הטכניון – מכון טכנולוגי לישראל הפקולטה למדעי המחשב

מרצה: פרופ' חגית עטיה סמסטר חורף התשפ"א

מתרגלים: אסף ישורון

אלעזר גרשוני

נועה שילר

גל גרימברג

מסדי נתונים

236363

מועד ב'

15 במרץ 2021

פירוט החלקים והניקוד:

שאלה	נושא	ניקוד	הערות
1	SQL	20	
2	RA	17	
3	Datalog	20	
4	Design	20	
5	MongoDB, NEO4J	23	

הנחיות לנבחנים

1. יש להשתמש רק בסימנים או פונקציות שנלמדו בתרגול או בהרצאה ומופיעים בשקפים של הקורס. כל שימוש בסימון שאינו כזה מחייב הסבר מלא של משמעות הסימון.
2. הזמן המוקצה למבחן הינו שלוש שעות, תכננו את הזמן בהתאם.
3. אין לכתוב בעפרון. יש להקפיד על כתב יד ברור ולסמן בצורה נקייה את השאלות והסעיפים.
4. ניתן להשתמש בכל חומר עזר שנגיש ישירות מה-Moodle. אין להשתמש במקלדת.

בהצלחה!

שאלה 1 — SQL:

נתון מסד הנתונים הבא, המכיל מידע על משתמשים ופרסומים ברשת חברתית:

Users(uID, Name, Age)

הטבלה מכילה מידע על המשתמשים ברשת. לכל משתמש שמור מספר מזהה שלו, שמו וגילו.

Posts(pID, uID)

הטבלה מכילה מידע על פרסומים (פוסטים) ברשת. לכל פוסט שמור מזהה ייחודי, ומספר מזהה של המשתמש שפרסם אותו.

Likes(uID, pID)

הטבלה מכילה מידע על פוסטים שסומנו ב-"אהבתי" (לייק) על ידי משתמשים. רשומה בטבלה משמעה שמשתמש עם מזהה uID אוהב את הפוסט עם מספר מזהה pID.

שימו לב: מפתחות הסכמות מסומנים בקו תחתון.

ענו על הסעיפים הבאים. ניתן להשתמש בשאילתות מקוננות ובמבטים (VIEWS).

א. (4 נקודות) כתבו קוד SQL המגדיר את הטבלה Likes (השתמשו בפקודה CREATE TABLE). שימו לב: מזהה של משתמש ומזהה של פוסט שניהם צריכים להיות ערכים מספריים חיוביים. בנוסף, עליכם לדאוג לכך שלא ניתן יהיה להכניס לטבלה רשומה בעלת uID שאינו מופיע בטבלה Users, וגם לא רשומה בעלת pID שאינו מופיע בטבלה Posts.

```
CREATE TABLE Likes (
uID INTEGER CHECK(uID>0) REFERENCES Users,
pID INTEGER CHECK(pID>0) REFERENCES Posts,
PRIMARY KEY(uID, pID)
);
```

ב. (4 נקודות) כתבו שאילתת SQL המחזירה את שמות כל המשתמשים שאהבו פוסט כלשהו. עליכם להחזיר את שמותיהם ללא חזרות.

```
SELECT DISTINCT users.name FROM
users INNER JOIN likes on users.uid=likes.uid;
```

ג. (6 נקודות) כתבו שאילתת SQL המחזירה את שמות כל המשתמשים שכל פוסט שלהם קיבל לייק אחד לפחות. שימו לב: יש להחזיר גם משתמשים שלא פרסמו פוסטים כלל (כיוון שהם מקיימים את התנאי באופן ריק).

```
CREATE VIEW posts_without_likes AS
SELECT uid, pid FROM posts WHERE pid not in (SELECT pid FROM likes);

SELECT name FROM users
WHERE uid not in (SELECT uid FROM posts_without_likes);
```

ד. (6 נקודות) מתכנני הרשת צריכים את עזרתכם בסינון משתמשים המפיצים כמות גדולה של תוכן לא איכותי ללא הבחנה. החזירו את המספרים המזהים של כל המשתמשים שפרסמו יותר פוסטים מאשר קיבלו לייקים.

```
CREATE VIEW likes_per_post AS
SELECT pid, COUNT(*) AS num_likes_per_post FROM likes GROUP BY pid;

CREATE VIEW posts_per_user AS
SELECT uid, COUNT(*) AS num_posts_per_user FROM posts GROUP BY uid;

CREATE VIEW likes_per_user AS
SELECT uid, SUM(num_likes_per_post) AS num_likes_per_user FROM
posts INNER JOIN likes_per_post on posts.pid=likes_per_post.pid
GROUP BY uid;

SELECT users.uid FROM
users INNER JOIN posts_per_user ON users.uid=posts_per_user.uid
INNER JOIN likes_per_user ON users.uid=likes_per_user.uid
WHERE num_posts_per_user > num_likes_per_user
```

