

פרויקט מסכם במערכות מידע ותכנון מערכות בסיסי נתונים

מגישים:

טל אילון – ת.ז. 311117428

עמיחי כלב – ת.ז. 308469675

אביהו מנחם – ת.ז. 204179832

1. חלק א': הנחות ותהליך תכנון בסיס הנתונים

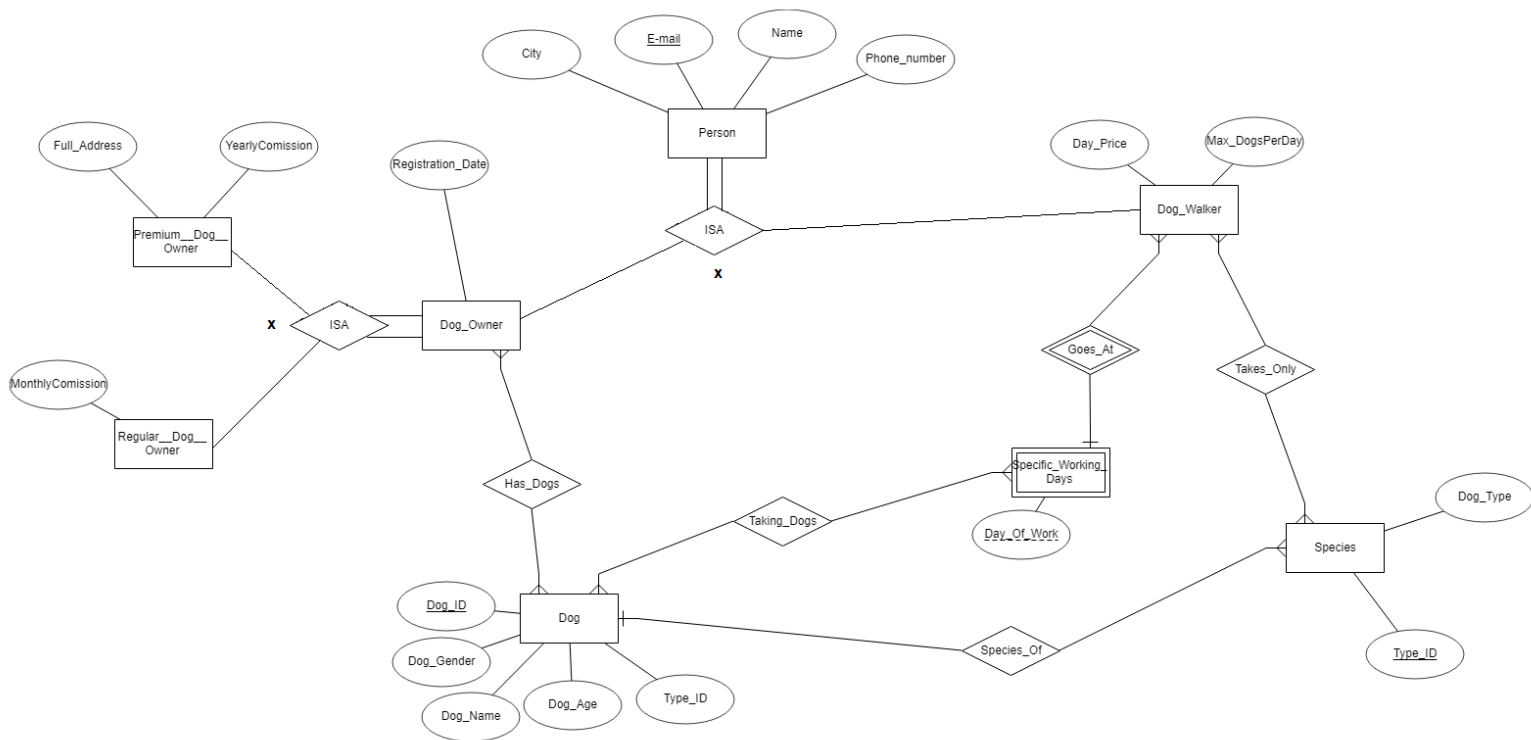
את מידול הפרויקט התחלנו בכך שהבחנו 33 ישויות עיקריות: בעל כלב רגיל, בעל כלב פרימיום ומוליך כלבים. כיוון שלבעל כלב ולמוליך כלבים יש תכונות משותפות, בחרנו לממש יישות על משותפת שתיקרא "אדם" והיא מורשה את תכונותיה לבעל כלב ולמוליך כלבים. גם לבעל כלב יש תכונות משותפות, ולכן גם הוא מוריש תכונות לסוג בעל כלב רגיל ולסוג בעל כלב פרימיום.

