בס"ד

**כריית מידע במשימות מתמטיות**

ספטמבר 2017 , אלול תשע"ז

****

מאת:

תמר באש

200956834

[tamarbash1@gmail.com](mailto:tamarbash1@gmail.com)

052-7555330

מנחה:

פרופ' רחל בן-אליהו זהרי  
rbz@jce.ac.il

# תקציר

פרויקט זה התחיל כשיתוף פעולה עם המחלקה להוראת המדעים במכון ויצמן, במטרה לפתח מערכת אינטראקטיבית לפתרון שאלות במתמטיקה. המערכת מיועדת למורים המעוניינים לבחון את רמת הידע של התלמידים בנושאים שונים מתוך תכנית הלימודים של כיתה י'. השאלות פותחו על ידי צוות מתחום הוראת המדעים, אני התבקשתי לפתח את המערכת האינטראקטיבית שתאפשר למורים לפרסם את השאלות, לתלמידים - לפתור את השאלות ולקבל ציונים ולמורים - לבחון את פתרונות התלמידים.

כבר בשלב הראשוני זיהינו כי היקף הפרויקט לא יהיה מספק ולכן הוספנו לו חלק שני של כריית מידע על בסיס הנתונים של המערכת. כלומר, לאחר שהתלמידים פתרו שאלות מתמטיות באמצעות המערכת, המידע נשמר בבסיס נתונים ואת המידע ניתן לנתח. כיום קיימות מערכות רבות לכריית מידע המתממשקות עם בסיסי נתונים שונים. החלטנו שבחלק זה נבחר מערכת כריית מידע קיימת שתתממשק עם המערכת שלנו ותאפשר ניתוח המידע בצורה פשוטה ונוחה.

בהמשך הוספנו חלק שלישי לפרויקט – חלק מחקרי העוסק בנושא meta metaqueries – שאילתות על, גם חלק זה שייך למחלקה של כריית מידע מתוך בסיסי נתונים. בחלק זה התבססתי על מאמר שנכתב ב2002 על ידי פרופ' רחל בן-אליהו זהרי ושותפיה למחקר. מאמר זה עוסק בmeta metaqueries ככלי לכריית מידע, באמצעותו ניתן לחפש קשרים בין טבלאות ועמודות בבסיס הנתונים. במסגרת חלק זה של הפרויקט פיתחתי נושא המופיע בסיכום המאמר כעניין שנשאר פתוח וכתבתי אלגוריתם שייתן מענה לחלק זה.

לצורך הפשטות, לאורך רוב מסמך זה אתייחס לשני החלקים הראשונים של הפרויקט (בתיאור הבעיה, הפתרון וכדומה) ובסוף המסמך אקדיש פרק נפרד לחלק שלישי המחקרי שנעשה.

תוכן עניינים

[תקציר 2](#_Toc494379868)

[הצהרה 4](#_Toc494379869)

[תודות 5](#_Toc494379870)

[רקע 6](#_Toc494379871)

[מהות ותיאור הבעיה 6](#_Toc494379872)

[תיאור הפתרון 6](#_Toc494379873)

[בניית המערכת 7](#_Toc494379874)

[מערכת ה-moodle 7](#_Toc494379875)

[בסיס הנתונים 10](#_Toc494379876)

[מערכות כריית מידע 11](#_Toc494379877)

[תוצאות שהתקבלו 13](#_Toc494379878)

[Metaqueries Generator - חלק מחקרי 15](#_Toc494379879)

[מבנה metaquery 16](#_Toc494379880)

[שימוש בחישובים קודמים כתוצאות ביניים 17](#_Toc494379881)

[אלגוריתם חיפוש בעץ 18](#_Toc494379882)

[האלגוריתם 19](#_Toc494379883)

[נכונות האלגוריתם 20](#_Toc494379884)

[מימוש 20](#_Toc494379885)

[אפשרויות הרחבה ושיפור ביצועים 21](#_Toc494379886)

[רשימת ספרות 22](#_Toc494379887)

# הצהרה

העבודה הזו נעשתה בהנחיית פרופ' רחל בן-אליהו זהרי

במכללה האקדמית להנדסה ירושלים – עזריאלי

המחלקה להנדסת תוכנה.

החיבור מציג את עבודתי האישית

ומהווה חלק מן הדרישות לקבלת תואר שני

בהנדסת תוכנה M.Sc..

# תודות

ברצוני להודות לכל אלו שעזרו לי במהלך ביצוע עבודה זו, ובמיוחד:

לפרופ' רחל בן-אליהו זהרי על הליווי הצמוד, הזמינות, החשיבה המשותפת וההכוונה לאורך כל הדרך

לד"ר יעקב אקסמן, ראש התכנית לתואר שני, על המוטיבציה והאכפתיות

לצוות הסיסטם של המכללה על העזרה בעניינים הטכניים

לצוות הוראת המדעים במכון ויצמן על הזמן שהקדישו עבורי והרצון הטוב

למשפחתי שתמכה, תומכת ומאמינה בי לאורך כל הדרך

ולבורא עולם על הכל.

# רקע

במכון ויצמן פיתחו משימות מתמטיות המתייחסות לתכנית הלימודים ל-5 יחידות במתמטיקה בכיתות י'. מטרת המשימות היא לאפשר למורים לקבל מידע רלוונטי על מצב התלמידים בזמן אמת. לכל משימה מצורפת טבלת שגיאה אפשריות ומשמעותן, כך שבהינתן שגיאה של תלמיד קל לזהות את הקושי שלו ולהושיט לו את העזרה שצריך. המשימות מגוונות ובמבנים שונים, רובן בנויות מכרטיסים שצריך לסדר בסדר מסוים.

# מהות ותיאור הבעיה

המערכת מכילה שאלות כתובות על דפים וכרטיסיות. מההיבט של הארגון פחות נוח לעבוד ככה, הן למורה והן לתלמיד. למורה קשה לנתח את תוצאות התלמידים והקשיים שלהם ולקבל תמונה רחבה של מצב הכיתה, לתלמידים העבודה מסורבלת, אי אפשר להסתכל שוב על פתרונות שנעשו מוקדם יותר.

מההיבט ההנדסי מתפספס פה מידע רב שניתן להשיג בקלות עם בניית מערכת מתאימה והפיכת השאלות לממוחשבות, כמו: שמירת היסטורית הפתרונות עבור התלמיד, בדיקה כמה זמן בממוצע לוקח לתלמיד לפתור שאלה, חישוב ממוצע כיתתי ועוד.

# תיאור הפתרון

הפתרון שבחרתי ליישם כולל:

1. מערכת המאפשרת לכתוב, לשתף ולפתור שאלות מתמטיות
2. בסיס נתונים השומר מידע על הפתרונות
3. מערכת כריית מידע קיימת, באמצעותה אפשר ללמוד ולהסיק מסקנות על הנתונים.

