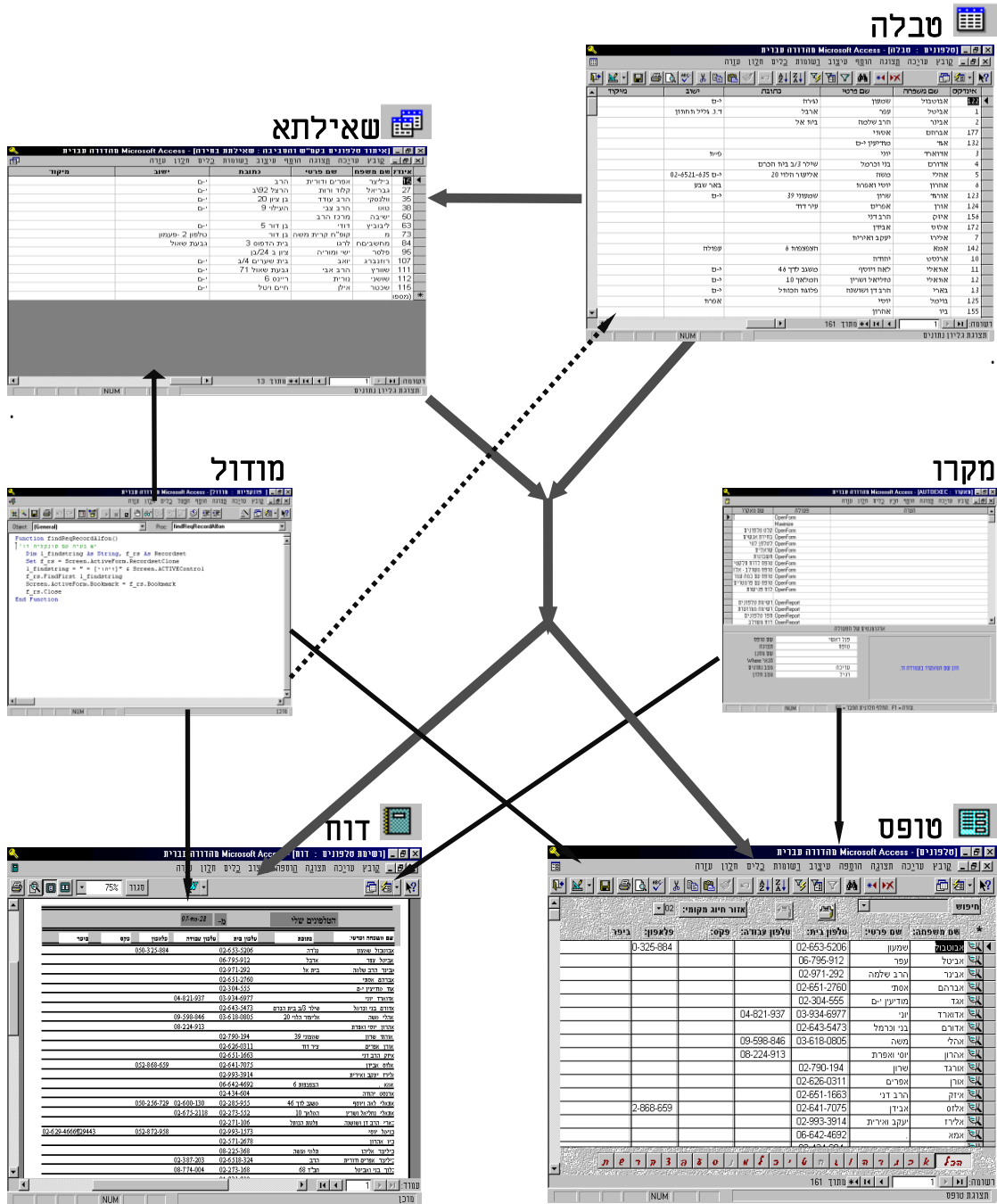
6	1. טפסים מתקדם
6	1.1. טופס עם פרמטרים
9	1.2. טופס עם סינון רשומות לפי אותיות האלף בית בעזרת כפתורים
13	1.3. שילוב קוד בטפסים
14	1.4. מיון וחיפוש לפי כל שדה
15	1.5. טופס מרובה עמודים
16	1.6. שילוב בקר לסיכומי רשומות
17	1.7. פתיחת טופס מטופס לפי קריטריונים
21	1.8. חיפוש ואיתור רשומות מתיבה משולבת
23	2. דוחות מתקדם
23	2.1. ספרור אוטומטי
23	2.2. הצגת תמונה בדוח
23	2.3. דוח סלקטיבי א'
24	2.4. דוח סלקטיבי ב'
27	3. מודולים (קוד ויזואל בייסיק VBA) - בסיסי
27	3.1. <u>הקדמה</u>
29	3.2. <u>איתור פקודות ופונקציות ותחבירם</u>
30	3.3. הצהרות ((declaration והגדרות (definition)
31	3.4. קבועים ומשתנים
32	3.5. טיפוסים משתנים וגודלם
36	3.6. גישה לאיברי האובייקט בשימוש בפקודת With
38	3.7. <u>מודולים Modules</u>
39	3.8. שגרות ופונקציות - SubRoutine & Function
44	3.9. <u>מילים שמורות</u>
45	3.10. <u>אופרטורים (מפעילים)</u>
47	3.11. <u>תוי בקרה כללים</u>
48	4. משפט השמה ומשפטי בקרה
49	4.1. משפט IF
49	4.2. משפט CASE
51	4.3. לולאת FOR
51	4.4. לולאת FOR EACH
52	4.5. לולאת Do Loop
52	4.6. לולאת DO While Until
53	4.7. לולאת WHILE
53	4.8. יציאה, דלג ל, וכתובת
55	4.9. <u>עבודה בסביבת הפיתוח</u>
56	4.10. שילוב קוד בטפסים/דוחות
56	4.11. <u>לכידת שגיאות</u>

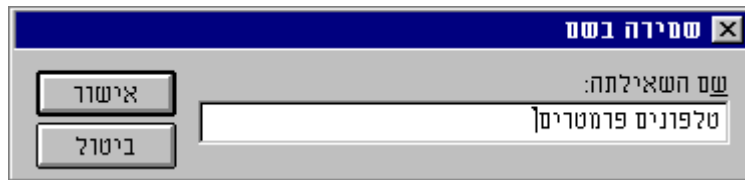
59	4.12. אובייקט הטופס כאוסף פקדים
60	4.13. מבנה בקרה – חזרה תוך שימוש באובייקט הטופס
62	4.14. בדיקת רצף באמצעות פקד תיבת רשימה
63	5. אובייקטים ואוספים
64	5.1. מבנה בסיסי של האובייקטים שבאקסס
65	5.2. מבנה של אובייקט אוסף
67	5.3. Application
69	5.4. DBEngine
71	5.5. Screen
72	5.6. Forms
73	5.7. TableDefs
74	5.8. Containers
78	5.9. Recordset
91	6. סיטמאות
91	6.1. קובץ קבוצות העבודה - System.Mdw
92	6.2. אבטחת מסד נתונים
96	6.3. יצירת מסד נתונים מאובטח
97	7. נספחים
97	7.1. תפריטים מותאמים
98	7.2. בונה התפריטים
99	7.3. מנהל לוח מתגים
100	7.4. פיתרונות בעזרת קוד
102	7.5. גרסת RunTime
103	7.6. שיכפול מסד נתונים
104	7.7. פיצול מסד נתונים
104	7.8. מנתח ביצועים
105	7.9. מנתח טבלאות
106	7.10. סכמה של מסדים מפוצלים ומשוכפלים
107	7.11. טבלת השוואות בין אקסס לבין וויזואל בייסיק
109	7.12. עיצוב תוכנה – הנחיות כלליות
110	7.13. תצורות בסיסי נתונים

סימנים מוסכמים בחוברת

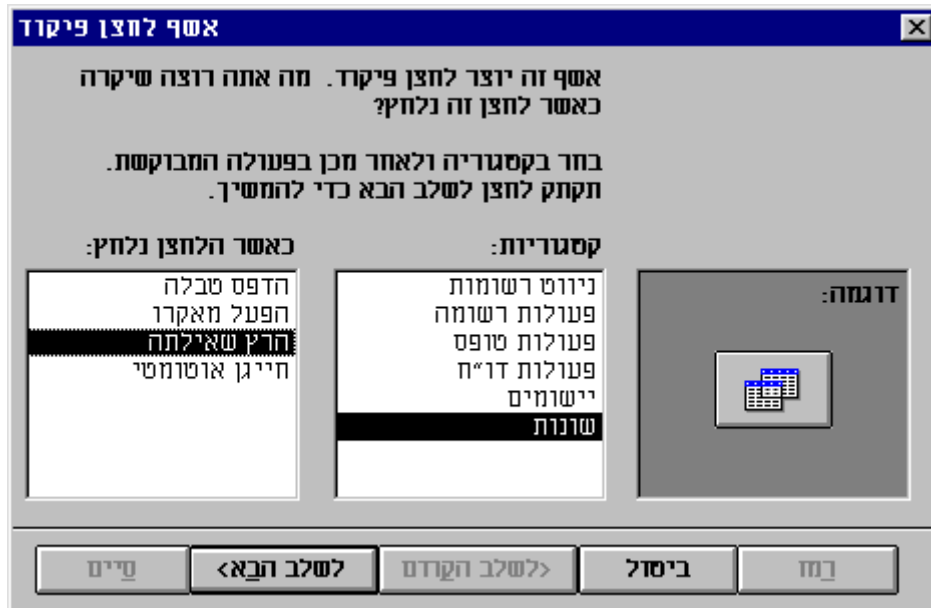
מתקדם 	טיפ 	הערה 
שים לב 	מקש קיצור 	תרגיל 
סיום פעולה 	הגבלה 	אזהרה 
	שאלה 	סיכום 

סכמת קשרי האובייקטים במסד נתונים

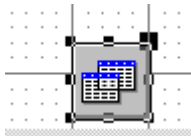




8. יש ליצור פקד כפתור נוסף שבמודול שלו תהיה פקודה להרצת שאילתא. אפשר להשתמש באשף פקדים כמודגם בתמונה.



9. נוצר פקד «כפתור» שבמודול הפרטי שלו יש את הפקודה הבאה:



Sub שאילתא_Click ()

DoCmd OpenQuery "טלפונים פרמטרים", A_NORMAL, A_EDIT

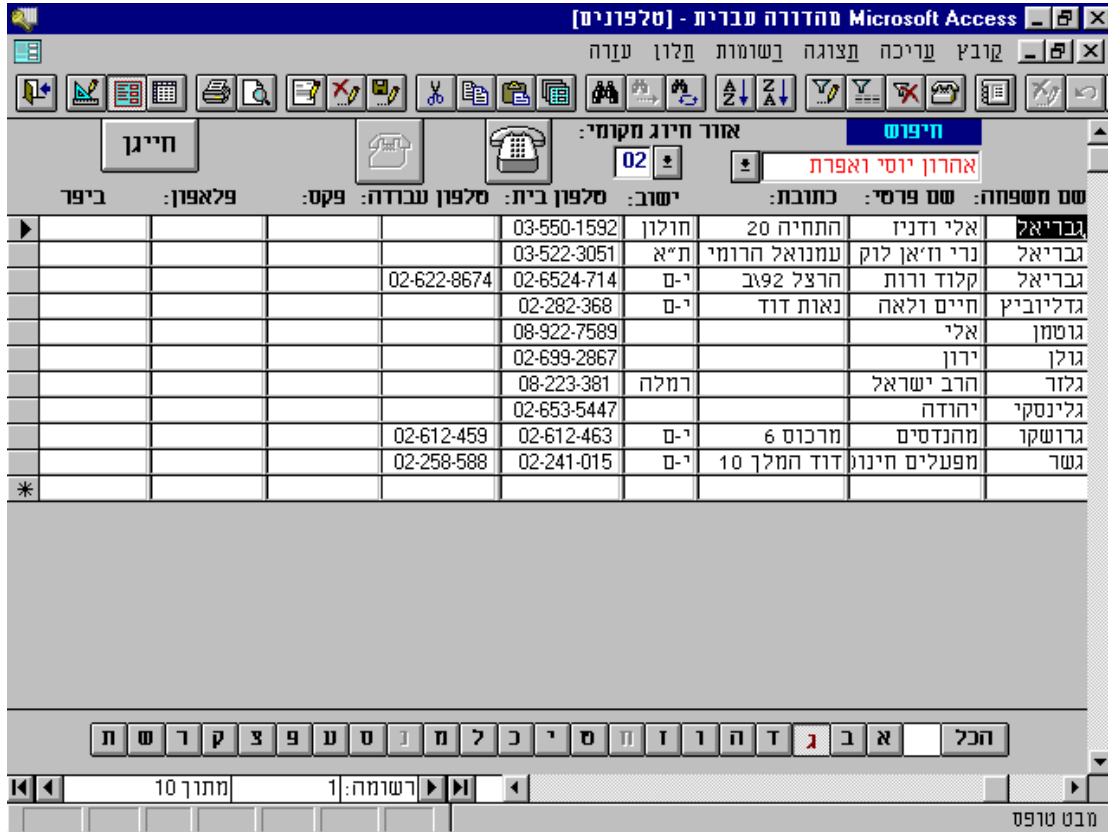
* פתיחה אוטומטית של תיבת רשימה, בזמן קבלת מוקד. אם רוצים לפתוח את תיבת הרשימה כל פעם שהמשתמש נכנס לפקד, יש לכתוב את שורות הקוד הבא, באירוע "בעת קבלת מוקד", שמדמה הקלדה של המקש F4 שהוא זה שפותח תיבות רשימה.

```
Sub_2 GotFocus()
SendKeys "{F4}"
End Sub
```


2.1. טופס עם סינון רשומות לפי אותיות האלף בית בעזרת כפתורים

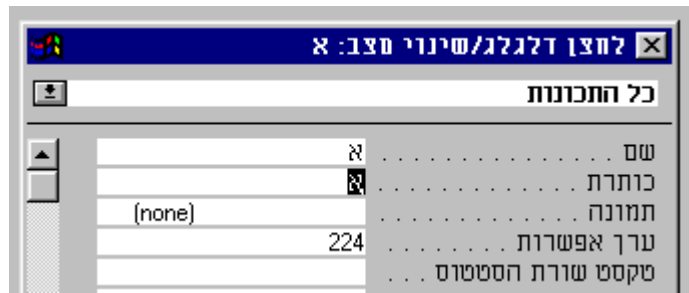
המטרה ליצור טופס שיהיה אפשר לקבל חתך רשומות לפי אות הפתיחה של הרשומה. למשל: יופיעו רק רשומות ששמות המשפחה שלהם מתחילים באות המבוקשת בדוגמא אנו רואים שמופיעים רק רשומות ששמות משפחה מתחילים באות "ג".

ראשית: תכונת «מבט ברירת מחדל» של הטופס צריך להיות «טפסים רציפים».



1. בשלב הראשון אנו צריכים ליצור פס אותיות שמורכב מפקד קבוצת אפשריות ובו כפתורי אותיות מ"א" ועד "ת". כאשר ערך של כל כפתור אות יהיה הערך האסקיי שלו. לדוגמא כפתור האות "א" יהיה 224. הכפתור «הכל» צריך להיות שווה ל 422 שהוא הערך האסקיי של כוכבית. (בכדי לקבל את כל הרשומות). פקד קבוצת אפשרות בנוי שכאשר אחד מהכפתורים שבו נלחץ, הוא מקבל את ערכו.





2. בשלב השני אנו צריכים לשנות את מקור הרשומה של הטופס באופן דינמי, מתוך הקוד.

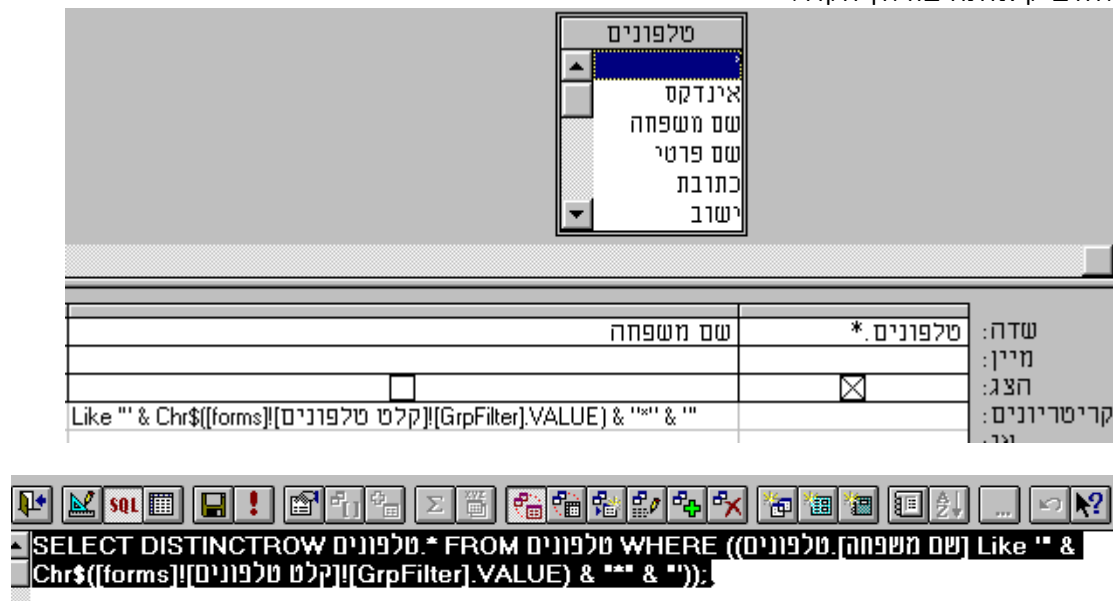


לשם כך אנו יוצרים שגרה באירוע «בזמן תקתוק» בקבוצת האפשרויות. האירוע יפעל כל פעם שהמשתמש מקליק על אחד מכפתורי האותיות. האירוע ישנה את מקור הרשומה של הטופס, כך שישאב רק חתך רשומות שמתחיל באות הכפתור שנלחץ. מומלץ ליצור את השאילתא בחלון «שאילתות» ולאחר מכן, להעתיק את קידוד ה SQL אל מסך קוד בייסיק. במקרה שלוחצים על כפתור «הכל» שערכו 42 הטופס יציג את כל הרשומות, כי ביטוי החתך יהיה כמו «**», שאמור לתת את כל הערכים.

```
Sub GrpFilter_Click ()
Me.RecordSource = "SELECT DISTINCTROW * FROM טלפונים WHERE (([שם משפחה] Like Chr$([forms]![טלפונים]![GrpFilter].[VALUE]) & '*')) ORDER BY פרטי.שם פרטי.שם משפחה);"
End Sub
```

טיפ

כדאי ליצור את השאילתא בחלון שאילתות, ולהעתיק אותה ללוח הגזירות, ולאחר מכן להדביק אותה בחלון הקוד.





יש לשים לב לאופן הכתיבה של השאילתא. השאילתא צריכה להיות מושמת כמחרוזת, לכן בתחילתה ובסופה יהיו גרשיים. אם יש ביטוי בתוך השאילתא שצריך להיות בגרשיים בדוגמא כוכבית "*" , יש להחליף את הגרשיים בגרש.

```
Me.RecordSource = "SELECT DISTINCTROW * FROM טלפונים WHERE  
((שם משפחה) Like Chr$([forms]![טלפונים]![GrpFilter].[VALUE]) & '*')) ORDER  
BY פרטי.טלפונים.שם משפחה.טלפונים.שם פרטי;"
```

3. בשלב השלישי אנו רוצים שרק כפתורי אותיות שיש להם נתונים בטבלה יהיו מאופשרים. אנו יוצרים אירוע חדש בעת טעינת הטופס. לכן אנו סורקים את הטבלה אות אות, בודקים אם יש רשומות שמתחילות בכל אחת מאותיות האלף בית, ואותיות שאין להם נתונים בטבלה "מכובות". טיפול מיוחד באותיות מנצפ"ך הסופיות "סןןן" = שורת הIF.

```
Sub Form_Load ()  
For i = 224 To 250  
If i = 234 Or i = 237 Or i = 239 Or i = 243 Or i = 245 Then GoTo daleg  
Me(Chr(i)).Enabled = DCount("*", "[שם משפחה]", "[טלפונים] LIKE " & Chr(i) & "*" )  
daleg:  
Next i  
End Sub
```

1.2.1. טופס אותיות דרך ב.

בשלב הראשון יש ליצור מחרוזת סינון. בכדי ליצור אותה אנו משימים אותה לתכונת הסינון.

פתרונות אפשריים בשורת הסינון

1. Me.Filter = "left([שם משפחה],1)=' ' & Chr\$(GrpFilter) & ' ' " .1
2. Me.Filter = "left([שם משפחה],1)=' ' " & Chr\$(GrpFilter) & " ' " .2
3. Me.Filter = "left([שם משפחה],1)= Chr\$(" & GrpFilter & ")" .3
4. Me.Filter = "left([שם משפחה],1)= Chr\$(forms![אלפון]! GrpFilter) .4
5. Me.Filter = "[שם משפחה] = like ' ' & Chr\$(Me.GrpFilter) & '*' " .5

5. אפשר להשתמש בפונקציה BuildCriteria - שמקבלת שם שדה וביטוי ויוצרת ביטוי למסנן - שבאובייקט Application שתחבירה הוא כזה

6. BuildCriteria ("ביטוי", סוג שדה, "[שם שדה]") .6
7. Me.Filter = BuildCriteria("left("& "[שם משפחה]" & ",1)", dbText, Chr\$(Me.GrpFilter)) .7
8. Me.Filter = BuildCriteria(" [שם משפחה] ", dbText, "like " & Chr\$(Me.GrpFilter) & "*") .8

כמובן שפתרון 4 הוא פחות אלגנטי, מפני שהוא מתאים רק לטופס הנוכחי, ובכל פעם שמשנעים אותו לטופס אחר, צריך לשנות בו את שם הטופס.

- כמו כן ישנה אפשרות להשתמש בפונקציית STR שהופכת מספרים למחרוזות -

בשלב השני יש להפעיל את תכונת המסנן על ידי שאנו משימים לתכונת מסנן מופעל את הערך אמת.

```
Me.FilterOn = IIf(GrpFilter = 100, False, True)
```

כאשר פקד "כל הרשומות" מכיל את הערך 100, והוא מבטל את הסינון כאשר לוחצים עליו.

אנו יכולים לשפר את הפתרון ולאפשר רק כפתורים של אותיות תחיליות שקיימות בשדה "שם משפחה" ע"י שורות הקוד הבאות

```
Dim i As Integer
```

```
For i = 224 To 250
```

```
    If i = 234 Or i = 237 Or i = 239 Or i = 243 Or i = 245 Then GoTo daleg
```

```
    Me(Chr(i)).Enabled = Dcount("*", "[אלפון]", "[שם משפחה]", like "*" & Chr(i) & "*")
```

```
daleg:
```

```
Next I
```

הסבר

ישנן, שתי צורות תחביר את כתיבת שם שדה. הראשונה מאפשרת להשתמש במשתנה

```
Me("שדה1").Enabled = TRUE
```

```
Me![שדה1].Enabled = TRUE
```

```
Me("א").Enabled
```

הוא בעצם הביטוי
Me("א").Enabled

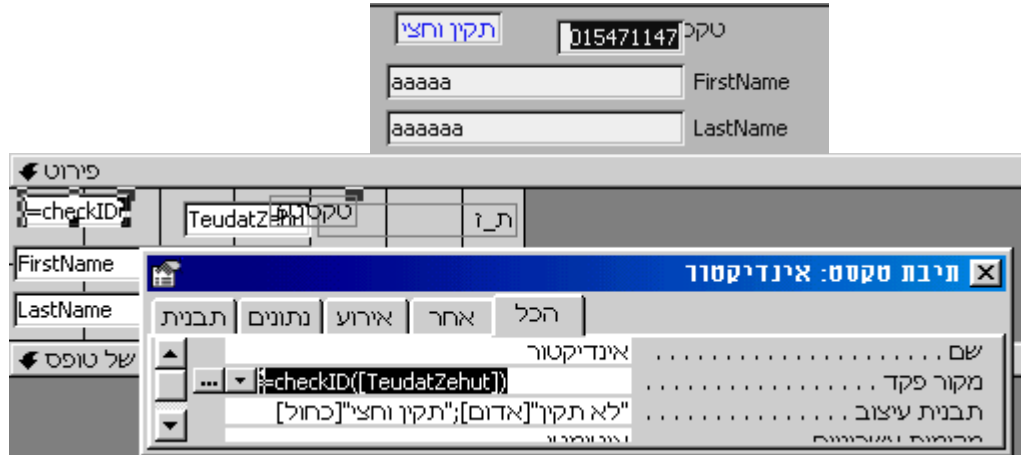
במקרה הראשון
Me(Chr(224)).Enabled

לכן הביטוי
Me(Chr(I)).Enabled

מה שמצביע על הפקד ששמו "א", ובעזרת פונקציית השאילתא אנו סורקים את הרשומות שמתחילות באותה אות, ובמקרה ואין רשומות שמתחילות באותה אות הערך המוחזר הוא 0 ששווה לביטוי FALSE ובמקרה שיש מספר כל שהוא של רשומות הערך הינו TRUE

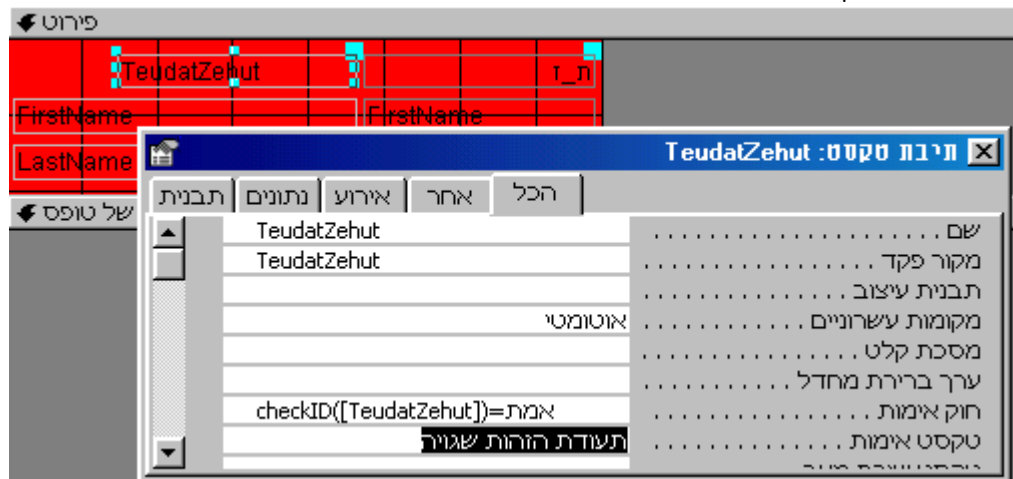
3.1. שילוב קוד בטפסים

נתונה לנו פונקציה לאימות תקינות של תעודת זהות ואנו רוצים לשלבה בטופס.
1. ישירות בפקד



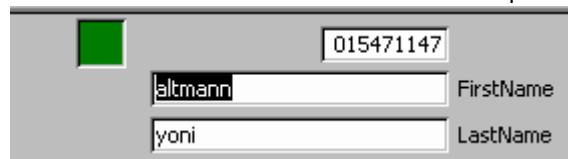
ללא שימוש בקוד, אלא באמצעות תכונות מקור פקד, ותבנית עיצוב. יתרונו שאין צורך ל"הסתבך" בקוד ושיש עדכון אוטומטי של המערכת.

2. בתכונת חוק אימות



אי אפשר יהיה לצאת מהפקד או לעדכן רשומה לפני שמקלידים את המספר הנכון.

3. בקוד



יצירת אינדיקטור גרפי שיראה לנו האם מספר הזהות הינו תקין או לא. באמצעות שני פקדים, ובאמצעות פקד אחד בלבד.

Private Sub Form_Current()	באירוע בנוכחי, שהוא בזמן מעבר רשומה
Public Function ההחלפה בין שני פקדים Me.takin.Visible = checkID(Me.TeudatZehut) Me.loTakin.Visible = Not Me.takin.Visible End Function	Public Function שני צבעים של פקד אחד If checkID(Me.TeudatZehut) Then Me.takin.BackColor = vbGreen Else Me.takin.Visible = vbRed End If End Function
Me.takin.BackColor = IIf(checkID(Me.TeudatZehut), vbGreen, vbRed)	

4.1. מיון וחיפוש לפי כל שדה

המטרה פקדים שמאפשרים מיון וחפוש בכל שדה. התיבה המשולבת העליונה מציגה רשימה של השדות שבטופס, וכאשר בוחרים שדה מסויים המיון של הרשומות נערך לפיו, בעוד שבתביבה המשולבת השנייה אנו מקבלים רשימת ערכים שמופיעים בשדה, ואנו יכולים לחפש רשומות לפי ערכים אלו.

לצורך העניין ניצור תיבת רשימה אחת ששמה "מיון" כאשר התכונות הבאות יהיו מוגדרות כאמור

לאחר מכן ניצור תיבת רשימה שניה ששמה יהיה "חיפוש" בעזרת האשף, כאשר נבחר שני עמודות בחיפוש, כאשר עמודת האינדקס חבויה.

לאחר מכן יש לכתוב את הקוד הבא:

```
Private Sub מיון_Click()
'מיון לפי שדה ושלוף רשימה של נתוני השדה לביצוע חיפוש
Dim fieldName As String
fieldName="[" & מיון & "]"
Me.OrderBy= "[אלפון]" & fieldName
Me.OrderByOn = True
Me.חיפוש.RowSource = "SELECT DISTINCTROW [אינדקס].[אלפון], " & fieldName & _
FROM [אלפון] WHERE (( " & fieldName & ") Not Is Null) ORDER BY " & fieldName
Me.חיפוש.Requery
End Sub

Private Sub חיפוש_AfterUpdate()
'מצא את הרשומה שתואמת את הפקד.
Me.RecordsetClone.FindFirst " = [אינדקס]" & Me![חיפוש]
Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub
```

5.1 . טופס מרובה עמודים

המטרה ליצור טופס ארוך שיש בו כמה עמודים. (שנוצרים ע"י הפקד סוף עמוד) ואפשרות למשתמש לדלג מעמוד לעמוד ע"י הקשה על כפתורים. בדוגמא יש טופס עם חמש עמודים, ולחיצה על כל כפתור מקפיצה את המסך לאזור המתאים בטופס.

1. יש ליצור את הטופס עצמו ואת הדפים בטופס ע"י הפקד המתאים. פקד עמוד חדש.



יש לייצר פקד קבוצת אפשרויות ששמו {מיתוג} שבו כפתורים דו מצביים. הכפתור הראשון מקבל סדר 1 השני 2 וכן הלאה.



3. בפקד קבוצת האירועים (ששמו מיתוג), באירוע «בעת תקתוק» אנו יוצרים את השגרה הבאה, כשהפרמטר שלו הינו הערך של פקד «מיתוג» עצמו.

```
Sub מיתוג_click ()
    Me.GotoPage me![מיתוג]
End Sub
```

4. יש לבטל סרגל גלילה של הטופס, בכדי למנוע שוטטות מדף לדף. בתכונת טופס «פטי גלילה» יש לבחור באופציה «אף אחד».

5. יש לבטל מקשי דפדוף כמו מקש PageDown ע"י כתיבת הקוד באירוע טופס. בכדי שלחיצת המקש תגיע לשגרה, יש לבחור תכונת «תצוגת מקש מקדימה» באירועי טופס לכן.

