## **Hibernate – Java Interview Questions for Experienced Professionals**

Spring boot:

Spring Boot Application Properties

Spring Boot Framework comes with a built-in mechanism for application configuration using a file called application.properties.

JPA(Java Persistence API ):

The primary focus of JPA is the ORM layer

*Spring Boot JPA is a Java specification for managing relational data in Java application*s. It allows us to access and persist(অবিরত) data between Java object class and relational database. JPA follows Object-Relation Mapping (ORM). It is a set of interfaces

Object-Relational Mapping (ORM) is the process of converting Java objects to database tables

The Java Persistence API (JPA) is a specification that defines how to persist data in Java applications.

It also provides a runtime **EntityManager** API for processing queries and transactions on the objects against the database. It uses a platform-independent object-oriented query language JPQL (Java Persistent Query Language).

In the context of persistence, it covers three areas:

* The Java Persistence API
* **Object-Relational** metadata
* The API itself, defined in the **persistence** package

JPA is not a framework. It defines a concept that can be implemented by any framework.

## **Why should we use JPA?**

JPA is simpler, cleaner, and less labor-intensive than JDBC, SQL, and hand-written mapping. JPA is suitable for non-performance oriented complex applications. The main advantage of JPA over JDBC is that, in JPA, data is represented by objects and classes while in JDBC data is represented by tables and records. It uses POJO to represent persistent data that simplifies database programming. There are some other advantages of JPA:

* JPA avoids writing DDL in a database-specific dialect of SQL. Instead of this, it allows mapping in XML or using Java annotations.
* JPA allows us to avoid writing DML in the database-specific dialect of SQL.
* JPA allows us to save and load Java objects and graphs without any DML language at all.
* When we need to perform queries JPQL, it allows us to express the queries in terms of Java entities rather than the (native) SQL table and columns.

Hibernate is one of the most popular Java ORM frameworks in use today. Its first release was almost twenty years ago, and still has excellent community support and regular releases. Additionally, **Hibernate is a standard implementation of the JPA specification**, with a few additional features that are specific to Hibernate. Let's take a look at some core features of JPA and Hibernate.

## **Object-Relation Mapping (ORM)**

In ORM, the mapping of Java objects to database tables, and vice-versa is called **Object-Relational Mapping.** The ORM mapping works as a bridge between a **relational database** (tables and records) and **Java application** (classes and objects).

### **1. What is Hibernate Framework?**

Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is Java-based ORM tool that provides a framework for mapping application domain objects to the relational database tables and vice versa.

Hibernate provides a reference implementation of Java Persistence API, that makes it a great choice as ORM tool with benefits of loose coupling. We can use the Hibernate persistence API for CRUD operations. Hibernate framework provide option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.

Similarly, hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

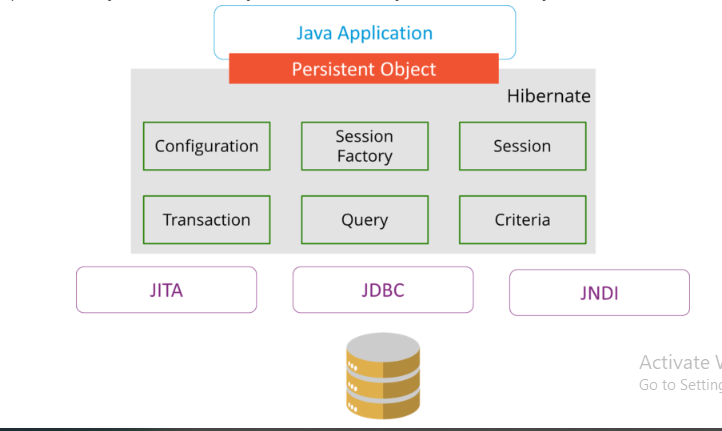
### **2. What are the important benefits of using Hibernate Framework?**

Some of the important benefits of using hibernate framework are:

1. Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.
2. Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.
3. Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism, and association.
4. Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because learning curve is small and there are tons of online documentation and help is easily available in forums.
5. Hibernate is easy to integrate with other Java EE frameworks, it’s so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.
6. Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it’s required.
7. Hibernate cache helps us in getting better performance.
8. For database vendor specific feature, hibernate is suitable because we can also execute native sql queries.

Overall hibernate is the best choice in current market for ORM tool, it contains all the features that you will ever need in an ORM tool.