Math System

DB

Data Mining System

## בניית המערכת

במטרה להתחיל לבנות את המערכת ביצעתי אפיון ראשוני בתיאום עם הארגון.

דרישות המערכת:

* המערכת תשמש מורים ותלמידים ולכן נדרשת לתמוך בשני סוגי משתמשים אלו.
* המערכת תתמוך בנוסף במשתמש מסוג admin האחראי להוסיף משתמשים למערכת ובעל הרשאות מקסימליות.
* המערכת תשמור נתונים על המשתמשים.
* המערכת תאפשר ליצור משימות משחקיות מסוגים שונים.
* מורים יוכלו לכתוב ולערוך שאלות מסוגים שונים ולפרסם אותן לתלמידים.
* מורים יוכלו לטעון שאלות קיימות.
* תלמידים יוכלו לפתור שאלות.
* מורים יוכלו לצפות בפתרונות התלמידים ולתת ציון במידת הצורך.
* תלמידים יוכלו לקבל את הציונים ולצפות בפתרונות.

בשלב הזה הארגון החליט שהוא מעוניין בעבודה בהיקף קטן משמעותית – לא מערכת שלמה עם משתמשים ואפשרות יצירת משימות אלא בסך הכל להפוך משימה או שתיים לממוחשבות ואינטראקטיביות. כאן נפרדו דרכינו והחלטתי להמשיך עם המשימה המקורית: מערכת עם משתמשים, אפשרות כתיבת משימות מתמטיות ופתירתן וחיבור למערכת כריית מידע.

כיוון שבניית המערכת אינה עיקר הפרויקט החלטנו לבנות מערכת בסיסית ברמת אבטיפוס במטרה למלא את בסיס הנתונים במידע ולהתקדם הלאה.

חיפשתי framework שיחסוך עבודה מיותרת ויאפשר לי לבנות במהירות מערכת על פי הדרישות שהוגדרו. לאחר חיפוש החלטתי להשתמש במערכת ה-moodle.

## מערכת ה-moodle

מערכת המודל הינה מערכת קוד פתוח חינמית המשמשת כסביבת הוראה ולמידה מקוונת. המערכת כוללת אפשרויות לניהול קורסים, הצגת חומר לימודים, עריכת מבחנים, פורומים לתקשורת עם תלמידים ומורים אחרים ועוד. moodle היא מערכת פופולרית ופועלת ביותר מ-200 מדינות. המערכת כתובה בphp ומחוברת לבסיס נתונים sql.

מערכת זו מיועדת לשמש כסביבת למידה ולכן מתאימה לצרכים שלנו מעל ומעבר למה שיספק לנו כל framework אחר. בשלב זה התמקדתי בלמידת המערכת, ניסיון להבין במה מתוך הדרישות שהגדרתי תומכת מערכת המודל ובמה לא והצצה לקוד ולבסיס הנתונים.

יתרונות השימוש במערכת ה-moodle

1. תמיכה בניהול משתמשים בעלי הרשאות שונות: מנהל, מורה ותלמיד.
2. גמישות רבה ואפשרות לשנות הגדרות כמעט בכל פעולה נדרשת כך שניתן בקלות להתאימה לדרישות הספציפיות שלנו.
3. כיוון שהמערכת פופולרית מאד ובעלת קוד פתוח ישנם מאות משתמשים סביב העולם שמשפרים אותה ומוסיפים לה תוספים חדשים (plugins) ולכן קיימים דברים רבים שבאופן בסיסי לא נתמכים במערכת אך ניתן להתקין plugin שתומך בהם.
4. הממשק מוכר ולכן המשתמשים במערכת שלנו, בהנחה שנתקלו בmoodle בעבר, ידעו להתמצא בו בקלות

חסרונות השימוש במערכת הmoodle-

1. קיימות אפשרויות רבות שאינן רלוונטיות למערכת שלנו
2. הממשק אינו ייחודי ומעט מסורבל

לאחר שבחרתי להשתמש בmoodle כבסיס למערכת התחלתי להקים את המערכת ולהכניס מידע.

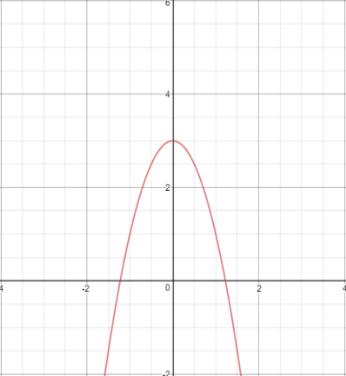
כתבתי משימות מתמטיות חדשות והשתדלתי להישאר נאמנה לסגנון המקורי. יצרתי קורס במערכת והכנסתי בו את המשימות.

דוגמה למשימה מתמטית (המשלבת גרירת כרטיסים):

נתון גרף של פונקציה מוזז אופקית או אנכית

התאימו בין הפונקציות והגרפים המתארים את ההזזה המתאימה

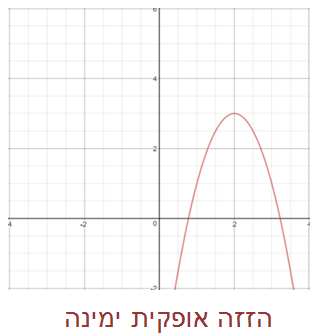
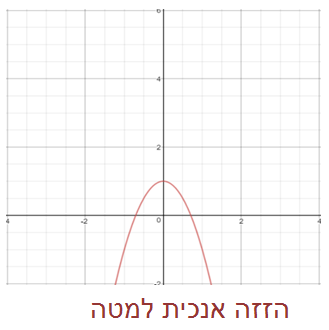
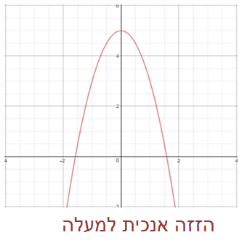
הפונקציה המקורית



y=-2x^2+3

y=-2(x-2)^2+3 y=-2x^2+5

y=-2(x+2)^2+3 y=-2x^2+1



משוב לתלמיד:

|  |  |
| --- | --- |
| בתבנית הבאה:  y = Af (Bx + C) + D  מייצג הזזה אופקית. C  C > 0 משמעותו הזזה שמאלה  C < 0 משמעותו הזזה ימינה | עבור שגיאה בהזזה אופקית |
| בתבנית הבאה:  y = Af (Bx + C) + D  מייצג הזזה אופקית. D  D > 0 משמעותו הזזה כלפי מעלה  D < 0 משמעותו הזזה כלפי מטה | עבור שגיאה בהזזה אנכית |

בשלב הבא יצרתי 100 משתמשים מסוג תלמיד ו-20 משתמשים מסוג מורה והזנתי את פרטיהם למערכת.