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 34 Then KeyCode = 0
End Sub
```

6.1. שילוב בקר לסיכומי רשומות

348,268.57	דולר
7,181.00	דולר קנדי
1,261.00	לי ש"ט
100.00	מרקים
93,620.00	פרנק
1,878,307.41	ש"ח

סיכומים	שם
	מקור הפקד
טבלה/שאילתה	סוג מקור שורה
SELECT [מטבע], [טופס פשפ], [מטבע]	מקור שורה
2	מספר העמודות
לא	כותרות עמודות
2 ס"מ; 2 ס"מ	רוחבי עמודות
1	עמודה מוצמדת

SumOf	כ"ס	מטבע	שדה:
Expression	Group By		סה"כ/סיכום:
			מיין:
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		הצג:

((תרומות - חתך Format(Sum [תרומות - חתך תרומות לפי טופס פשפ].מטבע, SELECT
 סכום תרומה[SumOf AS [Standard'תרומות לפי טופס פשפ].(סכום תרומה),
 FROM [תרומות - חתך תרומות לפי טופס פשפ]
 GROUP BY [תרומות - חתך תרומות לפי טופס פשפ].מטבע;

7.1 פתיחת טופס מטופס לפי קריטריונים

המטרה היא לפתוח מטופס «טפסים רציפים» טופס במבט על, כאשר מופיע בה הרשומה הנבחרת. בדוגמא אנו נפתח מטופס «אלפון» שכבר הכרנו את טופס «אלפון מבט על»

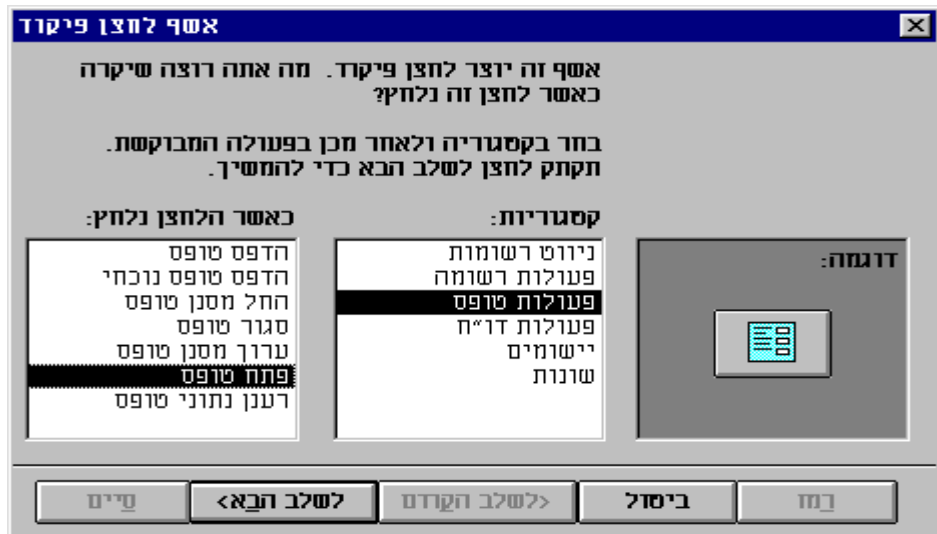
1. ראשית יש ליצור את טופס «אלפון מבט על» כאשר שדה הקשר בין שני הטפסים יהיה שדה «אינדקס»



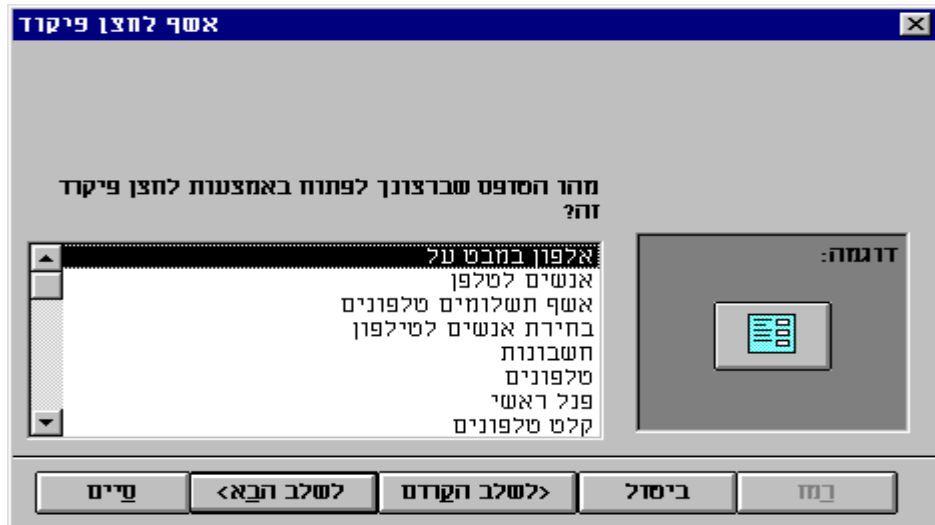
2. יש לעבור לטופס השני וליצור פקד כפתור פקודה חדש כאשר כפתור האשף דולק



3. יש לבחור «פעולות טופס» «פתח טופס» כנראה בתמונה



4. יש לבחור בטופס אותו אנו רוצים לפתוח.



5. יש לבחור באופציה הראשונה



6. יש לבחור את פקדי הקישור, בכדי למצוא רשומה תואמת בטופס שייפתח



7. יש לבחור תמונת כפתור וסיום



8. נוצר לנו בטופס כפתור שהקשה עליו תפתח לנו את הטופס השני



9. הקוד שמאחורי הכפתור הינו זה

```
Sub 53לחצן_Click ()
```

```
On Error GoTo Err_53לחצן_Click
```

```
Dim DocName As String
```

```
Dim LinkCriteria As String
```

```
DocName = "אלפון במבט על"
```

```
LinkCriteria = "[קלט טלפונים]=[איונדקס]=Forms![איונדקס]"
```

```
DoCmd OpenForm DocName, , LinkCriteria
```

```
Exit_53לחצן_Click:
```

```
Exit Sub
```

```
Err_53לחצן_Click:
```

```
MsgBox Error$
```

```
Resume Exit_53לחצן_Click
```

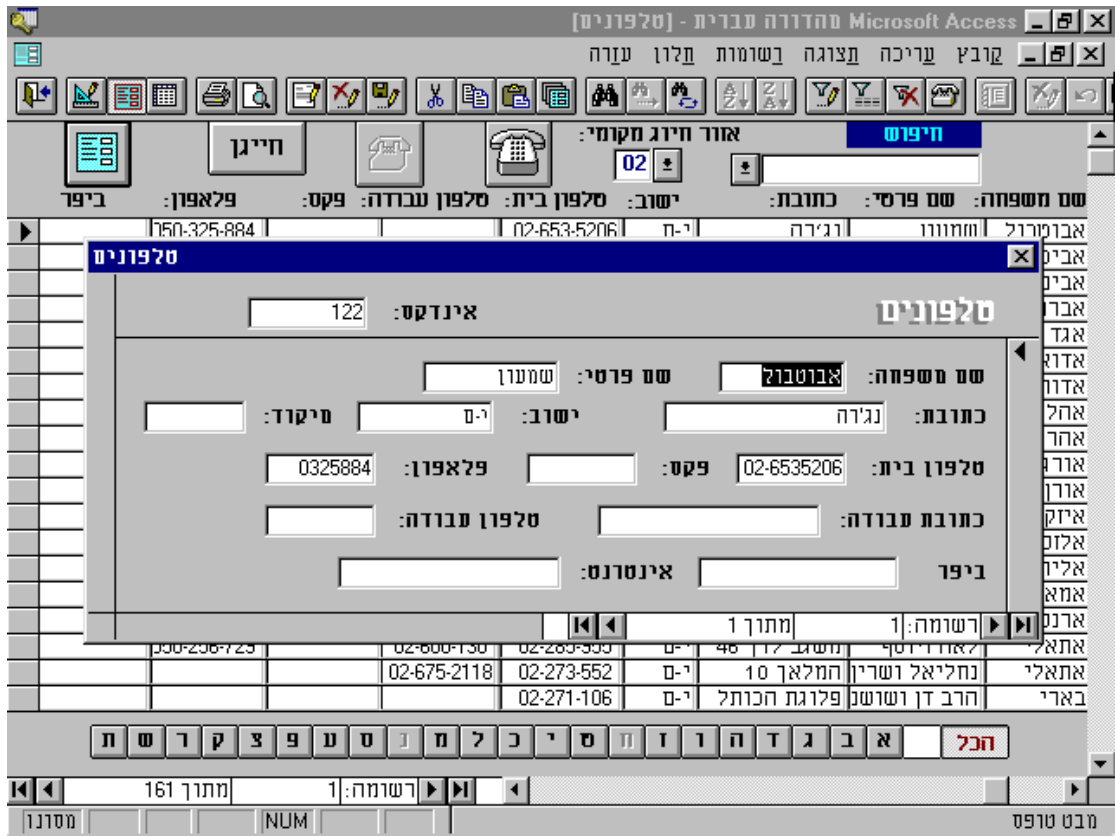
```
End Sub
```

בכדי לפתוח את הטופס בתיבת דיאלוג, יש לשנות את השורה הבאה כך:

```
DoCmd OpenForm DocName, , LinkCriteria, A_DIALOG
```

10. התוצאה בהקשה על הכפתור הינה זו

מדריך לאקס – מפתחים



בכדי להגיע להתאמה מירבית יש לשנות את תכונות הטופס «אלפון מבט על»

- _____ בוררי רשומות לא
- _____ לחצני ניווט לא
- _____ שנה גדל אוטומטית כן
- _____ מרכז אוטומטית כן
- _____ נפתח כן
- _____ בהתאם למוד לא
- _____ סגנון מסגרת דו-שיח

8.1. חיפוש ואיתור רשומות מתיבה משולבת

המטרה ליצור תיבה משולבת שהמשתמש יוכל לאתר בעזרתה כל רשומה בטופס.

בדוגמא אנו רואים תיבה משולבת שיש בה שלוש עמודות.

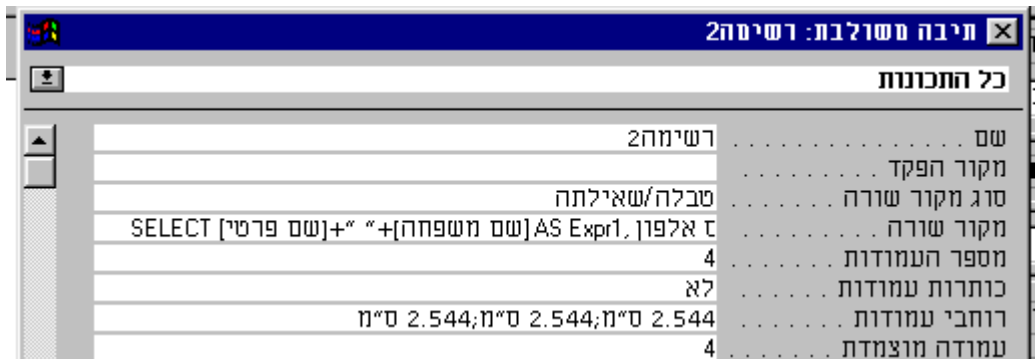


1. גם הטופס וגם התיבה צריכים להיות מבוססים על אותה טבלה, באופן ישיר או ע"י שאילתא. כמו כן, השדה שעל פיו רוצים לחפש את הנתון, צריך להיות כשדה בטופס.



2. תכונות התיבה המשולבת.

השדה שבעזרתו מבצעים את החיפוש חייב להיות השדה המוצמד, בתיבה המשולבת. בדוגמא אנו רואים שהשדה המוצמד הוא השדה הרביעי, אעפ"י שהוא איננו מופיע כלל בתיבה המשולבת.



3. אנו רואים שהתיבה המשולבת מורכבת מארבעה עמודות, שם, כתובת, ישוב, ואינדקס אלפון. אינדקס אלפון, כתובת, ישוב, ואינדקס אלפון. איתור הרשומות יהיה לפי שדה אינדקס אלפון.

שם: [שם משפחה]+ " " + [שם פרטי]	כתובת	ישוב	אינדקס אלפון	תורם הוראת קבע	שדה: מיון: הצג: קריטריונים:
סדר עולה					
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
True					

4. נותר לנו שלב הקוד. יש לקשור את אירוע בעת תקתוק של התיבה המשולבת לפונקציה שתבצע את העבודה.



5. הפונקציה תיכתב באחד המודולים הראשיים של המסד, בין היתר בכדי שיהיה אפשר להשתמש בה בכל טופס.

```
Function findReqRecord(theField as Variant)
Dim l_findstring As String, f_rs As Recordset
Set f_rs = Screen.ActiveForm.RecordSetClone
l_findstring = theField & " " & Screen.ActiveControl
f_rs.FindFirst l_findstring
Screen.ActiveForm.BookMark = f_rs.BookMark
f_rs.Close
End Function
```

הסבר:

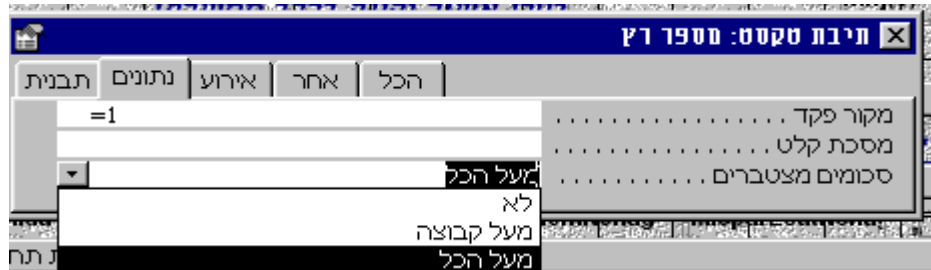
1. יוצרים שני משתנים. אחד מסוג טקסט, והשני מסוג סדרת-רשומות.
2. יוצרים סדרה של רשומות בזיכרון, לפי מקור הרשומה של הטופס הפעיל. ע"י השיטה «שכפל סדרת רשומות» RecordSetClone ז"א שעכשיו יש לי שכפול של כל רשומות שמופיעות בטופס הפעיל, ואפשר לבצע עליהן פעולות שונות.
3. משים את הערך אותו אני רוצה לחפש. אני בוחר לחפש בשדה «אלפון» את הערך שיישאב מהפקד תיבה משולבת.
4. מפעיל את השיטה «חפש ראשון» FindFirst על סט הרשומות עם התנאי אותו אני בוחר.
5. מכיוון שזהו שיכפול של הרשומות בטופס, מציאת הרשומה בקוד, עדיין לא תגרום להזזה של הרשומות בטופס לרשומה שנמצאה. לכן אני צריך להפעיל את השיטה סימון ספר BookMark, שמתאמת בין אוסף הרשומות של הטופס לאוסף סדרת הרשומות שבשכפול.
6. יש לסגור את האוסף המשוכפל, אחרת הוא יישאר בזיכרון.

הדפסת רשומה נוכחית בטופס
בתפריטים: קובץ <= הדפסה <= רשומה נוכחית
בקוד: לכתוב את הקוד הבא DoCmd.PrintOut acSelection

2. דוחות מתקדם

1.2 ספרור אוטומטי

בכדי ליצור פקד שיספרר אוטומטית את הרשומות יש לבחור פקד "תיבת טקסט" לתת לו את הערך 1 ב"מקור פקד" ובתכונת "סכומים מצטברים" יש לבחור, "מעל קבוצה" אם רוצים ספרור מחדש לכל קבוצה או "מעל הכל" אם רוצים רק ספרור אחד לכל הדוח.



2.2 הצגת תמונה בדוח

באם רוצים להציג תמונה בדוח כשהתמונה נשלפת מטבלה מסויימת יש ליצור אובייקט תמונה מאוגד ובמקור הפקד להשים לתוכו את הערך הבא:



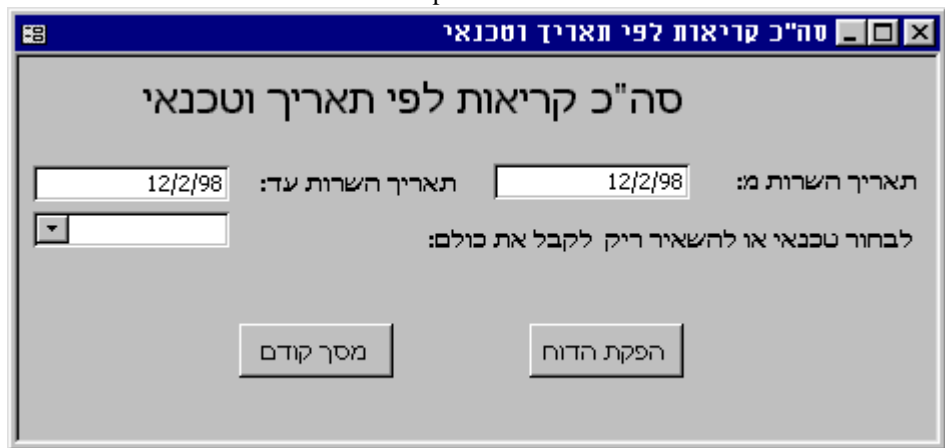
בהרצה נקבל את התוצאה הבאה



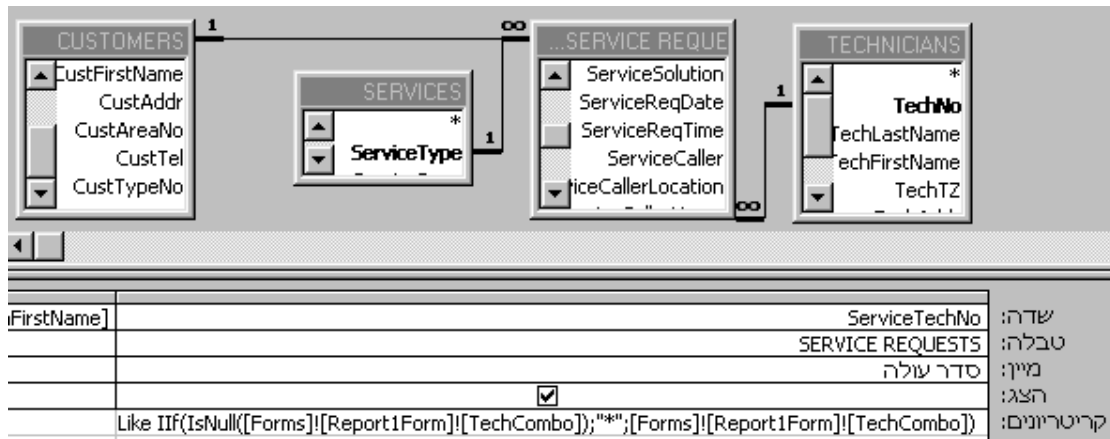
3.2 דוח סלקטיבי אי

ברצוננו ליצור דוח סלקטיבי שידפיס לפי אנשים שנבחרו בטופס פרמטר ואם לא נבחרו אנשים אנו רוצים שכולם יודפסו.

ראשית ניצור טופס פרמטר ששמו Report1Form כאשר שדה הפרמטר יהיה TechCombo



בשלב הבא ניצור דוח שיהיה מבוסס על שאילתא שהקריטריון שלה יבוסס על השדה בטופס.



התנאי בקריטריון אומר שאם שדה הפרמטר בטופס הינו ריק להציג את כל הרשומות, אחרת להציג רק את הרשומה שנבחרה בשדה הפרמטר בטופס.

בשלב הבא יש ליצור לחצן פקודה בטופס שמאחוריו יהיה הקוד שמריץ את הדוח.

```
DoCmd.OpenReport "report1", acViewPreview
```

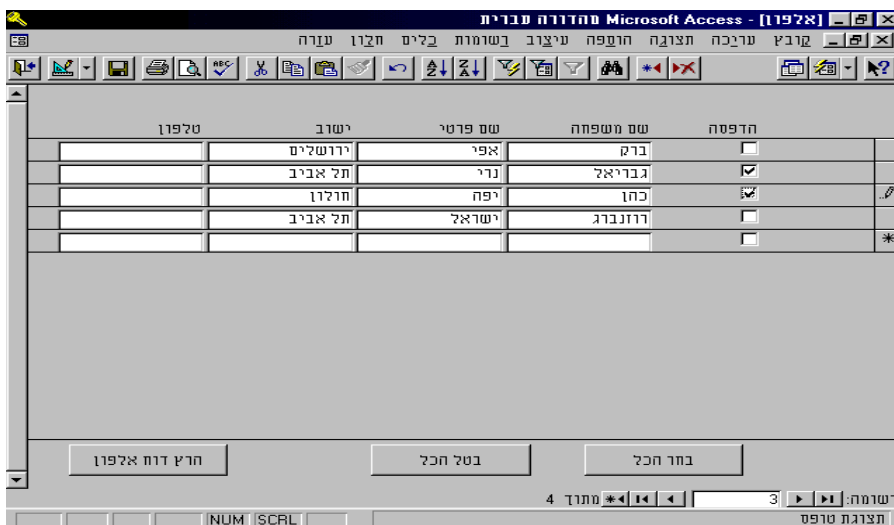
4.2. דוח סלקטיבי ב'

אנו רוצים להדפיס דוח סלקטיבי, באופן שרשימת האנשים שידפיס תהיה מבוססת על רשימת אנשים שאנו נבחר.

לצורך כך יש להגדיר בטבלה שדה נוסף «הדפסה» שהגדרתו תהיה כן/לא. יש לייצר טופס שיתן לנו רשימת אנשים עם השדה הזה, וכמובן שבפתיחה שלו יש להריץ שאילתא שתאפס את כל האנשים. ולבסוף אותו טופס יריץ לנו את הדוח הנדרש שכמובן צריך להיות מבוסס על שאילתא שבקריטריונים שלה תציג רק רשומות ששדה «הדפסה» ערכו אמת.



- 1 בטבלת «אלפון» יצרנו שדה חדש ושמו «הדפסה»



- 2 אנו יוצרים טופס המבוסס על הטבלה «אלפון» המשתמש צריך לסמן בפקד «הדפסה» את הרשומות אותן הוא רוצה להדפיס.

3 קוד מאחורי כפתורי הטופס

```
DoCmd.SetWarnings False
DoCmd.RunSQL "UPDATE אלפון.הדפסה SET אלפון = True;"
DoCmd.SetWarnings True
DoCmd.DoMenuItem 0, 5, 5, 0, acMenuVer70
```

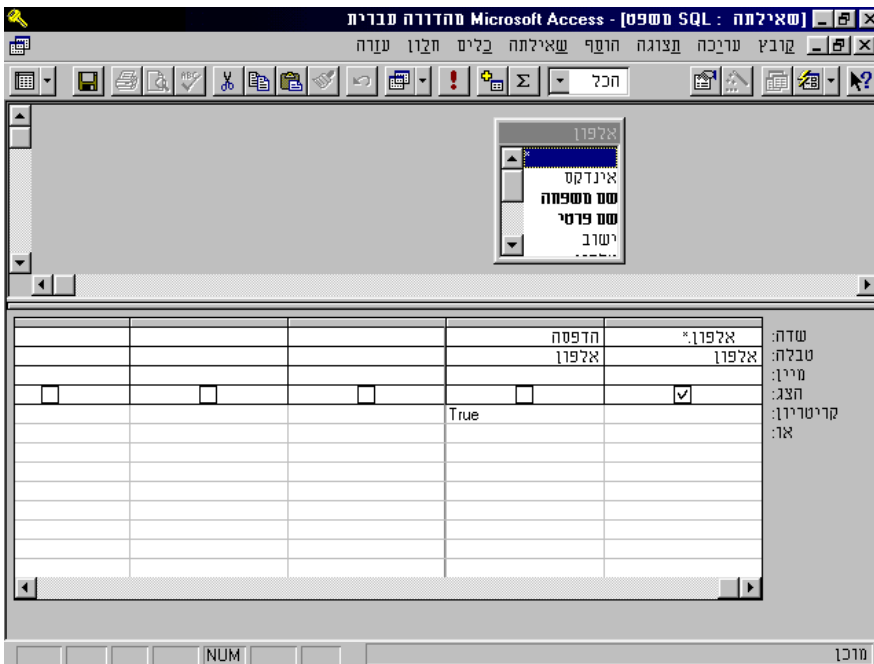
א כפתור «בחר הכל» שאילתא לבחור את כל הרשומות

```
DoCmd.SetWarnings False
DoCmd.RunSQL "UPDATE אלפון.הדפסה SET אלפון = FALSE;"
DoCmd.SetWarnings True
DoCmd.DoMenuItem 0, 5, 5, 0, acMenuVer70
```

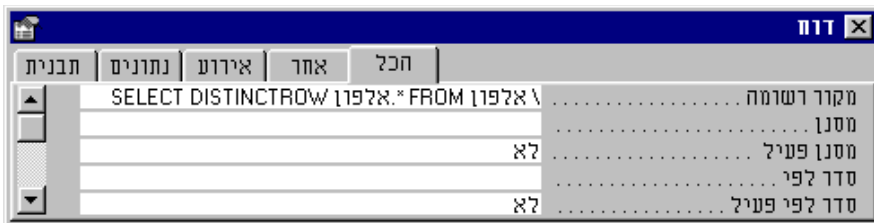
ב כפתור «בטל הכל» שאילתא לבטל בחירת כל הרשומות

```
DoCmd.DoMenuItem 0, 5, 5, 0, acMenuVer70, רשומות, '
רענון
DoCmd.OpenReport "acPreview", "אלפון",
```

ג כפתור «הרץ דוח אלפון» רענון נתוני הטופס, והרצת הדוח

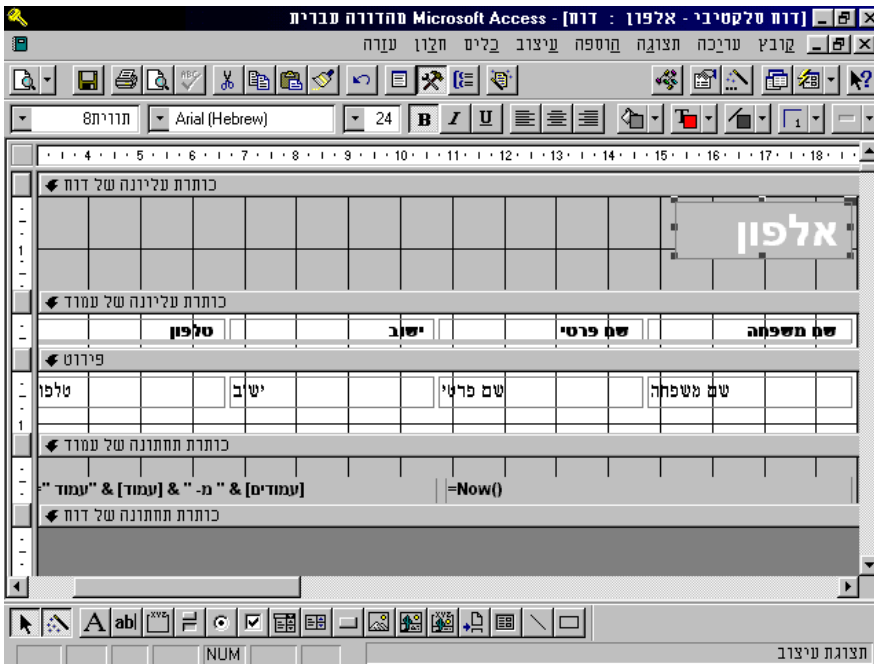


4 יש ליצור דוח שיציג רק רשומות ששדה «הדפסה» שלהם שווה אמת. לצורך כך יש ליצור את השאילתא הבאה, כאשר הקריטריון הינו שדה «הדפסה» שווה אמת.

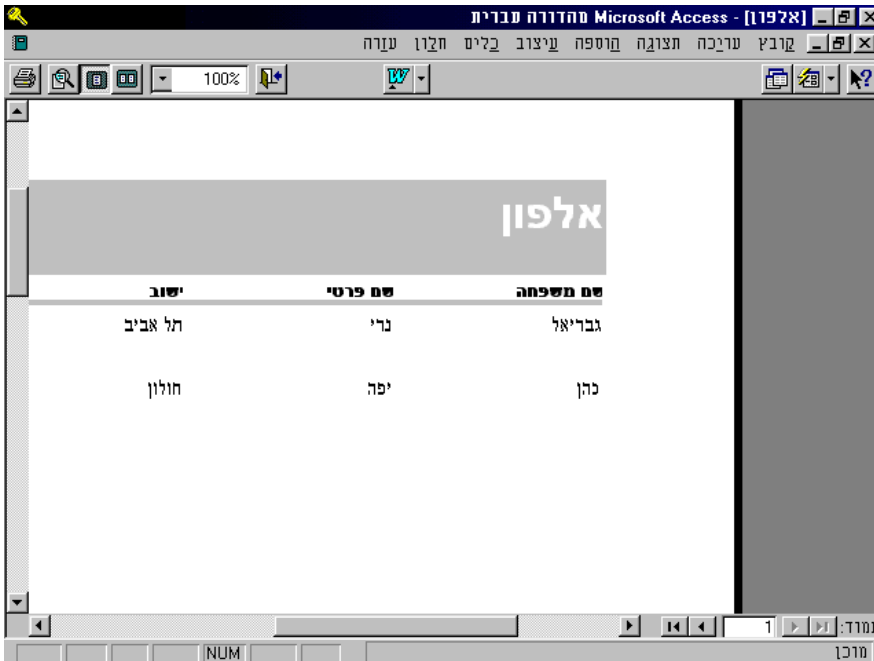


5 תכונת «מקור הרשומה» של הדוח הינו השאילתא שיצרנו.

6 למעט השינוי הזה, שאר הדוח הינו דוח רגיל.



7 הרצת הדוח הנ"ל לאחר הקשה על כפתור «הרצת דוח אלפון».



3. מודולים (קוד ויזואל בייסיק VBA) - בסיסי

1.3. הקדמה

- ראינו שישנן פקודות שאפשר לבצע על ידי השימוש במאקרויס, אך לרוב הן מוגבלות, כך שיש צורך לעתים להשתמש בקידוד. שימושים נפוצים בקידוד:
- יצירת אובייקטי אקסס ושינוי תכונותיהם באוטומציה.
 - הרצת פקודות מקרו באופן גמיש ובשימוש במשתנים, מתוך קידוד.
 - גישה ומניפולציה על נתוני המסד, בעזרת אובייקטי הגישה לנתונים.
 - משפטי קידוד לצורך חישובים או פעולות בכל המובן הרחב של הקידוד.
 - גישה לקוד חיצוני בקבצי DDL והפעלת תוכנות חיצוניות ב DDE.

חלון קידוד

The screenshot shows the VBA Editor with the following code:

```

Dim l_findstring As String, f_rs As Recordset
Set f_rs = Screen.ActiveForm.RecordsetClone
l_findstring = "=" & Screen.ActiveControl
f_rs.FindFirst l_findstring
Screen.ActiveForm.Bookmark = f_rs.Bookmark
f_rs.Close
End Function

Function FirstotOfNextMonth()
    FirstotOfNextMonth = DateSerial(Year(Now), Mon
End Function

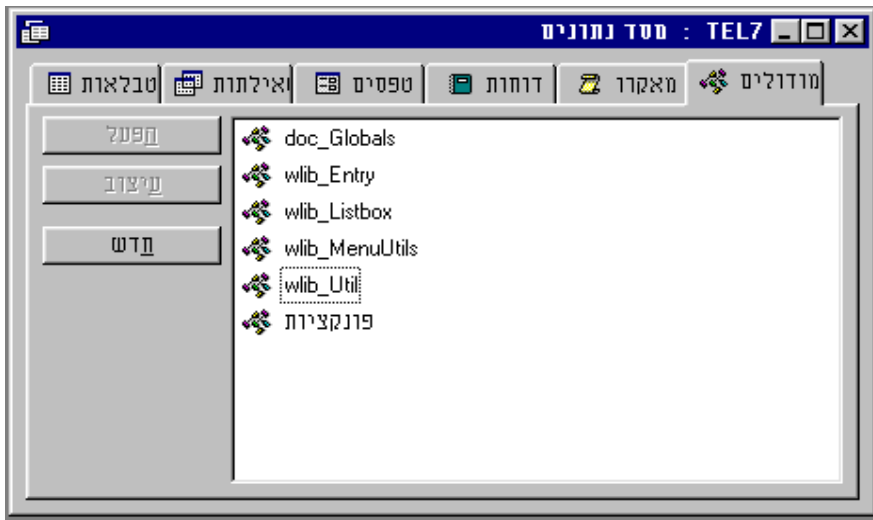
Function hipoch(inp As String) As String
    Dim b As String: b = ""
    If inp < "א" Then
        inp = Mid$(inp, InStr(1, inp, b) + 1) & "" & Left$(inp,
    End If
    hipoch = inp
End Function
    
