

טבלאות

את המידע לקחנו מאתר Kaggle,

הטבלה הראשונה והעיקרית ממנה לקחנו מידע נקראת [k-hotel-reviews-data-in-europe515](#), ומכילה כחצי מיליון רשומות, צילום מסך של הטבלה המקורית :

lng	lat	days_sinci	Tags	Reviewer_Total_Nun	Review_Ti	Positive_F	Total_Nun	Review_Ti	Negative_	Reviewer_	Hotel_Nan	Average_	Review_Da	Additional	Hotel_Add
4.915968	52.36058	0 days	['Leisure trip', 'Couple', 'Duplex Double Room', 'Sta	2.9	7	11	Only the p	1403	397	I am so ar	Russia	Hotel Aren	7.7	08/03/2017	194 s Graves
4.915968	52.36058	0 days	['Leisure trip', 'Couple', 'Duplex Double Room', 'Sta	7.5	7	105	No real cc	1403	0	No Negati	Ireland	Hotel Aren	7.7	08/03/2017	194 s Graves
4.915968	52.36058	3 days	['Leisure trip', 'Family with young children', 'Duplex Di	7.1	9	21	Location \	1403	42	Rooms ar	Australia	Hotel Aren	7.7	7/31/2017	194 s Graves
4.915968	52.36058	3 days	['Leisure trip', 'Solo traveler', 'Duplex Double Room '	3.8	1	26	Great loc	1403	210	My room	United Kir	Hotel Aren	7.7	7/31/2017	194 s Graves

הטבלה הייתה מבולגנת מאוד ולאחר סידור קפדני של המידע יצרנו ממנה 6 טבלאות מסודרות כמפורט בהמשך.

כפי שניתן לראות, בטבלה לעיל לא היו שמות ומשתמש ולכן היינו צריכות לחשוב כיצד להוסיף את השמות, כדי לעשות זאת איחדנו טבלאות נוספות מהן לקחנו שמות משתמש + מין (זכר או נקבה) : [Us-baby-names](#), [French-baby-names](#) שמות משתמש הם primary key ולכן הורדנו כפילויות. לצערנו גם לאחר תהליך זה לא קיבלנו מספיק שמות משתמש ולכן הוספנו באמצעות script בפייתון מספרים שונים בצמוד לשם לדוגמא : dan1, dan2, dan3 עד שהיו לנו מספיק שמות (מעל חצי מיליון). כמו כן הוספנו גם רשומה של סיסמה (מכיוון שסיסמה היא לא primary key הגדרנו בשלב זה סיסמה אחידה לכולם).

על מנת לקבל את הסכמה שלנו ביצענו חלוקה וסינון לטבלאות, בין היתר באמצעות סקריפטים בPython.

לסכמה שלנו קוראים hotels_reviews והיא מכילה 6 טבלאות, סידרנו את המידע באופן שבו הסכמה תתאים לאפליקציה, דאגנו לשמור על סדר, טיפוסים מתאמים וכמובן נמנענו לחלוטין משכפול מידע. בכל טבלה יש מפתח שמאפשר לזהות את הנתונים בקלות.

1. Countries

טבלה המכילה שמות מדינות, לצד ערכים מספריים (מפתחות) אליהם הם ממופים ועמודה האם המדינה נמצאת באירופה או לא (המערכת מיועדת לביקורות על בתי מלון באירופה בלבד).

תהליך יצירת הטבלה: מצאנו טבלה בה מופיעים מדינות מרחבי העולם וערך בוליאני האם הן נמצאות באירופה או לא. (בטבלה המקורית לא הייתה עמודה של מדינה)

מכילה 3 עמודות:

Key_User_Country - ערך מספרי, מפתח למדינה, עמודה זו משמשת כ primary key

Country - מחרוזת, שם המדינה

In_Europe – ערך בוליאני, האם המדינה נמצאת באירופה.

2. Hotels

טבלה המכילה את key של בית המלון(ערך מספרי), שמות בתי-המלון, כתובת ומפתח של מדינה בה נמצא בית המלון.