בהמשך, בבניית בסיס הנתונים נלקחו בחשבון ההנחות הבאות:

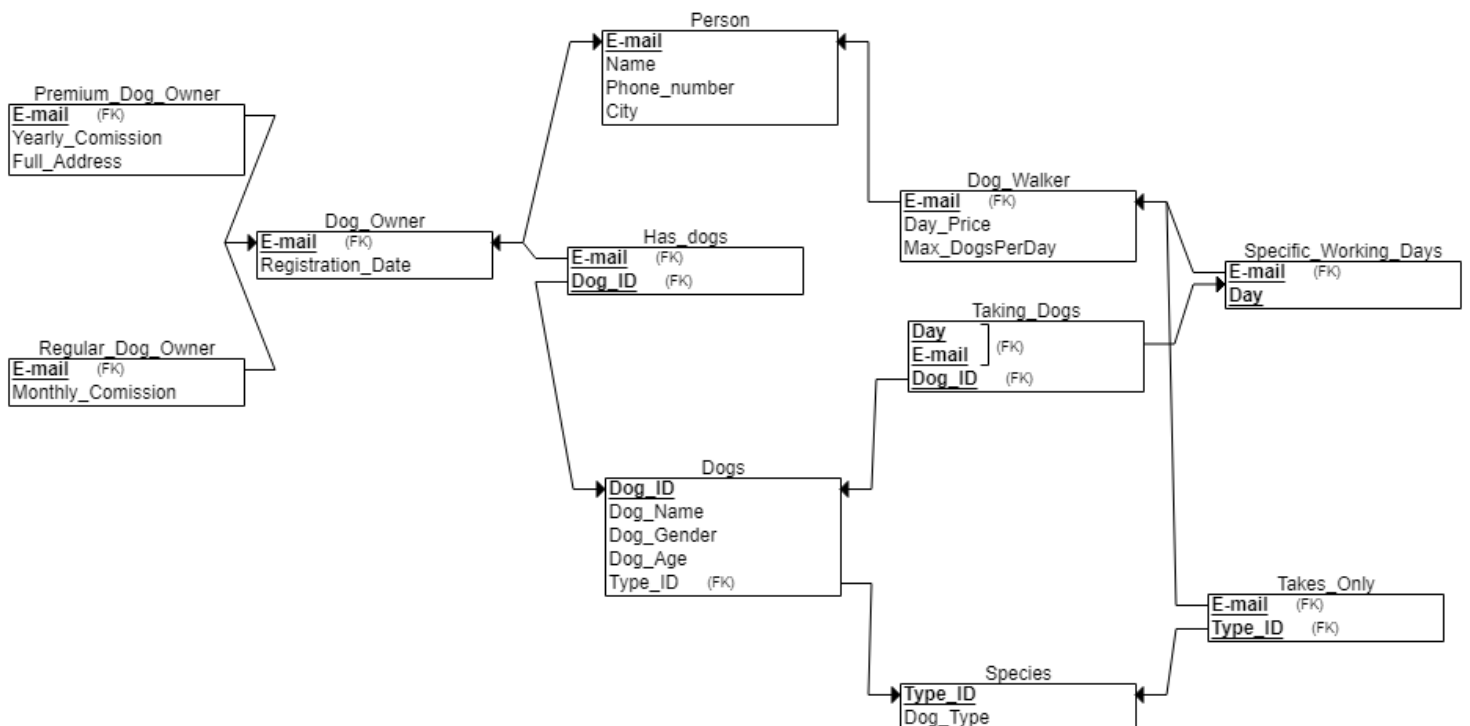
1. המשתמש שנרשם למערכת יכול להיות או בעל כלב או מוליך כלבים אבל לא שניהם.
2. בעל הכלב שנרשם למערכת יכול להיות או רגיל או פרימיום אבל לא שניהם.
3. כלב יכול ללכת לטיול פעם אחת ביום. מוליך כלבים יכול לקחת כמה כלבים באותו יום עד למקסימום אותו הגדיר בעת ההרשמה, בטיול יחיד מרוכז.
4. כל טיול אותו מבצע מוליך כלבים נחשב לטיול יומי יחיד עבור כלב מסויים.
5. התעריף אותו הצהיר מוליך הכלבים בעת ההרשמה למערכת רלוונטי לטיול יומי. כאשר מוליך כלבים מוציא יותר מכלב אחד ביום התעריף יוכפל במספר הכלבים אותו לוקח בטיול.
6. כאשר בעל כלב נרשם למערכת, הן אם הוא רגיל והן אם הוא פרימיום, תאריך ההרשמה שלו נרשם במערכת.
7. העמלה החודשית והעמלה השנתית שמורות במערכת ולא נתונות לשינוי על ידי בעל הכלב.
8. העמלה החודשית רלוונטית לבעל הכלב הרגיל, והעמלה השנתית רלוונטית לבעל הכלב הפרימיום.
9. מוליך כלבים לא יכול לשנות את פרטיו לאחר ההרשמה, לרבות את סוגי הכלבים שהוא הצהיר שהוא יכול לקחת בעת ההרשמה שלו לאתר, וימי העבודה שהוא יכול לעבוד בהם.
10. המערכת תומכת ברשימה סגורה של סוגי כלבים כאשר לא ניתן לשלב בין הסוגים השונים.
11. המערכת תומכת ברשימה סגורה של ערים בהן ניתן השירות.
12. התעריף של מוליך הכלבים קבוע ולא ניתן לשינוי לאחר הרשמה.
13. לכל כלב מס' יחודי במערכת.
14. לבעל כלב יכולים להיות כמה כלבים, וכלב יכול להיות שייך ליותר מבעל כלב אחד*.

