טעינת נתונים קשורים לאובייקטים

Eager Loading

Eager – "תאב", מבקש, מציינים במפורש שרוצים גם את ה navigation property

צריך לדעת שלפעמים גם אם לא מציינים ייטענו נתונים – במידה והם נטענו במסגרת שאילתא אחרת. זה חלק מהיכולות של EF.

* Include
* ThenInclude

using (var context = new BloggingContext())

{

var blogs = context.Blogs

.Include(blog => blog.Posts)

.ThenInclude(post => post.Author)

.ThenInclude(author => author.Photo)

.Include(blog => blog.Owner)

.ThenInclude(owner => owner.Photo)

.ToList();

}

אם רוצים שני תת-נתונים של אותו מאפיין, צריך פעמיים לעשות include ובכל פעם להגיע לתת מאפיין שאותו רוצים לטעון. בדוגמא להלן, טוענים את פוסט->כותב הפוסט, ופוסט->תגיות של הפוסט

using (var context = new BloggingContext())

{

var blogs = context.Blogs

.Include(blog => blog.Posts)

.ThenInclude(post => post.Author)

.Include(blog => blog.Posts)

.ThenInclude(post => post.Tags)

.ToList();

}

Explicit Loading

מבקשים במפורש עבור entity מסוים שרוצים לטעון את הנתונים הקשורים אליו.

using (var context = new BloggingContext())

{

var blog = context.Blogs

.Single(b => b.BlogId == 1);

context.Entry(blog)

.Collection(b => b.Posts)//רשימה – לבלוג יש הרבה פוסטים

.Load();

context.Entry(blog)

.Reference(b => b.Owner)//אובייקט יחיד – לבלוג יש אוביקט אחד של בעל הבלוג

.Load();

}

על Collection אפשר לבצע סינון – לבקש רק חלק המידע.

var goodPosts = context.Entry(blog).Collection(b => b.Posts).Query() .Where(p => p.Rating > 3).ToList();

Lazy Loading

Lazy – עצלן. רק כשמבקשים את המידע, הוא נטען לזכרון

יש דרכים מסוימות להגדיר lazy loading, אבל צריך ללמוד כמה דבירם נוספים קודם. כרגע מספיק לדעת שיש אפשרות כזו.

כלל חשוב:

אם רוצים להוסיף לטבלה שדה שלא מאפשר null, אם עושים את זה אחרי שכבר יש נתונים בטבלה, חייבים קודם להוסיף את השדה כאחד שכן מאפשר null, לעדכן לכל הרושמות הקיימות ערך מסוים ברירת מחדל, ואז להפוך את השדה לכזה שאינו מאפשר null

מחיקה של אובייקטים

ספר –

שדה סופר שהוא מפתח זר לטבלת סופרים, נגיד שאין ספר שלא מגודר לו סופר ולכן השדה לא מסומן כמאפשר null

שדה קטגוריה יתכן שלא מלא עבור כל ספר, ורוצים להשאיר את זה עם אפשרות שהוא יהי ריק

סופר -

קטגוריה -

אפשרויות של הגדרה מה יקרה כאשר מוחקים ערך ששדות מטבלאות אחרות מקושרים אליו:

ON DELETE NO ACTION – נקבל שגיאה כאשר ננסה למחוק ערך מטבלה שמקושרים אליה שדות מטבלה אחרת – זה ברירת מחדל עבור כל מפתח זר

ON DELETE CASCADE – למחוק את הערכים המקושרים, לדוגמא: אם מחוקים קטגוריה, מוחקים את כל הספרים שלה יוכל להיות הגיוני לדוגמא פציינט ותורות במרפאה, רוצים למחוק את כל התורות אם הפציינט כבר לא חבר במרפאה

ON DELETE SET NULL – מעדכנים את הערך של השדה המקשר ב null במקום להתייחס לערך בטבלה השנייה, שאותו כרגע מוחקים. זה מחייב שהשדה יאפשר null. לדוגמא: כאשר מוחקים קטגוריה, מעדכנים בשדה CategoryId של כל הספרים שבקטגוריה הזו, את הערך null במקום ה id של הקטגוריה הזו.

ON DELETE SET DEFAULT – צריך הגדרה של default על בעמודה

The DELETE statement conflicted with the REFERENCE constraint "FK\_\_Book\_\_CategoryId\_\_398D8EEE". The conflict occurred in database "C:\USERS\USER\SOURCE\REPOS\DBACCESS\DBACCESS\LIBRARY.MDF", table "dbo.Book", column 'CategoryId'.

The statement has been terminated

קורה כאשר מנסים למחוק רשומה שיש התיחסות אליה מטבלה אחרת, ומוגדר ב מפתח הזר ערך ברירת מחדל של No ACTION

הפתרון: או לשנות את הגדרת המפתח לפי מה שמעונינים שיקרה

או ללכת למחוק קודם את ההתיחסות לנתון שאותו מעונינים למחוק

ואז למחוק את הקטגוריה

Value cannot be null. ' – רצינו למחוק איזשהי רושמה, והיא לא נמצאה ב DB, ואז כששלחנו אותה ל remove בגלל שהיא Null קבלנו את השגיאה הזו.

הפתרון: לבדוק לפני שמבצעים remove שהערך הוא לא Null, ולבדוק למה חזר null אם חשבנו שיש לנו מה למחוק