# **Database**

**Database**

מאגר נתונים – תוכנה המאפשרת לשמור, לנהל ולתשאל כמות גדולה של נתונים.

היא תמיד חיצונית לתוכנת ה-REST API שאנו בונים.

המידע נשמר תמיד בדיסק (קבצים / פורמטים אחרים)

קיימים שני סוגי Databases:

1. Relational Database – מסד נתונים רלציוני

זהו מסד נתונים ששומר את המידע שבו בטבלאות.

בין הטבלאות יש קישורים (לכן – Relational)

בכדי לתשאל את המידע קיימת שפת SQL (Structured Query Language)

דוגמאות: MySQL, Oracle, MSSQL, Postgres, SQLite ועוד.

1. NoSQL Database

זהו מסד נתונים ששומר את המידע שבו בפורמט אחר מטבלאות. לדוגמה פורמט JSON.

על מסד נתונים כזה שפת SQL לא פועלת.

בכדי לתשאל את המידע קיים API פנימי של מסד הנתונים.

דוגמאות: MongoDB, ElasticSearch, Firestore, Cassandra ועוד.

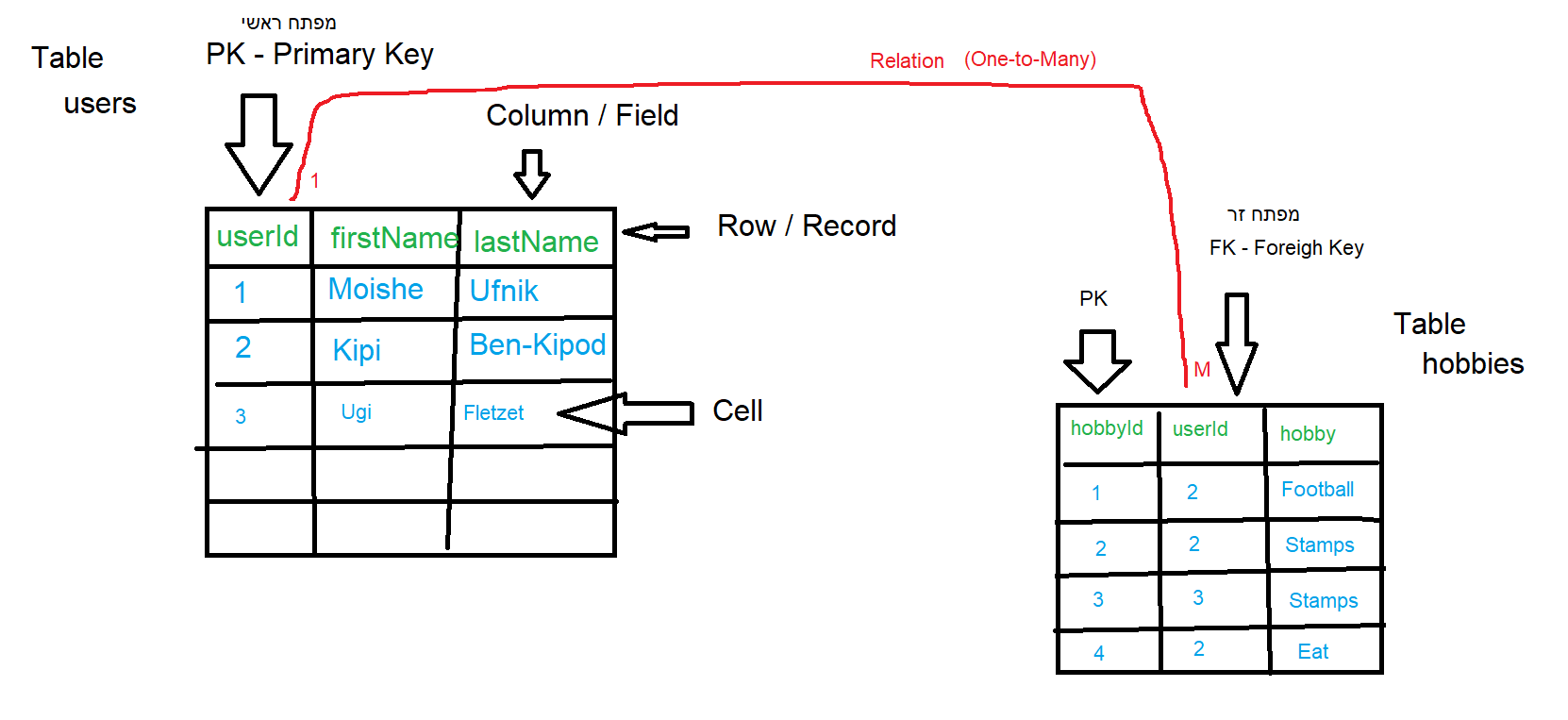
אם המידע מתאים יותר לטבלאות – הגיוני להשתמש במסד נתונים טבלאי.

אם המידע מתאים יותר לפורמט JSON – הגיוני להשתמש במסד נתונים NoSQL.

אם דרוש מידע עצום בגודלו עדיף לקחת NoSQL כי הם הרבה יותר מהירים.

אם דרושה מהירות – כנ"ל – עדיף NoSQL.

מונחים בעולם מסדי הנתונים הרלציוניים:



**XAMPP**

Cross – תוכנה זו קיימת עבור כל מערכות ההפעלה.

Apache – תוכנה המקבלה ל-Express – חושפת אתרים או שירותים

MySQL – מסד נתונים רלציוני מאוד פופולרי וחינאמי

Php – שפת תכנות

Perl – שפת תכנות

XAMPP זו תוכנה המכילה בתוכה את כל הנ"ל.