*הערה: במהלך תכנון בסיס הנתונים התכוונו לאפשר לכלב להיות שייך לכמה משתמשים, וזאת על מנת לאפשר לכלב שלו יש כמה בעלים (שייך למשפחה, שותפים וכו') להיות רשום במערכת תחת כמה בעלי כלבים. לפיכך התייחסנו לקשר בין כלב לבעל כלב כקשר רבים לרבים. עם זאת, במהלך העבודה על הפרויקט ראינו כי בזמן שעמד לרשותנו לא הספקנו לממש את הפונקציונליות הנדרשת במלואה. במידה שהיינו מספיקים לממש את הפונקציונליות, חשוב לציין כי היינו מוסיפים בהתאם הנחה שקיים תיאום בין בעלי הכלבים שלהם בעלות משותפת על אותו כלב, וזאת על מנת למנוע תשלום כפול על הטיולים.

2. חלק ב': תרשים ERD



3. חלק ג': סכימה טבלאית



*פקודות ה-DDL מצורפת גם כקובץ SQL.

```
CREATE schema DogWalkers;
```

```
USE DogWalkers;
```

```
CREATE TABLE Person
```

```
(
    Email VARCHAR(100) NOT NULL,
    P_Name VARCHAR(100) NOT NULL,
    Phone_number VARCHAR(10) NOT NULL,
    City VARCHAR(100) NOT NULL,
    PRIMARY KEY (Email)
);
```

```
CREATE TABLE Species
```

```
(
    Type_ID INT NOT NULL,
    Dog_Type VARCHAR(100) NOT NULL,
    Average_Height INT NOT NULL,
    Average_Weight INT NOT NULL,
    PRIMARY KEY (Type_ID)
);
```

```
CREATE TABLE Dogs
```

```
(
    Dog_ID INT NOT NULL auto_increment,
    Dog_Name VARCHAR(100) NOT NULL,
    Dog_Gender VARCHAR(100) NOT NULL,
    Dog_Age INT NOT NULL,
    Type_ID INT NOT NULL,
    PRIMARY KEY (Dog_ID),
    FOREIGN KEY (Type_ID) REFERENCES Species(Type_ID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE Dog_Owner
```

```
(
    Email VARCHAR(100) NOT NULL,
    Registration_Date DATE NOT NULL,
    PRIMARY KEY (Email),
    FOREIGN KEY (Email) REFERENCES Person(Email) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE Has_Dogs
```

```
(
    Email VARCHAR(100) NOT NULL,
    Dog_ID INT NOT NULL,
    PRIMARY KEY (Email,Dog_ID),
    FOREIGN KEY (Dog_ID) REFERENCES Dogs(Dog_ID) ON DELETE CASCADE ON UPDATE CASCADE,
);
```

```

FOREIGN KEY (Email) REFERENCES Dog_Owner(Email) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Regular_Dog_Owner
(
    Email VARCHAR(100) NOT NULL,
    Monthly_Comission INT NOT NULL,
    PRIMARY KEY (Email),
    FOREIGN KEY (Email) REFERENCES Dog_Owner(Email) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Premium_Dog_Owner
(
