

# BIG DATA & ELASTICSEARCH

מאת: יניב בוחבוט  
מנהל הפרויקט: ד"ר אלי יצחק



**SCE**  
הטכניון האקדמיות לחינוך עירש סמי שטמען

מהנדסים לעולם טוב יותר!  
PROJECT ORIENTED

---

# BIG DATA & ELASTICSEARCH

---

**מאט: יניב בוחבוט**  
**מנהל הפרויקט: ד"ר אלי יצחק**

**הוגש במחלקה להנדסת תוכנה**  
**המכלול האקדמי להנדסה ע"ש סמי שמעון**

## Contents

<b>1 מבוא ל BIG DATA</b>	
6.....	
6.....	6..... אפייניות Big Data 1.1
7.....	נפח 1.1.1
7.....	מהירות 1.1.2
7.....	גיאו... 1.1.3
8.....	אמינות 1.1.4
8.....	ערך 1.1.5
8.....	<b>2 מחסן נתונים</b>
8.....	סוגי טבלאות במחסן הנתונים 2.1
8.....	Data Mart, או מרכול נתונים... 2.2
8.....	טימיד(Dim Table) 2.3
8.....	עובדת (Fact Table) 2.4
9.....	סכמת כוכב 2.5
9.....	סכמת פתייחי שלג 2.6
10.....	<b>Hadoop ecosystem</b> 3
10.....	HDFS 3.1
11.....	MapReduce 3.2
13.....	<b>4 בסיסי נתונים NOSQL</b> 4
13.....	MongoDB 4.1
14.....	MongoDB vs. MySQL : השוואה בין שני המודלים 4.1.1
15.....	ארქיטקטורה של בסיס הנתונים MongoDB: 4.1.2
15.....	צבר שכפול 4.1.3
15.....	צבר פיזור 4.1.4
16.....	Cassandra 4.2
16.....	תכונות עיקריות 4.2.1
16.....	מודל נתונים ... 4.2.2
17.....	טור 4.2.2.1
17.....	רשומה 4.2.2.2
18.....	HBase 4.3
21.....	<b>5 מנועי חיפוש</b> 5
21.....	Apache Solr 5.1
21.....	Sphinx search 5.2
21.....	.Search technologies in cloud. 5.3
22.....	<b>ELASTIC SEARCH</b> 6
22.....	Elasticsearch תכונות כלליות... 6.1
22.....	מושגים מרכזיים ב Elasticsearch 6.2
23.....	יתרונות של Elasticsearch 6.3
23.....	חרוננו של Elasticsearch 6.4
23.....	השוואה בין RDBMSI Elasticsearch 6.5

24.....	<b>מетодולוגית הסימולציה עם ELASTICSEARCH/KIBANA</b>	<b>7</b>
27.....	ELASTICSEARCH	7.1
27.....	אכлом בתחום	7.1
28.....	Create Index API	7.1.1
28.....	Get API	7.1.2
30.....	Delete API	7.1.3
31.....	Update API	7.1.4
32.....	Multi Get API	7.1.5
34.....	BULK API	7.1.6
35.....	ELASTICSEARCH — DOCUMENT APIs	7.1.7
35.....	Elasticsearch -Mapping	7.2
36.....	Field Datatypes	7.2.1
36.....	Dynamic mapping	7.2.2
36.....	Explicit mapping	7.2.3
37.....	ELASTICSEARCH — AGGREGATIONS	7.3
37.....	Metrics Aggregations	7.3.1
37.....	Average Aggregation	7.3.2
37.....	Cardinality Aggregation	7.3.3
38.....	Max Aggregation	7.3.4
38.....	Min Aggregation	7.3.5
39.....	Bucket Aggregations	7.3.6
39.....	Geo Distance Aggregation	7.3.6.1
40.....	Filter Aggregation	7.3.6.2
40.....	ELASTICSEARCH — QUERY DSL	7.4
40.....	Full Text Queries	7.4.1
41.....	Match_query	7.4.1.1
41.....	Multi Match Query	7.4.1.2
42.....	Terms Query	7.4.1.3
43.....	Wildcard Query	7.4.1.4
44.....	type query	7.4.1.5
44.....	Range Query	7.4.1.6
45.....	Geo Queries	7.4.2
45.....	Geo point	7.4.2.1
47.....	Geo Distance sorting	7.4.2.2
47.....	Geo_distance query	7.4.2.3
49.....	Geo Distance Query	7.4.2.4
49.....	Geo Bounding Box Query	7.4.2.5
50.....	Geo polygon query	7.4.2.6
51.....	ELASTICSEARCH —KIBANA	8
52.....	Console Dev Tool	8.1
53.....	Discover	8.2
54.....	Visualize	8.3
56.....	Dashboard	8.4

57	9	סיכום
58	10	הפניות
59	10.1	ביבליוגרפיה
59	10.2	רשימת איורים

## הקדמה

בשנים האחרונות הנתונים גדלו, בקצב מהיר מאוד, בנתונים עצומים ובמגוון רחב של סוגים ותחומים שונים. על פי דוח של חברת נתונים בינלאומי IDC בשנת 2011 בלבד הנפח העולמי של נתונים שנוצרו בעולם היה ZETA 1.8BYTE

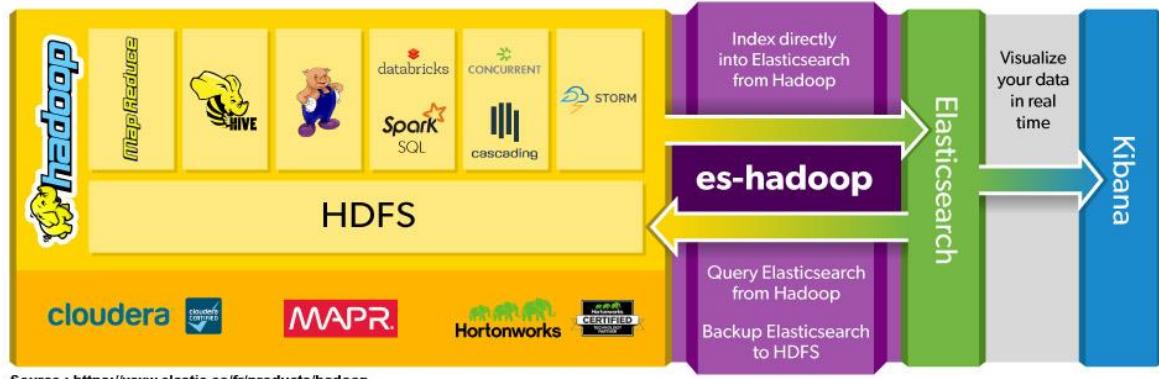
- גугл, פיסבוק, אפל, מיקרוסופט, NSA, ומגוון רחב של ארגונים הנעזרים בפתרונות בלתי נטפסות של מידע כגון שיחות טלפון מוקלטות ועד הרגלי קניה של לקוחותיהם, נתקלים באחד האתגרים הגדולים של טיפול ב Big Data מיצי תובנות ומסקנות מתוך המידע הזמין. בשנים האחרונות עלו לקדמת הבמה מספר טכנולוגיות המאפשרות לעבוד כמוניות גדולות של מידע בזמן קצר יחסית, באמצעות ביזור המידע בין מספר רב של מחשבים שבידיהם אותו במקביל. כמה מטכנולוגיות אלה הקשורות ל Big Data הם Elasticsearch מאגרי NOSQL ו Hadoop אשר נותנים מענה מושלם לחיפוש נתונים להררי DATA.

## מטרת הפרויקט והמחקר

מטרת הפרויקט הייתה ניתוח ההבדלים של סוגי מסדי הנתונים NOSQL הקיימים היום בשוק הקשורים לביג DATA, ביחיד עם זה לבצע סימולציה בעזרת כל חיפוש Elasticsearch.

פרויקט זה מבצע מחקר אשר מוסבר בפרק 2 - 5 בנושאים כגון : חשיבות תשתיות האחסון בארגונים, ארכיטקטורת האחסון של תשתיות ה Hadoop, שלבי עיבוד נתונים שלה, מגנוני מיצוי נתונים ומודל שלושת מסדי נתונים NOSQL השונים הפופולריים הקיימים היום בשוק. ולבסוף מחקר זה עבר על סוג מנوع החיפוש השונים המהווים את הפתרונות החדשניים של היום בדגם מפרק 6 על Elasticsearch.

יתרה מזו, מחקר זה יראה סימולציה של שאלות וഫונקציות שונות אנליטיות של מוצר - Elasticsearch / Kibana ותיעוד טכני במוגן רחב של אפשרויות כגון: אינדוקס, חיפוש, ניתוח ויזואלייזציה של נתונים על ידי תרשימים וגרפים



איור 1 ארכיטקטורת קישור בין מנוע חיפוש Elasticsearch של Big Data לבסיס נתונים.

## 1 מכוֹא ל BIG DATA

הכמות האדירה של הנתונים בעולם אילצה פיתוח דרכים חדשות בכך לצפות ולנתח את הנתונים, בסדרי גודל חדשים הנוגעים לאחסון, לאחסון, לשיטוף, ניתוח והדמיה של נתונים. BIG DATA, פשוטו ממשמעו "נתונים עתק", מתייחסים למערכות נתונים שהפכו להיות גדולים כל כך, שהם חורגים מיכולת האינטואיציה והניתוח של האדם, ואפילו מלאו של כל ניהול נתונים או מידע מבוסס מחשב.

כדי להתמודד עם הכמות האדירה של נפח הנתונים, תחום טכנולוגי חדש שנקרא ביג דטה. פתרונות אלה מיוצרים על ידי ענק האינטרנט, שנעודו לספק גישה בזמן אמת למסדי נתונים ענקיים. המונח ביג דטה מתייחס לדיסציפלינה חדשה בזאת של מספר תחומי: סטטיסטיקה, טכנולוגיה, מסד נתונים ועסקיות (שוויוק, כספים, משאבי אנוש וכו'). דיסציפלינה חדשה זו התאפשרהודות לכוח טכנולוגי אשר אפשר דברים שעדיין אין תיאורתיים בלבד. הדברים שאנו מדברים עליהם קשורים בעיקר לשתי סוגיות א. היקף הנתונים ב. מרכיבותם.

### היקף הנתונים:

מול הפיצוץ בהיקף המידע, ביג דטה שואפת להציג חילופה למסדי נתונים ומסדי נתונים מסורתיים (SQL Server, PLATFORUM, בינה עסקית...). התמודדות עם בעיות בكمיות גדולות מאוד, ענק האינטרנט, ובראשם YAHOO, FACEBOOK ו GOOGLE הי הראשונים שפרשו סוג זה של טכנולוגיה.

### מרכבותם:

ביג דטה מלאה בפיתוח של יישומים אנליטיים, אשר מעבדים את הנתונים, ניתוחים אלה נקראים Big Analytics או "ניתוח נתונים". הם עוסקים בנתונים כמותיים מורכבים בשיטות חישוב מבזירות וסטטיסטיות.

### Big Data 1.1 אפיינוי

לדברי GARTNER (חברה אמריקאית העוסקת במחקר וביעוץ בטכנולוגיית המידע) ביג דטה היא משפחה של כלים העוסקים בבעיה שלושת הכללות נפח, מהירות וגיוון. כלים אלה כוללים עיבוד של נתונים, רמה מסוימת של מהירות וגיוון רחב של מידע (מספר מקורות, בלתי מובנים, מבנים, נתונים Open data וכ"ד) כלומר, תדיות היצירה, איסוף, עיבוד, ניתוח ושיטוף של נתונים אלו מटבטים בכלים שנקרוים V3 ו V5<sup>1</sup>.

### כלל: V3

הנתונים הם אכן הבסיס לחומר גלם של תופעת ביג דטה ניתן לאפיין נתונים עתק לפי שלושה מאפיינים.<sup>2</sup>

- נפח (volume)
- מהירות (velocity)
- גיוון (variety)

### כלל: V5

- נפח (volume)
- מהירות (velocity)
- גיוון (variety)
- אמינות (Veracity)

<sup>1</sup> <https://www.gartner.com/technology/topics/data-analytics.jsp>

<sup>2</sup> <https://www.xsnet.com/blog/bid/205405/the-v-s-of-big-data-velocity-volume-value-variety-and-veracity>

<sup>3</sup> <https://hrboss.com/blog/2014-03-26/missing-vs-big-data-hr-5-v-model-here>

### נפח 1.1.1

הנפח מתאר את כמות הנתונים שנוצרו על ידי חברות או אנשים. עסקים בכל המגזרים יצרכו למצוא דרכי לנחל את הנפח ההולך וגדל של הנתונים שנוצר מדי ים. הערכה היא כי 90% מהנתונים שנאפסו מאז תחילת האנושות נוצרו במהלך השנים האחרונות. הנתונים מכילים את המידע הבא:

- קטיעי וידאו אנו מפרסמים.
- מידע האקלים.
- אוטומט GPS.
- רשומות עסקאות טרנסקציות.
- טקסטים מרשתות חברתיות.
- ועוד...

יש גידול שנתי של יותר מ -50%, נפח הנתונים הזמינים גדל באופן אקספוננציאלי. הנתונים המספריים שנוצרו ברחבי העולם עברו מ- 1.2 zettabytes בשנת 2010 ל- 1.8 zettabytes ב- 2011, ולאחר מכן ב- 2.8 zettabytes ב- 2012, ויסתכמו ב- 40 zettabytes ב- <sup>4</sup>2020. כדוגמא, Twitter יירה בינואר 2013, 7 טרה-בתים של נתונים בכל יום ו- 10 Facebook טרה-בייט. בשנת 2014.

### מהירות 1.1.2

המהירות מתארת את התדריות שבה נתונים נוצרים, נלכדים ומשותפים. בשל ההתפתחויות הטכנולוגיות האחרונות, צרכנים אף גם חברות מייצרים נתונים נוספים בזמן קצרים בהרבה. בrama זו של מהירות, חברות יכולות רק לנצל את הנתונים הללו אם הם נגישים ומשותפים בזמן אמיתי. כאשר בדיק בשלב של מהירות בזמן אמיתי ממערכות גדולות ככל כמה שניות, במקרה הטוב. אך, נתונים אלה אין כבר ערך מאז התחילה של מחזור של דור של נתונים חדשים.

### גוון 1.1.3

התפשטות של סוג נתונים ממוקורות כגון מדיה חברתית, אינטראקטיות בין מכונה והתקנים ניידים יוצרת מגוון גדול מעבר לנתוני טרנסקציות מסווגים. הנתונים אינם עוד חלק של מבנים קלים לשימוש(צריכה). סוג נתונים חדשים כוללים תוכן נתונים, מיקום גיאוגרפי, נתונים שנוצרו על ידי מכונות, נתונים מדידה, נתונים ניידים, תהליכי, נתונים RFID, נתונים מדיה חברתית, טקסט נתונים מהאלינטראנט. המידע החדש של מערכות Big Data אינם תמיד מגע בצורה של שורות ועמודות (מידע מובנה), ולכן קשה יותר לטפל במידע זהה בצורה מסווגית. מחלקים את נתונים data big באופן הבא:

1. נתונים מובנים : נתונים יחסיים (טבלאות, סכבות, פרוצדורות, פונקציות).
2. נתונים חיצי מובנים: הודיעות דואר אלקטרוני (ohoודעות מידיות יותר),
3. נתונים לא מובנים: PDF, Word, טקסט, מדיה (תמונה, וידאו, שיר).

## 1.1.4 אמינות

מהימנות הנתונים הפכה לקריטריון חיוני. האמינות מתייחסת לאמינות הנתונים. עם כל כך הרבה צורות של נתונים יש בעיה של אמינות הנתונים (נסתכל על tweets עם hashtags, שגיאות הקלדה, שפה בוטה, אמינות של תוכן). חוסר איות ודיק נובעת לעתים קרובות מהנפקים הגדולים.

## 1.1.5 עבר

מושג הערך הוא הרוח שנitin להפיק מהשימוש בנתוני Big Data של חברה למשל. ערך זה יכול להיות במספר צורות. בנוסף לזה פחות מ-1% מהחברות פיתחו פתרונות שמנצלות את הפוטנציאלי של הנתונים ובניהם שלם.

## 2 מחסן נתונים

### 2.1 סוג טבלאות במחסן נתונים

מחסן נתונים, או Data warehouse , מהוות את מרכז המידע של החברה. זהו מבנה (כמו מסד נתונים) שמטרתו, ביגוד למסדי נתונים, לקובץ נתונים עסקיים למטרות אналיטיות ולסייע בקבלת החלטות אסטרטגיית, יכול לכלול את כל סוג מסדי הנתונים כגון Fact table, Dim table, Data mart ועוד <sup>5</sup>. ההחלטה האסטרטגית היא פועלה של מכבלי החלטות בחברה, שמטרתה לשפר, מבחינה כמוותית או איכותית, את ביצועי החברה. בעיקרו, זו כמות של מידע עצומה מאורגן, אשר מנויינת על פי זמנים, מקורות נתונים רבים. מחסן הנתונים הוא המרכיב המרכזי של בניית עסקית . ואכן, מחסן הנתונים הוא הדרך הטובה ביותר שאנשי מקצוע מצאו כדי למדוד מידע למטרות אналיטיות.

### 2.2 Data Mart, או מרכול נתונים

מחסני נתונים באופן כללי נפוץ גדוֹל ומורכבים מאוד לתכנן, זו הסיבה לכך שחלק מהארגוני אימצו גישה חלפית של הקמה הדרגתית של מאגרי נתונים חלקיים וקטנים יותר הנקראים Data Marts. אלו יכולים לבצע חלוקה לפי פונקציונליות (mart למכירות, עבור הזמן, עבור משאבי אנוש). במובן זה the Data-mart הוא מעין מחסן נתונים של ארגון פנימי המהווה חלק מהארגון <sup>6</sup> ניתן למצאו בארגון מספר Data marts.

### 2.3 סימד(Dim Table)

בעת יצירת סכמה DB עבור מערכת מידע קלאסית, מדברים במקרים של טבלאות ייחודיים, טבלה היא יוצג של ישות ויחסים וטכנית ל קישור גופים אלה. ובכן ב- BI, אנחנו מדברים במקרים של מדד ועובדות. זהה גישה נוספת לנתונים, אנו מתכוונים לפ' מדדים, לzeros שמהם נרצה לעשות את הניתוח <sup>7</sup>. תיכון שהייה לחברה מימד לקוחות, מימד מוצר, מימד גיאוגרפיה (לניתוח גיאוגרפי) וכן הלאה. מימד הוא כל מה שאנו נשתמשים כדי לעשות את הניתוח שלנו.

### 2.4 עובדה (Fact Table)

טבלת ה Fact היא למעשה "לב" המערכת ומכליה נתונים לגבי הדבר שאותו רצים לנתח ולחקרו. מן הסתם, טבלאות אלו משתנות מבין ארגון אחד למשנהו ובין פרויקט לפרויקט. דוגמאות לטבלאות כאלה יכולות להיות: חברות תקשורת-טבלת שיחות מרכזית שתכיל פרטיים לגבי כל שיחה שבוצעה, כגון מספר טלפון , זמן התחלת שיחה וזמן סיום שיחה, משך זמן שיחה ועלות השיחה למיטלפן. אלה הם טבלאות המכילות מידע תפעולי או טבלאות למכירות (מכירות נתו, כמויות וSOCIALIM שhortzman, כמויות חשבוניות, כמויות מוחזרות, מחזורי מכירות וכו' ) לדוגמה או במלאי (מספר עותקים של מוצר במלאי, רמת مليי מלאי, תחלופה של

<sup>5</sup> <https://dzone.com/articles/difference-between-data-warehouse-and-data-mart>

<sup>6</sup> <https://www.sisense.com/glossary/data-mart>

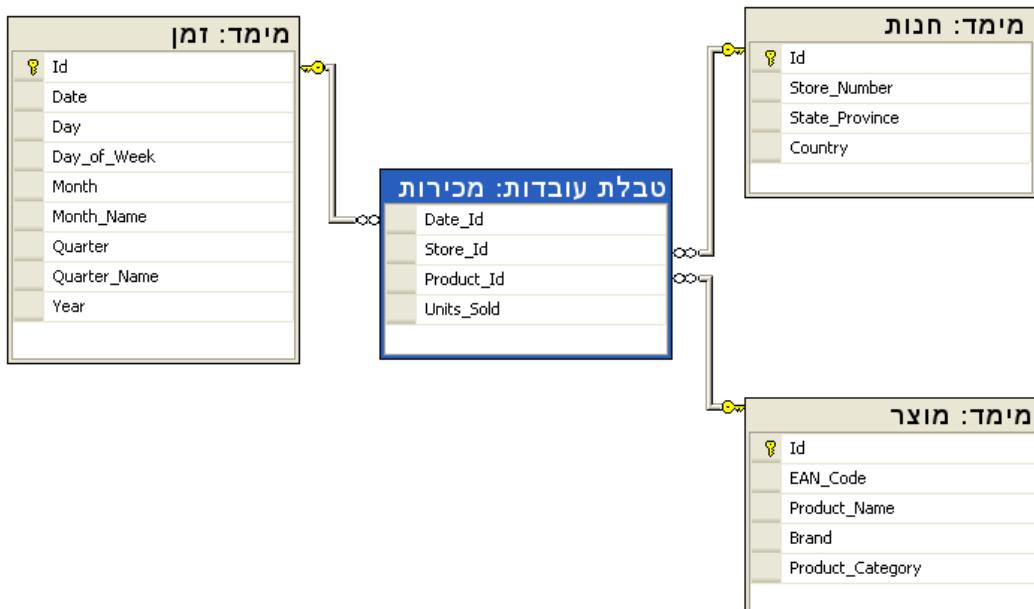
<sup>7</sup> <https://searchdatamanagement.techtarget.com/definition/dimension-table>

שטח וכו'), או על משאבי אנוש (ביצועי עובדים, מספר בקשות חופשה, מספר הפסיקות, תחלופת עובדים וכו'). עובדה היא כל מה שאנו חוננו רוצים לנתח

## 2.5 סכמת כוכב

סכמת כוכב היא למשה הסכמה הקלאסית והפשוטה ביותר שבה במרכז נמצא את טבלת העובדות ומסביבה טבלאות המידם.

דוגמא לסכמה זאת נראית בדיאגרמה הבאה:



איור 2 סכמת כוכב

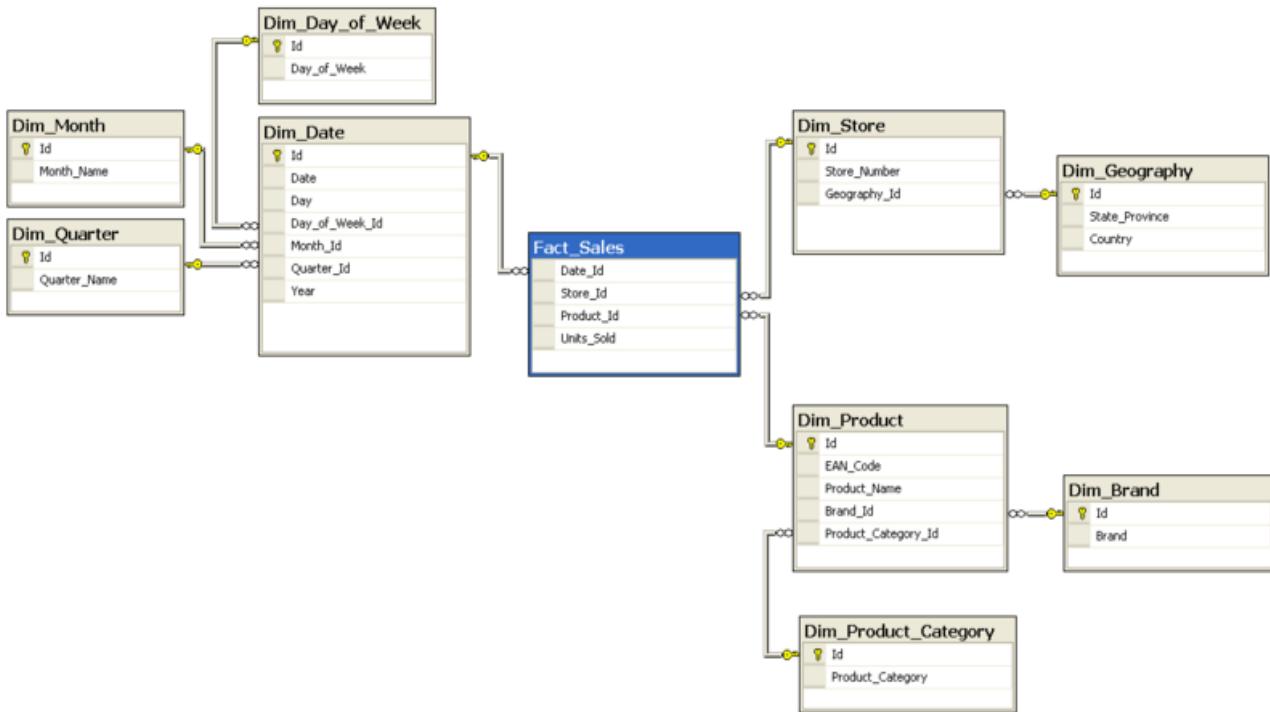
כפי שצויג בדיאגרמה, רואים שטבלת העובדות מכילה טבלאות מימד-נתונים כגון זמן, חנות ומוצר וזה מבהיר שטבלת העובדות מכילה את שמות בסיסי הנתונים הללו (זמן-id Date\_id חנות-id Store\_id מוצר-id Product\_id).  
כל טבלה יש מפתח ראשי בשורה 'מזהה' שלה, המתיחס לאחד מהשורות של המפתח הראשי של שלוש השורות בטבלה (Date\_Id, Store\_Id, Product\_Id).

## 2.6 סכמת פתיתו שלג

לעתים, כדי למנוע כפליות של נתונים, מנהתי המערכת נוטים לפרק ממדיים לממדים משנה. כלומר, המימד מכיל קוד של תוכנה וטבלה נוספת מפרטת תוכנה זאת<sup>8</sup>

דוגמא לסכמה זאת נראית בדיאגרמה הבאה

<sup>8</sup> [https://en.wikipedia.org/wiki/Snowflake\\_schema](https://en.wikipedia.org/wiki/Snowflake_schema)



איור 3 סכמת ה Snowflakes

סכמות כוכב ופתייה שלג נפוצות בעיקר בمخסני נתונים מד"ם (Data Warehouse) ובמרכזי נתונים (Data Marts) בהם מהירות שליפת המידע חשובה יותר מיעילות ביצוע מניפולציות על המידע. מכיוון שכרכובות בסכמות אלו אין מונרמולות לסדר גובה ולרבות מונרמולות עד לרמת נורמליזציה NF3 או נמוכה ממנה.

## Hadoop ecosystem 3

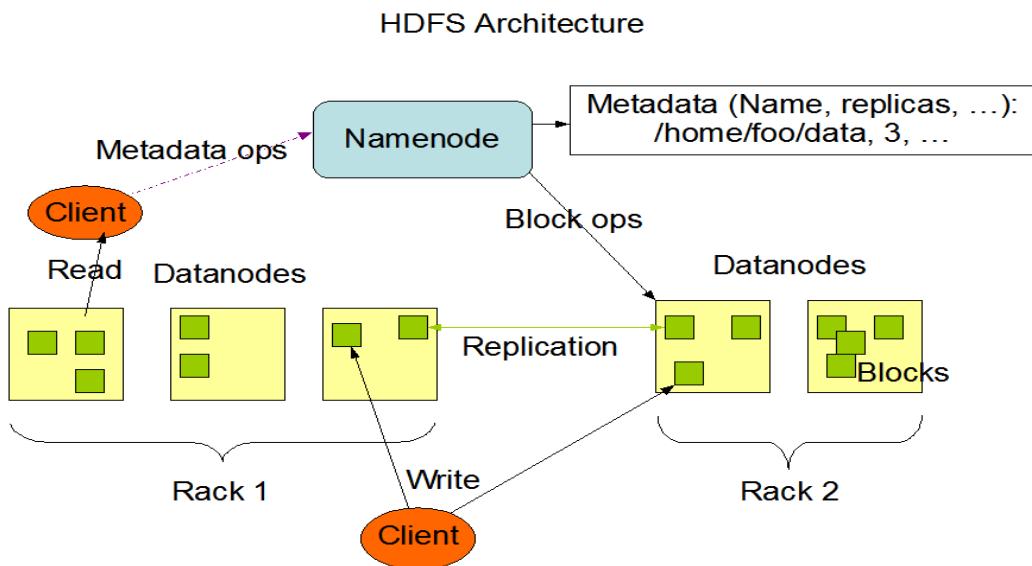
Hadoop היא מסגרת חופשית וקוד פתוח שנכתבה ב- Java כדי להקל על ייצור יישומים מבזרים (bihos לאחסון נתונים ועיבודם) ומאפשרת ליישומים לעבוד עם אלפי גטטים petabytes של נתונים. כך-shell צומת מרכיבת מרשתים מקובצות באשכול. כל המודולים של Hadoop מבוססי שעוריון הבסיס שיכלים בחומרה נפוצים יותר ולכן הם יכולים להיות מטופלים באופן אוטומטי על ידי מסגרת זו.

