

# פונקציות וביטויים



גירסה: 2  
מעודכן לתאריך: 14/02/2013  
נכתב ע"י: תחום הדרכה

## תוכן עניינים

3	מחרוזות תווים
4	מתימטי
5	לוגי
7	תאריך / שעות / זמן
9	המרת ערכי שעות (שעות דקות/עשרוני)
10	היעדרויות
15	מונים
17	נתונים יומיים DAILY
21	נתונים תקופתיים
21	נתונים שנתיים
22	נלווים יומי SALDAY
22	תעריפים
23	הסכמים וטבלאות
25	קובץ\טבלת עובדים
27	חדרי אוכל
29	משימות
30	נתונים מטבלת EMPLOG
31	נתונים מטבלת EMPVEC
32	נתונים מטבלת DEPVAL
33	סידור עבודה
35	אינדקס

## מחרוזות תווים

### **LEFT(Character\_Value, Number\_of\_Characters)**

שימוש: לקבלת התווים השמאליים מתוך מחרוזת תווים  
דוגמא: **LEFT("LAVIE",3)** יחזיר את מחרוזת התווים "LAV"

### **STR(Number,Length,Decimals)**

שימוש: הסבת ביטוי מספרי לתצוגה כמחרוזת תווים  
**Number** = ביטוי מספרי (נומרי)  
**Length** = אורך מחרוזת התוצאה כולל נקודה עשרונית  
**Decimals** = מספר מקומות מימין לנקודה העשרונית  
דוגמא: **STR(2.9,5,2)** יחזיר את המחרוזת "2.90"

### **SUBSTR(Character\_Value,Start\_Position,Number\_of\_Characters)**

שימוש: הוצאת חלק ממחרוזת תווים  
**Character\_Value** = ביטוי המחזיר מחרוזת תווים (המחרוזת המקורית)  
**Start\_Position** = מספר התו במחרוזת המקורית שיש להתחיל ממנו את מחרוזת התוצאה  
**Number\_of\_Characters** = אורך (בתווים) של מחרוזת התוצאה

דוגמא: נניח כי מספר העובד הוא בן 12 תווים. נניח כי מספר המקצוע מוכלל במספר העובד ומתחיל בתו השלישי באורך ארבעה תווים.  
במצב זה הביטוי **SUBSTR(Employee\_Number,3,4)** יחזיר את מספר המקצוע.  
התוצאה עבור עובד שמספרו "300275300104" תהיה "0275", כלומר מהתו השלישי למשך ארבעה תווים.

### **TRIM(Character\_Value)**

שימוש: הורדת רווחים (בלנק) מסוף מחרוזת התווים – הסרת רווחים מימין  
דוגמא: **TRIM("TK5WIN ")** יחזיר ("TK5WIN")

### **LTRIM(Character\_Value)**

שימוש: הורדת רווחים (בלנק) מסוף מחרוזת התווים – הסרת רווחים משמאל  
דוגמא: **TRIM(" TK5WIN")** יחזיר ("TK5WIN")

### **UPPER(Character\_Value)**

שימוש: הסבת מחרוזת תווים לאותיות אנגליות גדולות  
דוגמא: **Upper("LAVIE COMPUTERS")** יחזיר "LAVIE COMPUTERS"

### **Strrev()**

שימוש: הפונקציה מקבלת שדה מסוג string ומחזירה אותו הפוך.  
דוגמא: **strrev(abcde) -> edcba**

## מתימטי

### **VAL(Character\_Value)**

**שימוש:** להסבת מחרוזת המכילה ספרות למספר שניתן לבצע עליו פעולות. פונקציה חשובה ביותר לכל מקרה בו נרצה לבצע פעולות מתימטיות או השוואות על שדות מפתח במערכת כמו: מפעל, תת מפעל, מחלקה, עובד, הסכם, קודי מיון, מקצוע

### **דוגמא: VAL("29")=29**

אם נרצה להוסיף ערך למספר הסכם יש להפעיל הפונקציה ואח"כ לבצע פעולת חיבור. לבדיקת מספר עובד גדול מ-1000 יש להפעיל תחילה את הפונקציה ואז לבצע בדיקה

### **DIV\_LONG(Div<sub>1</sub>,Div<sub>2</sub>)**

**שימוש:** חלוקת ערכים, התוצאה המתקבלת הערך "השלם"  
**דוגמא:**  $10/4=2.5$  ולכן **DIV\_LONG(10,4)** תוצאה = 2, תוצאה זו מציגה את השלם ללא השבר.

בצורה זו ניתן לבדוד שנה מתוך השדה קוד תקופה **DIV\_LONG(PERIOD-> PERIOD\_KEY,100)**

לדוגמא אם התקופה הינה 12/2004 השדה **PERIOD\_KEY** יכיל את הערך - 200412 ולכן התוצאה שתתקבל הינה- 2004.

### **MOD(Div<sub>1</sub>,Div<sub>2</sub>)**

**שימוש:** קבלת ערך השבר העשרוני (לאחר הנקודה)  
**דוגמא:**  $10/4=2.5$  ולכן **MOD(10,4)** תוצאה = 5, תוצאה זו מציגה את השבר ולא את השלם.

בצורה זו ניתן לבדוד חודש מתוך השדה קוד תקופה **MOD(PERIOD-> PERIOD\_KEY,100)**

לדוגמא אם התקופה הינה 12/2004 השדה **PERIOD\_KEY** יכיל את הערך - 200412 ולכן התוצאה שתתקבל הינה- 12.

## לוגי

.T.

**שימוש:** לבדיקה אם ערך ביטוי לוגי הוא "אמיתי" TRUE

.F.

**שימוש:** לבדיקה אם ערך ביטוי לוגי הוא "שקרי" FALSE

.NOT.

**שימוש:** לשלילת תנאי לוגי

.AND.

**שימוש:** לחיבור בין שני תנאים לוגיים שחייבים להתקיים  
**דוגמא:** מספר עובד גדול מ- 1000 וגם מספר מחלקה קטן מ- 129  
(VAL(EMPLOYEE)>1000).AND.(VAL(DEPARTMENT)<129)

.OR.

**שימוש:** לחיבור בין שני תנאים לוגיים שרק אחד מהם חייב להתקיים  
**דוגמא:** מספר מפעל שווה ל- 4 או מספר מפעל שווה ל- 7  
(VAL(PLANT)=4).OR.(VAL(PLANT)=7)

DELETED()

**שימוש:** לבדיקה אם הרשומה בקובץ מסומנת כמבוטלת (רשומות מסוג זה בד"כ יש להוציא מתוך הדו"ח)  
הפונקציה מחזירה את הערך "אמת" TRUE במקרה שהרשומה מבוטלת

DEL()

**שימוש:** לבדיקה אם הרשומה בקובץ מסומנת כמבוטלת (רשומות מסוג זה בד"כ יש להוציא מתוך הדו"ח)  
הפונקציה מחזירה את הערך "\*" (כוכבית) במקרה שרשומה מבוטלת, אחרת BLANK

IIF(Logical\_Value,Result if True,Result if False)

**שימוש:** לקביעת ערך מותנה בקיום תנאי לוגי  
Logical Value - תנאי לוגי  
Result if True - ביטוי/ערך שישמש כתוצאה במקרה שהתנאי הלוגי מתקיים  
Result if False - ביטוי/ערך שישמש כתוצאה במקרה שהתנאי הלוגי לא מתקיים

**דוגמא:** אם נחלק "סה"כ לשכר" ב"תקן יומי" נקבל מספר המציין "ימי עבודה".  
אם התקן היומי = אפס אין משמעות לחלוקה זו ואז נקבע שרירותית אפס (כדי שלא תתקבל שגיאת חישוב) ולכן: IIF(STD>0,TIME2MIN(TOT\_WAGE)/STD,0)

**INTRANGE(VAL(Char\_Value),VAL-1,VAL-2,...,VAL-20)**

**שימוש:** בדיקת המצאות ערך בטווח כלשהו, כתיבה מקוצרת במקום שימוש בשורות תנאי מרובות המופרדות ב-"or".

**תחביר:** VAL(Char\_Val) יש להפוך כל ערך מסוג אלפא לערך מספרי  
**דוגמא:** בדוק האם שדה הסכם מהקובץ היומי שווה לערך 1 או 3 או 8 או 10

כן      הדפס Y

לא      הדפס N

IIF(INTRANGE(VAL(DAILY-  
>CONTRACT),1,3,8,10),"Y","N")

**שימוש בפונקציה זו אפשרי במערכות TK2000 ו-TKSQL**

**PAGENO()**

**שימוש:** לקביעת מספר דף במחולל דו"חות

**דוגמא:** אם נרצה להדפיס נתון מיוחד בעמוד הראשון של הדו"ח ניתן לבדוק אם  
**PAGENO()=1**

## תאריך / שעות / זמן

### **CTOD(Character\_Value)**

**שימוש:** הסבת מחרוזת תווים המייצגת תאריך במבנה MM/DD/YY למבנה תאריך פנימי (שניתן לבצע עליו חישובים)  
**דוגמא:** CTOD("11/15/97")

### **DTOC(Date\_Value)**

**שימוש:** הסבת תאריך פנימי למחרוזת במבנה MM/DD/YY  
**דוגמא:** אם התאריך היום הוא 5 ביוני 1998 אז DTOC(DATE())="06/05/1998"

### **STOD(Character\_Value)**

**שימוש:** הסבת מחרוזת תווים המייצגת תאריך במבנה CCYYMMDD למבנה תאריך פנימי (שניתן לבצע עליו חישובים)  
**דוגמא:** STOD("19970313") יחזיר את התאריך של ה- 13 במרץ 1997 במבנה פנימי

### **DTOS(Date\_Value)**

**שימוש:** הסבת תאריך פנימי למחרוזת במבנה "CCYYMMDD"  
**דוגמא:** אם התאריך היום הוא 5 ביוני 1998 אז DTOS(DATE())="19980605"

### **DATE()**

**שימוש:** לקבלת תאריך נוכחי (תאריך ממערכת ההפעלה) במבנה חישובי/ פנימי

### **DATE2STR(DATE(),"MM/DD/CCYY")**

**שימוש:** הדפסת התאריך בפורמטים שונים: DD/MM/YYCC, MM-DD-YY וכד'  
**תחביר:** DATE2STR (DATE,"DATE\_FORMAT")  
DATE2STR(DATE(),"MM/DD/CCYY")

**דוגמא:** DATE2STR(DATE(),"MM/DD/CCYY")  
DATE2STR(DATE(),"DD-MM-YY") - הדפסת התאריך הנוכחי  
[ DATE() ] במבנה DD-MM-YY - תאריך אירופאי

### **JULDATE(Date<sub>1</sub>)-JULDATE(Date<sub>2</sub>)**

**שימוש:** פונקציה לחישוב מספר הימים בין 2 תאריכים  
**תחביר:** שימוש בנתון תאריך, תחביר: "CCYYMMDD" JULDATE("19981231")  
**דוגמא:** JULDATE(EMPLOYEE->STAT0\_TILL)-JULDATE(EMPLOYEE->STAT0\_FROM)

### **DAY(Date\_Value)**

**שימוש:** לקבלת היום בחודש מתוך התאריך  
**דוגמא:** אם התאריך היום הוא 5 ביוני 1998 אז DAY(DATE())=5

### **YEAR(Date\_Value)**

**שימוש:** לקבלת השנה מתוך התאריך  
**דוגמא:** אם התאריך היום הוא 5 ביוני 1998 אז  $YEAR(Date())=1998$

**MONTH(Date\_Value)**

**שימוש:** לקבלת החודש מתוך התאריך  
**דוגמא:** אם התאריך היום הוא 5 ביוני 1998 אז  $MONTH(Date())=6$

**ENDPP()**

**שימוש:** לקבלת תאריך סיום תקופה, לשימוש בפונקציות תקופתיות בטבלאות K, העברה לשכר וכו'.  
**דוגמא:** בדיקה האם עובד סיים לעבוד לפני סוף תקופת השכר :  
 $EMPLOYEE \rightarrow HIRE\_TILL < ENDPP()$   
כלומר התנאי יתקיים במקרה שתאריך סיום העובדה יתרחש לפני תאריך סיום תקופת השכר הנוכחית.