### **3. Explain Hibernate architecture.**

Hibernate has a layered architecture which helps the user to operate without having to know the underlying APIs. Hibernate makes use of the database and configuration data to provide persistence services (and persistent objects) to the application. It includes many objects such as persistent object, session factory, transaction factory, connection factory, session, transaction etc. 

The Hibernate architecture is categorized in four layers.

* Java application layer
* Hibernate framework layer
* Backhand API layer
* Database layer

### **4. What are the differences between get and load methods?**

The differences between get() and load() methods are given below.

|  |  |  |
| --- | --- | --- |
| **No.** | **get()** | **load()** |
| 1) | Returns null if object is not found. | Throws ObjectNotFoundException if an object is not found. |
| 2) | get() method always hit the database. | load() method doesn’t hit the database. |
| 3) | It returns a real object, not a proxy. | It returns a proxy object. |
| 4) | It should be used if you are not sure about the existence of instance. | It should be used if you are sure that the instance exists. |

### **5. What are the advantages of Hibernate over JDBC?**

Some of the important advantages of Hibernate framework over JDBC are:

1. Hibernate removes a lot of boiler-plate code that comes with JDBC API, the code looks cleaner and readable.
2. Hibernate supports inheritance, associations, and collections. These features are not present with JDBC API.
3. Hibernate implicitly provides transaction management, in fact, most of the queries can’t be executed outside transaction. In JDBC API, we need to write code for transaction management using commit and rollback.
4. JDBC API throws SQLException that is a checked exception, so we need to write a lot of try-catch block code. Most of the times it’s redundant in every JDBC call and used for transaction management. Hibernate wraps JDBC exceptions and throw JDBCException or HibernateException un-checked exception, so we don’t need to write code to handle it. Hibernate built-in transaction management removes the usage of try-catch blocks.
5. Hibernate Query Language (HQL) is more object-oriented and close to Java programming language. For JDBC, we need to write native SQL queries.
6. Hibernate supports caching that is better for performance, JDBC queries are not cached hence performance is low.
7. Hibernate provides option through which we can create database tables too, for JDBC tables must exist in the database.
8. Hibernate configuration helps us in using JDBC like connection as well as JNDI DataSource for the connection pool. This is a very important feature in enterprise application and completely missing in JDBC API.
9. Hibernate supports JPA annotations, so the code is independent of the implementation and easily replaceable with other ORM tools. JDBC code is very tightly coupled with the application.

In case you are facing any challenges with these Java interview questions, please comment on your problems in the section below. Apart from this Java Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for structured training from edureka!

### 1) What is hibernate?

Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate, and retrieve data from the database.

[more details...](https://www.javatpoint.com/hibernate-tutorial)

### 2) What is ORM?

ORM is an acronym for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation, and data access.

### 3) Explain hibernate architecture?

Hibernate architecture comprises of many interfaces such as Configuration, SessionFactory, Session, Transaction, etc.

[more details...](https://www.javatpoint.com/hibernate-architecture) 

### 4) What are the core interfaces of Hibernate?

The core interfaces of Hibernate framework are:

* Configuration
* SessionFactory
* Session
* Query
* Criteria
* Transaction

### 5) Mention some of the advantages of using ORM over JDBC.

ORM has the following advantages over JDBC:

* Application development is fast.
* Management of transaction.
* Generates key automatically.
* Details of SQL queries are hidden.

### 6) Define criteria in terms of Hibernate.

The objects of criteria are used for the creation and execution of the object-oriented criteria queries.

### 7) List some of the databases supported by Hibernate.

Some of the databases supported by Hibernate are:

* DB2
* MySQL
* Oracle
* Sybase SQL Server
* Informix Dynamic Server
* HSQL
* PostgreSQL
* FrontBase

### 8) List the key components of Hibernate.

Key components of Hibernate are:

* Configuration
* Session
* SessionFactory
* Criteria
* Query
* Transaction

### 9) Mention two components of Hibernate configuration object.

Database Connection

Class Mapping Setup

### 10) How is SQL query created in Hibernate?

The SQL query is created with the help of the following syntax:

Session.createSQLQuery

### 11) What does HQL stand for?

Hibernate Query Language

### 12) How is HQL query created?

The HQL query is created with the help of the following syntax:

Session.createQuery

### 13) How can we add criteria to a SQL query?

A criterion is added to a SQL query by using the Session.createCriteria.

### 14) Define persistent classes.

Classes whose objects are stored in a database table are called as persistent classes.