**סוגי טבלאות:**

1. Base Table – טבלת בסיס

זו טבלה המכילה את המידע המרכזי / ראשי במערכת שלרוב לא תלוי במידע אחר.

דוגמאות: טבלת תלמידים, טבלת מרצים, טבלת מוצרים, טבלת שחקנים, טבלת סרטים...

1. Detailing Table – טבלת פירוט

היא מפרטת עמודה שהיתה אמורה להיות בטבלה אחרת.

עמודה מכילה בכל תא ערך בודד.

תא של עמודה לא יכול להכיל מערך פריטים.

אם דרושים מספר ערכים בתא – יש לבנות טבלה נוספת במקום עמודה זו.

לדוגמה, במקום עמודת טלפונים – טבלת טלפונים.

1. Listing Table – טבלת רשימה

זה המקביל ל-Enum.

מכילה רשימה סגורה של פריטים, לכל פריט יש שם וקוד.

מיועדת בכדי שבטבלאות אחרות נגדיר את הקוד במקום את השם לצורך מניעת כפילויות שמות, מניעת שגיאות בשמות עצמם וכו'.

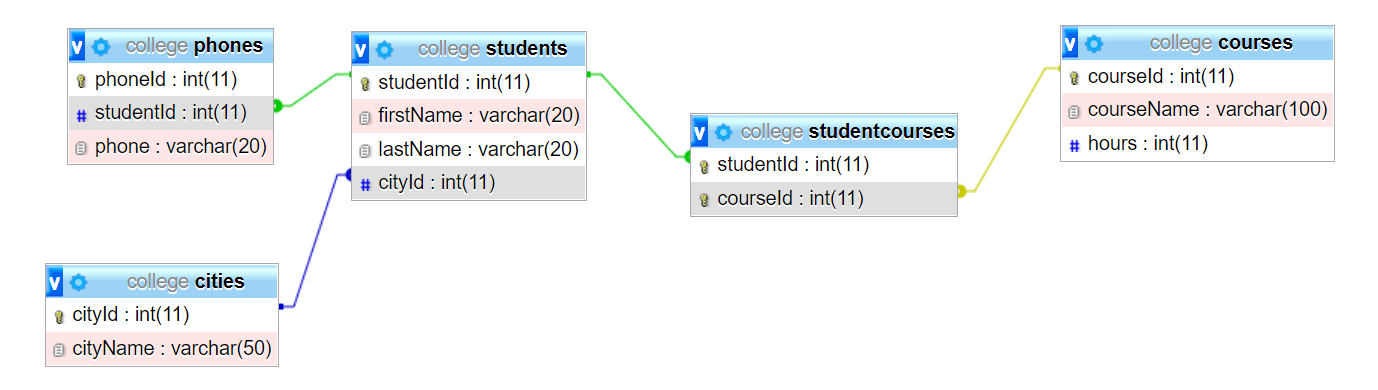
1. Association Table – טבלת נירמול

זו טבלה שמיועדת לייצג קשר של רבים לרבים בין שתי טבלאות.

כל מה שהיא מכילה זה עמודת מפתח ראשי מהטבלה הראשונה ועמודת מפתח ראשי מהטבלה השנייה.

שתי העמודות הללו מקושרות בקשר של יחיד לרבים עם הטבלאות שלהן.

אין צורך בעמודת מפתח ראשי נפרדת, אלא שתי העמודות הללו מייצגות יחד את המפתח הראשי.



**Relationship – קשר בין טבלאות**

שומר על תקינות המידע. ברגע שאנו מוסיפים מידע או מעדכנים מידע או מוחקים מידע, מסד הנתונים בודק האם אנו מבצעים פעולה חוקית מבחינת הקשרים שקיימים.

**סוגי קשרים**

1. One-to-Many

קשר יחיד לרבים.

צד אחד של הקשר הינו מפתח ראשי, צד שני של הקשר מפתח זר.

הוא נקרא יחיד לרבים כי כל ערך בעמודת המפתח הראשי יכולה להיות בעמודה זו רק פעם אחת, אך בעמודת המפתח הזר, ערך יכול להופיע מספר פעמים.

רוב הקשרים הם כאלו.

1. Many-to-Many

קשר של רבים לרבים.

רשומה בטבלה א' יכולה להיות קשורה למספר רשומות בטבלה ב', ורשומה בטבלה ב' יכולה להיות קשורה למספר רשומות בטבלה א'.

לדוגמה, תלמיד (רשומה בטבלה א') לומד הרבה קורסים (מס' רשומות בטבלה ב') וקורס (רשומה בטבלה ב') יכול להילמד ע"י מספר תלמידים (מס' רשומות בטבלה א')

אי אפשר לבצע טכנית קשר כזה במסד הנתונים ישירות בין שתי הטבלאות!

בכדי לבנות קשר כזה, יש לבנות את הסוג הרביעי של הטבלאות שציינו לעיל.

1. One-to-One

קשר של יחיד ליחיד.

זהו מצב בו שתי טבלאות מכילות מידע שונה לגבי אותו המפתח הראשי.

לדוגמה, טבלת לקוחות המכילה **קוד לקוח**, שם פרטי, שם משפחה, כתובת...

טבלת כרטיסי אשראי של לקוחות המכילה **קוד לקוח**, מס' כרטיס, סוג כרטיס, תוקף...

הקשר כאן הינו בין שני מפתחות ראשיים.