

מפתחות טבעיים לעומת מפתחות מלאכותיים

דיון נוקב ומעמיק מתרחש (כמעט) תמיד בחדרי הניתוח של צוותי הפרויקטים הנבנים בסביבה הטבלאית: האם להשתמש במפתחות "אמיתיים" לטבלה או להחליפם באמצעים מלאכותיים (SURROGATE KEYS).

ישנן דעות בעד ונגד לכאן ולכאן. במאמר זה נדבר על שתי הגישות, נדגיש את היתרונות והחסרונות בכל אחת מהן וננסה לעזור מניסיוננו על ידי הצגת הקווים המנחים אותנו בבחירת השיטה.

רקע

כאשר מתכננים אפליקציה, קובעים את הישויות שלה ואת הקשרים ביניהן. (וכמובן, עוד אי-אלו פרטים...). חלק מהגדרת הישות קובע את מאפייניה, כולל מזהיי הישות.

לישות, כידוע, יכולים להיות מסי' מזהים, אולם תמיד חייב להיות מזהה חד-ערכי.



הערה: אם אי אפשר למצוא מזהה כזה, חייבת להידלק נורה אדומה כי כנראה שמהו "התפקשש". יש להחזיר את הישות לשולחן הניתוחים על מנת לגלות מה חסר או מיותר בישות זו.

המזהה החד-ערכי (להלן יקרא בשם המוכר UID) יכול להיות מורכב ממאפיין אחד (כגון, מספר תעודת זהות) או ממספר מאפיינים, אשר שילוב שלהם ייתן תוצאה חד-ערכית (כגון, מספר טלפון + שם משפחה + שם פרטי).

תהליך קביעת ה-UID הוא חלק מתהליך תכנון המודל הלוגי, חלק בלתי נפרד מהניתוח. עפ"י המתודולוגיה, בשלב זה, יש לבצע תכנון לוגי. לא כאן המקום להפעיל שיקולי ביצועים, הוצאה מנירמול (Denormalization) והחלטה על שימוש במפתחות מלאכותיים.

כידוע, שלב הניתוח מהלך, לא מעט, על החבל הדק בין תיאוריה למציאות. בשלב זה, ה"נייר" סופג הכל: אין בעיות אחסון, אין בעיות ביצועים, המשתמשים מבינים מה הם רוצים. כאמור, עולם שכולו טוב... בשלב העיצוב, נמצאים כבר עם שתי הרגליים קרוב יותר לקרקע המוצקת ומשנים, לעיתים, את ההגדרות התיאורטיות על מנת ליישם את המציאות (הכואבת?!). לפיכך, ההחלטה על בחירת מפתח מלאכותי לטבלה צריכה להתבצע, אם בכלל, בשלב עיצוב בסיס הנתונים.

יש משהו מאד נוח בהגדרת מפתח מלאכותי – גם אם קיים מפתח טבעי. אך צריך לזכור כי מפתח מלאכותי הוא פתרון יישומי ולא לוגי.

שלב המודל הפיסי

בכל מודל לוגי – אנחנו מתעקשים להשתמש במפתחות הטבעיים. אולם כאשר עוברים למודל הפיסי יש לבדוק ולשקול את השימוש במפתחות הטבעיים או במפתחות מלאכותיים PRIMARY KEYS-כ.

רצוי מאד שלא לערב את המשתמש בשימוש שנעשה במפתחות המלאכותיים במודל הפיסי. לא נציג לו אותם בדיאגרמות, דוחות, מסכים וכו'. הם זרים לו ואינם מהווים חלק מעולם המערכת שלו.

בעת קבלת ההחלטה, צריך לזכור כי מפתחות מלאכותיים הם

• לא אינטואיטיביים למשתמשים (לפחות בהתחלה)

• לא המפתחות הטבעיים (מה שמפריע לחלק מהמנתחים ומתכנני מודל הנתונים).

