

SQL – lesson 1

Structured query language

נולד מתוך ההבנה שחייב מאגר נתונים!
עד כה היו בונים טבלה ובה מוזנים פרטי המשתמש האישיים וזה היה נשמר בקבצים: שורה עם פסיקים ביניהם.

מבנה זה היה בעייתי- כי לא ניתן היה לכתוב בקובץ מקביל ולאחות יחד את הנתונים והיה צריך לבנות כל מיני מנגנונים מסובכים, עד שזה הפך לשפת SQL

זה נחלק ל-2 רכיבים:

1. **מאחסן** את הנתונים – תוכנה- מאוד התפתח - big data
2. **שפה** – באמצעותה כותבים את הנתונים.

כיום קיימים מספר מאגרי נתונים עצומים שמקבלי מידע ממספר מקורות ויודעים לקלוט אותם כראוי.
אפאצ'י של סאן- המפורסמות והנפוצות בעולם. יש עוד המון נפוצות.

אחסנה

התוכנות שמאפשרות ל SQL לשמור את הנתונים בתוכם : כאן נכיר את SQLITE הוא כמו רכיב כי הוא מאוד קליל וכמעט ולא דורש התקנות, כיסחו לו המון אלמנטים של הגנה, אין בקרה על הנתונים שנכנסים בפנים. אפשר להצהיר על טיפוס מסוים ובפועל להזין טיפוס אחר – כאן האחריות של המתכנת שהנתונים יהיו תקינים!

על ההפסד הזה (חוסר דיוק) יש רווח עצום- ברגיל צריך שרתים רבי עוצמה להחזקת מאגרי נתונים כאלו עצומים אך ה- SQLITE יכול לשבת על טלפון סלולרי ולקלוט כמות עצומה של מידע!
ברמה זאת שלא חושבים פעמים ובוחרים את SQLITE מהירות הזנת נתונים ושליפתם מהירה ביותר. אנדרואיד עובד עם SQLITE : אנשי קשר וכדומה. ההצהרה על עבודה עם SQLITE היא מאוד פשוטה- מידי יוצר טבלאות להזנת משפטים הוא יורש מ (לא קלטתי)

שפה SQL

עליו להכיל 3 נקודות:

1. יצירה ומחיקה של טבלאות – חלקי או מלא.
2. הכנסה ומחיקה לעדכון נתונים.
3. שליפת נתונים – חלקיים או מלאים

מספר פקודות:

create- יצירה ומחיקה של טבלאות
drop-

insert- הכנסה, מחיקה ועדכון נתונים

update- הכנסה, מחיקה ועדכון נתונים

delete- הכנסה, מחיקה ועדכון נתונים

select- שליפת נתונים

בכדי להתחיל לעבוד איתו צריך להוריד

<http://sourceforge.net/projects/sqliteman/files/latest/download>

יורד אוטומטי.

הדבר השני- היכולת שלנו ללמוד לבד:

<http://www.w3schools.com/sql/default.asp>

ההמלצה ללמוד מהאתר הזה.

הציפייה ממפתח אנדרואיד היא לא שליטה מושלמת ב-SQL אלא הכרה בהיבטים של הזנת ושליפת נתונים.

תמיד תמיד תמיד – נשמור לנו מקום ל- key !

key – ערך ייחודי שלא חוזר פעמיים, היכולת הייחודית שלנו לאתר שורת נתונים ספציפית. זה למעשה נתון ייחודי למאגר הנתונים – למשל – אימייל, ת.ז וכדומה. נעדיף להשתמש ב- auto increment שהוא מנהל key אוטומטי, לכל שורה נוסף 1. ההמלצה שאם אנו בוחרים key משלנו נוסיף עליו **גם** את האוטומטי.

תחביר – **אותיות גדולות** = מילים שמורות. **אותיות קטנות** = ערכים.

INSERT

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

שם הטבלה

הערכים שמוזנים בטבלה

הנקודה פסיק בסוף איננו חובה- זה תלוי במערכת – המקובל הוא כן לשים! כל ערך לעטוף בגרש **בודד** .

במקרה והזנת הנתונים באוטו אינקרימנט לא צריך תת קידומת לכל שורה בטבלה.

