Topics:

1. **Introduction to Apache SOLR**
2. **Apache SOLR in Action**
3. **Search Algorithm**
4. **SOLR Indexing**
5. **Searching Using SOLR**
6. **Advanced Features in SOLR**
7. **Administration & SOLR Cloud**
8. **Project Support**
9. **Industry Knowledge**
10. **Solr Questions**

ASNWERS:

1. **Introduction to Apache SOLR**
   a. **What is Apache Lucene**

      Lucene is a open source Java based search engine library providing powerful indexing and search features.
      Lucene is not a complete application, but rather a code library and API that can easily be used to add search capabilities to applications.

2. **What is Apache Solr**

      Apache Solr is a scalable, ready-to-deploy enterprise search engine that was developed to search a large volume of text-centric data and returns results sorted by relevance. It is built on top of Apache Lucene.

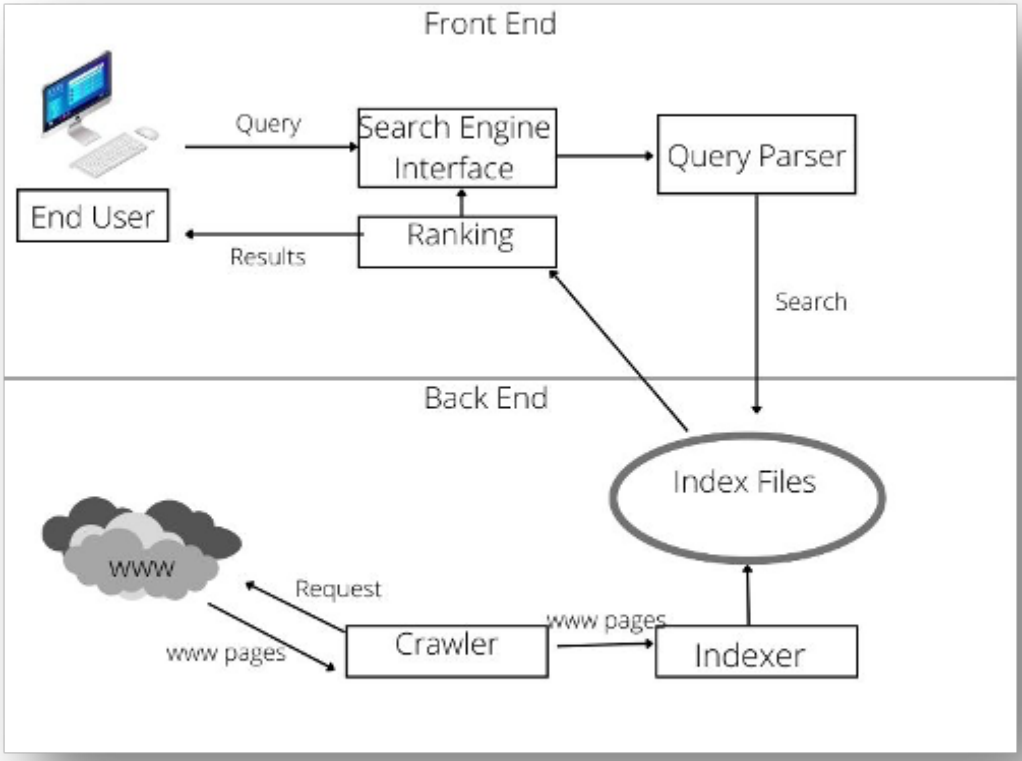
3. Search Engine and How it works?

      A Search Engine refers to a huge database of Internet resources such as webpages, images, etc. It helps to locate information on the World Wide Web.
      Users can search for information by passing queries into the Search Engine in the form of keywords or phrases. The Search Engine then searches in its database and returns relevant data/links to the user.

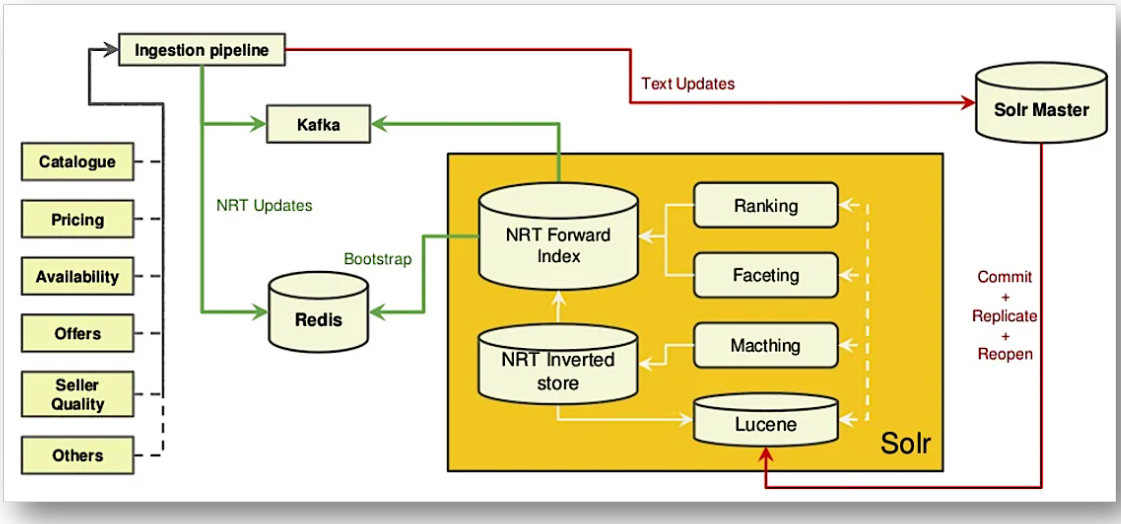
      Components of Search Engine:

- Input data/Web Crawler
- Database/Storage

- Search Interface



Architectural Implementation of Search Engine at Flipkart:



4. RDBMS vs NoSQL vs SOLR

	RDBMS	NoSQL	SOLR
Table Structure	• Tables consist of rows and columns. • Each column has a predefined data type. • Data is organized in a structured, tabular format.	• Table structure varies based on the NoSQL database type (document-based, key-value, column-family, graph-based). • NoSQL databases may not use tables and rows in the traditional sense.	• Solr does not use traditional tables. • It follows a document-centric approach where each document represents a unit of data. • Documents consist of fields
Schema	• RDBMS enforces a strict schema. • The schema defines table structures, column names, data types, and relationships between tables using foreign keys.	• NoSQL databases typically offer schema flexibility. • You can add fields to documents or change data structures without a rigid schema. • Different records/documents in the same collection may have varying structures.	• Solr provides some schema flexibility, allowing dynamic fields to be defined. • Fields can be added or changed without extensive schema modifications. • However, Solr still has a schema configuration to define field types and properties.
Data Relationships	• RDBMS supports complex relationships between tables using primary keys and foreign keys. • JOIN operations are used to combine data from multiple tables.	• Relationships are typically handled differently based on the NoSQL database type. • Some NoSQL databases support basic relationships, but complex joins are often avoided in favor of denormalization.	• Solr specializes in full-text search and indexing, making it ideal for text-centric data. • Documents are indexed and ranked based on relevance to search queries. • It is not a general-purpose database but excels in information retrieval.

5. Why would we prefer SOLR?

   We would prefer Solr for its exceptional full-text search capabilities, high performance, scalability, and flexibility, making it an ideal choice for applications requiring powerful and efficient search functionality.

6. SOLR Features

Feature	Description
Full-Text Search	Solr provides robust full-text search capabilities, making it easy to search and retrieve documents based on their content.
Distributed Search	Solr supports distributed search, allowing you to scale your search infrastructure horizontally for high availability and performance.
Near Real-Time Indexing	Solr offers near real-time indexing, enabling rapid updates and retrieval of freshly added or modified documents.
Faceted Search	Faceted search allows users to filter and navigate search results based on predefined facets or categories, enhancing user experience.
Scalability	Solr is highly scalable, with support for sharding and replication, making it suitable for handling large volumes of data and traffic.
Extensible Architecture	Solr's modular and extensible architecture enables easy customization and integration with other tools and systems.
JSON and XML Support	Solr accepts and returns data in various formats, including JSON and XML, making it versatile and compatible with different applications.
Text Analysis and Tokenization	Solr provides robust text analysis and tokenization capabilities, allowing for language-specific text processing and indexing.
Rich Query Language	Solr supports a powerful query language that enables complex and flexible search queries, including Boolean operators, proximity searches, and more.
Highlighting and Snippets	Solr can highlight search terms in the search results and provide snippets of text, helping users quickly identify relevant content.
Geospatial Search	Solr includes geospatial search capabilities, allowing you to perform location-based searches and spatial queries.
Data Import and Export	Solr provides tools for importing data from various sources (e.g., databases) and exporting search results, facilitating data integration and reporting.
Security	Solr offers security features like authentication, authorization, and encryption to protect sensitive data and control access to the search engine.
Monitoring and Logging	Solr includes monitoring and logging tools to help administrators track system performance, troubleshoot issues, and optimize resource usage.
Community Support	Solr has an active and vibrant open-source community, providing extensive documentation, forums, and resources for users and developers.

7. Installing & Setting up Solr

8. Solr Admin UI Quick Tour
9. Solr Architecture

2. Apache SOLR in Action

- Start Searching
- Explore Admin UI
- Analyzing Response
- Solr Schema & Configuration
- Update Handlers
- Field Types
- Analyzer
- Tokenizers
- Filters

3. Search Algorithm

- Inverted Index
- Forward Index

4. SOLR Indexing

- Concepts of Indexing in Solr
- Indexing JSON Documents
- Indexing PDF Documents
- Atomic Updates
- Hard vs Soft Commit
- Reindexing

5. Searching Using SOLR

- Sorting & Relevance in Solr Boosting
- Lucene scoring model
- Query Syntax in Solr Basic Parsers

6. Advanced Features in SOLR

- Boosting
- Faceting
- Highlighting
- Spell Checking
- Ranking
- Suggestions

- Pagination
- Grouping
- Spatial Search
- Real-Time Search
- Solr Plugin

7. Administration & SOLR Cloud

- Administering Solr Core
- Running on Tomcat/Jetty
- Managing Multiple Cores
- Log Management
- Solr on Cloud
- Solr & AWS

8. Project Support

- Summary of everything has been covered so far
- Basic Solr Project For Providing Search Capability(Like Google Search Engine)
- SOLR and ML Approach

9. Industry Knowledge

- How Solr is used in Top Tire Companies
- Architecture of Solr with regards to Industry
- Upgrading of Solr Infrastructure

10. Solr Questions

- What is Lucene?
- Why Indexing?
- Inverted Indexing vs Forward Indexing difference
- How does tokenizer & Filter fit into inverted indexing concept explain with an example ?
- What algorithm does apache solr uses by default to search documents ?
- Explain BM25
- What are the different algorithmns solr use to search data