שאלה 2 — RA:

סעיפים א-ד עוסקים בסכמה הכוללת את היחס $E(s,d)$. נתייחס ליחס כאל גרף מכוון של קשתות מ-s אל d, כאשר קבוצת הצמתים בגרף מוגדרת להיות הצמתים המופיעים ביחס E כצומת התחלה או צומת יעד.

א. (4 נקודות) כתבו שאילתה המחזירה את קבוצת הצמתים בגרף שאין להם קשת עצמית.

$$\left(\pi_s E \cup \rho_{s/d}\pi_d E\right) \setminus \pi_s \sigma_{s=d} E$$

ב. (5 נקודות) כתבו שאילתה המחזירה את כל הקשתות (a,b) כך שקיימת גם הקשת (b,a) בגרף.

$$E \cap \rho_{s/d,d/s} E$$

ג. (4 נקודות) כתבו שאילתה המחזירה את כל הזוגות (s,d) עבורן הגרף איננו טרנזיטיבי בצעד יחיד: קיים m כך ש $(s,m),(m,d) \in E$ אבל $(s,d) \notin E$.

$$\pi_{s,d}\left(\rho_{d/m}E \bowtie \rho_{s/m}E\right) \setminus E$$

ד. (4 נקודות) בהינתן יחס נוסף $D(d)$ המסמן קבוצה של צמתים בגרף, כתבו שאילתה המחזירה כל צומת s עבורו קיימים שני צמתים $d_1, d_2 \in D$ כך שיש קשת בין s לבין d_1 אך אין קשת בין s לבין d_2.

$$\pi_s(E \bowtie D) \setminus (E \div D)$$

שאלה 3 — Design

נתונה סכמה R = (A,B,C,D,E) עם קבוצת התלויות הפונקציונליות הבאה:

$$F = \{A \longrightarrow B, BC \longrightarrow A, B \longrightarrow CD, C \longrightarrow D\}$$

א. (3 נקודות) מצאו כיסוי מינימלי ל- F.
{A→B,B→A, B→C,C→D}

ב. (4 נקודות) איזה יחסים בפירוק R_1=(A,B,D), R_2=(A,C,E), R_3=(A,B,E) הם בצורה נורמלית Boyce-Codd (BCNF)?
R_1 הוא BCNF כי A ו-B הם מפתחות, וכל התלויות הן תלויות במפתח.
R_2 אינו BCNF כי A אינו מפתח על (אינו קובע את E).
R_3 אינו BCNF כי A אינו מפתח על (אינו קובע את E).

ג. (3 נקודות) האם הפירוק מסעיף ב' משמר את התלות $B \longrightarrow C$?
כן. מצירוף B→A (R_1) עם A→C (R_2)

--- המשך השאלה הינו בלתי-תלוי ---

תלות הכלה מאפשרת לבטא מידע כגון "כל מנהל (ביחס מנהלים) הוא עובד (ביחס עובדים)".

פורמלית, עבור סכמת מסד נתונים תלות הכלה $\mathrm{R}[A_1, \dots, A_m] \sqsubseteq \mathrm{S}[B_1, \dots, B_m]$, כאשר:

- R, S הם שמות יחסים בסכמה.
- $A_1, \dots, A_m$ סדרת אטריבוטים שונים של R.
- $B_1, \dots, B_m$ סדרת אטריבוטים שונים של S.

מתקיימת אם $\pi_{A_1,\dots,A_m}(\mathrm{R}) \sqsubseteq \pi_{B_1,\dots,B_m}(\mathrm{S})$, כאשר ההכלה מתקיימת לפי סדר האטרביוטים המועבר.
כלומר, עבור כל רשומה t ב-R קיימת רשומה t' ב-S כך שלכל $1 \le i \le m$ מתקיים כי $t[A_i] = t'[B_i]$.
לדוגמא, עבור $\mathrm{R} = \{(A,B,C),\{(3,2,2)\}\}$ ו-$\mathrm{S} = \{(C,D,E),\{(1,2,2),(2,1,2)\}\}$
מתקיימת תלות הכלה $\mathrm{R}[B,C] \sqsubseteq \mathrm{S}[D,E]$.