```

Annotations and their corresponding parts in the image:

- ביצוע קומפילציה למודולים טעונים**: Points to the 'Compile' button in the toolbar.
- שם פונקציה נוכחית**: Points to the 'Function' keyword in the code.
- עיון באובייקטי אקסס**: Points to the 'View Object' button in the toolbar.
- שם תחום**: Points to the 'Object' dropdown menu showing '[General]'.
- פונקציה חדשה**: Points to the 'New Function' button in the toolbar.
- מודול חדש**: Points to the 'New Module' button in the toolbar.
- חלון ניפוי שגיאות**: Points to the 'Error List' window showing an error for variable 'b'.
- שמות המשתנים שבשגרה/פונקציה נוכחית**: Points to the 'Context' column in the Error List window.
- חלון ניפוי שגיאות מכאן אפשר להריץ פונקציות ולקבל תוצאות**: Points to the 'Run' button in the toolbar.

אפי ברק © כל הזכויות שמורות - לשימוש אישי ולא מסחרי

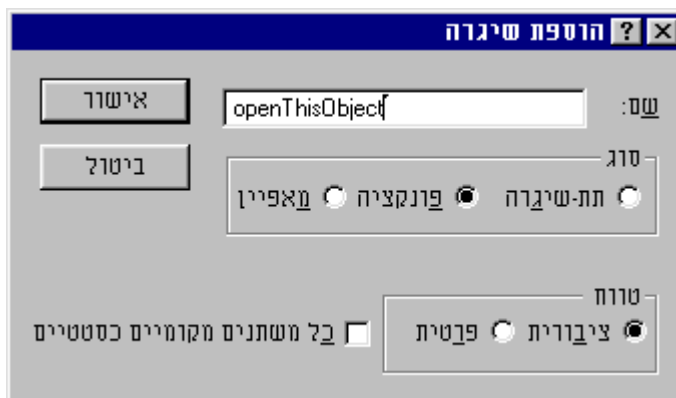
חלון ניפוי שגיאות יישאר פתוח בכל מקום במסד הנתונים



מודולים גלובליים

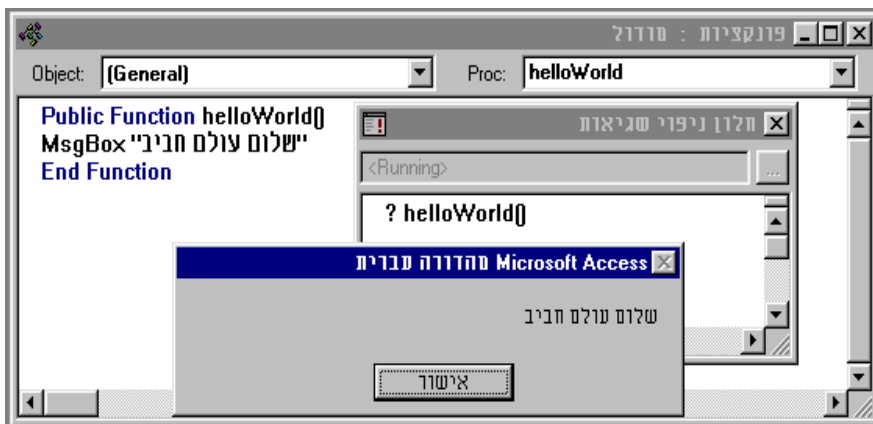
אפשר ליצור מספר רב של מודולים בכל מסד. כל מודול יכול להכיל מספר רב של משתנים/קבועים/פונקציות/ושגרות שתחומן הוא מסד הנתונים כולו.

הפניה אל פונקציה/שגרה גלובלית תעשה בק בשמה, בלא שם המודול בו היא נמצאת.



פונקציה חדשה

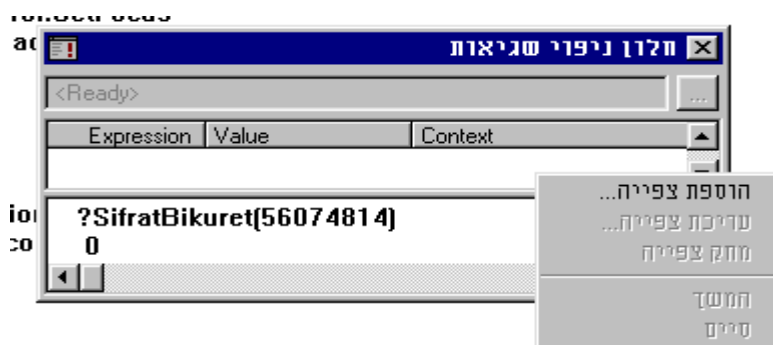
להוספת פונקציה חדשה יש לבחור בתפריט {הוסף} {שגרה חדשה...} יש לבחור אם מדובר בשגרה או בפונקציה, ויש להקיש את שמה.



דוגמא לשגרה

השגרה הנוכחית יש בה שורה אחת המייצרת תיבה צפה ובה ההודעה "שלום עולם".

בכדי להריצה באופן מידי, יש להקיש את שם השגרה בחלון {ניפוי שגיאות} ולהקיש ENTER

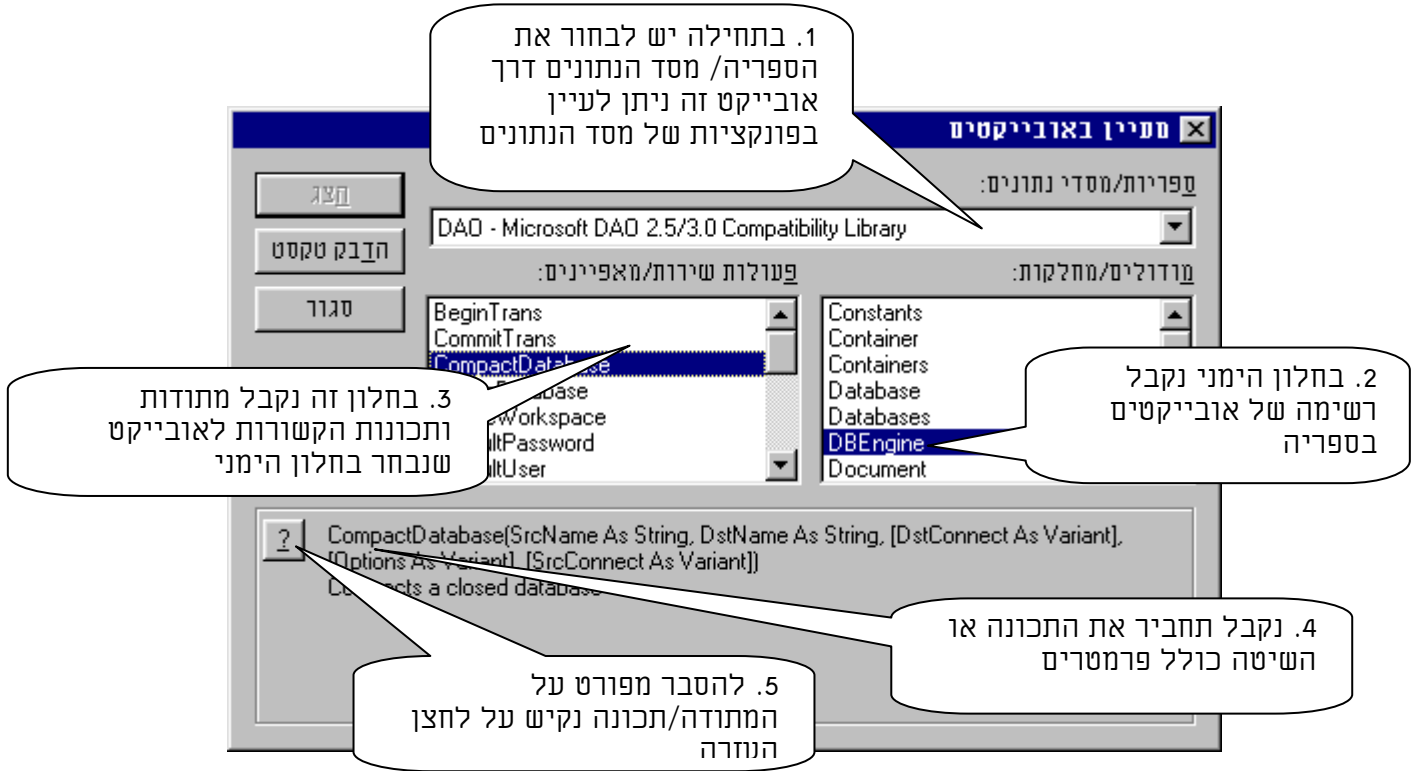


בהרצת פונקציה מחלון {ניפוי שגיאות} יש להוסיף סימן שאלה לפני פונקציה. פונקציה נוכחית בודקת ספרת ביקורת של מס' תעודת זהות ומחזירה במקרה שהיא תקינה 1- ובמקרה שהיא שלילית - 0

2.3. איתור פקודות ופונקציות ותחבירם

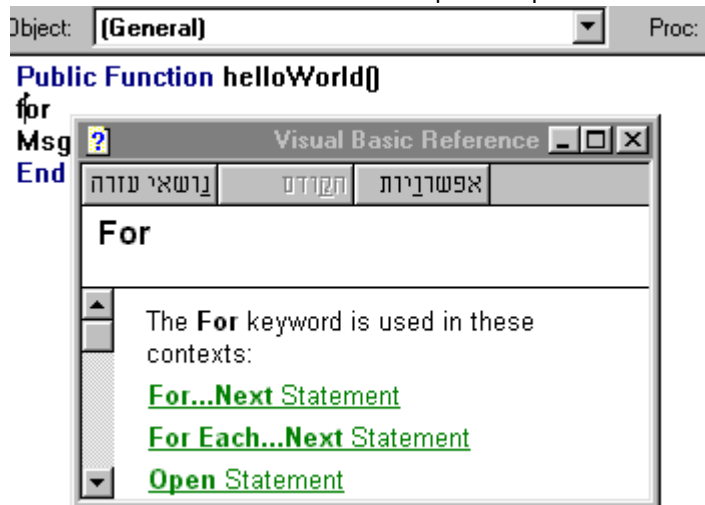
סורק האובייקטים

מתוך חלון זה אפשר לעיין בכל הפונקציות והתכונות שעומדים לרשות המפתח בכל המרחב של בסיס הנתונים, המסד, אובייקטי גישה, ויזואל בייסיק ועוד.



ברגע זה אנו מעיינים במתודות הקשורות למנוע בסיס הנתונים, במתודה "דחיסת מסד נתונים" בחלק התחתון אנו רואים את התחביר של המתודה, ובמקרה שנרצה הרחבה ופירוט נקיש על כפתור סימן השאלה, והעזרה תפתח בערך הזה.

במקרה שאנו רוצים עזרה על מילה שמורה של ויזואל בייסיק אנו יכולים לכתוב את המילה ולהקיש F1 והעזרה תפתח בערך המבוקש



3.3. הצהרות (declaration) והגדרות (definition)

לכל מודול יש אזור של הצהרות כלליות בו מצהירים על דרכי הפעולה של המודול, פונקציות המודול ועל טיפוסים משתנים חדשים, וכן מגדירים את המשתנים הגלובליים של המודול.
(באזור הזה אין אפשרות לכתוב קוד בשום דרך שהיא!!)



הבהרה - הבדל בין הצהרה להגדרה

בהצהרה יוצרים אב טיפוס חדש של משתנה או דרך פעולה מסוימת של התוכנה, בעוד שבהגדרה יוצרים ומגדירים בפועל את המשתנה עצמו בזיכרון, לצורך עבודה עימו. ההצהרה הינה תאורטית, ההגדרה הינה מעשית. אם הצהרנו על טיפוס חדש בגודל מוגבל, אנו יודעים שישנה אפשרות תאורטית ליצור משתנה כזה, אבל רק ברגע שנגדיר אותו, הוא יתפוס את הזכרון, ויהיה אפשר להשתמש בו. (לעיתים בעברית קוראים בשם הכרזה להגדרה)

הצהרות כלליות

Option Compare {<Database> | <Binary> | <Text> }

א. סוג השוואה בין נתונים באופרטורים השוואתיים.

בינארי: מבדיל בין אותיות גדולות לקטנות. טקסט: אינו מבדיל בין אותיות קטנות לגדולות. מסד נתונים: לפי סדר של הנתונים בעת שבסיס הנתונים יוצר.

Option Base {0 | 1}

ב. קביעת התחום התחתון של המערכים במודול

ברירת המחזל של האיבר התחתון במערך הוא איבר 0, יש אפשרות לכפות במודול מסוים שהאיבר התחתון יתחיל ב1, לשם הסדר הטוב.

Option Explicit

ג. כפיית הגדרת משתנים



ויזואל בייסיק בניגוד לפסקל C++ מאפשר ליצור משתנים באופן מרומז כאשר הוא נתקל בהם בפעם הראשונה, ולא מחייב להגדיר משתנים במפורש. בכדי לחייב הגדרת משתנים במפורש יש להצהיר את ההצהרה לעיל, במידה ולא מכריזים על משתנה נקבל את הודעת השגיאה הבאה:

כדאי להשתמש בכפיית הכרזת נתונים, לשם הסדר הטוב של התוכנה, לאפשר תחזוקה קלה, וכן בכדי שלא יוצרו משתנים לא קרואים, בגלל שגיאה דפוס אקראית, כך שהם ישבשו את כל מהלך ההרצה.

Option private Module

ד. כופה על מודול כללי להיות פרטי.

4.3 קבועים ומשתנים

המשתנים והקבועים הינם אבני היסוד שמהם מורכבת התוכנה.

קבועים - Constants

הינו כעין מיכל סטטי לנתונים, שרק ערך אחד יכול להיות מושם בתוכו, בתחילת ההרצה ואינו יכול להשתנות יותר במהלך הריצה.
 בהגדרת קבוע התוכנה מקצה כתובת בזכרון לאותו קבוע, בגודל הטיפוס של הערך המושם לקבוע, ושומרת את ערכו מבלי אפשרות לשינוי.
 אנו נשתמש בקבועים כאשר נרצה להתייחס במהלך התוכנית פעמים רבות לערך ספציפי שיכול מידי פעם להשתנות, כמו "שער הדולר". באם לא נשתמש בקבוע נצטרך לעבור בכל התוכנה ולשנות את כל ההתייחסויות לאותו ערך. כמו כן בכדי להתייחס לטווח ערכים מספריים שערכם המספרי לא אומר לנו דבר, אבל שמם כן, כמו קבועי הצבע. שימוש נוסף הוא לפשט שימוש בערכים מורכבים כמו PI בכדי להורות לתוכנה להגדיר קבוע חדש יש להשתמש במילה השמורה Const לפני שם הקבוע ולקבוע את ערכו באמצעות פעולת השמה.

הגדרת קבועים

קבוע הינו כעין משתנה שערכו סטטי, ואינו יכול להשתנות.
 ישנם קבועי מערכת שונים, וישנם קבועי משתמש, שמוגדרים בכניסה למודול.

Const PI = 3.14159265358979

קבועי מערכת

Vbcr = 13 תו אסקיי

Vblf = 10 תו אסקיי

VbRed - צבע אדום

משתנים - Variables

משתנה הינו כעין מיכל דינמי לנתונים, באופן שבמהלך הריצה אפשר לשנות את הנתונים שבו. במשתנה הערך הנתון בתוכם יכול להשתנות ע"י התוכנית או המשתמש תוך כדי הרצה. ואפשרות זו היא הנותנת לנו לעשות מניפולציות על משתנים והיא הנותנת לתוכנה את כוחה. בהגדרת משתנה, התוכנה מקצה כתובת בזכרון לאותו משתנה בגודל הטיפוס שהוגדר, ומאפשרת בכל עת לשנות את הערך הקיים, לערך חדש.

גם בקבועים וגם במשתנים אפשר לשמור בו ערכים מסוגים שונים בהתאם לטיפוס המשתנה, החל מנתון מספרי דרך נתון טקסטואלי וכלה בנתון תאריכי ועד למצביעים על אובייקטים.

בכדי להורות לתוכנה להגדיר משתנה, יש להשתמש במילה השמורה Dim או המילה השמורה Public לפני שם המשתנה ולקבוע את סוג הטיפוס שלו באמצעות המילה השמורה As. קביעת טיפוס המשתנה הינה אופציונלית בויזואל בייסיק.

הגדרת משתנים מקומיים Dim

משתנים שיוכרו רק בתוך המודול בלבד או הפונקציה.

Dim B [As String], C [As String]

Dim string4 As String * 4

Dim WormBorder Variant מסוג

הגדרת משתנים גלובליים Public

משתנים שיוכרו בכל הפרוייקט כולו, בתנאי שהם נרשמו במודול כללי.

Public FirstName As String

Public NumberOfChanges As Long

Public CreditCardNumber As String*80

Public myCard As String*65000

Public resumePrintNow As Integer

Public misparHafkadaLabank As Double

Public SchumLtashloom As Currency

ישנם פונקציות הבודקות את סוג הערך במשתנה, האם מספרי, תאריכי, ריק, כלום, והם מתחילות במילה Is, והם מופיעות כאן. המשתנה צריך להיות מושם בהם כפרמטר.
IsNumeric, IsDate, IsArray, IsEmpty, isFull, IsNull, IsObject

5.3. טיפוסים משתנים וגודלים

<p>טקסט רגיל String</p>		<p>טווח: 0 עד 2 מיליארד תווים. משמש לאחסנה של מחרוזות.</p>	<p>גודל: 10 בתים + בית לכל תו.</p>
<p>טקסט מוגבל String * [value]</p>		<p>טווח: 0 עד 65,400 תווים. משמש לאחסנה של מחרוזות.</p>	<p>גודל: בית לכל תו.</p>
<p>מספר</p> <p>כל נתון שנדרש לעשות עליו פעולה מתמטית או מיון מתמטי. ישנם ששה סוגי מספרים לפי גדלים שונים. במספרים מבדילים בין משתנים שלמים שיכולים לקבל מספרים שלמים בלבד, לבין משתני נקודה צפה שיכולים לקבל מספר עם שברים עשרוניים.</p>			
סוג	שם	גודל	מכיל
בוליאני	Boolean	1 בית	דוגמא True 0 TRUE/FALSE True =1 , False =0
שלם קצר	Byte	1 בית	0 עד 255
שלם בינוני	Integer	2 בתים	32,768- עד 32,768
שלם ארוך	Long	4 בתים	2,147,483,648- עד 2,147,483,648
נקודה צפה יחיד	Single	4 בתים	-3.4×10^{38} עד 3.4×10^{38}
נקודה צפה כפול	Double	8 בתים	-1.797×10^{308} עד 1.797×10^{308}
<p>תאריך/שעה Date</p>		<p>סוג שדה שתומך בתאריך ושעה, ומבצע בדיקת אימות לנתונים תאריכיים באופן אוטומטי. נתוני שעה מימין לנקודה, נתוני תאריך משמאל לנקודה. יש פונקציות תאריכיות רבות שעובדות עם סוג משתנה זה. גודלו: 8 בתים.</p>	
<p>מטבע Currency</p>		<p>סוג מספרי עם נקודה צפה שמוסיף לסכום סימן מטבע. ומחשב חישובי מטבע מדויקים. גודלו: 8 בתים.</p>	
<p>משתנה Variant</p>		<p>משתנה רב תכליתי, שמשנה את עצמו כמו זיקית לפי הערך שמושם בו. כמו כן נעשים בו שימושים רבים שיחודיים לתכנותיו, כמו האפשרות לקביעת פרמטרים אופציונליים ודינמיים בפונקציה או להשמה של רשימת ערכים למערך בעזרת הפונקציה Array. אחסון מספר: גודלו 16 בתים. אחסון טקסט: גודלו 22 בתים + גודל המחרוזת.</p>	
<p>אובייקט Object</p>		<p>משתנה שמצביע על אובייקט שהוגדר בתוכנה כגון טופס, פקד, אוסף רשומות ועוד. גודלו 4 בתים.</p>	

השמה - Casting

במקרה שישנה השמה ממשתנה שתומך רק בחלק מהיכולות של המשתנה השני, יתבצע שינוי אוטומטי של הערך שיושם במשתנה השני לפי התבנית שלו. לדוגמא אם מספר עם שברים

מדריך לאקסס – מפתחים

יושם למשתנה נקודה צפה, יתבצע עיגול של המספר לשלם והוא יושם למשתנה. אם תתבצע השמה למשתנה שאינו יכול לתמוך בסוג הערך, תופיע הודעת שגיאה.

שמות האובייקטים - תחביר ואפיון

תחביר השמות



שמות קבועים, משתנים, פקדים, שגרות ופונקציות אינם יכולים להתחיל בספרות ומספר תוים כמו סימן שאלה, סימן קריאה, סימנים מתמטיים, מקפים. רוב הסימנים הנזכרים אינם יכולים להיות מוכלים בשמותם. אסור שיהיה ריווח בין שני מילים המגדירים את האובייקט, וכמובן שאין לקרוא לאובייקט ב"מילה שמורה", בפונקציה קיימת, או בשם של אובייקט אחר שבתחום, כמו אובייקט אב.

נהוג שכל מילה חדשה במשתנה מתחילה באות גדולה. למשל JustDoIt

נוהגים לתת קידומת לכל משתנה ולכל פקד לפי סוגו, על פי השיטה ההונגרית, שהיא מוסכמה בינלאומית, בכדי לדעת משמו את סוג המשתנים שהוא יכול להכיל. (ראה נספח)

שמות משמעותיים

העיקר החשוב הוא שיש לתת שם משמעותי, לכל אובייקט שקיים בפרוייקט, זהו המפתח לפרוייקט ברור שקל לתכנת אותו. שם משמעותי הנו שם, שמתכנת זר יכול להבין ממנו, את תפקיד האובייקט גם בלי פירוש רש"י.

יש להתייחס לכל אחד מהאובייקטים כמו אל ילד קטן שהולך לגדול ושאהר כך אנחנו אמורים "לקרוא" לו הרבה. אין לתת שמות סתמיים כמו רצף תוים לא ברור או מספרים שלא אומרים דבר או להשאיר ברירות מחדל של התוכנה כמו "Form1", "Project1", ..

אין לתת "שמות חיבה וקיצור" כמו Num ל number וכדומה אלא לנקוב כל אובייקט בשמו המפורש. כמו בחברים יתכן ששם מקוצר יאמר לנו רבות, אך לאדם אחר יהיה סתמי ביותר.

אפשר לכתוב שמות באנגלית רגילה או עברית "מאונגלזת", אך אין לערבב אחד עם השני.

מישהו צריך לפתח את התוכנה ולתחזק אותה בעוד זמן כזה ואחר, ולכן היא צריכה להיות מובנת, בהירה וידידותית למתכנת, לא פחות ממה שהיא צריכה להיות ידידותית למשתמש. תוכנה צריכה להיות כמו חיבור ספרותי. תוכנה שלא ברורה היא תוכנה שנועדה לגניזה, ובאיזה שהוא שלב היא תרד לטמיון.

כאשר נותנים שם משמעותי, קל יותר לאחר מכן להפטר ממשתנים ש"סיימו" תפקידם, ואינם נצרכים, וכן להמנע ממשתנים כפולים, שאמורים לעשות את אותה עבודה.

הרכב השם

כל שם משתנה, פונקציה ושגרה רצוי מאוד מאוד שיכיל בתוכו לפחות שלוש מילים, בכדי שיובן משמו, מה תפקידו בכוח ואיזה אינטרסים של התוכנה הוא משרת. אין לפחד מגודלו של השם. יש תיבת רשימה שנוכל לבחור מתוכה את שמות האובייקטים של הפרוייקט מבלי שנצטרך להקליד כל פעם מחדש את שמו. כמובן שאין להגזים בשם אובייקט ולכתוב שם באורך הגלות, דבר שגם יכול להכביד על ההבנה של התוכנית. שם של פונקציה יתחיל בפועל, שיתאר את פעולתו. בעוד ששם משתנה יתחיל באות קטנה, שם פונקציה יתחיל באות גדולה.



בכדי לקבל את רשימת השמות שהוגדרו בתוכנה יש להקיש על מקשי CTRL + Space

בכדי לקבל רשימת תכונות ומתודות של אובייקט יש להקיש את שמו ואחרי שמו תו נקודה

מערך רגיל

```
Dim myArray (35) As String (0 -34)
Dim myArray (100 to 200) As String (100 - 200)
```

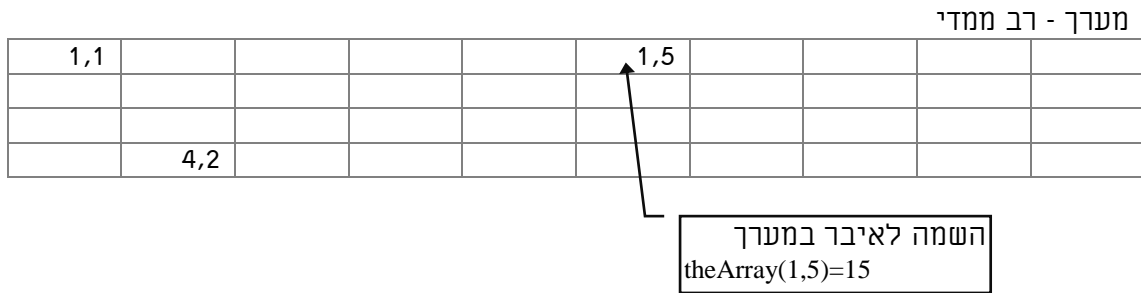
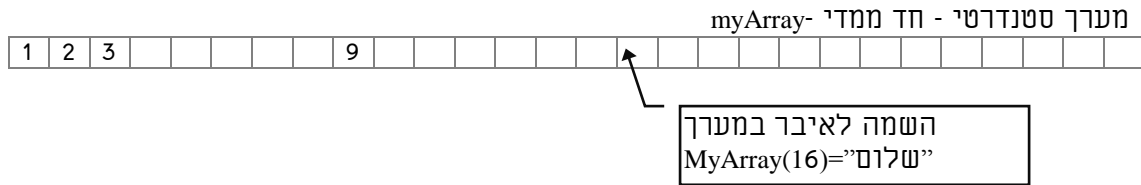
מערך רב ממדי

```
Dim MultyArray (10,10) As long
Dim MultyArray (1 to 10,1 to 10) As String
```

הגדרת מערך סטטי Array

מעריך הינו משתנה בעל איברים רבים, כשהגישה לאיבר מסוים שלו נעשית ע"י אינדקס.
ישנם סוגים שונים של מערכים.

השימוש במערכים הינו כאשר צריכים ליצור מספר רב של משתנים, מסיבה כל שהיא. או שיש להם מכנה משותף, כמו ימים בשבוע, או קריאת רשומות לזיכרון וכדו'.



כמובן שמעריך יכול להיות בנוי מכל סוג משתנה החל ממספרי וכלה בטקסט.

הגדרת מעריך דינמי

במקרה שאיננו יודעים לפני ההרצה מהו גודל המעריך שידרש, ואיננו רוצים לבזבז איברים בזכרון, יש אפשרות להכריז על מעריך בלי גודל ספציפי, ובזמן הריצה להגדיר את הגודל שלו כפי שנצטרך. כמו כן ישנה אפשרות להגדירו מחדש לאחר ההגדרה הראשונה במחיר של אתחול המעריך ופגיעה בנתונים הקיימים, וכן ישנה אפשרות להעלות את הגבול העליון של המעריך בלבד מבלי לפגוע בנתונים שכבר קיימים בו.

Dim myArray () as string	הגדרה של המעריך
Redim myArray(0,100)	קביעת גודל בזמן הרצה
Redim Preserve myArray(150)	הגדלת גבול עליון



אין אפשרות לבצע הגדרה מחדש של מעריך סטטי שגודלו כבר נקבע בהכרזה.

השמה מהירה

אפשר להשתמש בפונקציית Array להשמה למעריך דינמי מסוג ואריאנט.

```
Dim HWeekDays() As Variant
Dim HMonth() As Variant
HWeekDays = Array("שבת", "ששי", "חמישי", "רביעי", "שלישי", "שני", "ראשון")
HMonth = Array("תמוז", "סיון", "אייר", "ניסן", "אדר", "שבט", "כסלו", "חשוון", "אלול")
```

תרגיל: יש להגדיר מערך מסוג מחרוזת בעל 26 איברים ולהשים לתוכו את אותיות האלף בית העברי.

הצהרת טיפוס מוגדר משתמש - Types

טיפוס הינה הצהרה על אב טיפוס של משתנה. לתוכנה יש טיפוסים מובנים בתוכה, כפי שראינו. מלבד זאת אנחנו יכולים בעצמנו ליצור טיפוסים חדשים מהטיפוסים הפנימיים של התוכנה. לטיפוסים החדשים נקרא רשומות.

אין אפשרות להצהיר על טיפוסים חדשים במודול פרטי!

Type PratiIshim

Fname As String * 20

Lname As String * 15

Ktvet As String * 45

TeudatZeoot As long

tel As String * 12

fax As String *12

heahra As String

End Type

כאן יצרנו טיפוס חדש.

בעוד בTYPE אנחנו יוצרים טיפוס משתנה חדש במערכת, כמו טיפוסים בסיסיים שכבר קיימים בה כמו BYTE ו INTEGER הרי שב DIM אנחנו יוצרים משתנה ממשי בזכרון המחשב.

Public thePratim As PratiIshim

thePratim ששמו משתנה חדש

הצהרת טיפוס רשומה

רשומה הינה מבנה אחד שמורכב מהרבה משתנים שונים.

כשיש הרבה משתנים מסוגים שונים והם בעלי מכנה משותף רצוי לקבצם סביב שם אחד. כך שאפשר ליצור סוג טיפוס אב אחד, שמורכב מהרבה משתנים בקלות, וכל פעם שרוצים להגדיר משתנה כזה אין צורך להגדיר את כל המשתנים הכלולים בו. כך למשל אם רוצים לבנות רשומה של פרטים אישיים שכוללים שם, כתובת, טלפון, מספר תעודת זהות, ועוד אפשרם לקבצם ביחד למשתנה אחד.

הגדרת טיפוס רשומה

thePratim.Fname ="יוסף"

Fname למשתנה

גישה לאיבר ברשומה

6.3. גישה לאיברי האובייקט בשימוש בפקודת With

ישנה אפשרות לגשת באופן מקוצר לאיברים של אובייקט כולל מבנה רשומה באופן הבא:
תחביר:

With <שם אובייקט>

<ערך>=<תכונה>.

End With

דוגמא:

With thePratim

. Fname ="יוסף"

.Fax="02-5678989"

End With

הגדרת מערך רשומות

הגדרה על מערך של רשומות יש אפשרות ליצור מערך מסוג טיפוס רשומות שיצרנו.

Dim thePratim (100) as PratiIshim

השמה למשתנה כותרת החמישי במערך

thePratim (5). Fname = "יוסף"

תחום - טווח הכרה - Scope

תחום הוא האזור שבו המשתנה זמין, נגיש, ומוכר לפעילות. למשל משתנים מקומיים משני טפסים שרצים במקביל אינם זמינים אלו לאלו, אעפ"י שהם קיימים במקביל.

אם למשל הגדרתי משתנה באזור הראשי בטופס הוא יהיה נגיש רק למודולים בטופס, מעבר לטופס בטפסים אחרים או במודולים כללים גם אם הטופס פתוח כעת, המשתנה לו יהיה נגיש. הגדרת תחומי משתנים נעשית על פי האזור בו הוגדרו ועל פי הקידומת בהכרזתם האם הינם מקומיים (Dim) או גלובליים (Public):

משתנה שהוגדר בתוך פונקציה מסוימת, - יהיה מוכר רק בפונקציה. בפונקציה/שגרה אין אפשרות להגדיר משתנה Public אפילו בתוך פונקציה שהוגדרה Public, ואפילו היא במודול ראשי.

משתנה שהוגדר באזור ראשי של טופס - יהיה מוכר רק לשגרות/פונקציות הדוח או הטופס.

כ- DIM - יהיה מוכר לכל השגרות/פונקציות של אותו מודול בלבד, ולא יהיה נגיש לשגרות/פונקציות של מודולים אחרים.

כ- Public - יהיה מוכר לכל השגרות/פונקציות של הפרוייקט, כאשר יש להקדים לו את שם הטופס.

משתנה שהוגדר באזור ראשי באחד המודולים הראשיים.

כ- DIM - יהיה מוכר לכל השגרות/פונקציות של אותו מודול בלבד, ולא יהיה מוכר לשגרות/פונקציות של מודולים אחרים.

כ- Public - יהיה מוכר לכל השגרות/פונקציות של כל המודולים הקיימים.

פונקציה פרטית שהוכרזה כ Private

תוכר רק במודול שלה, ויהיה אפשר לקרוא לה רק מהמודול שלה

פונקציה גלובלית שהוכרזה כ Public

אם הוכרזה במודול פרטי של טופס היא תוכר בכל הפרוייקט רק בקידומת שם הטופס לדוגמא: פונקציה: HaktanTopes שנמצאת בטופס: closeMe צריכה להקראות כך: closeMe.HaktanTopes

אם הוכרזה במודול כללי, היא תוכר בשמה בכל הפרוייקט כולו.

הערה: בכל מקום שיש התייחסות לפונקציה הכוונה היא גם לשגרה.

אורך חיים

התחום בו המשתנה נוצר, קיים וחי בזיכרון המחשב, מעבר לו, המשתנה יסולק מהזיכרון, שיהיה פנוי לטובת משימות אחרות.

כך שמשנתה שהוכרז בפונקציה ייוצר בכניסה לפונקציה, ויעלם עם היציאה מהפונקציה. משנתה באזור הראשי בטופס, הוא יאותחל וייוצר בזיכרון עם פתיחת הטופס, ו"ימות" בעת שהטופס ייסגר.

משתנה במודול כללי יוכרז בעת הרצת התוכנית ויעלם בעת שהתוכנית תסגר.

ועם זאת, אפשר לשמר ערכו של משתנה מקומי ע"י הגדרתו כמשתנה סטטי באופן הבא:

Static Mone As Integer

ערכו הקודם יישמר בעת פתיחה חוזרת ונשנת של הפונקציה או הטופס. כמו כן אפשר להגדיר פונקציה שלמה כפונקציה סטטית, ואז כל הנתונים בה יישמרו, אעפ"י שהיא מקומית.

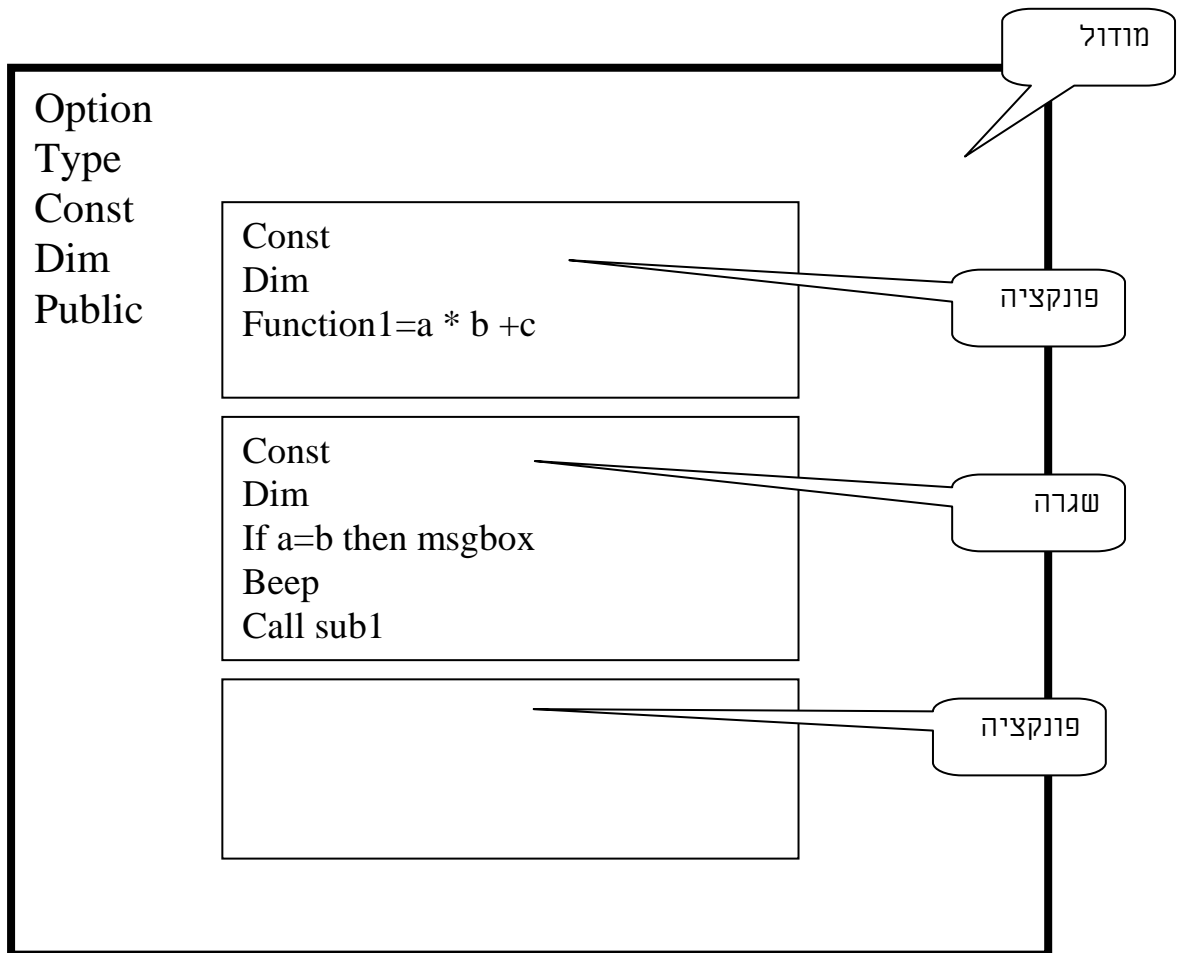
7.3 מודולים Modules

המודולים הינם מסגרות העל של הקידוד בויזואל בייסיק. כל הצהרה מכל סוג שהיא, הצהרה של פונקציה או שגרה, הגדרה של משתנה, קריאה לפונקציה או שגרה תכתב בתוך מודול כלשהו.

ישנם מודולים כלליים שמשרתים את כל הפרוייקט ולכן נקראים מודולים כלליים, וישנם מודולים פרטיים שמוצמדים לכל טופס/ דוח, והקידוד שבהם אמור לשרת רק את הטופס או הדוח הספציפי. מודול כללי יש ליצור כאובייקט נפרד, ולכל מודול כללי נוצר קובץ שבו הוא נשמר. לעומת זאת מודול פרטי נוצר באופן אוטומטי כאשר יוצרים טופס, נשמר בקובץ הטופס, והוא נמחק כאשר מוחקים את הטופס.

במקרה ויש כמה משתנים בעלי שם זהה במודולים שונים אפשר לפנות אליהם בתחביר הבא: שם המודול.שם משתנה. לעומת זאת הפניה אל הפונקציות שבמודולים תעשה ללא שם המודול. רק במקרה שיהיו שני פונקציות בעלות אותו שם בשני מודולים שונים, נקבל הודעת שגיאה בעת קריאה לפונקציה, ואז נצטרך לכתוב את שם המודול לפני הפונקציה למשל:

Call Module1.HaktenTopes1(Me)



הערה: אחרי פונקציה או שגרה, אין לכתוב דבר, מלבד משפטי הערה.

8.3. שגרות ופונקציות - SubRoutine & Function

השגרות והפונקציות הם הם מסגרות הקידוד. הפקודות, ומבני הבקרה, חייבים להיות אך ורק במסגרת תחום הפונקציה או השגרה.
מאידך השגרות והפונקציות חייבות להיכלל במודול, או מודול פרטי בטופס כזה ואחר או במודול כללי של כל הפרויקט.

שגרה הינה קטע קידוד שיש בו משפט או רצף משפטי פקודה, כאשר המטרה היא עצם ביצוע הקידוד. (השגרה לעיתים נקראת גם פרוצדורה או רוטינה)
אפשר להשתמש בשגרה בקריאה מתוך קידוד, ובשגרות המוצמדות לאירועים השונים בטפסים ודוחות.

מבנה בסיסי

[public | private] Sub <שם שגרה> ([פרמטרים])

<הליך>

End Sub

קריאה לשגרה: פרמטרים <שם שגרה> [CALL]

דוגמא: MsgBox "היום יום ראשון"

פונקציה הינה קטע קידוד שמתבצע במטרה להחזיר ערך מסוים.
אפשר להשתמש בפונקציה בהשמה למשתנה, בביטוי, ובתנאי במשפטי בקרה באזור קידוד, ובאירועי טופס ודוח בטפסים ודוחות, ובתנאי קריטריונים בשאילתא.

מבנה בסיסי

[Static | Public | Private] Function <שם פונקציה> ([פרמטרים]) [As <תוצאה>]

<הליך>

<פונקציה> | <ביטוי> | <תנאי> | <משתנה> | <קבוע> = <שם פונקציה>

End Function

הערה: יש אפשרות לא להגדיר פרמטרים של הפונקציה באופן מלא אלא בצורה מרומזת. פונקציה שלא יוגדר הערך שהיא מחזירה – תחזיר ערך מטיפוס VARIANT

דוגמאות:

Function Energy(mass As Single) As Single

Energy = mass * (3 * 10 ^ 8) ^ 2

End Function

פונקציה שמחזירה אנרגיה

עפ"י נוסחת תורת היחסות

Function MaxOf Items (item1, item2) As Integer

MaxOf Items = IIf(item1 >= item2, item1, item2)

End Function

פונקציה שמחזירה את הגדול

שבשני מספרים

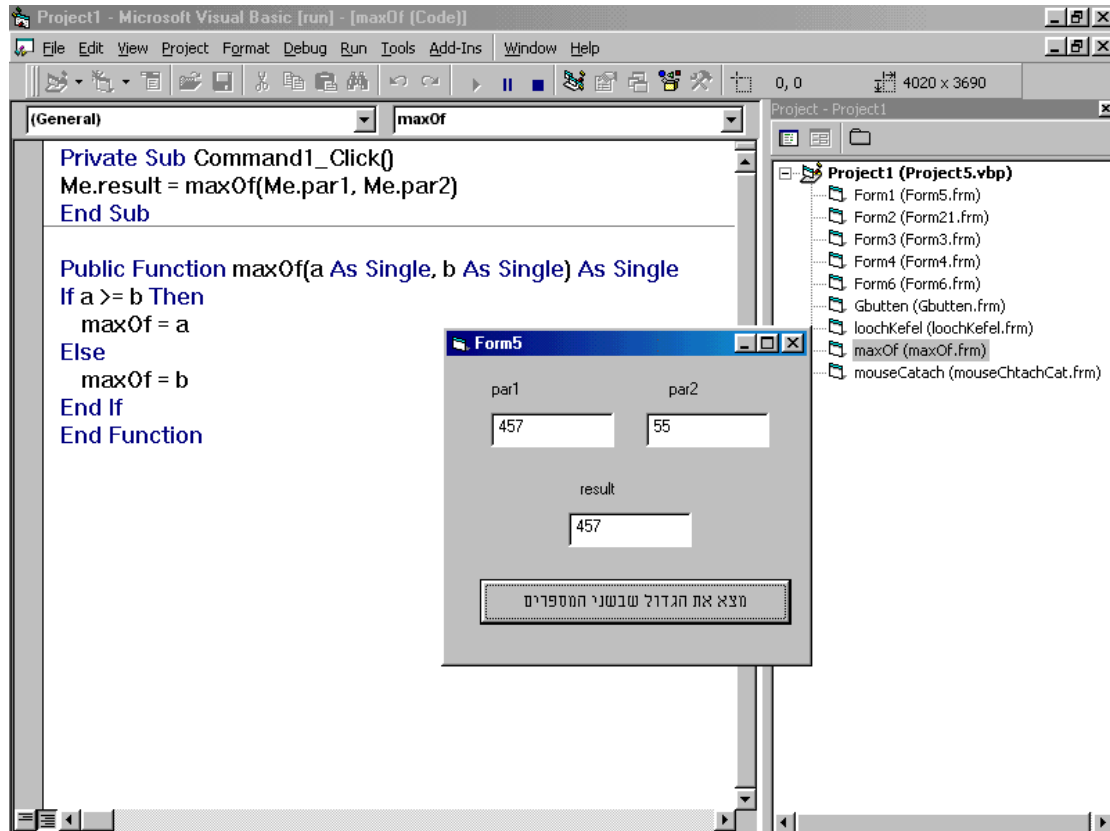
(פרמטרים) <שם פונקציה> = { <פקד> | <משתנה> }

קריאה לפונקציה:

maxOFNumbr = maxOf(12,78)

דוגמא:

ישום פונקציה בטופס



כשם שישנן פונקציות פנימיות של הויזואל בייסיק, כך ישנן שגרות אירוע פנימיות לפקדים השונים.

יש לשים לב שקריאה לפונקציה מחייבת סוגריים אחרי שם הפונקציה, לעומת זאת קריאה לשגרה מחייבת הורדת הסוגריים לאחר שם השגרה, אף אם יש לה פרמטרים. אפשר לקרוא לפונקציה כשם שקוראים לשגרה בלי להחזיר ערכים, אבל אז יש להקפיד על הורדת הסוגריים.

טווח פונקציה:

אפשר להכריז על שגרה/פונקציה כפרטית או ציבורית וכן כסטטית, ואז טווח משתנה יהיה בהתאם.

1.8.3. העברת פרמטר לפי כתובת ולפי ערך

ברירת המחדל בפרמטרים של VBA היא להעביר פרמטרים לפי כתובת, ולהשתמש בתוך הפונקציה באותו משתנה מקורי שהוגדר מוחץ לפונקציה, כך שאם מסיבה כל שהיא ערך הפרמטר ישתנה בפונקציה, הוא גם ישתנה במשתנה המקורי, בהגדרת הפונקציה, אם לא הגדרנו אחרת הפרמטר יוגדר כאילו שכתבנו את התחביר הבא:

Function SumAll(**ByRef** num As Integer)

באם רוצים להעביר פרמטר לפי ערך, שאז בביצוע הפונקציה יוגדר משתנה נוסף, ויועתק אליו תוכן המשתנה שהושם בפרמטר, יש להשתמש במילה השמורה **ByVal** לפני הגדרת הפונקציה.

Function SumAll(**ByVal** num As Integer)

היתרון של העברת פרמטר לפי כתובת הוא חיסכון במקום ובזמן חסרונו האפשרות לשנות משתנה באקראי ולהגיע לשיבוש נתונים.

2.8.3. פונקציות ושגרות עם פרמטרים אופציונליים ודינמיים

ישנה אפשרות להגדיר פונקציות שבהם חלק מהפרמטרים יהיו אופציונליים, וכן להגדיר פונקציות עם מספר פרמטרים דינמי.

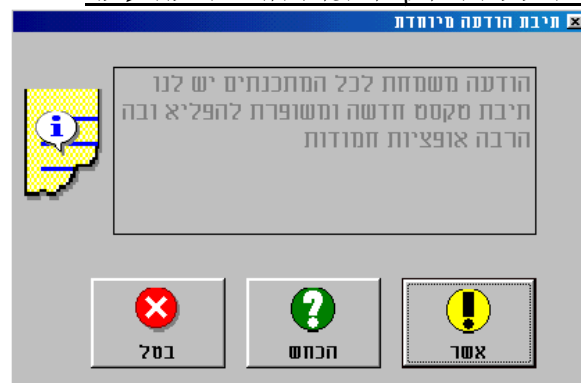
פונקציה בעלת פרמטר אופציונאלי

אנו ניצור פונקציה כזו כאשר נרצה לאפשר למשתמש לטפל בנתונים מסוימים אבל לא להכריחו. בכדי להגדיר פרמטר אופציונאלי יש להשתמש במילת המפתח **Optional** והמשתנה חייב להיות מסוג **Variant** בכדי לבדוק האם הפרמטר נרשם נשתמש בפונקציה **IsMissing()**