לבת ה Hadoop מרכיבת מחלק אחסון: **HDFS** (מערכת קבצים מבוזרת Hadoop), וקטע של עיבוד שנקרא **MapReduce**<sup>9</sup>. Hadoop מחלק את הקבצים לבlokים גדולים ומפיץ אותם על פני גטטים. כדי לעבוד את הנתונים, Hadoop מעביר את הקוד לכל צומת וכל צומת מעבד את הנתונים שיש לו. זה מאפשר לעבוד את כל הנתונים ב מהירות ובעילوت רבה יותר מאשר ארכיטקטורת מחשב העל הקונבנציונלי, אשר מסתמכ על מערכת קבצים מקבלים שבו חישובים נתונים מופצים על גבי רשותם ב מהירות גבוהה.

### HDFS 3.1

HDFS היא מערכת קבצים מבוזרת שפותחה על ידי Google FS - Hadoop. נכתבה ב Java, היא נועדה לאחסן כמויות גדולות מאוד של נתונים על מסך גדול של מכונות מצידות בכוננים קשיים. זה מאפשר את הפשתה של ארכיטקטורת האחסון הפיזי כדי לתפעל מערכת קבצים מבוזרת כאילו היה דיסק קשיח יחיד. ארכיטקטורת מחשב HDFS (הנראית גם אשכול HDFS) משתמשת על שני סוגים עיקריים של רכיבים:

<sup>9</sup> <http://www.corejavaguru.com/bigdata/hadoop/hdfs-architecture>



איור 4 רכיבי ארכיטקטורת HDFS

#### **Name Node**

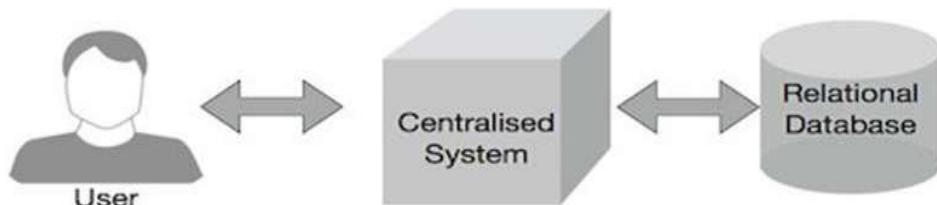
צומת שם, רכיב זה מנהל את מרחב השמות, אתüz מערכת הקבצים ואת מטא נתונים של קבצים וספריות. הוא מרכז את המיקום של בלוקים של הנתונים המבוזרים באשכול. הוא ייחודי אבל יש לו מופע משני שמנוהל את ההיסטוריה של השינויים במערכת הקבצים (תפקיד גיבוי). Name Node המשני מאפשר לאשכול Hadoop למשיך לפעול במקרה של כשל של Name Node המקורי.

#### **Data Node**

צומת נתונים, רכיב זה אחסון ומחזר את הבלוקים של נתונים. במהלך תהליך הקריה של הקובץ Name Node נשאל על מנת לאתר בלוקים של נתונים. עבור כל אחד מהם, ה Name Node מוחזר את הכתובת של ה Data Node הנגיש ביותר, ככלומר, כלומר, Data Node שיש לו את רוחב הפס הגדול ביותר.

## MapReduce 3.2

למערכות ארגוניות מסורתיות יש בדרך כלל שירות אחד מרכזי לאחסון ולביצוע נתונים. האיור הבא מציג תצוגה סכמתית של מערכת ארגונית מסורטיבית. המודל المسؤول הוא בהחלט לא מתאים לטיפול בكمויות עצומות של נתונים שנർחים עם הזמן ואינו נתמך על ידי שירותי מסדי נתונים סטנדרטיים



איור 5 מסדי נתונים סטנדרטיים בארגוניות

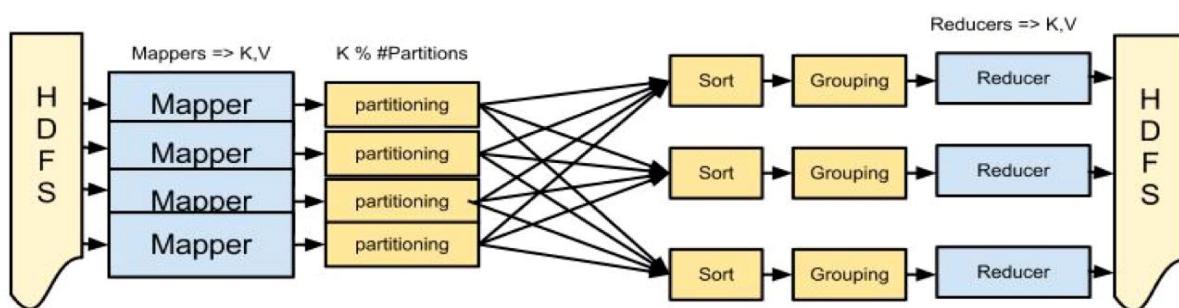
בנוסף, מערכת מרכזית יוצרת צוואר בקבוק יותר מדי בעת עיבוד מקבילי של קבצים רבים. Google פתרה את בעית צוואר הבקבוק באמצעות אלגוריתם שנקרא [MapReduce](#)<sup>10</sup>.

חלוקת MapReduce מחלק משימה לחלקים קטנים ומקצת אותם למספר מחשבים. מאוחר יותר, התוצאות נאספות במקום אחד ומשולבות כדי ליצור את התוצאה מערך נתונים.

MapReduce היא פרדיגמת תכנון המאפשרת להפיץ עיבודים מקבילים על נזינים של נתונים בדרך כלל מעל 1TB, באשלול מרכיב מסוימות או אפילו אלפי צמתים - מconiוט (שרתים), עם ארכיטקטורה Master / Slave, בזכות הפרדת נתונים וטיפולם. עיבוד נתונים מקבילי עם MapReduce הוא פשוט נעשה עם שתי פעולות הנקראות Reduce | Map.

- **Reduce** מכיל שתי שימושות חשובות, **Map** ו- **Reduce** :
- + **Map** לוקחת קבוצה אחת של נתונים וממיר אותה לקבוצת נתונים אחרת, שבה האלמנטים בודדים מפורקים לתוך tuples (מפתח/ערך).
- + **Reduce** לוקחת את הפלט של **Map** כקלט ומשלבת נתונים tuples אלה (זוגות מפתח/ערך) בקבוצה קטנה יותר של tuples. משימת **Reduce** מבוצעת תמיד לאחר עובדות **Map**.

cutet נסbir על כל אחד מהשלבים:



The MapReduce Pipeline

A mapper receives (Key, Value) & outputs (Key, Value)  
A reducer receives (Key, Iterable[Value]) and outputs (Key, Value)  
Partitioning / Sorting / Grouping provides the Iterable[Value] & Scaling

איור 6 שלבי האלגוריתם: MAP-REDUCE

- + **שלב "Split":** בשלב זה Framework HDFS מחלק את רשיימת הקלטיים (שימושות) לתוך פיצולים רבים וכל פיצול יוקצה למוכנה בשם mapper, כדי לעבד נתונים בכנה מידת גודל יותר, מה שהופך עיבוד עצמאי ולא היינו צריכים שייעבדו על ידי שני צמתים שונים.
- + **שלב ה "Map":** את ה mappers ביצעו את פעולה ה "map", על ה split -ים שהוקצו להם. ולאחר מכן יצרו רשימה ביןיהם של ערכי מפתח.
- + **שלב ה "Shuffle and Sort":** בשלב זה, ה HDFS Framework ממיין את הקבצים ביחס למפתח של כל אחד, ולאחר מכן מחליף את splitים בין הצמתים, וכל split מכיל את הערכים המשווים לאותו מפתח, אשר יוקצו לאחר מכן לReducer.

[https://www.sas.com/nl\\_nl/insights/big-data/hadoop.html](https://www.sas.com/nl_nl/insights/big-data/hadoop.html)<sup>10</sup>  
<http://ramkedem.com/mapreduce-%D7%95-hdfs-%D7%91%D7%9E%D7%99%D7%9C%D7%99%D7%9D-%D7%A4%D7%A9%D7%95%D7%98%D7%95%D7%AA><sup>11</sup>

שלב ה "Reduce" בשלב זה כל Reducer לפונקציה Reduce ומימוש פונקציות צבירה, על מנת לאוסף ולייצר את התוצאה הרצiosa.

למרות שהרעיון של MapReduce נראה פשוט, העיצוב שלו מורכב ולעתים קרובות הפתוohn געשה ע"י מפתחים מנוסים.

לטיכום, MapReduce מציעה ארכיטקטורת תוכנה ש解脱ת את המורכבות של המחשב המופץ, אבל נדרש את הארכיטקטורה הטכנית המאפשרת לנו להגדיר את הפיזול ואת ה-"Shuffle and Sort", שלבים שראים.

## 4. בסיסי נתונים NoSQL

מאגרי NoSQL הופיעו בשוק בזמן ש Hadoop היה בפיתוח. המוטיבציה העיקריית מאחוריה התנעה NoSQL היה לפתח מסדי נתונים בכדי לפחות את בעיית האחסון וגישה למידע שאינו ממודל במבנה טבלאי ייחודי אשר נפוץ בבסיסי נתונים ייחודיים. מבנה המידע שונה ממערכות בסיסי נתונים ייחודיים, ולכן ישן פעולות שמהירות יותר ב-NoSQL. כיום בסיסי נתונים מסווג SQL-NoSQL הופכים נפוצים יותר במערכות Big Data וכל אחד מנסה לפתח סוגים שונים של בעיות גישה לנ נתונים. ניתן לסכם את מסדי הנתונים NoSQL במאפיינים העיקריים של סביב ארבעה שכבות לוגיות:

- שכבת מודל נתונים, אשר הנתונים מבוססי עמודות, מסגר, גרפ, וכו'.
- שכבת הפצת נתונים המבטיחה הגדלה אופקית על צמתים מרובים הנמשכים על ידי עקרונות משפט CAP. היא מגיע יhud עם תמייה עבור מרכזי נתונים מרובים הקזאה דינמית (הוספה / הסרת צמת מהיצור של אשכול).
- שכבת התמדה עם גמישות של אחסון הנתונים או בדיקן או בשנייהם.
- שכבת משקל עם תמייה במשקי NoSQL שונים (REST, Thrift, API, JSON ספציפיים, וכו' ).

כרגע יש כ 150 מסדי נתונים NoSQL <sup>12</sup> זמינים על פני קוד פתוח וקטגוריות מוצרים מורשים.

אנו נסקור את שלושת מסדי הנתונים NoSQL הפופולריים אשר נמצאים בשימוש נרחב בחברות גדולות כמו גם בחברות קטנות ובינוניות ומשתלבות עם המנוע החיפוש Elasticsearch.

### MongoDB 4.1

MongoDB הוא אחד מסדי נתונים NoSQL בקוד פתוח הפופולרי ביותר, המספק鄙出ים גבוהים וזמינות עם יכולת לגודל באופן גמיש (scalability) עם תוספת של צמתים (Node) נוספים. MongoDB תומך גם במודול נתונים המבוסס על **ODBKit מסמך**. MongoDB מקודד בשפת התכנות C++ ומאחסן את הנתונים בפורמט BSON (JSON ביןארי), המבוסס על פורמט (JavaScript Object Notation (JSON) (JavaScript Object Notation (JSON). מאגרי MongoDB מכילים מערך נתונים אחד או יותר, המנהלים מסמכים שונים עם שדות נתונים שונים ומגוונים.

במסגר, ניתן להוסיף שדות, למחוק, לשנות ולשנות שם בכל עת. בנייתם למסדי נתונים ייחודיים, אין סכמה מוגדרת מראש. המבנה של המסמך הוא פשוט מאוד, הוא מורכב מזוגות Key/Value בצורה של מערכם אסוציאטיביים, המפתח הוא שם השדה, הערך התוכן שלה (בפורמט JSON) השניים מופרדים עם ":". כפי שמצוג בדוגמא לעיל. "ערך" יכול להיות מספר או טקסט, אך גם נתונים ביןaries (כגון תמונה) או אוסף של זוגות מפתח / ערך אחרים.

```
{
  "_id": Object ID ("4efa8d2b7d284dad101e4bc7"),
  "FName": "Bohbot",
  "Name": "Yaniv",
  "Age": "27",
  "Adresse": { "Street": "1 BEN ZVI", "Ville": "MODIIN"}
}
```

## השוואה בין שני המודלים : MongoDB vs. MySQL 4.1.1

MongoDB הוא מסד נתונים **מנוחה מסמכים**, בניגוד ל MySQL. מערכת ניהול זו פועלת אחרת במנוחים של אחסון נתונים, אם כי המבנה הבסיסי יש כמה קווי דמיון (similitude).

- + הנתונים מאוחסנים באוספים ולא בטבלאות. אוסף הוא קבוצה של מסמכים MongoDB. זה שווה ערך לטבלה RDBMS. אוסף קיים בתוך מסד נתונים אחד. מסמכים בתוך אוסף יכולים לכלול שדות שונים. בדרך כלל, כל המסמכים באוסף הם בעלי תכילת דומה.
- + מסמכי BSON מחליפים את הרשומות והשדות המוגדרים במסמכים אלה ואת עמודות השאלה של SQL.
- + שדה מורכב תמיד מערך ושם שדה. ערך זה יכול להציג על רשימות של מילימ או מספרים, טקסט או קובץ מלא.
- + מסמך MongoDB הוא הוסף של זוגות ערך / מפתח, בדיק כמו עם MySQL.

ההבדל העיקרי מסתמך בכך כלל על מצב העבודה של מסמכים, אשר פועלם על פי דפוס מסוים, בעוד שרשותם בMySQL יש להם את אותו הרכב.

יש להם אותה כמות ערכים ותמיד יש להם אותו סוג נתונים. למסרכי MongoDB, לעומת זאת, יש מבנה משליהם ויחודי. בדרך זו, ניתן להוציא שדות חדשים בכל עת, עם כל ערך מוגדר. מסד נתונים יחסיים כמו MySQL דורש שינוי מבני מלא. המפתח חייב להיות ייחודי בכל מסמן, אך ניתן למצוא אותו במסמכים אחרים. תהליך זה אינו אפשרי עם מסדי נתונים של MySQL. יש לקבוע את היחסים בין הטבלאות השוניים.

הבדל גדול נוסף מיצני נתונים, כמו מסד נתונים NoSQL MongoDB, מבוסן אינו משתמש ב- SQL כשפת שאלות ועובד נתונים בשפותו, זה מאפשר תקשורת בין MongoDB לבין הלוקוח המיעוד. לשם כך, מסד הנתונים משתמש בשיטות ספציפיות של שפת התכנות של הלוקוח המדובר, באמצעות מה שנקרא ספריות, כולל דרייברים שניתן להוריד בנפרד על דף רשמי.

עבור שאלות מורכבות ביותר, ניתן להשתמש ב MapReduce, כמו עם כל מסדי נתונים מוכווני מסמכים.

הטבלה הבאה מציגה את הקשר של המינוח MongoDB עם RDBMS:

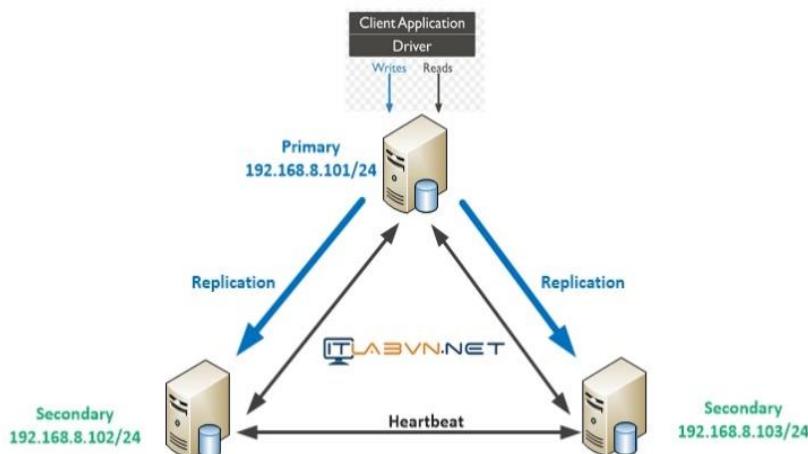
RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by Mongo db itself)

## 4.1.2 ארכיטקטורה של בסיס הנתונים MongoDB:

### 4.1.3 צביר שכפול

מתכונת בסיסית: מורץ תהליך אחד על מכונה אחת, במצב זה אין יכולת להתמודד עם עומסים. פתרון : שכפול נתונים לתוכה מסדר נתונים בცיבור: **Replica Set**

כל הנתונים משוכפלים לכל היותר ב 12 מחשבים בცיבור רצוי, מתקיים תהליך הצבעה ונבחר מחשב **Primary** ושאר המחשבים נחברים כ-**Secondary**<sup>13</sup>. פעולה קריאה מתבצעת מאחד המחשבים, פעולה עדכון מתבצעת על מחשב ראשי, מחשב ראשי שולח עדכנים למחשבים אחרים, במצב של של קיימת אפשרות להתמודד באופן אוטומטי. (אם מחשב ראשי נפל נבחר מחשב ראשי חדש והעבודה המשיכת).

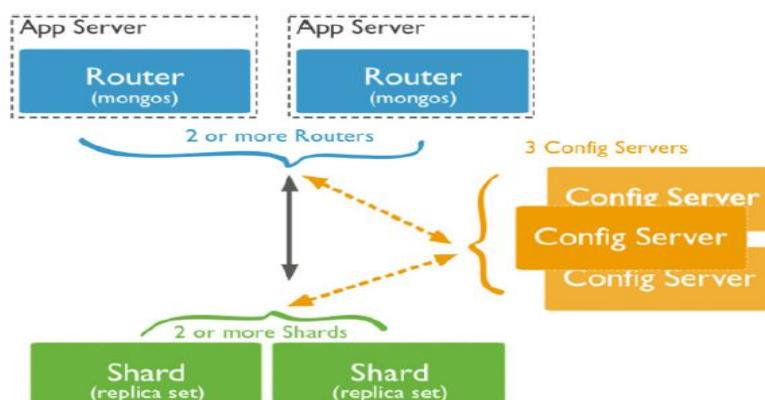


איור 7 תהליכי השכפול נתונים ב MongoDB

### 4.1.4 צביר פיזור

שיטה זו מפזרת את הנתונים בין מספר צבירים שכפול. דרוש מסדר נתונים אשר ינהל את פיזור הנתונים בצורה ייעילה על ידי מפתח ברמה של מסמך ולא ברמה של שורה בודדת בטבלה. לצורך כך נוצרת דרישת לשתי ישוות חדשות, שרת **קונפיגורציה**: המגדר היקן נמצא כל פריט.

ישות שאחריות על ניתוב בקשות של אפליקציות אל הצביר ובחזרה, لكن אפליקציה לא צריכה לנהל ניתוב של בקשות .



<sup>13</sup> <https://docs.mongodb.com/manual/replication>

## איור 8 מרכיבי ארכיטקטורת בסיס נתונים MongoDB

### Cassandra 4.2

בפרויקט Cassandra, Apache Foundation מונח זה תואם את האפשרות המוצעת על ידי ארכיטקטורת Cassandra כדי להוסיף מכונות חדשות הנקראות צמתים (Nodes).<sup>14</sup>

בתחילת Cassandra פותחה באופן פנימי על ידי פייסבוק לצרכים של ההודעות הפנימיות שלה. השימוש של Cassandra היה מוגבל לדרישות של הودעות פייסבוק. כתוצאה לכך תוכנות רבות של Cassandra לא יושמו מזמן, מכיוון שהם אינם נחוצים. ב-2008 פייסבוק החליטה להציגו לקרן אפקטיבי את Cassandra. הפרויקט נשאר שנתיים באינקובטור לפני שהפרק לפרויקט ברמה עליונה "של קרן אפקטיבי עד 2010".<sup>14</sup> Twitter, Netflix, Spotify, eBay ו-Cassandra הם השחקנים הגדולים של האינטרנט, כגון Facebook, Netflix, Spotify, eBay ו-Cassandra.

#### 4.2.1 תוכנות עיקריות

התוכנות העיקריות של מסד הנתונים NoSQL Cassandra מפורטים בסעיף זה.

- **עמירות בפני תקלות:** הנתונים של צומת (צומת הוא מופיע של Cassandra) משוכפים באופן אוטומטי לצמתים אחרים (מכונות שונות). לכן, אם הצומת אינה בשירות, הנתונים הנוכחיים זמינים דרך צמתים אחרים. המונח של שכפול מתייחס למספר הצמתים שבהם הנתונים משוכפים. יתר על כן, ארכיטקטורת Cassandra מגדירה את המונח אשכול קבוצה של לפחות שני צמתים(Nodes) ומרכז נתונים (Data center) כאשכולות delocalized (Data center) Cassandra מאפשרת שכפול בין מרכזי נתונים שונים (Data Centers). צמתים שנפלו ניתן להחליף ללא זמינות השירות.
- **מבודרת:** באשכול כל הצמתים שוויים. אין מושג של **שליט (master)**, **עבד (Slave)**, וגם אין תהיליך שיהיה חייב לניהל צואර בקבוק בחלק הרשות.
- **מודל נתונים עשיר:** מודל הנתונים המוצע על ידי Cassandra מבוסס על הרעיון של מפתח / ערך מאפשר לפתח מקרים רבים של שימוש בעולם האינטרנט.
- **אלסטי:** היא ליניארית, קצב הכתיבה והקריאה עולה באופן ליניארי כאשר שרת חדש נוסף לאשכול כמו כן Cassandra מבטיחה שלא תהיה השבתה של המערכת של ישומים.

#### 4.2.2 מודל נתונים

בישומים של העולם של האינטרנט, פייסבוק למשל, הנתונים אינם מתוארים באותו אופן, השימוש במאג'רי מידע יחסים אינו תמיד הולם. למעשה זה יוביל אותנו לאחסון נתונים עם ערכיSSH רבים.

יתר על כן, במסד נתונים יחסים מסורתיים, נהוג לנормל את הנתונים כדי למנוע יתרות נתונים וביעות המובנות הקשורות לעדכוןם. הנתונים מובנים בדרך מסוימת, מה שאורם לקריאה לעבור דרך של צירופים (JOIN) כדי לקבל את התוצאות הרצויות. בכלל אלו המבוזר של Cassandra על כמה מכונות, הוא לא מאפשר את זה. הרעיון של איחוד בטלאות בכמה מכונות לא הגיוני. במקרה זה Cassandra נדרש מעתנו כמה שיותר נתונים לא נורמליים (Denormalization).

לפיכך, חלק של העבודה על הנתונים יעשה במהלך שלב הכתיבה. לשם כך, הנתונים מקובצים למשפחות של עדות (columns families), אשר מבוססים על Big Table של Google Cassandra. מאחסן את הנתונים ברכף

וממונן לפי זוגות מפתח / ערך. קר מסויוגת Cassandra כמודד נתונים, מבליל להגיא עד לאחסון פיזי בעמודה (column oriented).

שם
ערך
חוותמת זמן

Keulkeul:300
Contact
email:bar@...
login:keulkeul
tel:+33549

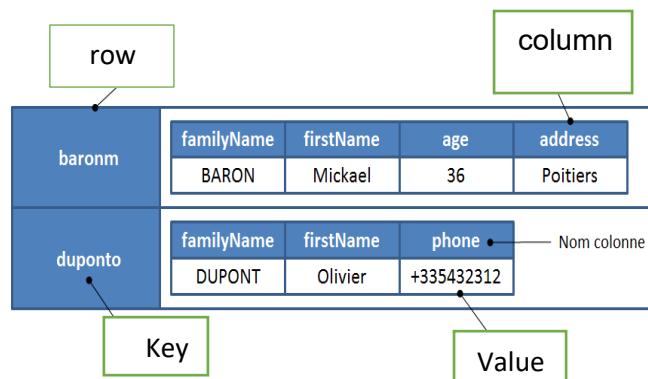
עמודה היא היחידה הקטנה ביותר במודול הנתונים של Cassandra זו שלישיה המכילה שם, ערך וחותמת זמן (האחרון משמש לקביעת העדכון האחרון). הגודל של השם יכול להכיל עד 64 KB הערך יכול להכיל 2 GB של נתונים.

הערך אינו נדרש. מחייבתו עלולה להוביל לביצועים משופרים. שם העמודה יכול להיחס כערך. לדוגמה באירור של להלן, שם העמודה מכילה את שם המשתמש ואת הציון. תיבת הערך יכולה להכיל מספר ערכים למשל אוסף של מחרוזות.

#### 4.2.2.2 רשותה

רשומה מורכבת מאוסף של עמודות, מזוינה על ידי מפתח, מפתח יכול להכיל עד 64 KB של נתונים והוא יכול להוכיח עד שני מיליארד עמודות, ניתן להשתמש בעמודות כמפתח הראשי. בתרשים להלן ניתן לראות דוגמא של שתי שורות .

- (1) מספר העמודות אינם זהה.
- (2) העמודות אינן בהכרח אותן הדבר.



איור 9 דוגמה לריבבי רשומות אשר מכילות נתונים.

#### : (Column Family)

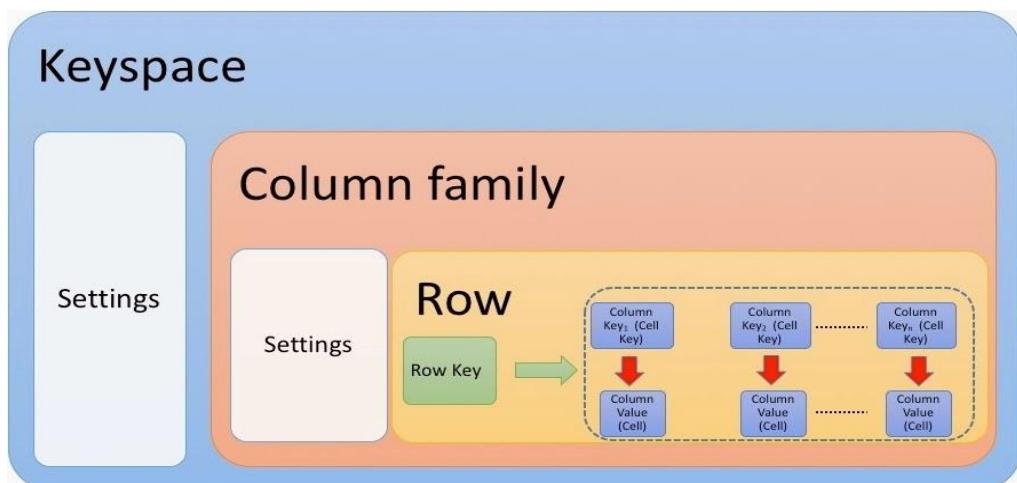
משפחה של עמודות או באנגלית column families היא קיבוץ לוגי של רשומות. במקביל עם העולם של מסדי נתונים ייחסימים, משפחה של עמודות היא סוג של טבלה. בעת הגדרת משפחת עמודות, תוכל להוציא מידע על מטא-דטה עברו עמודות. ניתן להשתמש במידע זה כדי להזין את השם והסוג של העמודות. עם זאת, זה בזמן של הוספה שורה שתבחרו אילו עמודות לנצל.

Persons				
familyName	firstName	age	address	
	BARON	Mickael	36	Poitiers
familyName	firstName	phone		
	DUPONT	Olivier	+335432312	

**איור 10 משפחה של עמודות.**

- סטטי: עדויות מוגדרות בעת יצירה או שינוי של משפחת העמדות.
  - דינמי: עדויות מוגדרות בעת יצירה או עריכה של שורה.

**Key space** : מכיל את טורי המשפחות בדיקן כמו בסיס נתונם מכיל טבלאות בעולם יחס'ים, משמשים בהם כדי לקווצת טורי משפחות ביחיד. במסדר הנתונים המסורתי יחס'ים האנלוגיה של key spaces ניתן לראותו כסכם מסדר נתונים.



