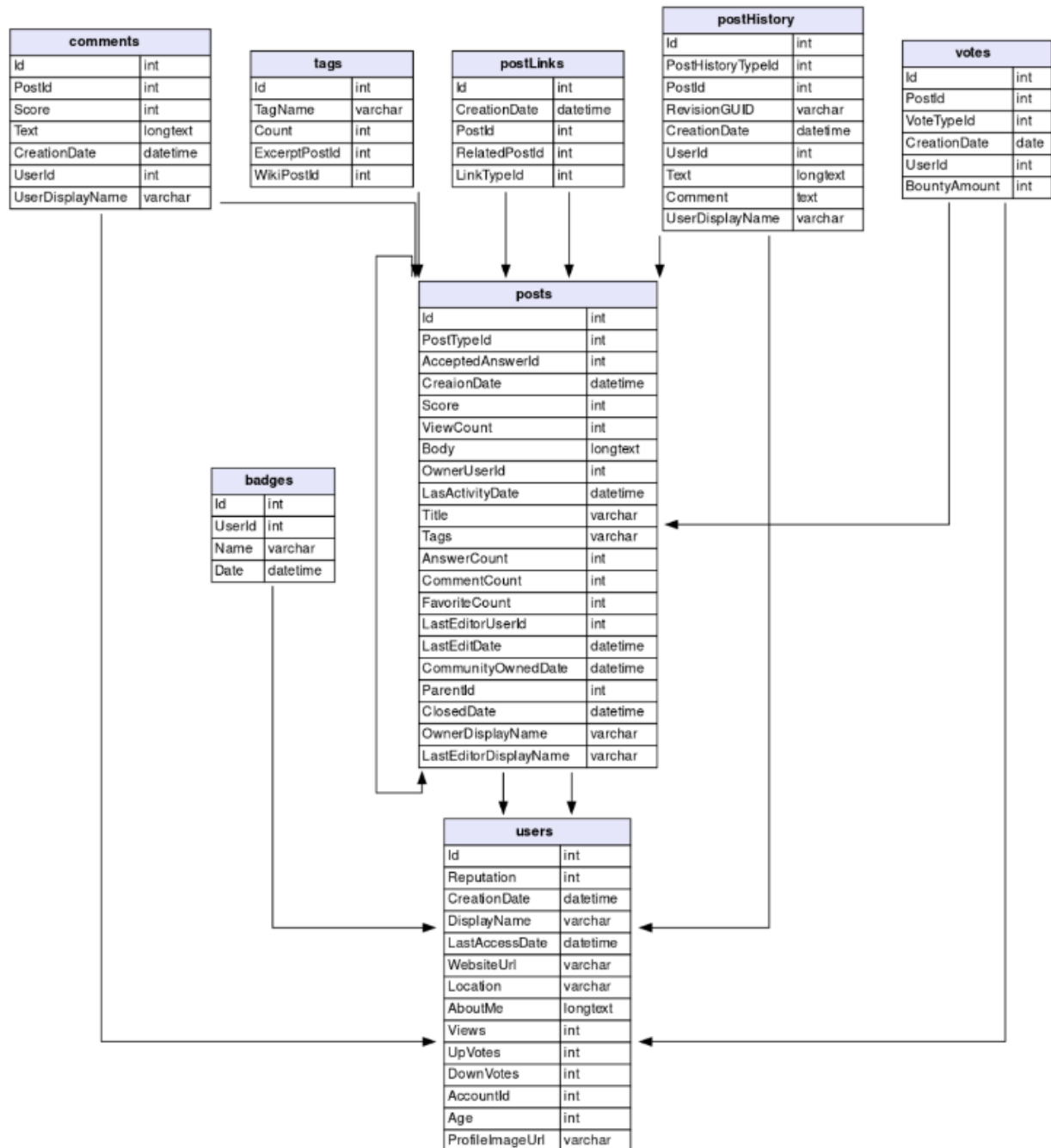


שאלת הגנת סכמה - עומרי נפתלי - סכמת מועד 2025 א'



טבלת comments :

- Id
- איום: כפילויות או NULL.
- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT.

מכיוון שלשדה Id אין קיום מחוץ לאתר או מפתח טבעי, הכי פשוט לתת בעצמנו ערכים רציפים.

· **PostId**

- איום: תגובה לפוסט שלא קיים (orphan).
- פתרון: FOREIGN KEY ל – posts(Id) עם ON DELETE CASCADE + אינדקס.

· **Score**

- איום: ערכים שליליים/לא סבירים.
- פתרון: CHECK (score >= 0) או טווח מוסכם.

· **Text**

- איום: טקסט ריק/NULL.
- פתרון: NOT NULL + CHECK (LENGTH(Text) > 0)

· **CreationDate**

- איום: תאריך עתידי
- פתרון: CHECK (CreationDate <= CURRENT_TIMESTAMP)

· **UserId**

- איום: תגובה למשתמש שלא קיים או NULL לא מוצדק.
- פתרון: FK ל – users(Id) ו NOT NULL.