### 15) What is SessionFactory?

SessionFactory provides the instance of Session. It is a factory of Session. It holds the data of second level cache that is not enabled by default.

[more details...](https://www.javatpoint.com/hibernate-architecture)

### 16) Is SessionFactory a thread-safe object?

Yes, SessionFactory is a thread-safe object, many threads cannot access it simultaneously.

### 17) What is Session?

It maintains a connection between the hibernate application and database.

It provides methods to store, update, delete or fetch data from the database such as persist(), update(), delete(), load(), get() etc.

It is a factory of Query, Criteria and Transaction i.e. it provides factory methods to return these instances.

[more details...](https://www.javatpoint.com/hibernate-architecture)

### 18) Is Session a thread-safe object?

No, Session is not a thread-safe object, many threads can access it simultaneously. In other words, you can share it between threads.

### 19) What is the difference between session.save() and session.persist() method?

|  |  |  |
| --- | --- | --- |
| **No.** | **save()** | **persist()** |
| 1) | returns the identifier (Serializable) of the instance. | Return nothing because its return type is void. |
| 2) | Syn: public Serializable save(Object o) | Syn: public void persist(Object o) |

### 20) What is the difference between get and load method?

The differences between get() and load() methods are given below.

|  |  |  |
| --- | --- | --- |
| **No.** | **get()** | **load()** |
| 1) | Returns **null** if an object is not found. | Throws **ObjectNotFoundException** if an object is not found. |
| 2) | get() method always **hit the database**. | load() method **doesn't hit** the database. |
| 3) | It returns the real object, not the proxy. | It returns **proxy object.** |
| 4) | It should be used if **you are not sure** about the existence of instance. | It should be used if **you are sure** that instance exists. |

### 21) What is the difference between update and merge method?

The differences between update() and merge() methods are given below.

|  |  |  |
| --- | --- | --- |
| **No.** | **The update() method** | **merge() method** |
| 1) | Update means to edit something. | Merge means to combine something. |
| 2) | update() should be used if the session doesn't contain an already persistent state with the same id. It means an update should be used inside the session only. After closing the session, it will throw the error. | merge() should be used if you don't know the state of the session, means you want to make the modification at any time. |

Let's try to understand the difference by the example given below:

1. ...
2. SessionFactory factory = cfg.buildSessionFactory();
3. Session session1 = factory.openSession();
5. Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing id of employee
6. session1.close();
8. e1.setSalary(70000);
10. Session session2 = factory.openSession();
11. Employee e2 = (Employee) session1.get(Employee.class, Integer.valueOf(101));//passing same id
13. Transaction tx=session2.beginTransaction();
14. session2.merge(e1);
16. tx.commit();
17. session2.close();

After closing session1, e1 is in detached state. It will not be in the session1 cache. So if you call update() method, it will throw an error.

Then, we opened another session and loaded the same Employee instance. If we call merge in session2, changes of e1 will be merged in e2.

### 22) What are the states of the object in hibernate?

There are 3 states of the object (instance) in hibernate.

1. **Transient**: The object is in a transient state if it is just created but has no primary key (identifier) and not associated with a session.
2. **Persistent**: The object is in a persistent state if a session is open, and you just saved the instance in the database or retrieved the instance from the database.
3. **Detached**: The object is in a detached state if a session is closed. After detached state, the object comes to persistent state if you call lock() or update() method.

### 23) What are the inheritance mapping strategies?

There are 3 ways of inheritance mapping in hibernate.

1. Table per hierarchy
2. Table per concrete class
3. Table per subclass

[more details...](https://www.javatpoint.com/hibernate-inheritance-mapping-tutorial)

### 24) How to make an immutable class in hibernate?

If you mark a class as mutable="false", the class will be treated as an immutable class. By default, it is mutable="true".

### 25) What is automatic dirty checking in hibernate?

The automatic dirty checking feature of Hibernate, calls update statement automatically on the objects that are modified in a transaction.

Let's understand it by the example given below:

1. ...
2. SessionFactory factory = cfg.buildSessionFactory();
3. Session session1 = factory.openSession();
4. Transaction tx=session2.beginTransaction();
6. Employee e1 = (Employee) session1.get(Employee.class, Integer.valueOf(101));
8. e1.setSalary(70000);
10. tx.commit();
11. session1.close();

Here, after getting employee instance e1 and we are changing the state of e1.

