

مقدمه‌ای بر الستیک سرچ

درس ارائه مطالب علمی و فنی

پارسا محمدیان

دانشگاه صنعتی شریف

۲۱ خرداد ۱۴۰۰



۱ Elasticsearch چیست؟

- ◀ آشنایی با ویژگی‌های معماری
- ◀ کاربردهای الستیک سرچ در صنعت

۲ مفاهیم اولیه‌ی الستیک سرچ

- ◀ آشنایی با اجزا
- ◀ مقایسه با دیتابیس‌های رابطه‌ای

۳ کارایی در عملیات مختلف

- ◀ جزئیات تست انجام شده
- ◀ عملیات ساختن مخزن
- ◀ عملیات بارگذاری داده
- ◀ عملیات جستجو زیررشته
- ◀ عملیات جستجو بازه
- ◀ عملیات تجمعی (Aggregation)

۴ الستیک سرچ کمی عمیق‌تر

۵ منابع



❶ Elasticsearch چیست؟

- ◀ آشنایی با ویژگی‌های معماری
- ◀ کاربردهای الستیک سرچ در صنعت

❷ مفاهیم اولیه‌ی الستیک سرچ

- ◀ آشنایی با اجزا
- ◀ مقایسه با دیتابیس‌های رابطه‌ای

❸ کارایی در عملیات مختلف

- ◀ جزئیات تست انجام شده
- ◀ عملیات ساختن مخزن
- ◀ عملیات بارگذاری داده
- ◀ عملیات جستجو زیررشته
- ◀ عملیات جستجو بازه
- ◀ عملیات تجمعی (Aggregation)

❹ الستیک سرچ کمی عمیق‌تر

❺ منابع



- ◀ موتور جستجو متن‌باز بر پایه‌ی لوسین (Lucene)
 - ◀ لوسین موتور جستجو متن‌باز نوشته شده به زبان جاوا است
- ◀ توزیع شده (Distributed)
- ◀ مقیاس‌پذیر (Scalability)
- ◀ امکان جستجو Full-text
- ◀ رابط HTTP و RESTful
- ◀ اسناد غیروابسته به قالب (Scheme-free) و بر پایه‌ی JSON



Elasticsearch چیست؟

کاربردهای الاستیک سرچ در صنعت



شکل: نمونه‌های کاربردهای مختلف و مشتری‌ها^۱

^۱ تصویر از [۱]

- ◀ خوشه یا Cluster ⇐ تعدادی از سرورهای الاستیک سرچ که به هم متصل هستند
- ◀ گره یا Node ⇐ هر یک از سرورهای الاستیک سرچ در Cluster
- ◀ سند یا Document ⇐ هر یک از اسناد متنی که بارگذاری می‌شود
- ◀ مخزن یا Index ⇐ مخازنی که دارای تعدادی سند با قالب یکسان هستند
- ◀ نگاشت یا Mapping ⇐ قالب هر Index
- ◀ تکه داده یا Shard ⇐ هر Index تعدادی Shard دارد
- ◀ بخش یا Segment ⇐ هر Shard تعداد Segment دارد



الستیک سرچ	دیتابیس‌های رابطه‌ای
Index	Database
Type ^a	Table
Document	Row
Field	Column
Mapping	Schema

^aRemoved since version 6.00

کمی نکات منفی در مورد الستیک!

◀ مفهوم Transaction ندارد

◀ پشتیبانی محدودتر از Join

نکته مهم

این محدودیت‌ها برای دیتابیس‌های NoSQL و Distributed قابل انتظار است.



- ▶ داده بارگذاری شده اطلاعات املاک تهران موجود در دیوار است. [۴]
- ▶ این داده شامل ۳۷۲۳۳۲ رکورد و حجم خام 143 MB است.
- ▶ تست‌ها بر روی یک کامپیوتر شخصی با رم 8 GB و پردازنده Core i7-8550 انجام شده است.

ابزارهای مورد استفاده

برای کوئری زدن به الاستیک از ابزار Kibana که یک وب اپلیکیشن است استفاده شده است. همچنین برای بارگذاری داده در الاستیک با استفاده از ابزار Curl درخواست HTTP فرستاده شده است. برای ارتباط با SQLServer از Azure Data Studio استفاده شده است.



