

مقدمه‌ای بر الستیک سرچ

درس ارائه مطالب علمی و فنی

پارسا محمدیان

دانشگاه صنعتی شریف

۲۲ تیر 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 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 الستیک سرچ در صنعت



شکل: نمونه‌های کاربردهای مختلف و مشتری‌ها^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 editor with the command `CREATE DATABASE Comparison;` and a Messages pane indicating successful execution at 12:47:51 AM. The right window is Kibana DevTools, showing a REST client with a PUT request to `/_bulk` that successfully indexed a document, returning a 200 status and a response body containing `"index": "comparison"`. A red box highlights the execution time in the SQL Server Messages pane, and another red box highlights the response time in the Kibana DevTools console.



کارایی در عملیات مختلف

عملیات بارگذاری داده

The image displays a workflow for data archiving and comparison. On the left, a SQL query in SQLQuery_1.sql creates a database named 'Comparison' and a table 'RealEstate' with various attributes. The query is executed, and the results are shown in the 'Messages' pane, indicating that 372392 rows were affected and the execution time was 00:04:21.288.

In the center, a terminal window shows the execution of a script that archives data from the 'RealEstate' table into a JSON file named '23-2-12.json'. The script uses the 'SQLCMD' command-line utility.

On the right, a REST client (Elastic) shows the response to a POST request to the endpoint 'localhost:5601/app/kibana#/dev_tools/console'. The response is a JSON object containing a 'count' of 372281, a 'total' of 1, and a 'shards' array with a 'successful' status of 1 and a 'failed' status of 0. The response is highlighted with a red arrow and the text 'Less than one minute!!!'.

```
--CREATE DATABASE Comparison;
--DECLARE @JSON NVARCHAR(MAX);
--SELECT @JSON=bulkcolumn FROM
--OPENROWSET (BULK 'F:\Temporary\archive\23-2-12.json', SINGLE_NCLOB)
--AS ImportData;
--SELECT * INTO RealEstate
--FROM OPENROWSET (@JSON)
--WITH
--(
--  [district] NVARCHAR(MAX),
--  [token] NVARCHAR(MAX),
--  [sub_category] NVARCHAR(MAX),
--  [category] NVARCHAR(MAX),
--  [business_type] NVARCHAR(MAX),
--  [price] BIGINT,
--  [credit] BIGINT,
--  [rent] BIGINT,
--  [title] NVARCHAR(MAX),
--  [location] NVARCHAR(MAX),
--  [area] INT,
--  [year] INT,
--  [room] INT
--);
```

```
SQLCMD -E -i "F:\Temporary\archive\23-2-12.json" -s SINGLE_NCLOB
```

```
curl -XPOST http://localhost:5601/app/kibana#/dev_tools/console
```

```
{
  "count": 372281,
  "total": 1,
  "shards": {
    "successful": 1,
    "skipped": 0,
    "failed": 0
  }
}
```

کارایی در عملیات مختلف

عملیات جستجو زیررشته

The image displays two side-by-side windows. The left window is SQL Server Enterprise Edition, showing a query execution plan and results. The right window is the Elasticsearch Dev Tools console, showing a PUT request to the /comparison endpoint and the resulting JSON response.

SQL Server Query and Results:

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

Results: Started executing query at line 1 (1857 rows affected) Total execution time: 00:00:01.258

Elasticsearch Dev Tools Console:

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

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": 6
        }
      }
    ]
  }
}
```



کارایی در عملیات مختلف

عملیات جستجو بازه

The image displays two side-by-side windows. The left window is SQL Server Enterprise Edition, showing a query execution plan and results for a range search. The right window is the Elasticsearch console, showing the same query and its results in JSON format.

SQL Server Query and Results:

```
-- (create test schema);
-- [rent] BIGINT,
-- [title] NVARCHAR(MAX),
-- [location] NVARCHAR(MAX),
-- [area] INT,
-- [year] INT,
-- [room] INT
-- ];

-- SELECT * FROM RealEstate WHERE title LIKE N'%مساغول%';
-- SELECT * FROM RealEstate WHERE room >= 1;
```

Results: Started executing query at line 1 (316700 rows affected). Total execution time: 00:00:05.753.

Elasticsearch Query and Results:

```
PUT /comparison
GET comparison/_count
GET comparison/_search
{
  "query": {
    "match": {
      "title": "مساغول"
    }
  }
}
GET comparison/_search
{
  "query": {
    "range": {
      "room": {
        "gte": 1
      }
    }
  }
}
```

Results (JSON):

```
{
  "took": 26,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 1.0,
    "hits": [
      {
        "_index": "comparison",
        "_type": "doc",
        "_id": "f1193k8KJ46GAtkzt1",
        "_score": 1.0,
        "_source": {
          "district": "برند",
          "token": "AV1grZzh",
          "sub_category": "apartment",
          "sell": "sell",
          "category": "residential",
          "business_type": "personal",
          "price": 240000000,
          "credit": null,
          "rent": null,
          "title": "مساغول واریز",
          "location": {},
          "area": 84,
          "year": 1399,
          "room": 2,
          "floor": 3,

```

کارایی در عملیات مختلف

عملیات تجمعی (Aggregation)

The image shows two side-by-side windows. The left window is Azure Data Studio, displaying a SQL query in a file named 'SQLQuery_1.sql'. The query is as follows:

```
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;
```

The right window is the Elastic console, showing the execution of the query. The console output is as follows:

```
1 PUT /comparison  
2  
3 GET comparison/_count  
4  
5 GET comparison/_search  
6 {  
7   "query": {  
8     "match": {  
9       "title": "مشارکان"  
10    }  
11  }  
12 }  
13  
14 GET comparison/_search  
15 {  
16   "query": {  
17     "range": {  
18       "room": {  
19         "gte": 1  
20       }  
21     }  
22   }  
23 }  
24  
25 POST comparison/_search  
26 {  
27   "aggs": {  
28     "room_avg": {  
29       "avg": {  
30         "field": "room"  
31       }  
32     }  
33   }  
34 }
```

The console output shows the results of the query, including the aggregation of the 'room' field. The results are as follows:

```
229 "type": "doc",  
230 "_id": "Bv1i93k8KJ4GATktz5",  
231 "_score": 1.0,  
232 "_source": {  
233   "district": "دزوم",  
234   "token": "wXoZSE0",  
235   "sub_category": "apartment",  
236   "category": "residential",  
237   "business_type": "real",  
238   "estate-business",  
239   "price": null,  
240   "credit": null,  
241   "rent": null,  
242   "title": "مدر مختوب در 280 مترمربع",  
243   "location": {  
244     "area": 280,  
245     "year": 1399,  
246     "rooms": 4,  
247     "agency": "پارمیان دور",  
248     "elevator": true,  
249     "parking": true,  
250     "storage": true  
251   },  
252   "aggregations": {  
253     "room_avg": {  
254       "value": 2.0316522661770797  
255     }  
256   }  
257 }
```



- [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.

