



פרויקט גמר SQL

נושא:

מגשימים נתונים

מאגר מידע עבור

תוכנית מגשימים



שם המרצה: איריס זרצקי

שם המגיש: ליחי סויסה

תז: 214907651

רקע תיאורטי

מגשימים – תוכנית הסייבר הלאומית היא תוכנית לימודים בתחום הסייבר והמחשבים לנוער מצטיין בעיקר בפריפריה הגאוגרפית והחברתית – שמטרתה להגדיל את מאגר המועמדים לשירות ביחידות הסייבר של צה"ל ומערכת הביטחון, כמו גם להביא להשתלבותם כמובילים בתעשיית ההיי-טק ובאקדמיה. התוכנית מופעלת כיום על ידי המרכז לחינוך סייבר.

בשלב הראשון נפתחה תוכנית "מגשימים" ב-2012 לתלמידי תיכון (י'-י"ב) בעלי יכולת גבוהה ונטייה לתחום המחשבים, בעלי מחויבות, התמדה ומשמעת עצמית. ב-2015 הורחב המיזם הלאומי לתלמידים בגילאים צעירים יותר והוא נקרא "ניצני מגשימים". התלמידים עוברים תהליך מיון לאחריו הם מתחילים בתהליך הכשרה הכולל מפגשים דו שבועיים (או חד שבועיים בניצני מגשימים) של לימודי מחשבים אינטנסיביים כל מפגש 3 שעות. בנוסף מתקיימות פעילויות גיבוש, העשרה חברתית, תרומה לקהילה, מחנות סייבר אינטנסיביים בחודשי הקיץ, וסיורים בחברות היי-טק. התוכנית מפוצלת ל 2 מסלולים: סייבר, ובינה מלאכותית. יש כ-3300 בוגרים של התוכנית (כולל מתוכנית ממריאות) 70% מתקבלים ליחידות סייבר וטכנולוגיה בצה"ל, 86% ממשיכים לאקדמיה או לתעסוקה בתחומי ההייטק.



מטרות פונקציונליות

במערכת זו ארצה לרכז בעיקר את הנתונים לגבי השנה השלישית והמרכזית בתוכנית - שנת הפרויקטים. שבה החניכים מתבקשים ליצור פרוייקט בתחום ידע כלשהו אשר יתרגם את יכולותיהם וכישוריהם אשר רכשו במהלך השנים למוצר מוגמר.

החניכים נעזרים בבוגרי התוכנית וברכזי התוכנית על מנת לייצר מוצר ברמה גבוהה אשר יעזור להם בעתיד בראיונות לעבודות וליחידות בצה"ל

מטרות המערכת שלי היא לתת גישה קלה ומהירה למידע אודות שנת פרויקטים למנהלי מערכות המידע במגשימים, מידע על הפרויקטים עצמם, על החניכים, על כיתות ורכזים ובעיקרון לשמור ולעקוב אחרי מידע לגבי בוגרי התוכנית וחניכים שנה ג'.

מכיוון שכ 60% מנרשמי התוכנית בכיתה י פורשים עד כיתה יב לא ראיתי לנכון לאגור מידע גם עליהם מכיוון שהסבירות שאדם יפרוש לפני שנת הפרויקטים גבוהה ולא ארצה לאחסן מידע לא שימושי במערכת