کارایی در عملیات مختلف

عملیات ساختن مخزن

The screenshot displays two application windows side-by-side. The left window is SQL Server Enterprise Manager, showing a query execution plan for the command `CREATE DATABASE Comparison;`. The 'Messages' pane at the bottom indicates the query was executed successfully at 12:47:51 AM, with a total execution time of 00:00:01.648. The right window is Kibana DevTools, showing a REST client request to `PUT /comparison` with a JSON body. The response shows the index was created successfully. The console output at the bottom right of Kibana shows a response time of 1207 ms, which is highlighted with a red box.

SQLQuery_1 - localhostmaster (Integrated) - Azure...

SQLQuery_1 - localhostmaster

Run Cancel Disconnect Change Connection master

```
1 CREATE DATABASE Comparison;
```

Messages

12:47:51 AM Started executing query at line 1
Commands completed successfully.
Total execution time: 00:00:01.648

Windows PowerShell

PS C:\Users\Parisa>

Elastic

127.0.0.1:5601/app/kibana#/dev_tools/console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Help

```
1 PUT /comparison
```

```
2 {
3   "shards_acknowledged": true,
4   "index": "comparison"
5 }
6
```

200 - OK 1207 ms



پارسا محمدیان (دانشگاه صنعتی شریف)



کارایی در عملیات مختلف

عملیات جستجو زیررشته

The image shows two side-by-side windows. The left window is SQL Studio, displaying a table schema for 'SQLQuery_1.sql' and a query: `SELECT * FROM RealEstate WHERE title LIKE N'%مشارکان%'`. The right window is the Elasticsearch console, showing a REST client request and its response. The request is a PUT to `/comparison` with a JSON body containing search parameters. The response is a 200 OK status with a 103ms execution time, returning a detailed document structure for a comparison.

SQL Studio Query:

```
SELECT * FROM RealEstate WHERE title LIKE N'%مشارکان%'
```

Elasticsearch Request:

```
PUT /comparison
{
  "query": {
    "match": {
      "title": "مشارکان"
    }
  }
}
```

Elasticsearch Response:

```
{
  "took": 27,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1767,
      "relation": "eq"
    },
    "max_score": 8.026751,
    "hits": [
      {
        "_index": "comparison",
        "_type": "doc",
        "_id": "rQJk93k8KJf4GAtk98Lq",
        "_score": 8.026751,
        "district": "مشارکان",
        "token": "AVL4887T",
        "sub_category": "shop-rent",
        "category": "commercial-rent",
        "business_type": "personal",
        "price": null,
        "credit": 300000000,
        "rent": 15000000,
        "title": "مشارکان",
        "location": { },
        "area": 78,
        "year": 1373,
        "room": 0
      }
    ]
  }
}
```

کارایی در عملیات مختلف

عملیات جستجو بازه

The image shows two side-by-side windows. The left window is SQL Server Enterprise Manager, displaying a table named 'RealEstate' with columns: id, title, location, area, year, and room. A query is executed: `SELECT * FROM RealEstate WHERE title LIKE N'%مشارکتی%';`. The right window is the Elasticsearch console, showing a PUT request to `/comparison` and a GET request to `/comparison/_search` with a range query for 'room' greater than or equal to 1. The search results are displayed in JSON format, showing a single hit for a property with room count 2.

```
1 PUT /comparison
2
3 GET /comparison/_count
4
5 GET /comparison/_search
6 {
7   "query": {
8     "match": {
9       "title": "مشارکتی"
10    }
11  }
12 }
13
14 GET /comparison/_search
15 {
16   "query": {
17     "range": {
18       "room": {
19         "gte": 1
20       }
21     }
22   }
23 }
```

```
1 {
2   "took": 26,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 10000,
13      "relation": "gte"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        "_index": "comparison",
19        "_type": "doc",
20        "_id": "f1193k8KJ46GAtkzt1",
21        "_score": 1.0,
22        "_source": {
23          "district": "پرنده",
24          "token": "AV1grZzh",
25          "sub_category": "apartment",
26          "sell": "residential",
27          "business_type": "personal",
28          "price": 240000000,
29          "credit": null,
30          "rent": null,
31          "title": "مساکن واریز",
32          "location": {},
33          "area": 84,
34          "year": 1399,
35          "room": 2,
36          "floor": 3,

```



کارایی در عملیات مختلف

عملیات تجمعی (Aggregation)

The image shows two side-by-side windows. The left window is SQL Studio, and the right window is the Elastic console.