נסתכל על קבוצות אטריבוטים W, X, Y, Z ושמות יחסים R, S, T

הוכיחו כי כללי ההיסק הבאים עבור תלויות הכלה הם נאותים (sound), כלומר כל טענה שניתן להוכיח בעזרתם היא נכונה:

ד. (3 נקודות) $\mathrm{R}[X] \sqsubseteq \mathrm{R}[X]$
לכל יחס R וקבוצת תכונות X, מתקיים $\pi_X(\mathrm{R}) = \pi_X(\mathrm{R})$ ולכן הכלל נאות.

ה. (3 נקודות) אם $\mathrm{R}[X] \sqsubseteq \mathrm{S}[Y]$ וגם $\mathrm{S}[Y] \sqsubseteq \mathrm{T}[Z]$ אז $\mathrm{R}[X] \sqsubseteq \mathrm{T}[Z]$
בגלל ש $\mathrm{R}[X] \sqsubseteq \mathrm{S}[Y]$ מתקיים $\pi_X(\mathrm{R}) \sqsubseteq \pi_Y(\mathrm{S})$
ובגלל ש $\mathrm{S}[Y] \sqsubseteq \mathrm{T}[Z]$ מתקיים $\pi_Y(\mathrm{S}) \sqsubseteq \pi_Z(\mathrm{T})$
מזה נובע כי $\pi_X(\mathrm{R}) \sqsubseteq \pi_Z(\mathrm{T})$
ולכן, לפי ההגדרה, $\mathrm{R}[X] \sqsubseteq \mathrm{T}[Z]$
והכלל נאות.

הוכיחו כי כלל ההיסק הבא המשלב תלויות הכלה ותלויות פונקציונליות הוא נאות (sound):

ו. (4 נקודות) נניח $|\mathrm{X}| = |W|$, אז מתלות הכלה $\mathrm{R}[XY] \sqsubseteq \mathrm{S}[WZ]$ ותלות פונקציונלית $W \longrightarrow Z$ על הסכמה S

ניתן להסיק את התלות $X \longrightarrow Y$ על הסכמה R.
זאת אומרת, אם שתי שורות t1, t2 ביחס R מקיימות t1[X]=t2[X] אז t1[Y]=t2[Y].

נסתכל על יחסים r, s שמקיימים את הסכמות ואת התלויות.
נניח כי ביחס r יש שתי שורות t1, t2 כך ש t1[X]=t2[X] (אנחנו צריכים להוכיח ש t1[Y]=t2[Y])
בגלל תלות ההכלה, ביחס s יש שתי שורות t1',t2' כך ש t1'[WZ]=t1[XY] וגם t2'[WZ]=t2[XY]
ולכן t1'[W]=t2'[W]
בגלל התלות הפונקציונלית, מקבלים ש t1'[Z]=t2'[Z]
וזה גורר ש t1[Y]=t2[Y]
מה שמוכיח כי $X \longrightarrow Y$

שאלה 4 — Datalog:

נתון מסד נתונים לתכנון מערכת שעות לסטודנטים, המכיל את הטבלאות הבאות:

Student(id, name)

Lecture(courseNumber, day, time)

Schedule(id, courseNumber, day, time)

הטבלה Student מכילה את מזהי ושמות הסטודנטים, הטבלה Lecture מכילה את זמני ההרצאה של הקורסים השונים והטבלה Schedule מכילה את מערכת השעות של הסטודנטים.

בסעיפים הבאים ניתן להשתמש ביחס Eq(x,y), המתקיים אמ"מ x שווה ל-y.

מערכת שעות של סטודנט מורכבת מכל ההרצאות המופיעות עם תעודת הזהות שלו ביחס Schedule.
מערכת שעות היא **חוקית** עבור סטודנט מסויים אם:

1. כל ההרצאות המופיעות במערכת השעות שלו מופיעות גם ביחס Lecture באותו הזמן בדיוק (כלומר, באותו יום ושעה).
2. אין שתי הרצאות חופפות במערכת המתקיימות בדיוק באותו זמן.
3. לכל קורס קיימת לכל היותר הרצאה אחת במערכת.

אם לסטודנט יש מערכת שעות ריקה (כלומר, אין עבורו רשומה ביחס Schedule) נגיד כי יש לסטודנט מערכת שעות חוקית.