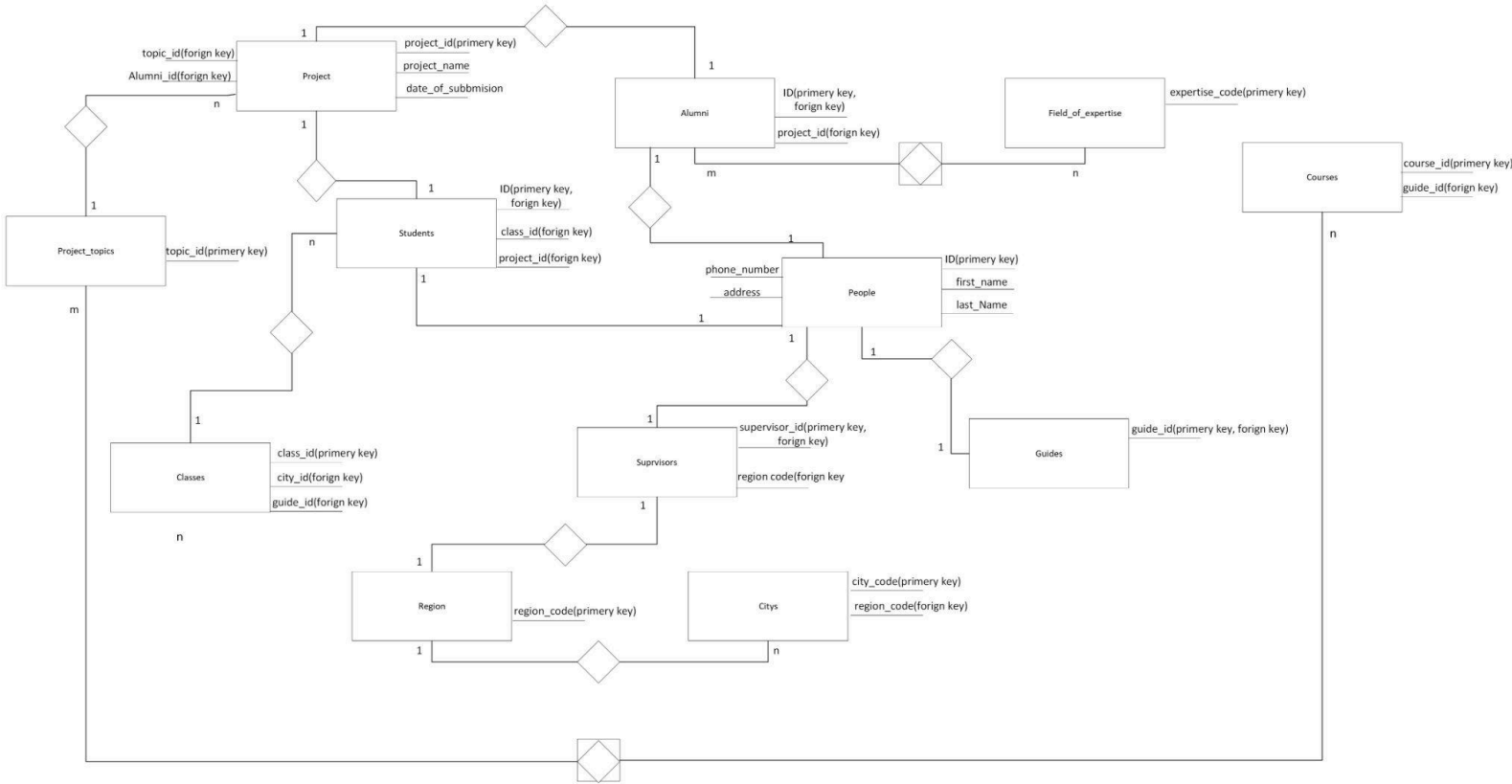
הנחות יסוד לנרמול

אנשים - מפתח כל אדם הוא ייחודי ומייצג את תעודת הזהות שלו
תלמידים - לכל תלמיד פרוייקט אחד ייחודי
התמחות בוגרים - לכל בוגר יש התמחות אחת בלבד
פרוייקטים - לכל פרוייקט מוגדר נושא אחד בלבד
בוגרים - כל בוגר יכול ללוות מספר פרוייקטים
כיתות - בכל כיתה מספר תלמידים ולכל תלמיד כיתה אחת
קורסים - בכיתה יש מספר קורסים
מדריכים - לכל כיתה יש מדריך אחד