## *איור 11 תיאור גרפי של Keyspace*

HBase 4.3

HBase הוא בסיס נתונים מבוצר מכון [עמודות](#), הפעול על Apache Hadoop. התוכן של עמודות HBase יכול לשמש כקלט ופלט עבור נתונים בשימוש שנוצר על ידי מנגנון [MapReduce](#) Hadoop .  
ישום קוד פתוח של המודול BigTable שפורסם על ידי [Google](#).<sup>15</sup>

HBase הוא מאגר נתונים **מוכווני عمودה**, כלומר, הוא מ אחסן נתונים לפי עמודות ולא לפי שורות. ב HBase אם אין נתונים ל משפט העמודה נתונה, הוא פשוט לא מ אחסן שם דבר בכלל, לעומת זאת מ סדי נתונים יחסיים אשר ח'יבים לאחסן ערכיו זהב. בנוסף, בעת אחזור נתונים ב- HBase, علينا לבקש רק את מ שפות העמודות הספציפיות הדרשיות לנו, כי יש מיליון טורים בשורה מסוימת, אנו צריכים לוודא שנחנו שואלים רק עבור הolumns שאנו ראמת אריכים

כמו ב HDFS, הארכיטקטורה HBase עוקב אחר המודל המסורתי Master/Slave שבו יש Master אשר לוחץ החלטות ו אחד או יותר אשר עושה את המשימה האמתית. ב HBase, המאסטר נקרא [H\\_Master](#)<sup>16</sup> ועבדים נקראים [RegionServer](#)

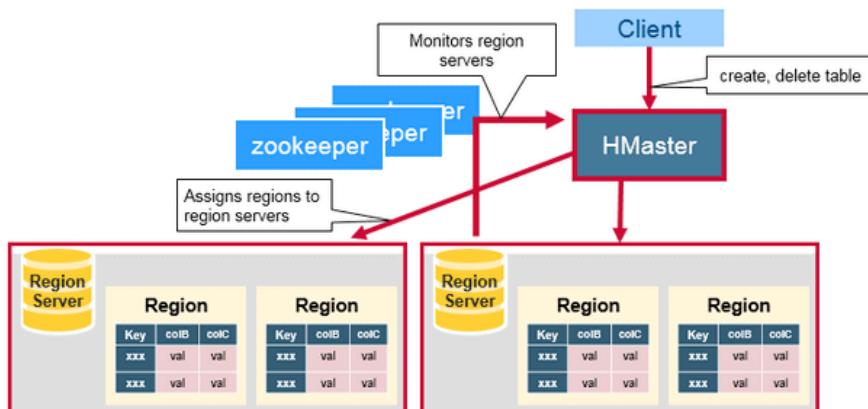
ארכיטקטורת הרכיבים:



איור 12 רכיבי הארכיטקטורה Hbase .

כפי שניתן לראות מהדיאגרמה שלעיל, בדרך כלל, לאשכול HBase יש צומת מאסטר אחד, הנקרא **H\_Master** ושרותי **HRegionServer**. כל שרת איזור מכיל איזוריים רבים הנקראים **HREGIONS**. גם איזוריים רבים הנקראים **ZooKeeper** כמו שירות תייאום מבוצרת כדי לשומר על מצב השירות באשכול. הנטונים ב-HBase משתמשים בטבלאות והטבלאות מאוחסנות באזוריים. כאשר טבלה גדולה מדי, הטבלה מחולקת למוחוזות רבים איזוריים אלה מוקצים לשרתים איזוריים ברחבי האשכול כל שרת איזור מארח בערך אותו מספר איזוריים.

### HBase H\_Master

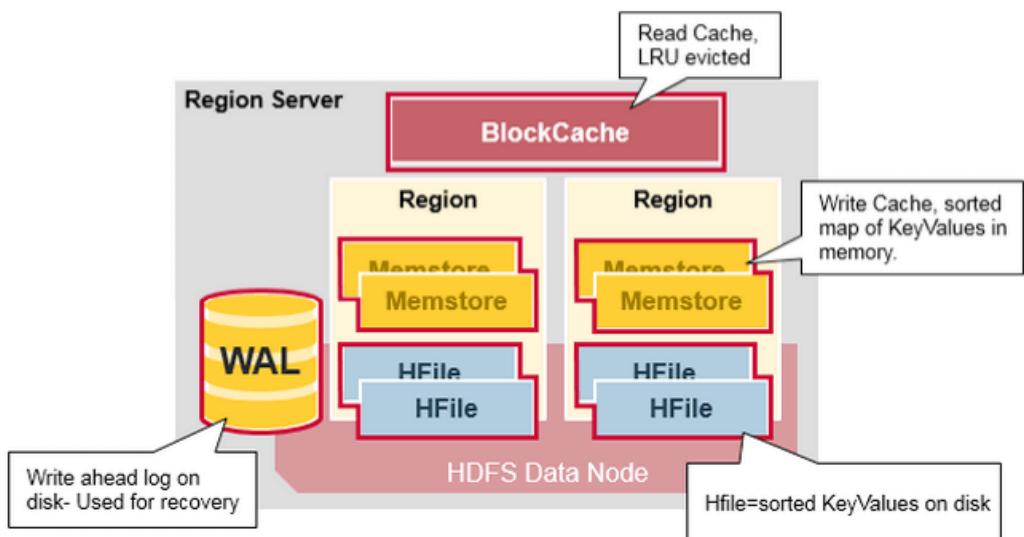


איור 13 רכיב ה HMaster מאשכול

HMaster הוא יישום של שרת ראשי הקצתת אזור, DDL (יצירה, מחיקת טבלאות) פועלות שטוטופלות על ידי HBase Master. שרת מסטר אחראי על ניהול כל המפעלים של RegionServer באשכול והוא משקע עבור כל השינויים במתה דתא באשכול מבוזר, המאסטר בדרך כלל פועל על שם Name Node.

### HBase RegionServer

HRegionServer הוא יישום של ה RegionServer הוא אחראי על השירות וניהול אזוריים באשכול מבוזר, פועל על Node Region Data אשר מוקצים לצמתים באשכול, שנקרא RegionServer, אלא מרשותם נתונים לקרוא וכתיבה שרת אזורי יכול לשרת כ-1,000 אזוריים.



איור 14 רכיבי Region Server

פועל על צומת נתונים HDFS ויש לו את הרכיבים הבאים:

- WAL: Write Ahead Log (יומן כתיבה קדימה) הוא קובץ במערכת הקבצים המבוזר WAL משמש לאחסון נתונים חדשים שעדיין לא התמידו לאחסון קבוע הוא משמש לשחזור נתונים במקורה של CISLON.
- Block Cache
- MemStore: הוא המת�כו קרייה הוא מאחסן נתונים הנקרים לעיתים קרובות בזיכרון לנוחים שלא שונשה בהם שימוש לאחרונה יגורשו כאשר הם מלאים.
- CF: הוא מטען כתיבה הוא מאחסן נתונים חדשים שטרם נכתבו לדיסק הוא ממוקם לפני כתיבה לדיסק יש MemStore אחד לכל משפחת טור לכל אזור.
- Hfiles: Hfiles אחסן את השורות כפי שמופיעין ה KeyValues על הדיסק.

כאשר Region Server מקבל את הבקשה, הוא מפנה את הבקשה לאזור (Region) ספציפי כל אזור מאחסן סט של שירותים ניטן להפריד את נתונים השירותים טורים רבים (CFs) הנתונים של CF מסוים מאחסן ב HStore אשר מורכב MemStore ומקבוצה של HFiles.

כל שרת אזור מכיל יומן כתיבה קדימה (הנקרא HLog או WAL) ואזוריים רבים כל אזור מורכב MemStore מרובים (HFile) הנתונים, חיים ב StoreFiles בקרה של משפחת טור (מוסבר להלן) ה- HStore מכיל שינויים בזיכרון של ה Store (נתונים).

המיופיע של אזוריים לאזור שרת נשמר בטבלה בשם META. כאשר מנוטים לקרוא או לכתוב נתונים מ- HBase הלוקה קורא את המידע מטבלת META ומתקשר שירות עם שרת האזור המתאים.

כל RegionServer מוסיף עדכונים (Puts, Deletes) ליאמן הרישום מראש (WAL) שלו, ולאחר מכן ל- MemStore עבור החנות המשופעת לא-WAL, יש את האפשרות של אובדן נתונים במקרה של כשל ב RegionServer לפני RegionServer חדש יושם על MemStore סמוך ו- StoreFiles HLog הוא יושם על WAL, ויש מופיע HLog אחד לכל RegionServer.



## 5 מנועי חיפוש

### Apache Solr 5.1

Apache Solr היא פלטפורמת חיפוש עסקית המבוססת על המנוע אינדוקס [Lucene](#)<sup>17</sup>. היכולות של Hadoop כוללות כתעת אינדוקס Solr כחלק מהחברה, שכן ניתן להשתמש בה- Solr ביעילות רבה בתנאים בה- Hadoop. Solr יש יכולות התאימה עצומות בין הרצוי למצוי, כולל ביטויים, תווים כללים, צירופים של מילים, קיבוץ של מילים, וכו'. עבור כל סוג הנתונים, Solr תומך באינדוקס של קבצים בפורמטים שונים מוכנים לשימוש בה- XML, JSON ו- CSV.

Solr מכיל גם תוספים אופציונליים לאינדוקס של תוכן עשיר (כגון PDF ועוד), זיהוי שפה, תוצאות חיפוש וכו'. Solr מספק גם מספר רב של אלגוריתמים הנקראים "פיטרים/יבטים" כדי לפצל ולהלך את התוצאות למספר קטגוריות. הוא מראה את המספרים עבור כל אחד. Solr מאפשר למשתמש "לחקור" את תוצאות החיפוש על פי היבטים אלה. Solr מספק גם תכונות מתקדמות כגון השלמה אוטומטית, איות ועוד.



### Sphinx search 5.2

חיפוש ספינקס הוא מנוע החיפוש שירות בקוד פתוח. בוגר [Solr](#) או [Elasticsearch](#) הוא משתמש במנוע האינדוקס שלו שנכתב בה- C +++. הספינקס מתגאה במספר רב של ביצועים ומדריגות. הוא מספק פונקציונליות דומה מ Solr ו- Elasticsearch. עם זאת, הוא דורש מוד נתונים אחר כדי לאחזר את התוכן בפועל. חיפוש ספינקס מספק זמינות גבוהה ופתרון של ניטור עם זאת, ל- Sphinx Search אין עדין אפשרות שילוב עם כלים ניהול משאים כגון YARN. עבור אבטחה, ONION Onion נתן להשתמש עם חיפוש ספינקס. כמו אתרים משתמשים ביום בספינקס: Craigslist, Melty, Pirate Bay, Netlog. האתר חילופי [Search technologies in cloud](#) 5.3

חיפוש כשירות בענן היה פופולרי כ Software as a service אשר מספק על ידי ספק ענן שונים לאחרונה. התכונות האופייניות של הצעה זו הם סביבה מחזור החצים כלו של פתרון חיפוש המקיים העלאת תוכן / מסמכים, אינדוקס, משקל משתמש להטאמה אישית לחיפוש על סמך מילוט מפתח. רובם גם תומכים במגוון שפות תכנות כמו JAVA, Python, .Net, Python וכו', כך ארגונים יכולים להשתמש במאגרי המשאבם הקיימים שלהם כדי למנף את השירותים הללו.

<http://lucene.apache.org/solr/features.html><sup>17</sup>  
<http://sphinxsearch.com/about/sphinx><sup>18</sup>

הספקים הפופולריים בתחום זה הם Elasticsearch, Amazon Cloud Search, Microsoft Azure Search (מיוחד על ידי חברת ELASTIC).



## elasticsearch

## ELASTIC SEARCH 6

### 6.1 Elasticsearch תכונות כלליות

התכונות הכלליות של Elasticsearch הן:

- Elasticsearch הוא מדרגי עד Petabytes של נתונים מבוקרים ולא מבוקנים.
- Elasticsearch יכול לשמש כתחליף של חניות מסוימות כמו MongoDB ו-RavenDB.
- Elasticsearch משתמש בדנורמליזציה (שינוי הישויות במסד נתונים שהוא במצב הרגיל: למשל מיזוג/פדרה של טבלאות) כדי לשפר את ביצועי החיפוש.
- Elasticsearch הוא אחד מנوعי החיפוש הפופולריים של הארגון, אשר כוונן בשימוש על ידי ארגונים גדולים רבים כמו ויקיפדיה, StackOverflow, GitHub, The Guardian ו-Co.
- Elasticsearch קוד פתוח זמין תחת רישיון גירסה 2.0.

### 6.2 מושגים מרכזיים ב Elasticsearch

#### • NODE (узם):

צומת הוא שירות יחיד המהווה חלק מהאשכול המחשבים, אחסן את הנתונים שלכם ומשתתףVIC ביכולות ייצור האינדקס והחיפוש של האשכולות. בדיקן כמו אשכול, צומת מזוהה על ידי שם אשר כבירה מחדל הוא מזהה אוניברסלי ייחודי אוניברסלי (UUID: Universally Unique Identifier) המוקצה לצומת בעת הפעלה. שם זה חשוב למטרות ניהול או זיהוי השירותים ברשת מחשבים שתואימים לצמתים באשכול Elasticsearch שלהם.

#### • Clusters (אשכול מחשבים):

אשכול הוא אוסף של צומת אחת או יותר (שירותים) שמחזקים יחד את כל הנתונים שלכם ומספקים יכולות אינדקס וחיפוש מאוחדות בכל הצמתים. אשכול מזוהה על ידי שם ייחודי אשר כבירה מחדל הוא "Elasticsearch".

#### • SHARD (שבר):

היא מחיצה (Horizontal Partition) של מידע של בסיס נתונים או מנוע חיפוש, כל מחיצה נקראת שבר (SHARD) כאשר כל שבר מנוהל ומתוחזק על מופיע בסיס נתונים נפרד כדי לאפשר עומס. האינדקסים מחולקים אופקי לשברים. משמעות הדבר היא שכל שבר מכיל את כל המאפיינים של המסמך, אך מכיל פחות מספר אובייקטים של JSON מאשר אינדקס. ההפרדה האופקית הופכת את השבר (SHARD) לצומת (NODE) עצמאית, אשר ניתן לאחסן בכל צומת. שבר ראשי הוא החלק המקורי של אינדקס ולאחר מכן, שברי אלה העיקריים משוכפלים בעותקים של שבר.

#### • INDEX (אינדקס):

האינדקס הוא אוסף של מסמכים בעלי מאפיינים דומים. אינדקס משתמש גם בReLUIN של שברים (SHARDS) כדי לשפר את הביצועים. לדוגמה, אפשרות לקבל אינדקס עבור נתוני לקוחות, אינדקס נוסף לקטלוג מוצרים, וכן אינדקס נוסף עבור נתונים. אינדקס מזוהה על ידי שם (זה חייב להיות בעוטיות קטנות) ושם זה משמש להתייחסות לאינדקס בעות ביצוע פעולות לאינדקס, חיפוש, עדכון ומחיקה נגד המסמכים שבתוכו. באשכול יחיד, אפשרותה להגדיר כמה שיותר אינדקסים.

- **SYPE(טיפוס):**  
בתוך אינדקס, באפשרותכם להגיד טיפוס אחד או יותר. טיפוס הוא קטגוריה לוגית / מחיצה של האינדקס שלכם אשר שם שלא היא למגרי תלי בכם.  
באופן כללי, סוג מוגדר עבור מסמכים בעלי קבוצת שדות מסווגים.  
לדוגמא, נניח שאתה מפעיל פלטפורמת בלוגים ומאתחן את כל הנתונים שלכם באינדקס יחיד.  
באינדקס זה, ניתן להגיד סוג עבור נתוני משתמש, סוג אחר עבור נתוני הבלוג, ועוד אחר עבור נתוני העורות.

- **DOCUMENTS (מסמך):**  
מסמך הוא יחידה בסיסית של מידע שנitin לצרף לאינדקס.  
לדוגמא, ניתן לקבל מסマー עבור לקוח ייחיד, מסマー אחר עבור מוצר בודד, ועוד מסマー עבור הזמנה אחת. זהו אוסף של שדות בפורמט JSON (JavaScript Object Notation) -שהוא פורמט נתונים אינטרנט בכל מקום. בתוך אינדקס / סוג (type), אפשר לאחסן מסמכים רבים.  
צריך לשים לב כי למרות שמסマー שוכן פיזית בתוך אינדקס, המסマー צריך להיות למעשה באינדקס / להקצת טיפוס בתוך אינדקס.  
כל מסマー שיר לטעיפוס בתוך אינדקס. כל מסマー משיר למזהה ייחודי, הנקרא SID.

- **REPLICAS (עותקים משוכפלים):**  
Elasticsearch מאפשר למשתמש ליצור עותקים משוכפלים של האינדקסים שלהם ואת השברים. שכפול לא רק עוזר להגדיל את זמינות הנתונים במקורה במקרה של כשל, אלא גם מאפשר את הביצועים של חיפוש על ידי חיפוש מקבלי של העותקים המשוכפלים האלה.

## 6.3 יתרונות של Elasticsearch

- Elasticsearch מפותחת על Java מה שהופך אותו תואם כמעט בכל פלטפורמה.
- Elasticsearch הוא כלי שעובד בזמן אמת, במילימ"ש לאחר הוספה המסマー לtower הבסיס נתונים לאחר שנייה אחת את המסマー ניתן לחפש במנוע זה.
- Elasticsearch מבזער, מה שהופך אותו קל בקנה מידת וולשבל אותו בכל ארגון גדול.
- יצירת גיבויים מלאים קלים באמצעות המושג של שער(gateway), אשר קיים ב-Elasticsearch.
- Elasticsearch משתמש באובייקטים של JSON כתגובה, מה שמאפשר להפעיל שירות Elasticsearch עם מספר רב של שפות תכנות שונות.
- Elasticsearch תומך כמעט בכל סוג מסマー למעט אלה שאינן תומכין עיבוד טקסט.

## 6.4 חסרונו של Elasticsearch

- לא- Elasticsearch אין תמיכה רבת-לשונית במונחים של טיפולسائلות בקשה ונתונים (אפשרי רק ב-JSON) בלבד לא- Apache Solr, שם ניתן בפורמטים CSV, XML או JSON.
- Elasticsearch יש גם בעיה של מצב פיצול המוח, אבל במקרים נדירים.

## 6.5 השוואה בין RDBMS ו Elasticsearch

ב Elasticsearch הוא אוסף של טיפוסים שונים בדיקן כמו בסיס נתונים הוא אוסף של טבלאות RDBMS מערכת ניהול נתונים. (כל טבלה הוא אוסף של שורות בדיקן כמו כל מיפוי ב-RDBMS הוא אוסף של אובייקטים)

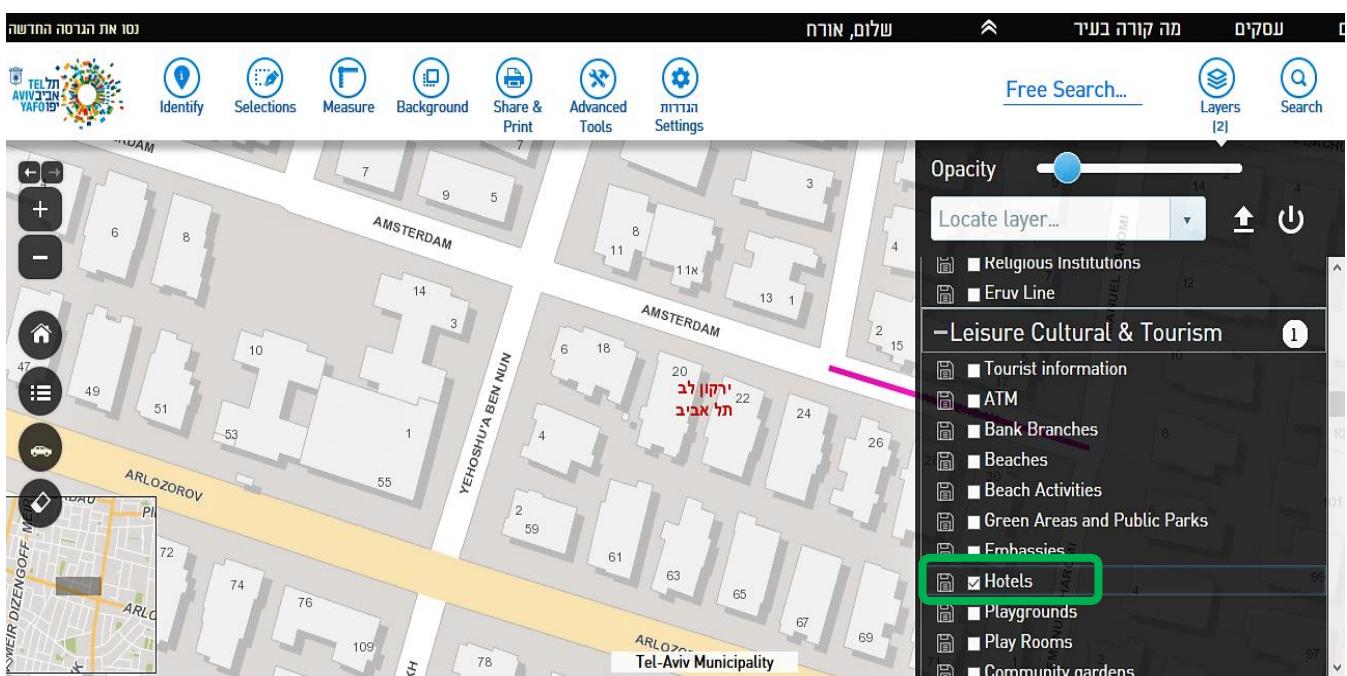
Elasticsearch	RDBMS
Index	Database
Shard	<b>Shard</b>
Mapping	<b>Table</b>
Field	<b>Field</b>
JSON Object	<b>Tuple</b>

## 7 מתודולוגית הסימולציה עם ELASTICSEARCH/KIBANA

לפני ההסבר על המתודולוגיה הסימולציה יש צורך להורד את Elasticsearch דרך אתר <https://www.elastic.co> ולחזור אליו. המתודולוגיה מבוססת על הורדת נתונים מאינטראנט של עיריית ת"א הקשורים לנקיות WIFI, בתים מלון, בנקים, בתים מrankhet ועוד, נקודות אלה נמצאים באזורי תל אביב. לאחר ההורדה של כל הנתונים אשר הזכרתי היה צריך להסב את הפורתט הנתונים של האתר Excel ל CSV ומפורט CSV היה צריך להסב לפורתט JSON ולאחר ההסבה היה צריך ליצור אינדוקס של הנתונים ב Elasticsearch. <http://www.convertcsv.com> הינו האתר אשר השתמשתי בו להסבה מ CSV ל JSON הינו:

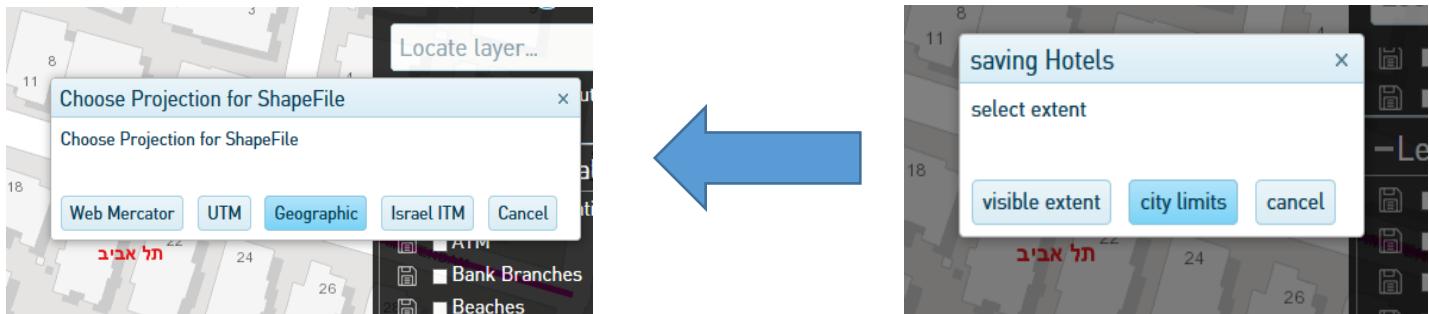
האתר של עיריית ת"א הוא : <https://gisn.tel-aviv.gov.il>.

לדוגמה נבחר מטור הרשימה את הנתונים של בתים מלון:



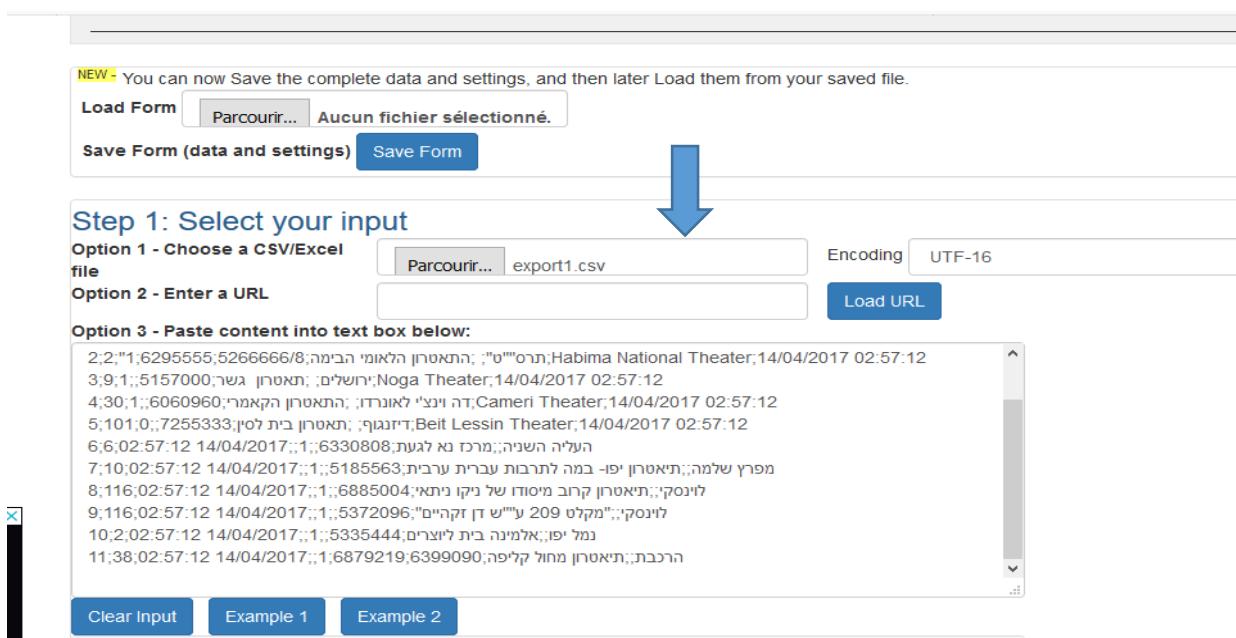
איור 15 אתר של עיריית ת"א שמיימנו תוכלן להורד נתונים

לאחר מכן נבחר את האופציה תיכון העיר ואת האופציה הגאוגרפית :



איור 16 בחרת סוג של נתונים

מתוך התיקייה שהורדתם בחרו את קובץ הנתונים בפורמט CSV ושמרו אותו כפורמט JSON במחשב, לאחר מכן יש להיכנס לאתר <http://www.convertcsv.com> בצד להפוך אותו ל JSON.



איור 17 אתר שבו תוכל לhmaיר את הקובץ לפורט JSON

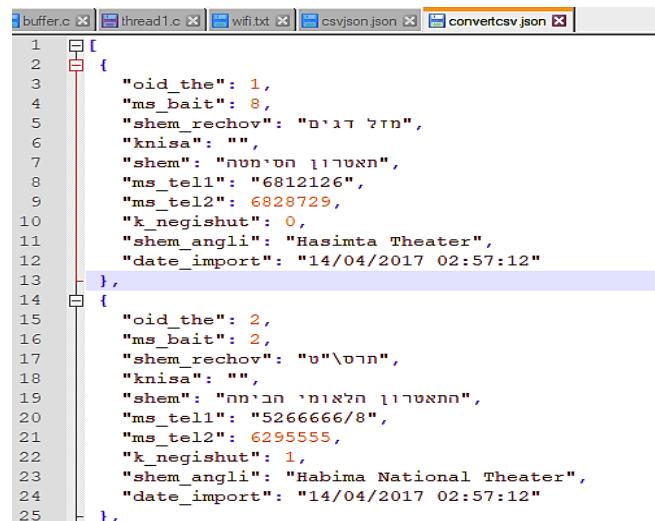
לאחר טיענת הקובץ יש לבחור מתוך האופציות את האופציה OUTPUT, בתוך הטבלה יש לסמך את כל העמודות שנמצאים בצד שמאל אשר ימשכו כפתחות בפורט JSON.