הדרך הטובה ביותר לביצוע שאילתות היא תמיד על סמך המפתח הטבעי. לכן זו צריכה להיות משימה הכרחית לזהות את המפתחות הטבעיים – לא רק על מנת לפעול נכון בשלב בניית המודל – אלא גם כיוון שהם מועמדים מצוינים לאינדקסים (ביצועים) אם נחליט ליישם מפתחות מלאכותיים.

מפתחות טבעיים קלים למשתמש בבואו לתחקר, לשאול, ליצור ולעבוד איתם במסכים. אין צורך ב-SEQUENCE-ים, קל לראות מה "אומרת" הרשומה בלי צורך להתחבר ל"אבא" וכו'.

מפתחות טבעיים לעומת מפתחות מלאכותיים

מצד שני, אי אפשר לעדכן מפתחות בטבלה זו אם ישנן רשומות בטבלאות אחרות שמצביעות אליה.

שיקול נוסף שיש לקחת בחשבון הוא השימוש במפתחות הזרים (FOREIGN KEYS). לעיתים עדיף להגדיר מפתח מלאכותי על מנת להמנע מריבוי השדות המשמשים בו.

שימוש במפתחות הטבעיים משאיר בפנינו 2 אפשרויות:

1. להגיד למשתמשים "סליחה, אבל במערכת החדשה, אסור לאנשים לשנות את שמם".
2. לאפשר שינוי באמצעות קוד תוכנה. כל שינוי מפתח – יחייב שינוי ברשומות כל ה"בנים" המצביעים אליו.

כמובן שאפשרויות אלו לא כל-כך מוצלחות... אם מפתח חיצוני יכול להשתנות בשלב כלשהו, זה מהווה מכשול במערכת אשר רצוי מלכתחילה לא ליפול עליו (גם אמא תמיד אמרה ש"לא מכניסים ראש בריא למיטה חולה") – כיוון שאחרת, מהר מאד נצטרך לספק את פתרון 2 המתואר לעיל – והרי כולנו כתבנו תוכנה – ואנו יודעים על ה-BUG-ים שלה... אולם, אם הערך לעולם, אבל, לעולם-לעולם, לא ישתנה, ניתן להשתמש בו בלי כל חשש כמפתח ראשי.

היתרון – עדכון המפתחות (הטבעיים), כמו גם קישור בין טבלאות (JOIN-ים) מאד פשוט. אם ה"פלסטיק" כל-כך טוב, ניתן, כמובן, להחליט על שימוש גורף במפתח מלאכותי. אך יש לזכור שלעיתים קרובות זה מעמיס, מיותר וכמובן יש מקומות שבהם נשלם גם ב"מקום" אחסון.

עליך לשאול את עצמך:

1. כמה JOIN-ים באמת אחסוך ע"י יצירת המפתח המלאכותי?
 2. באיזו תדירות ישתנה המפתח הטבעי?
 3. לאיזו צרה אכנס אם איישם את אחד משתי האפשרויות שתיארתי למעלה?
- ישנם מקומות בהם למפתח הטבעי אין את התכונות המצופות:

- לא ייחודי (2 אנשים עם אותו שם)
- שימוש חוזר במספרים "מתים" לאחר תקופה (מס' טלפון) באיזה שהוא שלב הוא יעודכן (החלפת מרכזת טלפון בבזק תגרום להוספת "6" למס' הטלפון לפני המס' הקיים) ואז מי שיצטרך לטפל בכל המקומות שזה משפיע, זה אנחנו!

גם אם נחליט על שימוש במפתח המלאכותי, רצוי להגדיר את המפתח הטבעי – ותמיד (כאמור) קיים מפתח כזה.

בכל מקרה שבו נחליט על שימוש במפתחות מלאכותיים, רצוי שנדאג ליצירת מפתחות ח"ע (UNIQUE INDEXES) לכל המפתחות הטבעיים. וזאת, כיוון שלמרות המפתחות המלאכותיים שיצרנו, שאילתות רבות מתבססות על השדות ששייכים למפתחות הטבעיים.

כמובן, רצוי להיזהר משימוש בלתי מבוקר במפתח מלאכותי, שאם לא כן, בקלות רבה נגיע לעיצוב מהיר של מודל רע.