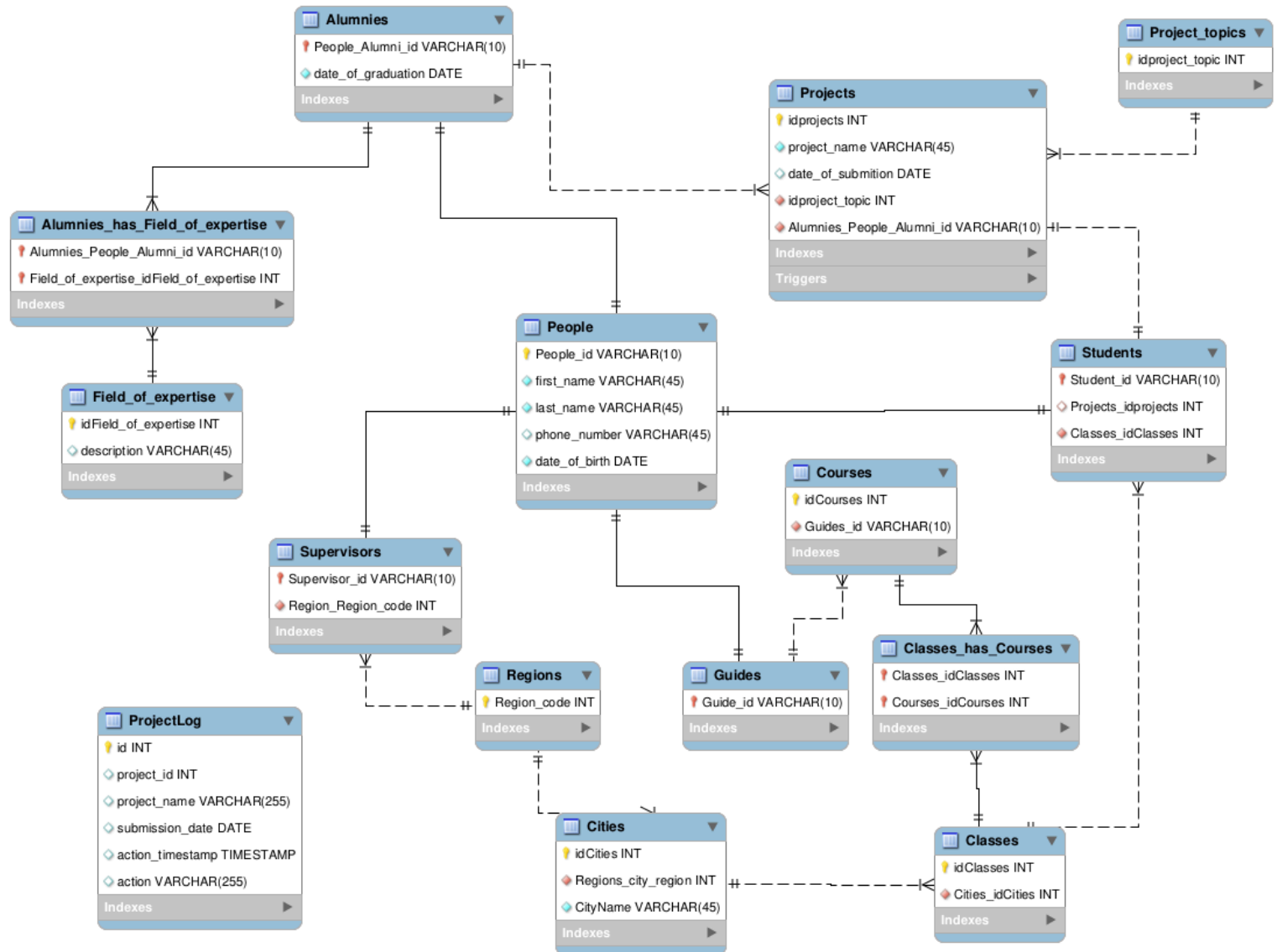
רשימת משתמשים

- מנהלי מערכות המידע של תוכנית מגשימים - גישה לכל המידע
- רכזים אזוריים - גישה למידע מוגבלת לאזור
- רכזי מוקדים - גישה למידע מוגבלת למוקד
- מדריכי פרוייקטים - גישה למידע מוגבלת לכיתה
- בוגרים חונכים - גישה למידע מוגבלת לפרויקט של החניכים שהם חונכים
- חניכים - גישה לרישום מידע וקריאת מידע אודות הפרוייקט שלהם