```
Public Function mat(arg1, arg2 As Single, Optional sug As Variant)
If IsMissing(sug) Then
    mat = arg1 + arg2
Else
    Select Case sug
    Case "+"
        mat = arg1 + arg2
    Case "-"
        mat = arg1 - arg2
    Case "*"
        mat = arg1 * arg2
    Case ":"
        mat = arg1 / arg2
    Case "^"
        mat = arg1 ^ arg2
    End Select
End If
End Function
```

אין לכתוב פרמטר רגיל לאחר פרמטר אופציונלי

תרגיל: ליצור msgbox חדש שי היה בו אופציות שלא קיימות ברגיל, כמו כפתורים גרפיים, תמונות, אפשרות לסגירה בעזרת השהיה, אפשרות לשלוט על גודל הגופן של ההודעה, אפשרות לפיצול שורות בעזרת תו בקרה פשוט ועוד כיד הדמיון הטובה, אך ביחד עם שלל התכונות אנו רוצים שמי שמשמש בו יוכל לכתוב רק פרמטר אחד בכדי להפעילו.



בעתיד יהיה אפשר לקמפל פרוייקט זה לקובץ DLL לשימוש חוזר בכל הפרוייקטים

פונקציה בעלת פרמטרים דינמיים

אנו ניצור פונקציה בעלת מספר לא מוגבל של פרמטרים, כאשר אנו רוצים לטפל במספר רב של נתונים באופן סדרתי, ואנו רוצים לאפשר למשתמש להעביר נתונים כפי רצונו.

שימוש אפשרי למשל, כאשר אנו רוצים שהמשתמש יעביר מספר מספרים בכדי שהפונקציה תחבר פעולה חשבונית ביניהם, או כאשר למשל אנו רוצים ליצור פונקציה שתבצע שליפה של נתון עפ"י אינדקס כדוגמת פונקצית **Choose**.

בכדי ליצור פונקציה עם מספר לא מוגבל של פרמטרים, נגדיר את הפרמטר כמערך מסוג **Variant**, עם המילה השמורה **ParamArray**.
כדוגמא הפונקציה מחברת בין כל הפרמטרים המוזנים לתוכה, תוך כדי שהיא משתמשת בלולאת **For Each**

```
Public Function SumAll(ParamArray num() As Variant)
```

```
Dim dbTotal As Single, var As Variant
```

```
For Each var In num
```


```
    dbTotal = dbTotal + var
```

```
Next var
```

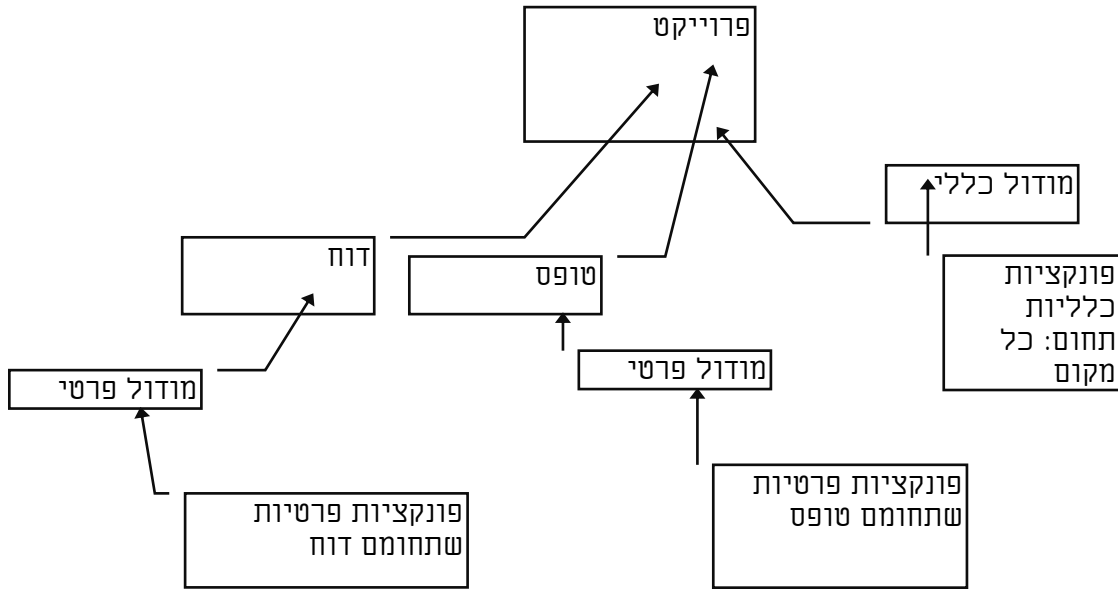
```
SumAll = dbTotal
```

```
End Function
```

הגבלות לגבי פונקציות

 אין לתת שם זהה לשני פונקציות אף הנמצאות בשני מודולים כלליים שונים, התוכנה לא תדע לאיזה פונקציה להתייחס, והיא איננה מבדילה ביניהם, אף לא לפי מספר הפרמטרים של הפונקציה.

תחומי פונקציות



מה אמרה שגרה אחת לשניה? "את לא פונקציה!!!"

9.3. מילים שמורות

מילים שמורות, הן מילים שהתוכנה שריינה לעצמה מראש, בכדי לבצע פעולות שונות בתוכנה. לכן אין אפשרות להשתמש במילים אלו, לבד מהפונקציונליות שהתוכנה שריינה להם, וליצור משתנים וקבועים ששמים כמילים השמורות. במקרה שננסה נקבל כמובן הודעת שגיאה.

דוגמאות למילים שמורות

dim
as
integer
byval
do
if
then
end
exit

דוגמא להגדרה לא חוקית

Dim do as byte

Do הינו מילה שמורה לתחילת לולאה, ולכן אי אפשר להגדירו כמשתנה.

01.3. אופרטורים (מפעילים)

האופרטורים הם תוי בקרה שגורמים לתוכנה לבצע מניפולציה על קטעי קוד ולהחזיר ערך מסוים, והם בעצם פועלים כמו פונקציות. ישנם אופרטורים שמחזירים ערכים גמישים כמו מספרים או מחרוזות, וישנם אופרטורים שמחזירים רק ערכים בוליאניים של אמת ושקר, כמו כן הם מתחלקים לשני סיווגים אונרים שפועלים על ביטוי אחד בלבד ובינאריים שפועלים על שני ביטויים.

	אופרטורים אריתמטיים מחזירים ערכים מספריים או מחרוזתיים
1 + 4	חיבור "+" פעולה: מחבר שני מספרים אחד לשני. אפשר לחבר גם בין מחרוזות.
7 - 9	חיסור "-" פעולה: מחסר שני מספרים אחד מהשני.
55 * 2	כפל - "*" פעולה: מכפיל שני מספרים אחד בשני.
34/88	חילוק "/" פעולה: מחלק שני מספרים ומחזיר ערך עשרוני.
88 \ 44	חילוק "\" פעולה: מחלק שני מספרים ומחזיר ערך שלם.
56 MOD 45	MOD פעולה: מחלק שני מספרים ומחזיר את השארית.
2 ^ 6	חזקה "^" פעולה: מעלה מספר בחזקה.
"T" & "אח"	שרשרת "&" פעולה: משרשר תווים אחד לשני.
Mishkal = "12121212"	שווה "=" פעולה: השמת ערך במשתנה.

	אופרטורים לוגיים מחזירים ערכים של אמת ושקר
a>5 And d>a	And
a>5 Or c=9	Or
a>5 Xor c=9	Xor
A = Not a	Not

	אופרטורים השוואתיים מחזירים ערכים של אמת ושקר
a>b	גדול מ - >
b<c	קטן מ - <
d=d	שווה ל - =
d >= d	גדול שווה - >=
d <= d	קטן שווה - <=
d <> not d	שונה מ - <>

קדימויות באופרטורים


יש סדר קדימויות בין האופרטורים השונים, כך למשל שהעלאה בחזקה קודמת לכפל וחילוק שקודמים לחיבור וחיסור שקודמים לפעולת השמה.

בכדי להקדים פעולה נמוכה בהיררכיה לגבוהה ממנה, נצטרך למסגר את הפעולה, בסוגריים עגולים.

באם אנחנו לא בטוחים בסדר קדימויות אופרטורים רצוי מאוד למסגר את האופרטורים שנרצה שיתקיימו בראשונה.

כך שאם נרצה להקדים פעולות חיבור לכפל, נצטרך לשימם בתוך סוגריים, באופן הבא. (a +b) * t

דוגמא:



$emc2 = m * (3 * 10 ^ 8) ^ 2$

11.3. תוי בקרה כללים

<u>תוי בקרה כללים</u>	
<code>k = k + 1 : nm = Val(Mid(mispar, lenOfMispar - k + 1, 1))</code>	הפרדה - תו נקודותיים " : " מפריד לשני משפטי קידוד שונים, או מציין שמדובר בכתובת.
<code>Dim a,d as long A=addDate("m",6,date())</code>	פסיק - תו פסיק " , " מפריד בהכרזת משתנים או בפרמטרים בפונקציה.
<code>'SUG: =2 מטבע=1 אגורות=</code>	הערה - תו גרש - ' גורם לתוכנה להתעלם ממשפט בשורה הנוכחית בלבד. כמו שורת הערה.
<code>"I love to love"</code>	גרשיים - "" מציין שמדובר בערך מחרוזתי.
<code>#01/01/98#</code>	סולמיות - "##" מציין שמדובר בערך תאריכי/שעה.
<code>Myfunc()</code>	סוגריים עגולים - "()" מציין שמדובר בפונקציה, או בסדר קדימויות.
	מעבר שורה - תו CRLF - ENTER גורם לתוכנה להתייחס לסוף פקודה.
<code>If I=10 and I=12 and _ I>55 then d=45</code>	קו תחתי - "_" קו תחתי בסוף שורה מחבר שתי שורות קוד לשורה אחת. כאילו היו פקודה שכתובה בשורה אחת.
תוי בקרה של אובייקטים	
<code>Me.refresh Me.controler1.requery</code>	נקודה - "." גורם לתוכנה להתייחס אל צד שמאלי של המשפט כאובייקט ואל הצד הימני כאל תת אובייקט או שם תכונה או שיטה של אותו אובייקט.

4. משפט השמה ומשפטי בקרה

משפט השמה הוא משפט בו התוכנה עושה השמה של ערך למשתנה או לפקד. הערך יכול להיות קבוע, אוסף קבועים, משתנה אחד, צירוף של כמה משתנים או פונקציה, או צירוף של כל הנתונים ביחד. בויזואל בייסיק משפט השמה חייב להיות בשורה נפרדת.

תחביר:

{<ביטוי> | <תנאי> | <פונקציה> | <משתנה>} = {שם משתנה | שם פקד. תכונה}

X=1

X=X+1

X=X/Y*Z+10

X=getFiratDate(#12/12/99#)+1

Text1.Caption="read my lipestik"

בכל הדוגמאות ערכו של המשתנה X הינו הערך שהושם לתוכו. בדוגמה הראשונה ערכו יהיה 1, בשניה ערכו + 1, בדוגמה האחרונה יהיה הערך שמחזירה הפונקציה + 1.

ההגבלה הראשית בהשמה הינה, שיש להשים את אותו סוג ערך שהמשתנה אמור לקבל, אחרת במקרה הטוב יהיה שיבוש נתונים ובמקרה הרע תיווצר שגיאת הרצה. לדוגמא אם אשים את הערך 4.7 למשתנה מסוג מספר שלם, תהיה קציצה של הנתונים אחרי הנקודה והוא יקבל לתוכו את הערך 4. לעומת זאת אם אשים את הערך המחרוזתי "שלוס" למשתנה מסוג מספרי, או אשים ערך מספרי למשנה מסוג מחרוזת, אקבל הודעת שגיאה, שתביא להפסקת הרצת התוכנה.

הגבלה שניה בהשמה היא, שיש להשים את גודל הנתונים שהמשתנה יכול להכיל, בכדי שלא תהיה גלישה. כך לדוגמא אם המשתנה הינו מסוג BYTE, אין להשים לתוכו מספר הגדול מ-255 אחרת תתבצע גלישה, כן אין להשים לתוך משתנה מחרוזת בגודל 4 את המילה "העולם" שהיא בגודל 5.

אין להשים ערך ריק NULL לכל משתנה חוץ מטיפוס Variant

מדון
מפני

משפטי בקרה

משפטים אלו מנהלים את התוכנה ומורים לה כיצד ל"זרום". איזו שורות יתבצעו, באיזה תנאי, וכמה פעמים.

בכל מקום בו יש <תנאי> התנאי חייב להחזיר ערך בוליאני של אמת - TRUE או שקר - FALSE התנאי יכול להיות ביטוי, צירוף של ביטויים, משתנה, ואף פונקציה.

<pre>If A < 0 And B > 3 Then MsgBox "מספר תקין"</pre>	<p>במשפט IF למשל כל התנאים הבאים נכונים.</p>
<pre>If A Then MsgBox "מספר תקין"</pre>	
<pre>If bdkSifratBikuret(54561812) Then MsgBox "מספר תקין"</pre>	
<pre>End If</pre>	

<i>IF משפט 1.4</i>	
<pre>If <תנאי> Then [משפט פקודה] [ElseIf <תנאי>] [משפט פקודה] [Else] [משפט פקודה] End If</pre>	<p>IF משפט ביצוע פקודות באופן סלקטיבי לפי התניה.</p>
<pre>If txtNatunLehachpala = "" Then CmdFirstPos.Enabled= False Else CmdFirstPos.Enabled= True End If</pre>	<p>דוגמא:</p>

<i>CASE משפט 2.</i>	
<pre>Select Case <משתנה> Case <ערך> [משפט פקודה] Case <ערך>, <ערך> [משפט פקודה] Case <ערך> To <ערך> [משפט פקודה] Case Is <ערך> <אופרטור השוואתי> [משפט פקודה] [Case Else] MsgBox "ערך שלא ניתן להתייחס אליו"</pre>	<p>CASE משפט ביצוע פקודות באופן סלקטיבי באופן מרובה תנאים. מתבצעת בדיקה של המשתנה מול הערכים שאנו מציינים לו בקייסים. אם מתברר שהשוואה זהה ונותנת ערך אמת, המשפט שאחרי ההשוואה מתבצע, והתוכנה מדלגת לסוף הקטע, ולא ממשיכה</p>

End Select

הערך יכול להיות ערך בודד, טווח ערכים, או כמה ערכים. כמו כן, אנו יכולים להשתמש גם באופרטורים השוואתיים ובולאנים בשימוש באופרטור IS

דוגמא

לבדוק את הקייסים האחרים. יעיל במקרה שיש הרבה תנאים לבדוק, שאז משפט ה if הופך להיות מסורבל.

Case 6

ערך בודד -

Case 6 To 77

טווח ערכים -

Case 6,13,77,88,101

מספר ערכים -

Case Is > 6

השוואה

```

Private Sub Command1_Click()
    selectCase
End Sub

Public Sub selectCase()
    Dim str As String

    Select Case Text1
        Case "ראשון", "Sunday", "sun": str = 1
        Case "שני", "Monday", "mon": str = 2
        Case "שלישי", "Tuesday", "Tus": str = 3
        Case "רביעי", "Wednesday": str = 4
        Case "חמישי", "thursday": str = 5
        Case "ששי", "Friday": str = 6
        Case "שבת", "Saturday", "Sat": str = 7
        Case Else: str = "אין יום כזה"
    End Select

    Text2 = str
End Sub
    
```

תרגיל: יש לכתוב פונקציה שממירה את ימי השבוע ממילים למספר

```

Private Sub calcMas_Click()
    Dim achoozMas As Currency

    Select Case bruto
        Case 0 To 2000: achoozMas = 0
        Case 2001 To 4000: achoozMas = 0.15
        Case 4001 To 6000: achoozMas = 0.3
        Case 6001 To 20000: achoozMas = 0.45
        Case 20001 To 100000000: achoozMas = 0.5
    End Select

    mas = bruto * achoozMas
    neto = bruto - mas
End Sub
    
```

תרגיל: יש לכתוב תוכנית שתקבל משכורת ברוטו ותחזיר לנו את המס שיש לשלם ואת המשכורת נטו. אנו יוצאים מתוך נקודת הנחה שיש רק מדרגת מס אחת.

```

Private Sub calcMas_Click()
    Dim achoozMas As Currency

    Select Case True
        Case bruto < 2000: achoozMas = 0
        Case bruto > 2001 And bruto < 4000: achoozMas = 0.15
        Case bruto > 4001 And bruto < 6000: achoozMas = 0.3
        Case bruto > 6001 And bruto < 20000: achoozMas = 0.45
        Case bruto > 20001: achoozMas = 0.5
    End Select

    mas = bruto * achoozMas
    neto = bruto - mas
End Sub
    
```

אפשרות לרשום תנאים בולאניים במשפט Case באופן הבא:

3.4 . לולאה FOR	
<p>For <משתנה> = <ערך1> To <ערך2> [Step <ערך>] [משפט פקודה] [Exit For] Next [<משתנה>]</p>	<p>הלולאה מאפשרת ביצוע קטע קוד מספר פעמים כפי שנקבע, בטווח שבין שני מספרים. המשתנה מאותחל בפעם הראשונה לערך 1 ובכל סיבוב הלולאה מועלה ערכו עד שמגיע לערך 2.</p>
<p>המשתנה I יאותחל לערך 1 והפקודה תתבצע מספר פעמים שבין ההפרש בין שני הערכים.</p> <p>המשתנה I הוא משתנה לכל דבר, ואפשר לשנותו בזמן הרצת הלולאה. הדילוג הינו ביחידה חיובית שלמה, באם רוצים לשלוט בערך הדילוג, יש להוסיף את האופציה Step ולציין את ערך הדילוג החדש.</p>	
<pre>For I = 1 To 10 For j = 1 To 10 Debug.pring (I*j); Next Debug.pring Next</pre>	<p>דוגמא : הדפסת לוח הכפל</p>

4.4 . לולאת FOR EACH	
<p>לולאה מיוחדת שפועלת על מערכים ואוספים, ומאפשרת לעבור עליהם באופן סדרתי, מבלי לבדוק את גודלם, כאשר הגישה לאיברים הינה סדרתית מהאיבר הראשון עד לאחרון באמצעות משתנה מצביע, שבכל סיבוב של הלולאה מצביע המשתנה על האיבר הבא באוסף. יתרונה של הפקודה הוא הפשטות, גם במעבר על כל האיברים מבלי אפשרות לטעות. וגם בהצבעה והשחרור האוטומטי של המצביע.</p>	
<pre>For Each <משתנה מצביע> In {<אוסף> <מערך>} [ערך]=[תכונה].<איבר - משתנה מצביע> [משפט פקודה] Next <משתנה></pre> <p style="text-align: right;">כאשר</p> <pre>Dim <משתנה מצביע> As object control variant</pre> <p style="text-align: right;">1. איפוס של מערך</p> <pre>Dim myArray(100) As Byte Dim currentId As Variant</pre>	<p>מאפשרת לנו ליצור לולאה לפי מספר האיברים הקיימים במערך או באוסף. כאשר ההתייחסות לאיבר הנוכחי היא בעזרת המשתנה שיכול להיות מסוג אובייקט כללי, ספציפי או וריאנט.</p> <p>באם רוצים להתייחס לאיבר ספציפי צריך להשתמש בתכונת Index</p>

```

For Each currentId In myArray
    CurrentId = 0
Next
        2. הדפסת שמות כל הפקדים שבטופס הטעון הראשון
For Each myControl In forms(0)
    Debug.print myControl.Name
Next
    
```

לולאה זו הינה ייחודית לשפת היוזואל בייסיק.

Do .5.4 לולאת Loop

Do	בכל מקרה משפט הפקודה יתבצע לפחות פעם אחת, ורק אז תתבצע בדיקה.	DO LOOP
[משפט פקודה]		
[Exit Do]		
[משפט פקודה]		
Loop {While Until} <תנאי>		

DO .6.4 לולאת While | Until

Do {While Until} <תנאי>	תתבצע בפעם הראשונה בדיקה, ורק אם התנאי עונה תתבצע הפקודה.	לולאת DO While Until כנ"ל. בדיקת התניה לאחר ביצוע פקודה.
[משפט פקודה]		
[Exit Do]		
[משפט פקודה]		
Loop		

בתנאי **UNTIL** הפקודה תתבצע כל זמן שהתנאי ערכו שקר
בתנאי **WHILE** הפקודה תתבצע כל זמן שהתנאי ערכו אמת

Do a = a + 1 Loop Until a = 10	Do Until a = 10 a = a + 1 Loop
--	--

```

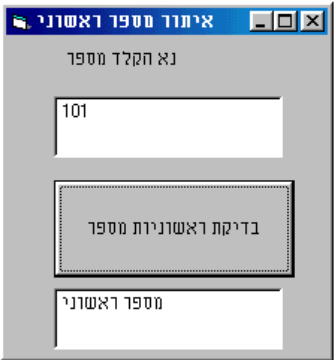
Dim myArray(100) As Integer, i As Integer
MyArray(58) = 1999
    
```

```

Do Until myArray(i) = 1999
    i = I + 1
Loop
MsgBox i
    
```

```

Do While myArray(i) <> 1999
    i = I + 1
Loop
MsgBox I
    
```

<pre>Private Sub Command1_Click() Dim i As Long i = 1 Do i = i + 1 Loop Until Text1 Mod i = 0 If Text1 = i Then Text2 = "מספר ראשוני" Else Text2 = "לא מספר ראשוני" End If End Sub</pre>		<p>בדיקת מספר ראשוני</p> <p>מה יקרה כאשר נקיש את המספר 1, וכיצד אנחנו יכולים לטפל בבעיה?</p> <p>הערה: בכדי לצאת מלולאה אין סופית יש להקיש CTRL+Break</p>
---	--	---

	<p><i>.7.4 לולאת</i> WHILE E</p>
--	--

<p>While <תנאי> [משפט פקודה] Wend</p>	<p>כל זמן שהתנאי מחזיר "אמת" הפעולה תתבצע שוב ושוב. אם בפעם הראשונה התנאי מחזיר "שקר" המשפט לא יתבצע כלל</p>	<p>ביצוע קטע באופן רב פעמי לפי התניה</p>
---	--	--

<pre>While nm = 0 k = k + 1: nm = Mid(mispar, lenOfMispar - k + 1, 1) Wend</pre>	<p>דוגמא:</p>
--	---------------

	<p><i>.8.4 יציאה, דלג ל, וכתובת</i></p>								
<p>Exit Function Exit Sub Exit For Exit Do End</p>	<table style="width: 100%;"> <tr> <td style="width: 30%;">יציאה מאזור פונקציה</td> <td style="width: 70%;">יציאה יזומה מאזור קוד.</td> </tr> <tr> <td>יציאה מאזור שגרה</td> <td>בכל המקרים התוכנית תמשיך לרוץ כאילו סיימה את המבנה שממנו יציאה.</td> </tr> <tr> <td>For</td> <td>יציאה מאזור לולאת</td> </tr> <tr> <td>DO</td> <td>יציאה מאזור לולאת הפסקת ביצוע הקוד</td> </tr> </table>	יציאה מאזור פונקציה	יציאה יזומה מאזור קוד.	יציאה מאזור שגרה	בכל המקרים התוכנית תמשיך לרוץ כאילו סיימה את המבנה שממנו יציאה.	For	יציאה מאזור לולאת	DO	יציאה מאזור לולאת הפסקת ביצוע הקוד
יציאה מאזור פונקציה	יציאה יזומה מאזור קוד.								
יציאה מאזור שגרה	בכל המקרים התוכנית תמשיך לרוץ כאילו סיימה את המבנה שממנו יציאה.								
For	יציאה מאזור לולאת								
DO	יציאה מאזור לולאת הפסקת ביצוע הקוד								

דילוג ל

<p>GoTo <כתובת></p>	<p>דלג ל הקפצת ביצוע תוכנית למקום אחר לפי כתובת שקיימת. באופן כללי לא נשתמש בפקודה זו, לבד ממקרה של לכידת שגיאות.</p>
----------------------------------	---

on error GoTo Err_YeshKabalot

דוגמא : לכידת
טעות

כתובת

<כתובת>:

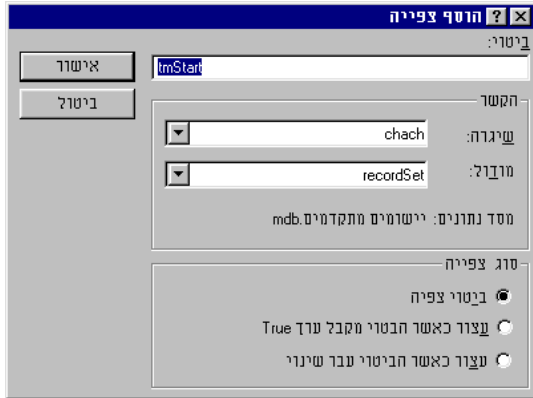
Err_YeshKabalot:

כתובת

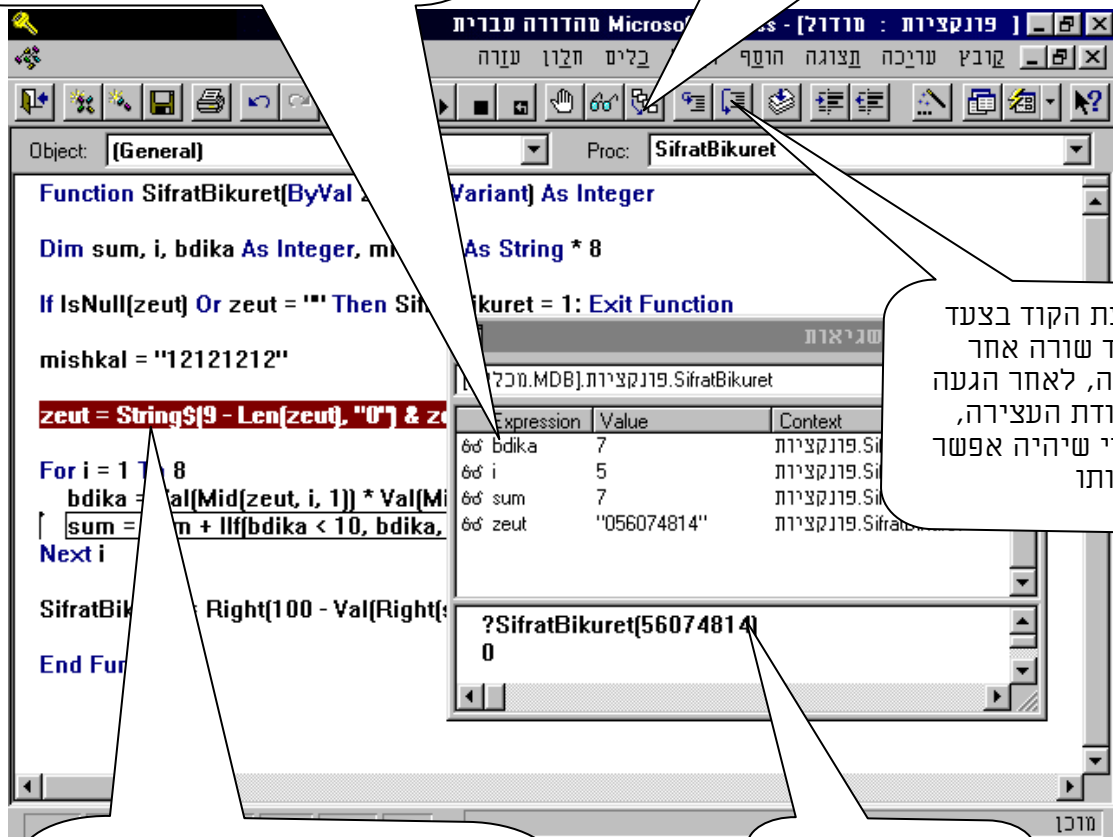
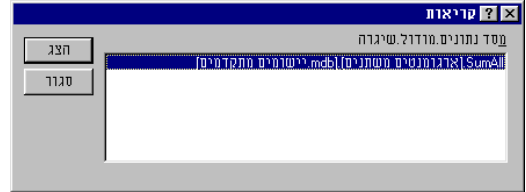
הכתובת מזוהה בשם שאחריו נקודותיים.
הכתובת הנה סימניה באזור כל שהוא בקוד
שמאפשרת למתכנת להקפיץ את מהלך הקוד
אל הכתובת, בעזרת פקודת Goto.
מבחינת התוכנית אין שום משמעות לכתובת
והתוכנה מתעלמת ממנה כאילו הייתה הערה.

עבודה בסביבת הפיתוח .9.4

חלון לצפייה לערכי המשתנים.
 בכדי להוסיף משתנים לצפייה יש להקיש
 {כלים} {הוספת צפייה}



אפשרות להגיע לקריאות פונקציה בזמן הרצה



הרצת הקוד בצעד בודד שורה אחר שורה, לאחר הגעה לנקודת העצירה, בכדי שיהיה אפשר לנפותו

הרצת פונקצייה/שגרה בכתיבת שמה ENTER

קביעת נקודת עצירה F9 {הפעל} {הוסף/הסר} מאפשר לדבג את הפונקציה ולראות ערכי משתנים בזמן הרצה. יש לשים נקודות עצירה במקומות בהן אנו רוצים לבדוק ערכי משתנים

01.4. שילוב קוד בטפסים/דוחות

אנו מניחים שקיימת לנו הפונקציה שמטרתה לבדוק תקינות מספר תעודת זהות, ואנו רוצים להפעילה על בקר "תעודת זהות" בטופס אלפון:

א. אפשר לבצע בדיקה לאחר הקלדה ולתת הודעת שגיאה במקרה שהמספר אינו תקין. הדבר יתבצע באופן הבא:

```
Sub תעודת_זהות_AfterUpdate ()
If Not SifratBikuret(Me![תעודת זהות]) Then MsgBox "מספר תעודת זהות שגוי"
End Sub
```

ב. אפשרות נוספת היא ליצור פקד מקושר לפונקציה, שבו ייכתב סטטוס הספרה של תעודת הזוהות. לשם כך יש לבסס את הטופס על שאילתא שבה ניצור שדה מחושב המבוסס על הפונקציה כשהוא מקבל כפרמטר את שדה תעודת הזוהות.