התחשבות במתכנתים

האם שימוש במפתח מלאכותי מקל או מכביד על המתכנת? מצד אחד, במשפט ה-SELECT, יש צורך ב-JOIN מורכב הרבה יותר, כאשר המפתח הטבעי מתבסס על יותר משדה אחד, אולם, מצד שני, השימוש במפתח המלאכותי מביא לעיתים בעצמו ל-JOIN-ים "מיותרים" כיוון שה"ידע" לרשומה רלוונטית ניתן להסקה עפ"י מפתח טבעי הקיים כחלק מנתוני הטבלה הנוכחית.

מפתחות טבעיים לעומת מפתחות מלאכותיים

יש נטיה של אנשי התוכנה (מתכנתים, בעיקר) להעדיף פקודות JOIN פשוטות על סמך שדה אחד ולא עפ"י סט של תנאים (...דבר שבסיס נתונים טבלאי אמור להיות במיוחד טוב בו).
אספקט אחר, הינו מהות השדה. האם שימוש ב-ID ברור ונוח יותר למתכנת לעומת השימוש ב-BANK_CODE וב-BRANCH_CODE ?

המראה הטבעי או המראה המלאכותי :

אז מה עדיף? טבעי או מלאכותי? כמובן, שבכל אפליקציה יש להפעיל את מערכת השיקולים הרלוונטית עבורה, שכן המערכות שונות האחת מהשניה ולכן קיימות מערכות בהן האסטרטגיה הטובה ביותר היא שימוש במפתחות מלאכותיים וישנם מערכות שבהם זה לא כך.

לסיכום, להלן הכללים המומלצים שיכולים לעזור בקבלת ההחלטה :

1. אם המפתח הראשי עלול להשתנות (מס' טלפון, זו דוגמא טובה...) – עדיף להשתמש במקומו במפתח הטבעי.
לא תמיד קל לדעת האם הוא נתון לשינוי, אך למשל...
 - אם מקור המידע למפתח הטבעי מתקבל למערכת שלנו ממערכת מידע אחרת, אל תאמין להם כשהם אומרים שהמפתחות שלהם לא ישתנו לעולם. **חוק מרפי מוכיח את ההפך כל הזמן!**
 - אם המפתח מוגדר ע"י המשתמש והוא אומר לך ש"לדעתו הוא לעולם לא ישתנה". זהירות!! רב הסיכויים הם נגדך...
2. אם המפתח הראשי מורכב מיותר מ-3 שדות (ישנם כאלו שיאמרו גם עד 5 שדות, אולם אנחנו חושבים ש-3 זה בדיוק במידה). מאד מייגע להתעסק עם מפתח מורכב שהרי בסופו של דבר, תאורטית לפחות, נמצא את עצמנו נשארים עם טבלה שכמעט כל השדות שבה הם המפתח הראשי.
המצב מחמיר כאשר קיים מפתח זר (FOREIGN KEY) אליו, אשר יחייב שדות רבים בטבלה הקשורה אשר, לעיתים, יסרב גם אותה. וזה הופך רע יותר כאשר ישנם קשרים של רבים לרבים (M:M).
במקרים אלו, כדאי לשקול שימוש במפתח מלאכותי.
3. כאשר הטבלה המדוברת הנה בסה"כ טבלת פענוח קטנה – לא כ"כ מומלץ להשתמש במפתח מלאכותי, שהרי המפתח הטבעי שלה הוא-הוא הערך המשתתף בטבלאות האחרות. ואז – למה ליצור כפילות ובעייתיות מתוך כוונה תחילה?
4. מפתח טבעי, אשר מהווה זיהוי ח"ע וודאי, שאינו מורכב ממס' שדות רב (כבר אמרנו שלוש?) והסיכוי לשינויו קלוש (לדוגמא, מספר תעודת זהות) – נעדיף להשתמש בו כמפתח ראשי.
5. אם בחרנו להשתמש במפתח מלאכותי – ניצור תמיד מפתח חד-ערכי לא ראשי (UNIQUE KEY) על סמך המפתח הטבעי.

מ פ ת ו ח נ ע י ם ! ! !