תרשים ERD מנורמל



תרשים DSD



פירוט אילוצי מערכת

אילוצי המערכת בבסיס הנתונים הם כדלקמן:

****אילוצ מפתח ראשי (Primary Key Constraint)**:** האילוצ הזה מבטיח שכל שורה בטבלה יכולה להיות מזוהה באופן ייחודי. בסכמה שלי, מפתחות ראשיים מוגדרים עבור טבלאות כמו `People`, `Alumnies`, `Projects`, `Project_topics`, `Field_of_expertise`, `Guides`, `Regions`, `Cities`, `Classes`, `Students`, `Courses`, `Supervisors`, `Classes_has_Courses` ו- `Alumnies_has_Field_of_expertise`.

****אילוצ מפתח זר (Foreign Key Constraint)**:** האילוצ הזה מקים קשר בין שתי טבלאות ושומר על אינטגריות רפרנציאלית. הוא מבטיח שערכים בעמודה (או עמודות) של טבלה אחת תואמים לערכים בעמודה המפתח הראשי או הערך הייחודי של טבלה אחרת. בסכמה שלי, מפתחות זרים מוגדרים עבור טבלאות כמו `Projects`, `Alumnies`, `Cities`, `Students`, `Guides`, `Supervisors`, `Classes_has_Courses` ו- `Alumnies_has_Field_of_expertise`.

****אילוצ לא ריק (Not Null Constraint)**:** האילוצ הזה מבטיח שעמודה לא יכולה להכיל ערכים ריקים. בסכמה שלי, אילוצי `NOT NULL` מוגדרים עבור עמודות כמו `People_id`, `first_name`, `last_name`, `date_of_birth`, `date_of_graduation`, `project_name`, `idproject_topic`, `Alumnies_People_Alumni_id`, `Regions_city_region`, `CityName`, `idClasses`, `Classes_idClasses`, `Classes_has_Courses`, `idCourses`, `Guides_id`, `Supervisor_id`, `Region_Region_code`, `idField_of_expertise` ו- `description`.

על ידי יישום ואכיפת אלו האילוצים, בסיס הנתונים שלי מבטיח אינטגרציה ויישום נכון של הנתונים, מונע נתונים לא תקינים או לא עקביים מלהיכנס או להתעדכן.

פירוט טבלאות במסד הנתונים

Projects
idprojects INT
project_name VARCHAR(45)
date_of_submission DATE
idproject_topic INT
Alumnies_People_Alumni_id VARCHAR(10)
Indexes

מבנה הטבלאות
ודוגמאות לנתונים

טבלאת Project
מפתח ראשי - idprojects תעודת זהות של אדם
שם פרטייקט
תאריך הגשה
נושא פרטייקט
בוגר מנחה

#	idprojects	project_name	date_of_submission	idproject_topic	Alumnies_People_Alumni_id
1	1	Compiler - quartz	2023-01-20	1	000001
2	2	TOR - onion	2023-02-15	2	000002
3	3	Operating system - atOS	2023-03-10	3	000003
4	198	test2	2024-03-05	1	000001
5	199	test	2024-03-05	1	000001

טבלאת Alumni
מפתח ראשי - תעודת זהות
תאריך סיום לימודים בתוכנית

Alumnies
People_Alumni_id VARCHAR(10)
date_of_graduation DATE
Indexes

#	People_Alumni_id	date_of_graduation
1	000001	2010-06-30
2	000002	2012-07-15
3	000003	2011-05-20
*	NULL	NULL

טבלאת Alumnies_has_Field_of_expertise
מפתח ראשי - תעודת זהות בוגר
מפתח ראשי - מספר מזהה של תחום התמחות

Alumnies_has_Field_of_expertise
Alumnies_People_Alumni_id VARCHAR(10)
Field_of_expertise_idField_of_expertise INT
Indexes