```
[אימות זהות] AS (אלפון![תעודת זהות]) SifratBikuret
```

בשלב השני אנו נגדיר את תכונת "תבנית" של פקד "אימות זהות" באופן הבא.

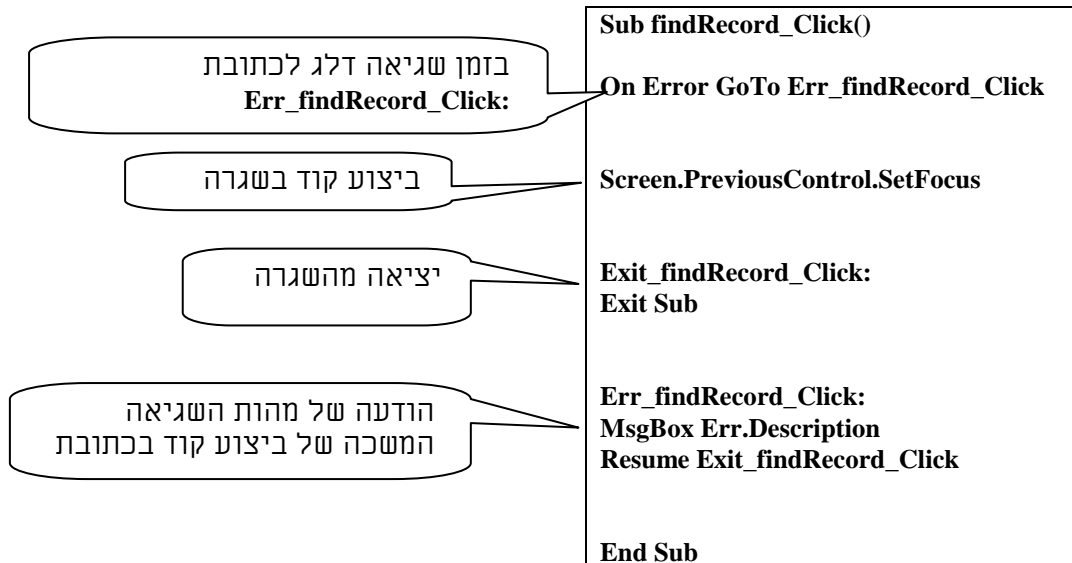


במקרה שהמספר תקין הערך שיוחזר הינו שלילי (-1) ולכן תיבת הטקסט "אימות זהות" תציג את הערך "תקין" בצבע כחול, במקרים האחרים יוצג הערך שגוי או "עדיפות למספר מלא" באדום.

יתרון השיטה הזו היא שתמיד ישנה תצוגה של מספר תעודת הזוהות, והמשתמש אינו צריך לנקוט בפעולה מיוחדת בכדי לוודא שהמספר אכן תקין.

11.4. לכידת שגיאות

בלי לכידת שגיאות, בעת שגיאה פעילות התוכנה תיעצר, בליווי הודעת שגיאה.



בפונקציה זו המטרה היא להחזיר ערכים לפי אינדקס

```
Public Function selectFrom(mispar As Byte) As String
Const st = "#1,30,#2,30,#3,29,#4,29,#5,77,#6,23,#7,21,#8,65,#9,12,#10,44"
selectFrom = Mid$(st, InStr(1, st, "#" & mispar) + IIf(mispar < 10, 3, 4), 2)
End Function
```

בפונקציה זו המטרה היא להמנע מתחביר רב בצירוף של תנאים לוגיים

כך שבמקום לכתוב את שורות הקוד הללו:

```
If (currentHyear.cycle = 2 Or currentHyear.cycle = 5 Or currentHyear.cycle = 7 _
Or currentHyear.cycle = 10 Or currentHyear.cycle = 13 Or currentHyear.cycle = 16 _
Or currentHyear.cycle = 18) Then
IsCurrentYearLeap = True
Else
IsCurrentYearLeap = False
End If
```

אפשר להסתפק בשורה אחת בלבד

```
IsCurrentYearLeap= isIn(currentHyear.cycle,2,5,10,13,16,18)
```

הפונקציה מקבלת שני פרמטר ראשון את הערך להשויה וכפרמטרים נוספים את הערכים שבתנאי ה - OR

```
Public Function isIn(mishtane, ParamArray num() As Variant)
Dim var As Variant, yes As Boolean
yes = False

For Each var In num
If mishtane = var Then yes = True
Next var

isIn = yes

End Function
```



תרגיל:

יש לכתוב פונקציה לבדיקת סיפרת ביקורת של תעודת זהות, ולשלב אותה בשאילתא ובטופס.

שיטת החישוב של משרד הפנים:

1. המספר שמבצעים עליו את החישוב הינו בן 8 ספרות. אם המספר קטן מ 8 ספרות משלימים אותו באפסים מצד שמאל (לדוגמא: 00000456=456)
2. מכפילים את הספרות המרכיבות את המספר במשקלים 1,2 מימין לשמאל
3. מסכמים את כל הספרות בתוצאה (יחידות ועשרות ביחד).
4. הספרה המשלימה לעשרת הבאה היא ספרת הביקורת.

לדוגמא:

1. מספר הזהות: 0 5 0 6 3 4 3 3
 2. הכפלה ב 1,2: 1 2 1 2 1 2 1 2
 3. סכום כל הספרות = 24 0 10 0 12 3 8 3 6
 4. השלמה לעשרת הבאה 6 = 30 - 24
- סיפרת הביקורת הינה 6



תרגיל:

יש לכתוב פונקציה שתקבל שכר חודשי ברוטו ותחזיר את סכום מס ההכנסה שעל בעל השכר לשלם לפי מדרגות המס הקיימות. יש לכתוב פונקציה שתאחזר את סכומי מדרגות המס ממערך, ופונקציה נוספת שתאחזר את מדרגות המס מטבלה, יש להשתמש בפונקציה = DLookup("[mas]![derug]", "[mas]", "[mas]![num]=1") ששולפת נתון משדה בטבלה לפי ערך בשדה אחר.
 (=DLookup("<ערך>=[שם שדה]", "[שם טבלה]", "[שם שדה]"),

אחוז המס	הכנסה
0 אחוז	0
30 אחוז	2215
45 אחוז	3158
50 אחוז	8313

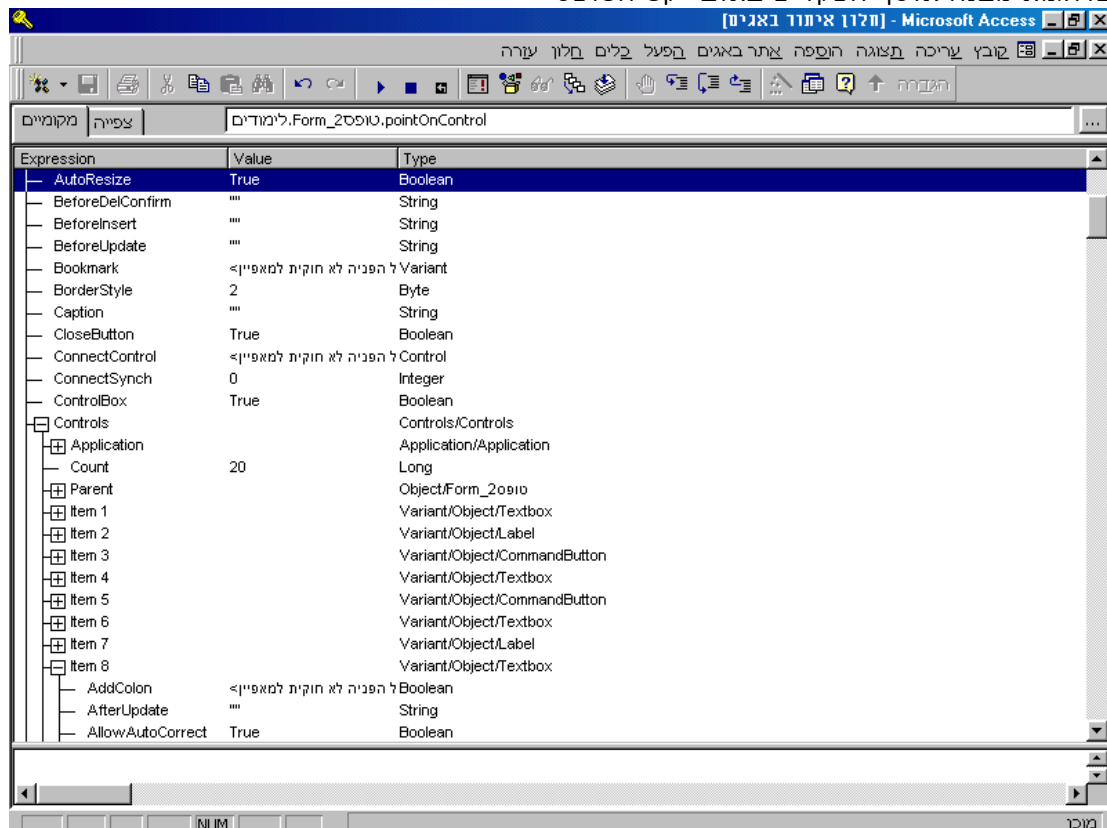
21.4. אובייקט הטופס כאוסף פקדים

הקדמה

פקדי מכולה אלו פקדים שמכילים פיזית פקדים אחרים. (טופס, פקד כרטסת, פקד מסגרת הינם אובייקטי מכולה)
 פקדי אוספים אלו אובייקטים שמכילים בתוכם אוסף של אובייקטים אחרים, לעיתים פקדים ולעיתים לא. (טופס, פקד תיבת רשימה, אובייקט ה-Forms הינם אובייקטי אוסף)
 לפקד אוסף יהיו תכונות Count שסופר את מספר הפריטים וitem, שמאפשר גישה אל תכונות כל פריט ופריט. הטופס שהינו בעצמו אובייקט אוסף של פקדים יש תכונות Controls שמאפשר להצביע ולהתייחס לכל פריט באוסף.

מבנה אוסף

בדוגמא מבנה אוסף הפקדים באובייקט הטופס



1.21.4. יצירת מצביע לפקד

לעיתים נרצה ליצור משתנה שיצביע על אובייקט מסוים בכדי לעשות כן אנו נכריז על המשתנה, ונהפוך אותו למצביע ע"י פקודת Set

```
Public Function pointOnControl()
Dim myc As Control
Set myc = Me.myControl
myc.BackColor = vbRed
Set myc = Nothing
End Function
```

כאשר אנחנו משנים את תכונת המשתנה כאשר הוא מצביע על אובייקט, תכונותיו של האובייקט ישתנו, והשינוי ישאר גם אם נשחרר את המצביע מלהצביע על האובייקט.

31.4. מבנה בקרה – חזרה תוך שימוש באובייקט הטופס

פקודת If

המטרה לעמעמם את כפתור "הדפס קבלות" במקרה שאין קבלות ולהפך. יש ליצור פקד בשם "מספר קבלה" ובתכונת מקור פקד פונקציית ספירה על טבלת

קבלות ממתינות: =DCount(";", "[קבלות]", "Null= [תאריך קבלה])

הדפס קבלות

קבלות

```
Private Sub Form_Open(Cancel As Integer)
```

```
    If מספר_קבלות > 0 Then
```

```
        Me.הדפס_קבלות.Enabled = True
```

```
    Else
```

```
        Me.הדפס_קבלות.Enabled = False
```

```
    End If
```

```
End Sub
```

הסבר: אנו יוצרים פקד ששמו "מספר קבלות" והוא סופר את מספר הרשומות שממתינות בתור הקבלות. באם אין קבלות ממתינות הוא הופך את תכונת הלחצן ללא מאופשר.

לא מאוגד

לא מאוגד

לא מאוגד

לא מאוגד

לא מאוגד

לא מאוגד

לא מאוגד

טקסט:ט

טקסט:ט

טקסט:ט

טקסט:ט

טקסט:ט

טקסט:ט

טקסט:ט

צבע פקד לפי סוגו

צבע רקע של פקדים באדום

הצג שמות פקדים והסתר אותם

הסט פקדי תיבת טקסט שמאלה טקסט ולהסתרם

בדוגמאות אנו משתמשים בפונקציית TypeName שפועלת על אובייקט ומחזירה את סוגו.

פקודת For

המטרה להציג את שמות כל הפקדים שבטופס בתיבת טקסט ולהסתרם

```
Private Sub Click() Click() הצג_שמות_פקדים_והסתר_אותם
```

```
    Dim i As Integer
```

```
    For i = 0 To Me.Count - 1
```

```
        myControl = myControl & Me.Controls.Item(i).Name & vbCrLf
```

```
        Me.Controls(i).Visible = False
```

```
    Next
```

```
End Sub
```

הסבר: אנו עוברים על כל אוסף הפקדים, כאשר אנחנו מצביעים כל פעם על פריט אחר באמצעות אובייקט ה"פקדים".

For Each פקודת

המטרה לצבוע את הרקע של כל הפקדים באדום למעט פקדי לחצני פקודה, מכיוון שאין להם תכונת רקע.

```
Private Sub צבע_רקע_של_פקדים_באדום_Click()
Dim thecontrol As Control
For Each thecontrol In Me
    If TypeName(thecontrol) <> "CommandButton" Then
        thecontrol.BackColor = vbRed
    End If
Next
End Sub
```

הסבר: אנו יוצרים משתנה מסוג מצביע פקד. בללואה אנו מבקשים שהיא תתבצע כמספר הפקדים שבטופס. בכל פעם שהיא מתבצעת המשתנה מצביע על הפקד הבא בסדרה עד לפקד האחרון.

Do While פקודת

המטרה להסיט את כל פקדי תיבת הטקסט לצד שמאל של הטופס

```
Private Sub הסט_הסט_פקדי_תיבת_טקסט_שמאלה_Click()
Dim i As Integer
Do While Me.Count > i
    If TypeName(Controls(i)) = "textBox" Then
        Me.Controls(i).Left = 0
    End If
    i = i + 1
Loop
End Sub
```

הסבר: אנו מסיטים את כל פקדי תיבת הטקסט אל שמאל המסך על ידי כך שאנו משימים לתכונת שמאל להם ערך 0.

Select Case פקודת

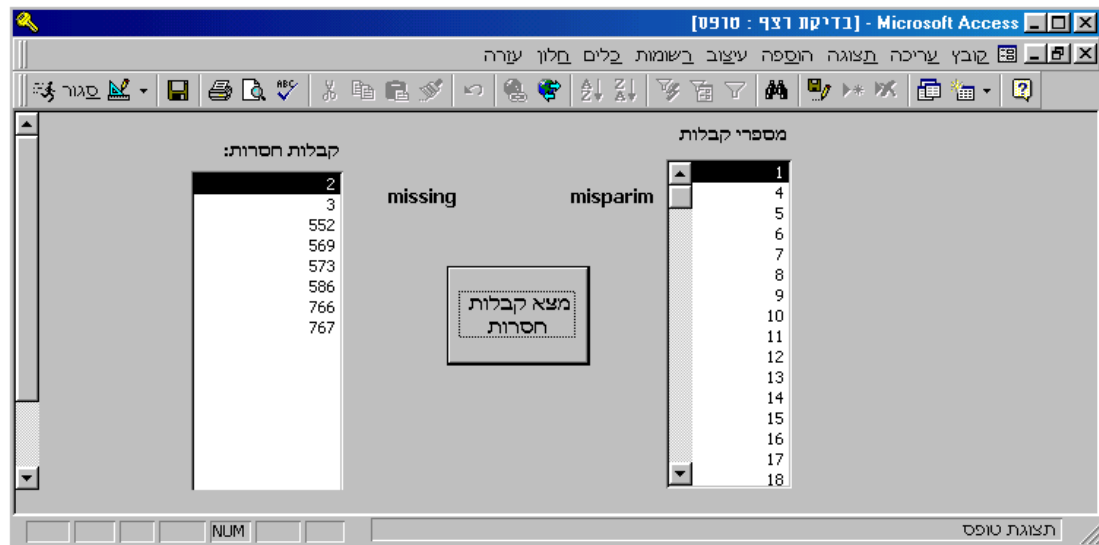
המטרה לבצע פעולה שונה בפקדים לפי סוגיהם

```
Private Sub צבע_פקד_לפי_סוגו_Click()
Dim thecontrol As Control
For Each thecontrol In Me
    Select Case TypeName(thecontrol)
        Case "label": thecontrol.BackColor = vbRed
        Case "textBox": thecontrol.BackColor = vbGreen
        Case "CommandButton": thecontrol.ForeColor = vbRed
    End Select
Next
End Sub
```

הסבר: אנו באופן סלקטיבי לפי סוג הפקד משנים את אחת התכונות שלו. אם הפקד הינו מסוג תווית אנו הופכים את הרקע שלו לאדום וכן הלאה

41.4. בדיקת רצף באמצעות פקד תיבת רשימה

המטרה היא לבדוק האם יש רצף במספרי הקבלות, כפי שדורש מס הכנסה. תחילה יש לשלוף את הנתונים לתיבת רשימה בשלב השני יש לדרוש את הקוד הבא, שסורק את כל הפריטים שבתיבת הרשימה, ורושם את מספר הקבלות החסרות בתיבת רשימה שניה.



פתרון א.

```

Private Sub מצא_קבלות_חסרות_Click()
Dim indexItem As Long, tzover As Long, str As String
tzover = 1
For indexItem = 0 To misparim.ListCount - 1
    Do While misparim.ItemData(indexItem) <> indexItem + tzover
        str = str & indexItem + tzover & “;”
        tzover = tzover + 1
    Loop
Next
missing.RowSourceType=“רשימת ערכים”
missing.RowSource = str
End Sub
    
```

פתרון ב'

```

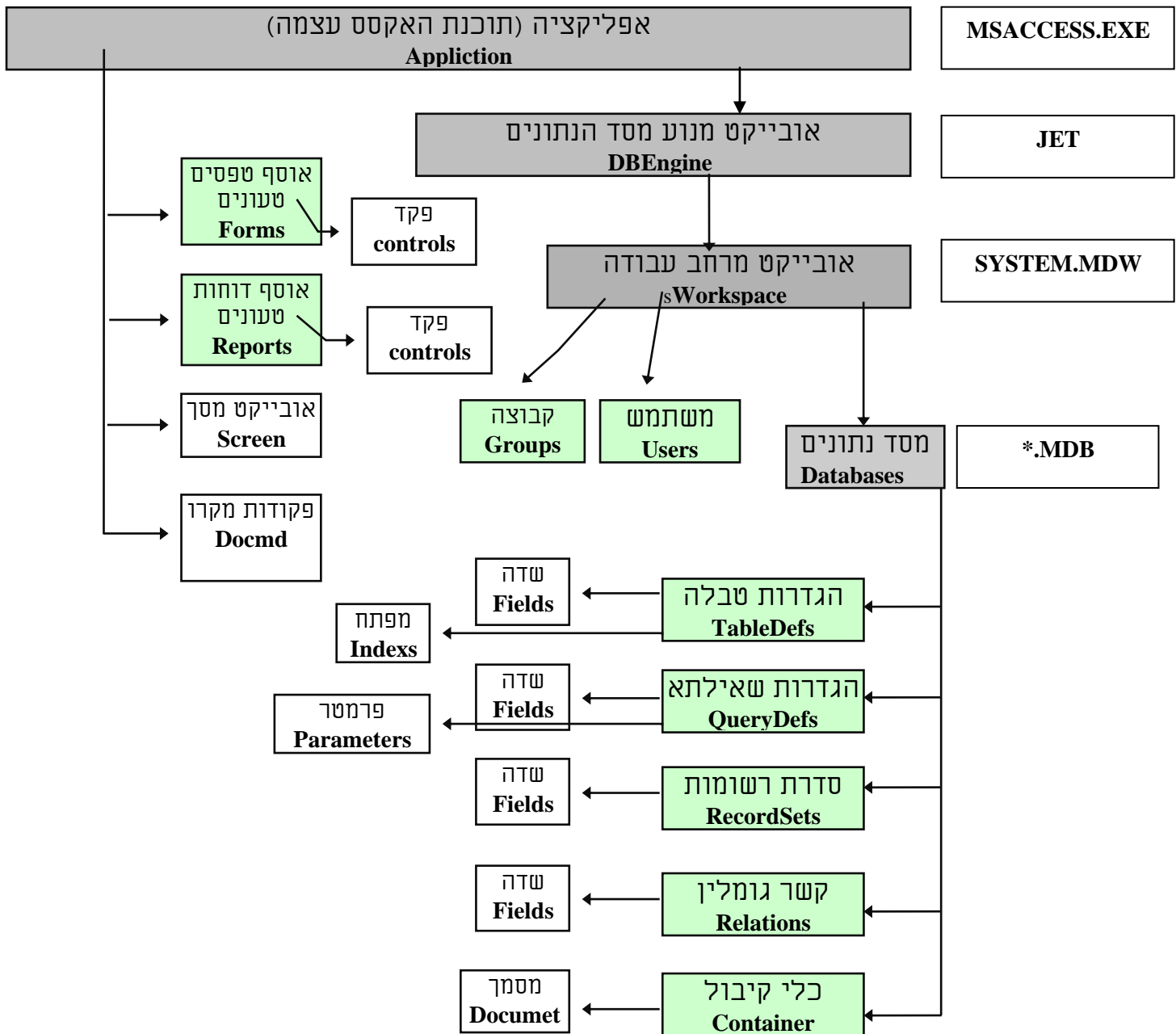
Private Sub מצא_קבלות_חסרות_Click()
Dim indexItem As Long, ratz As Long, str As String
ratz = 1
For indexItem = 0 To misparim.ListCount - 1
    If misparim.ItemData(indexItem) <> ratz Then
        str = str & ratz & “;”
        ratz = ratz + 1
    End If
    ratz = ratz + 1
Next
missing.RowSourceType=“רשימת ערכים”
missing.RowSource = str
End Sub
    
```

הערה: יש לשים לב שתכונת RowSource מקבלת מחרוזת בגודל של בין 255 לכ2004תווים.

5. אובייקטים ואוספים

הקדמה: בנוסף לאובייקטים הבסיסיים של אקסס שהכרנו אותם מחלון מסד נתונים, יש אובייקטים נוספים שאנו יכולים להגדירם כמשתנים, ולבצע בעזרתם מניפולציות כוללניות על בסיס הנתונים. נהוג לחלק את אובייקטי האקסס לשתי משפחות. אובייקטים רגילים, ואוספים שהינם אובייקטים המכילים אובייקטי בנים. כך לדוגמא חלון ניפוי שגיאות DEBUG הינו אובייקט עם תכונות ושיטות, אך האובייקט סדרת רשומות RecordSet הינו אוסף שמכיל בתוכו אובייקטי שדות.

ההיררכיה של אובייקטי הגישה למידע (DAO)



הקשר אל אובייקט בן עקרונית יהיה דרך אובייקט האב. בצורה הבאה: מנוע בסיס נתונים. מרחב העבודה. מסד נתונים נוכחי. סדרת רשומות

1.5 מבנה בסיסי של האובייקטים שבאקסס

1. ספריית אפליקציית האקסס

כאן נכללים כל האובייקטים וכל הפונקציות שקשורות להפעלת היישום ולצד הממשקי שלו, אובייקטי הטפסים, הדוחות והמודולים על כל מה שהם מכילים.

Access - Access.Constants (ac...)

Access.**Application**

Access.Application.Assistant

Access.Application.DoCmd

Access.Application.Screen

Access.Application.**Forms**

Access.Application.Forms(X).Controls

Access.Application.**Reports**

Access.Application.Reports(X).Controls

Access.Application.**Modules**

Access.Application.Modules(X).Lines

Access.Application.**References**

2. ספריית מנוע בסיס הנתונים JET

בו נכללים כל היכולות של בסיס הנתונים, הגדרת נתונים, שמירתם, שליפתם וכדומה.

DAO - DAO.LanguageConstants (db...)

DAO.**DBEngine**

DAO.DBEngine.**Workspaces**

DAO.DBEngine.Workspaces(X).Groups

DAO.DBEngine.Workspaces(X).Users

DAO.DBEngine.Workspaces(X).(**Connections**

DAO.DBEngine.Workspaces(X).(**Databases**

DAO.DBEngine.Workspaces(X).Databases(X).TableDefs

DAO.DBEngine.Workspaces(X).Databases(X).TableDefs(X).Fields

DAO.DBEngine.Workspaces(X).Databases(X).TableDefs(X).Indexes

DAO.DBEngine.Workspaces(X).Databases(X).(**QueryDefs**

DAO.DBEngine.Workspaces(X).Databases(X).QueryDefs(X).Fields

DAO.DBEngine.Workspaces(X).Databases(X).QueryDefs(X).Parameters

DAO.DBEngine.Workspaces(X).Databases(X).(**Relations**

DAO.DBEngine.Workspaces(X).Databases(X).Relations(X).Fields

DAO.DBEngine.Workspaces(X).Databases(X).(**Recordsets**

DAO.DBEngine.Workspaces(X).Databases(X).Recordsets(X).Fields

DAO.DBEngine.Workspaces(X).Databases(X).(**Containers**

DAO.DBEngine.Workspaces(X).Databases(X).Containers(X).Documents

DAO.DBEngine.**Errors**

3. ספריית הפונקציות של ויזואל בייסיק

כאן נכללים כל הפונקציות המוכרות של שפת הויזואל בייסיק ליישומים.

VBA - VBA.Constants (vb...)

VBA.FileSystem

VBA.Conversion

VBA.Financial

VBA.DateTime

VBA.Strings

VBA.ColorConstants (vb)

הערה: בכל מקום שכתוב "x" מדובר באינדקס של אובייקטים שמסמן את מספר האובייקט

באוספים מצויים השיטות הבאות: Count, Append, Delete, Refresh,

אפי ברק © כל הזכויות שמורות - לשימוש אישי ולא מסחרי

2.5. מבנה של אובייקט אוסף

תרשים של אובייקטים, מאוסף הגדרת הטבלאות ועד לשדות. השורה הראשונה שמודגשת היא אובייקט אוסף הטבלאות. מספר הטבלאות שמצויות באוסף (כולל טבלאות מערכת) הינו 37 פריטו הינו טבלה מספר אחד, כאשר תכונת הAttributes שווה ל0 מציינת שמדובר בטבלה רגילה. תכונת הConnect שהינה ריקה מציינת שמדובר בטבלת רגילה, ולא בטבלה מחוברת למסד אחר. אנו רואים שאובייקט הגדרת הטבלה – TableDef – מכיל בתוכו שלוש אוספים אוסף השדות- Fields אוסף האינדקסים - Indexes ואוסף התכונות – Properties כאשר לכל אחד מהם יש את התכונות שלו ואת הפריטים שלו.

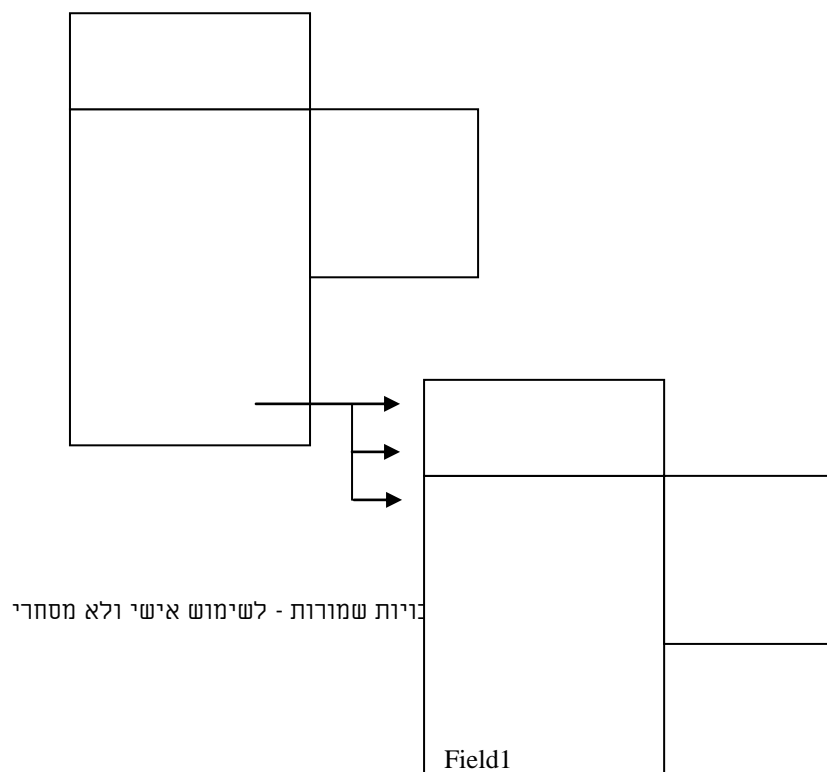
Watches		
Expression	Value	Type
CurrentDb.TableDefs		Object/TableDefs
Count	37	Integer
Item 1		Variant/Object/TableDef
Attributes	0	Long
ConflictTable	""	String
Connect	""	String
DateCreated	#15/05/2000 21:01:01#	Variant/Date
Fields		Fields/Fields
Count	6	Integer
Item 1		Variant/Object/Field
Item 2		Variant/Object/Field
Item 3		Variant/Object/Field
Item 4		Variant/Object/Field
Item 5		Variant/Object/Field
Item 6		Variant/Object/Field
Indexes		Indexes/Indexes
LastUpdated	#28/11/2000 14:56:28#	Variant/Date
Name	"Arazot"	String
Properties		Properties/Properties
RecordCount	61	Long
ReplicaFilter	<The replica is not a partial replica	Variant
SourceTableName	""	String
Updatable	True	Boolean
ValidationRule	""	String
ValidationText	""	String
Item 2		Variant/Object/TableDef
Item 3		Variant/Object/TableDef
Item 4		Variant/Object/TableDef
Item 5		Variant/Object/TableDef

אובייקט השדות

אנו רואים שאובייקט האוסף שדות, מכיל בתוכו פריטים הממוספרים מ 1 ואלך, כאשר סך השדות שבטלבה הנוכחית הם 6. כל פריט כזה הוא שדה, והוא פריט שיש לו תכונות שונות בעלי ערכים שונים, וכמו כן יש בו אוסף של תכונות, שאפשר לעבור עליהם באופן סדרתי.

Expression	Value	Type
68 CurrentDb.TableDefs		Object/TableDefs
Count	37	Integer
Item 1		Variant/Object/TableDef
Attributes	0	Long
ConflictTable	""	String
Connect	""	String
DateCreated	#15/05/2000 21:01:01#	Variant/Date
Fields		Fields/Fields
Count	6	Integer
Item 1		Variant/Object/Field
AllowZeroLength	False	Boolean
Attributes	1	Long
CollatingOrder	1033	Long
CollectionIndex	-8148	Integer
DataUpdatable	False	Boolean
DefaultValue	"0"	Variant/String
FieldSize	<Property can be set o	Long
ForeignName	<Invalid operation.>	String
Name	"t-kod-erez"	String
OrdinalPosition	0	Integer
OriginalValue	<Operation is not supp	Variant
Properties		Properties/Properties
Count	26	Integer
Item 1		Variant/Object/Property
Item 2		Variant/Object/Property
Item 3		Variant/Object/Property
Item 4		Variant/Object/Property
Item 5		Variant/Object/Property
Item 6		Variant/Object/Property

בדרך כלל לאובייקט אוסף, יהיו שיטות להוספת פריט באוסף, למחיקת פריט באוסף, לדענון האוסף, וכן תכונה שבה יצויין מספר הפריטים שבאוסף.



Application 3.5

זהו האובייקט הראשי של התוכנה דרכה אפשר לגשת ולהגדיר את האובייקטים בתוכנה, אמנם לרוב אפשר בהפניה מרומזת.
אובייקט האפליקציה Application שמכיל בתוכו שורה של תכונות ושיטות הקשורות ליישום בין התכונות הבולטות

Assistant	אובייקט עזר האופיס
DBEngine	אובייקט מנוע בסיס הנתונים
DoCmd	אובייקט פקודות המקרו
Forms	אובייקט הטפסים הפתוחים
Reports	אובייקט הדוחות הפתוחים
Modules	אובייקט המודולים
References	אובייקט ההפניות
MenuBar	תכונת התפריט
Parent	תכונת ההורה

בין השיטות הבולטות

CreateForm	צור טופס
CreateReport	צור דוח
CreateControl	צור פקד
DeleteControl	מחק פקד
CurrentDb	הצבע לבסיס נתונים נוכחי
CurrentUser	החזר שם משתמש נוכחי
Dsum, Dmax, Dfirst , Dlookup,	פונקציות צבירת תחום
DDEExecute	פונקציות לתקשורת בין יישומים
BuildCriteria	בנה ביטוי
AddToFavorites	הוסף למועדפים
NewCurrentDatabase	צור בסיס נתונים חדש כנוכחי
OpenCurrentDatabase	פתח בסיס נתונים כנוכחי
CloseCurrentDatabase	סגור בסיס נתונים נוכחי
Quit	סגור יישום
SetOption	שינוי האופציות של התוכנה
GetOption	קבלת הקביעות של התוכנה
LoadPicture	טען תמונה לטופס או פקד
RunCommand	הרץ פקודות תפריטים
Run	הרץ שגרה או פונקציה מישום אחר

לכל התכונות והשיטות אפשר להתייחס עם הקידומת Application או בלעדיה.

Application.MenuBar Application.CurrentObjectName Application.CurrentObjectType Application.Quit A_SAVE	תפריט התוכנה שם אובייקט נוכחי סוג אובייקט נוכחי יציאה מהתוכנה כולה לחלונות
--	---

--	--

DBEngine .4.5

אובייקט חשוב נוסף הינו אובייקט מנוע בסיס הנתונים DBEngine שמכיל בתוכו שורה של תכונות ושיטות הקשורות לקובץ בסיס הנתונים בין התכונות הבולטות

DefaultPassword	סיסמת ברירת מחדל
IniPath	נתיב של קובץ מנוע
Version	מספר גרסה של המנוע
SystemDB	נתיב של קובץ קבוצות עבודה

בין השיטות הבולטות

BeginTrans	התחלת טרנזקציה
CommitTrans	אישור כל העדכונים
Rollback	ביטול כל העדכונים
CompactDatabase	דחיסת מסד נתונים
RepairDatabase	תיקון מסד נתונים
OpenConnection	פתיחת חיבור למסד
CreateWorkspace	יצירת קבוצת עבודה חדשה
RegisterDatabase	רישום בסיס נתונים לעבודה עם ODBC

אובייקט חשוב נוסף הינו אובייקט מנוע בסיס הנתונים Workspace שמכיל בתוכו שורה של תכונות ושיטות הקשורות לאובייקטים של בסיס הנתונים בין התכונות הבולטות

Name	שם מרחב העבודה
UserName	שם המשתמש
Connections	מחרוזת החיבור

בין השיטות הבולטות

CreateGroup	צור קבוצה
CreateUser	צור משתמש
Close	סגור מרחב עבודה

גישה אל פריט באוסף

גישה אל פריט באובייקט אוסף יכולה להיות דרך מספרו או דרך שמו אם אנו רוצים לשלוף את גודלו של השדה הראשון בטבלה הראשונה של המסד אנו יכולים לעשות כן באחת משתי השיטות

`MsgBox DAO.DBEngine.Workspaces(0).Databases(0).TableDefs(0).Fields(0).Size`

`MsgBox DAO.DBEngine.Workspaces(0).Databases(0).TableDefs(0).Fields("acm").Size`

בכל מקום שכתוב 0 הכונה היא הפניה לאובייקט הראשון, בחלק מהאובייקטים, אובייקט 0 הינו האובייקט הנוכחי, כמו מרחב העבודה ובסיס הנתונים.

באותו האופן אנחנו יכולים לשנות תכונה למשל אם אנו רוצים לשנות את שם השדה הראשון בטבלת אלפון לKOOKI אנו רק צריכים לרשום את שורת הפקודה הבאה:

```
DAO.DBEngine.Workspaces(0).Databases(0).TableDefs.("אלפון")Fields(0).Name  
= "kooki"
```

Screen .5.5

זהו אובייקט המסך של האקסט, דרכו אפשר לגשת לאובייקטים פעילים במסך, כמו טופס/דוח פעיל, פקד פעיל ועוד.

טופס נוכחי - ActiveForm

```
Dim MyForm As Form
Set MyForm = Screen.ActiveForm
msgbox Screen.ActiveForm.Caption
```

דוח נוכחי - ActiveReport

פקד נוכחי - ActiveControl

```
Dim MyControl As Control
Set MyControl = Screen.ActiveControl
If MyControl.Name = "Field1" Then
    ... ' Do this.
Elseif MyControl.Name = "Button2" Then
    ... ' Do this.
End If
```

פקד קודם לנוכחי - PreviousControl

```
Sub Button0_Click ()
    Dim PrevControl As Control
    Set PrevControl = Screen.PreviousControl
    If TypeOf PrevControl Is TextBox Then
        MsgBox "A text box just lost the focus."
    End If
End Sub
```

Debug - חלון ניפוי שגיאות

הדפסה בחלון - Debug.print

Debug.print mishtane(I)

Docmd - פקודות מקרו

Forms .6.5

זהו אוסף של כל הטפסים הפתוחים והטעונים בזכרון המחשב, אפשר לגשת מכאן אל כל הטפסים הפתוחים כרגע, ולבצע עליהם מניפולציות שונות.

גישה אל הטופס ופקדיו

Forms("אנשים").name	שם הטופס עצמו
Forms("אנשים")(3).name	שם הפקד השלישי בטופס

הדפסה של כל כותרות הטפסים הפתוחים

```
Sub ListForms ()
    Dim X As Integer
    NumberForms = forms.count ' Count number of forms.
    Debug.Print "There are"; NumberForms; "open forms:"
    For X = 0 To NumberForms - 1
        Debug.Print forms(X).Caption ' Print form name.
    Next X
End Sub
```

סגירת כל הטפסים הפתוחים

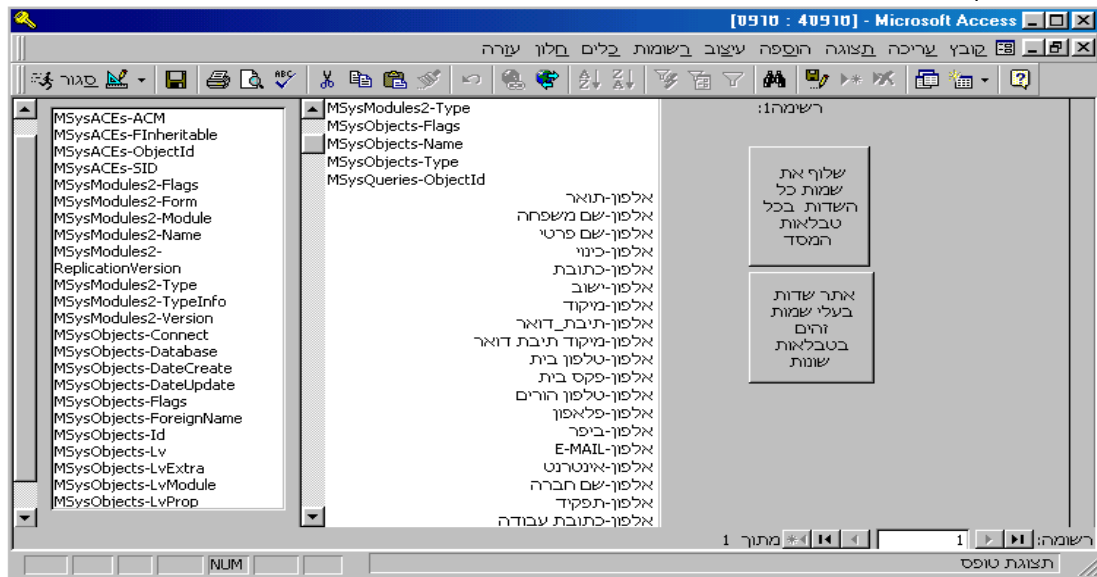
```
Sub closeAllOpenForms ()
    For i = forms.count - 1 To 0 Step -1
        DoCmd Close A_form, forms(i).name
    Next i
End Sub
```

הדפסת כל שמות הפקדים בטופס מסוים

```
Function printAllControls (theForm As String) As String
    Dim lfrm As Form, li As Integer
    Set lfrm = forms(theForm)
    For li = 0 To lfrm.count - 1
        Debug.Print lfrm(li).name
    Next li
    printAllControls = "מס' פקדים שהודפסו" & Str(lfrm.count - 1)
End Function
```


TableDefs .7.5

גישה אל שדות דרך קוד עומדת לפנינו משימה לשלוף את שמות כל השדות בכל טבלאות המסד ברשימה אחת, ולאחר מכן להשוות האם יש שמות זהים בטבלאות שונות.



הפתרון הוא ליצור מצביע לאוסף הטבלאות ולסרוק כל פעם את אוסף השדות בעזרת שתי לולאות

```
Private Sub Click()
    Dim l_Table As TableDef, l_field As Field, str As String
    For Each l_Table In DAO.DBEngine.Workspaces(0).Databases(0).TableDefs
        For Each l_field In l_Table.Fields
            str = str & l_Table.Name & "-" & l_field.Name & vbCrLf
        Next
    Next
    Me = שדות.str
End Sub
```

הפתרון הוא כמו בשאלה הקודמת, רק שכאן צריך ליצור עוד שתי לולאות בכדי לחפש מול כל שדה את כל השמות של כל השדות.