Col #	Field	Trim Left <input checked="" type="checkbox"/>
1	oid_the	<input checked="" type="checkbox"/>
2	ms_bait	<input checked="" type="checkbox"/>
3	shem_rechov	<input checked="" type="checkbox"/>
4	knisa	<input checked="" type="checkbox"/>
5	shem	<input checked="" type="checkbox"/>
6	ms_tel1	<input checked="" type="checkbox"/>
7	ms_tel2	<input checked="" type="checkbox"/>
8	k_negishut	<input checked="" type="checkbox"/>
9	shem_angli	<input checked="" type="checkbox"/>



Step 2: Choose input options  
Step 3: Choose output options

בשלב 5 יש להוריד את קובץ JSON בכדי להשתמש בו ב Elasticsearch (פרק 7.1.6). יופיעו דוגמאות של נתנים מייערת ת"א :



```

1 buffer.c x 2 thread1.c x 3 wifi.txt x 4 csvjson.json x 5 convertcsv.json x
1
2 {
3     "oid_the": 1,
4     "ms_bait": 8,
5     "shem_rechov": "רחוב דיזנגוף",
6     "knisa": "",
7     "shem": "תיאטרון הסימטה",
8     "ms_tel1": "6812126",
9     "ms_tel2": 6828729,
10    "k_negishut": 0,
11    "shem_angli": "Hasimta Theater",
12    "date_import": "14/04/2017 02:57:12"
13 },
14 {
15     "oid_the": 2,
16     "ms_bait": 2,
17     "shem_rechov": "חדרה",
18     "knisa": "",
19     "shem": "התיאטרון הלאומי הבינלאומי",
20     "ms_tel1": "5266666/8",
21     "ms_tel2": 6295555,
22     "k_negishut": 1,
23     "shem_angli": "Habima National Theater",
24     "date_import": "14/04/2017 02:57:12"
25 }

```

### Step 5: Generate output

Choose Conversion Type:

CSV To JSON

CSV To Keyed JSON

CSV To JSON Array

CSV To JSON Comma

Result Data:

```
[
  {
    "oid_the": 1,
    "ms_bait": 8,
    "shem_rechov": "רחוב דיזנגוף",
    "knisa": "",
    "shem": "תיאטרון הסימטה",
    "ms_tel1": "6812126",
    "ms_tel2": 6828729,
    "k_negishut": 0,
    "shem_angli": "Hasimta Theater",
    "date_import": "14/04/2017 02:57:12"
  },
  {
    "oid_the": 2,
    "ms_bait": 2,
    "shem_rechov": "חדרה",
    "knisa": "",
    "shem": "התיאטרון הלאומי הבינלאומי",
    "ms_tel1": "5266666/8",
    "ms_tel2": 6295555,
    "k_negishut": 1,
    "shem_angli": "Habima National Theater",
    "date_import": "14/04/2017 02:57:12"
  }
]
```

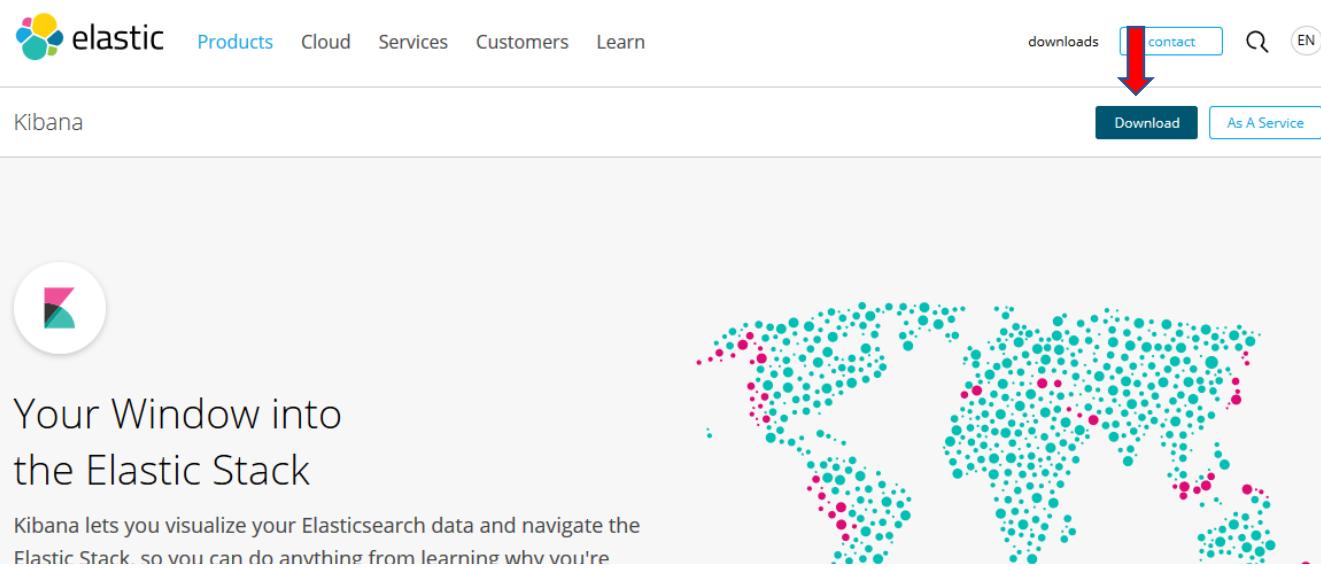
Save your result: convertcsv.json

Download Result

EOL: CRLF

איור 19 הורדת הקובץ הJSON

בהמשך בכדי להציג את הנתונים ע"י ממשק הורידו את התוכנה Kibana מהאתר: <https://www.elastic.co> KIBANA מספקת לטיומלוצית ניתוח הנתונים ממשק גרפי עם ריבוי הדמיות מגוונות ופונקציונליות רבות כדי לחפש נתונים בצורה יותר אינטראקטיבית ותוכלו גם לרשום את השאלות שלכם. כל הפונקציות הללו מנצלות את יכולות הצבריה (הగרציה) המוקפות של Elasticsearch. דרך תוכלו להציג את מקום בית מלאן על מפה ושירותיהם באמצעות תרשימים עוגה, היסטוגרמה של תקלות של נקודות WIFI המפוזרות בכל תל אביב, להציג שירותי של נתונים, ועוד. לאחר מכן Kibana הותקנו בהצלחה, תוכלו לגשת למשק הגרפי על ידי פיתוח דףדף בכתובת <http://localhost:5601>.



איור 20 אתר ה Elasticsearch

יש לבחור את גירסה המתאימה למחשב שלכם ולקוב אחריו שלבי התקינה:

### Installation Steps

New around here? View our [getting started page](#) to get acquainted.

- 1** Download and unzip Kibana
- 2** Open config/kibana.yml in an editor  
• Set `elasticsearch.url` to point at your Elasticsearch instance
- 3** Run `bin/kibana` (or `bin\kibana.bat` on Windows)

### Download Kibana

Want to upgrade? We'll give you a hand. [Migration Guide »](#)

Version:	6.3.0
Release date:	June 13, 2018
License:	<a href="#">Elastic License</a>
Downloads:	<a href="#">WINDOWS sha</a> <a href="#">MAC sha</a> <a href="#">LINUX 64-BIT sha</a> <a href="#">RPM 64-BIT sha</a> <a href="#">DEB 64-BIT sha</a>

#### איור 21 הורדת ה-Kibana

לאחר שניכסתם דרך הדףן לכתובת <http://localhost:5601>, כנסו לחילון DEVTOOL שם תוכלו לכתוב את השאלות כפי שמוצג בפרק הבא.

בנוסף לזה, אם יש צורך, יש להשתמש באתר <https://boundingbox.klokantech.com> אשר מספק קואורדינטות של מרחב גיאוגרפי של תל אביב בפורמט JSON GEOJSON. באתר זה השתמשתי בכדי להוריד קואורדינטות של אזורי Elasticsearch בת"א לצורך חיפוש גיאוגרפי של נתונים בתוך מאגר.

אתה זה מאפשר להשתמש בתחוםים גיאוגרפיים של "תיבה-תוcharת" ושל מצולעים במפה. באמצעותם, נוכל לבצע חיפושים הגיאוגרפיים ב-Elasticsearch. בהמשך הפרק של שאלות Elasticsearch יופיעו דוגמאות של חיפושים נתונים באמצעות התcona הגיאוגרפית שלהם.(פרק 7.4.2)

## 7.1 אקלס בטור ELASTICSEARCH 7.1.1 Create Index API

האינדקס API מוסיף או מעדכן מסמך JSON שהוקל באינדקס מסוים, מה שהופך אותו לבר חיפוש. הדוגמא הבאה מכניסה את מסמך JSON ל-אינדקס "Facebook", תחת סוג שנקרא "post" עם מזהה של 1:  
התוצאה של פעולה האינדקס הנ"ל הינה:

Console

```

1
2
3 PUT facebook/post/1
4 {
5   "user": "oshrat",
6   "post_date": "2009-11-15T14:12:12",
7   "Message": "This is a beautiful day!!!"
8 }
9
10
11
12
13

```

```

1 [
2   "_index": "facebook",
3   "_type": "post",
4   "_id": "1",
5   "_version": 1,
6   "result": "created",
7   "_shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  },
12  "created": true
13 ]

```

#### איור 22 הכנס אינדקס לטור המאגר של Elasticsearch

שדה `shards` מספקת מידע על תהליכי השכפול של פעולות האינדקס.

- Total - מציין את מספר עותקי שברים (শব্রিম) (ראשוניים ועוטקיים), כדי לבצע את פעולה האינדקס.
- Successful - מציין את מספר עותקי השברים שבهم הפעלת האינדקס הצלחה.
- Failed - מערך המכיל שגיאות הקשורות לשכפול במקורה של פעולה אינדקס נכשל בעותקי שבר(Shard).
- פעולות האינדקס היא מוצלחת במקרה מוצלח הוא לפחות 1.
- דיהוי אוטומטי - ניתן לבצע את פעולה האינדקס מבלי לציין את מספר המזהה. במקרה זה, מזהה יישור באופן אוטומטי. התוצאה של פעולה האינדקס הנ"ל הינה:

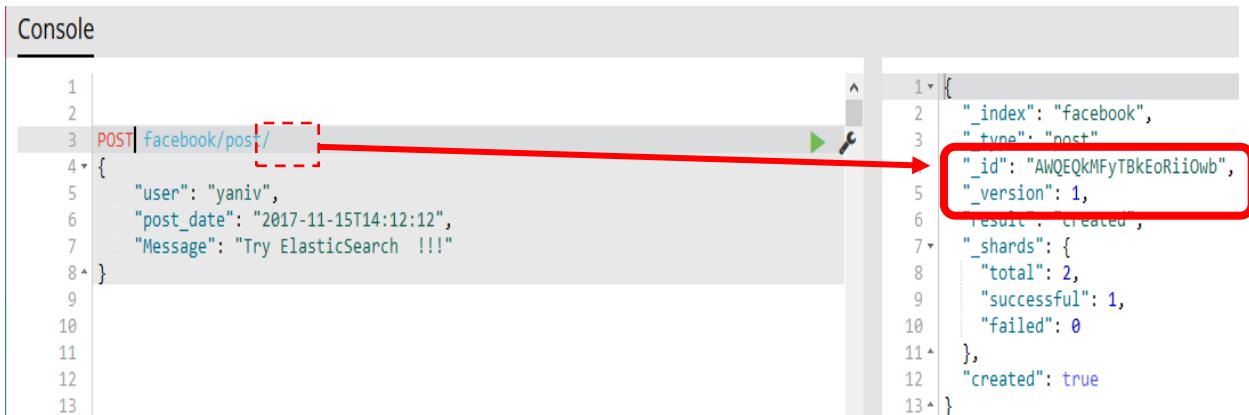
Console

```

1
2
3 POST | facebook/post/
4 {
5   "user": "yaniv",
6   "post_date": "2017-11-15T14:12:12",
7   "Message": "Try ElasticSearch !!!"
8 }
9
10
11
12
13

```

facebook/post/



```

1 {
2   "_index": "facebook",
3   "_type": "post",
4   "_id": "AWQE0kMFyTBkEoRiiOwb",
5   "_version": 1,
6   "_score": 1,
7   "_source": {
8     "user": "yaniv",
9     "post_date": "2017-11-15T14:12:12",
10    "Message": "Try ElasticSearch !!!"
11  }
12 }
13

```

אIOR 23 מספר ה `ID` של אינדקס אחריו שהוא נוצר

## Get API 7.1.2

הבא מחפשים מסמך JSON מאינדקס שנקרא `facebook`, תחת סוג הנקרא `post`, עם ערך מזהה 1 :

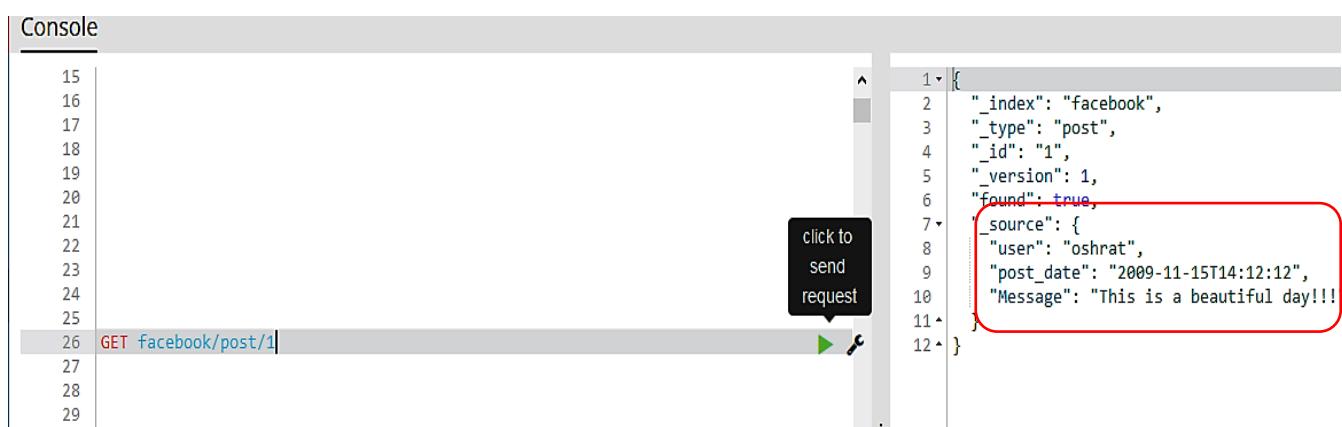
Console

```

15
16
17
18
19
20
21
22
23
24
25
26 GET | facebook/post/1
27
28
29

```

click to send request



```

1 {
2   "_index": "facebook",
3   "_type": "post",
4   "_id": "1",
5   "_version": 1,
6   "_found": true,
7   "_source": {
8     "user": "oshrat",
9     "post_date": "2009-11-15T14:12:12",
10    "Message": "This is a beautiful day!!!"
11  }
12 }
13

```

אIOR 24 מקור המידע אשר נשמר לאחר אינדקס נתון

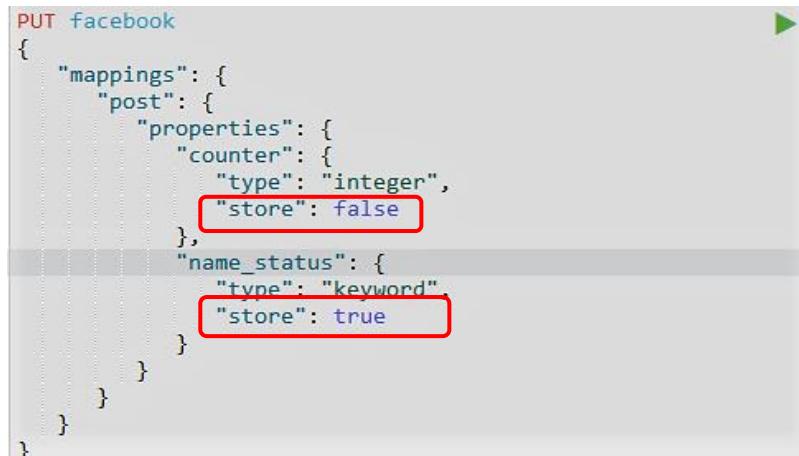
התוצאה הנ"ל כוללת את ה `_id`, `_type`, `_index` ו- `_version` של המסמך שאנו רוצים לאחזר, כולל המקוּר (`source`) בפועל של המסמך אם ניתן למצוא אותו (כפי שצוין על ידי השדה שנמצא בתגובה).

- סינון מקור: כבירית מחדל, פעולה ה- Get מוחזירה את התוכן של שדה המקור(source), אלא אם השתמשה בפרמטר של השדות מאוחסנים או אם שדה המקור מושבת. באפשרות לבטל את אחזור המקור באמצעות הפרמטר source.

```
GET facebook/post/1? _source=false
```

**שדות מאוחסנים** פועלת ה- get מאפשרת להגדיר קבוצה של שדות מאוחסנים שיוחזו על-ידי העברת הפרמטר storage\_fields. אם השדות המבוקשים אינם מאוחסנים, המערכת תתעלם מהם. לדוגמה, שקו את המיפוי הבא:

(1) בדוגמה זו, אנו מראים שבמיפוי ניתן להגדיר איזה מהשדות יהיו מאוחסנים או לא בתחום האינדקס.



```
PUT facebook
{
  "mappings": {
    "post": {
      "properties": {
        "counter": {
          "type": "integer",
          "store": false
        },
        "name_status": {
          "type": "keyword",
          "store": true
        }
      }
    }
  }
}
```

(2) כעת, אנחנו יכולים להוסיף מסמר שמכיל את השדות אלו.

```
PUT facebook/post/1
{
  "counter" : 1,
  "name_status" : ["Oshrat"]
}
```

(3) וננסה לאחזור אותו עם השדות :



```
"_index": "facebook",
"_type": "post",
"_id": "1",
"_version": 1,
"found": true,
"fields": {
  "name_status": [
    "Oshrat"
  ]
}
```

```
GET facebook/post/1?stored_fields=name_status,counter
```

(4) והתוצאה של הפעולה הנ"ל היא:  
ערכי שדות שנלקחו מהמසמר עצמו מוחזרים תמיד כמערך. מכיוון ששדה counter אינו מאוחסן, הבקשה פשוט מתעלמת ממנו בעת ניסיון להשיג את השדות המאוחסנים.

[渴求源碼](#)

השתמשו במקטע `{index} / {type} / {id} / _source`, כדי לקבל רק את השדה source של המסמר, ללא תוכן נוסף סיביו.

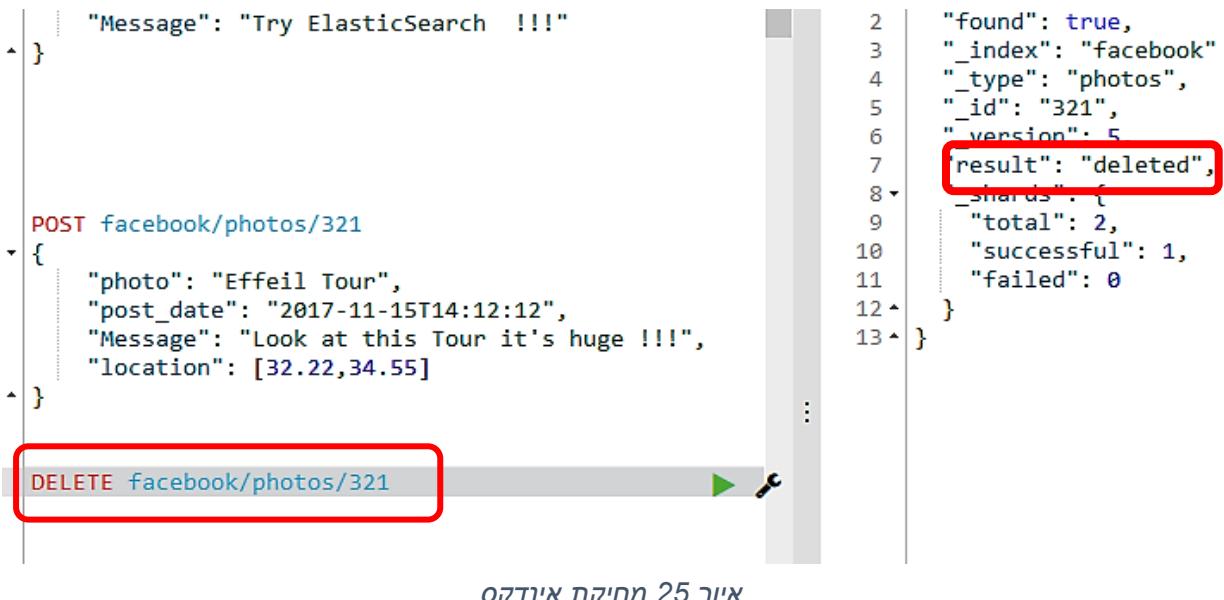
לדוגמה:

ניתן גם להשתמש באופן מסנני מקור כדי לקבוע אילו חלקיים של המקור יוחזרו:

GET facebook /post/1/\_source?\_source\_include=\*.id & \_source\_exclude=tags

### Delete API 7.1.3

ה- API למחיקה מאפשר למחוק מסמך JSON מוקלד מיינדקס ספציפי המבוסס על המזהה שלו. הדוגמא הבאה מוחק את מסמך JSON מהאינדקס שנקרא facebook, תחת סוג הנקרא photos, עם ערך 321 התוצאה של פעולה מחיקה לעיל היא:



The screenshot shows the Elasticsearch interface. On the left, there's a code editor pane with a red box around the DELETE request line: "DELETE facebook/photos/321". The right pane shows the response from the server, which includes a red box around the "result": "deleted" field in the JSON output.

```

    "Message": "Try ElasticSearch !!!"
}

POST facebook/photos/321
{
  "photo": "Effeil Tour",
  "post_date": "2017-11-15T14:12:12",
  "Message": "Look at this Tour it's huge !!!",
  "location": [32.22,34.55]
}

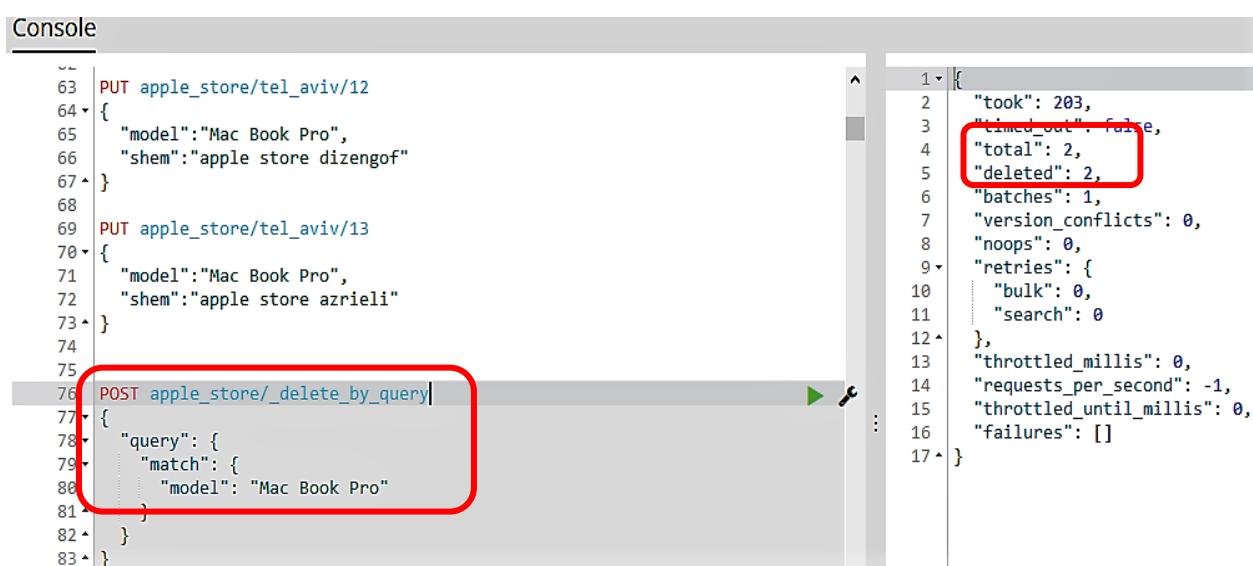
DELETE facebook/photos/321
  
```

```

2   "found": true,
3   "_index": "facebook"
4   "_type": "photos",
5   "_id": "321",
6   "_version": 5
7   "result": "deleted",
8   "_shards": [
9     "total": 2,
10    "successful": 1,
11    "failed": 0
12  }
13
  
```

איך 25 מחיקת אינדקס

### מחיקה על ידי ממashing API של שאליתה (delete by query)



The screenshot shows the Elasticsearch interface. On the left, there's a code editor pane with a red box around the POST request line: "POST apple\_store/\_delete\_by\_query". The right pane shows the response from the server, which includes a red box around the "deleted": 2 field in the JSON output.

```

63 PUT apple_store/tel_aviv/12
64 {
65   "model": "Mac Book Pro",
66   "shem": "apple store dizengof"
67 }
68
69 PUT apple_store/tel_aviv/13
70 {
71   "model": "Mac Book Pro",
72   "shem": "apple store azrieli"
73 }
74
75
76 POST apple_store/_delete_by_query|
77 {
78   "query": {
79     "match": {
80       "model": "Mac Book Pro"
81     }
82   }
83 }
  
```

```

1 {
2   "took": 203,
3   "timed_out": false,
4   "total": 2,
5   "deleted": 2,
6   "batches": 1,
7   "version_conflicts": 0,
8   "noops": 0,
9   "retries": {
10     "bulk": 0,
11     "search": 0
12   },
13   "throttled_millis": 0,
14   "requests_per_second": -1,
15   "throttled_until_millis": 0,
16   "failures": []
17
  
```

איך 26 מחיקה של אינדקסים מרובים בשאליתה אחת

השימוש הפשוט ביותר ב- API פשוט מבצע מחיקה על כל מסמך התואם לשאליתה. הנה ה-

במהלך ביצוע של `delete_by_query`, מספר שאלות חיפוש מבוצעות ברצף על מנת למצוא את כל המסמכים המתאימים למחיקה.

בכל פעם שנמצאת אצווה של מסמכים, מtbodyת בקשה מקבילה מתאימה כדי למחוק את כל המסמכים האלה. ניתן גם למחוק מסמכים של אינדקסים רבים וסוגים רבים בו-זמנית, בדיק כמו ה- API של החיפוש (Search API):

```
POST book,store/price,location/_delete_by_query
{
  "query": {
    "match_all": {}
  }
}
```

## Update API 7.1.4

ה- API לעדכון מאפשר לנו לעדכן מסמך על בסיס סקריפט נוסף. הפעולה מהזורה את המסמך מהאינדקס, מפעילה את הסקריפט ומאנדרסת את התוצאה (יש אפשרות גם למחוק). הוא משתמש בטיפוס של המסמך כדי להבטיח שלא התרחשו עדכונים במהלך ה `"GET"` או `"REINDEX"`.

לדוגמא, אנו מאיינדקסים מסמך פשוט:

```
#####
PUT book/thriller/15
{
  "counter":1,
  "tag":["thrill"]
}
#####
#
```

איור 27 אינדקס לוגי עדכון

עכשו, אנחנו יכולים להריץ סקריפט שיגדל את ה `counter` למשל:

```
POST book/thriller/15/_update
{
  "script": {
    "source": "ctx._source.counter += params.count",
    "lang": "painless",
    "params": {
      "count": 4
    }
  }
#####
#
```

```
1
2   "_index": "book",
3   "_type": "thriller",
4   "_id": "15",
5   "_version": 4,
6   "result": "updated",
7   "_shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  }
12 }
```

איור 29 שאלת עדכון עם SCRIPT

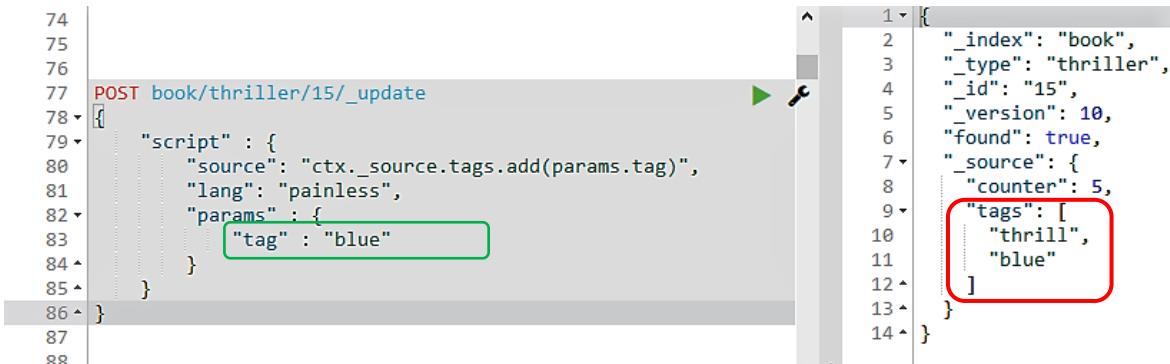
```
#
#####
#
```

`GET book/thriller/15`

```
5   "_version": 6,
6   "found": true,
7   "source": {
8     "counter": 5,
9     "tag": [
10       "thrill"
11     ]
12 }
```

איור 28 אינדקס לאחר עדכון בעמצעות SCRIPT

בנוסף לדוגמא שלם, אנו יכולים להוסיף-tag לרשימת התגיות (שםו לב, אם התג קיימ, הוא עדין יוסיף אותו לרשימה):



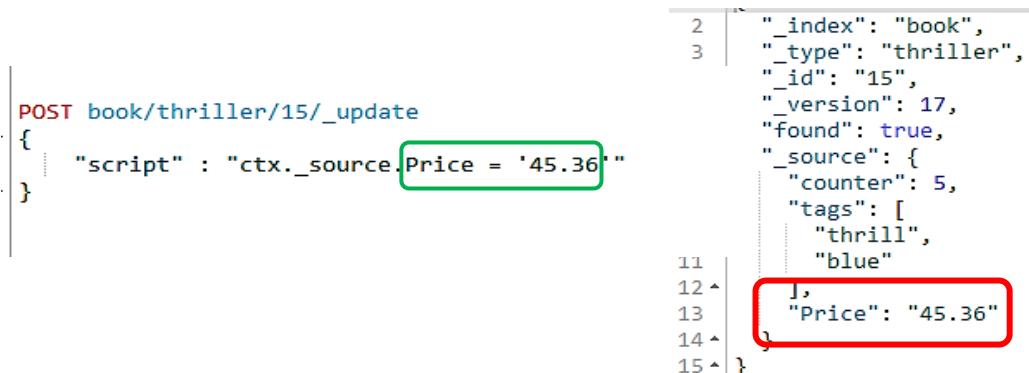
```

74
75
76
77 POST book/thriller/15/_update
78 {
79   "script" : {
80     "source": "ctx._source.tags.add(params.tag)",
81     "lang": "painless",
82     "params" : {
83       "tag" : "blue"
84     }
85   }
86 }
87
88
  
```

The screenshot shows a debugger interface with a code editor on the left and a results panel on the right. The code editor contains a POST request to update a document with a script that adds a tag ('blue') to the document's tags array. The results panel shows the updated document with the new tag added to the tags array.