**TIME()**

**שימוש:** קבלת השעה ממערכת ההפעלה (עבור מחולל דו"חות בלבד)

**CalcDu(entrance\_exit\_type, entrance\_exit\_net, entrance\_exit\_comparison)**

**שימוש:** השוואת משכים בין זוגות של כניסות ויציאות והחזרת שעות העבודה בניכוי חפיפה. לדוגמא משמש לניכוי משכי חפיפה בין ססיות לבין כניסות ויציאות רגילות לנוכחות.

**תחביר:**

1. **entrance exit type** – סוג כניסה ויציאה לחישוב המשך -  $0 = \text{כניסה/יציאה}$  רגילים,  $1 = \text{ק"פ}$ ,  $2 = \text{כוננות}$ .
2. **entrance exit net** – מספרי זוגות כניסות ויציאות לחישוב משך הנוכחות **נטו** בניכוי החפיפה.
3. **entrance exit comparison** – מספרי זוגות כניסות ויציאות לחישוב המשך **בהשוואה** לסעיף 2.  
הפונקציה תחזיר משך מחושב בהשוואה בין זוגות כניסות ויציאות בסעיף 3 בניכוי סעיף 2. המשך בניכוי החפיפה יוצג בשעות דקות ( ללא התחשבות בעגולים, הפסקות וכו' ).

**לדוגמא:**  $CalcDu(0, "5,6", "1,2,3,4")$



## המרת ערכי שעות (שעות דקות/עשרוני)

### TIME2DEC(Time\_Expression)

**שימוש:** הסבת שדה זמן להצגה עשרונית

**דוגמא:** TIME2DEC(4:45)=4.75

### DEC2TIME(Decimal\_Expression)

**שימוש:** הסבת שדה זמן עשרוני להצגת שדה זמן על בסיס שעות/דקות

**דוגמא:** DEC2TIM(2.9)=2:54

### TIME2MIN(Time\_Expression)

**שימוש:** הסבת שדה זמן לדקות. לצורך ביצוע פעולות וחישובים על שדות זמן

**דוגמא:** TIME2MIN(3:15)=195

### MIN2TIME(Time\_Expression)

**שימוש:** הסבת מספר המייצג דקות לשדה זמן HH:MM. לצורך ביצוע פעולות וחישובים על

שדות זמן

**דוגמא:** MIN2TIME(500)=8:20

### LROUND(Time\_Field,Rounding\_Interval,Grace)

**שימוש:** עיגול ערך בשעות לפי מרווח עיגול ונקודת עיגול

**תחביר:** Time\_Field - שדה שעות דקות סטנדרטי במערכת לביא

**Rounding\_Interval** - מרווח העיגול (התוצאה בדקות תהיה תמיד כפולה של ערך

זה)

**Grace** - עד לערך זה עיגול כלפי מטה (קטן שווה) מעל ערך זה כלפי מעלה

**דוגמא:** נניח כי השדה Period->Cat00 (מונה 0 בקובץ התקופתי) מכיל 172.11 שעות

התוצאה של הפונקציה LROUND(Period->Cat00,15,8) תהיה 172.15

## היעדרויות

### ABS\_NAME(ABS\_CODE)

**שימוש:** לקבלת שם/תאור קוד היעדרות  
**דוגמא:** ABS\_NAME(5) יחזיר את תאור קוד היעדרות מס' 5

### ABS\_SHRT(ABS\_CODE)

**שימוש:** הדפסת התיאור המקוצר של קוד היעדרות  
**תחביר:** ABS\_SHRT(ABS\_CODE\_NO)  
**דוגמא:** הדפסת התיאור המקוצר של ק. היעדרות מספר 20  
ABS\_SHRT(20) דו"ח תקופתי  
ABS\_SHRT(GETEXEPT("C1",8)) - קובץ קונפיגורציה - כרטיס שנתי YTD (GETEXEPT("C1",8))  
אין צורך בהוספת הפונקציה ABS\_SHRT בחוץ YTD יש להגדיר סוג שדה "קוד היעדרות".  
מספר קוד היעדרות ירשם בקובץ YTD כערך עשירי. תוכנית הדו"ח תתייחס לסוג השדה  
בחוץ YTD  
קוד היעדרות = ערך 4 ותשלוף את התיאור המקוצר של קוד היעדרות.

### ABS\_HOUR(Absence\_Code)

**שימוש:** לקבלת סה"כ שעות היעדרות מתוך הקובץ התקופתי לפי קוד היעדרות  
**דוגמא:** אם קוד היעדרות 5 = חופשה אז ABS\_HOURS(5) יחזיר את שעות החופשה.

### ABS\_DAYS(Absence\_Code)

**שימוש:** לקבלת סה"כ ימי היעדרות מתוך הקובץ התקופתי לפי קוד היעדרות  
**דוגמא:** אם קוד היעדרות 5 = חופשה אז ABS\_DAYS(5) יחזיר את ימי החופשה

### S\_ABS\_DC(Abs<sub>0</sub>,Abs<sub>1</sub>,...,Abs<sub>40</sub>)

**שימוש:** סיכום קודי היעדרות בשעות מקובץ תקופתי PERIOD. תוצאה: ערך עשירי  
**דוגמא:** S\_ABS\_DC(5,6,10)

### S\_ABS\_DD(Abs<sub>0</sub>,Abs<sub>1</sub>,...,Abs<sub>40</sub>)

**שימוש:** סיכום קודי היעדרות בימים מקובץ תקופתי PERIOD. תוצאה: בימים  
**דוגמא:** S\_ABS\_DD(5,6,10)

### S\_ABS\_HH(Abs<sub>0</sub>,Abs<sub>1</sub>,...,Abs<sub>40</sub>)

**שימוש:** סיכום קודי היעדרות בשעות מקובץ תקופתי PERIOD. תוצאה: בשעות  
**דוגמא:** S\_ABS\_HH(5,6,10)

### YEAR4ABS (Employee\_Number, Date, Absence\_Code)

**שימוש:** קבלת שנת המעקב לקוד היעדרות ולעובד מבוקש לכול שנה, קלנדארית ולשנה שאינה קלנדארית.

**תחביר:** Number\_Employee - מספר עובד  
Date - תאריך לשליפה

**Absence\_Code** - מספר הקוד שברצונך להציג את שנת המעקב שלו.

**דוגמא:** לקבלת תקופת המעקב עבור השנה הנוכחית לקוד היעדרות חופשה (2) לעובד המבוקש.

YEAR4ABS (EMPLOYEE->EMPLOYEE,ENDPP(),2)

לקבלת יתרת העובד מבוססת על תאריך תחילת העבודה של העובד המסוים.

BALANCE\_H(EMPLOYEE->EMPLOYEE,  
YEAR4ABS(EMPLOYEE->EMPLOYEE,ENDPP(),2)

**GETABSFL(EMPLOYEE,"ABS\_CODE,FIELD\_NO","YEAR)**

**שימוש:** פונקציה כללית לשליפת נתונים מקובץ ABSFOLLOW

**תחביר:** EMPLOYEE - מס' עובד

ABS\_CODE - מס' קוד היעדרות מטבלה H

FIELD\_NO - מס' השדה בקובץ ABSFOLLOW

YEAR - שנה

**דוגמא:** שליפת יתרה היסטורית בימים לשנה הנוכחית עבור קוד היעדרות מספר 2:

GETABSFL(EMPLOYEE->EMPLOYEE,"2,5",DIV\_LONG(PERIOD->PERIOD\_KEY,100))

שליפת יתרה היסטורית בימים לשנה הנוכחית קלנדארית עבור קוד היעדרות מספר 2:

GETABSFL(EMPLOYEE->EMPLOYEE,"2,5",  
YEAR4ABS(PERIOD\_EMPLOYEE,ENDPP()))

**GETABSPR(EMPLOYEE,"ABS\_CODE,FIELD\_NO","DATE)**

**שימוש:** פונקציה כללית לשליפת נתונים מקובץ ABSFOLLOW עבור נתונים המחושבים לתת-תקופות.

**תחביר:** EMPLOYEE - מס' עובד

ABS\_CODE - מס' קוד היעדרות מטבלה H

FIELD\_NO - מס' השדה בקובץ ABSFOLLOW

DATE - תאריך מסוים הנמצא בתת-התקופה המבוקשת (ניתן להשתמש גם ב-ENDPP)

**הערה:** תת-תקופה: בד"כ בשימוש בקודי היעדרות "תקשיריים" שתקופת המעקב עבורם היא פחות משנה, לדוגמא: לימודים/בחירות, מחלת ילד וכד'.

**BALANC\_D(Employee#,Year,Abs\_Code)**

**שימוש:** קבלת יתרת קוד היעדרות בימים מקובץ ABSFOLLOW

**תחביר:** Employee# - מס' עובד שיועבר כמחרוזת ASCII

Year - שנה עבורה נדרשת היתרה

Abs\_Code - קוד היעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא:** BALANC\_D(PERIOD->EMPLOYEE,1998,23)

BALANC\_D(PERIOD->EMPLOYEE,DIV\_LONG(PERIOD->PERIOD\_KEY,100),23)

**BALANC\_H(Employee#,Year,Abs\_Code)**

**שימוש:** קבלת יתרת קוד היעדרות בשעות מקובץ ABSFOLLOW

**תחביר:** Employee# - מס' עובד שיועבר כמחרוזת ASCII

Year - שנה עבורה נדרשת היתרה

Abs\_Code - קוד היעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא:** BALANC\_H(PERIOD->EMPLOYEE,YEAR4ABS(PERIOD->

>EMPLOYEE,ENDPP(),5),5

הצגת יתרה בשעות עבור קוד היעדרות 5 עבור שנה 5.