#	Alumnies_People_Alumni_id	Field_of_expertise_idField_of_expertise
1	000001	1
2	000002	2
3	000003	3
*	NULL	NULL

Field_of_expertise
idField_of_expertise INT
description VARCHAR(45)
Indexes

טבלאת Filed_of_expertise
מפתח ראשי - מספר זיהוי של התמחות
תיאור התמחות

#	idField_of_expertise	description
1	1	networking
2	2	low-level
3	3	web
*	NULL	NULL

People
People_id VARCHAR(10)
first_name VARCHAR(45)
last_name VARCHAR(45)
phone_number VARCHAR(45)
date_of_birth DATE
Indexes

#	People_id	first_name	last_name	phone_number	date_of_birth
1	000001	John	Doe	1234567890	1990-05-15
2	000002	Jane	Smith	0987654321	1992-08-20
3	000003	Michael	Johnson	9876543210	1988-03-10
4	000004	John	exit	1234567890	1990-05-15
5	000005	Tim	Mcnil	0987654321	1992-08-20
6	000006	Ben	Dover	9876543210	1988-03-10
7	000007	Nig	Mansen	1234567890	1990-05-15
8	000008	Crolin	Tory	0987654321	1992-08-20
9	000009	bobson	jake	9876543210	1988-03-10
10	000010	bon	jole	9876876542	1988-02-12
*	NULL	NULL	NULL	NULL	NULL

טבלאת People
מפתח ראשי -
תעודת זהות
שם פרטי
שם משפחה
מספר טלפון
תאריך לידה

טבלאת Project_topics
מפתח ראשי - מספר מזהה של נושא פרוייקט

Project_topics
idproject_topic INT
Indexes

#	idproject_topic
1	1
2	2
3	3
*	NULL

טבלאת Guides
מפתח ראשי - תעודת זהות מדריך

Guides
Guide_id VARCHAR(10)
Indexes

#	Guide_id
1	000001
2	000002
3	000003
*	NULL

טבלאת Courses

מפתח ראשי - קוד זיהוי קורס
תעודת זהות של
מדריך

Courses
idCourses INT
Guides_id VARCHAR(10)
Indexes

#	idCourses	Guides_id
1	1	000001
2	2	000002
3	3	000003
*	NULL	NULL

טבלאת Classes_has_Courses

מפתח ראשי - קוד זיהוי כיתה
מפתח ראשי - קוד זיהוי קורס

Classes_has_Courses
Classes_idClasses INT
Courses_idCourses INT
Indexes

#	Classes_idClasses	Courses_idCourse
1	101	1
2	102	2
3	103	3
*	NULL	NULL

טבלאת Classes

מפתח ראשי - קוד כיתה
קוד עיר

Classes
idClasses INT
Cities_idCities INT
Indexes

#	idClasses	Cities_idCities
1	101	1
2	102	2
3	103	3
*	NULL	NULL

טבלאת Supervisors

מפתח ראשי - תעודת זיהוי רכז
קוד אזור

Supervisors
Supervisor_id VARCHAR(10)
Region_Region_code INT
Indexes

#	Supervisor_id	Region_Region_code
1	000007	100
2	000008	200
3	000009	300
*	NULL	NULL

טבלאת Cities

מפתח ראשי - קוד עיר

קוד אזור

שם עיר

Cities	
idCities INT	
Regions_city_region INT	
CityName VARCHAR(45)	
Indexes	

#	idCities	Regions_city_region	CityName
1	1	100	Beer-Sheva
2	2	200	Tel-aviv
3	3	300	Karmiel
*	NULL	NULL	NULL

טבלאת Regions

מפתח ראשי - קוד אזור

Regions	
Region_code INT	
Indexes	

#	Region_code
1	100
2	200
3	300
*	NULL

טבלאת Students

מפתח ראשי - תעודת זהות תלמיד

קוד פרוייקט

קוד כיתה

Students	
Student_id VARCHAR(10)	
Projects_idprojects INT	
Classes_idClasses INT	
Indexes	