א. (6 נקודות) כתבו תכנית datalog המגדירה את היחס Legal(id) המכיל את תעודות הזהות של סטודנטים שלהם יש מערכת שעות חוקית.

$$classNotExists(id) \leftarrow schedule(id,c,d,t), \neg lecture(c,d,t).$$
$$sameTime(id) \leftarrow schedule(id,c1,d,t), schedule(id,c2,d,t), \neg eq(c1,c2).$$
$$moreThanOneLec(id) \leftarrow schedule(id,c,d1,t), schedule(id,c,d2,t), \neg eq(d1,d2).$$
$$moreThanOneLec(id) \leftarrow schedule(id,c,d,t1), schedule(id,c,d,t2), \neg eq(t1,t2).$$

$$noLegal(id) \leftarrow classNotExists(id).$$
$$noLegal(id) \leftarrow sameTime(id).$$
$$noLegal(id) \leftarrow moreThanOneLec(id).$$

$$legal(id) \leftarrow student(id,n), \neg\, notLegal(id).$$

ב. (5 נקודות) כתבו שאילתת RC המחזירה את תעודות הזהות של סטודנטים שלהם יש מערכת שעות חוקית.

$$\{id : \exists n\big(student(id,n)\big) \wedge$$
$$\forall c,d,t\big(schedule(id,c,d,t) \rightarrow lecture(c,d,t)\big) \wedge$$
$$\forall c_1,c_2,d,t\,\Big(\big(schedule(id,c_1,d,t) \wedge schedule(id,c_2,d,t)\big) \rightarrow eq(c_1,c_2)\Big) \wedge$$
$$\forall c,d_1,t_1,d_2,t_2\,\Big(\big(schedule(id,c,d_1,t_1) \wedge schedule(id,c,d_2,t_2)\big)$$
$$\rightarrow \big(eq(d_1,d_2) \wedge eq(t_1,t_2)\big)\Big)\}$$

ג. (4 נקודות) נגיד כי סטודנט לקח קורס מסויים אם קיימת הרצאה כלשהי של הקורס במערכת השעות שלו. מסלול בין שני סטודנטים עם תעודות זהות id_1, id_2 הוא הרצף $v_1,\ldots,v_n$ כך שמתקיים $v_1=id_1$ ו-$v_n=id_2$ ולכל i מתקיים כי v_i ו-v_{i+1} הם מספרי ת"ז של סטודנטים שלקחו את אותו הקורס (לא בהכרח את אותה ההרצאה).כתבו תכנית datalog המגדירה את היחס OddPath(x,y) המכיל את

כל זוגות הסטודנטים שקיים מסלול באורך אי-זוגי ביניהם.

$$tookSameClass(id1, id2) \leftarrow student(id1, n1), student(id2, n2), \neg eq(id1, id2)$$
$$schedule(id1, c, d1, t1), schedule(id2, c, d2, t2)$$

$$oddPath(x, y) \leftarrow tookSameClass(x, y).$$
$$oddPath(x, y) \leftarrow evenPath(x, z), tookSameClass(z, y).$$

$$evenPath(x, z) \leftarrow oddPath(x, z), tookSameClass(z, y).$$

נתונה התכנית הבאה:

$$A(id) \leftarrow Schedule(id, c1, d1, t1), Schedule(id, c2, d2, t2), \neg Eq(c1, c2).$$
$$B(id) \leftarrow Student(id, n), Schedule(id, c, d, t), \neg A(id).$$

ד. (2 נקודות) כתבו במילים מה יהיה הפלט של התכנית המרובדת עבור הפרדיקט B.
ב-B יהיו תעודות הזהות של כל הסטודנטים שיש להם בדיוק קורס אחד במערכת השעות שלהם (ייתכן ויש כמה הרצאות במערכת השעות עבור אותו הקורס).

נתון ה-EDB הבא:

Student	
id	**name**
1	Alice
2	Bob
3	Mallory

Lecture		
courseNumber	**day**	**Time**
236363	Mon	12:30
234123	Mon	14:30

Schedule			
id	**courseNumber**	**day**	**Time**
1	236363	Mon	12:30
2	234123	Mon	14:30
3	236363	Mon	12:30

ה. (3 נקודות) ציינו את כל המודלים המינימליים של התוכנית עבור ה-EDB הנתון.

$B(1), B(2), B(3)$
$A(1), B(2), B(3)$
$B(1), A(2), B(3)$
$B(1), B(2), A(3)$
$A(1), A(2), B(3)$
$B(1), A(2), A(3)$
$A(1), B(2), A(3)$
$A(1), A(2), A(3)$

שאלה 5 – mongoDB/Neo4j:

חברת Taub security גילתה כי ישנם ניסיונות פריצה לחוות המחשבים הפקולטית. החברה החליטה לנתח את הרשת באמצעות **גרף Neo4j** כמתואר להלן.