**ABSUSE\_H(Employee#,Year,Abs\_Code)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ ABSFOLLOW - בשעות

**תחביר: Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Year** - שנה עבורה נדרש הניצול  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא: ABSUSE\_H(PERIOD->EMPLOYEE,1998,23)**  
 ABSUSE\_H(PERIOD->EMPLOYEE, DIV\_LONG(PERIOD->PERIOD\_KEY,100),23)

**ABSUSE\_D(Employee#,Year,Abs\_Code)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSFOLLOW - **בימים**  
**תחביר: Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Year** - שנה עבורה נדרש הניצול  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא: ABSUSE\_D(PERIOD->EMPLOYEE,1998,23)**  
 ABSUSE\_D(PERIOD->EMPLOYEE, DIV\_LONG(PERIOD->PERIOD\_KEY,100),23)

**ABSTRX\_H(Employee#,Per\_Key,Abs\_Code)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX - **בשעות**  
**תחביר: Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Per\_Key** - תקופת צבירה עבורה נדרש הנתון  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא: ABSTRX\_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,4)**

**ABSTRX\_D(Employee#,Per\_Key,Abs\_Code)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX - **בימים**  
**תחביר: Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Per\_Key** - תקופת צבירה עבורה נדרש הנתון  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)

**דוגמא: ABSTRX\_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,4)**

**TRX\_R\_H(Employee#,Per\_Key,"Abs\_Code, Mode, Justification Code,...")**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד - **בשעות**

**תחביר: Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Per\_Key** - תקופת צבירה עבורה נדרש הנתון  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
**Mode** - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** קוד הסיבה שיוחזר, במקרה שהוגדר Mode=0 ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר Mode=1 ניתן להגדיר את טווח קודי הסיבה שיוחזר. יש "לעטוף" את פרמטרים 3 – הסוף בגרשיים.

**דוגמאות:** `TRX_R_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,"2,1,1001,1004")`  
`TRX_R_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,"2,0,1001,1002,9053,9054")`

**TRX\_R\_D(Employee#,Per\_Key,"Abs\_Code, Mode, Justification Code,...")**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד - בימים

**תחביר:** `Employee#` - מס' עובד שיועבר כמחרוזת ASCII  
`Per_Key` - תקופת צבירה עבורה נדרש הנתון  
`Abs_Code` - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
`Mode` - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** - קוד הסיבה שיוחזר, במקרה שהוגדר `Mode=0` ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר `Mode=1` ניתן להגדיר את טווח קודי הסיבה שיוחזר. יש "לעטוף" את פרמטרים 3 – הסוף בגרשיים.

**דוגמאות:** `TRX_R_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,"2,1,1001,1004")`  
`TRX_R_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,"2,0,1001,1002,9053,9054")`

**TRX\_REASON\_H(Employee#,Per\_Key,Abs\_Code, Mode, Justification Code,...)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד - בשעות

**תחביר:** `Employee#` - מס' עובד שיועבר כמחרוזת ASCII  
`Per_Key` - תקופת צבירה עבורה נדרש הנתון  
`Abs_Code` - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
`Mode` - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** - קוד הסיבה שיוחזר, במקרה שהוגדר `Mode=0` ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר `Mode=1` ניתן להגדיר את טווח קודי הסיבה שיוחזר.

`TRX_REASON_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,2,1,1001,1004)`

**דוגמאות:**

`TRX_REASON_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,2,0,1001,1002,9053,9054)`

**הערה:** הפונקציה `TRX_REASON_H` מאפשרת להגדיר ביטוי עבור קוד ההיעדרות בניגוד לפונקציה `TRX_R_H` שאינה מאפשרת זאת.

**לשימוש במערכת TKSQL בלבד !**

**TRX\_REASON\_D(Employee#,Per\_Key,Abs\_Code, Mode, Justification Code,...)**

**שימוש:** הצגת נתוני "ניצול לתקופה" מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד - בימים

**תחביר:** `Employee#` - מס' עובד שיועבר כמחרוזת ASCII  
`Per_Key` - תקופת צבירה עבורה נדרש הנתון  
`Abs_Code` - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
`Mode` - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** - קוד הסיבה שיוחזר, במקרה שהוגדר `Mode=0` ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר `Mode=1` ניתן להגדיר את טווח קודי הסיבה שיוחזר.

`TRX_REASON_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,2,1,1001,1004)`

**דוגמאות:**

`TRX_REASON_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD_KEY,2,0,1001,1002,9053,9054)`

**הערה:** הפונקציה **TRX\_REASON\_D** מאפשרת להגדיר ביטוי עבור קוד ההיעדרות בניגוד לפונקציה **TRX\_R\_D** שאינה מאפשרת זאת.

**לשימוש במערכת TKSQL בלבד !**

**TRX\_EX\_H(Employee#,Per\_Key,Abs\_Code,Transaction Type, Mode, Justification Code,...)**

**שימוש:** הצגת נתונים מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד – בשעות לכל סוגי הרשומות לפי בחירה  
**תחביר:** **Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Per\_Key** - תקופת צבירה עבורה נדרש הנתון  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
**Transaction Type** – סוג הרשומה (1=עדכון לניצול, 2=עדכון למכסה, 3=עדכון לזכאות,  
 12 = עדכון לניצול ולמכסה, 13=עדכון לניצול ולזכאות,  
 23=עדכון למכסה ולזכאות, 123=עדכון

לניצול למכסה ולזכאות )

**Mode** - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** קוד הסיבה שיוחזר, במקרה שהוגדר Mode=0 ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר Mode=1 ניתן להגדיר את טווח קודי הסיבה שיוחזר.

**דוגמאות** **TRX\_EX\_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,2,23,1,1001,1004)**  
**TRX\_EX\_H(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,2,23,0,1001,1002,9053,9054)**

**הערה:** הפונקציה **TRX\_EX\_H** מאפשרת להגדיר ביטוי עבור קוד ההיעדרות בניגוד לפונקציה **TRX\_R\_H** שאינה מאפשרת זאת.

**לשימוש במערכת TKSQL בלבד !**

**TRX\_EX\_D(Employee#,Per\_Key,Abs\_Code,Transaction Type, Mode, Justification Code,...)**

**שימוש:** הצגת נתונים מקובץ - ABSTRX עבור קודי סיבה מסויימים בלבד – בימים לכל סוגי הרשומות לפי בחירה  
**תחביר:** **Employee#** - מס' עובד שיועבר כמחרוזת ASCII  
**Per\_Key** - תקופת צבירה עבורה נדרש הנתון  
**Abs\_Code** - קוד ההיעדרות (עד 3 ספרות לפי טבלה H)  
**Transaction Type** – סוג הרשומה (1=עדכון לניצול, 2=עדכון למכסה, 3=עדכון לזכאות,  
 12 = עדכון לניצול ולמכסה, 13=עדכון לניצול ולזכאות,  
 23=עדכון למכסה ולזכאות, 123=עדכון

לניצול למכסה ולזכאות )

**Mode** - 0 = עד 20 קודי סיבה בודדים, 1 = טווח קודי סיבה  
**Justification Code** קוד הסיבה שיוחזר, במקרה שהוגדר Mode=0 ניתן להגדיר עד 20 קודי סיבה שונים מופרדים בפסיקים, במקרה שהוגדר Mode=1 ניתן להגדיר את טווח קודי הסיבה שיוחזר.

**דוגמאות** **TRX\_EX\_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,2,23,1,1001,1004)**  
**TRX\_EX\_D(EMPLOYEE->EMPLOYEE,PERIOD->PERIOD\_KEY,2,23,0,1001,1002,9053,9054)**

**הערה:** הפונקציה TRX\_EX\_D מאפשרת להגדיר ביטוי עבור קוד היעדרות בניגוד לפונקציה TRX\_R\_D שאינה מאפשרת זאת.

**לשימוש במערכת TKSQL בלבד !**

**ATTACHEXIST(Employee#, Date, From\_Abs\_Code, Till\_Abs\_Code)**

**שימוש:** הפונקציה בודקת האם קיימת אסמכתא בטבלת attach.  
**Employee** – מספר עובד (מחרוזת של 12 תווים)  
**Date** – תאריך מבוקש (מחרוזת של 8 תווים בפורמט CCYYMMDD)  
**From\_Abs\_Code** – מקוד היעדרות  
**Till\_Abs\_Code** – עד קוד היעדרות

הפונקציה מחזירה 1 אם קיימת אסמכתא לעובד, לתאריך ולקוד היעדרות מסוים או לכל קודי היעדרות, ומחזירה 0 אם לא קיימת אסמכתא.  
**דוגמא:** בדיקה האם קיימת אסמכתא לעובד מספר 1, בתאריך 31/12/2008, עבור קוד היעדרות 3.

```
ATTACHEXIST(DAILY->EMPLOYEE, DAILY->DATE,0,999)
ATTACHEXIST("1","20081231",3,3)
```

**לשימוש במערכת TKSQL בלבד !**

## מונים

**CAT\_NAME(Category Number)**

תאור מונה מתוך טבלה A  
**שימוש:** לקבלת תאור (שם) מונה שעות לפי מספר המונה  
**דוגמא:** כאשר רוצים להציג תאור של מונה מס' 7 CAT\_NAME(7)

**CAT\_NAME()**

**שימוש:** קבלת שם מונה  
**דוגמא:** CAT\_NAME(8)

**DAYCATDC()**

**שימוש:** סיכום ערכי מונים (כערכים עשרוניים) מקובץ יומי - DAILY  
**תחביר:** DAYCATDC(P1,P2,...,P49)  
**P** = מספר מונה

ניתן להוסיף סימן מינוס (-) מקדים לכל מונה למעט מונה מספר אפס (0 - רגילות)

**מקרה פרטי - טיפול בערך שלילי של מונה 0 - רגילות**  
 לצורך טיפול בערך שלילי של מונה 0 יש לבצע הכפלה של ערך מונה זה ב- (-1).  
**לדוגמא:** צבירת הערכים הבאים:

+ 1- X (מונה 0)

+ 1- X (מונה 1)

(מונה 2)

=====

מונה 0 - מונה 1 + מונה 2

(DAYCATDC(0)\*-1)+DAYCATDC(-1,2)

#### DAYCATHH()

**שימוש:** סיכום ערכי מונים (כערכי שעות/דקות) מקובץ יומי - DAILY  
**תחביר:** ראו פירוט עבור פונקציה DAYCATDC  
**דוגמא:**  $DAYCATHH(1,2,3)$  יסכם את מונים 1,2,3 מהקובץ היומי בשעות.  
 ניתן גם להחסיר מונים  $DAYCATHH(1,-2,3)$

#### PERCATDC()

**שימוש:** סיכום ערכי מונים (כערכים עשרוניים) מקובץ תקופתי - PERIOD  
**תחביר:** ראו פירוט עבור פונקציה DAYCATDC  
**דוגמא:**  $PERCATDC(1,2,3)$  יסכם את מונים 1,2,3 מהקובץ התקופתי בעשרוני.  
 ניתן גם להחסיר מונים

#### PERCATHH()

**שימוש:** סיכום ערכי מונים (כערכי שעות/דקות מקובץ תקופתי) - PERIOD  
**תחביר:** ראו פירוט עבור פונקציה DAYCATDC  
**דוגמא:**  $PERCATDC(1,2,3)$  יסכם את מונים 1,2,3 מהקובץ התקופתי בשעות דקות.  
 ניתן גם להחסיר מונים

#### TCDCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ TCDAY). תוצאה בשעות/דקות  
**דוגמא:**  $TCDCATF(0,1.50,2,2.00)$   
 מונה 0 X 1.5 + מונה 2 X 2.00

#### TCPCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ TCDAY). תוצאה בשעות/דקות  
**דוגמא:**  $TCPCATF(0,1.50,2,2.00)$   
 מונה 0 X 1.5 + מונה 2 X 2.00

#### TKDCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ DAILY). תוצאה בשעות/דקות  
**דוגמא:**  $TKDCATF(0,1.50,2,2.00)$   
 מונה 0 X 1.5 + מונה 2 X 2.00

#### TKPCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ PERIOD). תוצאה בשעות/דקות  
**דוגמא:**  $TKPCATF(0,1.50,2,2.00)$   
 מונה 0 X 1.5 + מונה 2 X 2.00

#### DADCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ DADAY). תוצאה בשעות/דקות  
**דוגמא:**  $DADCATF(0,1.50,2,2.00)$   
 מונה 0 X 1.5 + מונה 2 X 2.00



### DAPCATF(Cat,Factor,.....)

**שימוש:** פונקציה לסיכום מונים + הכפלה בפקטור (קובץ DAPER). תוצאה בשעות/דקות

**דוגמא:** DAPCATF(0,1.50,2,2.00)

מונה 0 X 1.5 + מונה 2 X 2.00

## DAILY יומיים

### ERRDESCR(Error#)

**שימוש:** שלילת תיאור למספר שגיאה נתון

**דוגמא:** Errdescr(Daily->Err\_Ex1) יחזיר את תיאור שגיאה למס' השגיאה המופיע ב-

DAILY

**הערה:** קובץ ERRORS.DBF חייב להיות פתוח

### ערכים המשמשים לפונקציות חריגים יומיים

בקובץ החריגים היומי 11 ווקטורים של חריגים. כל ווקטור מכיל את הנתונים הבאים:

Value Code	תאור
1. תחילה	- תחילת תחום החריג (שעות/דקות)
2. סיום	
3. בפועל	- הפרש בין סיום לתחילה בדקות
4. בפועל מעוגל	- ערכו המעוגל של התחום בדקות
5. ערך לתשלום	- כמה דקות מהתחום אושרו לתשלום ע"י תוכנית חישוב יומית
6. ערך להיעדרות	- כמה דקות חויבו כהיעדרות (אם הוגדר קוד היעדרות)
7. מציין לשכר	קובע איך טופל החישוב לשכר
	0 = לפי ההסכם
	1 = לא לתשלום
	2 = לתשלום ללא הגבלה
8. קוד היעדרות	- קוד היעדרות חוקי לפי טבלת קודי היעדרות
9. עדכון ידני	- ערך בדקות שהוזן ע"י המשתמש (דורס את חישובי המערכת)
10. סוג החריג	- קובע את מהות התחום:

Event Code	
Early Entry = 1	הקדמה בכניסה
Late Entry = 2	איחור בכניסה
Early Exit = 3	הקדמה ביציאה
Late Exit = 4	יציאה מאוחרת
Early Entry2 = 5	הקדמות ואיחורים עבור שמרות מפוצלות
Late Entry2 = 6	
Early Exit2 = 7	
Late Exit2 = 8	
Break = 9	- הפסקה
Meal = A	- ארוחה
Other = B1..B9	- אחר - אי נוכחות / "יציאות במהלך העבודה"

**C1..C9 = העדרויות חלקיות - לשימוש רק  
ברשומות המוגדרות כ"העדרויות חלקיות"  
היעדרות חלקית=99 בסוג פעילות.**

קוד שגיאה/חריג (11) קוד בן 3 ספרות שהוכנס באופן אוטומטי ע"י תוכנית החישוב

**GETEVENT("Event\_Code",Value\_Code)**

**שימוש:** שליפה וסיכום לפי "סוג חריג או תחום" הפונקציה סורקת את כל הווקטורים **ומסכמת** "ערך נדרש" ל"סוג חריג" **תחביר:** **Event\_Code** - סוג חריג (סעיף 10) ערכים 1 עד B. כדי לציין את כולם - C **Value\_Code** - קוד של הערך הנדרש: **1 = בפועל** - (בפונקציה יש לסמן ערך 1. המשמעות מקבילה ל- 3 ברשימה מעלה) **2 = בפועל מעוגל** - (בפונקציה יש לסמן ערך 2. המשמעות מקבילה ל- 4 ברשימה מעלה) **3 = ערך לתשלום** - (בפונקציה יש לסמן ערך 3. המשמעות מקבילה ל- 5 ברשימה מעלה)

**דוגמא:** הביטוי **GETEVENT("9",2)** יחזיר את סה"כ ההפסקות שהורדו לעובד ביום נתון בערך המעוגל

**GETEXEPT("Event\_Code",Value\_Code)**

**שימוש:** הפונקציה סורקת את כל הווקטורים ושולפת את הערך הראשון המתאים ל"סוג חריג" **תחביר:** **Event\_Code** - סוג חריג או תחום (שדה 10) ערכים 1 עד C9, כולל יציאות במהלך העבודה B1-B9 **Value\_Code** - קוד של הערך הנדרש: 1-9, 11 **דוגמא:** הביטוי **GETEXEPT("9",1)** יחזיר את התחלת תחום ההפסקה הראשונה שהורדה לעובד ביום נתון **דוגמא:** הביטוי **GETEXEPT("C1",8)** יחזיר את מספר קוד היעדרות בשורת החריגים הראשונה **דוגמא:** הביטוי **GETEXEPT("B2",8)** יחזיר את מספר קוד היעדרות המתייחסת לאינטרוול השני של אי נוכחות - יציאה במהלך העבודה

**הערה:** ההבדל בין פונקציה זו ל- **GETEVENT** היא בכך שהיא משמשת לשליפת ערך כלשהו, כאשר אין משמעות לסיכום, **לדוגמא**, שליפת ערך קוד העדרות, אין משמעות לסיכום מספר קוד העדרות.

**PUTEXEPT("Event\_Code",Value\_Code,Value)**

**שימוש:** הפונקציה סורקת את כל הווקטורים ומעדכנת את הערך הראשון המתאים ל"סוג חריג" **תחביר:** **Event\_Code** - סוג חריג (סעיף 10) ערכים 1 עד C9. **Value\_Code** - קוד של הערך הנדרש: 

1-2	התחלת/סיום תחום
7	מציין לשכר
8	קוד היעדרות
9	עדכון ידני

**דוגמא:** הביטוי PUTEXEPT("2",8,4) כתיבת קוד העדרות 4 לכל מי שנרשם לו איחור.  
**דוגמא:** הביטוי PUTEXEPT("C2",8,5) כתיבת הערך 5 בשדה קוד העדרות לחריג  
 "העדרות חלקית" בשורה שניה.  
**דוגמא:** הביטוי PUTEXEPT("B2",8,11) כתיבת הערך 5 בשדה קוד העדרות של האינטרוול  
 השני של אי נוכחות (יציאה במהלך העבודה)

**הערה:** בתוכנית "עדכון קבצים כללי" - "שדה לעדכון" לבחור שדה פנוי שיחושב מחדש  
 (לדוגמא: מונה קיזוז, קוד עדכון)

#### DAY\_ABS(Abs\_Code,Value\_Code)

**שימוש:** שליפה וסיכום לפי קוד היעדרות נתון. הפונקציה מחזירה ערך בדקות.

**תחביר:** Abs\_Code - קוד היעדרות חוקי לפי טבלת היעדרויות

Value\_Code - קוד של הערך הנדרש:

1 = ערך היעדרות

2 = ערך לתשלום

**דוגמא:** Day\_Abs(4,2) יתקבל סיכום של ערך לתשלום עבור קוד היעדרות מס' 4

**הערה:** השימוש בפונקציה הינו עבור וקטור חריגים בקובץ היומי בלבד.

#### DSEARCH(Field\_no\_in\_AbsVec,Value\_to\_be\_Found)

**שימוש:** חיפוש ווקטור בו קיים ערך נתון בשדה נתון. הפונקציה מחזירה מספר ווקטור.

**תחביר:** Field\_No\_in\_AbsVec - מספר שדה בווקטור ערך מ-1 עד 11

(- ערך שיש למצוא כמחרוזת תווים בין גרשיים

הפונקציה סורקת את כל הווקטורים ומחפשת "ערך נתון" במספר "שדה נתון"

**דוגמא:** מצא ווקטור חריגים שבו מוגדרת הפסקה

DSEARCH(10,"9")

מצא ווקטור חריגים בו מופיע קוד היעדרות 23

DSEARCH(8,"23")

#### DAY\_VEC(Vec\_No,Field\_No)

**שימוש:** הפונקציה מחזירה כל שדה שנבחר מתוך הווקטור לפי מספר הווקטור. הפונקציה

מחזירה ערך מספרי.

**תחביר:** Vec\_No - מספר ווקטור נתון שהתקבל מהפעלת פונקצית DSEARCH()

Field\_No - מספר שדה נתון (1 עד 11)

**הערה:** השימוש בפונקציה זו באמצעות פונקצית DSEARCH

#### DAYNAMEL(DAILY->DAY\_TYPE)

**שימוש:** שליפת תאור סוג יום לתאריך נתון בתוכנית יבוא/יצוא (יצוא).

**תחביר:** עבור שליפת תאור מקוצר של סוג היום - יש להשתמש בפונקציה DAYNAMES

עבור שליפת תאור סוג היום - יש להשתמש בפונקציה DAYNAMEL

**הערה:** בפרופיל היצוא יש להגדיר שדה אלפא (כמו שם עובד) ולהפעיל עליו את ביטוי היצוא.

#### GET\_NOTE(EMPLOYEE,DATE,Maximum\_Length)

**שימוש:** שליפת ההערות (מטבלת **note**) מה- **iBrowse** או **DAILY** והצגתם בדוח.  
**תחביר:** **GET\_NOTE** – מחזיר שדה **NOTE** מטבלת **NOTE**.

**EMPLOYEE** – מספר עובד

**DATE** – תאריך מבוקש

**Maximum\_Length** – מספר תווים להצגת ההערה, מקסימום עד 256 תווים.

**דוגמא:** **TRIM(GET\_NOTE(DAILY EMPLOYEE,->DAILY-> DATE,80))**

פונקציה זו קיימת בגרסת **SQL** בלבד.

#### **GETTRANSMIT(EMPLOYEE,DATE)**

**שימוש:** שליפת נתונים מטבלת **tkclock** לצורך זיהוי אישור הממונה ב- **iBrowse**. באמצעות הפונקציה ניתן לדעת האם הנתונים הזמניים של העובד בתאריך מסויים, אושרו על ידי הממונה והועברו ל-**daily**. הנתונים יוצגו בדו"חות היומיים.  
**EMPLOYEE** = מספר עובד.  
**DATE** = תאריך.

הפונקציה מקבלת כפרמטר **מספר עובד ותאריך** ומחזירה אחת מהאפשרויות הבאות:  
0 – נתוני העובד לאותו היום לא אושרו והועברו ל- **Daily**.  
1 – נתוני העובד לאותו היום אושרו והועברו ל- **Daily**.  
(-1) – לא נמצאה רשומה עבור העובד לאותו היום.

#### **דוגמא:**

**GETTRANSMIT(DAILY->EMPLOYEE, DAILY->DATE)**

**GETTRANSMIT("1","20081231")**

הפונקציה תחזיר האם נתוני העובד אושרו והועברו ל-**daily** לאו (0 / 1 / -1) עבור נתוני עובד מספר 1 בתאריך 31/12/2008.