#	Student_id	Projects_idprojec	Classes_idClasse
1	000004	1	101
2	000005	2	102
3	000006	3	103
4	000010	NULL	101
*	NULL	NULL	NULL

רשימת השאלות ופירוט ייעוד

```
10 -- all Alumnies that helped with a project
11 • SELECT p.project_name, CONCAT(ppl.first_name, " ", ppl.last_name) AS alumniName
12 FROM Projects p
13 INNER JOIN People ppl ON p.Alumnies_People_Alumni_id = ppl.People_id
14 WHERE EXISTS (
15     SELECT 1
16     FROM Alumnies a
17     WHERE a.People_Alumni_id = ppl.People_id
18 );
```

project_name	alumniName
Compiler - quartz	John Doe
test2	John Doe
test	John Doe
TOR - onion	Jane Smith
Operating system - atOS	Michael Johnson

1

שאלתה למציאת כל הבוגרים שעזרו בפרוייקט (אחד או יותר) ואת שמות הפרוייקטים שהם עזרו בהם

שימוש: על מנת לתת מידע לרכזי התוכנית על בוגרים בולטים או לתת מידע לבוגרים שרוצים לדעת בכמה פרוייקטים הם עזרו

```
19
20 -- earliest project latest project and total number of projects
21 • SELECT MAX(date_of_submission) AS latest_project_submitted,
22     MIN(date_of_submission) AS earlies_project_submitted,
23     COUNT(*) AS total_projects
24 FROM Projects;
25
```

#	latest_project_submitt	earlies_project_submitt	total_project
1	2024-03-05	2023-01-20	5

2

שאלתה למציאת הפרוייקט הראשון שהוגש, הפרוייקט האחרון שהוגש, ומספר הפרוייקטים הקיימים במאגר

שימוש: נתונים עבור רכזי פרוייקטים ועבור מנהלי התוכנית

```

26 -- get total of projects from a topic with at least 1 project
27 • SELECT idproject_topic ,COUNT(*) AS total_projects
28 FROM Projects p
29 GROUP BY idproject_topic
30 HAVING COUNT(*) > 0;
31

```

3

שאלתה לצפייה בסכום הפרויקטים
השייכים לנושא מסוים בו קיים לפחות
פרויקט אחד

#	idproject_topic	total_projects
1	1	3
2	2	1
3	3	1

שימוש: מידע המשמש רכזים ומדריכים
ובוגרים לדעת מהם הנושאים הכי
מבוקשים שצריך בהם הכי הרבה
בוגרים שיעזרו/שיעורי העשרה

```

32 -- get supervisors with regions and phone number
33 • SELECT
34     p.first_name AS supervisor_first_name,
35     p.last_name AS supervisor_last_name,
36     sup.Supervisor_id,
37     r.Region_code,
38     p.phone_number
39 FROM
40     Supervisors sup
41 JOIN
42     People p ON sup.Supervisor_id = p.People_id
43 JOIN
44     Regions r ON sup.Region_Region_code = r.Region_code
45 order by supervisor_first_name;
46

```

4

שאלתה לקבלת מידע על כל הרכזים
האזוריים וקוד האזור שעליו הם
אחראיים

שימוש: מידע למדריכים מתחת לאזור
מסויים כדי לדעת מי מנהל האזור
שלהם ובנוסף מידע למנהלי התוכנית
ליצירת קשר מהיר

הערה: ניתן לפרק שאלתה זו
לפרוצדורה מאוחסנת אשר מקבלת
קוד אזור ומוציאה מידע על רכז
האזור הספציפי

#	supervisor_first_name	supervisor_last_name	Supervisor_id	Region_code	phone_number
1	bobson	jake	000009	300	9876543210
2	Crolin	Tory	000008	200	0987654321
3	Nig	Mansen	000007	100	1234567890