-DELETE

```
DELETE FROM table_name  
WHERE some_column=some_value;
```

החלק הראשון נזין את שם הטבלה – החלק הזה עלול למחוק את כל הטבלה.

אם נרצה רק חלק ממנה יש לנו מילה שמורה – WHERE אם יש id לשורה נוסיף לפקודה id=5; מאפשר למחוק- טבלה, עמודה, שורה.

הספירה מתחילה מ-1.

כשאנחנו עובדים עם ArrayAdapter יש 4 פרמטרים – הרביעי נותן את האידי האמיתי של הדאטא בייס.

עוד דבר חשוב- נניח ומחקתי את שורה מספר 3 – בפעם הבאה שאני יעשה אינסרט כבר **לא** יהיה שימוש במספר 3 ! כשאנחנו מתעסקים בקבצים חיצוניים – זה אזור שהוא מחוץ לג'אווה ומה שקורה שם מחוץ לקלעים אינו ברור לנו ואין לנו שליטה עליו לא נוכל להגיד לו לצמצם את הטבלה זה תפקידו- לנהל את ה-key, לצמצם רווחים וכו'. יש חשיבות לכך שאין מחזור שורות זה מספר ייחודי – העיקר שלכל אחד יהיה מפתח ייחודי משלו.

אחת המהפכות של השנתיים האחרות הוא הביג דאטא והוא מתחיל להאפיל על SQLITE והוא מתבסס על הענף יש לו יכולות עצומות, השימוש בו הופך לנפוץ יותר ויותר ומאומץ בחום ע"י החברות הגדולות.

עם זאת, מבחינת השאילות של SQL לא נרשמת מגמת שינוי והוא מוסיף להיות שקוף למשתמש.

UPDATE

```
UPDATE table_name  
SET column1=value1, column2=value2, ...  
WHERE some_column=some_value;
```

שם עמודה = ערך

אם לא נשים ערך ב-WHERE- הוא יעדכן את כל העמודה בערך החדש, כך: id=4 למשל הוא יחפש קודם כל את שורה 4 ואז יבצע את העדכון. כדי לבחור קבוצות שורות לעדכון אנחנו נשתמש במילים שמורות (לא במסגרת הקורס).

SELECT

```
SELECT column_name, column_name  
FROM table_name;
```

הכול – select* כוכבית אומר הכול ולא מגביל לעמודה ספציפית – וכשנרצה עמודה ספציפית:

```
Select* from user WHERE id=2
```

בשורה הראשונה מאפשר לבחור מספר עמודות אך באותה שורה וללא WHERE

- היום לומדים SQL מנותק מאנדרואיד כדי למנוע בלבול. בהמשך נלמד בהקשר של אנדרואיד.

שימוש ב-SQLITE:

עבר ימני על טבלה- הזנת שם עמודה

בספר הלימוד שלנו מראים שימוש עם share comender אך צור לא ממליץ להתחיל ממנו כי הוא קשה יותר ללמידה ופחות מוחשי בעניין הדברים שמאחורי הקלעים.

Not null = בשורה שבה מגדירים עמודות, אם נסמן בוי זה אומר שזהו שדה חובה
id – PK integer – מגדיר את העמודה כ id

לג'אווה יש כלים חזקים שלא מאפשרים הזנת נתון שאינו תואם את הטיפוס ולכן כאן "הכול נלעס"

העבודה מול העורך – לא לסמוך על הסידור שלו- חייב לעשות מיון או לחלק לפי קבוצות ולדעת איזה פריט לבקש ספציפי ולא לסמוך על המיון שלו.

מסקנות-

אין מחויבות לטיפוס שהגדרנו. הוא אוכל הכול! אך בגלל שאנחנו עובדים מול ג'אווה הוא כבר מגן לנו על הנתונים. זה מוסיף יתרון בביצועים.
• זה לא יעבוד מהר בלולאת אינסרט.

CREATE TABLE

```
CREATE TABLE table_name  
(  
    column_name1 data_type(size),  
    column_name2 data_type(size),  
    column_name3 data_type(size),  
    ....  
);
```

בג'אווה נצטרך רק שם שמור כדי לעשות את המונה האוטומטי כך: _id