תהליך יצירת הטבלה : לקחנו מהטבלה המקורית את העמודה של כתובת בתי המלון, הורדנו כפילויות, ובאמצעות סקריפט הפרדנו את כתובת בית המלון והמדינה לעמודות נפרדות, באמצעות סקריפט נוסף שינינו את שם המדינה לערך מספרי – key המתאים לפי טבלה 1.

מכילה 4 עמודות :

Key_Hotel - ערך מספרי, מפתח לבית המלון, עמודה זו משמשת כ primary key
Hotel_Name – מחרוזת, שם בית המלון
Key_Hotel_Country – ערך מספרי, מפתח של המדינה בה נמצא בית המלון
Hotel_Address – כתובת בית המלון למעט מדינה

3. Users

טבלה המכילה את שמות המשתמשים, סיסמה, מין ומדינת המוצא שלהם, לצד ערכים מספריים (מפתחות) אליהם הם ממופים

תהליך יצירת הטבלה : יצרנו שמות משתמש וסיסמאות כמפורט בתחילת המסמך, בנוסף, עם אותו סקריפט שינינו את המדינה של כותב הביקורת לערך מספרי – key המתאים לפי טבלה 1.

מכילה 5 עמודות :

Key_User - ערך מספרי, מפתח למשתמש, עמודה זו משמשת כ primary key
Name – מחרוזת, שם משתמש
Password – מחרוזת, סיסמא
Gender – מחרוזת, מין
Key_User_Country – ערך מספרי, מפתח של המדינה בה מתגורר המשתמש

4. TypeOfTrip

טבלה המכילה את סוגי הטיול (פנאי/ עסקים), לצד ערכים מספריים (מפתחות) אליהם הם ממופים.

תהליך יצירת הטבלה : בטבלה המקורית תחת העמודה Tags היו מספר רב של מאפייני הטיול, בחרנו 2 מאפיינים (עסקים או פנאי) ומחקנו את השאר. התאמנו לכל רשומה

אופציה אחת - האם הטיול היה לעסקים או פנאי, במקרה ולא היה רשום הוספנו באופן רנדומלי.

מכילה 2 עמודות :

Key_Trip – ערך מספרי, מפתח לסוג הטיול (פנאי או עסקים), עמודה זו משמשת כ primary key
Type_Of_Trip - מחרוזת, סוג הטיול

5. Compositions

טבלה המכילה את סוגי הרכב הטיול (יחיד/ זוגי/ משפחה), לצד ערכים מספריים (מפתחות) אליהם הם ממופים.

תהליך יצירת הטבלה : בדומה לתהליך שעשינו בטבלה 4, בטבלה המקורית תחת העמודה Tags היו מספר רב של מאפייני הטיול, בחרנו 3 סוגי הרכבים (יחיד/ זוגי/ משפחה), ומחקנו את השאר. התאמנו לכל רשומה אופציה אחת, במקרה ולא היה רשום הוספנו באופן רנדומלי.

מכילה 2 עמודות :

Key_Composition – ערך מספרי, מפתח לסוג הרכב הטיול, עמודה זו משמשת כ primary key
Composition – מחרוזת, סוג הרכב הטיול

6. Reviews

טבלה המכילה את כל הביקורות על בתי-המלון. את המשתמש שכתב את הביקורת, בית המלון עליו נכתבה הביקורת, תאריך, תוכן הביקורת השלילית, תוכן הביקורת החיובית, דירוג המשתמש, סוג הטיול, סוג הרכב הטיול ומספר הלילות שהמשתמש שהה בבית-המלון, לצד ערכים מספריים (מפתחות) אליהם הם ממופים.

תהליך יצירת הטבלה : עבור כל ביקורת יצרנו מפתח ייחודי, והתאמנו אותה למשתמש ייחודי מטבלה 3. כמו בטבלאות הקודמות החלפנו את הערכים key שלהם מהטבלאות המתאימות (טבלה 3 עבור user, טבלה 4 עבור סוג הטיול, טבלה 5 עבור הרכב הטיול...)