איור 30 הוספה נתן באמצעות ה-SCRIPT

אנחנו יכולים גם להוסיף שדה חדש למסמר על ידי הסкриיפט:



```

POST book/thriller/15/_update
{
  "script" : "ctx._source.Price = '45.36'"
}
  
```

The screenshot shows a debugger interface with a code editor on the left and a results panel on the right. The code editor contains a POST request to update a document with a script that sets a new field ('Price') to '45.36'. The results panel shows the updated document with the new field added to the source object.

איור 31 הוספה שדה חדש לתוך העימדקם באמצעות ה-SCRIPT

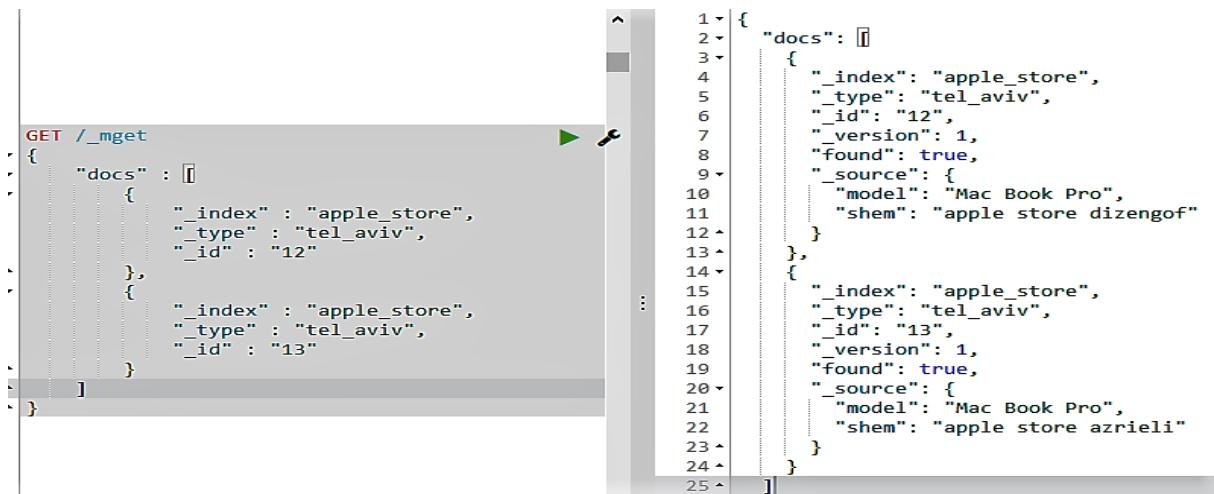
לחלווף, להסיר שדה מתוך המסמר:

```

POST book/thriller/15/_update
{
  "script": "ctx._source.remove('new_field')"
}
  
```

### Multi Get API 7.1.5

Multi-GET API מאפשר להשיג מספר מסמכים על בסיס אינדקס, טיפוס (אופציונלי) או id. התגובה כוללת מערך מסמכים עם כל המ속דים שהוחזרו, כל אלמנט דומה במבנה למסמר שסופק על ידי הממשק ה-Get API של הנה כמה דוגמאות של חיפוש:



```

GET /_mget
{
  "docs" : [
    {
      "_index" : "apple_store",
      "_type" : "tel_aviv",
      "_id" : "12"
    },
    {
      "_index" : "apple_store",
      "_type" : "tel_aviv",
      "_id" : "13"
    }
  ]
}

```

```

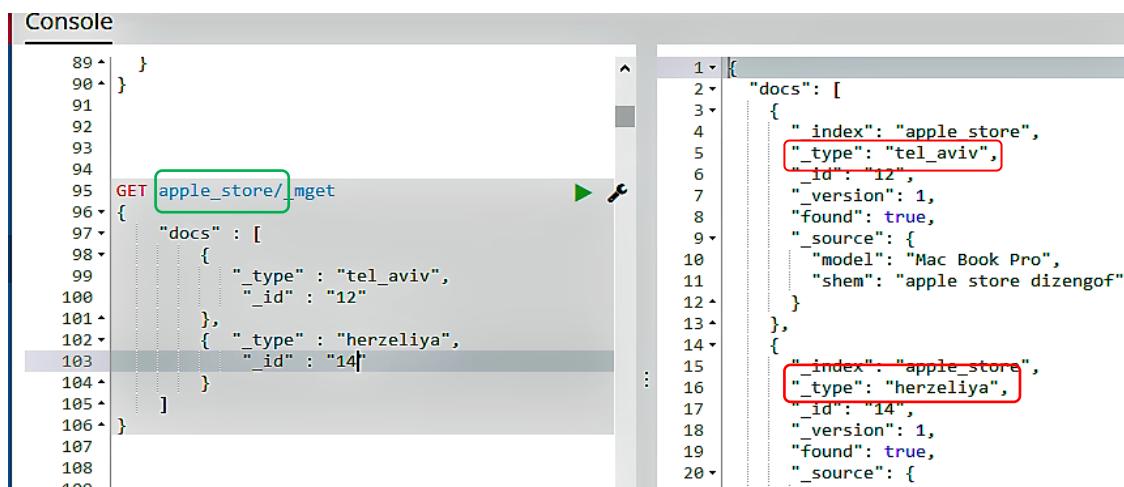
1  {
2   "docs": [
3    {
4      "_index": "apple_store",
5      "_type": "tel_aviv",
6      "_id": "12",
7      "_version": 1,
8      "found": true,
9      "_source": {
10        "model": "Mac Book Pro",
11        "shem": "apple store dizengof"
12      }
13    },
14    {
15      "_index": "apple_store",
16      "_type": "tel_aviv",
17      "_id": "13",
18      "_version": 1,
19      "found": true,
20      "_source": {
21        "model": "Mac Book Pro",
22        "shem": "apple store azrieli"
23      }
24  }
25 ]
}

```

איור 32 איחזר מרובת מסמכים

ניתן להשתמש גם בנקודות האינדקס (במקרה זה אין צורך בגוף):

- אם מחפשים למשל שני מסמכים וכל אחד מהטיפוס שונה, נציין לפחות את סוג ה "type" שלו באופןם בשדות "type" וונרשום את ה id שלהם. זה אותו דבר למספר רב יותר של מסמכים.



```

89  }
90 }
91
92
93
94
95 GET apple_store/_mget
96 {
97   "docs" : [
98     {
99       "_type" : "tel_aviv",
100      "_id" : "12"
101    },
102    {
103      "_type" : "herzeliya",
104      "_id" : "14"
105    }
106  ]
107
108 }

```

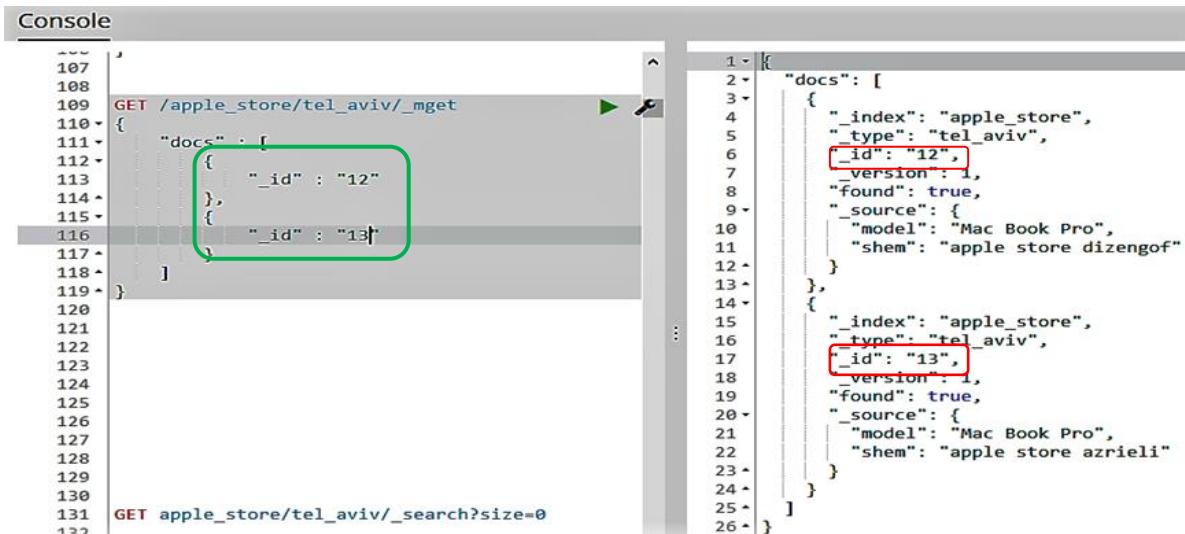
```

1  [
2   "docs": [
3    {
4      "_index": "apple_store",
5      "_type": "tel_aviv",
6      "_id": "12",
7      "_version": 1,
8      "found": true,
9      "_source": {
10        "model": "Mac Book Pro",
11        "shem": "apple store dizengof"
12      }
13    },
14    {
15      "_index": "apple_store",
16      "_type": "herzeliya",
17      "_id": "14",
18      "_version": 1,
19      "found": true,
20      "_source": {
21        "model": "Mac Book Pro"
22      }
23  }
24 ]
}

```

איור 33 שיטת איחזר מסמכים מרובים ראשונה

אם נרצה לחפש בתוך אינדקס מסמכים מסוימים, שאנו מכירים את המספר ה `_id` שלהם ואת הטיפוס שלהם, כך נרשום את קטע קוד.



```

107
108
109 GET /apple_store/tel_aviv/_mget
110 {
111     "docs": [
112         {
113             "_id": "12"
114         },
115         {
116             "_id": "13"
117         }
118     ]
119 }
120
121
122
123
124
125
126
127
128
129
130
131 GET apple_store/tel_aviv/_search?size=0
132

```

```

1  [
2     "docs": [
3         {
4             "_index": "apple_store",
5             "_type": "tel_aviv",
6             "_id": "12",
7             "_version": 1,
8             "found": true,
9             "_source": {
10                 "model": "Mac Book Pro",
11                 "shem": "apple store dizengof"
12             }
13         },
14         {
15             "_index": "apple_store",
16             "_type": "tel_aviv",
17             "_id": "13",
18             "_version": 1,
19             "found": true,
20             "_source": {
21                 "model": "Mac Book Pro",
22                 "shem": "apple store azrieli"
23             }
24     ]
25 ]
26

```

אייר 34 שיטת אחזור מסמכים מרובים שנייה

כברירת מחדל, השדה `_source` יוחזר עבור כל מסמך (אם הוא מאוחסן). בדומה למשك ה API - של GET לאחזר רק חלקים מהמקור (או בכלל לא) באמצעות הפורטט `source`. ניתן גם להשתמש בפורטט `_source`, כדי לציין ביריות מחדל, אשר ישמשו כאשר אין הוראות לכל מסמך.

```

GET _mget
{
    "docs": [
        {
            "_index": "warehouse",
            "_type": "marine",
            "_id": "1",
            "_source": false
        },
        {
            "_index": "hotel",
            "_type": "luxe",
            "_id": "2",
            "_source": ["field3", "field4"]
        }
    ]
}

```

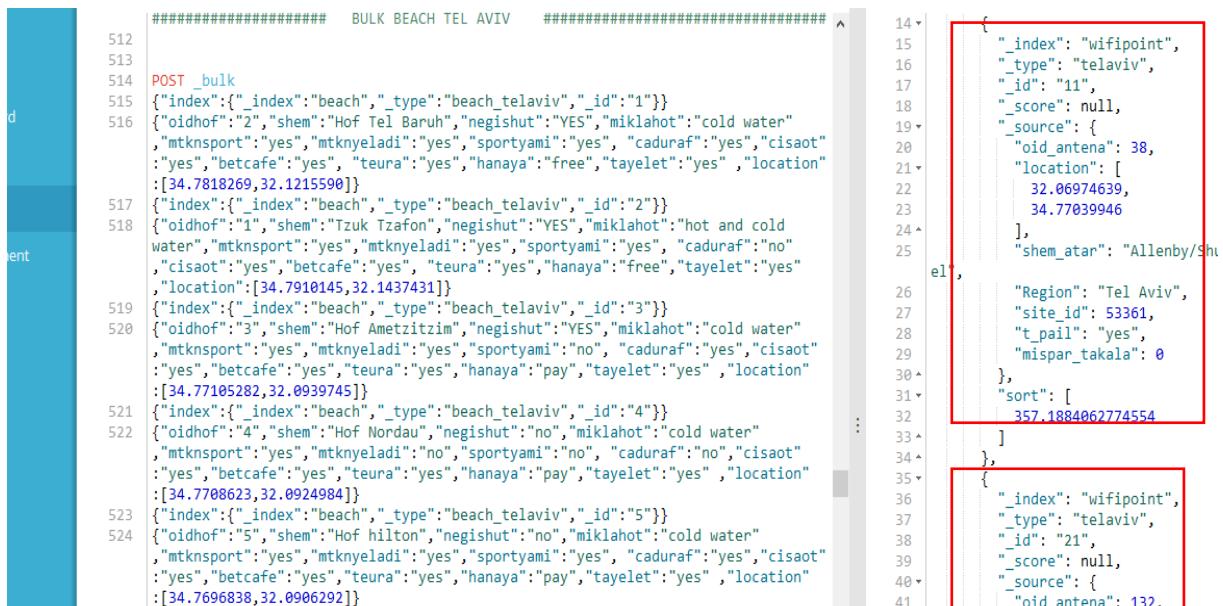
## BULK API 7.1.6

ה API BULK מאפשר לבצע פעולות רבות של אינדקס/מחיקה /עדכן של נתונים מרובים בבת אחת. זה יכול להגדיל באופן משמעותי את מהירות האינדקס. האפשרויות הן `bulk`, `bulk/{index}` / `bulk/{type}` / `bulk/{Index}`. כך יראה המבנה של Bulk אם נרצה לאנדקס / למחוק / לעדכן.

דוגמא להלן של איך לרשום שאלתה זו.

POST college/sce/\_bulk

```
{"index": {"_index": ": college", "_type": "sce", "_id": "1"}}
{"name": "Sami Shamoon College", "description": "SCE", "street": "jabotinsky",
"City": "Ashdod", "state": "IS", "zip": "178754", "Location": [31.89585, 76.8405],
"fees": 2000, "tags": ["junior Secondary", "beautiful campus"], "rating": "3.5"
{"update": {"_id": "45678", "_type": "sce", "_index": "college"} } ← עדכון של האינדקס
{"doc": {"fee": "86565 $"} }
{"delete": {"_index": "college", "_type": "sce", "_id": "102"} } ← מחיקת המסמך
```



```
512 ##### BULK BEACH TEL AVIV #####
513
514 POST _bulk
515 {"index": {"_index": "beach", "_type": "beach_telaviv", "_id": "1"}}
516 {"oidhof": "2", "shem": "Hof Tel Baruh", "negishut": "YES", "miklahot": "cold water",
"mtknsport": "yes", "mtknyeladi": "yes", "sportyami": "yes", "caduraf": "yes", "cisaot":
"yes", "betcafe": "yes", "teura": "yes", "hanaya": "free", "tayelet": "yes", "location":
[34.7818269, 32.1215590]}
517 {"index": {"_index": "beach", "_type": "beach_telaviv", "_id": "2"}}
518 {"oidhof": "1", "shem": "Tzuk Tzafon", "negishut": "YES", "miklahot": "hot and cold
water", "mtknsport": "yes", "mtknyeladi": "yes", "sportyami": "yes", "caduraf": "no"
"cisaot": "yes", "betcafe": "yes", "teura": "yes", "hanaya": "free", "tayelet": "yes"
"location": [34.7910145, 32.1437431]}
519 {"index": {"_index": "beach", "_type": "beach_telaviv", "_id": "3"}}
520 {"oidhof": "3", "shem": "Hof Ametitzim", "negishut": "YES", "miklahot": "cold water",
"mtknsport": "yes", "mtknyeladi": "yes", "sportyami": "no", "caduraf": "yes", "cisaot":
"yes", "betcafe": "yes", "teura": "yes", "hanaya": "pay", "tayelet": "yes", "location":
[34.77105282, 32.0939745]}
521 {"index": {"_index": "beach", "_type": "beach_telaviv", "_id": "4"}}
522 {"oidhof": "4", "shem": "Hof Nordau", "negishut": "no", "miklahot": "cold water",
"mtknsport": "yes", "mtknyeladi": "no", "sportyami": "no", "caduraf": "no", "cisaot":
"yes", "betcafe": "yes", "teura": "yes", "hanaya": "pay", "tayelet": "yes", "location":
[34.7708623, 32.0924984]}
523 {"index": {"_index": "beach", "_type": "beach_telaviv", "_id": "5"}}
524 {"oidhof": "5", "shem": "Hof hilton", "negishut": "no", "miklahot": "cold water",
"mtknsport": "yes", "mtknyeladi": "yes", "sportyami": "yes", "caduraf": "yes", "cisaot":
"yes", "betcafe": "yes", "teura": "yes", "hanaya": "pay", "tayelet": "yes", "location":
[34.7696838, 32.0906292]}
```

```
14 *
15 {
16   "_index": "wifipoint",
17   "_type": "telaviv",
18   "_id": "11",
19   "_score": null,
20   "_source": {
21     "oid_antena": 38,
22     "location": [
23       32.06974639,
24       34.77039946
25     ],
26     "shem_atar": "Allenby/Shm
el",
27     "Region": "Tel Aviv",
28     "site_id": 53361,
29     "t_pall": "yes",
30     "mispar_takala": 0
31   },
32   "sort": [
33     357.1884062774554
34   ]
35 },
36 {
37   "_index": "wifipoint",
38   "_type": "telaviv",
39   "_id": "21",
40   "_score": null,
41   "_source": {
42     "oid_antena": 132,
```

BULK אינדקס נתונים עירית ת"א ע"י שיטת ה 35 או'

איור זה מתאר את פקודת Bulk ב Elasticsearch עם נתונים חופים השונים של תל אביב.

## ELASTICSEARCH — DOCUMENT APIs 7.1.7

ממתק תכונות היישומים (API) ברשות הוא קבוצה של קראיות פונקציה או הוראות תכונות אחרות כדי לגשת לרכיב התוכנה באותו יישום אינטרנט מסוים.

לדוגמא, ה API - של Facebook מסיע למפתח ליצור יישומים על ידי גישה לנוטונים או לתכונות אחרות של פייסבוק; זה יכול להיות תאריך לידה או עדכון>Status. Elasticsearch מושפע מREST API אשר נגיש באמצעות JSON מעל HTTP. Elasticsearch משתמש במוסכמות הבאות:

## Elasticsearch -Mapping 7.2

המיופיע הוא תהליך הגדרת האופן של מסמך, והשדות שהוא מכיל שבו מאוחסנים ומאנדקסים. לדוגמה, אנו משתמשים במיפויים להגדרת:

- + אילו שדות של מחרוזת יש להתייחס אליהם כשדות טקסט מלאים.
- + אשר שדות מכילים מספרים, תאריכים, או geolocations.
- + את הפורמט של ערכי התאריך.

כליים מותאמים אישית כדי לשלוט על המיפוי עבור שדות שנוספו באופן דינמי.

## Field Datatypes 7.2.1

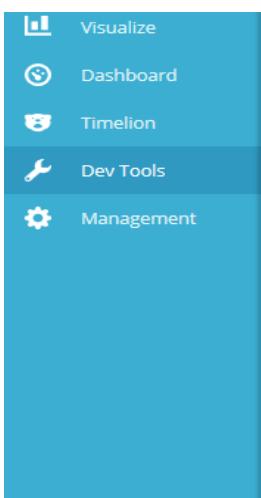
לכל שדה יש סוג נתונים שיכל להיות:  
 סוג פשוט כמו Boolean, Double, Long, Date, keyword, texte או IP. סוג התומך בטבע היררכי של JSON כגון geo\_point, geo\_shape או אובייקט או מוקן. או סוג מיוחד כמו שדה בדרכים שונות למטרות שונות. לדוגמה, שדה מחרוזת יכול להיות באינדקס כשדה טקסט לחיפוש טקסט מלא וכשדה של מילוי מפתח למילוי או לאיסוף.

## Dynamic mapping 7.2.2

שדות וסוגי מיפוי לא צריכים להיות מוגדרים לפני שימוש בהם עם מיפוי דינמי, שמורות של שדות חדשים יתווסף באופן אוטומטי, פשוט על ידי יצירת אינדקס של מסמר. ניתן להוסיף שדות חדשים הן לשוג המיפוי בرمמה העליונה והן לאובייקט הפנימי ולשדות המוקנים. ניתן להגדיר כללי מיפוי דינמיים כדי להתאים אישית את המיפוי המשמש לשדות חדשים.

## Explicit mapping 7.2.3

אם אנו יודעים יותר על הנתונים שלנו מאשר Elasticsearch יכול לנחש, בעוד שמייפוי דינמי יכול להיות שימושי בהתחלה, בשלב מסוים אנו נרצה ליצור ליצין מיפויים מפורטים יותר משלנו. ניתן ליצור מיפוי שדה בעט יצירת אינדקס, ונitin להוסיף שדות לאינדקס קיים באמצעות משק ה-PUT mapping API.



```

628 ##### Mapping theatres telaviv #####
629 #####
630 #####
631 PUT theatres
632 {
633   "mappings": {
634     "theatre_telaviv": {
635       "properties": {
636         "oidthe": { "type": "integer" },
637         "msbait": { "type": "integer" },
638         "Address": { "type": "text" },
639         "knisa": { "type": "integer" },
640         "shem": { "type": "text" },
641         "kneghishut": { "type": "keyword" },
642         "phonenumb": { "type": "keyword" },
643         "shemangli": { "type": "keyword" },
644         "location": { "type": "geo_point" }
645       }
646     }
647   }
648 }
```

איך 36 מיפוי השדות של אינדקס

איור זה מתאר את המיפוי שדות של אינדקס theatre אשר נתונים סופקו מהאתר של ירידת ת"א.

## ELASTICSEARCH — AGGREGATIONS 7.3

### Metrics Aggregations 7.3.1

מסגרת הצבירה מסייעת לספק נתונים מוצברים על סמך שאילתת חיפוש. הוא מבוסס על קטעי קוד פשוטים הנקרים ארגזיות, שיכולים להיות מרכיבים על מנת לבנות סיכומים מורכבים של הנתונים. המבנה הבסיסי של הצבירה מוצג להלן:

```
"aggregations" : {
    "<aggregation_name>" : {
        "<aggregation_type>" : {
<aggregation_body>
    }
    [ , "meta" : { [<meta_data_body>] } ]?
    [ , "aggregations" : { [<sub_aggregation>]+ } ]?
}
}
```

ישנם סוגים שונים של ארגזיות, כל אחד עם המטרה שלו:

### Average Aggregation 7.3.2

צבירה של ערכים בעלי ערך ייחד המחשבת את הממוצע של ערכים מספריים שחולצו מהמסמכים המוצברים.



The screenshot shows the Elasticsearch Dev Tools Management interface. On the left, there's a code editor with a scroll bar containing lines 157 through 172. Lines 160-166 show a POST request to /wifipoint/\_search?size=0 with the following body:

```
160 POST /wifipoint/_search?size=0
161 {
162
163     "aggs" : {
164         "avg_grade" : { "avg": { "field" : "price" } }
165     }
166 }
```

To the right, the results of the search are displayed. The results section starts at line 10 and includes:

```
10     "hits": {
11         "total": 5,
12         "max_score": 0,
13         "hits": []
14     },
15     "aggregations": {
16         "avg_grade": {
17             "value": 4153.8
18         }
19     }
20 }
```

איור 37 שאילתת ממוצע

איור זה מתאר את הצבירה הממוצעת של מחיר WiFi point באינדקס wifi אשר נתוני סופקו מהאתר של יריעת ת"א.

### Cardinality Aggregation 7.3.3

זו צבירה של ערכים בעלי ערך ייחד המחשבת ספירה משוערת של ערכים נפרדים, ניתן לחץ ערכים משודוטים ספציפיים במסמך או שנוצרים על-ידי סקריפט.



```

169
170
171
172 ##### Cardinality Aggregation HOW MANY DIFFERENT FIELDS #####
173 ##### Cardinality Aggregation HOW MANY DIFFERENT FIELDS #####
174 POST /wifipoint/_search?size=0
175 {
176     "aggs" : {
177         "type_count" : {
178             "cardinality" : {
179                 "field" : "mispar_takala"
180             }
181         }
182     }
183 }
184
185 #####

```

איור 38 שאלת זו סופרת את כל התקנות השונות של WIFI

. Elasticsearch מתרגם את הצבירה של ערכים השונים של תקנות באינדקס Wifi Point ב

#### Max Aggregation 7.3.4

ציבורה של ערכים בעלי ערך יחיד המציג את הערך המרבי בין הערכים המספריים שחולצו מהמסמכים המוצטברים.



```

138
139
140
141
142
143
144 ##### AGGREGATION MAX WIFIPOINT TAKALA #####
145 POST /wifipoint/_search?size=0
146 {
147     "aggs" : {
148         "max_mispar_takala" : { "max" : { "field" : "mispar_takala" } }
149     }
150 }
151 #####
152
153

```

איור 39 שאלת מקסימלית

. Elasticsearch מתרגם את הצבירה של בעלי ערך המקסימלי באינדקס wifi point ב

#### Min Aggregation 7.3.5

ציבורה של ערכים בעלי ערך יחיד המציג את הערך המינימלי בין ערכים מספריים שחולצו מהמסמכים המוצטברים.



```

670     "avg_price" : { "avg" : { "field" : "knisa" } }
671   }
672
673 }
674 ###### AGGREGATION MIN PRICE OF KNISA IN THEATRES AT TLV ######
675
676 POST /theatres/_search?size=0
677 {
678   "aggs" : {
679     "min_price" : {
680       "min" : { "field" : "knisa" } }
681     }
682   }
683
684 ###### #####
685

```

```

6   "successful": 5,
7   "skipped": 0,
8   "failed": 0
9 },
10 },
11 "hits": {
12   "total": 6,
13   "max_score": 0,
14   "hits": []
15 },
16 "aggregations": {
17   "min_price": {
18     "value": 45
19   }
20 }

```

איור 40 שאילתת המינימלית

. Elasticsearch מター את הצבירה שמחזירה את הערך price המינימלי באינדקס Théâtres ב

### Bucket Aggregations 7.3.6

#### Geo Distance Aggregation 7.3.6.1

שאילתת זו צוברת קבוצות נתונים אשר עליהם היא פועלת על שדות geo point שלם ומצבירה זו נינתן למין אוטם לפי טווחים שהמשמש יכול להגדיר. המשמש יכול להגדיר נקודות מזוקה וטווחים שונים. הצבירה מעריכה את המרחק של כל מסמך מנקודות המזוקה וקובעת לאיזה קבוצה נתונים הם שייכים על סמך הטווחים (מספר שיר לקבוצת נתונים, אם המרחק בין המספר לבין המיקור נמצא בתחום המרחק של הקבוצה הזו).



```

692 }
693 }
694
695
696 ## Geo Distance Aggregations for find pharmacies in rings around Tel Aviv
697 #####
698 POST /pharmacies/_search?size=0
699 {
700   "aggs" : {
701     "rings_around_telaviv" : {
702       "geo_distance" : [
703         "location",
704         "origin" : "32.0767,34.78476",
705         "ranges" : [
706           { "from": 500, "key": "first_ring" },
707           { "from": 1000, "to": 3000, "key": "second_ring" },
708           { "from": 1500, "key": "third_ring" }
709         ]
710       }
711     }
712   }

```

```

14 },
15 "aggregations": {
16   "rings_around_telaviv": {
17     "buckets": [
18       {
19         "key": "first_ring",
20         "from": 500,
21         "doc_count": 11
22       },
23       {
24         "key": "second_ring",
25         "from": 1000,
26         "to": 3000,
27         "doc_count": 9
28       },
29       {
30         "key": "third_ring",
31         "from": 1500,
32         "doc_count": 5
33       }
34     ]
35   }

```

איור 41 שאילתת מין גאוגרפיה

איור זה מター את צבירת נתונים מהאינדקס של בת"א אשר מחזירה את כמות הנתונים שנמצאים בטווחים השונים(טבעות שונות) מנקודה שהגדרתי מהאינדקס pharmacies ב Elasticsearch .