## בסיס הנתונים

בסיס הנתונים של moodle הינו טבלאי ומכיל טבלאות רבות (כ-350). לא קל להבין על מה אחראית כל טבלה ואילו טבלאות יש לעדכן בהוספת אובייקטים למערכת.

עם זאת, moodle מספקת כלים לניהול בסיס הנתונים והוספת אובייקטים בקלות באמצעות ממשק המשתמש. כלים אלו מאפשרים גמישות רבה והתאמה אישית לדרישות המערכת. לדוגמה, באופן רגיל עבור המשתמשים המערכת שומרת שדות בסיסיים כמו שם פרטי, שם משפחה, שם משתמש וסיסמא, אימייל ועוד. אני רציתי לשמור כמה שיותר מידע אישי על המשתמשים כמו גיל, מין וכו. המערכת מאפשרת לי להגדיר שדות חדשים והם יתווספו לטבלאות המתאימות בבסיס הנתונים. כמובן שביצירת משתמש, נוכל להזין ערך מתאים בשדה החדש וזה יישמר בבסיס הנתונים.

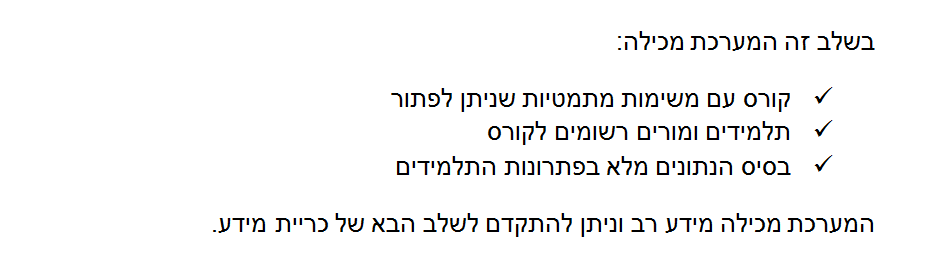
שדות שבחרתי לשמור עבור המשתמשים במערכת (אלו שאינם מסומנים בכוכבית הוגדרו על ידי):

עבור מורה:

* שם פרטי\*
* שם משפחה\*
* גיל (תאריך לידה)
* מין
* בית ספר
* וותק (שנת התחלת עבודה)
* השכלה

תלמיד:

* שם פרטי\*
* שם משפחה\*
* גיל (תאריך לידה)
* מין
* בית ספר
* מורה
* ציונים/ממוצע



מערכות כריית מידע

בחיפושיי אחר מערכת מתאימה לעבודה עם moodle נתקלתי במונח Education Data Mining – כריית מידע בחינוך.

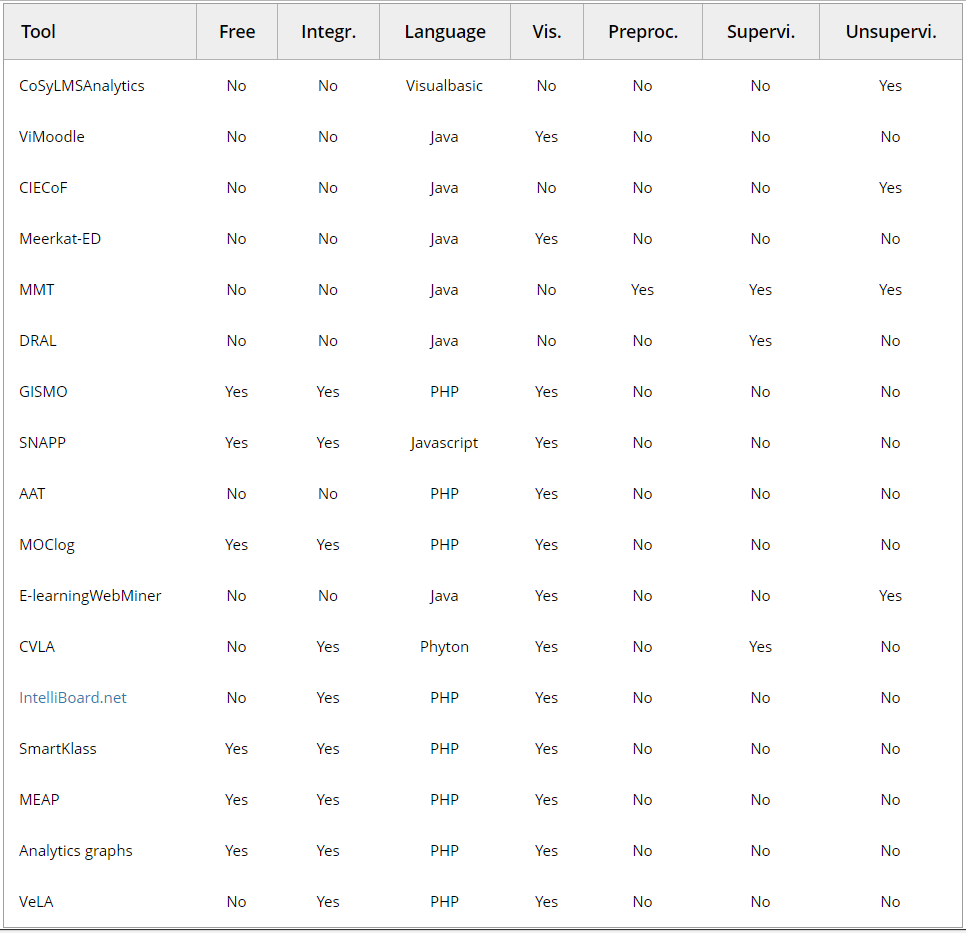
כריית מידע בחינוך מיישמת שיטות של [כריית מידע](https://he.wikipedia.org/wiki/%D7%9B%D7%A8%D7%99%D7%99%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) על מנת לענות על צרכים שונים של מערכות חינוך (לדוגמה [אוניברסיטאות](https://he.wikipedia.org/wiki/%D7%90%D7%95%D7%A0%D7%99%D7%91%D7%A8%D7%A1%D7%99%D7%98%D7%94) ו[בתי ספר](https://he.wikipedia.org/wiki/%D7%91%D7%99%D7%AA_%D7%A1%D7%A4%D7%A8) ,(ובהם הערכת חומרי הלמידה והתאמתם, הערכת תהליכי למידה, זיהוי התנהגויות של לומדים ועוד. מטרת התחום היא לפתח ולשפר את השיטות לחקר הנתונים הללו על מנת לגלות תובנות חדשות על האופן שבו אנשים לומדים (מתוך ויקיפדיה).