```
Private Sub Click()
    Dim l_Table As TableDef, k_Table As TableDef, l_field As Field, K_field As Field
    Dim str, l_fieldName As String
    For Each l_Table In DAO.DBEngine.Workspaces(0).Databases(0).TableDefs
        For Each l_field In l_Table.Fields
            For Each k_Table In DAO.DBEngine.Workspaces(0).Databases(0).TableDefs
                For Each K_field In k_Table.Fields
                    If l_field.Name = K_field.Name And l_Table.Name <> k_Table.Name Then
                        str = str & l_Table.Name & "-" & l_field.Name & ";" &
                    End If
                Next K_field
            Next k_Table
        Next l_field
    Next l_Table
    Me.RowSource = שדות1.str
End Sub
```

Containers .8.5מכולות Containers

מיכל האקסס הינו אוסף של תיעוד כל האובייקטים הקיימים במסד, בעזרתו אפשר לגשת לכל אובייקט קיים, ותכונה של כל אובייקט ולשנותם.

לעומת ה forms שנותן גישה רק לטפסים פתוחים, דרך האובייקט הזה אפשר לגשת אף לטפסים הסגורים.

אובייקט אוסף המכולות בנוי כך שיש בו אוסף של אובייקטים שונים ואלו הם

Databases
Forms
Modules
Relationships
Reports
Scripts
SysRel
Tables

Private Sub Click() רשימת_כל_מסמכי_התיעוד_שבמכולה

Dim dcm **As** Document, ctrLoop **As** Container
'מעבר סידרתי על כל אובייקטי המכולה

For Each ctrLoop **In** CurrentDb.Containers

 Debug.Print ctrLoop.Name

'מעבר סידרתי על כל מסמכי המכולה

For Each dcm **In** ctrLoop.Documents

 Debug.Print dcm.Name

Next

 Debug.Print

Next

End Sub

כל אחד מאובייקטים אלו מכיל Documents - אוסף מסמכי תיעוד של כל האובייקטים מהסוג שלו.

לכל אחד מהאובייקטים יש כמה תכונות, כמו שם האובייקט, תאריך היצירה, תאריך העדכון, שם הבעלים, שם המשתמש.

תרגיל להעלות לזכרון את כל הטפסים של מסד הנתונים

Private Sub Click() פתיחת_כל_הטפסים_של_המסד

Dim l_db **As** DATABASE, l_ctr **As** Container, dcm **As** Document

'יצירת מצביע לבסיס נתונים נוכחי

Set l_db = DBEngine.Workspaces(0).Databases(0)

'יצירת מצביע למכולה של הטפסים

Set l_ctr = l_db.Containers("forms")

'מעבר על כל אובייקטי המסמך של מכולת הטפסים

For Each dcm **In** l_ctr.Documents

'הדפסת תכונות של מסמך בודד בחלון מידי

 Debug.Print dcm.Name & "-" & dcm.Owner & "-" & dcm.LastUpdated

'פתיחת הטופס אל הזכרון

DoCmd.OpenForm dcm.Name

Next

'שחרור המצביעים

Set ctrLoop = **Nothing**

Set l_db = **Nothing**

End Sub

הערה: אי אפשר לגשת מאובייקט זה אל כל תכונתיו של אובייקט מסויים אלא אל מקצת התכונות שנועדו לתיעוד, כך שאם רוצים לשנות תכונותיו של טופס, אי אפשר לשנות אותם, כמו שמשנים את תכונות הגדרת הטבלה או השאילתא, אלא יש לפתוח אותו תחילה בזכרון ואז לשנותו.

אובייקטים שמכיל בתוכו המיכל

מוגדר ב	מכיל
Databases	מנוע בסיס נתונים JET
Tables	מנוע בסיס נתונים JET
Relationships	מנוע בסיס נתונים JET
Forms	אקסס טפסים שמורים
Reports	אקסס דוחות שמורים
Scripts	אקסס מקרו-ים שמורים
Modules	אקסס מודולים שמורים

פתיחת כל הטפסים בבסיס הנתונים הנוכחי

```

Sub openAllForms ()
    Dim mydb As Database, l_container As Container, l_xx As Integer
    DoCmd Hourglass True
    Set mydb = DBEngine.Workspaces(0).Databases(0)
    Set l_container = mydb.containers("Forms")
    For l_xx = 0 To l_container.Documents.count - 1
        MsgBox "פּוֹתֵחַ אֵת:" & Chr(10) & l_container.Documents(l_xx).name
        DoCmd OpenForm l_container.Documents(l_xx).name, A_DESIGN
    Next l_xx
    DoCmd Hourglass False
End Sub
    
```

הדפסת כל שמות האובייקטים

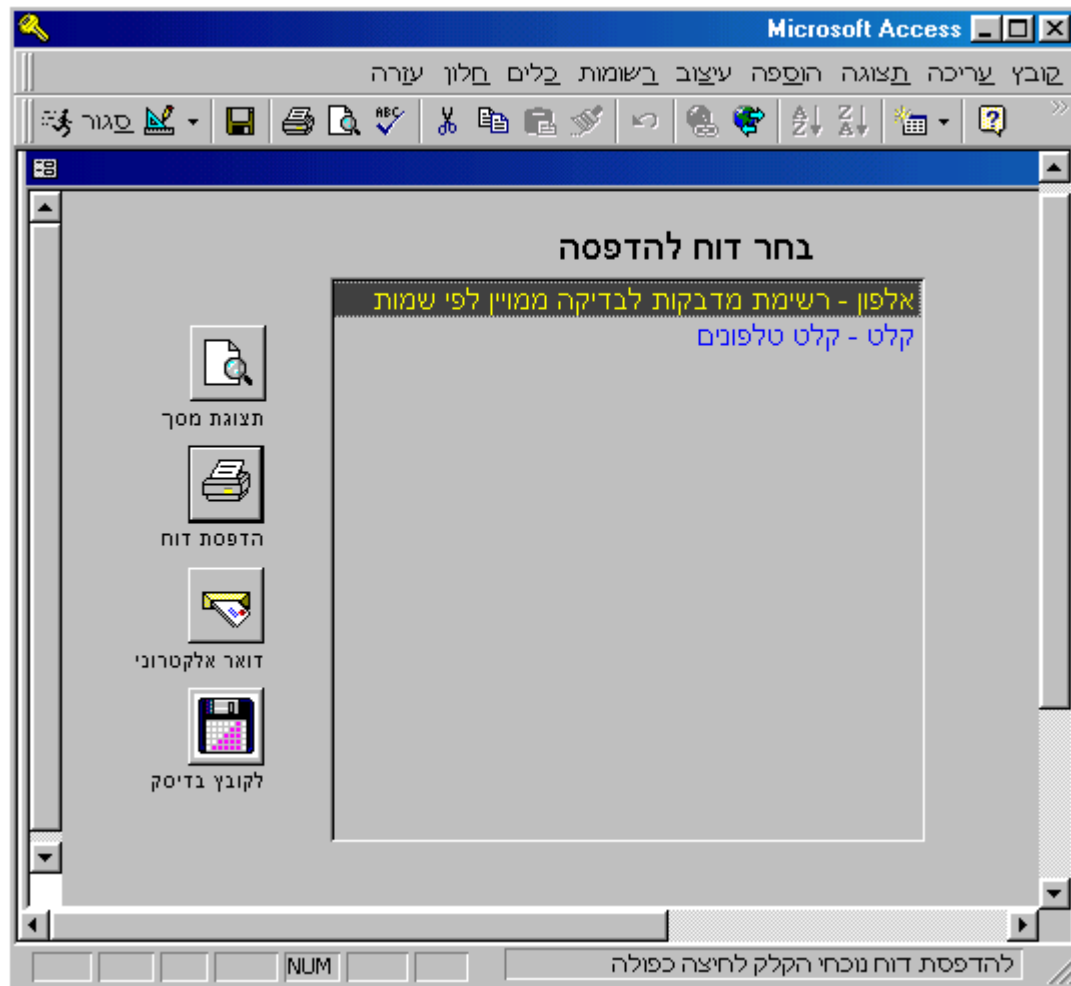
```

Function printAllObject ()
    Dim mydb As Database, l_container As Container, l_xx As Integer

    Set mydb = DBEngine.Workspaces(0).Databases(0)

    For i = 1 To 7
        Debug.Print Choose(i, "Tables", "Forms", "Reports", "Scripts", "Modules",
        "Databases", "Relationships")
        Set l_container = mydb.containers(Choose(i, "Tables", "Forms", "Reports",
        "Scripts", "Modules", "Databases", "Relationships"))
        For l_xx = 0 To l_container.Documents.count - 1
            Debug.Print l_container.Documents(l_xx).name
        Next l_xx
    Next i
End Function
    
```

המטרה היא ליצור טופס שבאופן אוטומטי יכיל את כל דוחות המסד, כך שהמשתמש יקבל רשימת דוחות קיימים ויוכל להדפיס כל דוח או לשמור לקובץ או בדואר אלקטרוני. כל זמן שלא נבחר דוח, הכפתורים צריכים להיות מעומעמים, בכדי שלא תתבצע הדפסת דוח.



ראשית יש לכתוב את הקוד שאוסף את שמות דוחות, באירוע בעת טעינת טופס

```
Private Sub Form_Load()
```

```
    Dim l_db As DATABASE, l_rowSource As String, Areport As Object
```

```
    Dim containerOfRrports As Container
```

```
    'יש ליצור מצביע אל בסיס הנתונים
```

```
    Set l_db = DBEngine.Workspaces(0).Databases(0)
```

```
    'יש ליצור מצביע אל המכולה של אוסף הדוחות
```

```
    Set containerOfRrports = l_db.Containers("Reports")
```

```
    'יש ליצור לולאה שתעבור על כל מסמכי הדוחות, ותשלוף את שמם
```

```
    For Each Areport In containerOfRrports.Documents
```

```
        l_rowSource = l_rowSource & Areport.Name";" &
```

```
    Next Areport
```

```
    'יש לקבוע את תכונות תיבת הרשימה
```

```
    Me!MyCombo.RowSource = l_rowSource
```

```
    Me!MyCombo.RowSourceType = "Value List"
```

```
    Me!MyCombo.ColumnCount = 1
```

```
    Me!MyCombo.ColumnWidth = "1"
```

אפי ברק © כל הזכויות שמורות - לשימוש אישי ולא מסחרי

```

Me!MyCombo.BoundColumn = 1
Me![MyCombo].Value"" =
'יש לשחרר מצביעים
l_db.Close
Set containerOfRrports = Nothing
'כל זמן שלא נבחר שום דוח, נעשה עמעום של הכפתורים של ההדפסה
changeStatusOfControls False

```

End Sub

```

Private Sub MyCombo_Click()
ברגע שנבחר דוח מבוטל העמעום
changeStatusOfControls True

```

End Sub

```

Public Function changeStatusOfControls(status As Boolean(
Me.[תצוגת מסך]!Enabled = status
Me.[הדפסת דוח]!Enabled = status
Me.[דואר אלקטרוני]!Enabled = status
Me.[יצא לקובץ]!Enabled = status

```

End Function

לאחר מכן יש לכתוב את הקוד לכל אחד מפקדי הכפתורים של הטופס

```

Private Sub מסך_תצוגת_Click()
DoCmd.OpenReport Me![MyCombo].Value, A_PREVIEW

```

End Sub

```

Private Sub הדפסת_דוח_Click()
DoCmd.OpenReport Me![MyCombo].Value, A_NORMAL

```

End Sub

```

Private Sub דואר_אלקטרוני_Click()
DoCmd.SendObject A_REPORT, Me![MyCombo].Value

```

End Sub

```

Private Sub יצא_לקובץ_Click()
DoCmd.OutputTo A_REPORT, Me![MyCombo].Value

```

End Sub

Recordset .9.5

האוסף "סדרת רשומות" הינו אובייקט מסוג טבלה, שמאפשר לנו לגשת לנתוני טבלה או לשכפל אותם, באופן שנוכל לעבוד איתם כאיברים במערך. בעזרת אובייקט זה, אנו יכולים לחפש נתונים בטבלאות, לעדכון, להוסיף ולמחוק רשומות, באופן סדרתי, ובשימוש בכל פקודות התכנות של VBA

תכונות ושיטות של האובייקט

<u>Properties</u>	<u>methods</u>
rd.AbsolutePosition rd.PercentPosition rd.Bookmarkable rd.Updatable rd.Restartable rd.NoMatch rd.EditMode rd.LockEdits rd.RecordCount rd.Type rd.DateCreated rd.LastModified rd.LastUpdated rd.Transactions rd.ValidationRule rd.ValidationText rd.Properties rd.Name	<p>שיטות הקשורות ליצירת סדרת רשומות</p> rd.OpenRecordset(Type:=, Options:=) RecordSetClone rd.CopyQueryDef rd.Clone rd.CacheSize rd.CacheStart rd.INDEX rd.Sort rd.Requery NewQueryDef:= rd.GetRows(cRows:=)
	<p>שיטות הקשורות לגישה את רשומות בסדרה</p> rd.Seek Comparison:=, Key1:= rd.FindFirst Criteria:= rd.FindLast Criteria:= rd.FindNext Criteria:= rd.FindPrevious Criteria:= rd.Move Rows:=, StartBookmark:= rd.MoveFirst rd.MoveLast rd.MoveNext rd.MovePrevious rd.BOF rd.EOF rd.Bookmark
	<p>שיטות הקשורות לעדכון רשומות</p> rd.Delete rd.Edit rd.UPDATE rd.AddNew rd.CancelUpdate
	rd.Close

בתחילה יש לפתוח אובייקט מסוג זה, ע"י שימוש בשיטת פתיחה או שיכפול, לאחר מכן יש לנוע ברשומות של האובייקט בכדי למצוא את הרשומה המתאימה, אפשר לשלוף נתונים מרשומה, וכן לעדכן רשומה, ולבסוף יש לסגור את האובייקט

דוגמאות לשימוש במשתנה אוסף רשומות

בדיקת רצף מספרים של קבלות בטבלה

Sub bdikatRetzef ()

Dim MyDB **As** Database, myRecSet **As** Recordset, l_sql, myMsg **As** String, i **As** Integer

Set MyDB = DBEngine.Workspaces(0).Databases(0)

l_sql = "SELECT [תרומות ירושלים].* FROM [תרומות ירושלים] ORDER BY [תרומות ירושלים].[מספר קבלה];"

myMsg = "חסרות הקבלות הבאות" & Chr(10)

Set myRecSet = MyDB.OpenRecordset(l_sql)

myRecSet.MoveFirst

For i = 1 **To** DCount("[תרומות ירושלים]", "[תרומות ירושלים] קבלה")

If myRecSet.[מספר קבלה] <> i **Then**

myMsg = myMsg & Chr(10) & Str(i)

i = i + 1

End If

myRecSet.MoveNext

Next

MsgBox myMsg, 24, "התרעה על קבלות חסרות"

myRecSet.Close

End Sub

החלפת שם משפחה בשם משפחה אחר במאה הרשומות הראשונות

Function UpdAlfon (SQLSTRING **As** String)

Dim MyDB **As** Database, myRecSet **As** Recordset, l_sql **As** String

Set MyDB = DBEngine.Workspaces(0).Databases(0)

l_sql = "SELECT * FROM אלפון";"

Set myRecSet = MyDB.OpenRecordset(l_sql)

myRecSet.MoveFirst

For i = 1 **To** 100

If myRecSet.[שם משפחה] = "אבוטבול" **Then**

myRecSet.Edit

myRecSet.[שם משפחה] = "אביטל"

myRecSet.Update

End If

myRecSet.MoveNext

Next i

myRecSet.Close

End Function

החלפת כל המופעים של שמות משפחה בטבלה
נכתב ע"י יוסי כהן

Function updalfon()

Dim mydb **As** DATABASE, myrecset **As** Recordset

Set mydb = DBEngine.Workspaces(0).Databases(0)

Set myrecset = mydb.OpenRecordset("SELECT [שם משפחה] FROM אלפון;")

myrecset.FindFirst "[שם משפחה]='לוי'"

Do Until myrecset.NoMatch

If myrecset. "[שם משפחה]='לוי'" **Then**

myrecset.Edit

myrecset. "[שם משפחה]='כהן'"

myrecset.UPDATE

End If

myrecset.FindNext "[שם משפחה]='לוי'"

Loop

myrecset.Close

End Function

מחיקת כל רשומה שלישית בטבלה
נכתב ע"י יוסי כהן

Function DeleteThird()

Dim mydb **As** DATABASE, myrecset **As** Recordset, i **As** Integer

Set mydb = DBEngine.Workspaces(0).Databases(0)

Set myrecset = mydb.OpenRecordset("SELECT [שם משפחה] FROM אלפון;")

i = 1

myrecset.MoveFirst

Do Until myrecset.EOF

If i Mod 3 = 0 **Then** myrecset.Delete

i = i + 1

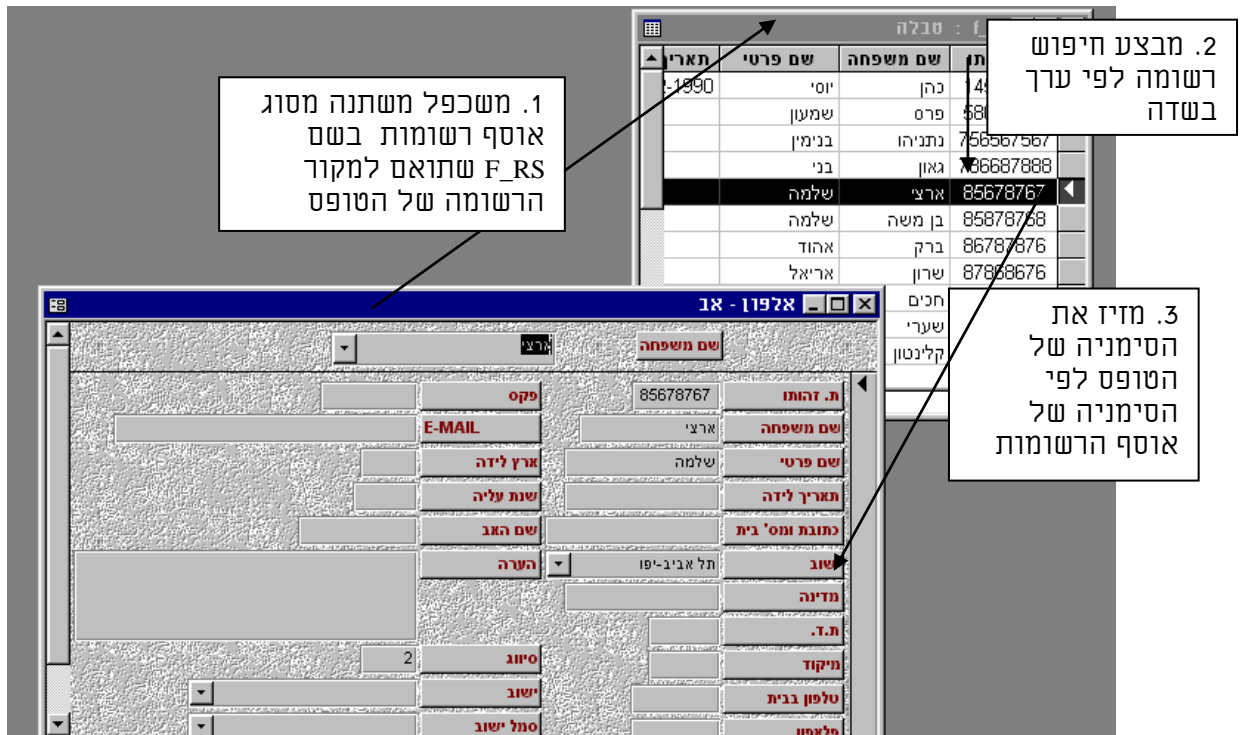
myrecset.MoveNext

Loop

myrecset.Close

End Function

איתור רשומה בטופס בעזרת סדרת רשומות



```

Function findReqRecord ()
Dim l_findstring As String, f_rs As Recordset
Set f_rs = Screen.ActiveForm.RecordSetClone
l_findstring = "[אלפון].[אינדקס אלפון]" & Screen.ActiveControl
f_rs.FindFirst l_findstring
Screen.ActiveForm.BookMark = f_rs.BookMark
f_rs.Close
End Function
    
```

שליפת תאריך יצירת טבלה וזמן עדכון אחרון

```

Dim Creation, DesignChange As Variant, MyTable As Recordset, MyDB As Database
Set MyDB = DBEngine.Workspaces(0).Databases(0)
Set MyTable = MyDB.OpenRecordset("Employees", DB_OPEN_TABLE)
Creation = MyTable.DateCreated
DesignChange = MyTable.LastUpdated
MyTable.Close
    
```

פתיחת אוסף רשומות עפ"י טבלה

```

Function EnumerateRecordset () As Integer
Dim mydb As Database, MyData, TempRecordset As Recordset
Dim i, J As Integer
Set mydb = DBEngine.Workspaces(0).Databases(0)
    
```

```

Set MyData = mydb.OpenRecordset("Orders", DB_OPEN_SNAPSHOT)
Debug.Print
' Enumerate all Recordset objects.
For J = 0 To mydb.Recordsets.count - 1
    Set TempRecordset = mydb.Recordsets(J)
    Debug.Print
    Debug.Print "Enumeration of Recordset objects("; J; "): "; TempRecordset.name
    Debug.Print
    ' Enumerate fields.
    Debug.Print "Fields: Name, Type, Value"
    For i = 0 To TempRecordset.Fields.count - 1
        Debug.Print " "; TempRecordset.Fields(i).name
        Debug.Print ", "; TempRecordset.Fields(i).Type
        Debug.Print ", "; TempRecordset.Fields(i).Value
    Next i
Next J
MyData.Close
EnumerateRecordset = True
End Function

```

הוספת רשומות חדשות מטבלה קיימת לטבלה חדשה

נכתב ע"י רוזי בדיחי

```

Sub CopyTable()
Dim db As Database, rstFrom, rstTo As Recordset
Set db = CurrentDb()
Set rstFrom = db.OpenRecordset("AddressBook")
Set rstTo = db.OpenRecordset("Addresses")
Do Until rstFrom.EOF
    With rstTo
        .AddNew
        !LastName = rstFrom.Last_Name
        !FirstName = rstFrom.First_Name
        !Address = rstFrom.Address
        !City = rstFrom.City
        !StateOrProvince = rstFrom.State
        !PostalCode = rstFrom.Zip_Code
        !HomePhone = rstFrom.Home_Phone_No
        !WorkPhone = rstFrom.Work_Phone_no
        !EmailAddress = rstFrom.Email_address
        !Birthdate = rstFrom.Birthday
        !FaxNumber = rstFrom.Fax_Number
        !MobilePhone = rstFrom.Mobile_Phone
        .Update
    End With
    rstFrom.MoveNext
Loop
rstFrom.Close
rstTo.Close
End Sub

```

מספור רץ של שדה אחד לפי ערך משתנה בשדה אחר

המטרה היא למספר שדה במספור רץ אשר מתחיל כל פעם שערך בשדה אחר משתנה בדוגמא יש לנו מצביעים שנמצאים בקלפיות שונים ובכל קלפי יש ליצור מספר רציף למצביעים.

```
Dim db As Database, rs As Recordset
Dim index As Integer, Ballot As Integer
Set db = CurrentDb
Set rs = db.OpenRecordset("Voters1")
rs.MoveFirst
index = 1
Ballot = rs!Ballot
```

יש ליצור לולאה שתעבור על כל הרשומות

```
Do Until rs.EOF
    rs.Edit
    rs!kalpiId = index
    rs.Update
    rs.MoveNext
    If rs.EOF Then Exit Do
    במקרה ששדה "קלפי" שונה, יש להתחיל את המספור מחדש אחרת יש להוסיף 1
    If Ballot = rs!Ballot Then
        index = index + 1
    Else
        index = 1
    End If
    Ballot = rs!Ballot
Loop
rs.Close
```

שליפה מכספומט

יש לנו טבלת לקוחות שיש בהם מספרי כרטיס וסיסמה. אנו צריכים לבדוק האם מספר הכרטיס שהוקלד כסימולציה להכנסת כרטיס כספומט הינו תקין, ואז לאפשר למשתמש להקליד את הקוד הסודי שלו, ואם ישנה התאמה בין מספר הכרטיס לקוד הסודי, נאפשר לו להקליד את סכום המשיכה. בכדי לפתור את הבעיה יש לעבוד עם אובייקט אוסף רשומות.

אנו נראה כמה דוגמאות של פתיחת אובייקט אוסף רשומות ואיתור נתונים בו, ולבסוף את הפתרון בשלמותו.

פתרון א.

בפתרון זה אנו מפעילים את השיטה FindFirst לחיפוש של רשומה תואמת

```
Public Function findKod1()
Dim mydb As DATABASE, kaspomatRs As Recordset
Set mydb = CurrentDb
Set kaspomatRs = mydb.OpenRecordset("kaspomat", DB_OPEN_DYNASET)
kaspomatRs.MoveFirst

kaspomatRs.FindFirst "[kaspomatNum]= " & str$(inputNum(

If kaspomatRs.NoMatch Then
inputNumMsg.Caption="כרטיס לא תקין"
Else
inputNumMsg.Caption="אנא הקלד קוד סודי"
End If
kaspomatRs.Close
mydb.Close
End Function
```

פתרון ב.

בפתרון זה אנחנו עוברים בלולאה על כל הרשומות באמצעות שיטת MoveNext, עד אשר אנחנו מאתרים את הרשומה או עד שאנו מגיעים לסוף הרשומות

```
Public Function findKod2()
Dim mydb As DATABASE, kaspomatRs As Recordset
Set mydb = CurrentDb
Set kaspomatRs = mydb.OpenRecordset("kaspomat", DB_OPEN_DYNASET)
kaspomatRs.MoveFirst

Do Until kaspomatRs.EOF Or kaspomatRs!kaspomatNum = Val(inputNum)
kaspomatRs.MoveNext
Loop

If kaspomatRs.EOF Then
inputNumMsg.Caption="כרטיס לא תקין"
Else
inputNumMsg.Caption="אנא הקלד קוד סודי"
End If
kaspomatRs.Close
mydb.Close
```

End Function

פתרון ג.

בפתרון זה אנחנו יוצרים אוסף רשומות "מפולטר" מאוסף רשומות קיים כך, שאם המספר קיים, אנו אמורים לקבל רשומה אחת ואם לא לא נקבל רשומות בכלל באוסף הרשומות החדש.

```

Public Function findKod3()
Dim mydb As DATABASE, kaspomatRs As Recordset, kaspomatFilter As Recordset
Set mydb = CurrentDb
Set kaspomatRs = mydb.OpenRecordset("kaspomat", DB_OPEN_DYNASET)

kaspomatRs.Filter = "[kaspomatNum]= " & str$(inputNum)

Set kaspomatFilter = kaspomatRs.OpenRecordset

If kaspomatFilter.RecordCount = 0 Then
    inputNumMsg.Caption="כרטיס לא תקין"
Else
    inputNumMsg.Caption="אנא הקלד קוד סודי"
End If
kaspomatRs.Close
mydb.Close
End Function

```

פתרון ד.

בפתרון זה אנחנו פותחים את אוסף הרשומות בשאילתא שכבר חותכת את הנתונים, שתואמים כמו במקרה הקודם.

```

Public Function findkod4()
Dim mydb As DATABASE, kaspomatRs As Recordset
Dim sqlString As String
Set mydb = CurrentDb
sqlString = "SELECT kaspomat.* FROM kaspomat WHERE kaspomatNum=" &
inputNum

Set kaspomatRs = mydb.OpenRecordset(sqlString, DB_OPEN_SNAPSHOT)

If kaspomatRs.RecordCount = 0 Then
    inputNumMsg.Caption= "כרטיס לא תקין"
Else
    inputNumMsg.Caption= "אנא הקלד קוד סודי"
End If
kaspomatRs.Close
mydb.Close

End Function

```

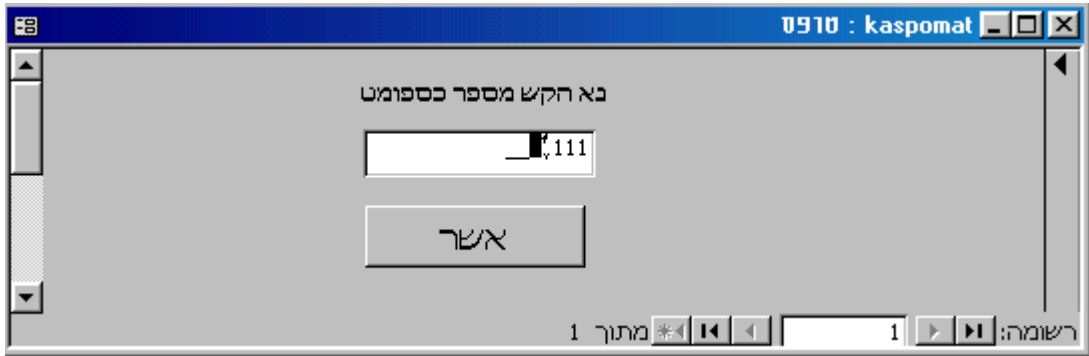
הפתרון בשלמותו

ניצור טופס ובו שלושה פקדים

פקד תווית בשם: inputNumMsg שערך ברירת המחדל שלו יהיה "נא הקש מספר כספומט"

פקד תיבת טקסט בשם: inputNum

ולפקד לחצן פקודה בשם: כספומט



ללחצן הפקודה תהיה שגרת אירוע בעת הקלקה, שתבצע את הקוד הבא:

```
Private Sub כספומט_Click()
Select Case inputNumMsg.Caption
Case: "נא הקש מספר כספומט", "כרטיס לא תקין"
    If findKod() Then
        inputNumMsg.Caption="אנא הקלד קוד סודי"
        kod = inputNum
        inputNum.InputMask = "password"
    Else
        inputNumMsg.Caption="כרטיס לא תקין"
    End If
Case: "אנא הקלד קוד סודי", "ססמא שגויה"
    If findPsw() Then
        inputNumMsg.Caption="הקלד סכום משיכה"
        inputNum.InputMask = "99999999"
    Else
        inputNumMsg.Caption="ססמא שגויה"
    End If
Case: "הקלד סכום משיכה"
    MsgBox "זהו זה?"
End Select
inputNum = Null
inputNum.SetFocus
End Sub
```

הפונקציה בודקת האם יש מספר כרטיס כמו שהוקש, ומחזירה אמת ושקר בהתאמה

```
Public Function findKod() As Boolean
Dim mydb As DATABASE, kaspomatRs As Recordset
Set mydb = CurrentDb
Set kaspomatRs = mydb.OpenRecordset("kaspomat", DB_OPEN_DYNASET)
kaspomatRs.MoveFirst

kaspomatRs.FindFirst "[kaspomatNum]= " & str$(inputNum)

findKod = Not kaspomatRs.NoMatch
```

```
kaspmatRs.Close  
mydb.Close  
End Function
```

הפונקציה בודקת האם הססמה שהוקשה אכן תואמת לבעל הכרטיס, ומחזירה אמת ושקר בהתאמה.

```
Public Function findPsw()  
Dim mydb As DATABASE, kaspmatRs As Recordset  
Dim sqlString As String  
Set mydb = CurrentDb  
sqlString = "SELECT kaspmat.* FROM kaspmat WHERE kaspmatNum=" & kod  
  
Set kaspmatRs = mydb.OpenRecordset(sqlString, DB_OPEN_SNAPSHOT)  
  
findPsw = kaspmatRs("psw") = Val(inputNum)  
kaspmatRs.Close  
mydb.Close  
End Function
```


תרגילים באובייקטים

- א. יש ליצור פונקציה שתבדוק האם טבלה מסוימת קיימת במסד הנתונים הנוכחי או כל מסד אחר. במקרה שהטבלה קיימת היא תחזיר אמת ובמקרה שלא היא תחזיר שקר.
- ב. יש ליצור פונקציה שתבדוק האם טופס מסוים פתוח כרגע במסד.
- ג. יש ליצור שגרה שבודקת איזה משאילתות המסד משומשות בידי טפסים ודוחות.
- ד. יש ליצור שגרה שפותחת את כל הטפסים במבט עיצוב, משנה את צבע הרקע, (או תכונה אחרת) וסוגרת אותם.
- ה. יש ליצור טופס ובו תיבת רשימה שמציגה את כל הטבלאות במסד, תיבת רשימה נוספת שמציגה את כל השדות שבטבלה הנבחרת, פקד תיבת טקסט שאפשר להקליד בו מילה, וכפתור שגורם לכלך שהמילה נחפשת באותו שדה.
- ז. יש ליצור שגרה שיוצרת תיעוד לגבי הטבלאות של המסד, ורושמת בטבלה את כל השדות של כל הטבלאות עם פירוט של תכונות כמו סוג, גודל, והאם שדה אינדקס.
- ח. יש ליצור טופס שבונה פונקציות תחום של SQL, ובו אפשרות לבחור טבלה, לבחור שדה, לבחור פונקציה, ולבחור ביטוי חתך של הפונקציה.
- ט. יש ליצור שגרה שעושה אופטימיזציה לשדות, ולאחר שהיא ובודקת וסורקת ומוצאת את אורך הרשומה הארוכה ביותר בכל שדה היא משנה את גודל השדה בהתאמה.
- י. יש לכתוב שגרה שמצמידה את כל הפקדים המסומנים, בטופס או בדוח הנוכחי אחד לשני.

אוספים דינמיים (אוספי זמן ריצה)

ביכולתנו ליצור אוספים בזמן ריצה, ע"י שיוך פקדים מסוגים שונים ואף משתנים לאוסף חדש שאנחנו יוצרים. אנחנו יכולים ליצור רשימות דינמיות, של רשומות משתנים במקום הקצאה דינמית.

לאובייקט אוסף יש את התכונות הבאות:

תחביר	תכונה או שיטה
Count	מספר הפריטים שבאובייקט
Item(0)	גישה אל פריט באובייקט
Add	הוספת פריט חדש
Remove (0)	השמטת פריט קיים

בשיטה זו אנו יצרנו מראש סדרת פקדים מסוגים שונים בטופס, יצרנו בקוד אוסף חדש ששמו **myCollection** ושייכנו אליו פקדים ואף משתנים בעזרת שיטת Add. בדוגמא אנחנו עושים השמה של כל הפקדים לתוך תיבת טקסט Text3, ולאחר שאנו מסירים את המשתנה **strFirstname** מתוך אוסף הפקדים אנחנו יכולים לשנות את צבע הרקע שלהם.