### Filter Aggregation 7.3.6.2

כבריה זו מסננת את האוסף של נתונים של כל המסמכים שאנו מגדירים מתוך אינדקס מסוים, לעיתים קרובות זה ישמש כדי לחדד את הקשר הצבירה אל קבוצה מסוימת של מסמכים.



```

192     }
193   }
194 ##### Filter Aggregation #####
195
196 POST /wifipoint/_search?
197 {
198   "aggs" : {
199     "telavivs" : {
200       "filter" : { "term": { "_type": "telaviv" } },
201       "aggs" : {
202         "avg_t" : { "avg" : { "field" : "price" } }
203       }
204     }
205   }
206 }
207 #####
208

```

```

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

איור 42 שאילתת המסננים

בדוגמא לעיל, אנו מחשבים את המחיר הממוצע של כל המחיצרים של נקודות wifi שהם נמצאים בתל אביב (מאוטו טיפוס).

### ELASTICSEARCH — QUERY DSL 7.4

ב Elasticsearch, החיפוש מתבצע באמצעות שאילתת המבוססת על JSON השאיתה מתבצעת על שבי סעיפים:

- + **סעיפים שאילתת הعلا - סעיפים אלה הם term, range query או match**, אשר מחפשים ערך ספציפי בתחום ספציפי.
- + **סעיפים שאילתת מרכיבים - שאילתות אלה הן שילוב של סעיפים שאילתת עליים ושאליות מרכיבות אחרות כדי להלץ את המידע הרצוי.**

Elasticsearch תומך במספר גדול של שאילתות, שאילתת מתחילה במילת מפתח ולאחר מכן יש אפשרות להוסיף תנאים ומסננים בתוך הגוף של האובייקט JSONoSוגים שונים של שאילתות תוארו להלן:

### Full Text Queries 7.4.1

שאילתות אלה משמשות לחיפוש גוף מלא של טקסט כמו פרק, הودעת דוא"ל או מאמר, שאילתת זו פועלת בהתאם למונח המשיך לוותה אינדקס או מסמך מסוים. בסעיף זה נדון בשאלות השונות של שאילתות טקסט מלאות השאילתות בקבוצה זו הן:

#### Match\_query 7.4.1.1

שאילתת זו תואמת טקסט או ביטוי עם הערכים של שדה אחד או יותר לדוגמה:  
שיםו לב, כאן "shem\_atar" הוא שם של שדה של מסמך כלשהו המאוחסן, אתם יכולים להחלף את השם בכל שדה אחר במקום שדה זה.  
בדוגמה לעיל, אנו חיפשנו מסמך מתוך האינדקסים המוכלים ב Elasticsearch, שמכיל את שם האתר שבחרנו.  
קיבלו תוצאה אחת המכילה אותה וזהו נקודת wifi.wifi.



```

board
lion
Tools
Management

1 13
2 14
3 15
4 16
5 17
6 18
7 19
8 20
9 21
10 22
11 23
12 24
13 25
14 26
15 27
16 28
17 29
18 30
19 31
20 32
21 33
22 34
23 35
24 36
25 37
26 38
27 39
28 40
29 41
30 42
31 43
32 44
33 45
34 46
35 47
36 48
37 49
38 50
39 51
40 52
41 53
42 54
43 55
44 56
45 57
46 58
47 59
48 60
49 61
50 62
51 63
52 64
53 65
54 66
55 67
56 68
57 69
58 70
59 71
60 72
61 73
62 74
63 75
64 76
65 77
66 78
67 79
68 80
69 81
70 82
71 83
72 84
73 85
74 86
75 87
76 88
77 89
78 90
79 91
80 92
81 93
82 94
83 95
84 96
85 97
86 98
87 99
88 100
89 101
90 102
91 103
92 104
93 105
94 106
95 107
96 108
97 109
98 110
99 111
100 112
101 113
102 114
103 115
104 116
105 117
106 118
107 119
108 120
109 121
110 122
111 123
112 124
113 125
114 126
115 127
116 128
117 129
118 130
119 131
120 132
121 133
122 134
123 135
124 136
125 137
126 138
127 139
128 140
129 141
130 142
131 143
132 144
133 145
134 146
135 147
136 148
137 149
138 150
139 151
140 152
141 153
142 154
143 155
144 156
145 157
146 158
147 159
148 160
149 161
150 162
151 163
152 164
153 165
154 166
155 167
156 168
157 169
158 170
159 171
160 172
161 173
162 174
163 175
164 176
165 177
166 178
167 179
168 180
169 181
170 182
171 183
172 184
173 185
174 186
175 187
176 188
177 189
178 190
179 191
180 192
181 193
182 194
183 195
184 196
185 197
186 198
187 199
188 200
189 201
190 202
191 203
192 204
193 205
194 206
195 207
196 208
197 209
198 210
199 211
200 212
201 213
202 214
203 215
204 216
205 217
206 218
207 219
208 220
209 221
210 222
211 223
212 224
213 225
214 226
215 227
216 228
217 229
218 230
219 231
220 232
221 233
222 234
223 235
224 236
225 237
226 238
227 239
228 240
229 241
230 242
231 243
232 244
233 245
234 246
235 247
236 248
237 249
238 250
239 251
240 252
241 253
242 254
243 255
244 256
245 257
246 258
247 259
248 260
249 261
250 262
251 263
252 264
253 265
254 266
255 267
256 268
257 269
258 270
259 271
260 272
261 273
262 274
263 275
264 276
265 277
266 278
267 279
268 280
269 281
270 282
271 283
272 284
273 285
274 286
275 287
276 288
277 289
278 290
279 291
280 292
281 293
282 294
283 295
284 296
285 297
286 298
287 299
288 300
289 301
290 302
291 303
292 304
293 305
294 306
295 307
296 308
297 309
298 310
299 311
300 312
301 313
302 314
303 315
304 316
305 317
306 318
307 319
308 320
309 321
310 322
311 323
312 324
313 325
314 326
315 327
316 328
317 329
318 330
319 331
320 332
321 333
322 334
323 335
324 336
325 337
326 338
327 339
328 340
329 341
330 342
331 343
332 344
333 345
334 346
335 347
336 348
337 349
338 350
339 351
340 352
341 353
342 354
343 355
344 356
345 357
346 358
347 359
348 360
349 361
350 362
351 363
352 364
353 365
354 366
355 367
356 368
357 369
358 370
359 371
360 372
361 373
362 374
363 375
364 376
365 377
366 378
367 379
368 380
369 381
370 382
371 383
372 384
373 385
374 386
375 387
376 388
377 389
378 390
379 391
380 392
381 393
382 394
383 395
384 396
385 397
386 398
387 399
388 400
389 401
390 402
391 403
392 404
393 405
394 406
395 407
396 408
397 409
398 410
399 411
400 412
401 413
402 414
403 415
404 416
405 417
406 418
407 419
408 420
409 421
410 422
411 423
412 424
413 425
414 426
415 427
416 428
417 429
418 430
419 431
420 432
421 433
422 434
423 435
424 436
425 437
426 438
427 439
428 440
429 441
430 442
431 443
432 444
433 445
434 446
435 447
436 448
437 449
438 450
439 451
440 452
441 453
442 454
443 455
444 456
445 457
446 458
447 459
448 460
449 461
450 462
451 463
452 464
453 465
454 466
455 467
456 468
457 469
458 470
459 471
460 472
461 473
462 474
463 475
464 476
465 477
466 478
467 479
468 480
469 481
470 482
471 483
472 484
473 485
474 486
475 487
476 488
477 489
478 490
479 491
480 492
481 493
482 494
483 495
484 496
485 497
486 498
487 499
488 500
489 501
490 502
491 503
492 504
493 505
494 506
495 507
496 508
497 509
498 510
499 511
500 512
501 513
502 514
503 515
504 516
505 517
506 518
507 519
508 520
509 521
510 522
511 523
512 524
513 525
514 526
515 527
516 528
517 529
518 530
519 531
520 532
521 533
522 534
523 535
524 536
525 537
526 538
527 539
528 540
529 541
530 542
531 543
532 544
533 545
534 546
535 547
536 548
537 549
538 550
539 551
540 552
541 553
542 554
543 555
544 556
545 557
546 558
547 559
548 560
549 561
550 562
551 563
552 564
553 565
554 566
555 567
556 568
557 569
558 570
559 571
560 572
561 573
562 574
563 575
564 576
565 577
566 578
567 579
568 580
569 581
570 582
571 583
572 584
573 585
574 586
575 587
576 588
577 589
578 590
579 591
580 592
581 593
582 594
583 595
584 596
585 597
586 598
587 599
588 600
589 601
590 602
591 603
592 604
593 605
594 606
595 607
596 608
597 609
598 610
599 611
600 612
601 613
602 614
603 615
604 616
605 617
606 618
607 619
608 620
609 621
610 622
611 623
612 624
613 625
614 626
615 627
616 628
617 629
618 630
619 631
620 632
621 633
622 634
623 635
624 636
625 637
626 638
627 639
628 640
629 641
630 642
631 643
632 644
633 645
634 646
635 647
636 648
637 649
638 650
639 651
640 652
641 653
642 654
643 655
644 656
645 657
646 658
647 659
648 660
649 661
650 662
651 663
652 664
653 665
654 666
655 667
656 668
657 669
658 670
659 671
660 672
661 673
662 674
663 675
664 676
665 677
666 678
667 679
668 680
669 681
670 682
671 683
672 684
673 685
674 686
675 687
676 688
677 689
678 690
679 691
680 692
681 693
682 694
683 695
684 696
685 697
686 698
687 699
688 700
689 701
690 702
691 703
692 704
693 705
694 706
695 707
696 708
697 709
698 710
699 711
700 712
701 713
702 714
703 715
704 716
705 717
706 718
707 719
708 720
709 721
710 722
711 723
712 724
713 725
714 726
715 727
716 728
717 729
718 730
719 731
720 732
721 733
722 734
723 735
724 736
725 737
726 738
727 739
728 740
729 741
730 742
731 743
732 744
733 745
734 746
735 747
736 748
737 749
738 750
739 751
740 752
741 753
742 754
743 755
744 756
745 757
746 758
747 759
748 760
749 761
750 762
751 763
752 764
753 765
754 766
755 767
756 768
757 769
758 770
759 771
760 772
761 773
762 774
763 775
764 776
765 777
766 778
767 779
768 780
769 781
770 782
771 783
772 784
773 785
774 786
775 787
776 788
777 789
778 790
779 791
780 792
781 793
782 794
783 795
784 796
785 797
786 798
787 799
788 800
789 801
790 802
791 803
792 804
793 805
794 806
795 807
796 808
797 809
798 810
799 811
800 812
801 813
802 814
803 815
804 816
805 817
806 818
807 819
808 820
809 821
810 822
811 823
812 824
813 825
814 826
815 827
816 828
817 829
818 830
819 831
820 832
821 833
822 834
823 835
824 836
825 837
826 838
827 839
828 840
829 841
830 842
831 843
832 844
833 845
834 846
835 847
836 848
837 849
838 850
839 851
840 852
841 853
842 854
843 855
844 856
845 857
846 858
847 859
848 860
849 861
850 862
851 863
852 864
853 865
854 866
855 867
856 868
857 869
858 870
859 871
860 872
861 873
862 874
863 875
864 876
865 877
866 878
867 879
868 880
869 881
870 882
871 883
872 884
873 885
874 886
875 887
876 888
877 889
878 890
879 891
880 892
881 893
882 894
883 895
884 896
885 897
886 898
887 899
888 900
889 901
890 902
891 903
892 904
893 905
894 906
895 907
896 908
897 909
898 910
899 911
900 912
901 913
902 914
903 915
904 916
905 917
906 918
907 919
908 920
909 921
910 922
911 923
912 924
913 925
914 926
915 927
916 928
917 929
918 930
919 931
920 932
921 933
922 934
923 935
924 936
925 937
926 938
927 939
928 940
929 941
930 942
931 943
932 944
933 945
934 946
935 947
936 948
937 949
938 950
939 951
940 952
941 953
942 954
943 955
944 956
945 957
946 958
947 959
948 960
949 961
950 962
951 963
952 964
953 965
954 966
955 967
956 968
957 969
958 970
959 971
960 972
961 973
962 974
963 975
964 976
965 977
966 978
967 979
968 980
969 981
970 982
971 983
972 984
973 985
974 986
975 987
976 988
977 989
978 990
979 991
980 992
981 993
982 994
983 995
984 996
985 997
986 998
987 999
988 1000
989 1001
990 1002
991 1003
992 1004
993 1005
994 1006
995 1007
996 1008
997 1009
998 1010
999 1011
1000 1012
1001 1013
1002 1014
1003 1015
1004 1016
1005 1017
1006 1018
1007 1019
1008 1020
1009 1021
1010 1022
1011 1023
1012 1024
1013 1025
1014 1026
1015 1027
1016 1028
1017 1029
1018 1030
1019 1031
1020 1032
1021 1033
1022 1034
1023 1035
1024 1036
1025 1037
1026 1038
1027 1039
1028 1040
1029 1041
1030 1042
1031 1043
1032 1044
1033 1045
1034 1046
1035 1047
1036 1048
1037 1049
1038 1050
1039 1051
1040 1052
1041 1053
1042 1054
1043 1055
1044 1056
1045 1057
1046 1058
1047 1059
1048 1060
1049 1061
1050 1062
1051 1063
1052 1064
1053 1065
1054 1066
1055 1067
1056 1068
1057 1069
1058 1070
1059 1071
1060 1072
1061 1073
1062 1074
1063 1075
1064 1076
1065 1077
1066 1078
1067 1079
1068 1080
1069 1081
1070 1082
1071 1083
1072 1084
1073 1085
1074 1086
1075 1087
1076 1088
1077 1089
1078 1090
1079 1091
1080 1092
1081 1093
1082 1094
1083 1095
1084 1096
1085 1097
1086 1098
1087 1099
1088 1100
1089 1101
1090 1102
1091 1103
1092 1104
1093 1105
1094 1106
1095 1107
1096 1108
1097 1109
1098 1110
1099 1111
1100 1112
1101 1113
1102 1114
1103 1115
1104 1116
1105 1117
1106 1118
1107 1119
1108 1120
1109 1121
1110 1122
1111 1123
1112 1124
1113 1125
1114 1126
1115 1127
1116 1128
1117 1129
1118 1130
1119 1131
1120 1132
1121 1133
1122 1134
1123 1135
1124 1136
1125 1137
1126 1138
1127 1139
1128 1140
1129 1141
1130 1142
1131 1143
1132 1144
1133 1145
1134 1146
1135 1147
1136 1148
1137 1149
1138 1150
1139 1151
1140 1152
1141 1153
1142 1154
1143 1155
1144 1156
1145 1157
1146 1158
1147 1159
1148 1160
1149 1161
1150 1162
1151 1163
1152 1164
1153 1165
1154 1166
1155 1167
1156 1168
1157 1169
1158 1170
1159 1171
1160 1172
1161 1173
1162 1174
1163 1175
1164 1176
1165 1177
1166 1178
1167 1179
1168 1180
1169 1181
1170 1182
1171 1183
1172 1184
1173 1185
1174 1186
1175 1187
1176 1188
1177 1189
1178 1190
1179 1191
1180 1192
1181 1193
1182 1194
1183 1195
1184 1196
1185 1197
1186 1198
1187 1199
1188 1200
1189 1201
1190 1202
1191 1203
1192 1204
1193 1205
1194 1206
1195 1207
1196 1208
1197 1209
1198 1210
1199 1211
1200 1212
1201 1213
1202 1214
1203 1215
1204 1216
1205 1217
1206 1218
1207 1219
1208 1220
1209 1221
1210 1222
1211 1223
1212 1224
1213 1225
1214 1226
1215 1227
1216 1228
1217 1229
1218 1230
1219 1231
1220 1232
1221 1233
1222 1234
1223 1235
1224 1236
1225 1237
1226 1238
1227 1239
1228 1240
1229 1241
1230 1242
1231 1243
1232 1244
1233 1245
1234 1246
1235 1247
1236 1248
1237 1249
1238 1250
1239 1251
1240 1252
1241 1253
1242 1254
1243 1255
1244 1256
1245 1257
1246 1258
1247 1259
1248 1260
1249 1261
1250 1262
1251 1263
1252 1264
1253 1265
1254 1266
1255 1267
1256 1268
1257 1269
1258 1270
1259 1271
1260 1272
1261 1273
1262 1274
1263 1275
1264 1276
1265 1277
1266 1278
1267 1279
1268 1280
1269 1281
1270 1282
1271 1283
1272 1284
1273 1285
1274 1286
1275 1287
1276 1288
1277 1289
1278 1290
1279 1291
1280 1292
1281 1293
1282 1294
1283 1295
1284 1296
1285 1297
1286 1298
1287 1299
1288 1300
1289 1301
1290 1302
1291 1303
1292 1304
1293 1305
1294 1306
1295 1307
1296 1308
1297 1309
1298 1310
1299 1311
1300 1312
130
```

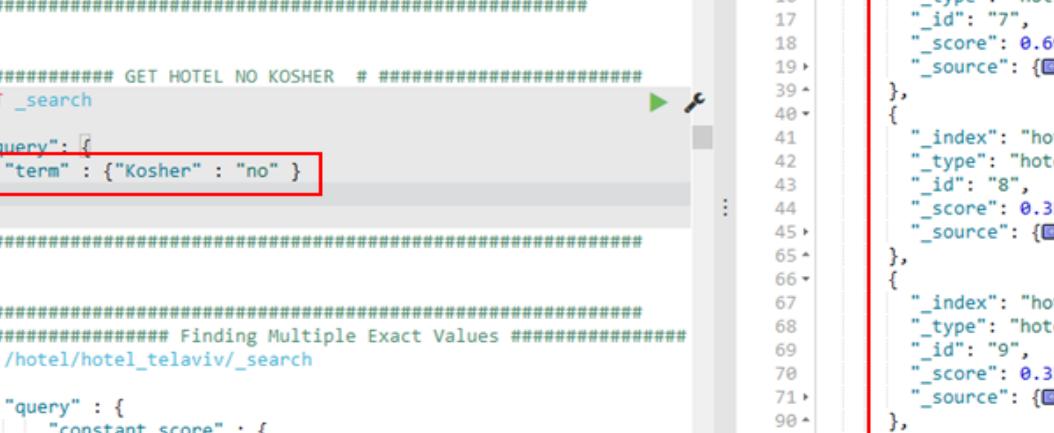
**איור 44 שאילתת חיפוש מושכים מרובים באמצעות התאמת של השדות**

בוגרמא לעיל ניתן לראות שכאר שאלות מזינים את שאילתת multi match בנתוני חיפוש כגון: "address" ו- "multi" אשר מכיל את המילה "houlda11", במטרה לחפש את המילה/משפט "houlda11" במספר רב של מילים אשר נמצאו במהלך חיפושם. נמצוא שהמשפט קיים בשני אינדקסים שונים כגון: "Hotel", "Wifipoints".

## Terms Query 7.4.1.3

שאילתת המונח מוצאת מסמכים המכילים את המונח המדוייק.

```
368     "shem_rechov": "Bluch 32"
369   }
370 }
371 #####
372 ##### GET HOTEL NO KOSHER # #####
373
374
375 ##### GET HOTEL NO KOSHER # #####
376 POST _search
377 {
378   "query": [
379     {"term" : {"Kosher" : "no" } }
380   ]
381 }
382 #####
383
384
385 #####
386 ##### Finding Multiple Exact Values #####
387 GET /hotel/hotel_telaviv/_search
388 {
389   "query" : {
390     "constant_score" : {
391       "filter" : {
392         "terms" : {
393           "Classification " : [3,4]
```

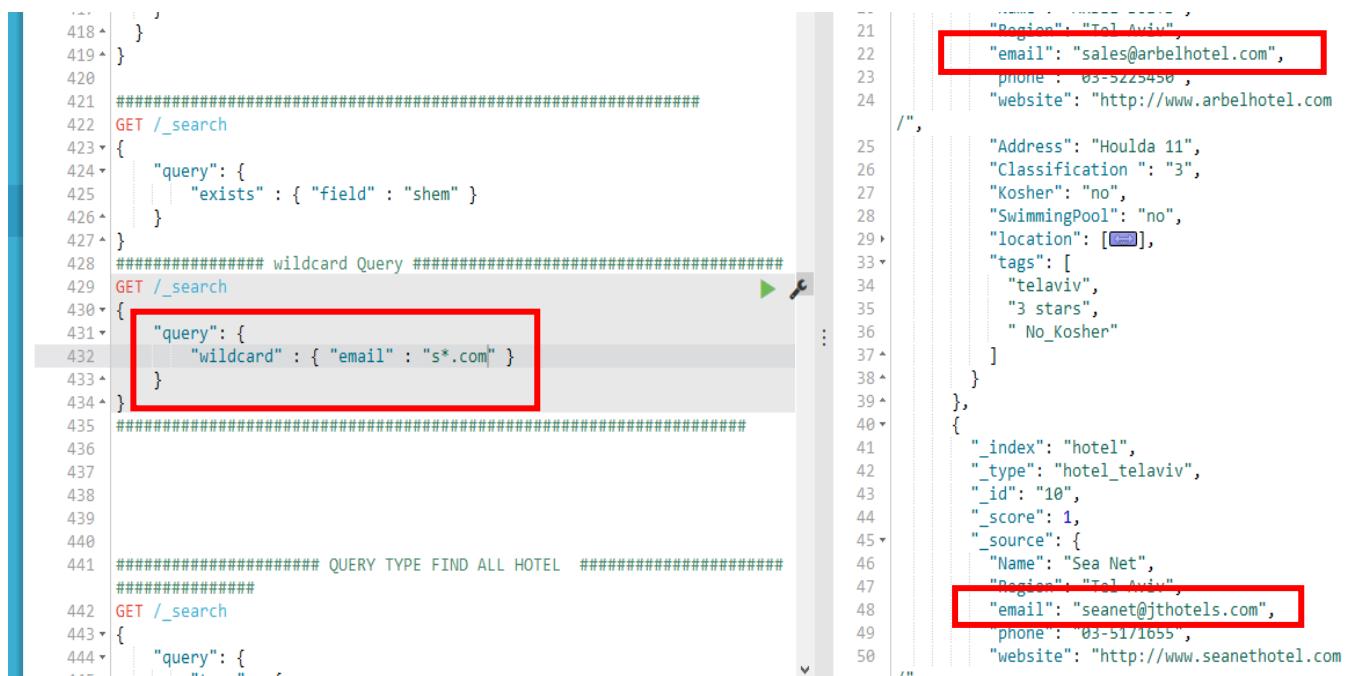


א/or 45 שאלות מונח

בדוגמא לעיל, כאשר מזינים את "שאילתת-המונה", עברר "סח" והמפתח שהוא שדה "Kosher", במטרה למצוא מסמכים במאגר אשר מכילים את הערך "סח" נמצא מספר תוצאות שמכילות את הערך בשדה Kosher. ניתן לראות, שרוב המסמכים שנמצאו, הם אנדקס' Hotel אשר מכילים בשדותיהם את הערך "סח" (הכוונה לא כשרים).

#### Wildcard Query 7.4.1.4

השאילתת זו מוצאת מסמכים הכללים שdots שתואמים תווים כללים לא מוגדרים (לא מנותח). התווים הכלליים שנתמכים הם "\*" אשר תואם כל רצף תווים ו "?" אשר תואם כל תו בודד. שימוש לב Ci השאילתת זו יכולה להיות איטית, שכן היא צריכה לחזור על פני מונחים רבים. כדי למנוע שאילתות מאוד איטיות, מונחים כללים לא צריכים להתחילה עם אחד מהתווים כגון "\*" או "?" (אחרת השאילתת תחפש את כל האפשרויות).



```

418 }
419 }
420
421 #####
422 GET /_search
423 {
424   "query": {
425     "exists" : { "field" : "shem" }
426   }
427 }
428 #####
429 GET /_search
430 {
431   "query": {
432     "wildcard" : { "email" : "s*.com" }
433   }
434 }
435 #####
436
437
438
439
440
441 ##### QUERY TYPE FIND ALL HOTEL #####
442 #####
443 GET /_search
444 {
445   "query": {
446     "term" :

```

```

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

איור 46 שאילתת חיפוש טקסט על ידי השלמה ביטוי

בדוגמא לעיל, אשר מדובר בשאילתת wildcards, השאילתת זו מורכבת משני חלקים: מהערך, אשר יהיה מורכב מביטוי רגולרי/משפט אשר לא דע קטיעים ממנו ונשלים עם התו "\*" אשר מחליף את כל הריצפים של תווים אפשריים ומפתח שהוא שדה "email" בדוגמה זו. לאחר ביצוע השאילתת קיבלנו מספר תוצאות שמכילות את הביטוי רגולרי/משפט שהוא מתחילה עם האות "s" ומסתיים עם ".com". ובניהם רצף של תווים שונה. כפי שניתן לראות שרוב המסמכים שהוא מצא במאגר אשר מכילים בשדותיהם "email", הם אנדקס' Hotel אשר שמותיהם של הכתובות מייל שונים. השאילתת זו שימושית מאוד במקרים מסוימים.

## type query 7.4.1.5

שאילתה זו מסננת מסמכים התואמים לשוג המסר/המייפוי שסופק.

The screenshot shows a code editor with two panes. The left pane contains a JSON search query:

```
395
396    }
397  }
398 }
399 #####
400 GET /_search
401 {
402   "query": {
403     "type" : {
404       "value" : "hotel_telaviv"
405     }
406   }
407 }
408 #####
409 ######      Sorting By Distance      #####
410 #####
```

The value "hotel\_telaviv" is highlighted with a blue box. The right pane shows the search results as a JSON object:

```
10  "hits": {
11    "total": 10,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "hotel",
16        "_type": "hotel_telaviv",
17        "_id": "5",
18        "_score": 1,
19        "_source": {
20          "Name": "Dan Tel Aviv",
21          "Region": "Tel Aviv",
22          "email": "dantelaviv@danhoteles
.com",
23          "phone": "03-5202525",
24          "website": "http://www.danhoteels
.com",
25          "Address": "AYARKON 22"
```

The result for \_id "5" is highlighted with a red box, showing the document details: Name: "Dan Tel Aviv", Region: "Tel Aviv", email: "dantelaviv@danhoteles.com", phone: "03-5202525", website: "http://www.danhoteels.com", and Address: "AYARKON 22".

**איור 47 שאילתת סינון לפי טיפוס המשמה**

בוגר לדוגמה, אנו חיפשנו מילים שהם מואתו סוג. שאלתה זו מורכבת משני חלקים: מהערך, אשר יהיה שם של סוג המספר שאננו מנסים לאחזר ומפתח שהוא שדה "value" כברירת מחדל. לאחר ביצוע השאלה קיבלנו מספר מסמכים שמקילות בדיקות מסוג זה ולא אחר.

## Range Query 7.4.1.6

מוצאים ממכים עם שדות בעלי מונחים בטוויה מסוימים. סוג השאלה של Lucene תלוי בסוג השדה, עבור שדות Numeric Range Query הנקרא TermRangeQuery, ואילו בשדות מספר / תאריך, השאלה היא .

**איור 48** שאלת זו מוצאת לפי טווח המחרים של תיקון WIFI

Elasticsearch תומך בשני סוגי של נתונים גיאוגרפיים: שדות **geo\_point** התומכים בזוגות קווי אוֹרְקִי ושדות **geo\_shape** התומכים בנקודות, קוּים, מעגלים, מצולעים, רב-מצולעים וכו'". השאלות בקבוצה זו הן:

- geo\_distance** מוצא מסמך עם נקודות גיאוגרפיות בתחום המרחק שצוין של נקודה מרכזית.
- geo\_distance\_range** כמו השאלה geo\_distance, אבל טווח מתחיל במרחב מסוים מן הנקודה המרכזית.
- geo\_bounding\_box** חיפוש מסמכים עם נקודות גיאוגרפיות שנoverlap לתחומי המלבן שצוין.
- geo\_polygon** מצא מסמכים עם נקודות גיאוגרפיות בתחום המצלע שצוין.

#### Geo point 7.4.2.1

שדות הטיפוס של **geo\_point** מקבל זוגות קוּי אוֹרְקִי, אשר ניתן להשתמש בהם:

- כדי למצוא נקודות גיאוגרפיות בתחום תיבת תחמת, מרחק מסוים של נקודה מרכזית, או בתחום מצולע.
- כדי לצבור מסמכים בצורה גיאוגרפית או לפי מרחק מנקודה מרכזית.
- לשלב את המרחק הציין של הרלונטיות של מסמך.
- מיון מסמכים לפי מרחק.

```
PUT ImageGeoPoint
{
  "mappings": {
    "image": {
      "properties": {
        "location": {
          "type": "geo_point"
        }
      }
    }
  }
}
```

בשלב ראשון נצטרך למפות את האינדקס שאנו יוצרים, כדי להוסיף במאפיינים שלו (properties) את המאפיין location ולציין שהוא מסוג **geo\_point**. בסחות זה נוכל לתשאל אילו שאלות גיאוגרפיות.

```
PUT ImageGeoPoint/image/1
{
  "text": "Geo-point as an object",
  "location": {
    "lat": 41.12,
    "lon": -71.34 ①
  }
}

PUT ImageGeoPoint/image/2
{
  "text": "Geo-point as a string",
  "location": "41.12,-71.34" ②
}
```

```
PUT VideoGeopoint/video/3
{
  "text": "Geo-point as a geohash",
  "location": "drm3bt3ev3e86" ③
}

PUT TextGeopoint/text/4
{
  "text": "Geo-point as an array",
  "location": [ -71.34, 41.12 ] ④
}
```

שנארבע דרכי לציין נקודה גיאוגרפית, כפי שמוצג להלן :

**1** נקודה גיאוגרפית מבוטא כאובייקט, עם מפתחות **Lat** ו- **Ion**.

**2** נקודה גיאוגרפית מבוטא כמחרוזת בפורמט: **"lat, lon"**.

**3** נקודה גיאוגרפית akan מבוטא כפורמט **Geohash**. Geohash היא דרך נוחה להביע מיקום (בכל מקום בעולם) באמצעות מחרוזת אלפאנומרית קצרה, עם דיקרב יותר עם מחרוזות ארוכות יותר).