מחשב הוא צומת עם מספר ip ייחודי, מחשב **אישי** הינו מחשב מחוץ לחוות המחשבים עם תווית PrivateComputers ואילו למחשב בחוות המחשבים יש תווית TaubComputers.

פורט הוא צומת המכיל תכונת id ייחודית המייצגת את מספר הפורט.

מחשב א' **מתחבר** למחשב ב' אם קיימת קשת מכוונת מסוג Connect היוצאת מצומת א' לצומת ב'.

מחשב **פתח פורט** x אם הוא מחובר בקשת מכוונת מסוג OpenPort לצומת בעל תווית פורט עם id = x.

הטיפוסים הם: ip) – String) (id) – integer

א. (6 נק') החברה רוצה למצוא את מספר המחשבים האישיים עם כתובת ip המתחילה ב127 אשר לא התחברו למחשב עם כתובת ip המכילה את תת המחרוזת 555. כתבו שאילתה המבצעת זאת.

```
match (c1:PrivateComputers)-[:Connect]->(c2)
 WHERE c2.ip CONTAINS '555' with c1, collect(DISTINCT c1.ip) AS bad
match (c3:PrivateComputers)
WHERE c3.ip =~'127.*' AND  NOT c3.ip in bad
return count(*)
```

ב. (6 נק') החברה רוצה למצוא את הip של כל המחשבים **האישיים** אשר מקיימים את התנאי הבא:

כל מחשב **מחוות המחשבים** אליו התחברו, התחבר לפחות לשני פורטים שונים.

כתבו שאילתה המבצעת זאת.

```
match ((c:TaubComputers)-[:OpenPort]-> (M) )
 with c, count(DISTINCT M.id) AS number where number > 1
 with c, collect(c.ip) as TwoOrMore
 match (c1:PrivateComputers)-[r:Connect]->(c2)
 where c2.ip in TwoOrMore
 return c1.ip
```

בעקבות שינויים ארגוניים,החברה החליטה לעבור למסד נתונים מסוג **MongoDB**. לרשותכם מסד נתונים ובו אוסף (Collection) בודד הנקרא Computers אשר מכיל מידע על חיבורי. כל מסמך באוסף הוא מהצורה הבאה:

```
{
        "_id": <ObjectId>,
        "ip_client": <string>,
        "connections": [
```

```
        {
            “ip_server”: <string>,
            "attempts" : <int>
        },
        …
        {
            “ip_server”: <string>,
            "attempts" : <int>
        }}
```

דוגמא למסמך אפשרי:

```
{
    “_id”: ObjectId(056ab84901a07b),
    “ip_client”: “127.0.0.1",
    “connections”: [
        {
            “ip_server”: "198.2.3.4 ",
            "attempts" : 5
        }
        {
            “ip_server”: "198.6.17.1 ",
            "attempts" : 15
        }
]}
```

ג. (6 נק') החברה גילתה שיש טעות במספר הנסיונות וצריך להוסיף 5 ניסיונות לכל ערך מסוג attempts. כתבו שאילתה המחזירה עבור כל ip_client את סכום ניסיונות ההתחברות (attempts) הנמצאים במערך connections לאחר התיקון ואת הסכום לפני התיקון, ממוינים מהסכום הגדול ביותר לקטן ביותר.

לדוגמא, עבור ה collection הנתון, הפלט יהיה:

“127.0.0.1" : prev_sum: 20

Sum: 30

```
db.getCollection('Q3').mapReduce
(
        function() { for (var idx = 0; idx < this.connections.length; idx++) {
                        emit(this.ip_client, this.connections[idx].attempts); }},
        function(key, values) {
            reducedVal = { sum: 0, prev_sum: 0 }
            reducedVal.prev_sum = Array.sum(values);
            reducedVal.sum = Array.sum(values) +  values.length *5;
            return reducedVal;
          }

        , {out: "Result :"}

).find({}).sort({value: -1});
```

ה. (5 נק') כתבו שאילתה שמחזירה את השדות ip_server, ip_client ושדה נוסף safe המכיל true אם החיבור לip_server הכיל פחות מ- 10 ניסיונות התחברות (attempts), אחרת, שדה זה לא יופיע כלל.

```
db.getCollection('Computers').aggregate([
{ $unwind: '$connections'},{
    $project: {
      "_id": 0,
      "ip_client": 1,
      "connections.ip_server" : 1,
      "safe": {
        $cond: [
          { $gte: [ "$connections.attempts", 10 ] },
          "$$REMOVE",
          true ]}}} ])
```