ככל שגובר השימוש בסביבות עבודה לימודיות מקוונות כך גובר הצורך במערכות כריית מידע בחינוך. מערכות אלו נותנות תמונה רחבה על הנעשה בכיתה, עשויות לאתר קשיים אצל תלמידים ואפילו לחזות הצלחה או כישלון. בתחום זה קיימים עקרונות ייחודיים כמו קוד אתי להסרת כל סממן זיהוי ממשתמשי המערכת לפני תהליך כריית הנתונים.

מערכות מתאימות לmoodle

בשלב זה גיליתי יתרון נוסף לשימוש במערכת הmoodle: מאחר וmoodle- היא סביבת למידה פופולרית וחינמית, רבים מהמחקרים העוסקים בכריית מידע בחינוך בחרו ליישם את המחקר דווקא על מערכת זו, וקיימים מאמרים רבים המשווים בין מערכות כריית מידע המתאימות לעבודה מול moodle.

הטבלה הבאה לדוגמה, מובאת מתוך מאמר המציג כלי חדש לכריית מידע המותאם לmoodle ומשווה בין מערכות קיימות:



הדרישות שלי ממערכת כריית מידע: שתוכל להתממשק עם בסיס הנתונים של moodle ולהציג את הנתונים באופן ויזואלי.

סקרתי מספר מערכות כריית מידע:

**MDM tool**

פותח בספרד על ידי קבוצת מחקר בראשות פרופסור Cristóbal Romero מאוניברסיטת קורדובה, לאחר שבמאמרים קודמים בחן את הנושא של כריית מידע בלמידה מרחוק והכלים השונים. מיישם טכניקות כריית מידע שונות.

חינמי אך לא זמין להורדה (חדש מ2017).

**GISMO - moodle plugin**

**G**raphical **I**nteractive **S**tudent **Mo**nitoring Tool for Moodle

מטרתו של כלי זה לדמות מה קורה בכיתה בשיעורי למידה מרחוק ולהציג את המידע על פעילויות התלמידים באופן ויזואלי. חינמי ומתחבר למערכת כplugin-.

**FMDEV**

מאפשר גילוי מידע מתוך בסיס הנתונים של moodle , מספק משוב מהיר על התקדמות התלמידים בקורסי למידה מרחוק. לא זמין (חדש מ2017)

**SmartKlass – moodle plugin**

מודד ומנתח את תהליך הלמידה בכל עת בקורסיMoodle . מטרתו לעזור למורים לעקוב אחר ההתקדמות של התלמידים בקורסי למידה מרחוק. זהו תוסף dashboard.

**AAT - Academic Analytics Tool**

מאפשר זיהוי חומרי למידה מורכבים. מספק ייצוג גרפי, משולב במודל כבלוק. לא חינמי.

**TADA-Ed**

**T**ool for **A**dvanced **D**ata **A**nalysis for **Ed**ucation

מאפשר ניתוח וגילוי דפוסים רלוונטיים בתרגילים הניתנים אונליין, משתמש בטכניקות שונות לכריית מידע. גם כלי זה פותח באקדמיה לשימוש מורים אך אינו זמין להורדה.

מבין הכלים הנ"ל בחרתי להשתמש ב-GISMO כיוון שהוא זמין להורדה, חינמי, קל ונוח לשימוש ומציג את המידע באופן ויזואלי.

# תוצאות שהתקבלו

לאחר חיבור התוסף GISMO ניתן להציג באופן ויזואלי נתונים על פעולות הסטודנטים בקורס, ציונים, גישה למשאבים ועוד.

Moodle platform

Moodle database

Math System

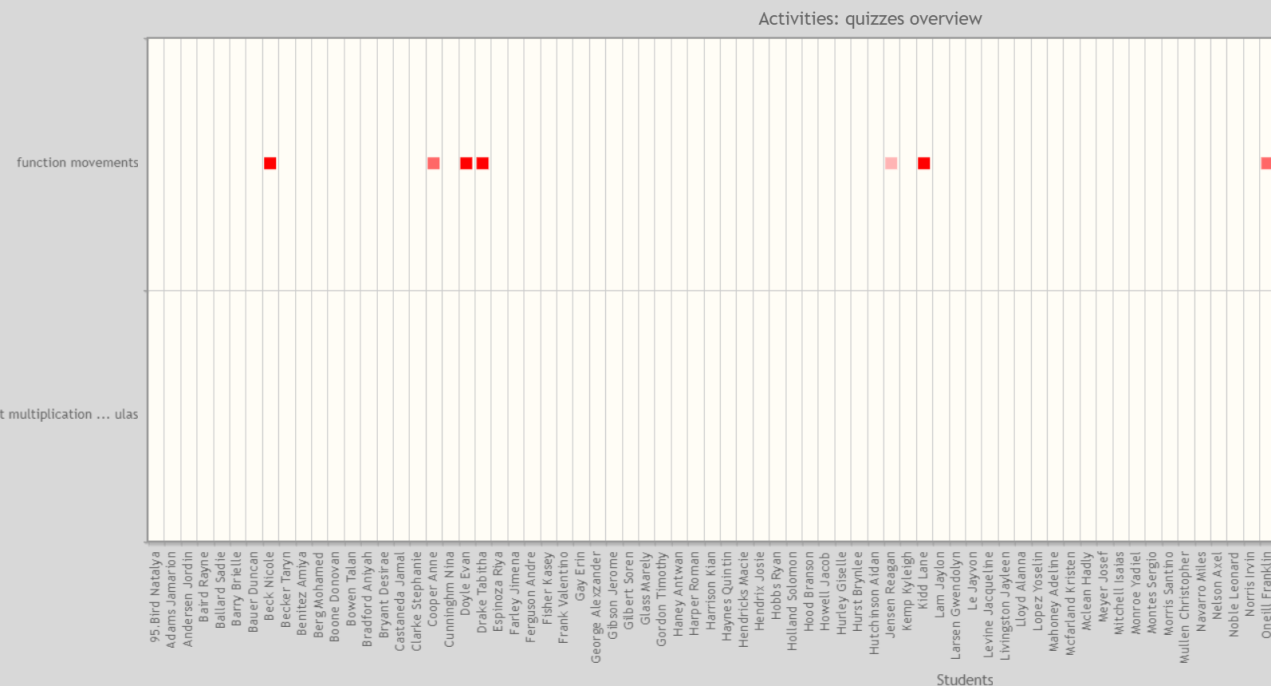
DB

Data Mining System

GISMO plugin

חשוב לציין שגם moodle מאפשרת הצגת דוחות וסטטיסטיקות, אך שם הנתונים הם ברמת התלמיד ואילו GISMO נותן תמונה רחבה של כל הכיתה.