**4** נקודה גיאוגרפית מבוטא כמערך עם הפורמט **[lon, lat]**.

**5** שאלתת התיבה הגיאוגרפית תמצא את כל הנקודות הגיאוגרפיות הנפלות בתחום התיבה (תחום גאוגרפי שוגדר ע"י קואורדינטות של המשתמש).

```
GET ImageGeoPoint/_search
{
  "query": {
    "geo_bounding_box": {
      "location": {
        "top left": {
          "lat": 42,
          "lon": -72
        },
        "bottom right": {
          "lat": 40,
          "lon": -74
        }
      }
    }
  }
}
```

#### Geo Distance sorting 7.4.2.2

שאילתת זו מאפשרת מין גיאוגרפי למרחק, באמצעות שאלתת geo distance. דוגמא, בהנחתה שהceneנו למסמכים שלנו במאפיין את המאפיין הבא location.pin, שהוא שדה מסוג geo\_point אשר מכיל את הקואורדינטות בהתאם עבור כל מסמך. בדוגמה להלן, השאלתת מכילה את הקואורדינטות אשר מהם יתחל הדירוג של המסמכים בהתאם לנקודה הזו בשדה location. שאלתת זו ניתן גם לציין באיזה יחידות תיחשב הדירוג (km,m), או אם נרצה שהדירוג יהיה עולה או יורם (asc,desc) כפי שניתן לראות התוצאות בסדר עולה.



```

1 ######      Sorting By Distance      #####
2
3 GET /_search
4 {
5   "sort" : [
6     {
7       "_geo_distance" : {
8         "location" : [34.770225,32.0658416],
9         "order" : "asc",
10        "unit" : "m",
11        "mode" : "min",
12        "distance_type" : "arc"
13      }
14    }
15  ],
16  "query" : {
17    "term" : { "Region": "Tel Aviv" }
18  }
19 }
20 ###### in result we have sort:0.82334 =0.823km #####
21
22 ###### TRY 1 essaie #####
23 ###### find all document wifi point from a geopoint #####
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113

```

איור 49 שאלה מין מסומנים לפי הטווח שלהם מנוקודה

#### Geo\_distance query 7.4.2.3

שאילתת זו מסננת את המסמכים אשר התוצאות שלהם ממוקמות למרחק מסוים מנוקודה גיאוגרפית שבוחרים. מילאתי את המיפוי והאינדקס של מסמך הבא:  
לדוגמא: אינדקס של משחח פשוט עם נקודת מקום "Pin" צריך להיות מוגדר גם במיפוי שלו וגם הכנסת האינדקס



```

PUT /hotel_telaviv
{
  "mappings": {
    "location": {
      "properties": {
        "pin": {
          "properties": {
            "location": {
              "type": "geo_point"
            }
          }
        }
      }
    }
  }
}

PUT / hotel_telaviv /location/1
{
  "pin": {
    "location" : {
      "lat" : 40.12,
      "lon" : -71.34
    }
  }
}

```

לאחר מכן ניתן לבצע את השאלה הפשוטה הבאה באמצעות מסנן geo\_distance:

```
GET /hotel_telaviv/location/_search
{
  "query": {
    "bool": {
      "must": [
        "match_all": {}
      ],
      "filter": {
        "geo_distance": {
          "distance": "200km",
          "pin. Location" : {
            "lat" : 40,
            "lon" : -70
          }
        }
      }
    }
  }
}
```

פורמטים מקובלים:

באוטו אופן שבו סוג pointGeo יכול לקבל ייצוגים שונים של נקודה גיאוגרפית, המסנן יכול לקבל את זה גם:

```
"pin. Location" : {
  "lat" : 40,
  "lon" : -70
}
}

"filter": {
  "geo_distance": {
    "distance": "12km",
    "pin. Location": [-70, 40]
  }
}
```

קו אופקי ואנכי בתור מאפיינים:

קו אופקי ואנכי כמערך בפורמט [lat, lon] -כאן CD' להתאים לחסן GeoJson.

מיקום מרובה למסמר geo\_distance יכול לעבוד עם מספר מיקומים / נקודות למסמר ברגע שמייקום / נקודה מתאימים למסנן.



```
#####
# wifipoint/telaviv/_search?
GET /_search
{
  "query": {
    "bool": {
      "must": [
        "match_all": {}
      ],
      "filter": {
        "geo_distance": {
          "distance": "2000km",
          "pin.location": {
            "lat": 32.0800334,
            "lon": 34.7665656
          }
        }
      }
    }
  }
}

#####
PUT /my_locations
{
  "mappings": {
```

hits": [
 {
 "\_index": "wifipoint",
 "\_type": "telaviv",
 "\_id": "5",
 "\_score": 1,
 "\_source": { }
 },
 {
 "\_index": "wifipoint",
 "\_type": "telaviv",
 "\_id": "2",
 "\_score": 1,
 "\_source": { }
 },
 {
 "\_index": "wifipoint",
 "\_type": "telaviv",
 "\_id": "4",
 "\_score": 1,
 "\_source": { }
 },
 {
 "\_index": "wifipoint",
 "\_type": "telaviv",
 "\_id": "1",
 "\_score": 1,
 "\_source": { }
 }
]

אייר 50 מסנן מוכנים מרוביים לפי שאלה GEOdistance

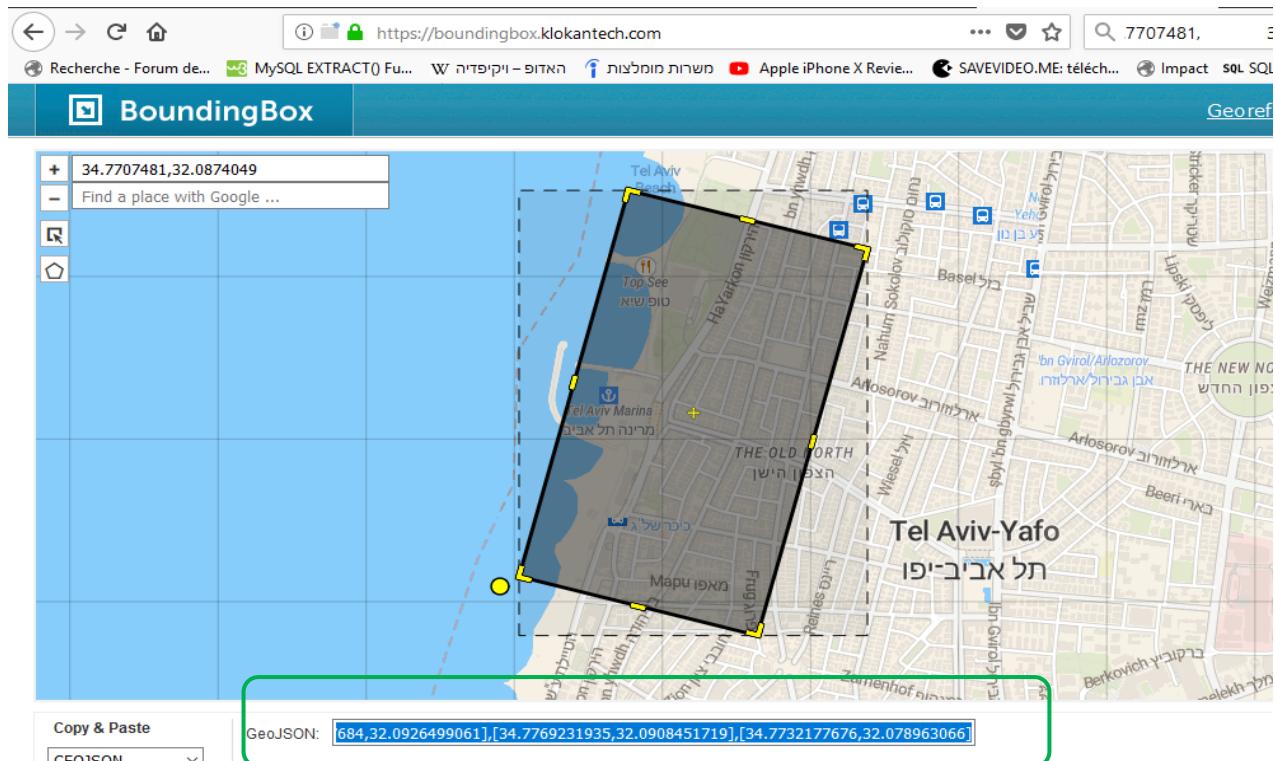
## Geo Distance Query 7.4.2.4

מסננים מסמכים שהיימם בטוח מנקודה מסוימת

```
GET /_search
{
  "query": {
    "bool": {
      "must": {
        "match_all": {}
      },
      "filter": {
        "geo_distance_range": {
          "from": "50km",
          "to": "500km",
          "pin.location" : {
            "lat" : 38,
            "lon" : -74
          }
        }
      }
    }
  }
}
```

## Geo Bounding Box Query 7.4.2.5

שאילתת המאפשרת סינון מסמכים אשר נמצאים במאגר נתונים ובודקת התאמות אשר מtabססות על מיקום של נקודות הגיאוגרפיות של המסמכים באמצעות תיבת התוחמתה. היא תמצא את כל הנקודות הגיאוגרפיות הנופלות בתחום התיבה, לצורך ההדגמה, סיפקנו לעצמנו קואורדינטות של תיבת התוחמתה של תל אביב באתר בשם [GeoJson boundingbox.com](https://boundingbox.klokantech.com).



איור 51 אתר ה [boundingbox.com](https://boundingbox.klokantech.com)

לאחר הזנה של קואורדינטות לטור השאלתה החזירה לנ-31 מסמכים שונים שנופלים בתחום הספציפי זהה.



```

{
  "query": {
    "bool": {
      "must": [
        "match_all": {}
      ],
      "filter": {
        "geo_polygon": {
          "location": [
            {
              "points": [
                [32.029289,34.73901],
                [32.029289,34.851841],
                [32.146805,34.851841],
                [32.146805,34.73901],
                [32.029289,34.73901]
              ]
            }
          ]
        }
      }
    }
  }
}

#####
# Bounding Box Tel Aviv 1 #####
GET /_search
{
  "query": {
    "bool": {
      "must": [
        "match_all": {}
      ],
      "filter": {
        "geo_polygon": {
          "location": [
            {
              "points": [
                [32.029289,34.73901],
                [32.029289,34.851841],
                [32.146805,34.851841],
                [32.146805,34.73901],
                [32.029289,34.73901]
              ]
            }
          ]
        }
      }
    }
  }
}

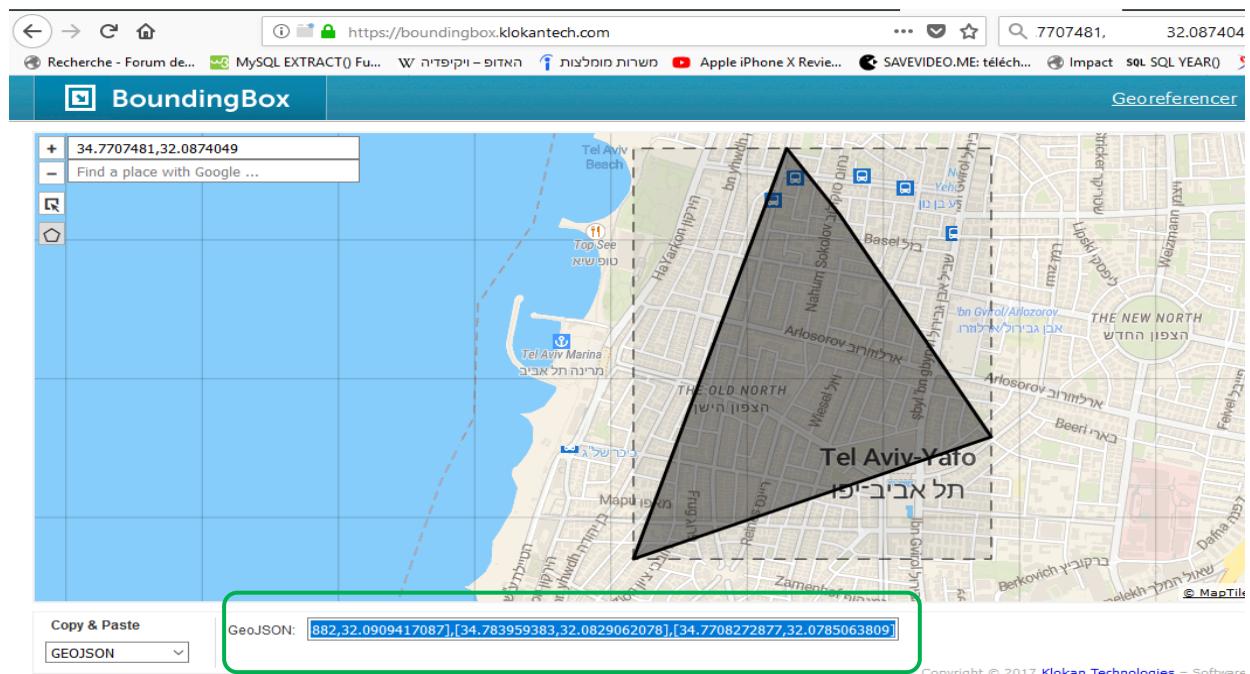
```

The screenshot shows a Kibana search interface. On the left, the search query is displayed, including a geo\_polygon filter for a bounding box around Tel Aviv. On the right, the search results are shown, with two documents highlighted by red boxes. The first result is from the 'theatres' index, and the second is from the 'wifipoint' index. Both results have a score of 1.

איור 52 שאלת סינון מושגים על ידי תבנה התוחמת.

#### Geo polygon query 7.4.2.6

שאילתת זו מtabססת על אותו עיקנון של האxo**bounding box**,היא תכלול התאמות שימושיות שמשתיכות רק למצולע של נקודות. להלן דוגמא: היא תמצוא את כל הנקודות הגיאוגרפיות הנופלות בתחום המצלע, לצורך ההדגמה, סיפקנו לעצמנו קואורדינטות של מצלע של תל אביב באתר בשם [bbox.com](#) בפורמט [GeoJson](#).



איור 53 אשפרות לקבל קואורדינטות מה מצלע

לאחר הזנה של קואורדינטות לערך השאלתה כפי שמתואר להלן, השאלה החזירה לנו מספר רב של מסמכים שונים שנופלים בתחום של המצלע זהה. קיבלנו אינדקסים שונים לחלוטן.

```

57 GET /_search
58 {
59   "query": {
60     "bool": {
61       "must": {
62         "match_all": {}
63       },
64       "filter": {
65         "geo_polygon": {
66           "location": {
67             "points": [
68               [34.7732177676,32.078963066],
69               [34.7651553425,32.0807680348],
70               [34.7688607684,32.0926499061],
71               [34.7769231935,32.0908451719],
72               [34.7732177676,32.078963066]
73             ]
74           }
75         }
76       }
77     }
78   }
79 }
80 ###### Bounding Box Tel Aviv 2 center #####
81
82
83 GET hotel/hotel_telaviv/_search
  
```

איור 54 חיפוש נתונים לפי תיכון גאוגרפי של ת"א על ידי קואורדינטות המצלע

## ELASTICSEARCH —KIBANA 8



התפקיד העיקרי של Kibana הינו בהציג המידע בצורה תרשימים וגרפים, הוא מסתמך על כל הכוח של ElasticSearch כדי לפעול בפשטות ובייעילות את כל נתוניםיו. Kibana מאפשר לצרף את נתוניםיו לתוך תרשימים וגרפים שיכולים ליצור לוח מחוונים (Dashboard) דינמי או אינטראקטיבי. ניתן בקלות לעבור מבניית דוחות פשוטות לפיתוח לסכימות מורכבות עם שימוש בתמורות מתמטיות או מסננים לפי היבטים(facets). התכונות עיקריות של Kibana הן:

- ✚ ייצור דוחות: טבלאות, גרפים, גיאוגרפיים.
- ✚ ייצור Dashboard.
- ✚ עבודה על נתונים בזמן אמת.
- ✚ יכולת ליצור אוסףים ומסננים.
- ✚ חילוץ מידע ועוד.

ה Kibana מחולק לחמש חלקים עיקריים שנרחיב עליהם בפרק הבא.

## Console Dev Tool 8.1

דף זה מכיל כלים המאפשרים אינטראקציה עם הנתונים ב-KIBANA (CONSOLE) התוסף מסויף (CONSOLE) אשר מספק ממשק משתמש כדי לקיים אינטראקציה עם API REST של ELASTICSEARCH. ב��וף יש שני תחומים עיקריים: העורך, שבו המשתמש מחבר בקשות ל-ELASTICSEARCH וחילונית התגובה, המציגת את התשובות לבקשתו.



The screenshot shows the Kibana interface with the 'Dev Tools' sidebar open. The 'Console' tab is active, displaying a code editor with two red boxes highlighting specific sections of the code. The first red box encloses the following code block:

```

156     "sum_t" : { "sum" : { "field" : "price" } }
157   }
158 }
159 #####
160 ##### Cardinality Aggregation HOW MANY DIFFERENT FIELDS
161 POST /wifipoint/_search?size=0
162 {
163   "aggs" : {
164     "type_count" : {
165       "cardinality" : {
166         "field" : "mispar_takala"
167       }
168     }
169   }
170 }
171 #####
172 ##### AGGREGATION THAT COUNT HOW MANY WIFI POINT HAVE TAKALA
173 #####
174 POST /wifipoint/_search?size=0
175 {
176   "aggs" : {
177     "count_mispar_takala" : { "value_count" : { "field" :
178       "mispar_takala" } }
179   }
180 }

```

The second red box encloses the resulting JSON response from the Elasticsearch query:

```

1 {
2   "took": 8,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10   "hits": {
11     "total": 5,
12     "max_score": 0,
13     "hits": []
14   },
15   "aggregations": {
16     "type_count": {
17       "value": 4
18     }
19   }
20 }

```

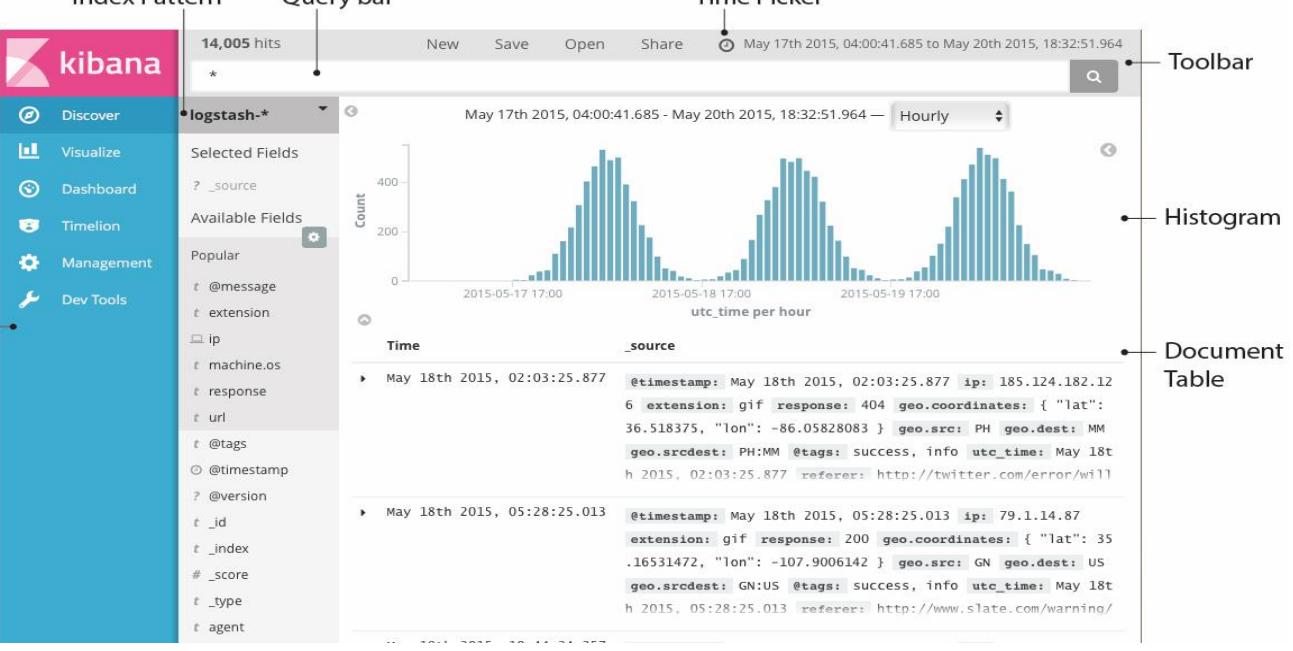
Below the screenshot, two red boxes contain explanatory text:

בצד זה מופיע כל התוצאות לאחר שהמשתמש הריץ את השאלות

בעד זה המשתמש רושם את השאלות

## Discover 8.2

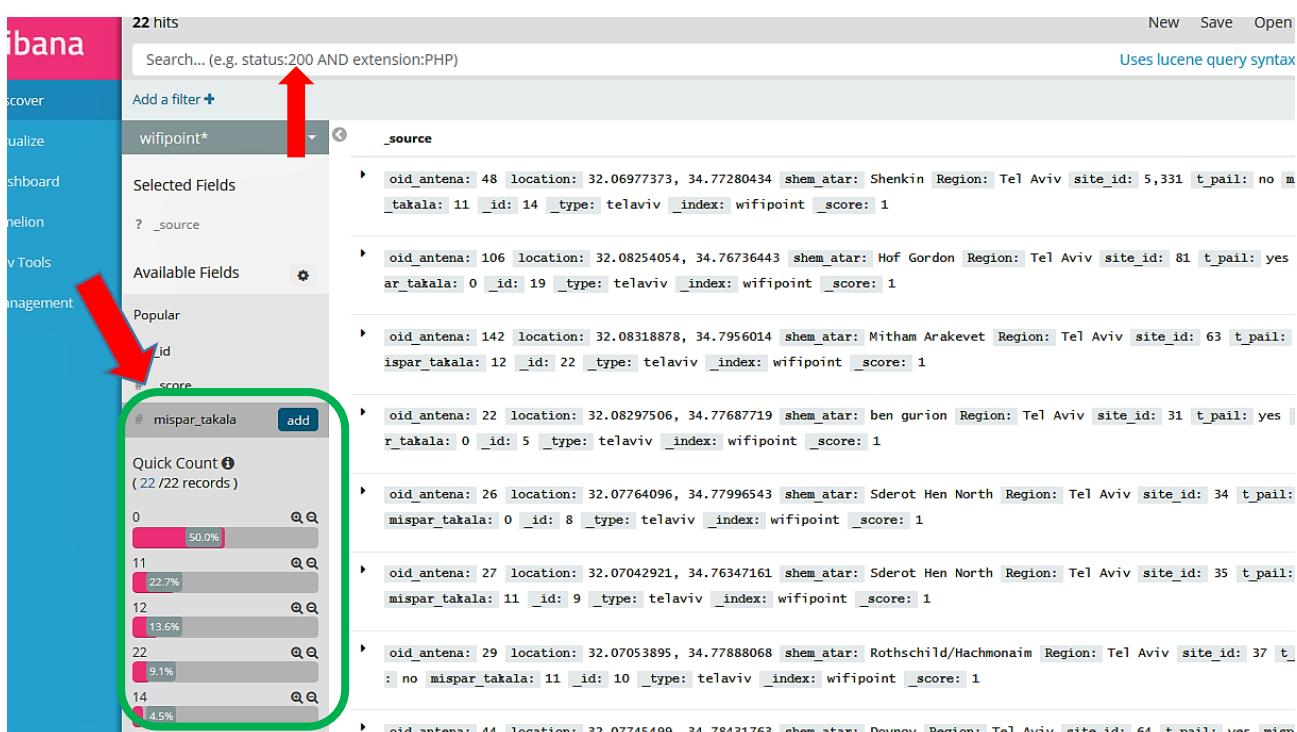
הדף DISCOVER קריית נתונים יכולה להיות אינטראקטיבית. בדף זה, יש לנו גישה לכל המסמכים בכל אינדקס המתאים למבנה האינדקס שנבחרה. בנוסף לזה ניתן לשולח שאלות חיפוש, לסנן תוצאות חיפוש ולהציג נתונים מסוימים. יש גם הדרישה של מספר המסמכים לשאלות החיפוש ולסיטוטים קות של ערכי שדות. יתר על כן אם שדה זמן מוגדר עבור מודל האינדקס שנבחר, התפלגות המסמכים בזמן מוצגת בהיסטוגרמה בחלק העליון של הדף.



The screenshot shows the Kibana Discover interface. On the left is a sidebar navigation with links to Discover, Visualize, Dashboard, Timelion, Management, and Dev Tools. The main area has three sections: Index Pattern (set to logstash-\*), Query bar (with a dropdown showing 14,005 hits and a search bar containing an asterisk), and Time Picker (set to May 17th 2015, 04:00:41.685 to May 20th 2015, 18:32:51.964). A Toolbar includes New, Save, Open, Share, and a magnifying glass icon. To the right is a Histogram showing 'Count' vs 'utc\_time per hour' from May 17th to May 19th. Below the histogram is a Document Table with two rows of log entries. Labels on the right side point to the Toolbar, Histogram, and Document Table.

איור 55 דף הDiscovery

בחילון זה תוכלו גם לקבל את "שיעור הרלוונטיות" של החיפושים שלכם כשתייחסו על אחד מהנתונים אשר נמצא בחילון השמאלי. בדוגמא להלן, קיבלנו באמצעות החיפושים של הנתון של מספר תקלות WIFI בת"א מי כל החיפושים שביצענו. בנוסף לזה, תוכלו לחפש את נתונים שלכם בצורה מהירה מאד באמצעות הסרגל חיפוש אשר משתמש בשאלות של Lucene, וגם תוכלו ליצור רשימות נתונים בהםם תוכלו לסנן את נתונים שתרצו לשימושם לתוך dashboard.

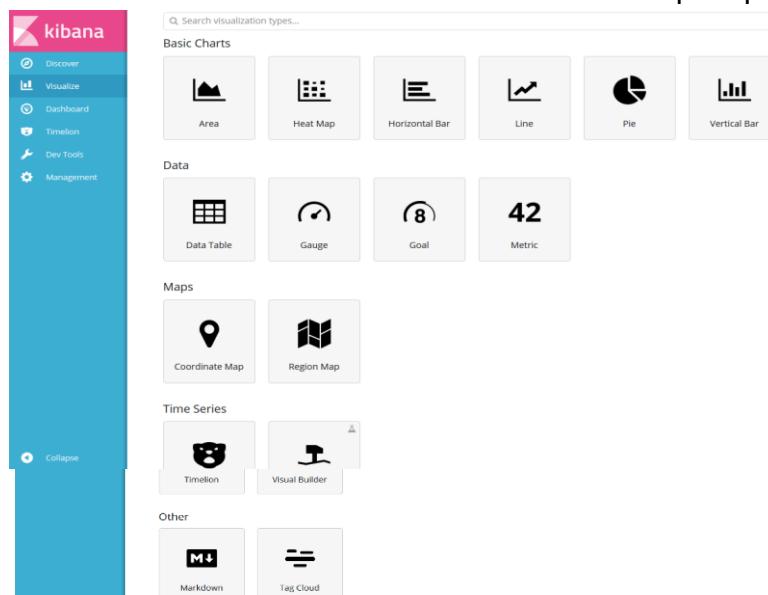


The screenshot shows the Kibana Visualize interface. At the top is a search bar with the placeholder 'Search... (e.g. status:200 AND extension:PHP)'. Below it is a 'Selected Fields' dropdown set to 'wifipoint\*' and an 'Add a filter +' button. A red arrow points to the search bar. In the bottom-left corner, there is a green-highlighted 'Quick Count' chart titled '(22/22 records)' showing the distribution of 'mispar\_takala' values. The chart has six bars with the following data: 0 (50.0%), 11 (22.7%), 12 (13.6%), 22 (9.1%), 14 (4.5%), and 1 (0.0%). The main area displays a list of '\_source' documents, each with fields like 'oid\_antena', 'location', 'shem\_atar', 'Region', 'site\_id', 't\_pail', and 'mispar\_takala'. A red arrow points to the 'Selected Fields' dropdown.

איור 56 דף הDiscovery עם המרכיבים שלו

## Visualize 8.3

הויזואלייזציה הוא המקום שבו נוכל ליצור, לשנות ולהציג הדמויות מותאמות אישית. ישנו מספר סוגים שונים של ויזואלייזציות החל מגրפים ליניארים, לען של נקודות, להיסטוגרמות, לתרשי עוגה, למפות מרוצפות (להציג נתונים על מפה) ולטבלאות נתונים ועוד. בנוסף לזה ניתן לשף ויזואלייזציות גם עם משתמשים אחרים שיש להם גישה למופע KIBANA של המשתף.