## נתונים תקופתיים

### **OTHERPERVAL (PeriodAddition, FieldNumber)**

**שימוש:** שלילת נתון מתקופה אחרת (לא נוכחית) מטבלת Period.  
**PeriodAddition** = מספר תקופות קדימה (חיובי), אחורה (שלילי), אפס – תקופה נוכחית.  
**FieldNumber** = מספר שדה מטבלת PERIOD.

#### דוגמא:

כאשר התקופה הנוכחית 200905

OTHERPERVAL (-12,75)

הפונקציה מחזירה עבור תקופה 200805 ערך של שדה 75 (Cat01).

OTHERPERVAL (1,17)

הפונקציה מחזירה עבור תקופה 200906 ערך של שדה 17 (Wage\_Total).

#### הערה:

אם רשומת ה-PERIOD הרצויה אינה קיימת, הפונקציה מחזירה אפס או מחרוזת ריקה לפי סוג שדה המבוקש.

לשימוש במערכת TKSQL בלבד !

## נתונים שנתיים

### **GETANUAL(Subject\_No,Field\_No)**

**שימוש:** שלילת נתונים מקובץ ANNUAL - שימוש בדו"חות תקופתיים

#### תחביר:

**Subject** - מס' נושא

**Field\_Num** - מס' שדה בקובץ (ערכים אפשריים: 0 - אפס, או מספר השדה בקובץ)

הפונקציה מקבלת את ערכי קוד התקופה (שנה ומספר תקופה) עפ"י התקופה המורצת באמצעות הדו"ח ופונה לקובץ ANNUAL עפ"י מספר הנושא ומספר השדה המבוקשים.

כאשר השדה מוגדר עם ערך 0 - הפונקציה תעביר באופן אוטומטי את הערך המחושב לתקופת הדו"ח - ובכך תאפשר חיסכון תפעולי.

כאשר מספר השדה = אפס, הפונקציה מבצעת חישוב אוטומטי של מספר השדה לשליפה ע"י הוספה של ערך קבוע 12 למספר התקופה (היסט של 12 - הגעה למערך התקופות בקובץ). באופן זה ניתן לשלוף את הערכים המחושבים לתקופת הדיווח ללא צורך בעדכון הפונקציות בדו"חות המיוחדים.

**לדוגמא:** תקופת הדו"ח היא 02/2000. השדה המוגדר בפונקציה = 0. התוכנית תחשב 02 + 12 ותשלוף את הערך בשדה מספר 14 בקובץ (ANNUAL.PERIOD\_02).

אם הדו"ח מופק לתקופה 03/2000 והפונקציה שנרשמה היא GETANUAL(100,0), אז הדו"ח יהיה מדפיס את ערך שדה מספר 15 (ANNUAL.PERIOD\_03) לנושא מספר 100 לשנת 2000.

אותה תוצאה היתה מתקבלת אם הפונקציה היתה נכתבת בצורה זו: GETANUAL(100,15)

**דוגמא:** GETANUAL(100,0)  
GETANUAL(200,7)

## SALDAY נלווים יומי

**GETSALD(Code)**

**שימוש:** הפונקציה מחזירה את ערך הקוד בקובץ SALDAY - נלווים לשכר יומי  
**דוגמא:** GETSALD(200)

## תעריפים

**GET\_RATE(Code,Date)**

**שימוש:** שלילת קוד תעריף לתאריך מסוים  
**תחביר:** Code – מספר קוד תעריף מטבלת קודי תעריף.  
Date "CCYYMMDD"  
**הערה:** בדרך כלל אין שימוש בפונקציה זו כשלעצמה. הפונקציה מופעלת ע"י הפונקציות ORDERATE וכו'.

**ORDERATE(N,EndPP())**

**שימוש:** קבלת ערך תעריף להזמנת עבודה (ORDER) – בהעברה לשכר בלבד - תמחיר משימות  
**תחביר:** N - מס' התעריף הנדרש, ערכים 1 עד 3 : 1 - תעריף עלות, 2 - תעריף חיוב, 3 - תעריף (3)  
**ENDPP()** - פונקציה פיקטיבית ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את התעריף הנכון לסוף התקופה.

**PARTRATE(N,EndPP())**

**שימוש:** קבלת ערך תעריף לחלק (PART) – בהעברה לשכר בלבד - תמחיר משימות  
**תחביר:** N - מס' התעריף הנדרש, ערכים 1 עד 3 : 1 - תעריף עלות, 2 - תעריף חיוב, 3 - תעריף (3)  
**ENDPP()** - פונקציה פיקטיבית ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את התעריף הנכון לסוף התקופה

**BATRATE(N,EndPP())**

**שימוש:** קבלת ערך תעריף לאצווה (BATCH) – בהעברה לשכר בלבד - תמחיר משימות  
**תחביר:** N - מס' התעריף הנדרש, ערכים 1 עד 3 : 1 - תעריף עלות, 2 - תעריף חיוב, 3 - תעריף (3)  
**ENDPP()** - פונקציה פיקטיבית ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את התעריף הנכון לסוף התקופה.

### PROCRAATE(N,EndPP())

**שימוש:** קבלת ערך תעריף לתהליך (PROCESS) – בהעברה לשכר בלבד - תמחיר משימות

**תחביר:** N- מס' התעריף הנדרש, ערכים 1 עד 3 : 1 - תעריף עלות, 2 - תעריף חיוב, 3 - תעריף (3)

**ENDPP()** - פונקציה פיקטיבית ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את התעריף הנכון לסוף התקופה

### DPTRATE(N,EndPP())

**שימוש:** קבלת ערך תעריף למחלקה (DPARTMENT) – בהעברה לשכר - בלבד - תמחיר מחלקתי

**תחביר:** N- מס' התעריף הנדרש, ערכים 1 עד 3 : 1 - תעריף עלות, 2 - תעריף חיוב, 3 - תעריף (3)

**ENDPP()** - פונקציה פיקטיבית ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את התעריף הנכון לסוף התקופה

## הסכמים וטבלאות

### TAB\_NAME("Table\_Name",Table\_Number)

**שימוש:** לקבלת שם טבלה

**דוגמא:** TAB\_NAME("CONTRACT",VAL(GETCONTR(ENDPP)))

150, TAB\_NAME("TABLEF"), יחזיר את שם טבלת שעות נוספות שמספרה 150

### GETCONTR(DATE())

### GETCONTR(ENDPP())

**שימוש:** שליפת מספר הסכם לתאריך נתון (לתאריך נתון - שימוש ב-DATE, לסוף תקופה - ENDPP)

**לדוגמא:** DATE

לקבלת ההסכם שהוא בתוקף היום - תאריך מערכת ההפעלה - GETCONTR(DATE())

לקבלת ההסכם שבתוקף ביום מסוים לדוגמא "31.12.1998" GETCONTR(STOD("19981231"))

**דוגמא:** ENDPP()

לקבלת הסכם שהוא בתוקף בתקופת שכר מסוימת GETCONTR(ENDPP())  
ENDPP פונקציה ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את ההסכם הנכון לסוף התקופה

### GETFIELD("FILE\_NAME,KEY\_FIELD,RETURN\_FIELD,",VALUE)

### גרסת 32 ביט + SQL !!!

**שימוש:** שליפת נתונים מקובץ לפי שדה מפתח

**תחביר:** FILE\_NAME - שם הקובץ שממנו ישלפו הנתונים

KEY\_FIELD - שדה המפתח שיזהה את הרשומה שתוחזר

RETURN\_FIELD - שם השדה שיוחזר

VALUE - הערך שהתכנית תחפש בשדה המפתח.

הערך בשדה VALUE חייב להיות זהה בסוג ובאורך לשדה KEY\_FIELD

**דוגמא:** שליפת שם של קוד מיון 2, לפי מספר הקוד ששוין לעובד בקובץ העובדים-  
GETFIELD("SORT2,NUMBER,NAME,"EMPLOYEE->SORT2)  
מס' SORT2 המשויך לעובד ייבדק מול שדה NUMBER בקובץ SORT2, ויוחזר  
השדה NAME בהתאמה.

#### הערות :

- הפונקציה נתמכת בתצורה זו בגרסאות 32 ביט ו-SQL
- בגרסת 32 ביט – חובה להשתמש באותיות גדולות (UPPERCASE)
- יש לשים לב "לעטוף" את שלושת הערכים המזהים הראשונים במרכאות ולהוסיף סימן פסיק לפני המרכאות הסוגרות.
- הפונקציה תחזיר ערך מהסוג והאורך של שדה RETURN\_FIELD

#### מגבלות בגרסת 32 ביט :

- בגרסת 32 ביט ניתן להחזיר ערכים רק כאשר הוגדר שדה KEY\_FIELD שהוא שדה האינדקס של הקובץ שממנו מחזירים נתונים. שדה KEY\_FIELD חייב להיות שדה מפתח חד-חד-ערכי.
- בגרסת 32 ביט ניתן להחזיר נתונים רק מקבצים שנפתחים על ידי התכנית שבה משתמשים
- נכון ל-20.05.2004 ניתן להשתמש בפונקציה בדוחות נוכחות יומיים, תקופתיים ודוחות עובדים (תכניות RSYSTEM, RPERIOD, RDAILY) בלבד.
- בגרסת 32 ביט ניתן להחזיר ערכים מהקבצים הבאים בלבד :  
ACTIVITY, AUTOLEVEL1, AUTOLEVEL2, BANDS, BASES, CCNTR, CELL, CITIES, CLASS, DACAT, DAKEY, DPARTMNT, MACHINE, MARITAL, ORDER, SITES, SORT0, SORT1, SORT2, PLANTS, PROFS, TABLEB.

#### גרסת SQL :

- בגרסת SQL מגבלות אלו אינן קיימות

**GETFIELD("FILE\_NAME","KEY\_FIELD","RETURN\_FIELD",VALUE)**

#### **תצורה חלופית - לשימוש בגרסת SQL בלבד !!!**

**שימוש:** שליפת נתונים מטבלה לפי שדה מפתח  
**תחביר:** FILE\_NAME – שם הטבלה שממנה ישלפו הנתונים  
KEY\_FIELD – שדה המפתח שיזהה את הרשומה שתוחזר  
RETURN\_FIELD – שם השדה שיוחזר  
VALUE – הערך שהתכנית תחפש בשדה המפתח.  
הערך בשדה VALUE חייב להיות זהה בסוג ובאורך לשדה KEY\_FIELD

**דוגמא:** שליפת שם של קוד מיון 2, לפי מספר הקוד ששוין לעובד בקובץ העובדים-  
GETFIELD("SORT2","NUMBER","NAME",EMPLOYEE->SORT2)  
מס' SORT2 המשויך לעובד ייבדק מול שדה NUMBER בקובץ SORT2, ויוחזר  
השדה NAME בהתאמה.  
שליפת שם של אצוה לפי מספר קוד מיון 2 שהוגדר לעובד  
GETFIELD("BATCH","NUMBER","NAME",STR(VAL(EMPLOYEE->SORT2),20,0))

#### הערות :

- מבנה זה של הפונקציה נתמך בגרסת SQL בלבד. מערכת SQL תומכת במבנה זה ובמבנה של גרסת 32 ביט.
- תיאורטית הפונקציה מסוגלת לשלוף כל נתון מכל טבלה במערכת ולקבל כל אחד מהשדות בטבלה כשדה המזהה לרשומה.
- ניתן להשתמש בפונקציה מכל תכנית במערכת.