5

```

48 -- get all students with no project
49 • SELECT s.Student_id, concat(p.first_name, " ", p.last_name) as studentName
50 FROM Students s
51 LEFT JOIN Projects pr ON s.Projects_idprojects = pr.idprojects
52 join People p on p.people_id = s.student_id
53 WHERE pr.idprojects IS NULL;
54

```

שאלתה למציאת כל החניכים ללא
פרויקט

שימוש: משמש מדריכים ורכזים על
מנת לעזור לתלמידים לבחור
פרויקט או לבדוק כמה חניכים
מהמחזור ללא פרויקט

Student_id	studentName
000010	bon jole

6

```

55  -- count classes in each city
56  • SELECT c.CityName, COUNT(cl.idClasses) AS class_count
57  FROM Cities c
58  LEFT JOIN Classes cl ON c.idCities = cl.Cities_idCities
59  GROUP BY c.CityName;
60

```

שאלתה לספירת כמות הכיתות בכל עיר

שימוש: מידע למנהלי התוכנית ולמנהלי כוח אדם כדי לדעת מה כמות המדריכים ורכזי הכיתות הנדרשת

CityName	class_count
Beer-Sheva	2
Tel-aviv	2
Karmiel	1

7

פרוצדורה מאוחסנת המקבלת קוד אזור ומחזירה את המידע על רכז האזור

שימוש: מידע תמציתי וספציפי יותר מאשר השאלתה הכללית, מיועד לתלמידים או בוגרים שרוצים ליצור קשר עם רכז האזור שלהם

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `getRegionSupervisor`(in requestedRegionCode int)
2 BEGIN
3 SELECT
4     p.first_name AS supervisor_first_name,
5     p.last_name AS supervisor_last_name,
6     sup.Supervisor_id,
7     r.Region_code,
8     p.phone_number
9 FROM
10     Supervisors sup
11 JOIN
12     People p ON sup.Supervisor_id = p.People_id
13 JOIN
14     Regions r ON sup.Region_Region_code = r.Region_code
15
16 where requestedRegionCode = r.Region_code;
17
18 END

```

8

פרוצדורה מאוחסנת שמקבלת תעודת זהות של מדריך ומחזירה רשימת חניכים של אותו מדריך

שימוש: מידע על חניכי המדריך עבור המדריך

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `getAllGuidesStudents`(in requestedGuide varchar(10))
2 BEGIN
3 -- get all students of specific guide
4 SELECT
5     co.idCourses,
6     co.Guides_id,
7     concat(g.first_name, " ", g.last_name) as guideName,
8     concat(s.first_name, " ", s.last_name) as studentName,
9     stu.student_id
10 FROM
11     Courses co
12 JOIN
13     Students stu ON co.idCourses = stu.Projects_idprojects
14 JOIN
15     People s ON stu.Student_id = s.People_id
16 join
17     People g on co.Guides_id = g.people_id
18 WHERE
19     co.Guides_id = requestedGuide;
20 END

```

9

```
CREATE DEFINER='root'@'localhost' PROCEDURE `getAllStudentsUnderSupervisor`(in requestedSupervisor varchar(10))
BEGIN
-- get info on all students that are from a specific supervisor region
SELECT
concat(p.first_name, " ", p.last_name) as studentName,
s.Student_id,
c.idClasses,
ci.idCities,
r.Region_code,
ci.CityName
FROM
Students s
JOIN
Classes c ON s.Classes_idClasses = c.idClasses
JOIN
Cities ci ON c.Cities_idCities = ci.idCities
JOIN
Regions r ON ci.Regions_city_region = r.Region_code
JOIN
Supervisors sup ON r.Region_code = sup.Region_Region_code
join
People p on s.student_id = p.people_id
WHERE
sup.Supervisor_id = requestedSupervisor;
END
```

פרוצדורה מאוחסנת
המקבלת תעודת זהות של
רכז אזור ומחזירה את כל
החניכים שהוא אחראי
עליהם

שימוש: בדיקת רישומים
לימי שיא(לא במערכת)
בדיקת מספרי חניכים
ומידע עליהם עבור רכזי
האזור