דוגמה לגרף:



גרף זה מספק מידע בנוגע למי וכמה מהסטודנטים הגישו בוחן וציוניהם.

כלי זה אמנם מספק ויזואליזציה של המידע ולא מיישם אלגוריתמים של כריית מידע, עם זאת ניתן להשתמש בו לכריית מידע ולהיעזר בו להבנת תהליכי למידה. לדוגמה: האם גישות רבות למשאבים משפיעות בהכרח על ציונים גבוהים יותר? הגרפים יכולים לענות על השאלה בקלות יחסית.

# Metaqueries Generator - חלק מחקרי

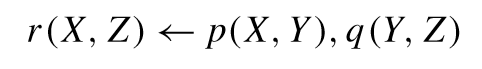
חלק זה מתבסס על המאמרR. Ben-Eliyahu-Zohary, E. Gudes, G. Ianni, Metaqueries: **Semantics, complexity, and efficient algorithms, Artificial Intelligence** (2003)*.*

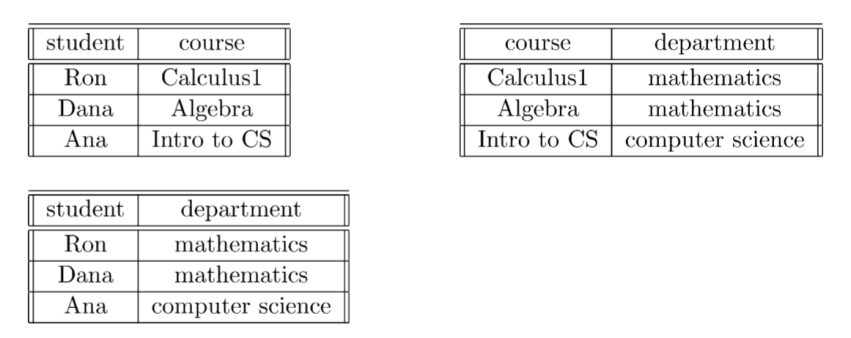
המאמר מגדיר את המושג Metaquery (שאילתת על) ומרחיב כמה מהנושאים הקשורים אליו.

metaquery היא למעשה תבנית שאנו מנסים למצוא התאמה אליה בבסיס הנתונים ובאופן זה לגלות קשרים נסתרים בין הטבלאות.

דוגמה מתוך המאמר:

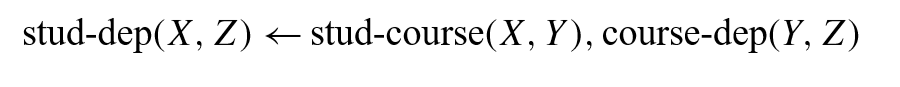
Metaquery





התאמה שנמצאה בטבלאות בבסיס הנתונים

כלל שניתן להסיק



ההתאמה שמצאנו גילתה לנו כלל חדש: אם סטודנט לוקח קורס והקורס שייך למחלקה מסוימת, אזי מסתבר שהסטודנט שייך לאותה מחלקה.

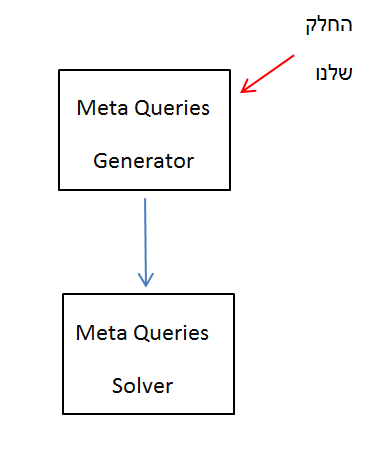
לרוב הכללים המתגלים לא יהיו נכונים במאה אחוז מהמקרים, לכן קיימים ערכי support ו-confidence המייצגים את רמת הוודאות של הכלל. כלל הוא בעל ערך אם רמת הוודאות שלו (ערכי support ו-confidence) גבוהה מ-threshold שקבענו.

פתרון ל-metaqueryהוא טבלאות ועמודות ספציפיות בבסיס הנתונים שהרשומות בהן מקיימות את תבנית ה-metaquery עם ערכי support ו-confidence גבוהים מה-threshold שנקבע.

הבעיה של מציאת פתרון לmetaquery היא NP-hard.

עד כאן מתוך המאמר.

במחקר הנוכחי לא נתעכב על מציאת פתרון לmetaquery נתון אלא נרצה לייצר metaqueries שאותן יש לבדוק.

מתוך סיכום המאמר:

"Several research topics remain open in the field of data mining using metaqueries. One interesting direction is to develop a system that will generate the metaqueries automatically. A simple way to do this is to go over all the possible combinations. A more sophisticated approach would be to learn in which direction the interesting information can be found from answers to preliminary metaqueries…" *(Metaqueries: Semantics, complexity, and efficient algorithms,* *[Rachel Ben-Eliyahu-Zohary](http://www.sciencedirect.com/science/article/pii/S0004370203000730" \l "!)**,* *[Ehud Gudes, Giovambattista Iann](http://www.sciencedirect.com/science/article/pii/S0004370203000730" \l "!)i).*

בהנחה שיש מכונה שיכולה לענות על metaqueries, נרצה לספק לה את התבניות שאותן היא תפתור על פי סדר מסוים. כיוון שחישוב metaquery הוא פעולה יקרה, נשאף שתהליך זה יהיה מתוחכם ויעיל יותר מאשר ייצור רנדומלי של כל התבניות האפשריות. תוצר המחקר יהיה אלגוריתם המייצר metaqueries.

תכונות האלגוריתם:

1. מייצר את כל הmetaquries העשויות להיות בעלות פתרון
2. מביא להוזלת עלויות חישוב ה-metaqueries

## מבנה metaquery

דוגמא לתבניות:

r(X, Z) ← p(X, Y ), q(Y, Z)

r(X, Z) ← p(X, Y ), q(Y, Z), s(Z,W)

בצד שמאל תמיד יופיע יחס אחד, כלומר טבלה מתוך בסיס הנתונים עם משתנים המייצגים עמודות ממנה (כולן או חלקן). לכל היותר תופיע טבלה על כל עמודותיה, לכל הפחות תופיע טבלה עם שתי עמודות בלבד.

בצד ימין תופיע טבלה אחת או יותר עם משתנים המייצגים עמודות מתוך הטבלאות. לכל היותר יופיעו כל הטבלאות על כל עמודותיהן. לכל הפחות יופיעו 2 טבלאות בלבד ו-2 משתנים בכל טבלה, כאשר אחד המשתנים משותף (מאותו type).