After changing the state, we are committing the transaction. In such a case, the state will be updated automatically. This is known as dirty checking in hibernate.

### 26) How many types of association mapping are possible in hibernate?

There can be 4 types of association mapping in hibernate.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

### 27) Is it possible to perform collection mapping with One-to-One and Many-to-One?

No, collection mapping can only be performed with One-to-Many and Many-to-Many.

### 28) What is lazy loading in hibernate?

Lazy loading in hibernate improves the performance. It loads the child objects on demand.

Since Hibernate 3, lazy loading is enabled by default, and you don't need to do lazy="true". It means not to load the child objects when the parent is loaded.

### 29) What is HQL (Hibernate Query Language)?

Hibernate Query Language is known as an object-oriented query language. It is like a structured query language (SQL).

The main advantage of HQL over SQL is:

1. You don't need to learn SQL
2. Database independent
3. Simple to write a query

### 30) What is the difference between first level cache and second level cache?

|  |  |  |
| --- | --- | --- |
| **No.** | **First Level Cache** | **Second Level Cache** |
| 1) | First Level Cache is **associated with Session**. | Second Level Cache is associated with **SessionFactory**. |
| 2) | It is **enabled** by default. | It is **not enabled** by default. |

## **Hibernate Interview Questions for beginners**

### **Q1. What is Hibernate?**

[*Hibernate*](https://www.edureka.co/blog/what-is-hibernate-in-java/) is one of the most popular [*Java frameworks*](https://www.edureka.co/blog/java-frameworks/) that simplify the development of Java application to interact with the database. It is an Object-relational mapping (ORM) tool. Hibernate also provides a reference implementation of Java API.

It is referred as a framework which comes with an abstraction layer and also handles the implementations internally. The implementations include tasks like writing a query for [*CRUD*](https://www.edureka.co/blog/node-js-mysql-tutorial/) operations or establishing a connection with the databases, etc.

Hibernate develops persistence logic, which stores and processes the data for longer use. It is a lightweight tool and most importantly **open-sourced**which gives it an edge over other frameworks.

### **Q2. What are the major advantages of Hibernate Framework?**

* It is open-sourced and lightweight.
* Performance of Hibernate is very fast.
* Helps in generating database independant queries.
* Provides facilities to automatically create a table.
* It provides query statistics and database status.

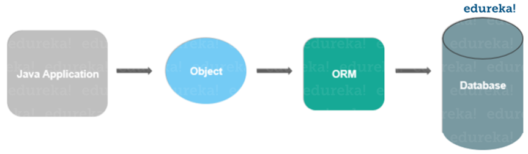
### **Q3. What are the advantages of using Hibernate over JDBC?**

Major advantages of using Hibernate over JDBC are:

1. Hibernate eliminates a lot of boiler-plate code that comes with [*JDBC API*](https://www.edureka.co/blog/connect-mysql-database-in-java/), the code looks cleaner and readable.
2. This Java framework supports [*inheritance*](https://www.edureka.co/blog/inheritance-in-java/), associations, and collections. These features are actually not present in JDBC.
3. HQL (Hibernate Query Language) is more object-oriented and close to Java. But for JDBC, you need to write native SQL queries.
4. Hibernate implicitly provides transaction management whereas, in JDBC API, you need to write code for transaction management using commit and rollback.
5. JDBC throws SQLException that is a checked exception, so you have to write a lot of try-catch block code. Hibernate wraps JDBC exceptions and throw JDBCException or HibernateException which are the unchecked exceptions, so you don’t have to write code to handle it has built-in transaction management which helps in removing the usage of try-catch blocks.

### **Q4. What is an ORM tool?**

It is basically a technique that maps the object that is stored in the database. An ORM tool helps in simplifying data creation, manipulation, and access. It internally uses the Java API to interact with the databases.



### **Q5. Why use Hibernate Framework?**

Hibernate overcomes the shortcomings of other technologies like [*JDBC*](https://www.edureka.co/blog/connect-mysql-database-in-java).

* It overcomes the database dependency faced in the JDBC.
* Changing of the databases cost a lot working on JDBC, hibernate overcomes this problem with flying colors.
* Code portability is not an option while working on JDBC. This is easily handled by Hibernate.
* Hibernate strengthens the object level relationship.
* It overcomes the [*exception-handling*](https://www.edureka.co/blog/java-exception-handling) part which is mandatory while working on JDBC.
* It reduces the length of code with increased readability by overcoming the boilerplate problem.