· **UserDisplayName**

- איום: הערך כאן הוא עותק של שם המשתמש מטבלת users, ולכן עלול להיות שונה מהערך המקורי (בגלל עדכון או באג).
- פתרון: ליצור טריגר שמעתיק את השם מטבלת users בעת יצירה/עדכון של הרשומה וליצור מנגנון ניטור שמזהה פערים בין הערכים.

טבלת tags :

Id .

- איום: כפילויות או NULL.

- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT

TagName .

- איום: כפילויות ברישיות שונות/רווחים.

- פתרון: UNIQUE על LOWER(TagName), נירמול (TRIM), בדיקת תווים.

Count .

- איום: עלול להיות שונה ממספר ההופעות האמיתי של התגית בפוסטים, משום שהוא כפילות של מידע שצריך עדכון בכל שינוי. זה יוביל לאי־התאמות ולסטטיסטיקות שגויות.

- פתרון: יש שתי דרכים להתמודד –

1. לחשב בכל פעם את מספר הפוסטים שמכילים את התגית ולא לשמור את Count כלל. זה מונע כפילות אך מוותר על היתרון של חישוב מוקדם.

2. להשאיר את Count כעמודה ייעודית, ולעדכן אותה בכל שינוי רלוונטי (הוספה או הסרה של תגית), למשל באמצעות טריגרים או לוגיקה באפליקציה. בנוסף, להפעיל ניטור תקופתי שמזהה פערים בין הערך המחושב לערך השמור.

ExcerptPostId / WikiPostId .

- איום: מצביע לפוסט לא מתאים/לא קיים.

- פתרון: FOREIGN KEY ל – posts(Id) עם ON DELETE SET NULL; טריגר שמוודא סוג/מאפיין מתאים בפוסט.

טבלת PostLinks

Id .

- איום: כפילויות או NULL.

- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT

- **CreationDate**

- איום: תאריך עתידי

- פתרון: CHECK (CreationDate <= CURRENT_TIMESTAMP)

- **PostId / RelatedPostId**

- איום: orphan או self-link.

- פתרון: FK ל – posts(Id) + CHECK(PostId <> RelatedPostId)

- **LinkTypeId**

- איום: ערך לא מוכר/טעות הקלדה.

- פתרון: FK ל – linkTypes(LinkTypeId); ייחוד עסקי לפי צורך (למשל, PostId, UNIQUE((RelatedPostId, LinkTypeId)

טבלת postHistory

- **Id**

- איום: כפילויות או NULL.

- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT

- **PostHistoryTypeId**

- איום: ערך לא חוקי/לא מתועד

- פתרון: FK ל – postHistoryTypes(PostHistoryTypeId)

- **PostId**

- איום: orphan.

- פתרון: FK ל- posts(Id) + אינדקס.

- **RevisionGUID**

- איום: הערך יכול להיות לא ייחודי או NULL, מה שיגרום לכך ששתי גרסאות שונות יקבלו מזהה זהה או שלא יקבלו מזהה כלל.
- פתרון: להגדיר את השדה כ- NOT NULL ו- UNIQUE, כדי להבטיח שכל גרסה מקבלת מזהה ייחודי.
- איום נוסף: הערך עשוי להיות מחרוזת כלשהי שאינה בפורמט של GUID לדוגמה טקסט חופשי שהוזן בטעות.
- פתרון: להוסיף בדיקה של מבנה הערך (בצד בסיס הנתונים אם יש תמיכה בביטוי רגולרי, או בצד האפליקציה) כדי לאכוף שמדובר במחרוזת בפורמט תקני של GUID.
- פתרון: UNIQUE + NOT NULL.

CreationDate

- איום: תאריך עתידי
- פתרון: CHECK (CreationDate <= CURRENT_TIMESTAMP)

UserId / UserDisplayName

- ראו טבלת comments.UserId / UserDisplayName.

Text / Comment

- איום: טקסט ריק כאשר נדרש תוכן.
- פתרון: CHECK (Text/Comment IS NULL OR LENGTH(Text/Comment) > 0).

טבלת votes :

Id

- איום: כפילויות או NULL.
- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT

PostId

- איום: orphan.
- פתרון: FK ל- posts(Id) + אינדקס.

· **VoteTypeId**

- איום: ערך לא מוכר/טעות הקלדה.
- פתרון: (FK ל- voteTypes(TypeId); אינדקס.

· **UserId**

- איום: NULL לא מוצדק/יתומים.
- פתרון: (FK + NOT NULL ל- Id) users.

· **BountyAmount**

- איום: שלילי או לא עקבי עם סוג ההצבעה.
- פתרון: (CHECK (BountyAmount >= 0); טריגר שמחייב $BountyAmount > 0$ כאשר VoteTypeId הוא Bounty, ואוסר ב- VoteTypeId אחרים.

טבלת posts :

· **Id**

- איום: כפילויות או NULL.
- פתרון: PRIMARY KEY NOT NULL AUTO_INCREMENT

· **PostTypeId**

- איום: ערך לא קיים/לא מתועד.
- פתרון: (FK ל- postTypes(PostTypeId) + אינדקס.