הסכמה היא סופית (מספר הטבלאות ומספר העמודות בכל טבלה) ולכן גם מספר הmetaqueries האפשריים הוא סופי. נרצה לייצר ולבדוק את כל הmetaqueries האפשריים כיוון שלא נרצה לפספס קשר משמעותי.

## שימוש בחישובים קודמים כתוצאות ביניים

בהינתן metaquery ופתרון עבורה, ניתן להשתמש בפתרון זה כתוצאות ביניים לפתרון מהיר ויעיל עבור שאילתה המערבת משתנה/יחס אחד נוסף באופן הבא:

נניח יש לנו פתרון ל:

r(X, Z) ← p(X, Y ), q(Y, Z)

ואנחנו רוצים למצוא פתרון ל:

r(X, Z) ← p(X, Y ), q(Y, Z), s(X,Z)

(נזכיר, פתרון ל metaquery הוא טבלאות ועמודות שערכי support ו-confidence של ההתאמה שלהן עומדים ב-threshold שקבענו)

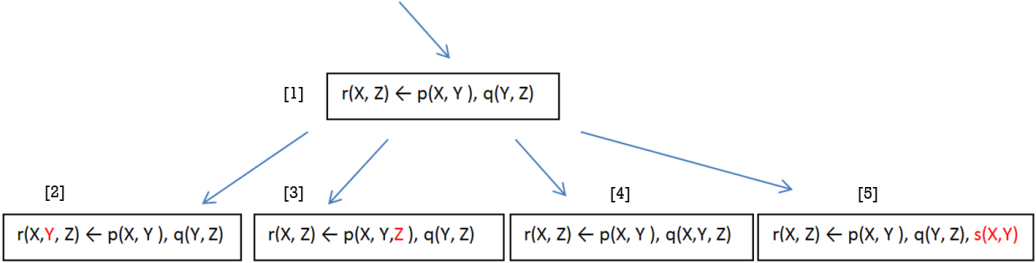
בהינתן פתרון של שאילתה, שאילתה זהה עם הוספת יחס נוסף תהיה קלה יותר לפתרון: נעבור על מרחב הפתרון הקודם ונבדוק אילו מהיחסים עדיין מתקיימים. כיוון שהוספנו אילוץ, מרחב הפתרון יהיה בהכרח קטן/שווה לפתרון הקודם. לכל היותר נעבור על כל הDB, אך ברוב המקרים נצמצם את מרחב החיפוש.

אם כן, עם כל הוספת יחס ל-metaquery נקטין את מרחב החיפוש והדרך לפתרון תהיה קצרה ופשוטה יותר. ניתן להתייחס ל metaquery המקורי כאל "אב" ול metaqueries החדשים כאל "בנים". אם נחשב את הבנים לאחר חישוב האב נוכל להשתמש בתוצאת החישוב של שאילתת האב ולהפוך את חישוב הבנים לזול ומהיר יותר. כמובן שאם לשאילתה אין פתרון גם לבנים אין פתרון ואין טעם לבדוק אותם.

נציע דרך לפתרון אוסף השאילתות באופן יעיל:  
נשמור את כל השאילתות במבנה נתונים של עץ כאשר לכל metaquery יהיו בנים שמהווים את אותה שאילתה בתוספת יחס אחד נוסף. סדר פתרון השאילתות יהיה על פי העומק בעץ, כאשר העלים יחושבו אחרונים. ככל שמתקרבים לעלים כך הפתרון קל יותר. נרצה לעבור על כל העץ: בכל פעם לחשב metaquery ולייצר את הבנים, לחשב את הבנים וכו' עד לסיום.

לדוגמה:

שאילתה [1] היא ה"אב", שאילתות [5]-[2] הן דוגמאות לבנים. כאשר נייצר את התבניות, נייצר קודם את [1] ונשלח לפתרון. לאחר מכן נייצר את הבנים ונספק לsolver- את פתרון האב לצורך חישוב מהיר יותר.



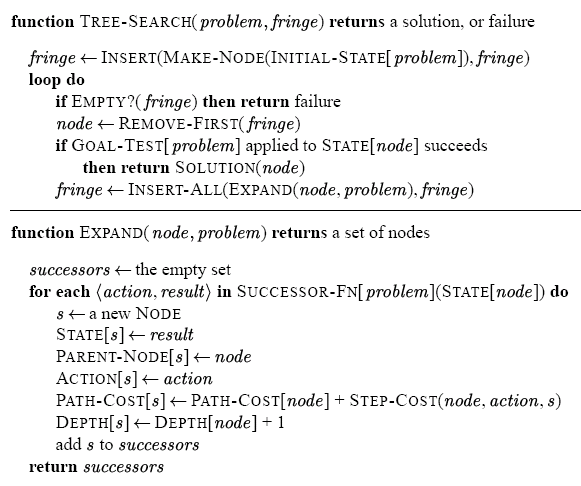
למעשה, הפתרון המוצג כאן דומה לאלגוריתם חיפוש בעץ.

## אלגוריתם חיפוש בעץ

אלגוריתם חיפוש בעץ משתמש במבנה של עץ כדי למצוא פתרון לבעיות חיפוש. בבעיות חיפוש יש מרחב מצבים, מצב התחלה ומצב מטרה המסודרים כקדקודים בעץ, כאשר ניתן לבצע פעולות שונות כדי לעבור ממצב למצב. האלגוריתם מחפש שביל בעץ (כלומר סדרת פעולות) שיוביל ממצב התחלה למצב מטרה. האתגר: הבעיה יכולה להיות אקספוננציאלית. מבנה העץ לא ידוע מראש והוא נבנה תוך כדי ריצה. לכן אי אפשר להשתמש באלגוריתמים פשוטים של גרפים.

אלגוריתם חיפוש בעץ בונה את עץ החיפוש תוך כדי ריצה ומשתמש בתור כדי לשמור קדקודים. בתחילה מכניסים לתור את מצב ההתחלה, בכל איטרציה שולפים קדקוד מהתור ובודקים אם זה מצב מטרה. אם כן – סיימנו. אם לא – מייצרים את הבנים ומכניסים אותם לתור בסדר מסוים.

אלגוריתם חיפוש בעץ גנרי:



האלגוריתם שלנו יעבוד באופן דומה, כאשר:

* אצלנו אין "מצב מטרה". נרצה לחשב את כל השאילתות האפשריות.
* אין מצב התחלה אחד. בתחילה יוכנסו לתור כל השאילתות הכוללות מספר מינימלי של יחסים (2 טבלאות).
* כאשר לשאילתה אין פתרון אין צורך לייצר את הבנים.
* הסדר בתוך התור ייקבע על פי עומק הקדקוד בעץ.
* האלגוריתם אינו מסתיים בהגעה למצב מטרה אלא לאחר שבדקנו את כל השאילתות האפשריות, כלומר כאשר התור ריק.

