We integrate a variety of features into the Xome® application to assist home buyers in finding suitable properties. Users can input their desired location, price range, and other search criteria to find available properties that match their preferences. If they are not satisfied with the listings returned to them, they can receive notifications for when new properties hit the market.

Tools like Elasticsearch Percolator are especially beneficial in various scenarios as a solution for sending notifications. As a reverse search, it is given a document and finds all the search queries that match the document.

By reversing the traditional search process, developers can address specific use cases and enhance the performance of their search applications. Understanding the Percolator’s capabilities and incorporating it into an [Elasticsearch workflow](https://www.xome.com/blog/improving-property-search-performance-with-elasticsearch/) opens up new possibilities for developers who are optimizing search functionality.

## **Using the percolator field type and the percolate query**

Elasticsearch Percolator has two components: the percolate field type and the percolate query.

The [percolator field type](https://www.elastic.co/guide/en/elasticsearch/reference/current/percolator.html) stores queries that are registered with the Percolator. When indexing a document containing a query into a percolate field, Elasticsearch parses the JSON structure of the query and stores it in a format that can be efficiently matched against incoming documents.

The [percolate query](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-percolate-query.html) then uses the query stored in the percolator field type to match provided documents. Elasticsearch retrieves the document upon execution of the percolate query to be matched and compares it against the registered queries stored in the percolate field. If there is a match between the document and any of the registered queries, Elasticsearch returns the IDs of the matching queries.

## **Use case: Sending home alerts and notifications to users**

When users are not satisfied with their property search results, they can save their search criteria and set up alerts for whenever a new listing becomes available. This can be done efficiently using Elasticsearch Percolator.

For notifications to occur, a daily job runs and takes all the saved criteria, and searches for new listings in the last one day to send an alert. Sending instant notifications becomes a bit more complex. Normally the system would need to iterate over the criteria of all users for each new listing to find who to alert, but with Percolator, it’s simplified.

Create an index containing a percolator field

For a general code example, we might have a schema that looks something like this for the property listings:

{

  "listings": {

    "mappings": {

      "properties": {

        “id”: {“type”: “integer”},

        "address": {"type": "text"},

        "city": {"type": "text"},

        "state": {"type": "keyword"},

        "zip": {"type": "keyword"},

        "price": {"type": "integer"},

        "bedrooms": {"type": "integer"},

        "bathrooms": {"type": "float"}

        ....

      }

    }

  }

}

For percolation, an index needs to be created that contains a percolator field. The Percolator index may contain fields like shown below:

{

“query”: {“type”: “percolator”},

"userid": {"type": "integer"},

"emailaddress": {"type": "keyword"},

....

}

These fields may vary based on requirements, except for the “query” field which would always need to be included.

Insert into the Percolator index.

Suppose a user is searching for a property with 2 bedrooms in Los Angeles, CA, within a price range of $100,000 to $200,000. The query used to search for that property would go in the “query” field of the Percolator.

The following document would then be inserted into Elasticsearch:

{

"userid": 12123,

"emailaddress": "username@emailaddress.com".

"query": {

"bool": {

"must": [

{

"range": {

"price": {

"gte": 100000,

"lte": 200000

}

}

},

{

"term": {

"bedrooms": 2

}

},

{

"term": {

"state": {

"value": "CA"

}

}

},

{

"match\_phrase": {

"city": {

"query": "Los Angeles"

}

}

}

]

}

}

}

Once a new listing is returned, saved searches are queried with the following:

{

"query": {

"bool": {

"must": [

{

"percolate": {

"field": "query",

"id": "232206669",

"index": "listings"

}

}

]

}

}

}

The above query would return all the saved searches that match the listing with id 232206669. From there, notifications are sent to the users returned by the above query for this listing.

Through using Elasticsearch Percolator, real estate applications can ensure timely alerts and notifications tailored to each user’s unique requirements. This streamlines the process of sending alerts and demonstrates a commitment to enhancing user engagement and satisfaction.