```
Dim myCollection As New Collection
```

```
Dim strFirstname As String
```

```
Private Sub Form_Load()
```

```
Dim ctr As Variant
```

```
strFirstname = "stam"
```

```
myCollection.Add Text1
```

```
myCollection.Add Text2
```

```
myCollection.Add Label1
```

```
myCollection.Add Check1
```

```
myCollection.Add Option1
```

```
myCollection.Add strFirstname
```

```
For Each ctr In myCollection
```

```
Text3 = Text3 & vbCrLf & ctr
```

```
Next ctr
```

```
For Index = 1 To myCollection.Count
```

```
Text4 = Text4 & vbCrLf & myCollection.Item(Index)
```

```
Next Index
```

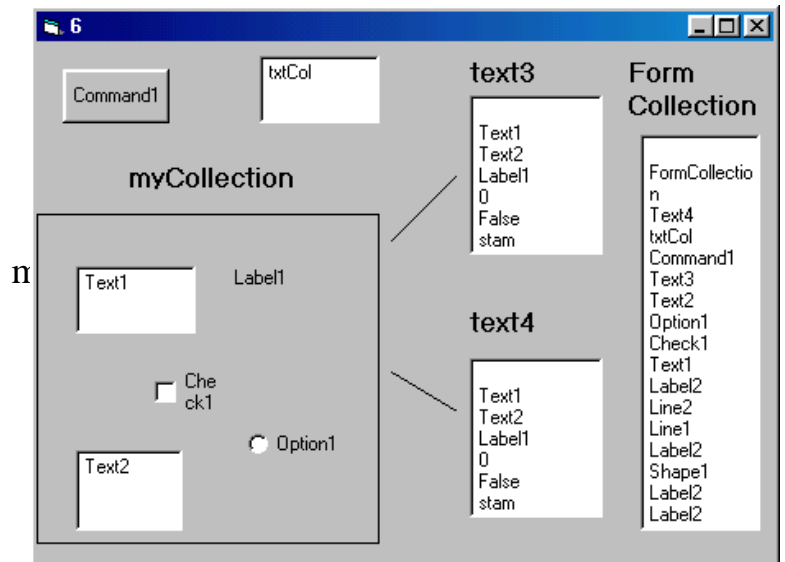
```
Me.Caption = myCollection.Count
```

```
For Each ctr In Me
```

```
FormCollection = FormCollection & vbCrLf & ctr.Name
```

```
Next ctr
```

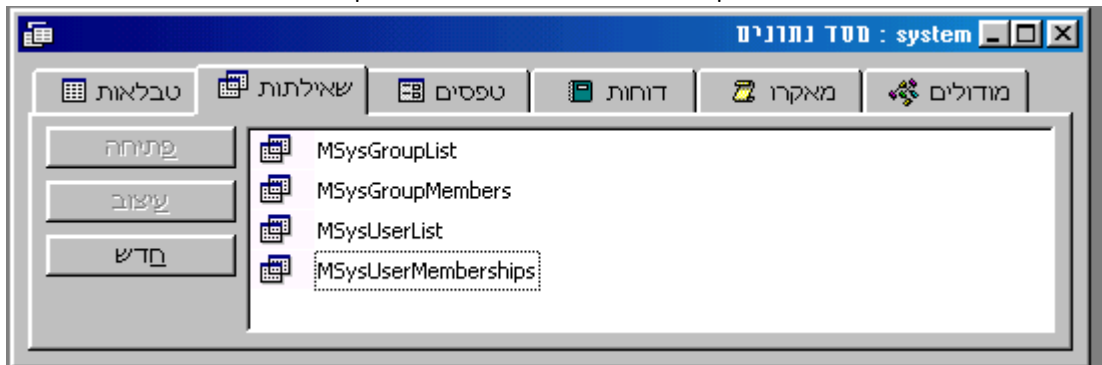
```
End Sub
```



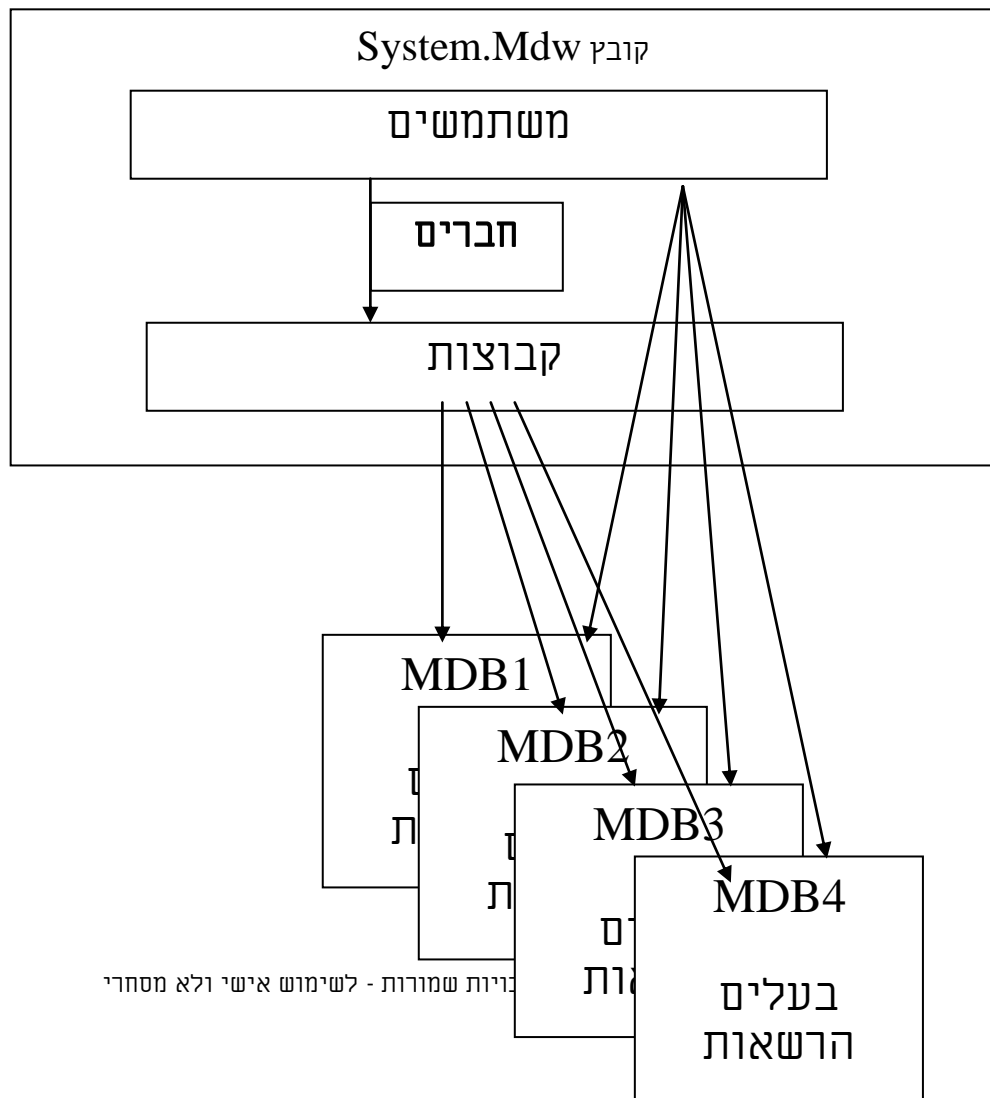
6. סיסמאות

1.6 קובץ קבוצות העבודה - System.Mdw

קובץ **System.Mdw** הינו בעיקרון קובץ בפורמט של קובץ Mdb לכל דבר, ואפשר לפתחו באמצעות האקסס. יחודו של הקובץ שמערכת האקסס, חייבת להיות מחוברת לקובץ מסוג כזה, אחרת אי אפשר להריץ אותה כלל. ישנה תוכנה מיוחדת שיוצרת את הקובץ הזה. בקובץ הזה ישנה הגדרה של כל שמות המשתמשים, ססמאות שלהם, הגדרה של כל שמות הקבוצות, ורשימה של משתמשים שחברים בכל קבוצה. לעומת זאת בקבצי בסיס הנתונים הרגילים יהיו רשימת הבעלים של כל אובייקט, ורשימת ההרשאות של כל משתמש וכל קבוצה לגבי כל אחד מהאובייקטים ולגבי מסד הנתונים גופו



שאלות בקובץ System.mdw



2.6 . אבטחת מסד נתונים

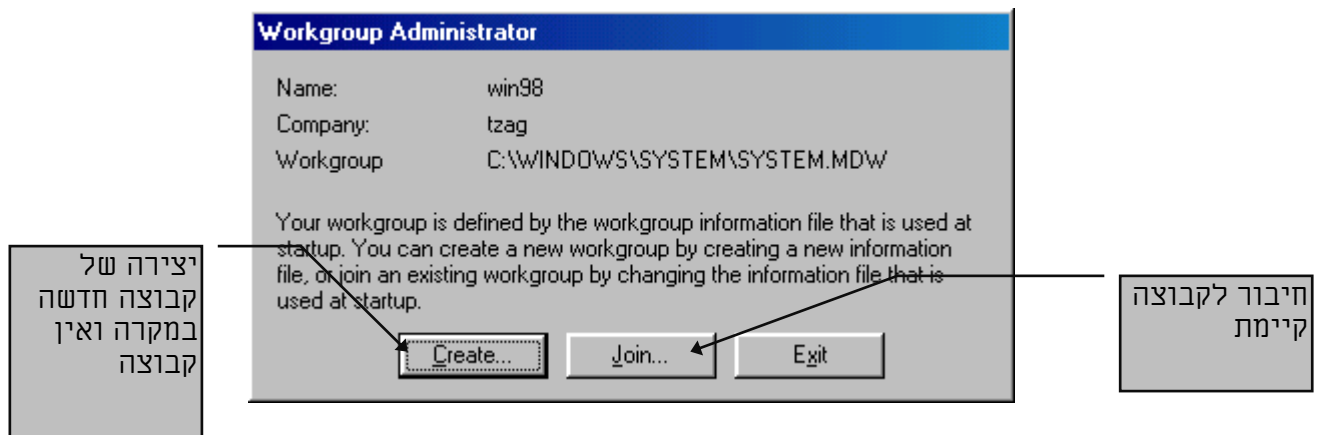
הקדמה:

לעתים רצוי לאבטח מסד נתונים. כאשר מסד הנתונים מאובטח, אפשר לקבוע מה יכולים משתמשים וקבוצות משתמשים לעשות עם אובייקטים של מסדי נתונים. בנוסף לכך, בד"כ נרצה להגן על זכויות היוצרים של היישום שלנו, את שני הדברים מיישמים בעזרת מנגנון האבטחה.

מערכת האבטחה של Microsoft Access מורכבת ממספר חלקים שיש להגדיר: **חשבונות קבוצות עבודה, משתמשים וקבוצות, וכן בעלויות, והרשאות.**

1. יש ליצור או לבחור קבוצת עבודה שבה יעשה השימוש במסד הנתונים. ע"י הרצת התוכנה WRKGADM.EXE

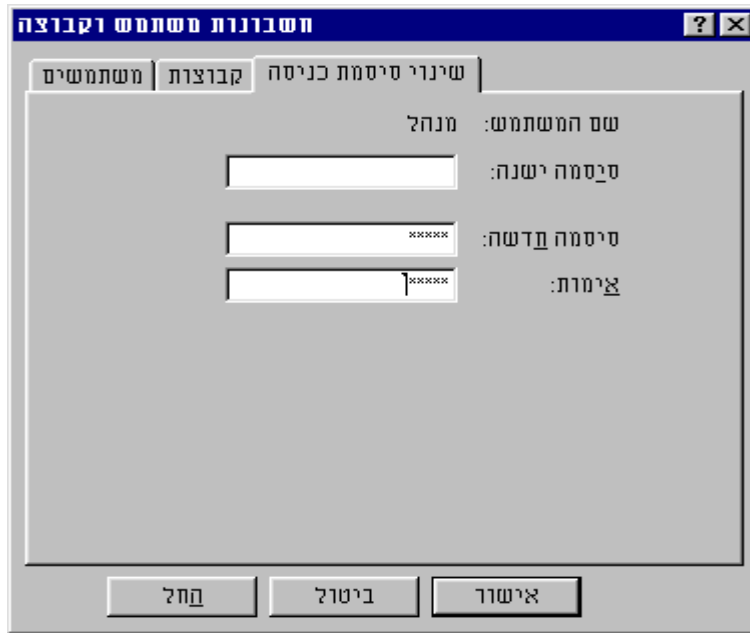
קבוצת עבודה זו מוגדרת על ידי מסד נתוני המערכת (בדרך כלל, הקובץ SYSTEM.MDW) שבו משתמשת תוכנת Microsoft Access בעת אתחול התוכנה.



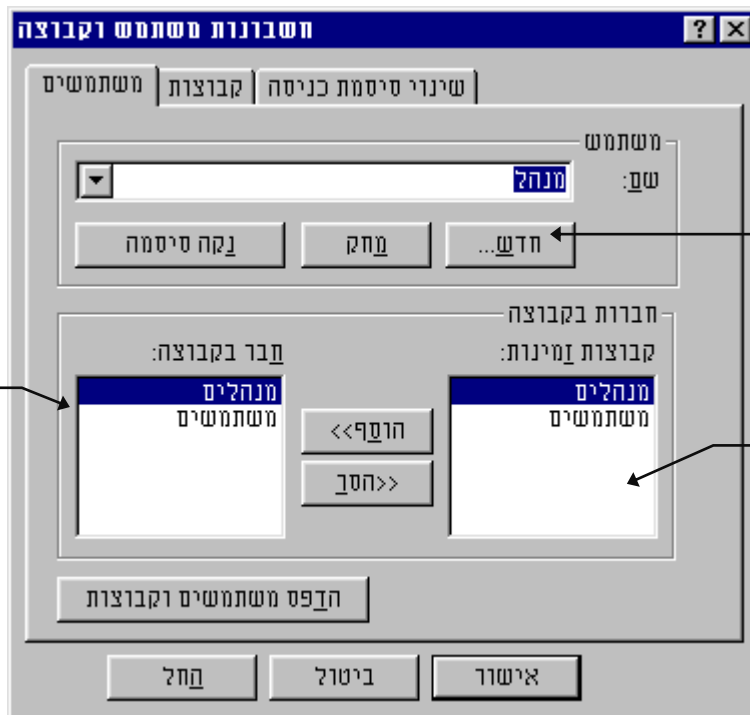
2. יש ליצור חשבון משתמש אחד או יותר או בחר בחשבון משתמש אחד או יותר, על מנת לנהל את קבוצת העבודה ולהיות בעלים של מסד הנתונים ושל האובייקטים שלו.

חשבונות **המנהל והמשתמש** חשובים כי יש להם הרשאות שלא ניתן להסיר. החשבונות שתגדיר יחליפו את חשבון המנהל המוגדר מראש, שהוא חשבון ברירת המחדל לבעלים ולמנהל.

חשבון המנהל מותקן עם כל עותק של Microsoft Access. כאשר הסיסמא הינה ריקה בכדי להפעיל מנגנון הרשאות יש להקליד סיסמא בשם משתמש «מנהל».



3. בכדי ליצור משתמשים חדשים יש להקליק חדש ולהוסיף את השם



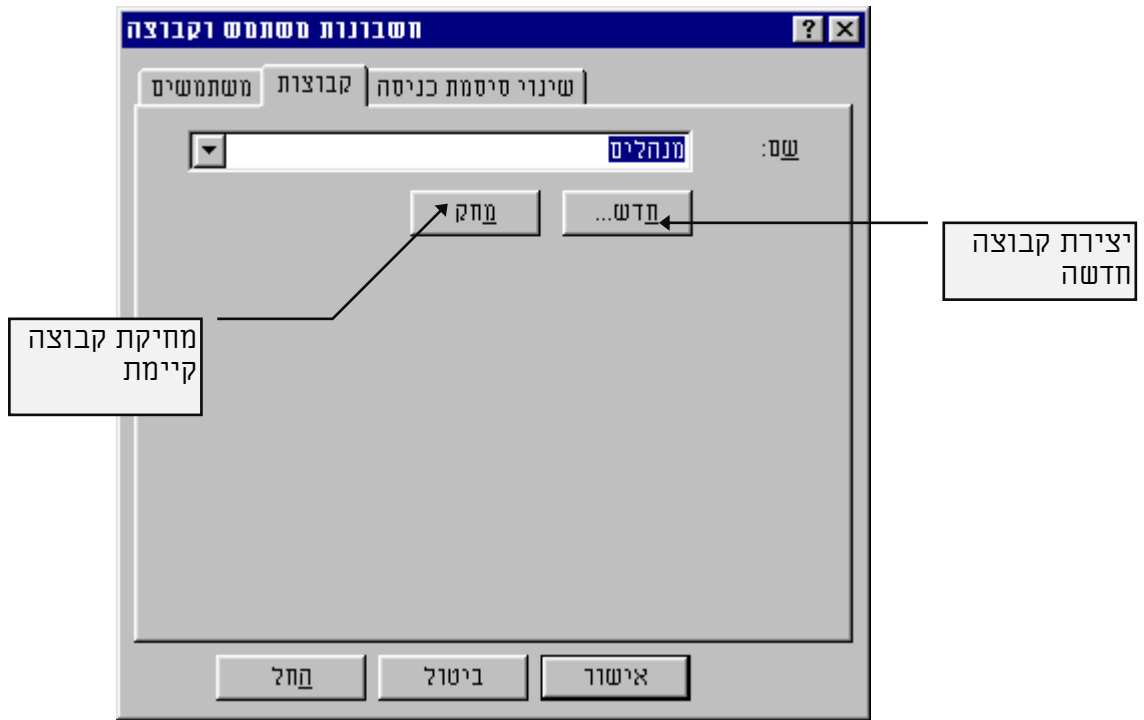
יצירת שם משתמש חדש

החלת חברות של משתמש בקבוצות הקיימות. הרשאות הקבוצה עוברת למשתמש

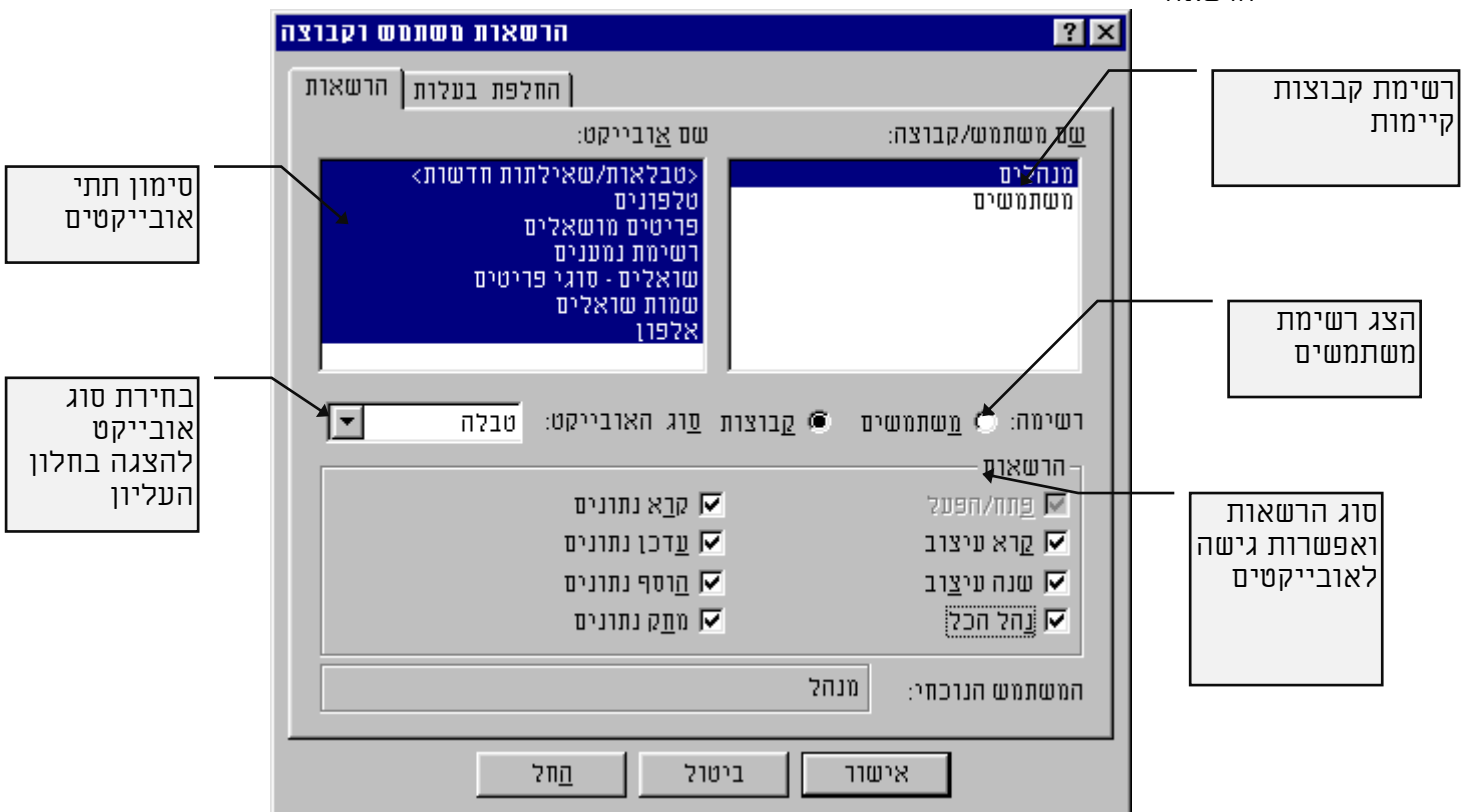
קבוצות שקיימות במרחב העבודה

סיסמאות לחשבונות, לפי הצורך. כדי לאבטח את מסד הנתונים, חשוב להוסיף סיסמה לחשבון המנהל ולחשבון הבעלים. בנוסף, אפשר שתוצה להוסיף סיסמה לחשבונות שתיצור עבור משתמשים, או להנחות את המשתמשים ליצור לעצמם סיסמאות.

4. בכדי ליצור קבוצות יש לבחור חדש ולרשום את שם הקבוצה



5. בכדי להכיל הרשאות של קבוצות או משתמשים יש לבחור שם משתמש/קבוצה לבחור סוג אובייקט ובאותו אובייקט לבחור את תתי האובייקטים הרצויים, ולהכיל עליהם סוג הרשאה



- * צור חשבונות משתמשים עבור כל אחד מהמשתמשים של מסד הנתונים וצור קבוצות על מנת לעזור לך לנהל את חשבונות המשתמשים.
- הסר את הרשאות ברירת המחדל והקצה הרשאות חדשות עבור מסד הנתונים ועבור האובייקטים שלו.

חשוב יש להסיר את ההרשאות של קבוצת המנהלים ושל קבוצת המשתמשים הכוללים את כל המשתמשים בקבוצת העבודה. אם לא תעשה זאת, המשתמשים עשויים לקבל הרשאות לא רצויות לאובייקטים. אחרי הסרת ההרשאות, למשתמשים תהיינה רק ההרשאות שאתה תגדיר.

דוגמא: הגבלת קבוצה להריץ טפסים מבלי שיהיה אפשר לבצע כל שינוי אחר.

שם משתמש/קבוצה: מנהלים

שם אובייקט: <טפסים חדשים>

רשימה: משתמשים קבוצות סוג האובייקט: טפס

הרשאות:

<input checked="" type="checkbox"/> פתח/הפעל	<input checked="" type="checkbox"/> קבא נתונים
<input type="checkbox"/> קרא עיצוב	<input checked="" type="checkbox"/> עדכן נתונים
<input type="checkbox"/> שנה עיצוב	<input checked="" type="checkbox"/> חוסף נתונים
<input type="checkbox"/> נהל הכל	<input checked="" type="checkbox"/> ממשק נתונים

המשתמש הנוכחי: מנהל

בדוגמא זו בחרתי בקבוצת «משתמשים»

אפשר רק לפתוח טפסים. כלומר להריץ אותם, אך אי אפשר לעבור למבט עיצוב

בכל הטפסים של מסד הנתונים

6. שינוי בעלות של אובייקטי מסד הנתונים

אובייקט: טלפונים

בעלים נוכחי: מנהל

אובייקט חדש: טבלה

בעלים חדש: מפתח

רשימה: משתמשים קבוצות

החלף בעלות

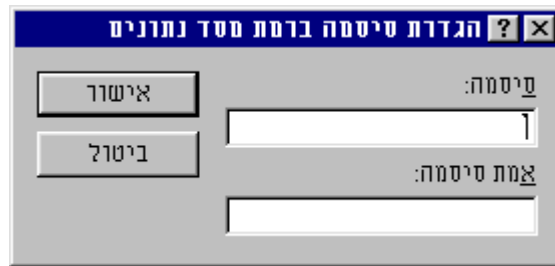
המשתמש הנוכחי: מנהל

אובייקטים לשינוי בעלות.

שם בעלים

החלף בעלות

7. סיסמת אבטחה פרטית של מסד הנתונים, מגבילה את פתיחתו ממי שאין לו הרשאה פרטית.



8. הצפנת מסד נתונים מונעת את קריאתו על ידי תוכנות עזר ועל ידי מעבדי תמלילים.



הערות

אבטחת נתונים ב Microsoft Access תלויה במערכת הקשרים בין חשבונות משתמשים בקבוצת העבודה והרשאות שמגדירים עבור מרכיבים שונים של מסד הנתונים. תוכנת Microsoft Access מאחסנת את פרטי חשבונות המשתמשים יחד עם הגדרות הקבוצה והסיסמאות, בקובץ מסד הנתונים (.MDB). וגם בקובץ מרחב העבודה ששמו באקסס 2 SYSTEM.MDA כך שכאשר מעבירים מסד נתונים מאובטח יש להעתיק גם את קובץ מרחב העבודה.

כאשר תיצור מחדש את חשבונות המשתמשים והקבוצות, הקפד לזהות אותם באופן ייחודי, על ידי הגדרת מספר אישי (PID).

אקסס שומר נתוני סרגלי כפתורים בהתאם לשם המשתמש, כך שאם נעשה שינוי בכפתורים הוא יבוא לידי ביטוי באותו שם. כנ"ל ברשימת בקבצים אחרונים, שהינה לפי שם משתמש.

3.6. יצירת מסד נתונים מאובטח

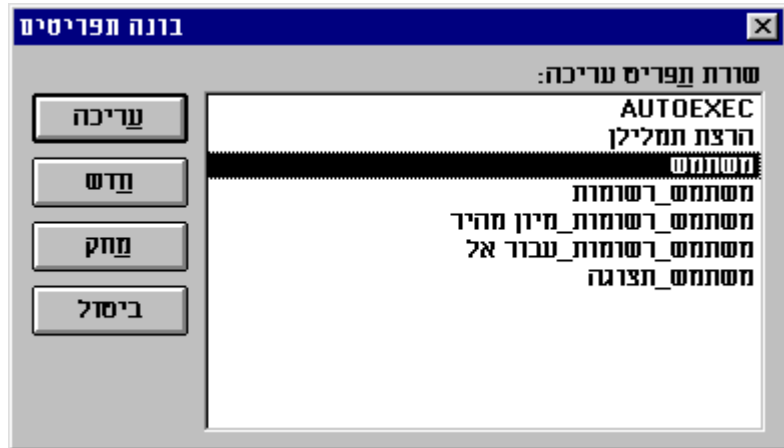
1. יש להכניס סיסמא ב«מנהל». מנגנון ההרשאות פועל!!
2. יש ליצור שם משתמש חדש לדוגמא «מפתח» ולתת לו חברות בקבוצת «מנהלים». (לקבוצת «מנהלים» יש הרשאות מלאות).
3. יש לבטל את החברות של «מנהל» בקבוצת «מנהלים».
4. יש לבטל את כל ההרשאות של קבוצת «משתמשים» (מכיוון שאין אפשרות לבטל את החברות של «מנהל» וכל משתמש אחר בקבוצת «משתמשים»).
5. יש להעביר בעלות על כל האובייקטים במסד לשם המשתמש החדש שהוא «מפתח». (מכיוון שהבעלים של האובייקטים יש להם הרשאה מליאה).
6. מסד הנתונים מאובטח!!!

בכדי לבטל מנגנון הרשאות יש להיכנס בסיסמת «מנהל» ולרוקן סיסמא «מנהל» לכלום.

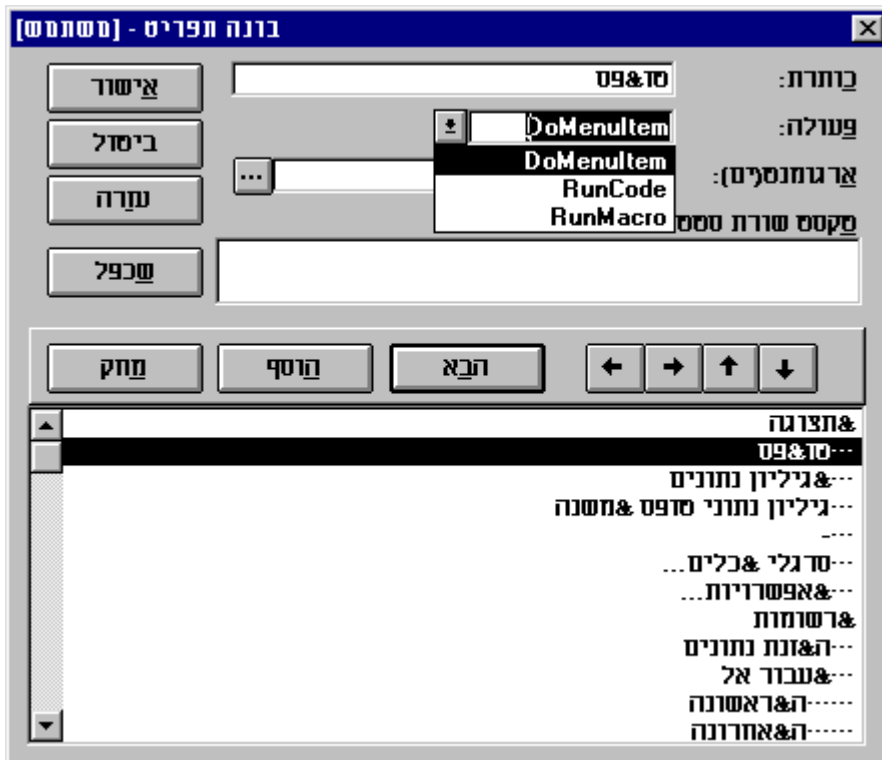


2.7. בונה התפריטים

אנו נכנסים לאותו תפריט מקרויים



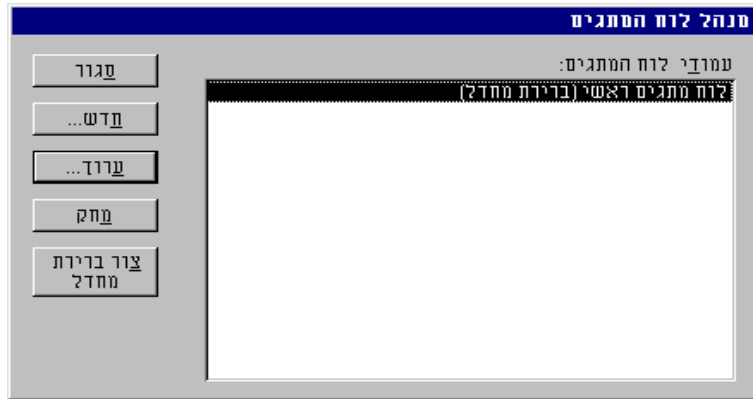
בונה התפריט בכבודו ובעצמו



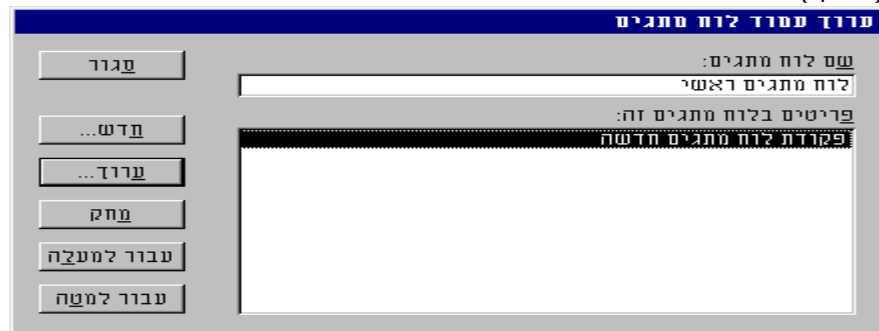
יש לבצע את כל השינויים הרצויים, ובסיומם להקיש «אישור» במקרה של תפריט חדש הבונה יבנה את המקרויים המתאימים, ובמקרה ויש הוא ישנה את המקרויים.

3.7. מנהל לוח מתגים

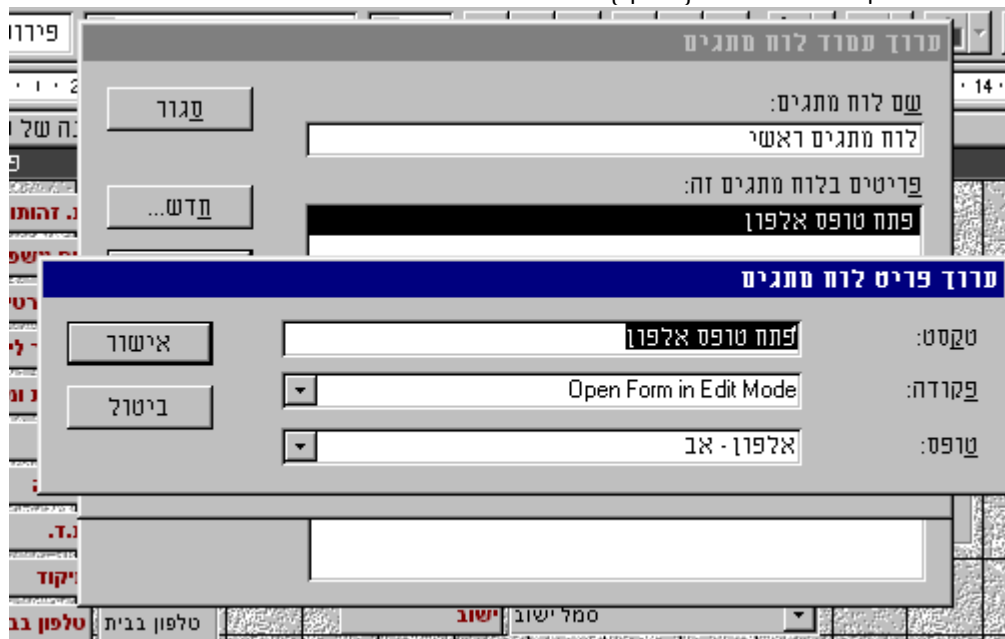
לוח מתגים הינו בעצם טופס תפריטים להפעלת המנהל יש להגיע דרך {מנהל} {תוספות} {מנהל לוח מתגים} נקבל את מנהל המתגים ובו חלון עם שמות לוחות מתגים קיימים. בכדי ליצור פריטי לוח מתגים יש להקיש על כפתור {ערוך}



נקבל את חלון "ערוך עמוד לוח מתגים" בכדי ליצור פריטי לוח מתגים יש להקיש על כפתור {ערוך}



בשלב זה נקיש על כפתור {ערוך} בכדי לעצב את הפריט בלוח המתגים



4.7 . פיתרונות בעזרת קוד



יש לנו טבלה שיש בה נתונים מעורבים בעברית ואנגלית. אנו רוצים למיין דוח באופן שיתן עם כל אות עברית את האות הלועזית התואמת, בראשונה כל השמות שמתחילים באות א' ואחריו אלו שמתחילים באות A וכן הלאה, כיצד אפשר לפתור בעיה זו?

תשובה:

ישנם שני דרכים:

1. א. יש ליצור טבלה או מערך שבו כל אות בעברית תהיה מול האות האנגלית.
- ב. בשלב השני יש ליצור פונקציה שתחזיר אות עברית לפי אות לועזית.
- ג. בשלב השלישי יש ליצור בשאילתא, שדה מחושב שיתן את האות הראשונה של השם, אם עברי באופן פשוט ואם לועזי את התרגום שלו לעברית.

2. אפשר לבסס המרה על פונקצית CASE שתבצע את הכל יותר מהר, ובאופן גמיש יותר, כן שמול אות אחת יהיו כמה אותיות.

1.