לכן אצלנו האלגוריתם עובד באופן הבא:

נכניס לתור שאילתות ראשוניות הכוללות מספר מינימלי של יחסים ומשתנים. בכל פעם נשלוף קדקוד (metaquery) וננסה לפתור. אם מצאנו פתרון – נייצר את הבנים ונכניס אותם לתור, תוך שמירה על תוצאת החישוב של שאילתת האב. אם אין פתרון לא נייצר בנים אלא נמשיך ונוציא מהתור את ה- metaquery הבא. כאשר התור ריק - סיימנו.

הסדר בתוך התור ייקבע על פי עומק הקדקוד בעץ (אפשר לחשב בקלות בעזרת מספר המשתנים המעורבים בשאילתה) – נחשב תחילה שאילתות גבוהות יותר בעץ ונתקדם לכיוון העלים.

## האלגוריתם

MQs\_Generator(DB\_schema, support\_value,confidence\_value, fringe)

{

Fringe 🡨 initial\_MQs (schema)

loop do

If Empty (fringe) then return

MQ 🡨 remove\_first (fringe)

Sol 🡨 solver (MQ, schema, support\_value, confidence\_value) // calc MQ by external machine

If (sol)

send\_report(MQ, sol)

MQ\_collection🡨 Expand (MQ, schema, fringe, sol)

Insert(MQ\_collection, fringe)

}

הפונקציה Expand מקבלת metaquery ומייצרת את כל הבנים, באמצעות הוספת משתנה או יחס אחד נוסף בכל פעם.

הפונקציה Insert מכניסה לתור את השאילתות החדשות שנוצרו. לפני הכנסת metaquery לתור תבדוק שאינה קיימת, כיוון שאותה שאילתה יכולה להיווצר על ידי שני אבות שונים. הסדר בתוך התור ייקבע לפי הרמה של הקדקוד בעץ.

## נכונות האלגוריתם

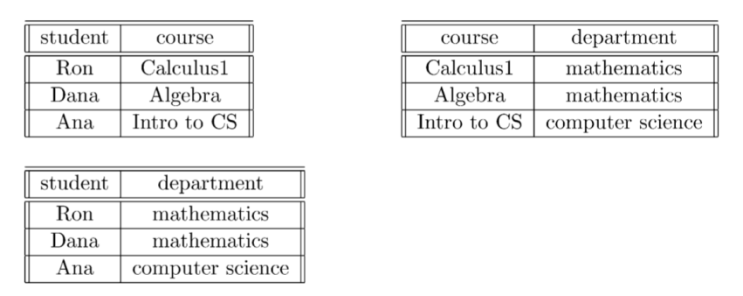
יצירת כל ה-metaqueries העשויות להיות בעלות פתרון

אנו מתחילים מהשאילתות המינימליות האפשריות ועבור כל שאילתה מייצרים את כל השאילתות האפשריות בעלות יחס אחד נוסף. גם עבור השאילתות החדשות ממשיכים ומייצרים את כל השאילתות האפשריות בעלות יחס אחד נוסף, וכך ניתן להגיע עד לשאילתה המקסימלית, המערבת את כל הטבלאות וכל העמודות של בסיס הנתונים. נפסיק לייצר בנים לשאילתה רק אם אין לה פתרון ובהכרח לשאילתות הבנים אין פתרון ואין טעם לייצר אותם.

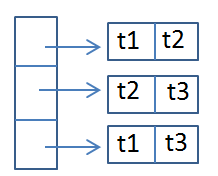
הוזלת עלויות החישוב

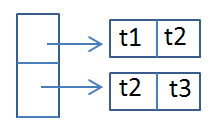
כיוון שאנו נעזרים בתוצאות ביניים, רוב השאילתות יחושבו באופן מהיר יותר ללא צורך לעבור על כל הטבלאות בבסיס הנתונים. שאילתות רבות לא ננסה לפתור כלל כיוון שנזהה שהן בנים לשאילתה שאין לה פתרון. כיוון שנפתור את השאילתות על פי סדר הרמות בעץ, בהכרח נזהה שאילתה שכבר קיימת בתור ולא ננסה לפתור אותה יותר מפעם אחת.

## מימוש

את הסכמה של בסיס הנתונים ניתן לייצג באמצעות מערך דו ממדי שגודלו כמספר הטבלאות בסכמה, וכל אינדקס במערך מכיל את מספר וסוג העמודות השייכות לאותה טבלה. לדוגמה, עבור הסכמה:

נממש אותה כמערך דו ממדי כך:



metaquery תמומש גם היא כמערך דו ממדי, אשר גודלו כמספר הטבלאות המעורבות ב-metaquery וגודל כל מקום במערך כמספר המשתנים בטבלה זו המעורבים ב-metaquery. המקום הראשון במערך הינו ראש ה-metaquery, והשאר הינו גוף ה-metaquery.

לדוגמה, השאילתה r(x,y) -> p(y,z) לסכמה המוזכרת למעלה, תיוצג כך:

בתחילת הריצה מאתחלים את התור ב- metaqueries המינימליות. metaquery מינימלית היא metaquery המערבת 2 טבלאות בלבד ו-2 משתנים בכל טבלה, כאשר אחד המשתנים משותף (מאותו type).metaquery מקסימלית מערבת את כל הטבלאות וכל המשתנים הקיימים בסכמה.

בהינתן metaquery, הבנים הם כל האפשרויות להוספת משתנה אחד לטבלאות המעורבות ב- metaquery המקורי או להוסיף טבלה חדשה עם שני משתנים, כאשר לפחות משתנה אחד משותף לטבלאות הקודמות.

**הערכת ביצועים**

נניח N מספר הטבלאות בסכמה, M מספר העמודות המקסימלי בטבלה. יש אפשרויות לבחירת הטבלאות עבור metaquery (כל טבלה ניתן לבחור או לא לבחור) ו-אפשרויות לבחירת העמודות עבור כל טבלה.



בסך הכל האלגוריתם יכול לייצר לכל היותר = metaqueries שונים.



למעשה המספר יהיה קטן הרבה יותר כיוון שבכל פעם שנמצא metaquery ללא פתרון לא נמשיך לפתח את הבנים. בנוסף, קיימים אילוצים ליצירת metaquery לדוגמה: בהתווסף טבלה חדשה היא צריכה להכיל משתנה עם type שכבר קיים בטבלאות קודמות, גם אילוצים אלו יקטינו את מספר ה-metaqueries האפשריים.