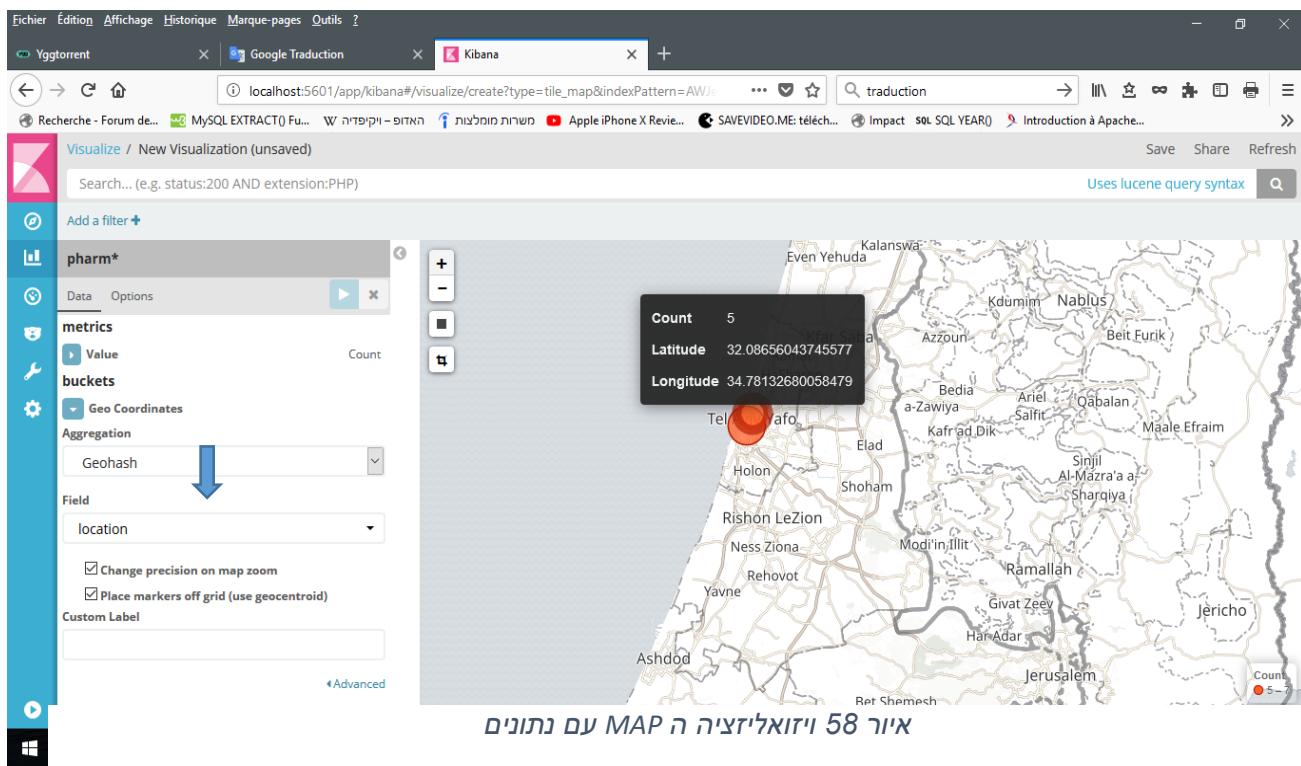


ויזואלייזציות השונות הן:

- Line, Area, and Bar charts
- Pie Charts
- Region Maps
- Heatmap Chart
- Data Table
- Coordinate Maps
- Tag Clouds
- Metric

איור 57 דף ה VISUALIZE והתקנות השונות

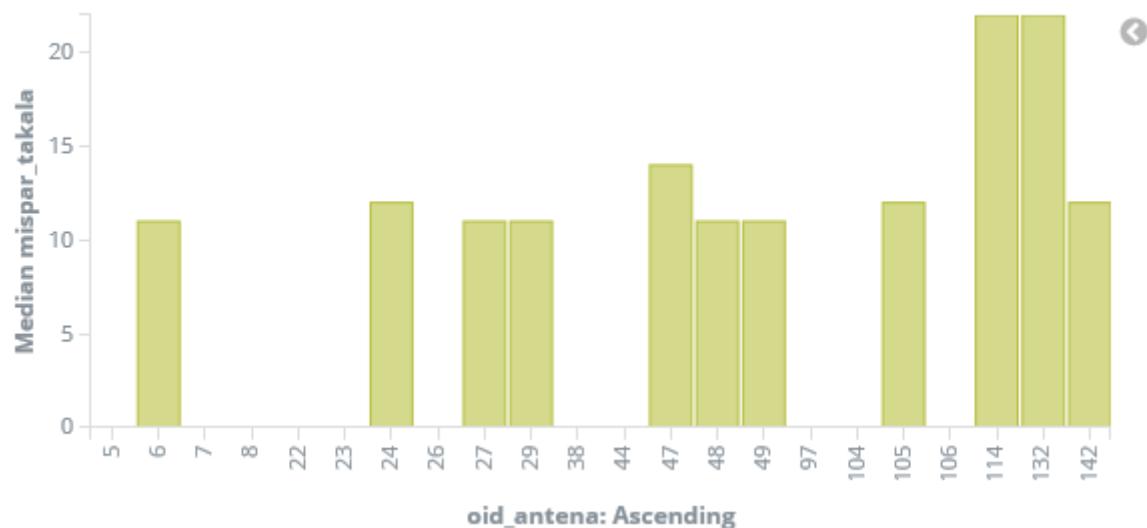
נדגים בכמה ויזואלייזציות השונות: ויזואלייזציה ה "Coordinate Map" מאפשרת להציג את נתונים על המפה. כאן, בדוגמה להלן מצאנו 5 בתים מrank חסונים באותו אזור אשר חיפשנו עם Elasticsearch.



איור 58 ויזואלייזציה ה MAP עם נתונים

לאחר שביצינו את השאלות ב Elasticsearch, ויזואלייזציה של «Bar Charts» מאפשרת להציג את הנתונים כפי שניתן ליראות להן. כאן, בדוגמה להלן מצאנו לכל נקודה WiFi את כמות התקלות של כל אחד מהם בתוך המאגר Elasticsearch.

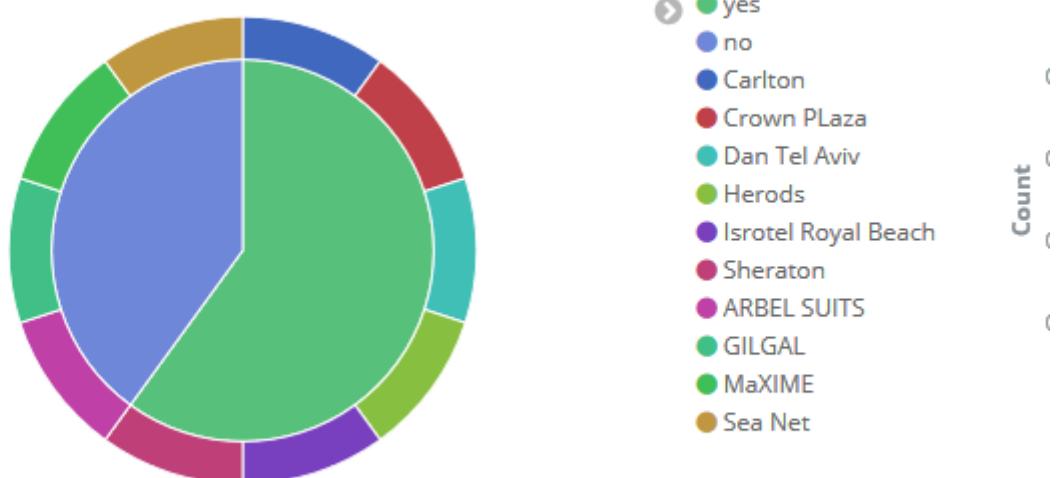
visualization wifi id /mispar takala



איור 59 ויזואלייזית כמות התקלות WiFi על ידי BAR CHART

ויזואלייזיה של "Pie Charts" מאפשרת להציג נתונים בצורה עוגה. כאן, בדוגמה להלן מצאנו לכל בית מלון בתל אביב והסבירה האם הוא שומר כשרות או לא.

pie chart hotel casher 3



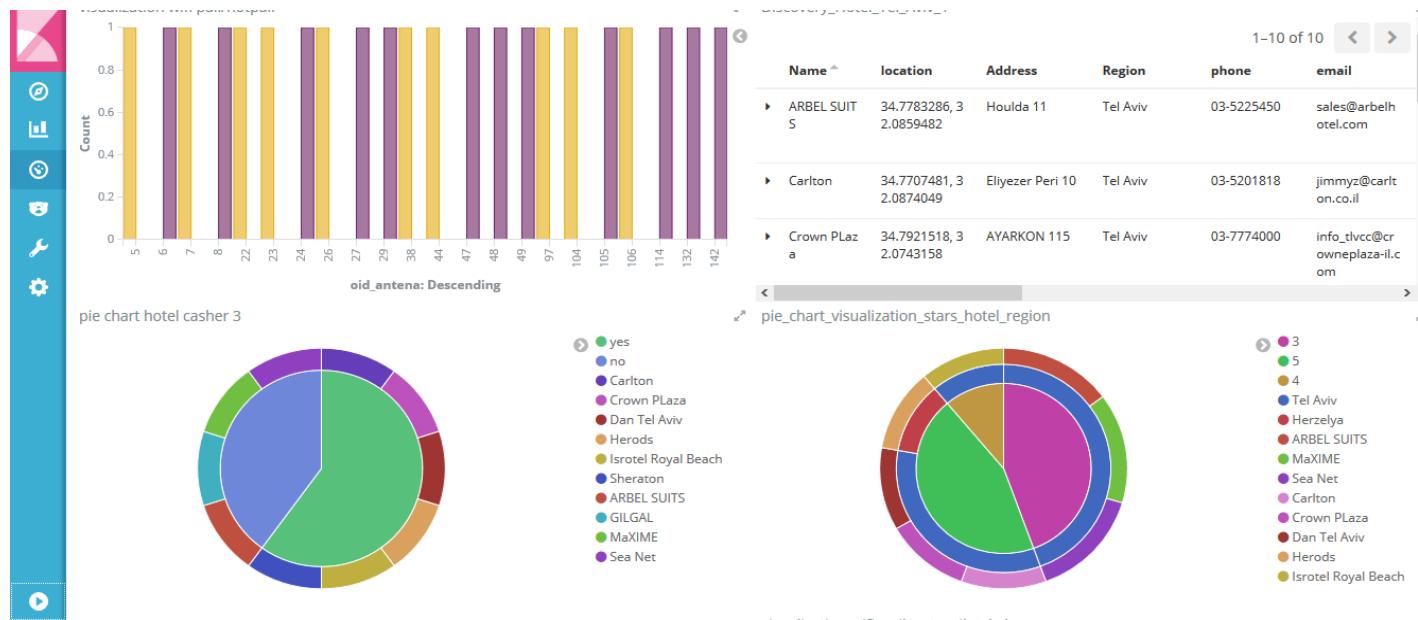
visualization wifi id /mispar takala

איור 60 ויזואלייזית ה PIE CHART של נתונים בתי מלון

## Dashboard 8.4

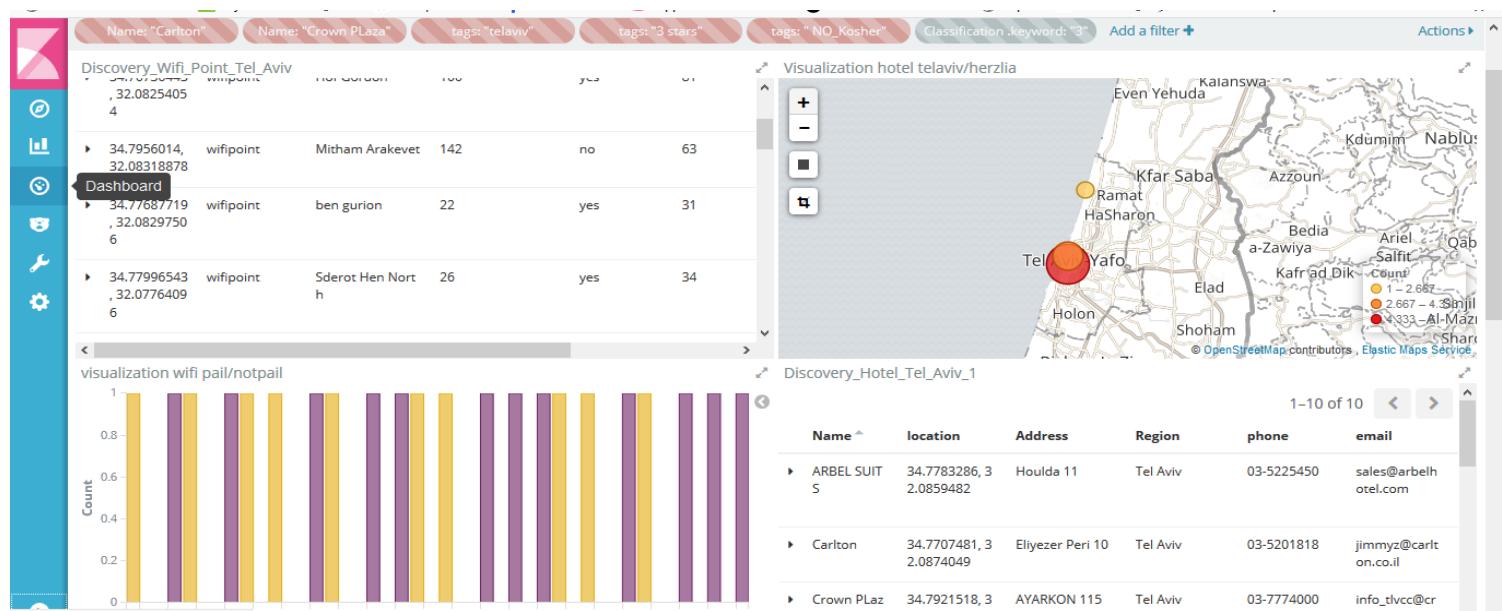
להלן מוחונים (DASHBOARD) של KIBANA (DASHBOAR) הוא המקום שבו נוכל ליצור, לשנות ולהציג לוחות מוחונים מותאמים אישית שלנו. במציאות לוח מוחונים, נוכל לשלב מספר רב של ויזואלייזציות בדף אחד, ולאחר מכן לסנן אותם על ידי מתן שאלת חיפוש או על ידי בחירת מסננים על ידי לחיצה על האלמנטים שבהדמיה. לוח מוחונים שימושיים כאשר ברצוננו לקבל סקירה כללית של הימנין LOGS (התאמות) בין הדמויות ויומנין שונים.

. להלן דוגמה של הרכבות השונות שניתן לעשות עם ה Dashboard.



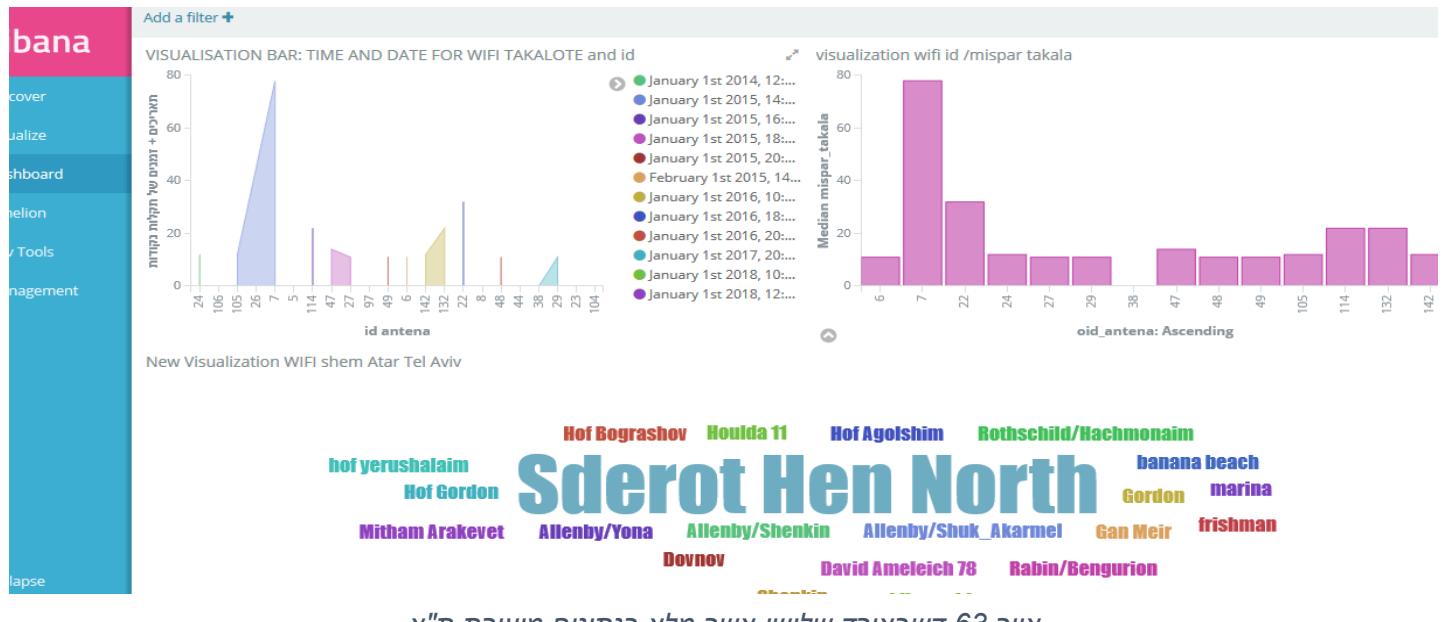
איך 61 דashboard אשר מלא נתונים מייערת ת"א

.Pie Charts של רשימת נתונים של בתי מלון מטבלת Discovery ותרשיימי



איך 62 דashboard שני אשר מלא נתונים מייערת ת"א

כאן, תמונה זו מראה dashboard מורכב משילוב של רשימה נתונים של בתים מלון ונקודות WiFi בתל אביב מטבלת . Coordinate Map Discovery



איור 63 דשאורד שלישי אשר מלא בנתונים מעירת ת"א

תמונה זו מראה dashboard מורכב משילוב של היסטוגרמה על תקלות של WiFi ווש בתל אביב ותרשיים של ענ"ק תגיות אשר מציג את חשיבותם וכפילות השמות ביחס לגודלם בענ"ק.

## 9 סיכום

המחקר על Big data שנעשה כחלק מפרויקט גמר לצורך קבלת תואר מהנדס תוכנה חישב בפניו את כל הכלים הטכנולוגיים המשמשים לעיבוד וניתוח מידע מתוך מאגרי נתונים המכילים מידע עתיק.

כלי החיפוש האלו כדוגמת Cassandra, MongoDB, Hadoop, MySQL ועוד אשר נועגות בתחום של ה Big Data מספקות תוצאות מאוד מהירות מבחינת הביצועים ביחס לחיפוש נתונים בסיסי נתונים עתיק ומורכבים ממגוון רחב של נתונים. מה שלכורה בסיסי נתונים הסטנדרטים בكمויות עתיק אינם אפשריים.

בחוק מהמחקר של הפרויקט נחשפה לעולם של Big Data, ניסיתי כמה שיותר להסביר את מרכיבותו והאתגרים השונים אשר מהנדסים נתקלים בהם ומנסים להבין אותם. אם זה למשל להוציא ממגר נתונים של חברה ערך עסקי לחברה עצמה או אם זה לנתח או לשפר את ביצועי שירותיהם ע"י ניתוח קבצי לוג. ביחד עם זה, גיליתי כלים חדשים אשר מפורטים בגוף מסמך זה. כיוון כלים אלואפשרים עבודה בתעשייה ההיברידית מתחתיות ה Hadoop, מודל הנטונים החדשן עד למאגר נתונים NOSQL . דוגמא : כל הכלים האלה יכולם לספק ניתוח מדויק מאוד לצורך שיפור באיכות השירות עבור לקוחות החברה.

הגעתנו למסקנה כי ניתוח והבנה של Big Data היום, מספק המבנה טובה יותר של מה שהחברות רוצחות להשיג ביחס ללקוחות שלהם בכל התחומיים העסקיים.

ביחד עם זה בחלק הסייעיה, ניסיתי לתאר כמה שיותר את הפוטנציאלי האדיר של הכלי Elasticsearch, מחייש טקסט מורכב או מונחים עד ניתוחים גאוגרפיים מורכבים. הסקתי מסקנה שמערכות הקוד פשוטה ומאפשרת עבודה נוכה לכתיבת שאלות שאלות שאנו צריכים בטור מאגר של Elasticsearch.

בנוסף לכך, הריאתי שכל KIBANA מאפשר הצגת התוצאות בצורה גרפית אינטראקטיבית המאפשרת למשתמש לנתח בצורה קלה וברורה כל נתון.

לסיכום הממחקר והסימולציה שבצעתי אמורים לתרום להבנה והכרה טובה יותר של שיטות שונות בחיפוש.

## 10 הפניות

1. **Gartner**. - <https://www.gartner.com/technology/topics/data-analytics.jsp>.
2. <http://xsnet.com>. - <https://www.xsnet.com/blog/bid/205405/the-v-s-of-big-data-velocity-volume-value-varietiy-and-veracity>.
3. **hrboss.com** - 26 03 2014. - <https://hrboss.com/blog/2014-03-26/missing-vs-big-data-hr-5-v-model-here>.
4. **computerworld.com**- <https://www.computerworld.com/article/2493701/data-center/by-2020--there-will-be-5-200-gb-of-data-for-every-person-on-earth.html>.
5. **dzone.com**. - <https://dzone.com/articles/difference-between-data-warehouse-and-data-mart>.
6. **sisense.com** - <https://www.sisense.com/glossary/data-mart/>.
7. **Searchdatamanagement**- <https://searchdatamanagement.techtarget.com/definition/dimension-table>.
8. **wikipedia**. - [https://en.wikipedia.org/wiki/Snowflake\\_schema](https://en.wikipedia.org/wiki/Snowflake_schema).
9. **corejavaguru.com**. - <http://www.corejavaguru.com/bigdata/hadoop/hdfs-architecture>.
10. **sas.com**. - [https://www.sas.com/nl\\_nl/insights/big-data/hadoop.html](https://www.sas.com/nl_nl/insights/big-data/hadoop.html).
11. **ramkedem.com**. - <http://ramkedem.com/mapreduce-%D7%95-hdfs-%D7%91%D7%9E%D7%99%D7%9C%D7%99%D7%9D%D7%A4%D7%A9%D7%95%D7%98%D7%95%D7%AA/>.
12. **cnblogs.com**. - <http://www.cnblogs.com/xiaoit/p/4569203.html>.
13. **mongodb.com**. - <https://docs.mongodb.com/manual/replication/>.
14. **dzone.com**. - <https://dzone.com/articles/nosql-cassandra-in-plain-english>.
15. **engineering.salesforce.com**. - <https://engineering.salesforce.com/investing-in-big-data-apache-hbase-b9d98661a66b>.
16. **mapr.com**. - <https://mapr.com/blog/in-depth-look-hbase-architecture/>.
17. **sphinxsearch.com**. - <http://sphinxsearch.com/about/sphinx/>
18. **Solr** - <http://lucene.apache.org/solr/features.html>

## 10.1ביבליוגרפיה

**Big Data Concepts, Theories, and Applications** Shui Yu Song Guo Yu, Shui Guo, So, :Springer International Publishing Cham.

**Big Data Related Technologies, Challenges and Future Prospects** Chen, M ; Mao, Shiwen ; Zhang, Yin ; Leung, Victor Chung Ming.2014 - .Springer

**Big Data and Analytic** .Morabito Vincenzo : - .Springer.2015

**Big Data A Primer.** Hrushikesha Mohanty, Prachet Bhuyan, Deepak Chenthati *Editors* :springer 2015

**Big Data and Web search engine.** BigDataBench: a Big Data Benchmark Suite fromWeb Search Engines, Institute of Computing Technology, Chinese Academy of Sciences, Wanling Gao1, Yuqing Zhu1, Zhen Jia1 2016

## 10.2רשימת איורים:

5.....	איור 1 ארכיטקטורת קישור בין מנוע חיפוש Elasticsearch של Big Data לבסיס הנתונים.
9.....	איור 2 סכמת כוכב .....
10.....	איור 3 סכמת ה Snowflakes .....
11.....	איור 4 רכיבי ארכיטקטורת ה HDFS .....
11.....	איור 5 מסדי נתונים סטנדרטיים בארגונים .....
12.....	איור 6 שלבי האלגוריתם: MAP-REDUCE: .....
15.....	איור 7 תהליכי השכפול נתונים בסיס נתונים MongoDB .....
16.....	איור 8 מרכיבי ארכיטקטורת בסיס נתונים MongoDB .....
17.....	איור 9 דוגמה לדריבוי רשומות אשר מכילות נתונים. ....
18.....	איור 10 משפחה של עמודות. ....
18.....	איור 11 תיאור גרפי של Keyspace .....
19.....	איור 12 רכיבי הארכיטקטורה Hbase .....
19.....	איור 13 רכיב ה HMaster מaszcol Hbase .....
20.....	איור 14רכיבי ה Region Server .....
24.....	איור 15 אתר של ייירת ת"א שמיימו תוכלו להויר נתונים .....
25.....	איור 16 בחרת סוג של נתונים .....
25.....	איור 17 אתר שבו תוכלו להמיר את הקובץ לפורמט JSON .....
25.....	איור 18 בחירת השדות שהו בקובץ ה JSON .....
26.....	איור 19 הורדת הקובץ הJSON .....
26.....	איור 20 אתר ה Elasticsearch .....
27.....	איור 21 הורדת ה Kibana .....
27.....	איור 22 הכנס אינדקס לתוך המאר של Elasticsearch .....
28.....	איור 23 מספר ה ID של אינדקס אחריו שהוא נוצר .....
28.....	איור 24 מקור המידע אשר נשמר לאחר אינדקס נתן .....
30.....	איור 25 מחלוקת אינדקס .....
30.....	איור 26 מחיקה של אינדקסים רבים בשאלתה אחת .....
31.....	איור 27 אינדקס לנפי עדכן .....
31.....	איור 28 אינדקס לאחר עדכן בעמצעות SCRIPT .....
31.....	איור 29 שאלהת עדכן עם SCRIPT .....
32.....	איור 30 הוספה נתן באמצעות ה SCRIPT .....
32.....	איור 31 הוספה שדה חדש לתוך הענידקס בעמצעות ה SCRIPT .....
33.....	איור 32 איחזור מרובה מסמכים .....

33.....	איור 33 שיטת אחזור מסמכים מרובים ראשונה
34.....	איור 34 שיטת אחזור מסמכים מרובים שנייה .....
35.....	איור 35 אינדקס נתוני עירית ת"א ע"י שיטת ה BULK .....
36.....	איור 36 מיפוי השדות של אינדקס .....
37.....	איור 37 שאלת מתוציא .....
38.....	איור 38 שאלת זו סופרת את כל התקנות השונות של WIFI .....
38.....	איור 39 שאלות מקסימלית .....
39.....	איור 40 שאלת המינימלית .....
39.....	איור 41 שאלת מין גאוגרפי .....
40.....	איור 42 שאלת המסננים .....
41.....	איור 43 שאלת חיפוש טקסט לפי התאמה עם השדה .....
42.....	איור 44 שאלת חיפוש שימושים מרובים באמצעות התאמה של השדות .....
42.....	איור 45 שאלת מונח .....
43.....	איור 46 שאלת חיפוש טקסט על ידי השלמת ביטוי .....
44.....	איור 47 שאלת סינון לפי טיפוס המשמש .....
44.....	איור 48 שאלת זו מוצאת לפי טווח המחרירים של תיקון WIFI .....
47.....	איור 49 שאלת מין מסמכים לפי הטווח שלהם מנוקודה .....
48.....	איור 50 מסנן שימושים מרובים לפי שאלת GEOdistance .....
49.....	איור 51 אתר ה <a href="http://boundingbox.com">boundingbox.com</a> .....
50.....	איור 52 אשפרות לקבל קואורדינטות מה מצלע .....
51.....	איור 53 חיפוש נתונים לפי תיכון גאוגרפי של ת"א .....
53.....	איור 54 דף ה <a href="#">DISCOVERY</a> .....
53.....	איור 55 דף ה <a href="#">DISCOVERY</a> עם המרכיבים שלו .....
54.....	איור 56 דף ה <a href="#">VISUALIZE</a> והפתרונות השונות .....
54.....	איור 57 ויזואליזציה ה MAP עם נתונים .....
55.....	איור 58 ויזואליזית כמות התקנות WIFI על ידי BAR CHART .....
55.....	איור 59 ויזואליזית ה PIE CHART של נתונים בת מילון .....
56.....	איור 60 דשبورד אשר מלא נתונים מעירית ת"א .....
56.....	איור 61 דשبورד שני אשר מלא נתונים מעירית ת"א .....
57.....	איור 62 דשبورד שלישי אשר מלא נתונים מעירית ת"א .....

[www.sce.ac.il](http://www.sce.ac.il)

**קמפוס באר שבע**  
ביאליק 95, באר שבע 84100

**קמפוס אשדוד**  
ד'בוטינסקי 84, אשדוד 77245