### **Q6. What are the different functionalities supported by Hibernate?**

* Hibernate is an ORM tool.
* Hibernate uses Hibernate Query Language(HQL) which makes it database-independent.
* It supports auto DDL operations.
* This Java framework also has an Auto Primary Key Generation support.
* Supports cache memory.
* Exception handling is not mandatory in the case of Hibernate.

### **Q7. What are the technologies that are supported by Hibernate?**

Hibernate supports a variety of technologies, like:

* XDoclet Spring
* [*Maven*](https://www.edureka.co/blog/create-selenium-maven-project/)
* Eclipse Plug-ins
* J2EE

### **Q8. What is HQL?**

HQL is the acronym of Hibernate Query Language. It is an Object-Oriented Query Language and is independent of the database.

### **Q9. How to achieve mapping in Hibernate?**

Association mappings are one of the key features of Hibernate. It supports the same associations as the relational database model. They are:

* One-to-One associations
* Many-to-One associations
* Many-to-Many associations

You can map each of them as a uni- or bidirectional association.

### **Q10. Name some of the important interfaces of Hibernate framework?**

Hibernate interfaces are:

* **SessionFactory** (org.hibernate.SessionFactory)
* **Session** (org.hibernate.Session)
* **Transaction** (org.hibernate.Transaction)

### **Q11. What is One-to-One association in Hibernate?**

In this type of mapping,  you only need to model the system for the entity for which you want to navigate the relationship in your query or domain model. You need an entity attribute that represents the association, so annotate it with an @OneToOne annotation.

### **Q12. What is One-to-Many association in Hibernate?**

In this type of association, one object can be associated with multiple/different objects. Talking about the mapping, the One-to-Many mapping is implemented using a [*Set Java*](https://www.edureka.co/blog/sets-in-java/) collection that does not have any redundant element. This One-to-Many element of the set indicates the relation of one object to multiple objects.

### **Q13. What is Many-to-Many association in Hibernate?**

Many-to-Many mapping requires an entity attribute and a @ManyToMany annotation. It can either be unidirectional and bidirectional. In **Unidirectional**, the attributes model the association and you can use it to navigate it in your domain model or JPQL queries. The annotation tells Hibernate to map a Many-to-Many association. The **bidirectional** relationship, mapping allows you to navigate the association in both directions.

### **Q14. How to integrate Hibernate and Spring?**

[*Spring*](https://www.edureka.co/blog/spring-tutorial/) is also one of the most commonly used Java frameworks in the market today. Spring is a JavaEE Framework and Hibernate is the most popular ORM framework. This is why Spring Hibernate combination is used in a lot of enterprise applications.

Following are the steps you should follow to integrate Spring and Hibernate.

1. Add Hibernate-entity manager, Hibernate-core and Spring-ORM dependencies.
2. Create Model classes and corresponding DAO implementations for database operations. The DAO classes will use SessionFactory that will be injected by the Spring Bean configuration.
3. Note that you don’t need to use Hibernate Transaction Management, as you can leave it to the Spring declarative transaction management using @Transactional annotation.

### **Q15. What do you mean by Hibernate Configuration File?**

Hibernate Configuration File mainly contains database-specific configurations and are used to initialize SessionFactory. Some important parts of the Hibernate Configuration File are Dialect information, so that hibernate knows the database type and mapping file or class details.

## **Hibernate Interview Questions for intermediate**

### **Q16. Mention some important annotations used for Hibernate mapping?**

Hibernate supports JPA annotations. Some of the major annotations are:

1. **javax.persistence.Entity:** This is used with model classes to specify they are entity beans.
2. **javax.persistence.Table:** It is used with entity beans to define the corresponding table name in the database.
3. **javax.persistence.Access:** Used to define the access type, field or property. The default value is field and if you want Hibernate to use the getter/setter methods then you need to set it to a property.
4. **javax.persistence.Id:** Defines the primary key in the entity bean.
5. **javax.persistence.EmbeddedId:** It defines a composite primary key in the entity bean.
6. **javax.persistence.Column:** Helps in defining the column name in the database table.
7. **javax.persistence.GeneratedValue:**It defines the strategy to be used for the generation of the primary key. It is also used in conjunction with javax.persistence.GenerationType enum.