- אם הוגדר שדה KEY\_FIELD שאינו שדה אינדקס בטבלה וישנן מספר רשומות שמתאימות לערך שהוגדר בשדה VALUE, הפונקציה תחזיר ערך מאחת הרשומות באופן אקראי.
- הפונקציה תחזיר ערך מהסוג והאורך של שדה RETURN\_FIELD

## קובץ\טבלת עובדים

PRT\_TIME(Date())  
PRT\_TIME(ENDPP())

**שימוש:** שליפת אחוז משרה לעובד לתאריך נתון (לתאריך נתון - שימוש ב-DATE, לסוף תקופה - ENDPP)

**תחביר:**

**לדוגמא DATE:**

לקבלת אחוז המשרה שהוא בתוקף היום - תאריך מערכת ההפעלה  
PRT\_TIME(Date())  
לקבלת אחוז המשרה שבתוקף ביום מסוים לדוגמא 31.12.1998  
PRT\_TIME(STOD("11981231"))

**לדוגמא ENDPP():**

לקבלת אחוז המשרה שהוא בתוקף בתקופת שכר מסוימת  
PRT\_TIME(ENDPP())

**הערה:** ENDPP פונקציה ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר. התוכנית שולפת את אחוז המשרה הנכון לסוף התקופה.

EMP\_ABS(Date())  
EMP\_ABS(ENDPP())

**שימוש:** שליפת קוד היעדרות לעובד לתאריך נתון (לתאריך נתון - שימוש ב-DATE, לסוף תקופה - ENDPP)

**תחביר:** לדוגמא DATE:

לקבלת קוד היעדרות שהוא בתוקף היום - תאריך מערכת ההפעלה  
EMP\_ABS(Date())  
לקבלת קוד היעדרות שבתוקף ביום מסוים  
EMP\_ABS(STOD("11981231")) 31.12.1998 **לדוגמא**

**לדוגמא ENDPP():**

לקבלת קוד היעדרות שהוא בתוקף בתקופת שכר מסוימת  
EMP\_ABS(ENDPP())

ENDPP פונקציה ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר.

התוכנית שולפת את קוד היעדרות הנכון לסוף התקופה

פונקציה זו מתייחסת לשדה קוד היעדרות בשתי הגרסות. בגרסת bit 32 פונקציה זו מתייחסת גם לסטאטוס.

EMP\_STAT(Date())

## EMP\_STAT(ENDPP())

**שימוש:** שלילת סטאטוס לעובד לתאריך נתון (לתאריך נתון - שימוש ב-DATE , לסוף תקופה - ENDPP)

**תחביר: לדוגמא DATE:**

לקבלת סטאטוס שהוא בתוקף היום - תאריך מערכת ההפעלה

EMP\_ABS(ABS(EMP\_DATE))

**לקבלת סטאטוס שבתוקף ביום מסוים**

EMP\_ABS(STOD("11981231")) 31.12.1998 μα ΔΙ

**:ENDPP()** αμφισβητούμενο

לקבלת סטאטוס שהוא בתוקף **בתקופת שחר** מסוימת EMP\_ABS(ENDPP())

ENDPP פונקציה ההופכת את מספר התקופה וקוד הצבירה לתאריך סוף תקופת השכר.

## התוכנית שולפת את הסטאטוס הנכון לסוף התקופה

פונקציה זו קיימת בגרסת SQL בלבד, כיוון שבה יש הפרדה בין השדה סטאטוס לקוד היעדרות.

**EMPFLEX**

ישנן 4 פונקציות המתייחסות ל- EMPFLEX שההבדל ביניהן הוא סוג הנתון :

EMPFLEXINT(EMPLOYEE->EMPLOYEE, FIELD\_NUM) - **נומרי שלם**

EMPFLEXDEC(EMPLOYEE->EMPLOYEE, FIELD NUM) - **נומרי עשרוני**

EMPFLXSTR(EMPLOYEE->EMPLOYEE.FIELD NUM) - **מחזורת**

תאריך - EMPFLEXDATE(EMPLOYEE->EMPLOYEE, FIELD NUM)

כך שיש לבדוק מה סוג הנתון ולפיו לבחור את הפונקציה לשליפה .

לדוגמה אם השדה עשירני יש להשתמש בפונקציה שמחזירה ערך עשירני :

EMPFLEXDEC(EMPLOYEE-&gt;EMPLOYEE,13)

במידה והפונקציה מנוהלת היסטורית ב- EMPLOG אז מתווסף פרמטר נוסף לפונקציה והוא תאריך:

EMPFLEXINT(EMPLOYEE->EMPLOYEE, FIELD NUM, date) - נומרי שלם

EMPFLEXDEC(EMPLOYEE->EMPLOYEE, FIELD NUM, date) - נומרי עשרוני

EMPFLXSTR(EMPLOYEE->EMPLOYEE, FIELD\_NUM, date) - מחזורת

EMPFLXDATE(EMPLOYEE->EMPLOYEE, FIELD NUM, date) - תאריך

לדוגמא:

```
empflexdec(employee->EMPLOYEE,14,'20060401')
```

```
empfexdec(employee->EMPLOYEE,14,STR(PERIOD-14,10,0)
>PERIOD KEY,6,0)+'01')
```

**CHILD COUNT(ITEMTYPE,EMPLOYEE,AGE,DATE)**

**שימוש:** קבלת מספר ילדים **עד** גיל מבוקש נכון לתאריך.

**תחביר: ITEMTYPE** – מספר הווקטור ממנו יש להחזיר את מספר הילדים, כפי שהוגדר

בכרטיס עובד.

5 – וקטור ילדים (Children)      9 – וקטור ילדים ובני משפחה (family)

**EMPLOYEE** – מספר העובד הרצוי.  
**AGE** – ערך מספרי המשקף את הגיל המבוקש לחישוב מספר הילדים.  
**DATE** – התאריך הרצוי  
**דוגמא:** לקבלת מספר הילדים של עובד 1, עד גיל חמש, נכון לתאריך 31/03/2004 כפי שמוגדר בווקטור 9.  
 CHILD\_COUNT(9,1,5,20040331)

פונקציה זו קיימת בגרסת SQL בלבד.

## חדרי אוכל

**DND\_UNIT(Meal-n,Meal-n1,...,Meal-n20)**

**שימוש:** סכימת מספר יחידות של ארוחות - קובץ ארוחות יומי - DNDAILY  
**תחביר:** DND\_UNIT(Meal-n,Meal-n1,...,Meal-n20)  
**דוגמא:** DND\_UNIT(1,2,20)

**DND\_EMPL(Meal-n,Meal-n1,...,Meal-n20)**

**שימוש:** סכימת חיובי עובד עפ"י ארוחות - קובץ ארוחות יומי - DNDAILY  
**תחביר:** DND\_EMPL(Meal-n,Meal-n1,...,Meal-n20)  
**דוגמא:** DND\_EMPL(1,2,20)

**DND\_SITE(Meal-n,Meal-n1,...,Meal-n20)**

**שימוש:** סכימת חיובי אתר/קבלן עפ"י ארוחות - קובץ ארוחות יומי - DNDAILY  
**תחביר:** DND\_SITE(Meal-n,Meal-n1,...,Meal-n20)  
**דוגמא:** DND\_SITE(1,2,20)

כל הפונקציות היומיות הנ"ל מתייחסות לרשומה הראשונה בקובץ חד"א לעובד/תאריך  
 בשלב זה 6.11.2001 לא ניתן להחיל פונקציות יומיות בדו"חות מיוחדים יומיים חד"א

**DNP\_UNIT(Meal-n,Meal-n1,...,Meal-n20)**

**שימוש:** סכימת מספר יחידות של ארוחות - קובץ ארוחות תקופתי - DNPERIOD  
**תחביר:** DNP\_UNIT(Meal-n,Meal-n1,...,Meal-n20)  
**דוגמא:** DNP\_UNIT(1,2,20)

**DNP\_EMPL(Meal-n,Meal-n1,...,Meal-n20)**

**שימוש:** סכימת מספר יחידות של ארוחות - קובץ ארוחות תקופתי - DNPERIOD  
**תחביר:** DNP\_EMPL(Meal-n,Meal-n1,...,Meal-n20)  
**דוגמא:** DNP\_EMPL(1,2,20)

**MEAL\_FOUND(Att\_Pair\_Strat, Att\_Pair\_End, Meal\_Site, Diner\_type, Meal\_Number\_N...1-25)**

**שימוש:** הפונקציה מאפשרת לדעת האם העובד אכל בטווח הכניסות והיציאות או מחוץ לטווח הכניסות והיציאות.

הפונקציה מחזירה:

**TRUE** – שעת הארוחה בתוך טווח הכניסות והיציאות או

**FALSE** – שעת הארוחה מחוץ לטווח הכניסות והיציאות.

**Att\_Pair\_Strat** = זוג ראשון.

**Att\_Pair\_End** = זוג אחרון. לבדיקת כל הזוגות יש לכתוב 0 בשני הפרמטרים

הראשונים. לא ניתן לכתוב 0 רק באחד הפרמטרים.

**Meal\_Site** = מספר האתר בו בוצעה הארוחה כפי שמופיע בטבלת הארוחות היומית.

**Diner\_Type** = סוג הסועד כפי שמופיע בטבלת הארוחות היומית.

**Meal\_Number\_N** = שדה גמיש. מספר הארוחה כפי שמופיע בטבלת הארוחות

היומית (1-25). ניתן להכניס את מספרי הארוחות הרצויים לבדיקה, ע"י הפרדה עם

פסיקים. לבדיקת כל הארוחות יש לכתוב 0.

\*לכל אחד מהמצבים ניתן לכתוב פרמטר 0 המתייחס למצב ALL.

**דוגמא:** בדיקה אם יש ארוחה כלשהי בתוך טווח הנוכחות.

התייחסות לכל זוגות הנוכחות, כל סוגי האתרים, כל סוגי הסועד וכל הארוחות.

**MEAL\_FOUND(0,0,0,0,0)**

בדיקה אם יש ארוחות 4, 6, 14, 25 בזמן הנוכחות של זוגות 1 עד 3 בסוג אתר 2 ולסוג סועד 1.

**MEAL\_FOUND(1, 3, 2, 1, 4, 6, 14, 25)**

### הערות:

1. ארוחות ללא שעה לא יבדקו.

2. הבדיקה מתבצעת לפי שעת הארוחות בלבד (גם כאשר אין ערך בשדה כמות).

3. הפונקציה לא תומכת בפריטים EXP\_UNITS ממערכת חד"א.

4. כאשר עובדים עם טבלה D בסיום חישוב ארוחות, כל הכמויות מסתכמות בשורת הארוחות הראשונה באותו יום.

### DNENDPP()

**שימוש:** שלילת תאריך סוף תקופה לשימוש בטבלת DNPERIOD.

### PERFRDINVAL(FieldNumber)

**שימוש:** שלילת שדה מרשומה תקופתית Period עבור ביטויי חדרי אוכל DNPER לצורך העברות שכר, יצוא/בוא וכד'.

**FieldNumber** = מס' השדה אותו אנו מעוניינים לשלוף מטבלת Period.

**דוגמא:** העברה לשכר מחדרי האוכל רק עבור עובדים שעבדו מעל מספר ימי עבודה מסויים

בחודש. שדה מספר ימי עבודה בפועל אינו קיים ב-DNPER. לכן הפונקציה מקבלת מ-

DNPER מספר עובד, קוד תקופה וקוד צבירה, ושולפת לפיהם את השדה המבוקש מה-Period.

