## 1. Compare NoSQL & RDBMS

| Criteria | NoSQL | RDBMS |
|---|---|---|
| Data format | Does not follow any order | Organized and structured |
| Scalability | Very Good | Average |
| Querying | Limited as no Join Clause | Using SQL |
| Storage mechanism | Key-Value Pair, document, column storage, etc. | Data & relationship stored in different tables |

## 2. What is NoSQL?

NoSQL encompasses a wide variety of different database technologies that were developed in response to a rise in the volume of data stored about users, objects and products. The frequency in which this data is accessed, and performance and processing needs.

## 3. What are the features of NoSQL?

When compared to relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

- Large volumes of structured, semi-structured, and unstructured data
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture

## 4. Explain “Polyglot Persistence” in NoSQL?

The term polyglot programming, to express the idea that applications should be written in a mix of languages to take advantage of the fact that different languages are suitable for tackling different problems.

Complex applications combine different types of problems, so picking the right language for each job may be more productive than trying to fit all aspects into a single language.

## 5. Does NoSQL Database Interact With Oracle Database?

NoSQL Database supports retrieving records through the Oracle Database External Table functions. This makes it possible to perform some queries from Oracle Database and retrieve records from NoSQL Database.

## 6. When should I use a NoSQL database instead of a relational database?

A relational database enforces ACID. So, you will have schema based transaction oriented data stores. It’s proven and suitable for 99% of the real world applications. You can practically do anything with relational databases.But, there are limitations on speed and scaling when it comes to massive high availability data stores.

A columnar database is, simply a database that stores data in a columns rather than rows, and from the outside looks like your typical row-based DBMS.

There are a variety of columnar databases: Sybase IQ, InfoBright, and Redshift, to name a few (there are many others).

One of the main benefits of a columnar database is that data can be highly compressed. The compression permits columnar operations — like MIN, MAX, SUM, COUNT and AVG— to be performed very rapidly.

**Columnar Benefits**

- Columnar databases typically provide better query performance when the number of fields required for the query are a subset of the transaction's fields
- Columnar databases can provide more efficient use of space (data compression)

**Columnar Drawbacks**

- Writing transactional data to a columnar database takes more time than to a row-based database (warning, don't use a columnar store for your transactional systems)
- If your query contains all the columns in a transaction, your query will typically perform worse compared to a row-based database