    Email VARCHAR(100) NOT NULL,
    Yearly_Comission INT NOT NULL,
    Full_Address VARCHAR(100) NOT NULL,
    PRIMARY KEY (Email),
    FOREIGN KEY (Email) REFERENCES Dog_Owner(Email) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Dog_Walker
(
    Email VARCHAR(100) NOT NULL,
    Day_Price INT NOT NULL,
    Max_DogsPerDay INT NOT NULL,
    PRIMARY KEY (Email),
    FOREIGN KEY (Email) REFERENCES Person(Email) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Specific_Working_Days
(
    Email VARCHAR(100) NOT NULL,
    Day_of_Work INT NOT NULL,
    PRIMARY KEY (Email, Day_of_Work),
    FOREIGN KEY (Email) REFERENCES Dog_Walker(Email) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Takes_Only
(
    Email VARCHAR(100) NOT NULL,
    Type_ID INT NOT NULL,
    PRIMARY KEY (Email, Type_ID),
    FOREIGN KEY (Email) REFERENCES Dog_Walker(Email) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Type_ID) REFERENCES Species(Type_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Taking_Dogs
(
    Email VARCHAR(100) NOT NULL,
    Day_of_Work INT NOT NULL,
    Dog_ID INT NOT NULL,

```

```
PRIMARY KEY (Email, Day_of_Work, Dog_ID),
FOREIGN KEY (Email, Day_of_Work) REFERENCES Specific_Working_Days(Email, Day_of_Work) ON DELETE
CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Dog_ID) REFERENCES Dogs(Dog_ID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

5. חלק ה': מענה על השאלות

*כלל השאלות מצורפות גם בקבצי SQL.

שאלת ראשונה

```
SELECT Person.p_name as Dog_Owner_Name, COUNT(DISTINCT Day_of_Work)
AS Number_Of_Days, SUM(day_price) AS Total_Payment

FROM Has_Dogs AS hd LEFT JOIN Person on Person.email=hd.email
LEFT JOIN Taking_Dogs AS td ON hd.Dog_ID = td.Dog_ID
LEFT JOIN Dog_Walker ON Dog_Walker.email=td.email

WHERE Person.city = "Tel Aviv" OR Person.city = "Haifa"

GROUP BY Dog_Owner_Name, Person.email;
```

פלט לדוגמה:

	Dog_Owner_Name	Number_Of_Days	Total_Payment
▶	Beyonce Knowles	0	NULL
	Dwayne Johnson	7	515
	Nicolas Cage	3	220
	Vin Diesel	0	NULL
	zion	1	60

הסבר:

התקבל פלט ובו כל בעלי הכלבים מתל אביב וחיפה, גם כאלו שאין להם כלבים שרשומים לטיוולים ולכן עבור אותם בעלי כלבים, מס' הימים יהיה 0 והתשלום יהיה NULL, כלומר 0.

שאלת שנייה