### **Q17. What is Session in Hibernate and how to get it?**

Hibernate Session is the interface between Java application layer and Hibernate. It is used to get a physical connection with the database. The Session object created is lightweight and designed to be instantiated each time an interaction is needed with the database. This Session provides methods to create, read, update and delete operations for a constant object. To get the Session, you can execute HQL queries, SQL native queries using the Session object.

### **Q18. What is Hibernate SessionFactory?**

SessionFactory is the factory class that is used to get the Session objects. The SessionFactory is a heavyweight object so usually, it is created during application startup and kept for later use. This SessionFactory is a thread-safe object which is used by all the threads of an application. If you are using multiple databases then you would have to create multiple SessionFactory objects.

### **Q19. What is the difference between openSession and getCurrentSession?**

This getCurrentSession() method returns the session bound to the context and for this to work, you need to configure it in Hibernate configuration file. Since this session object belongs to the context of Hibernate, it is okay if you don’t close it. Once the SessionFactory is closed, this session object gets closed.

[[](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum)](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

### [Java Certification Training Course](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

* *[Instructor-led Sessions](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[Real-life Case Studies](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[Assignments](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*
* *[Lifetime Access](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)*

[Explore Curriculum](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=curriculum" \t "_blank)

openSession() method helps in opening a new session. You should close this session object once you are done with all the database operations. And also, you should open a new session for each request in a multi-threaded environment.

### **Q20. What do you mean by Hibernate configuration file?**

The following steps help in configuring Hibernate file:

1. First, identify the POJOs (Plain Old Java Objects) that have a database representation.
2. Identify which properties of POJOs need to be continued.
3. Annotate each of the POJOs in order to map the Java objects to columns in a database table.
4. Create a database schema using the schema export tool which uses an existing database, or you can create your own database schema.
5. Add Hibernate Java libraries to the application’s classpath.
6. Create a Hibernate XML configuration file that points to the database and the mapped classes.
7. In the Java application, you can create a Hibernate Configuration object that refers to your XML configuration file.
8. Also, build a Hibernate SessionFactory object from the Configuration object.
9. Retrieve the Hibernate Session objects from the SessionFactory and write down the data access logic for your application (create, retrieve, update, and delete).

### **Q21. What are the key components of a Hibernate configuration object?**

The configuration provides 2 key components, namely:

* Database Connection: This is handled by one or more configuration files.
* Class Mapping setup: It helps in creating the connection between Java classes and database tables.

### **Q22. Discuss the Collections in Hibernate**

Hibernate provides the facility to persist the Collections. A [*Collection*](https://www.edureka.co/blog/java-collections/) basically can be a List, Set, Map, Collection, Sorted Set, Sorted Map. java.util.List, java.util.Set, java.util.Collection, etc, are some of the real interface types to declared the persistent collection-value fields. Hibernate injects persistent Collections based on the type of interface. The collection instances generally behave like the types of value behavior.

### **Q23. What are the collection types in Hibernate?**

There are five collection types in hibernate used for one-to-many relationship mappings.

* Bag
* Set
* List
* Array
* Map

### **Q24. What is a Hibernate Template class?**

When you integrate Spring and Hibernate, Spring ORM provides two helper classes – HibernateDaoSupport and HibernateTemplate. The main reason to use them was to get two things, the Session from Hibernate and Spring Transaction Management. However, from Hibernate 3.0.1, you can use the SessionFactory getCurrentSession() method to get the current session. The major advantage of using this Template class is the **exception translation** but that can be achieved easily by using @Repository annotation with service classes.

### **Q25. What are the benefits of using Hibernate template?**

The following are the benefits of using this Hibernate template class:

* Automated Session closing ability.
* The interaction with the Hibernate Session is simplified.
* Exception handling is automated.

### **Q26. Which are the design patterns that are used in Hibernate framework?**

There are a few design patterns used in Hibernate Framework, namely:

* Domain Model Pattern: An object model of the domain that incorporates both behavior as well as data.
* Data Mapper: A layer of the map that moves data between objects and a database while keeping it independent of each other and the map itself.
* Proxy Pattern: It is used for lazy loading.
* Factory Pattern: Used in SessionFactory.

### **Q27. Define Hibernate Validator Framework**

Data validation is considered as an integral part of any application. Also, data validation is used in the presentation layer with the use of Javascript and the server-side code before processing. It occurs before persisting it in order to make sure it follows the correct format. Validation is a cross-cutting task, so we should try to keep it apart from the business logic. This Hibernate Validator provides the reference implementation of bean validation specs.

