

סיכום שיעור

המשך נורמליזציה, בניית בסיס הנתונים שאיתו נעבוד, מבוא ל - Operator overloading ב- C#

- השלמנו את הנושא של 3rd, 2nd normal form. על ידי דוגמאות שבאחת ראינו שדות שתלויים במפתח חלקי (מתוך מפתח שמשתרע על יותר משדה אחד) וזה יוצר כפילות נתונים רבה - תיקון על ידי פיצול הטבלה ל-3 טבלאות, 1 עבור כל ישות שלה מפתח ראשי מתאים (אחד משני השדות שהרכיבו את ה- primary key של הטבלה המקורית) + טבלה שמתארת את היחס בין שתי הישויות בטבלאות הניפרדות (2nd normal form)

בדוגמה האחרונה היה דפקט קצת קשה יותר לאבחנה, אחד השדות היה תלוי בשדה אחר שאינו מפתח ראשי (כלומר תלות עקיפה במפתח הראשי) - פיצלנו את הטבלה לשתיים בצורה שאחת תכיל את המפתח הראשי (שם עובד) ואת אחד השדות (העיר שבה הוא גר) וטבלה שניה שמייחסת את העיר למדינה.

הרעיון של הנירמול זה להפחית כפילות נתונים גם מבחינת שימורם וגם מבחינת עידכונם ופישוט מקסימלי למבנה הטבלאות, אך ורק לנתונים ההכרחיים בכל טבלה ע"מ לשלוף כל סוג של נתון מהטבלאות. כל זה שייך לתחום של תכנון בסיסי נתונים שבחלקו הינו מדע ובחלקו גם אמנות. יש סיכוי שנעבוד בחברות Startup ולכן כדאי לדעת מעט על נושא זה, גם אם זה לא העיקר של הלימוד.

- תארנו את בסיס הנתונים של הסטודנטים עם הספרים והספריות בדף:
http://www.lamed-oti.com/school/tmy/yud_alef/t-sql_syntax.pdf

בדף גם מצויינים הפרטים על סוגי השדות והחוקים ליצירת הטבלאות (primary ו- foreign keys).

- בנושא של העמקת הידע ב- C#, למען ביסוס מעמדנו כמהנדסי תוכנה לעתיד, הזכרנו קצת רקע בנושא של העמסת אופרטורים (Operator Overloading) שהוא למעשה מה שניקרא בז'רגון המהנדסים: Syntactic sugar (סוכר תחבירי) כיוון שהוא דרך אלגנטית (קריא: פשוטה) לבצע פעולות על עצמים מטיפוסים שהגדרנו באופן דומה לביצוע הפעולות על טיפוסים קיימים בשפה (כמו int או char). למשל פעולות חיבור או השוואה.

- לשם כך נבנה מחלקה (class) פשוטה של תאריכים (כפי שמתואר בשיעורי הבית), ובהמשך ניראה כיצד לבנות את הסוכר התחבירי עבורה.

- שיעורי בית

1. לבנות מחלקה בשם date עם משתנה פרטי בשם day שיכיל את מספר הימים שחלפו מאז תחילת הספירה ועד לתאריך שלפיו בונים את העצם. אנו מניחים לשם הפשטות שאין שנים מעוברות ובכל שנה היו 365 יום על פי החודשים כך שבינואר 31, בפברואר 28 וכך הלאה עד דצמבר 31.

2. יהיו לנו 2 Constructors. אחד מקבל כפרמטר מספר שלם חיובי, והשני מקבל 3 מספרים בפורמט d,m,y (d זה יום בחודש, m זהו חודש ו-y זו שנה). הבונה מבצע בדיקות תקינות וזורק אותנו החוצה במידה והקלט שגוי (בעזרת הפקודה: System.Environment.Exit(0)). אם הקלט הוא מספר יחיד שלם חיובי, אז המשתנה הפרטי day יקבל ערך זה, אחרת (3 שלמים), נודא ש-d הוא בין 1 ל-31, ש-m הוא בין 1 ל-12 וכן ש-d הוא מספר שמתאים ל-m - למשל: 29,2,2014 אינו חוקי, כי בפברואר אמרנו שתמיד יהיו עד 28 יום (אין שנים מעוברות). הבונה יחשב את מספר הימים שעברו מהיום 1/1/1 ועד היום המבוקש (בהינתן קלט של (d, m,y) בנוסף ניצור גם פעולה (method) עבור הדפסה פשוטה של התאריך כמספר שלם, כך שאם בתוכנית יכתב:

```
date d1= new date(1,2,1);
d1.print();
```

יודפס המספר: 31

3. לבנות את הטבלאות המתוארות בדף הקרוי:

http://www.lamed-oti.com/school/tmy/yud_alef/t-sql_syntax.pdf

4. תאור במילים על איך למצוא אם לתלמיד בעל מספר זיהוי (ID) x, קיים לפחות ספר אחד בעל זיהוי (ISBN) של y - בספריה שנמצאת בעיר מגוריו של הסטודנט.