מכילה 9 עמודות :

Key_Review – ערך מספרי, מפתח לביקורת, עמודה זו משמשת כ primary key
Key_User – ערך מספרי, מפתח של כותב הביקורת
Key_Hotel – ערך מספרי, מפתח של בית המלון עליו נכתבה הביקורת
Date - תאריך כתיבת הביקורת
Negative_Review – מחרוזת, ביקורת שלילית על בית המלון

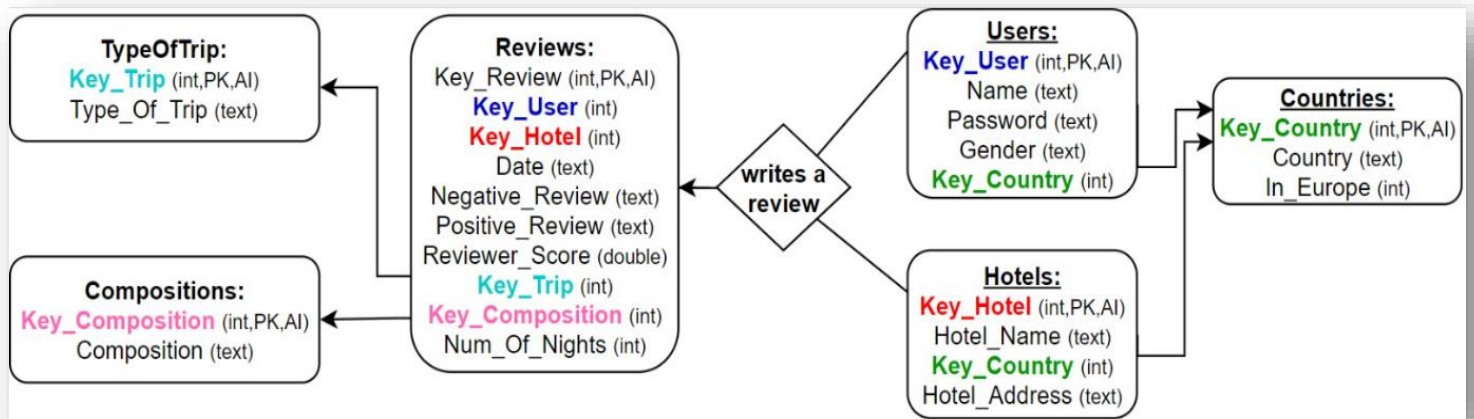
Positive_Review – מחרוזת, ביקורת חיובית על בית המלון
Reviewer_Score – ערך מספרי (double), דירוג מספרי שנתן המשתמש לבית המלון
Key_Trip – ערך מספרי הממופה באמצעות טבלה ומציג את סוג הטיול (עסקים או פנאי)
Key_Composition – ערך מספרי הממופה באמצעות טבלה ומציג את סוג הרכב הטיול
Num_Of_Nights – ערך מספרי הממופה באמצעות טבלה ומציג את מספר הלילות בו
שהה המבקר בבית המלון

שינויים נוספים שעשינו בטבלאות :

ALTER TABLE schema_name.table_name MODIFY COLUMN Key INT **Auto_Increment** **ALTER**
עדכון אוטומטי עבור רשומת Key.
את השינוי הזה עשינו בכל הטבלאות כדי שעבור כל טבלה נוכל להוסיף רשומה חדשה והמפתח
שלה יתעדכן באופן אוטומטי להיות המספר הבא בטבלה.

ALTER TABLE schema_name.table_name **ADD PRIMARY KEY Key** **ALTER**
שינינו בכל הטבלאות את עמודת Key להיות primary key - מפתח ייחודי.