· **AcceptedAnswerId**

- מפנה לפוסט שאינו תשובה/לא שייך לשאלה.
- פתרון: טריגר שמאמת שהפוסט המופנה הוא Answer ל- ParentId של השאלה.

· **CreationDate / LastActivityDate / LastEditDate / ClosedDate / CommunityOwnedDate**

- איום: אי סדר כרונולוגי.
- פתרון: CHECK-ים כרונולוגיים (למשל $LastActivityDate \geq CreationDate$).

· **Score**

- איום: אי-עקביות מול votes.
- פתרון: טריגרים לעדכון/חישוב ב – VIEW.

· **ViewCount / AnswerCount / CommentCount / FavoriteCount**

- ראו comments.Count ובנוסף ראו תשובה לשאלה 2 סעיף c במבחן.

· **Body / Title**

- איום: ריק/קצר מדי/ארוך מדי
- פתרון: NOT NULL + CHECK על אורך.
- ראו תשובה לשאלה 2 סעיף a במבחן.

· **Tags**

- איום: חוסר נורמליזציה וקושי אכיפה.
- פתרון: להחליף בטבלת קשר postTags(PostId, TagId) עם UNIQUE(PostId, TagId).
- ראו תשובה לשאלה 2 סעיף b.

· **OwnerUserId / OwnerDisplayName / LastEditorUserId / LastEditorDisplayName**

- איום: NULL כפול/חוסר עקביות.
- פתרון: FK של המזהה ל – users(Id).

· **ParentId**

- איום: תשובה ללא הורה/שאלה עם ParentId.
- פתרון: FK ל- posts(Id).

טבלת badges :

· **Id**

- איום: כפילויות.
- פתרון: PRIMARY KEY.

- **UserId**
- איום: orphan.
- פתרון: FK ל- users(Id).

- **Name**
- איום: וריאציות כתיב/רישיות.
- פתרון: מילון/נרמול; אם המדיניות היא "פעם אחת" אז UNIQUE.

- **Date**
- איום: תאריך עתידי.
- פתרון: CHECK (Date <= CURRENT_TIMESTAMP).

טבלת users :

- **Id**
- איום: כפילויות.
- פתרון: PRIMARY KEY.

- **Reputation**
- איום: שלילי/לא עקבי עם אירועי מערכת
- פתרון: CHECK (Reputation >= 0); עדכון מבוקר בפרוצדורות.

- **CreationDate / LastAccessDate**
- איום: אי-סדר כרונולוגי
- פתרון: CHECK (LastAccessDate >= CreationDate AND LastAccessDate <= CURRENT_TIMESTAMP).

· **DisplayName**

- איום: ריק/ארוך מאוד/תווים בעייתיים.
- פתרון: CHECK על אורך/תווים, נירמול.

· **WebsiteUrl / ProfileImageUrl**

- איום: URL לא תקין
- פתרון: וידוא פורמט בעזרת REGEXP.
- איום נוסף: גם אם ה-URL בפורמט נכון, ייתכן שהעמוד שהוא מצביע אליו לא קיים בפועל (404) או שאינו נגיש (למשל הדומיין לא רשום).
- פתרון: ניתן לבצע בדיקה יזומה (lookup) בזמן ההזנה כדי לוודא שהעמוד זמין. עם זאת, חשוב להדגיש שבדיקה כזו היא חלקית בלבד – ייתכן שהעמוד יהיה קיים ברגע הבדיקה אך יוסר או ייפול מאוחר יותר. לכן גם אחרי בדיקה מוצלחת, עדיין עלולה להיווצר אי-התאמה בעתיד.
- בנוסף, קשה לוודא שזה אכן האתר של הגולש, מכיוון שכל אחד יכול לשים איזה כתובת שהוא רוצה.

· **Views / UpVotes / DownVotes**

- איום: שליליים.
- פתרון: CHECK (≥ 0).

· **AccountId**

- איום: מצביע ליישות לא קיימת (אין accounts בסכמה).
- פתרון: להוסיף טבלת FK + accounts.

· **Age**

- איום: הערך עשוי להיות לא סביר (למשל גיל שלילי או 200), וגם עצם קיום השדה בעייתי – גיל משתנה כל הזמן, ולכן שמירתו כעמודה סטטית יוצרת סיכון לאי-התאמה בין הנתון לבין הגיל האמיתי של המשתמש. בנוסף, שם השדה מטעה: המשתמש עלול לחשוב שמדובר בגיל עדכני בעוד שמדובר בגיל בזמן יצירת החשבון או בעת ההזנה.
- פתרון: עדיף לשמור **תאריך לידה** (BirthDate) ולא גיל. מתאריך הלידה ניתן לחשב את הגיל העדכני בכל שליפה, וכך נמנע מהצורך לעדכן את השדה מדי שנה. ייתכן שבמערכת כמו Stack Overflow נמנעו מלבקש תאריך לידה מסיבות של פרטיות או צמצום חשיפת נתונים אישיים, ולכן נשמר רק גיל חד-פעמי בעת ההרשמה. אם בוחרים להשאיר את Age כפי שהוא – רצוי לכל הפחות להוסיף בדיקות סבירות להבהיר שהגיל מייצג את הערך בזמן ההזנה.