### הערה:

1. הפונקציה מחזירה סוגי ערכים שונים בהתאם לסוג שדה המקור המבוקש.

## משימות

**TCDEF\_Level\_No(Employee,DTOS(DATE()),Row\_Number**

**שימוש:** שליפת מספר רמת הדיווח, מקובץ ברירות מחדל TCDEF, הנכונה לתאריך המבוקש  
**תחביר:** TCDEF\_Level\_No( מספר הרמה לשליפה TCDEF\_1 = הזמנה ORDER  
TCDEF\_2 = אצווה BATCH  
TCDEF\_3 = חלק PART  
TCDEF\_4 = תהליך PROCESS  
EMPLOYEE - מספר העובד - יש לקשר מקובץ עובדים  
DATE() - לתאריך המערכת = תאריך נוכחי, או כל שדה תאריכי אחר  
Row\_Number - מספר השורה בקובץ ברירות מחדל TCDEF (עד 10 שורות)

**דוגמא:** נדרש לשלוף את מספר "תהליך" (רמה 4) המוגדר בשורה השלישית של קובץ  
"משימות ברירות מחדל" לתאריך הנוכחי

TCDEF4(EMPLOYEE->EMPLOYEE,DTOS(DATE()),3)

**הערה:** שימוש בפונקציה זו מחייב קישור לקובץ עובדים EMPLOYEE - כקובץ ראשי.

**TCDAYORDER(In ,Out)**

**שימוש:** שליפת נתונים מטבלת המשימות TCDAY, עבור טווח שעות מסויים. הנתונים מומרים לערך מספרי.

In = שעת כניסה

Out = שעת יציאה

**דוגמא:** TCDAYORDER(8.00,17.00)

הפונקציה תחזיר את מספר המשימה המתאימה לשעות הנוכחות מ 8.00 ועד 17.00.

### הערות:

1. ברשומות המשימה השייכות לטווח השעות של סוג הדיווח הנבדק (כניסה/יציאה, כניסה ק"פ יציאה ק"פ), קיימת רק הזמנה אחת. יתכנו מספר רשומות משימה לזוג אחד אבל בכול מספר הזמנה זהה.

במצב בו קיימות מספר רשומות הזמנה בטווח השעות הנבדק, יוחזר הערך מהרשומה הראשונה.

2. טווח הפעולה של הפונקציה **לא כולל** משימות ברירת מחדל, אם בבדיקה מעורבת רק משימת ברירת מחדל הפונקציה תחזיר ערך מזהה לשגיאה (-1). הבדיקה מתבצעת רק על משימות שיש להן שעת התחלה ושעת סיום,

3. יתכן כי בחישוב ראשון נתוני המשימה אינם מלאים, ולכן לצורך קבלת ערכים באמצעות הפונקציה נדרש חישוב חוזר.

**לשימוש במערכת TKSQL בלבד !**

**GET\_NOTE\_TC(EMPLOYEE,DATE,SEQ\_NO,Maximum\_Length)**

**שימוש:** שליפת ההערות מהמשימות היומיות (מטבלת job\_notes). הנתונים יוצגו בדו"חות משימה יומיים.

EMPLOYEE = מספר עובד.

DATE = תאריך.

SEQ\_NO = מספר רשומה.

Maximum\_Length = מספר תווים להצגת ההערה, מקסימום 256 תווים.

**דוגמא:**

GET\_NOTE\_TC(" 1","20081231", 25, 100)

הצגת הערות עבור עובד מספר 1, לתאריך 31/12/2008, לרשומת משימה 25, אורך הערה 100 תוים.

**TCENDPP()**

**שימוש:** שלילת תאריך סוף תקופה לשימוש בטבלת TCPER.

**נתונים מטבלת EMPLOG**

**EMPLGDEC(EMPLOYEE,GROUPCODE,PERIOD)**

**שימוש:** שלילת ערך שדה מספר 7 - MANUPD\_DEC - מקובץ EMPLOG  
**תחביר:** **EMPLOYEE** מספר עובד (מקובץ עובדים)

<b>GROUPCODE</b>	מספר נושא
<b>PERIOD</b>	תקופה נוכחית
0	תקופה קודמת, 2- שתי (2) תקופות קודמות וכן הלאה
1-	

אין מתבצע החיפוש של רשומה מבוקשת בהתייחס לשדה התקופה:  
מציאת הרשומה הנכונה לתקופה המבוקשת:

- (1) מצא תקופה נוכחית עפ"י תאריך המחשב
- (2) חשב את מספר התקופה המבוקשת עפ"י הערך בפרמטר PERIOD (1- וכד')
- (3) מצא את הרשומה הנכונה לתקופה המבוקשת

**דוגמא:** תאריך נוכחי 21.10.2001

רשומות קיימות בקובץ ל- 10.2000  
10.2001 - ו

- (1) התקופה הנוכחית = 10.2001 (עפ"י תאריך המחשב)
- (2) התקופה המבוקשת הכתובה בנוסחה היא 1-, כלומר מבוקשת רשומה התקפה

- (3) לתקופה 9.2001 ( 10.2001 - 1 ) = 9.2001  
בקובץ קיימת רשומה לתאריך 10.2000  
לפיכך תקופה קודמת = מ- 10.2000 עד 9.2001 - התקופה המבוקשת

לשליפת נתונים

**דוגמא:** **EMPLGDEC(EMPLOYEE->EMPLOYEE,101,0)**

**EMPLGNUM(EMPLOYEE,GROUPCODE,PERIOD)**

**שימוש:** שלילת ערך שדה מספר 9 - MANUPD\_NUM - מקובץ EMPLOG  
**תחביר:** ראו פירוט בפונקציה EMPLGDEC  
**דוגמא:** כנ"ל

**EMPLGCOD(EMPLOYEE,GROUPCODE,PERIOD)**

**שימוש:** שלילת ערך שדה מספר 10 - MANUPD\_COD - מקובץ EMPLOG  
**תחביר:** ראו פירוט בפונקציה EMPLGCOD  
**דוגמא:** כנ"ל

## נתונים מטבלת EMPVEC

לשימוש במערכת TKSQL בלבד !

**EMPVECDATE(VectorType, Date [,ColumnNumber][,SequenceNumber])**

**שימוש:** שלילת נתוני **תאריכים** מטבלת EMPVEC – TKSQL בלבד !!!  
הפונקציה סורקת את כל הווקטורים ומחזירה את הערך שהתבקש.

**תחביר:** VectorType - סוג הווקטור

Date - תאריך הנתון

ColumnNumber - מספר סוג השדה לשליפה, אם לא סופק ערך – ברירת המחדל

היא 1.

SequenceNumber - מספר המופע לשליפה, אם לא סופק ערך – ברירת המחדל

היא 1 או הערך העדכני לתאריך שסופק, לפי סוג הנתון שהתבקש.

סוגי הווקטורים:

1. קוד היעדרות
2. חלקיות משרה
3. טבלת אחוזי משרה GZ
4. בונסים (תוספות) למונים
5. שמות ילדים (children)
6. קודי היעדרות - מכסות ברירת מחדל
7. תאריכים - נתוני תקופות תקשי"ר
8. ערכי ברירת מחדל לנושאים שנתיים
9. שמות ילדים ובני זוג (family)
10. קבוצת כיסוי (עבור APM)
11. בקשת היעדרות

מספר סוג השדה:

ColumnNumber	שם שדה	סוג שדה
1	DATEFROM	DATETIME
2	DATETILL	DATETIME

**דוגמא:** שלילת תאריך תחילת קוד היעדרות הנוכחי - EmpVecDate(1,Date())

שלילת תאריך סיום קוד היעדרות הנוכחי - EmpVecDate(1,Date(),2)

**EMPVECDEC(VectorType, Date [,ColumnNumber][,SequenceNumber])**

**שימוש:** שלילת נתוני **ערכים עשרוניים** מטבלת EMPVEC – TKSQL בלבד !!!  
הפונקציה סורקת את כל הווקטורים ומחזירה את הערך שהתבקש.

**תחביר:** VectorType - סוג הווקטור

Date - תאריך הנתון

ColumnNumber - מספר סוג השדה לשליפה, אם לא סופק ערך – ברירת המחדל

היא 1.

SequenceNumber - מספר המופע לשליפה, אם לא סופק ערך – ברירת המחדל

היא 1 או הערך העדכני לתאריך שסופק, לפי סוג הנתון שהתבקש.

מספר סוג השדה:

ColumnNumber	שם שדה	סוג שדה
1	VALNUM	DECIMAL(9,4)
2	VALNUM2	DECIMAL(9,4)

**דוגמא:** שליפת אחוז המשרה הנוכחי - EmpVecDec(2,Date())  
שליפת מכסת ב"מ בשעות לקוד ההיעדרות הראשון - EmpVecDec(6,Date(),2,1)

**EMPVECINT(VectorType, Date [,ColumnNumber][,SequenceNumber])**

**שימוש:** שליפת נתוני **ערכים שלמים** מטבלת TKSQ – EMPVEC בלבד !!!  
הפונקציה סורקת את כל הווקטורים ומחזירה את הערך שהתבקש.

**תחביר:** VectorType - סוג הווקטור

Date - תאריך הנתון

ColumnNumber - מספר סוג השדה לשליפה, אם לא סופק ערך – ברירת המחדל היא 1.

SequenceNumber - מספר המופע לשליפה, אם לא סופק ערך – ברירת המחדל היא 1 או הערך העדכני לתאריך שסופק, לפי סוג הנתון שהתבקש.

מספר סוג השדה:

ColumnNumber	שם שדה	סוג שדה
0	VALTYPE	TINYINT
1	VALINT	INT

**דוגמא:** שליפת מספר טבלת GZ נוכחית - EmpVecInt(3,Date())  
שליפת שם קוד ההיעדרות הנוכחי - ABS\_NANE(EmpVecInt(1,Date()))

**EMPVECSTR(VectorType, Date [,ColumnNumber][,SequenceNumber])**

**שימוש:** שליפת נתוני **מחרוזות טקסט** מטבלת TKSQ – EMPVEC בלבד !!!  
הפונקציה סורקת את כל הווקטורים ומחזירה את הערך שהתבקש.

**תחביר:** VectorType - סוג הווקטור

Date - תאריך הנתון

ColumnNumber - מספר סוג השדה לשליפה, אם לא סופק ערך – ברירת המחדל היא 1.

SequenceNumber - מספר המופע לשליפה, אם לא סופק ערך – ברירת המחדל היא 1 או הערך העדכני לתאריך שסופק, לפי סוג הנתון שהתבקש.