10

```
CREATE DEFINER='root'@'localhost' PROCEDURE `getProjectsBetweenDates`(in startDate date, in endDate date)
BEGIN
-- all projects between these dates or null dates
SELECT *
FROM Projects p
WHERE date_of_submission BETWEEN startDate AND endDate;
END
```

פרוצדורה מאוחסנת על
מנת לקבל את כל
הפרוייקטים בין תאריך
התחלה למועד סיום
המתקבלים כקלט

שימוש: בדיקת תקופות
עם הכי הרבה
פרוייקטים\הכי מעט פרוייקטים
בעבור מנהלי התוכנית והרכזים

11

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `getStudentsOfCity`(in idOfCity int)
2 BEGIN
3 -- id of all students from certain city
4 SELECT Student_id, concat(p.first_name, " ", p.last_name) as studentName, cityName
5 FROM Students s
6 join People p on p.people_id = s.student_id
7 join Cities on idCities = idOfCity;
8 END
```

פרוצדורה מאוחסנת
למציאת מידע על
תלמידים מעיר מסויימת
על פי קוד עיר כקלט
שימוש: מידע על
תלמידים לרכזי אזור
ומדריכים בעיר מסויימת

12

```
• CREATE DEFINER='root'@'localhost' PROCEDURE `getStudentSubmittedAfterAlumniGrad`(in alumniGradDate date)
BEGIN
    -- Let's say we want to retrieve all students who have submitted projects
    -- that match the field of expertise of any alumni who graduated after a certain date.

    SELECT DISTINCT s.*
    FROM Students s
    JOIN Projects p ON s.Projects_idprojects = p.idprojects
    JOIN Alumnies a ON p.Alumnies_People_Alumni_id = a.People_Alumni_id
    JOIN Alumnies_has_Field_of_expertise af ON a.People_Alumni_id = af.Alumnies_People_Alumni_id
    WHERE a.date_of_graduation > alumniGradDate
    AND p.idprojects IN (
        SELECT p.idprojects
        FROM Projects p
        JOIN Alumnies a ON p.Alumnies_People_Alumni_id = a.People_Alumni_id
        JOIN Alumnies_has_Field_of_expertise af ON a.People_Alumni_id = af.Alumnies_People_Alumni_id
        WHERE af.Field_of_expertise_idField_of_expertise =
            (SELECT Field_of_expertise_idField_of_expertise
             FROM Alumnies_has_Field_of_expertise
             WHERE Alumnies_People_Alumni_id = a.People_Alumni_id));
    END
```

פרוצדורה מאוחסנת
למציאת כל התלמידים
שהגישו עבודה בתחום
ההתמחות של בוגר שסיים
ללמוד אחרי תאריך הנתון
כקלט

שימוש: עבור חניכים אשר
הגישו פרוייקט ורוצים
לבדוק כמה בוגרים התמחו
בתחום של הפרוייקט
שלהם לאורך התקופה
שאחרי ההגשה

13

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `switchStudentClass`(in newClass int, in requestedStudent int)
2 BEGIN
3 -- update query to move student to different class
4 UPDATE Students
5 SET Classes_idClasses = newClass -- new class
6 WHERE Student_id = requestedStudent; -- student id
7 END
```

פרוצדורה מאוחסנת
לשינוי כיתה של
תלמיד

שימוש: למקרה
שתלמיד עבר כיתה
באותה עיר/עבר עיר וצריך לשנות את הכיתה שלו

14

מזניק עבור טבלאת פרויקטים שמכניס נתונים LOG לתוך טבלאת עזר בשם ProjectLog

```
1 • CREATE DEFINER='root'@'localhost' TRIGGER `project_submission_trigger` AFTER INSERT ON `Projects` FOR EACH ROW BEGIN
2 -- Action: Insert into ProjectLog table
3 INSERT INTO ProjectLog (project_id, project_name, submission_date, action_timestamp, action)
4 VALUES (NEW.idprojects, NEW.project_name, NEW.date_of_submission, NOW(), 'New project submitted');
5 END
```