```
SELECT Dog_Type, AVG(Day_Price) as Average_Day_Price,
COUNT(email) as Num_Of_Active_Dogwalkers

FROM (SELECT DISTINCT DW.email, S.Dog_Type, DW.Day_Price
FROM Taking_Dogs as TD LEFT JOIN Dogs ON TD.Dog_ID =
Dogs.Dog_ID
LEFT JOIN Species as S ON Dogs.Type_ID = S.Type_ID
LEFT JOIN Dog_Walker as DW ON TD.email = DW.email) as
Data_Table

GROUP BY Dog_Type;
```

פלט לדוגמה:

	Dog_Type	Average_Day_Price	Num_Of_Active_Dogwalkers
▶	Chihuahua	60.0000	1
	Dalmatian	55.0000	2
	Golden Retriever	20.0000	1
	Labrador	54.5000	10
	Pomeranian	75.0000	2

הסבר :

כל כלב שנלקח לטיול נקלח על ידי מוליך כלבים שהוא בהכרח פעיל – מכך נגזור את סוג הכלב, נבצע ממוצע של המחירים ליום של מוליכי הכלבים ונסכום את מס' המוליכים בהתאם לסוג הכלב באופן מבחין, כלומר כל מוליך שלקח כלב מסוג מסויים לפחות פעם אחת במהלך השבוע ייספר פעם אחת.

שאלתה שלישית

```
SELECT dog_name
FROM (SELECT Dogs.dog_name ,COUNT(*) as max_trip
      FROM Taking_Dogs as tk JOIN Dogs on tk.dog_id=Dogs.dog_id
      GROUP BY Dogs.dog_name
      HAVING max_trip<0.5*(SELECT max(max_trip)
                           FROM (SELECT tk.dog_id, count(*) as
                                max_trip
                                FROM Taking_Dogs as tk
                                JOIN Dogs on
                                     tk.dog_id=Dogs.dog_id
                                GROUP BY dog_id) as d)) as d;
```

פלט לדוגמה :

	dog_name
▶	israel
	Moti
	Prince
	Rexy
	Shlomo
	Shoko
	Sokolov
	Ugy
	Yafa

הסבר :

נמצא תחילה את הכלב שיוצא הכי הרבה פעמים לטיול, וממנו נמצא את הכלבים שיוצאים לטיול בפחות מ50% ממספר הטיולים הגבוה ביותר שנמצא, ונחזיר רשימה של שמות הכלבים האלו.

שאלתה רביעית

```
SELECT DISTINCT p_name,dog_name
FROM ( SELECT dw.email, p_name ,min(d.dog_age) as MIN_AGE
      FROM ((Dog_Walker as dw JOIN Person as p ON
            dw.email=p.email) JOIN Taking_Dogs AS td ON
            dw.email=td.email)
            JOIN Dogs AS d ON td.dog_id=d.dog_id
      GROUP BY dw.email, p_name
      HAVING min(d.dog_age)) as t
      JOIN Taking_Dogs AS td ON t.email=td.email
      JOIN Dogs AS d ON td.dog_id=d.dog_id

WHERE min_age=dog_age
ORDER BY p_name ASC;
```

פלט לדוגמה :

	p_name	dog_name
▶	Abraham Lincoln	Sokolov
	Adir Miler	Prince
	Angelina Jolie	Rexy
	Barack Obama	Yafa
	Ben Affleck	Shmuel
	Brad Pitt	Shoko
	Jennifer Lawrence	Shmuel
	Johnny Depp	Ugy
	Morgan Freeman	Shoko
	Nicole Kidman	Shmuel
	Shahar Hason	Shmuel
	Tom Cruise	Ugy
	אביהו מנחם	Moti

הסבר :

תחילה נמצא את גיל הכלב הכי צעיר עבור כל מוליך כלבים שרשום במערכת, ולאחר מכן נחזיר את שם הכלב הצעיר ביותר בצירוף שם המוליך. במידה ויש כמה כלבים שגילם הצעיר ביותר, יוצגו כל הצעירים ביותר תחת אותו מוליך.