סכמה



שאלות

כל השאלות נמצאות בקובץ מצורף – sql_queries.txt (מספור בהתאמה להסבר כאן)

נסמן שאלתה מורכבת בצהוב

1. כל הרשומות מהטבלה compositions. **SELECT**
2. באיזה בית מלון התארחו הכי הרבה מהרכב מסוים במדינה נבחרת, הרכב לבחירת המשתמש מהרשימה (יחיד, זוג, משפחה, קבוצה, חברים). **SELECT**
3. כל הרשומות מהטבלה countries.
4. מדינה לפי מפתח של המדינה. **SELECT**
5. כל המדינות שנמצאות באירופה. **SELECT**
6. כל המדינות באירופה בהן יש בית מלון המופיע במסד הנתונים. **SELECT**
7. המדינה של כותב הביקורת. **SELECT**
8. כל בתי המלון. **SELECT**
9. כל בתי המלון לפי מדינה מסוימת שהמשתמש בוחר. **SELECT**
10. X בתי המלון באירופה בעלי הדירוג הממוצע הגבוה ביותר. **SELECT**
(X לבחירת המשתמש)
11. X בתי המלון במדינה מסוימת בעלי הדירוג הממוצע הגבוה ביותר. **SELECT**
(X ומדינה לבחירת המשתמש)
12. רשימה של כל בתי המלון מסודרים לפי המפתח. **SELECT**
13. כמות מבקרים (באחוזים) לפי מדינה שביקרו במלון מסוים לבחירת המשתמש. **SELECT**
14. כל הביקורות. **SELECT**
15. כל הביקורות של משתמש מסוים. **SELECT**
16. כל הביקורות על בית מלון מסוים לבחירת המשתמש. **SELECT**
17. הכנסת ביקורת חדשה לטבלת הביקורות. **INSERT**
18. מחיקת ביקורת מסוימת. **DELETE**
19. מחיקת כל התגובות של משתמש מסוים. **DELETE**
20. כל הפרטים מהטבלה typeoftrip. **SELECT**
21. הכנסת משתמש חדש. **INSERT**
22. כל הפרטים של משתמש מסוים לפי שם משתמש (primary key) **SELECT**
23. כל הפרטים של משתמש מסוים לפי key ייחודי של המשתמש **SELECT**
24. עדכון סיסמא עבור משתמש מסוים **UPDATE**
25. כמות מבקרים שהגיעו ממדינה מסוימת שביקרו בבית מלון לבחירת המשתמש **SELECT**

מבנה הקוד

בנינו אפליקציית web בjava, סביבת העבודה היא eclipse

השתמשנו בשרת מסוג 9 tomcat server

מסד הנתונים שלנו הוא mysql workbench 8.0CE

ספריות ג'אווה בהן השתמשנו:

java-json

mysql-connector-java-8.0.13

קבצי הפרויקט -

קוד המקור נמצא בתיקיה src, התיקיה מכילה ארבעה packages :

1. **connections** – בתיקיה זו, נמצא קלאס שיוצר חיבור עם Mysql בהתאם לנתונים שלנו.
2. **databases** – המודלים האחרניים לקשר בין האפליקציה לDB. מכיל את השאילתות שנשלחות לשרת.
3. **object** – הקלאסים המכילים את האובייקטים. כל אובייקט מייצג טבלה אחרת מהסכמה ושדות האובייקט מייצגות את הרשומות.
4. **servlets** – קבצי צד-שרת המנהלים את השאילתות ומבצעים קישור בין השאילתות לקבצי ה-jsp.

בתיקיה WebContent נמצאים:

- כל קבצי ה**JSP** – קבצי עמוד האינטרנט.
- תיקיית WEB-INF המכילה את הקובץ **web.xml**, ובנוסף, תיקיית **lib** המכילה את הספריות בהן השתמשנו.
- תיקיית **img** המכילה את התמונות.
- תיקיית **css** המכילה את העיצוב.

מהלך השימוש באפליקציה :

תחילה המשתמש יצטרך להירשם או להתחבר. לאחר מכן תוצגנה לו אפשרויות מגוונות לביצוע שאילתות כמפורט במסמך user manual, במקרה של תקלה תוצג הודעה למשתמש.