```
Function EngChrToHeb (chrEng As String) As String
Dim chrHeb As String * 1
chrEng = UCase$(chrEng)
Select Case chrEng
Case "A": chrHeb = "א"
Case "B": chrHeb = "ב"
Case "CH": chrHeb = "ח"
Case "CA": chrHeb = "כ"
Case "C", "CE": chrHeb = "ס"
Case "D": chrHeb = "ד"
Case "E": chrHeb = "א"
Case "F", "P": chrHeb = "פ"
Case "G": chrHeb = "ג"
Case "H": chrHeb = "ה"
Case Else: chrHeb = chrEng
End Select
EngChrToHeb = chrHeb
End Function
```

2.

```
SELECT DISTINCTROW EngChrToHeb(Mid$([1,1],[שם אדם].[שם אנשים].[תורנות - אנשים]) AS OT,
[תורנות - אנשים].[שם אדם])
FROM [תורנות - אנשים]
ORDER BY EngChrToHeb(Mid$([1,1],[שם אדם].[שם אנשים].[תורנות - אנשים]));
```

שם אדם	OT
efi	א
Gadi	ג
Pini	פ
	*

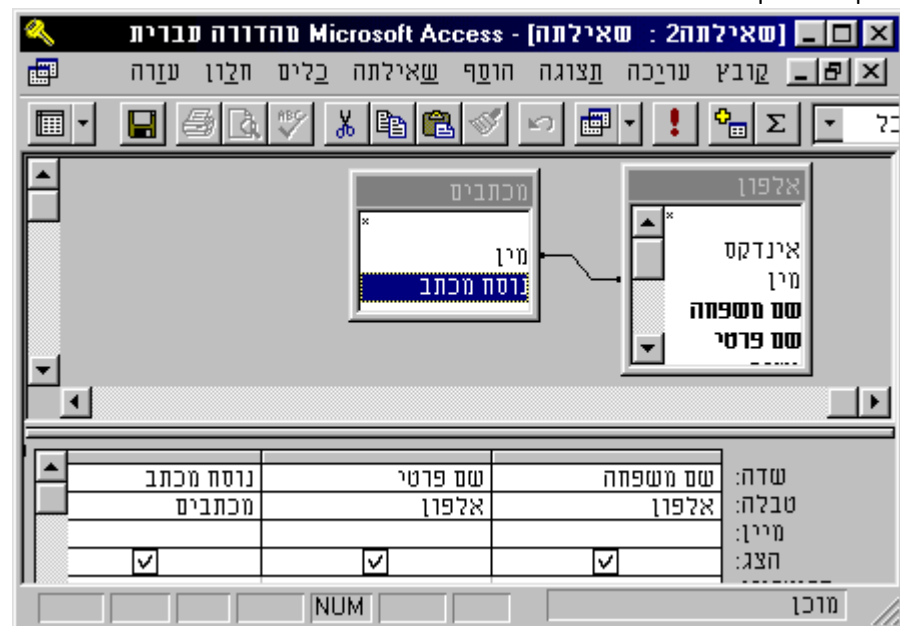


אנו רוצים לסווג אנשים לפי גברים ונשים ולשלוח מכתב שונה עפ"י המינים.

תשובה: יש ליצור שדה סיווג מין: גברים/נשים כשהמין הוא גבר נקליד את האות "ז" וכאשר הוא אשה נקליד את האות "נ". לאחר מכן ניצור טבלה חדשה ובה שני שדות שדה ראשון «מין», ושדה שני «נוסח מכתב». הרשומה הרשונה תהיה מכתב לגברים ושדה «מין» יכיל את האות "ז", והרשומה השניה תהיה מכתב לנשים, ושדה «מין» יכיל את האות "נ".



ניצור שאילתא שתהיה מבוססת על שתי הטבלאות כאשר השדה המקשר ביניהם יהיה שדה «מין» באופן הבא:

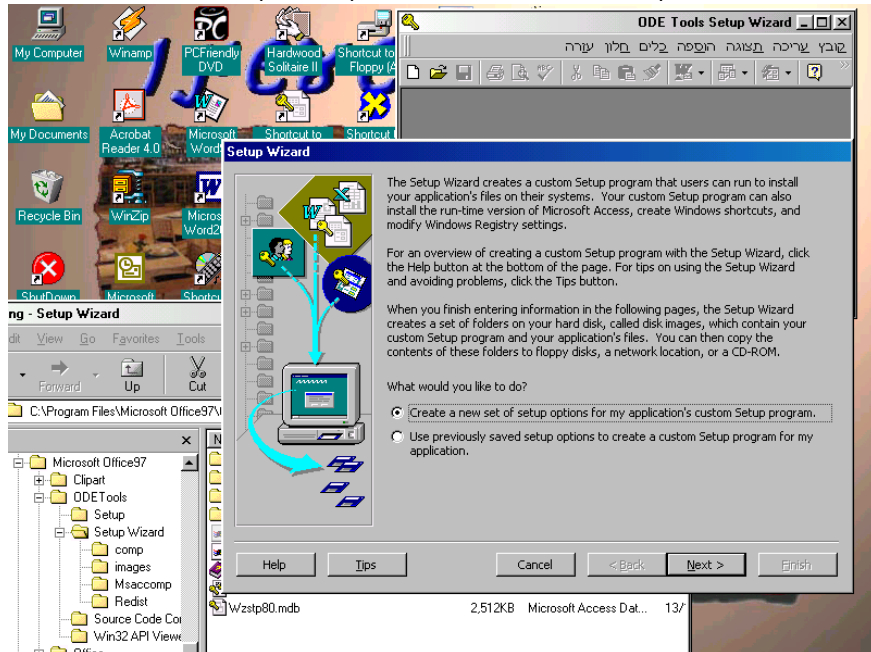


כך שכל רשומת אלפון תמשוך אחריה רשומה ממכתבים בהתאמה לשדה «מין» על בסיס אותה שאילתא ניצור דוח.

5.7 גרסת Runtime

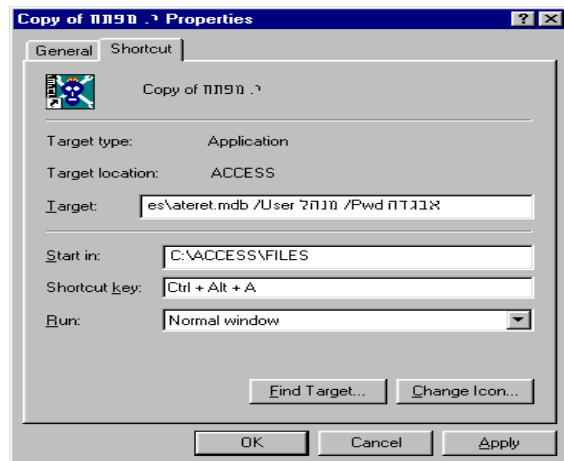
לערכת האופיס מפתחים מצורף תקליטור שלם ובו אוסף של תוכנות למפתחים שיש בו בין היתר גרסת Runtime של האקסס שמאפשרת להריץ ישום אקססי מבלי יכולת לפתחו. כך שמפתח יכול ליצור ישום ולחלקו עם גרסת Runtime של האקסס במספר עותקים שירצה. בתקליטור זה יש תוכנות נוספות כמו אשף התקנות שמאפשר ליצור תוכנת התקנה לישום שלך, כולל עדכון הרישום, כן יש בו תוכנה לניהול מסדים משוכפלים. ועוד מאשפים נוספים לאקסס, ועד למאמדים טכניים.

בכדי ליצור קובץ התקנה של גרסת ההרצה, ישנו אשף התקנה ששמו Wzstp80.mdb, וכאשר פותחים אותו באקסס, הוא יוצר תוכנת התקנה לאקסס, בכמה שלבים.



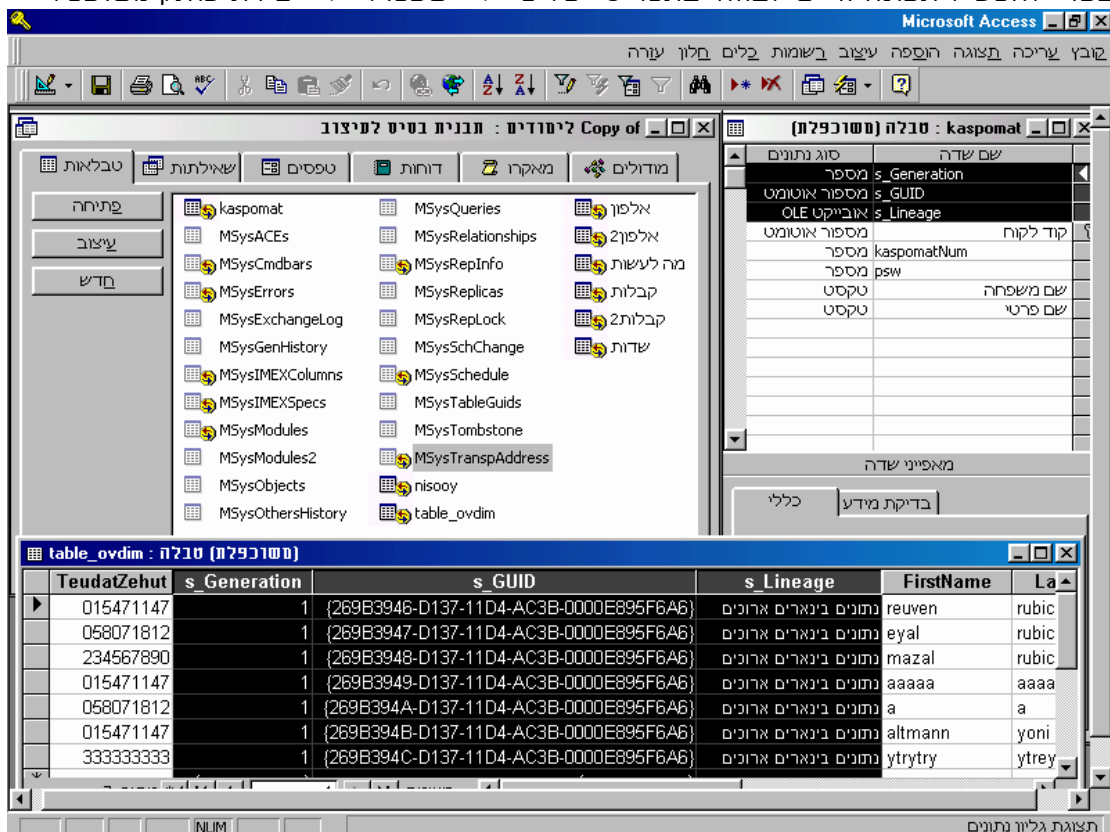
הכנה לקובץ להרצת במוד RUNTIME

1. לשנות את הפונט בכל המקומות ל ARIAL (או פונט אחר שתומך בעברית).
2. להראות/להסתיר סרגלי כלים בכניסות ויציאות מטפסים/דוחות. (ה RUNTIME מעלים את כל סרגלי הכלים הסטנדרטיים).
3. ליצור קובץ INI בלי הקטע של התוספים, אחרת יבקש את התוספים בכל כניסה לתוכנה.
4. ליצור דיסקטי התקנה, ולנסות להתקין אותם במחשב ניטרלי. (לעיתים יש בעיות בקבצים מחוברים)
5. ליצור צלמיות לתיקון ולדחיסה. REPER/ COMPACT/
6. להגן על התוכנה בעזרת הרשאות. אפשר ליצור צלמית עם שם משתמש וסיסמא בתוכה.



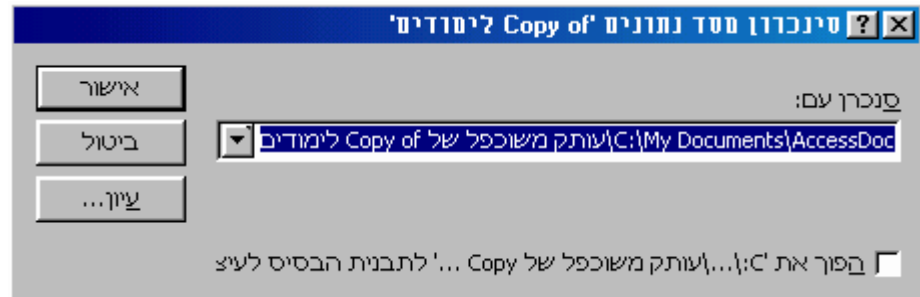
6.7. שיכפול מסד נתונים

הטכניקה של מסד נתונים משוכפל הינה במטרה להחזיק כמה מסדי נתונים מנותקים במקומות שונים כאשר אנו רוצים שהנתונים יהיו מעודכנים בכולם. המסד המשוכפל הינו מסד ראי של המסד נתונים מקורי, השינויים ברשומות במסד נתונים אחד ישתקפו במסד הנתונים השני, בעת בקשת סנכרון בין מסדי הנתונים. טכניקה זו מאפשרת לנו להחזיק שני מסדים בלי קשר ביניהם במהלך העבודה, ואעפ"י כן הנתונים יעודכנו מאחד לשני. כמו כן טכניקה זו מאפשרת לנו להפיץ עותקים של התוכנה, ולעדכן שינויים בפיתוח באופן קל אצל המשתמש. הפיתוח הינו אפשרי רק במסד במסטר. שינוי הפיתוח במסד המסטר יועתק אל שאר המסדים בעת בקשת סנכרון. במקרה של התנגשות בנתונים בעת סנכרון, ישנו פותר התנגשות שמציג את שני הרשומות הסותרות ומאפשר למנהל להכריע בדבר. בכדי להפעיל תכונה זו יש לבחור בתפריט "כלים" -> "שכפול" -> "יצירת עותק משוכפל"



אנו רואים תבנית בסיס של מסד משוכפל. נוספו מספר טבלאות מערכת, ונוספו בכל טבלה שלוש שדות נוספים בכדי לנטל את כל השיכפול.

בכדי לבצע סנכרון של בסיס הנתונים יש לבחור את האופציה "סנכרון כעת" בתפריט דלעיל



במקרה של התנגשויות יופיע לנו מסך מתאים

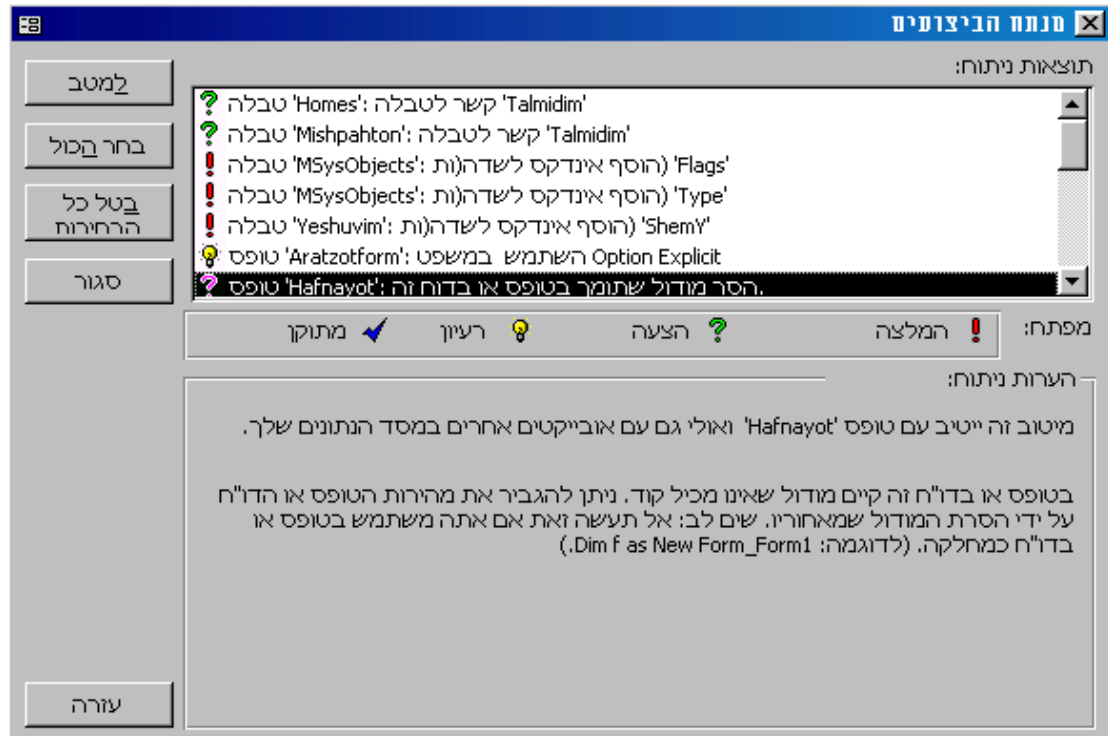
7.7. פיצול מסד נתונים

במקרה של פיתוח אצל לקוח, הקובץ הישן שנמצא אצל הלקוח מבחינת הנתונים הוא המעודכן ביותר, ובכל פעם שהמפתח מביא עותק חדש בדרך הרגילה הוא צריך לעדכן בעותק החדש את הנתונים שנוספו אצל הלקוח, בשיטה הזו הוא לא יצטרך לעדכן מחדש כל עותק חדש שהוא מביא ללקוח, מכיוון שקובץ הממשק הוא לבדו משתנה.



8.7. מנתח ביצועים

מנתח הביצועים הינו אשף, שתפקידו לבדוק ולנתח את כל האובייקטים בבסיס הנתונים, להמליץ, או להציע או לתקן שיפורים בכל אחד מהאובייקטים הקיימים. אפשר לבחור את כל האובייקטים או חלקם בשימוש באשף זה. אשף זה נמצא בתפריטים ב"כלים" < "נתח" < "ביצועים".



9.7 . מנתח טבלאות

אשף ניתוח טבלאות

מנתח הטבלאות: מסתכל על הבעיה

ייתכן שהטבלה או הגליון האלקטרוני שלך מאחסנים אותו המידע פעמים רבות. שכפול מידע עלול לגרום לבעיות.

ראשית, שכפול מידע מבוזבז שטח. הצג בפני דוגמה. <<

שנית, שיכפול מידע עלול להוביל לטעויות. הצג בפני דוגמה. <<

מוצרים וספקים			
מוצר	קוד ספק	ספק	כתובת
Ravioli Angelo	PAST	Pasta Buttini s.r.l.	dei Gelsomini, 153
Gnocchi di nonna	PAST	Pasta Buttini s.r.l.	dei Gelsomini, 153
Carnarvon Tigers	PAVL	Pavlova, Ltd.	74 Rose St.
Outback Lager	PAVL	Pavlova, Ltd.	74 Rose St.
Pavlova	PAVL	Pavlova, Ltd.	74 Rose St.
Vegie-spread	PAVL	Pav, Ltd.	74 Rose St.

מידע על ספק חוזר על עצמו.

שם ספק אינו מאוית כראוי.

סיים <הבא> <הקודם> ביטול

מנתח הטבלאות הוא אשף המאפשר לנו לנרמל את בסיס הנתונים שלנו באופן אוטומטי, או לתת לנו לבחור לפצל את הטבלאות לפי רצוננו.

אשף ניתוח טבלאות

אילו שדות מכילים מידע החוזר על עצמו? רצוי להעביר שדות אלו לטבלאות חדשות. כל טבלה צריכה להכיל נתונים אודות נושא מסוים. באפשרותך לגרור שדות כדי ליצור טבלאות חדשות, ולהעביר שדות בין טבלאות. כמו כן, באפשרותך לשנות את שמות הטבלאות ולקבוע שדות מפתח ראשי.

טבלה 1: TeudatZehut, LastName, בדיקת מידע ל- טבלה 2

טבלה 2: זיהוי ייחודי מחולל, FirstName

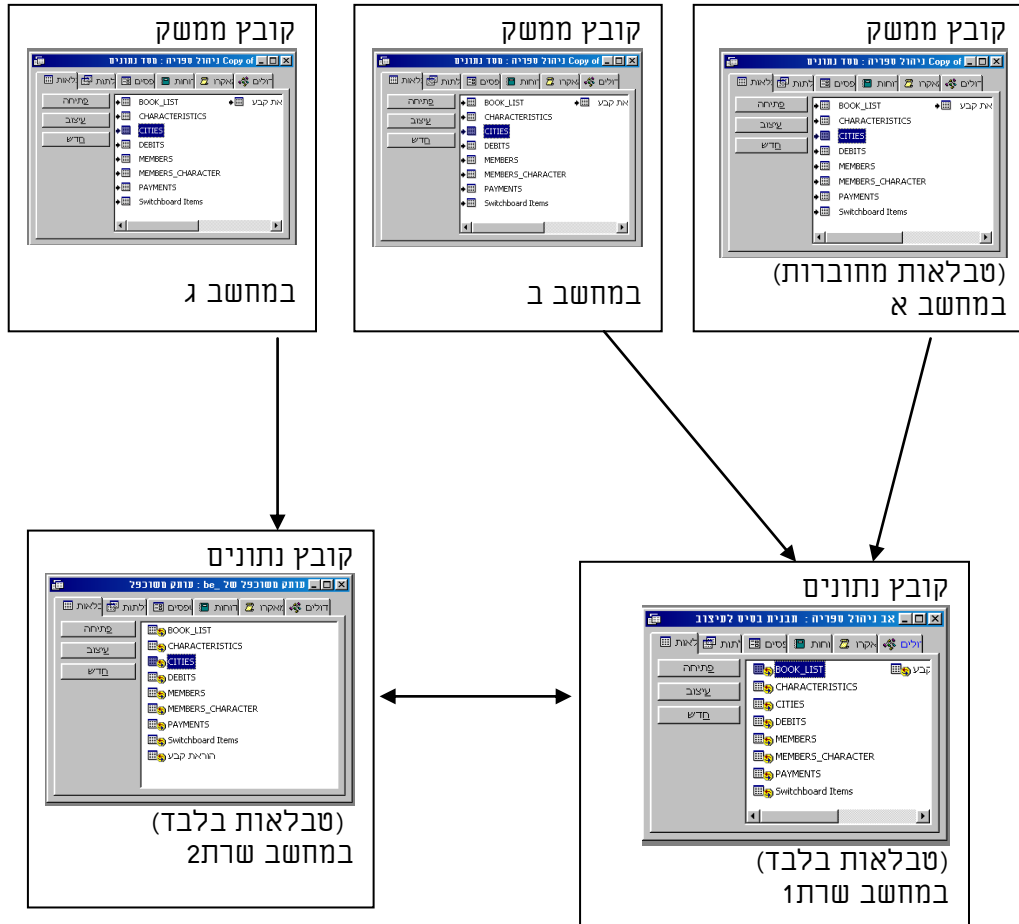
שם טבלה: טבלה 1

אישור ביטול

סיים <הבא> <הקודם> ביטול

01.7. סכמה של מסדים מפוצלים ומשוכפלים

הקבצים העליונים הינם קבצי ממשק שאין בהם טבלאות פיזיות. הטבלאות הפיסיות נמצאות בקבצים התחתונים. כך שאפשר לעבוד על אותם נתונים ממחשב א וב. הטבלאות עצמן נמצאות המחשב שרתו ובמחשב שרת2, שהם מסדים משוכפלים שמאפשרים לסנכרן בין נתונים שעודכנו בנפרד בין שני המחשבים.



11.7. טבלת השוואות בין אקסט לבין ויזואל בייסיק

תכונה	אקסט	ויזואל בייסיק
יעוד	ממזכירות, מפתחים ועד מתכנתים	מתכנתים בלבד.
קלות ומהירות פיתוח	התוכנה מאפשרת בקלות ובמהירות רבה ליצור יישום מלא, תוך אוטומציה בתהליכים שונים.	פיתוח במיוחד של מסד נתונים עדיין קצת קשה, איטי, ומסורבל.
פיתוח ללא קוד	ישנה אפשרות ליצור יישום מלא ללא שורת קוד אחת, כך שגם לא מתכנתים יכולים לפתח בו, ואף לבטא קשרים מורכבים כמו יחס של יחיד לרבים בטופס משולב.	בלתי אפשרי. עדיין בכדי ליצור טופס עדכון פשוט של טבלה, יש לרשום קוד לעדכון, ולנפות באגים, בפעולות מאוד פשוטות ובסיסיות בסיס נתונים.
פיתוח ידדוטי	תחביר גמיש, תיקון אוטומטי של חלק מהשגיאות. מאפשר שמות עם רווחים.	תחביר מאוד נוקשה, שאסור לסטות ממנו.
פיתוח ויזואלי	פיתוח ויזואלי במיטבו, החל מיצירת הטבלה, דרך קשרי גומלין, יצירת שאילתות לפי דוגמה, טפסים ודוחות.	פיתוח ויזואלי, שחוזק רבות בגרסה האחרונה, עם "סביבת הנתונים" ודוחות, ושעדיין צריך כמה כלים להשלמה.
גמישות בפיתוח ויצירתו	ישנם קונבנציות שקשה לסטות מהם, והפיתוח הינו במסלול מאוד מסוים. כך למשל כל טופס מאוגד אל טבלה או שאילתא אחת בלבד. ולעיתים תפיסה זו מסרבלת.	התוכנה בנויה ממודולים בסיסיים, כך שאפשר לשחק ולעצב אותה כרצוננו. כך למשל אפשר ליצור בטופס אחד, קישור אל כמה טבלאות ללא קוד. יצירתיות ומודולריות במיטבם.
סרבול היישום	כל יישום קטן ככל האפשר, יצריך את סביבת הפיתוח הגדולה והמסורבלת של האקסט.	אפשר ליצור יישום קטן, זריז ומהיר, שמכיל בתוכו את עצמו ותו לא.
כלים מובנים	ישנם הרבה כלים מובנים לטיפול בבסיסי נתונים, כמו מנגנון שכפול מסד נתונים, מנגנון הרשאות וסמאות פנימי, יבוא ויצוא נתונים, עבודה עם בסיסי נתונים זרים, יצירת דפי אינטרנט דינמיים דמויי טפסים.	גם אם יש כמה אשפים, עדיין אין כלים מובנים כמו באקסט, ויש ליצור את חלקם מהתחלה.
יצירת קבצי EXE	לא אפשרי. יש Run Time גדול ומסורבל.	אפשרי לייצור קובץ EXE זריז ומהיר. כמו כן אפשר ליצור פקדי אקטיבX, וקבצי DLL.
אשפי פיתוח	ישנם אשפים מעולים לפיתוח מהיר בכל אספקט באקסט, כולל אשפי מסדים, אשפי שדות, אשפי פיצול, אשפי קישור מחדש.	ישנם מספר אשפים, אבל לא מספיק.
פיתוח ללא באגים	הפיתוח בסביבה זו, מוגן בפני טעויות שונות של המשתמשים, ומאידך מכיל בתוכו באגים שונים.	עדיין מספר הבאגים גבוה יחסית לאקסט.
תלות במערכת	קשה לשנות מדפוסי הפיתוח שמכתיב האקסט. אמנם אפשר להטמיע פקדי אקטיבX, אבל הסביבה תומכת בהם בכבדות.	סביבה עצמאית לחלוטין. יצירתיות מלאה בידי המפתח ליצור יישום כרצונו.
תמיכה בפיתוח	ישנם כלים לתמיכה בפיתוח כמו בונה הביטויים.	אין כלים לתמיכה בפיתוח.
תמיכה בעברית	תמיכה מלאה בעברית, בכל הממשק כולו, כולל בקבצי עזרה. תוכנה אידיאלית לאנשים שהאנגלית אינה שפה חזקה אצלם.	תמיכה חלקית בעברית.
טיפול בקבצים	טיפול קל ופשטני בקבצים. כל הנתונים כולם יכולים ל"שבת" בקובץ אחד.	יש לטפל בהרבה בקבצים. כל טופס וכל דוח וכל אובייקט נשמרים בקובץ עצמאי.

21.7. עיצוב תוכנה – הנחיות כלליות

עם הפנים למשתמש הקצה

לבחון עם המשתמש את אב הטיפוס ואת התוכנה עצמה לאחר הפיתוח, ולראות מה מתאים לו ומה לא. בסופו של דבר הוא הפוסק האחרון והוא הבוס, והוא זה שיחליט איך להשתמש בתוכנה, מה יעיל ונוח בעבורו, והאם בכלל להשתמש בה או לזרוק אותה לאשפה. לדוגמא: טופס רציני שעבדתי עליו הרבה מאוד, לא היה בשימוש בכלל, בגלל שעלה לאט מדי.

שמות כפתורים ותפריטים ידידותיים ומשמעותיים למשתמש קצה

למשל עדכון ודוחות לא אומר הרבה למשתמשים רגילים. עדיף לכתוב קלט מידע ואחזור מידע למשל. הסמנטיקה חשובה כאן מאוד. יש למצוא ביטויים משפת היום יום, ולא מקורס מנתחי מערכות.

הקפדה על חזותיות של התוכנה

התפיסה החזותית אצל דוב האנשים חזקה מהפיסה המילולית. כשמתרגלים לתמונה אין צורך לקרוא את כל הטקסט. התמונה גם מבהירה את הכוונה ואפשר להשתמש בה להמחשה, מעבר לזה שתמונות שוברות את השגרה והופכות את העבודה לחווייתית ומהנה יותר. לכן תמיד יש לשלב בכפתורים תמונות עם כיתוב גם יחד, כמו כן רצוי לשלב בטפסים השונים צלמיות ותמונות, שיהיו מעניין הטפסים.

התמצאות בתוכנה

חשוב שההתמצאות בתוכנה תהיה קלה. שהמשתמש יוכל לנווט בקלות להיכן שהוא רוצה, מבלי להיות תלוי בזכירה של צירוף מקשים כזה ואחר, או מסלול ארוך של פתיחת טפסים. יש ליצור תפריטים וסרגלי כלים שתהיה בהם הפונקציונליות הנדרשת. יש לקבץ טפסי תפריטים בכרטסת אחת, כך שכל התפריטים יהיו נגישים. לקבץ טפסי תחזוקה של טבלאות סמלים לטופס אחד כך שיהיה אפשר לגשת ולעדכן בטופס אחד את כל הטבלאות.

יישום ארגונומי

ליצור אפשרות בכל מקום לעבוד עם המקלדת ללא עכבר. עבודה שוטפת עם עכבר בבסיס נתונים דורשת מאמץ רב, איטית יותר ומעייפת את המשתמש. כמו כן, ליצור מקשים חמים לכל מה שצריך בעזרת מקרו הקצאת מקשים. לאפשר למשתמש לבצע כמה שיותר בכמה שפחות פעולות. לעיתים צריך ליצור מסלולים שונים לשם כך.

זהירות מסינוור המשתמש

יש גבול לקליטה האנושית. לכן יש למעט בריבוי של אובייקטים וצבעים במקום אחד, אעפ"י שזה חסכוני, הדבר מסנוור את המשתמש. יש צורך לבזר ולהפריד נתונים, וכן לא להגזים בצבעים ולהשתמש בהם באופן סולידי. למשל אסור להשתמש באדום ככה סתם לרקע של טופס. אדום הוא צבע שמשתמשים בו במינון נמוך, בעיקר לאזהרות והתראות. נהוג להשתמש באותו מסך בגוונים שונים של אותו צבע, למשל גווני אפור שונים, או גווני כחול שונים.

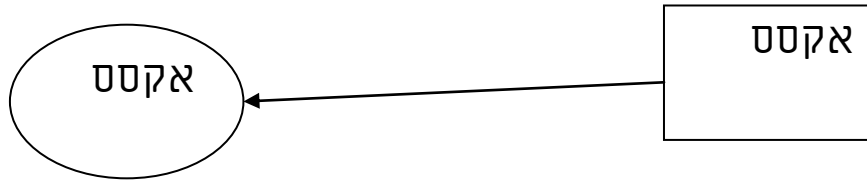
השהיית משתמש

אסור לתת למשתמש לחכות יותר מדי זמן, בפעולות עדכון או שליפת מידע. במהלך הפיתוח אין שימת לב לעניין זה, כי בדרך כלל עובדים עם מספר רשומות קטן. בבדיקות הסופיות יש להכניס מספר רשומות שידמו את המצב האמיתי, בדרך כלל עשרות אלפי, ולבדוק את זמן התגובות. חוסר אינדקסים או ריבוי שלהם וכן ריבוי תיבות רשימה בטופס וכן ריבוי של תת טפסים בטופס אחד גורמים להאטה. לא להגזים באף אחד מהם. כמו כן, יש לבדוק את פלטפורמת המיחשוב של המשתמש ואת תצורת העבודה (רשת או לא) בכדי למטב את המערכת.

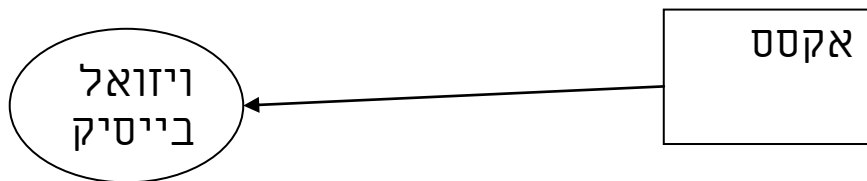
31.7. תצורות בסיסי נתונים

ישום חזית

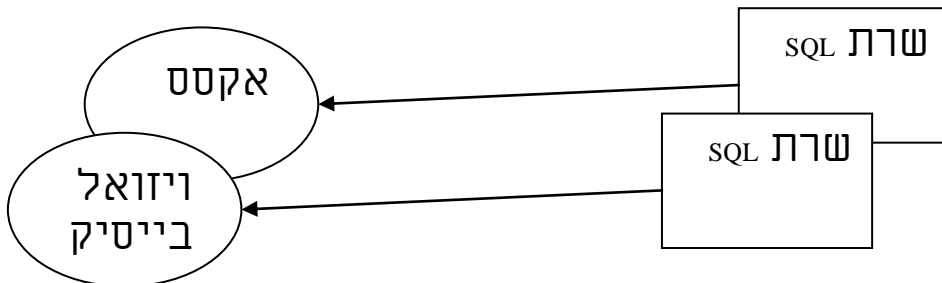
בסיס נתונים



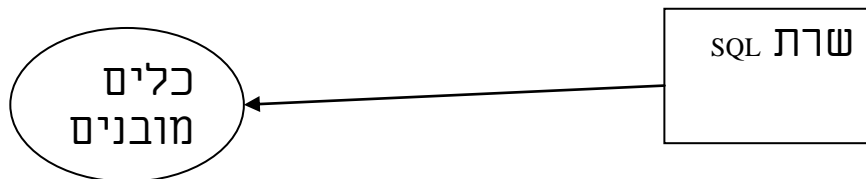
יתרונות: מהירות וקלות פיתוח, מיעוט באגים, עלות תוכנה ופיתוח נמוכה.
חסרונות: עבודה איטית ברשת, ובריבוי משתמשים, אפליקציה כבידה ומסורבלת. חוסר גמישות בפיתוח.



יתרונות: ישום מודולרי, קטן וזריז, גמישות רבה בפיתוח. עלות תוכנה נמוכה.
חסרונות: עבודה איטית ברשת, ובריבוי משתמשים, פיתוח איטי ומורכב יחסית לפיתוח באקסס, כתיבה מרובה של קוד, אין מספיק כלים לאוטומציה של פיתוח.



יתרונות: עבודה מהירה בריבוי משתמשים, אפשרויות חזקות יותר בעבודה עם נתונים, אמיונות נתונים גבוהה יותר.
חסרונות: מערכת מורכבת יותר, עבודה עם שתי מערכות בו זמנית, עלות תוכנה גבוהה יותר, נדרשת מומחיות רבה יותר.



יתרונות: בנוסף ליתרונות בסעיף הקודם, תיאום בין המסד ליישום החזית.
חסרונות: עלות גבוהה יותר, נדרשת מומחיות רבה יותר, לימוד ספציפי של כל מערכת, תלות ביישום החזית של התוכנה, מגבלות התלויים הכלים המובנים.