## אפשרויות הרחבה ושיפור ביצועים

**fixed metaqueries**

במאמר נכתב כי קיים מקרה בו מענה על metaquery הוא פולינומיאלי, והוא כאשר מדובר בסכמה קבועה וב- metaquery קבועה:

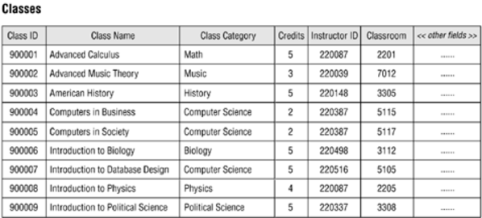
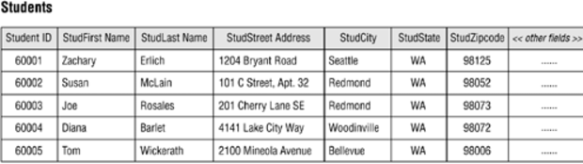
"We identify several tractable classes: answering a fixed query on a database with a fixed schema (but varying relation size) can be done in polynomial time"

ניתן להשתמש במקרה זה ולבחור מראש מספר metaqueries קבועים (לדוגמה, metaquery המייצגת יחס טרנזיטיביות) וסכמה קבועה של בסיס הנתונים. ישנן סכמות שהן קבועות כמו בסיס נתונים רלציוני, את האחרות נצטרך להתאים. באופן הזה יהיו לנו מספר metaquery קבועות שניתן לחשבן בזמן פולינומיאלי והן יהיו הראשונות שנבחן.

**significant metaqueries**

בחלק זה נרצה שיהיה ברשותנו ידע מקדים על שאילתות משמעותיות. כמובן שמידע וודאי ומדויק לא נוכל לקבל מראש. אך ננסה למצוא דרכים להעריך האם metaquery תהיה משמעותית. לפני הכל, metaquery משמעותית היא metaquery שתספק לנו מידע בעל ערך. מידע בעל ערך חייב להיות מידע שכיח. אחרת, אם קיים קשר כלשהו אך הוא נדיר, אין לו משמעות אמתית עבורנו. זהו בדיוק התפקיד של ערכי הsupport והconfidence- , לוודא שקשרים שמצאנו קיימים בשכיחות גבוהה מספיק בבסיס הנתונים. אם כך, השאלה היא: איך נוכל לחזות ערכי support ו-confidence גבוהים? נציע דרך לעשות זאת באמצעות ביצוע מדגם על בסיס הנתונים:

ננסה להחיל את התבניות על חלק קטן של בסיס הנתונים, כלומר לפתור את ה-metaquery, אך על מדגם מתוך בסיס הנתונים. המדגם עשוי להיות על השורות או על העמודות.



ככל שנמצא יותר התאמות כך נשער ערכי support וconfidence גבוהים יותר. אפשר לבחור את החלק מבסיס הנתונים באופן רנדומלי או לבקש מהמשתמש שיצביע לנו על טבלאות ועמודות חשובות בבסיס הנתונים וזה החלק שנדגום. לדוגמה, אם נחפש מידע מעניין בבסיס נתונים של בנק, המשתמש יוכל להצביע על כך ששדה המשכורת הוא חשוב, לעומת שדה השם שהוא חשוב פחות. אם כך, לאחר שבחרנו טבלאות ועמודות לדגימה, כל metaquery תיבדק על חלק זה ותקבל ציון על החשיבות שלה בהתאם למספר ההתאמות שנמצא.

נוכל להשתמש בזיהוי metaqueries משמעותיות לצורך סידור הקדקודים בתור. הסדר בתוך התור ייקבע על ידי שני מדדים:

1. עומק הקדקוד בעץ
2. בין הקדקודים השייכים לאותה רמה בעץ נבחר לחשב את המשמעותיים ביותר תחילה.

# רשימת ספרות

**Knowledge Discovery in Databases**

[1] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. **Knowledge discovery in databases: an overview**. AAAI / MIT Press (1991). <http://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1011/929>

[2] Romero, C., Ventura, S. **Educational Data Mining: a Survey from 1995 to 2005**. Expert Systems with Applications, 1, 33, 135-146 (2007). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.4774&rep=rep1&type=pdf>

**Data mining tools**

[3]J. M. Luna, C. Castro, C. and Romero, **MDM tool: A data mining framework integrated into Moodle**, Computer Applications in Engineering Education, Vol. 25 (2017). <http://onlinelibrary.wiley.com/doi/10.1002/cae.21782/full>

[4] Angelo F. Dias Gonçalves and Alexandre M. A. Maciel, Rodrigo L. Rodrigues, **Development of a Data Mining Education Framework for Data Visualization in Distance Learning Environments** (2016) <http://ksiresearchorg.ipage.com/seke/seke17paper/seke17paper_130.pdf>

[5] Merceron, A., & Yacef, K. **Tada-ed for educational data mining**. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 7(1), 267–287 (2005). <http://www.imej.wfu.edu/articles/2005/1/03/index.asp>

[6] Romero, C., & Ventura, S. **Data mining in education**. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(1), 12–27 (2013). <http://onlinelibrary.wiley.com/doi/10.1002/widm.1075/full>

[7] Romero, C., Ventura, S., Salcines, E. **Data mining in course management systems: Moodle case study and tutorial**. Computer & Education, 51(1), 368-384 (2008). <http://research.moodle.net/95/1/Romero%20(2008)%20Data%20mining%20in%20course%20management%20systems-%20Moodle%20case%20study%20and%20tutorial.pdf>

**EDN – Education data mining**

[8] **Wikipedia** - <https://en.wikipedia.org/wiki/Educational_data_mining>

[9] Ben-Zadok, G., Leiba, M., & Nachmias, R. **Comparison of online learning behaviors in school vs. at home in terms of age and gender based on log file analysis**. Interdisciplinary Journal of E-Learning and Learning Objects, 6, 305–322 (2010). <http://www.ijello.org/Volume6/IJELLOv6p305-322Zadok703.pdf>

**Metaqueries**

[10] R. Ben-Eliyahu-Zohary, E. Gudes, G. Ianni, **Metaqueries: Semantics, complexity, and efficient algorithms,** Artificial Intelligence, 149(1):61–87 (2003) <http://www.sciencedirect.com/science/article/pii/S0004370203000730>

[11] Wei-Min Shen, KayLiang Ong, Bharat Mitbander, Carlo Zaniolo, **Metaqueries for Data Mining**. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA ,pp. 375-397 (1996).

<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9217B6E3598E0D2D564A921A117392A0?doi=10.1.1.28.4694&rep=rep1&type=pdf>

[12] F. Angiulli, R. Ben-Eliyahu-Zohary, G. Ianni, and L. Palopoli. **Computational properties of metaquerying problems**. ACM Trans. on Comput. Logic, 4(2):149–180 (2003). <https://arxiv.org/pdf/cs/0106012.pdf>