מספר סוג השדה:

ColumnNumber	שם שדה	סוג שדה
1	VALCHAR	CHAR(30)
2	VALCHAR2	CHAR(30)
3	VALCHAR3	CHAR(30)

**דוגמא:** שליפת שם הילד התשיעי - EmpVecStr(5,Date(),1,9)

## **נתונים מטבלת DEPVAL**



### DEPVAL(DEPARTMENT, FILENUM, FIELDNUM, DATE)

**שימוש:** קבלת ערך השפעה מחלקתית רצויה מטבלת DEPVAL. פותח במקור עבור איכילוב, שם נתוני המכסות נאגרים ברמה מחלקתית.

**תחביר:** **DEPARTMENT** – מספר המחלקה ממנה יש לקבל את ערך ההשפעה המחלקתית.

**FILENUM** – מספר טבלת עובדים

**FIELDNUM** – מספר שדה בטבלה

**DATE** – תאריך מבוקש

**דוגמא:** קבלת הערך של שדה מספר 99 מטבלת עובדים (מס' טבלה 1) עבור מחלקה 1, מעודכן לסוף תקופה. `DEPVAL(1,1,99,ENDPP())`

פונקציה זו קיימת בגרסת SQL בלבד.

## סידור עבודה

### Getsched(Employee, Date, Pair Number/Duration, In/Out)

**שימוש:** קבלת שעת כניסה \ יציאה מסידור העבודה מהזוג הראשון \ השני. ניתן גם להציג משך בין הכניסות והיציאות המתוכננות בסידור.

**תחביר:** **Employee** – מספר עובד

**Date** – תאריך מבוקש

**Pair Number/Duration** – הגדרת סוג השדה הרצוי:

זוג ראשון - 1

זוג שני - 2

משך בין כניסות ויציאות מתוכננות - 3.

**In/Out** – כניסה מתוכננת - 1 יציאה מתוכננת - 2, אם מוגדר משך (3), בשדה זה

יש להגדיר 0.

**דוגמא:** קבלת שעת היציאה המתוכננת בסידור מהזוג הראשון.

`getsched(DAILY->Employee, DAILY->Date, 1, 2)`

פונקציה זו קיימת בגרסת SQL בלבד.

### Getsched1(Employee, Date, Pair Number/Duration, In/Out)

**שימוש:** שלפת שעת כניסה/יציאה (מהזוג הראשון/השני) או משך בין הכניסות והיציאות המתוכננות בסידור העבודה.

**Employee** – מספר עובד

**Date** – תאריך מבוקש

**Pair Number/Duration** – הגדרת טווח השעות הרצוי:

זוג ראשון - 1

זוג שני - 2

משך בין כניסות ויציאות מתוכננות - 3.

**In/Out** – הגדרת סוג השדה הרצוי:

כניסה מתוכננת - 1,

יציאה מתוכננת - 2,

אם הוגדר "משך" (3) בטווח השעות הרצוי, יש להגדיר 0.

**דוגמא:** קבלת שעת היציאה המתוכננת בסידור העבודה מהזוג הראשון.

`Getsched1(DAILY->Employee, DAILY->Date, 1, 2)`

**הערה:** כאשר עובדים עם מודול התראות יש להשתמש בפונקציה זו במקום GETSCHED, כיון שפונקציה זו מקבלת רשומת סידור עבודה וירטואלית אם לא קיימת רשומה אמיתית בטבלה CRNSCHED.

## אינדקס

.	6
.AND.....	6
.F.....	6
.NOT.....	6
.OR.....	6
.T.....	6
<b>A</b>	
ABS_DAYS(Absence_Code).....	11
ABS_HOUR(Absence_Code).....	11
ABS_NAME(ABS_CODE).....	11
ABS_SHRT(ABS_CODE).....	11
ABSTRX_D(Employee#,Per_Key,Abs_Code).....	13
ABSTRX_H(Employee#,Per_Key,Abs_Code).....	13
ABSUSE_D(Employee#,Year,Abs_Code).....	13
ABSUSE_H(Employee#,Year,Abs_Code).....	13
ATTACHEXIST(Employee#,Date,From_Abs_Code,Till_Abs_Code).....	16
<b>B</b>	
BALANC_D(Employee#,Year,Abs_Code).....	12
BALANC_H(Employee#,Year,Abs_Code).....	12
BATRATE(N,EndPP()).....	27
<b>C</b>	
CalcDu.....	9
CAT_NAME().....	17
CAT_NAME(Category Number).....	16
CHILD_COUNT(ITEMTYPE,EMPLOYEE,AGE,DATE).....	31
CTOD(Character_Value).....	8
<b>D</b>	
DADCATF(Cat,Factor,.....).....	18
DAPCATF(Cat,Factor,.....).....	18
DATE().....	8
DATE2STR(DATE(),"MM/DD/CCYY").....	8
DAY(Date_Value).....	8
DAY_ABS(Abs_Code,Value_Code).....	23
DAY_VEC(Vec_No,Field_No).....	23
DAYCATDC().....	17
DAYCATHH().....	17
DAYNAMEL(DAILY->DAY_TYPE).....	23
DEC2TIME(Decimal_Expression).....	10
DEL().....	6
DELETED().....	6
DEPVAL(DEPARTMENT,FILENUM,FIELDNUM,DATE).....	38
DIV_LONG(Div <sub>1</sub> ,Div <sub>2</sub> ).....	5
DND_EMPL(Meal-n,Meal-n1,...,Meal-n20).....	32

DND_SITE(Meal-n,Meal-n1,...,Meal-n20) .....	32
DND_UNIT(Meal-n,Meal-n1,...,Meal-n20) .....	32
DNENDPP() .....	33
DNP_EMPL(Meal-n,Meal-n1,...Meal-n20).....	32
DNP_UNIT(Meal-n,Meal-n1,...,Meal-n20).....	32
DPTRATE(N,EndPP()) .....	28
DSEARCH(Field_no_in_AbsVec,Value_to_be_Found) .....	23
DTOC(Date_Value) .....	8
DTOS(Date_Value) .....	8

## E

EMP_ABS(DATE()).....	30
EMP_ABS(ENDPP()).....	30
EMPFLEX .....	31
EMPLGCOD(EMPLOYEE,GROUPCODE,PERIOD).....	36
EMPLGDEC(EMPLOYEE,GROUPCODE,PERIOD) .....	35
EMPLGNUM(EMPLOYEE,GROUPCODE,PERIOD).....	35
EMPVECDATE(ItemType, Date [,ColumnNumber][,SequenceNumber]) .....	36
EMPVECDEC(ItemType, Date [,ColumnNumber][,SequenceNumber]) .....	37
EMPVECINT(VectorType, Date [,ColumnNumber][,SequenceNumber]).....	37
EMPVECSTR(VectorType, Date [,ColumnNumber][,SequenceNumber]) .....	37
ENDPP() .....	9
ERRDESCR(Error#) .....	19

## G

GET_NOTE(EMPLOYEE,DATE,Maximum_Length) .....	23
GET_NOTE_TC(EMPLOYEE ,DATE ,SEQ_NO ,Maximum_Length) .....	34
GET_RATE(Code,Date).....	27
GETABSFL(EMPLOYEE,"ABS_CODE,FIELD_NO," ,YEAR) .....	12
GETABSPR(EMPLOYEE,"ABS_CODE,FIELD_NO," ,DATE) .....	12
GETANUAL(Subject_No,Field_No) .....	25
GETCONTR(DATE()) .....	28
GETCONTR(ENDPP()) .....	28
GETEVENT("Event_Code",Value_Code) .....	21
GETEXEPT("Event_Code",Value_Code).....	21
GETFILED("FILE_NAME,KEY_FIELD,RETURN_FIELD," ,VALUE).....	28
GETSALD(Code) .....	27
Getsched(Employee,Date,Pair Number/Duration, In/Out) .....	38
Getschedl(Employee,Date,Pair Number/Duration, In/Out) .....	39
GETTRANSMIT(EMPLOYEE ,DATE).....	24

## I

IIF(Logical_Value,Result if True,Result if False) .....	6
INTRANGE(VAL(CHAR_VALUE),VAL-1,VAL-2,...,VAL-20) .....	7

## J

JULDATE(Date <sub>1</sub> )-JULDATE(Date <sub>2</sub> ) .....	8
---------------------------------------------------------------	---

## L

LEFT(Character_Value, Number_of_Characters).....	4
--------------------------------------------------	---

LROUND(Time_Field,Rounding_Interval,Grace).....	10
LTRIM(Character_Value) .....	4
<b>M</b>	
MEAL_FOUND(Att_Pair_Strat, Att_Pair_End, Meal_Site, Diner_type, Meal_Number_N...1-25).....	32
MIN2TIME(Time_Expression) .....	10
MOD(Div <sub>1</sub> ,Div <sub>2</sub> ) .....	5
MONTH(Date_Value) .....	9
<b>O</b>	
ORDERATE(N,EndPP()) .....	27
OTHERPERVAL (PeriodAddition, FieldNumber) .....	24
<b>P</b>	
PAGENO().....	7
PARTRATE(N,EndPP()).....	27
PERCATDC() .....	17
PERCATHH() .....	17
PERFRDNVAL(FieldNumber) .....	33
PROCRAATE(N,EndPP()).....	27
PRT_TIME( DATE()) .....	30
PRT_TIME( ENDPP()) .....	30
PUTEXEPT("Event_Code",Value_Code,Value) .....	21
<b>S</b>	
S_ABS_DC(Abs <sub>0</sub> ,Abs <sub>1</sub> ,...,Abs <sub>40</sub> ).....	11
S_ABS_DD(Abs <sub>0</sub> ,Abs <sub>1</sub> ,...,Abs <sub>40</sub> ) .....	11
S_ABS_HH(Abs <sub>0</sub> ,Abs <sub>1</sub> ,...,Abs <sub>40</sub> ) .....	11
STOD(Character_Value) .....	8
STR(Number,Length,Decimals) .....	4
Strev() .....	4
SUBSTR(Character_Value,Start_Position,Number_of_Characters) .....	4
<b>T</b>	
TAB_NAME("Table_Name",Table_Number .....	28
TCDAYORDER(In ,Out) .....	34
TCDCATF(Cat,Factor,.....).....	17
TCDEF_Level_No(Employee,DTOS( DATE()),Row_Number .....	34
TCENDPP() .....	35
TCPCATF(Cat,Factor,.....).....	17
TIME() .....	9
TIME2DEC(Time_Expression).....	10
TIME2MIN(Time_Expression) .....	10
TKDCATF(Cat,Factor,.....).....	18
TKPCATF(Cat,Factor,.....) .....	18
TRIM(Character_Value).....	4
TRIM(GET_NOTE(DAILY<-EMPLOYEE,DAILY->DATE,80)).....	23
TRX_EX_ D(Employee#,Per_Key,Abs_Code,Transaction Type, Mode, Justification Code,...).....	15

TRX_EX_H(Employee#,Per_Key,Abs_Code,Transaction Type, Mode, Justification Code,...)	15
TRX_R_D(Employee#,Per_Key,"Abs_Code, Mode, Justification Code,...")	14
TRX_R_H(Employee#,Per_Key,"Abs_Code, Mode, Justification Code,...")	13
TRX_REASON_D(Employee#,Per_Key,Abs_Code, Mode, Justification Code,...)	15
TRX_REASON_H(Employee#,Per_Key,Abs_Code, Mode, Justification Code,...)	14, 15, 24, 34
<b>U</b>	
UPPER(Character_Value)	4
<b>V</b>	
VAL(Character_Value)	5
<b>Y</b>	
YEAR(Date_Value)	9
YEAR4ABS (Employee Number, Date, Absence Code)	11