SQL Studio (Left):

- File: SQLQuery_1.sql - localhost:Comparison (Integrated) - Azure Data Studio
- Run: Run | Cancel | Disconnect | Change Connection | Comparison
- SQL Query:

```
18 -- [rent] BIGINT,  
19 -- [title] NVARCHAR(MAX),  
20 -- [location] NVARCHAR(MAX),  
21 -- [area] INT,  
22 -- [year] INT,  
23 -- [room] INT  
24 -- ];  
25  
26 -- SELECT * FROM RealEstate WHERE title LIKE N'%مشارکان%';  
27  
28 -- SELECT * FROM RealEstate WHERE room >= 1;  
29  
30 SELECT AVG(room) AS 'roomAverage'  
31 FROM RealEstate;
```

- Results: Messages
- 12:22:15 AM: Started executing query at line 1
Warning: Null value is eliminated by an aggregate or other SET operation.
(1 row affected)
Total execution time: 00:00:00.019

Elastic Console (Right):

- URL: localhost:5601/app/kibana#/dev_tools/console
- Dev Tools: Console | Search Profiler | Grok Debugger | Painless Lab | BETA
- History: Settings | Help
- 200 - OK 216 ms
- PUT /comparison
- GET comparison/_count
- GET comparison/_search
- POST comparison/_search
- Aggregations:
- room_avg: {
 value: 2.0316522661770797
}



کارایی در عملیات مختلف

عملیات تجمعی (Aggregation)

The image displays two side-by-side windows. The left window is the SQLQuery_1.sql editor in Azure Data Studio, showing a series of SQL queries for testing aggregation functions. The right window is the Elastic console, showing the results of these queries as JSON documents.

SQLQuery_1.sql (Left Window):

```
18 -- [rent] BIGINT,  
19 -- [title] NVARCHAR(MAX),  
20 -- [location] NVARCHAR(MAX),  
21 -- [area] INT,  
22 -- [year] INT,  
23 -- [room] INT  
24 -- ];  
25  
26 -- SELECT * FROM RealEstate WHERE title LIKE N'%مشارکان%';  
27  
28 -- SELECT * FROM RealEstate WHERE room >= 1;  
29  
30 SELECT AVG(room) AS 'roomAverage'  
31 FROM RealEstate;
```

Elastic Console (Right Window):

The console shows the results of the queries. The first query (line 26) returns a document with the following fields:

```
{  
  "_type": "doc",  
  "_id": "Bv1i93k8KJ4GATkzt5",  
  "_score": 1.0,  
  "_source": {  
    "district": "دزوم",  
    "token": "wXoZSE0",  
    "sub_category": "apartment",  
    "sell": true,  
    "category": "residential",  
    "business_type": "real-estate-business",  
    "price": null,  
    "credit": null,  
    "rent": null,  
    "title": "280 متر محوط در منطقه مرکزی",  
    "location": {  
      "area": 280,  
      "year": 1399,  
      "room": 4,  
      "agency": "پارسیان دور - انبار",  
      "floor": 2,  
      "elevator": true,  
      "parking": true,  
      "storage": true  
    }  
  }  
}
```

The second query (line 30) returns a document with the following fields:

```
{  
  "room_avg": 2.0316522661770797  
}
```



سند ۱ :

سند ۲ :



- [1] Howard Chen. *Elastic Q1 FY2021 Analysis*. URL: <https://blog.publiccomps.com/elasticsearchq1fy2021/> (visited on 06/06/2021).
- [2] Zachary Tong Clinton Gormley. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. O'Reilly Media, 2015. ISBN: 1449358543.
- [3] *Elastic Stack and Product Documentation*. URL: <https://www.elastic.co/guide/index.html> (visited on 04/19/2021).
- [4] Amirali Taheri. *tehran-real-estate-prices*. URL: <https://www.kaggle.com/amiralitaheri/tehranrealestateprices> (visited on 06/11/2021).

اگر جایی منبع ذکر نشده از [۳] و [۲] استفاده شده است.