### **Q28. What is Dirty Checking in Hibernate?**

Hibernate incorporates Dirty Checking feature that permits developers and users to avoid time-consuming write actions. This Dirty Checking feature changes or updates fields that need to be changed or updated, while keeping the remaining fields untouched and unchanged.

### **Q29. How can you share your views on mapping description files?**

* Mapping description files are used by the Hibernate to configure functions.
* These files have the **\*.hbm** extension, which facilitates the mapping between database tables and Java class.
* Whether to use mapping description files or not this entirely depends on business entities.

**Q30. What is meant by Light Object Mapping?**

The means that the syntax is hidden from the business logic using specific design patterns. This is one of the valuable levels of ORM quality and this Light Object Mapping approach can be successful in case of applications where there are very fewer entities, or for applications having data models that are metadata-driven.

## **Hibernate Interview Questions for experienced**

### **Q31. What is meant by Hibernate tuning?**

Optimizing the performance of Hibernate applications is known as Hibernate tuning.

The performance tuning strategies for Hibernate are:

1. SQL Optimization
2. Session Management
3. Data Caching

### **Q32. What is Transaction Management in Hibernate? How does it work?**

Transaction Management is a property which is present in the Spring framework. Now, what role does it play in Hibernate?

Transaction Management is a process of managing a set of commands or statements. In hibernate, Transaction Management is done by transaction interface. It maintains abstraction from the transaction implementation (JTA, JDBC). A transaction is associated with Session and is instantiated by calling session.beginTransaction().

**Q33. How do you integrate Hibernate with Struts2 or Servlet web applications?**

You can integrate any Struts application with Hibernate. There are no extra efforts required.

1. Register a custom ServletContextListener.
2. In the ServletContextListener class, first, initialize the Hibernate Session, store it in the servlet context.
3. Action class helps in getting the Hibernate Session from the servlet context, and perform other Hibernate task as normal.

**Q34. What are the different states of a persistent entity?**

It may exist in one of the following 3 states:

* Transient: This is not associated with the Session and has no representation in the database.
* Persistent: You can make a transient instance persistent by associating it with a Session.
* Detached: If you close the Hibernate Session, the persistent instance will become a detached instance.

**Q35. How can the primary key be created by using Hibernate?**

A Primary key is a special relational database table column designated to uniquely identify all table records. It is specified in the configuration file hbm.xml. The generator can also be used to specify how a Primary key can be created in the database.

|  |  |
| --- | --- |
| 1  2  3  4 | <id name="ClassID" type="string" >  <column name= "columnID" length="10" >  <generator/>  </id> |

**Q36. Explain about Hibernate Proxy and how it helps in Lazy loading?**

* Hibernate uses a proxy object in order to support Lazy loading.
* When you try loading data from tables, Hibernate doesn’t load all the mapped objects.
* After you reference a child object through getter methods, if the linked entity is not present in the session cache, then the proxy code will be entered to the database and load the linked object.
* It uses Java assist to effectively and dynamically generate sub-classed implementations of your entity objects.

**Q37. How can we see Hibernate generated SQL on console?**

In order to view the SQL on a console, you need to add following in Hibernate configuration file to enable viewing SQL on the console for debugging purposes:

|  |  |
| --- | --- |
| 1 | <property name="show\_sql">true</property> |

**Q38. What is Query Cache in Hibernate?**

Hibernate implements a separate cache region for queries resultset that integrates with the Hibernate second-level cache. This is also an optional feature and requires a few more steps in code.

### Programming & Frameworks Training

### [FULL STACK WEB DEVELOPMENT INTERNSHIP PROGRAM](https://www.edureka.co/full-stack-web-development?utm_source=blogbanner&utm_campaign=trending_course" \o "Full Stack Web Development Internship Program" \t "_blank)

### [Full Stack Web Development Internship Program](https://www.edureka.co/full-stack-web-development?utm_source=blogbanner&utm_campaign=trending_course" \o "Full Stack Web Development Internship Program" \t "_blank)

*[Reviews](https://www.edureka.co/full-stack-web-development?utm_source=blogbanner&utm_campaign=trending_course" \o "Full Stack Web Development Internship Program" \t "_blank)*

**[5](https://www.edureka.co/full-stack-web-development?utm_source=blogbanner&utm_campaign=trending_course" \o "Full Stack Web Development Internship Program" \t "_blank)**[(4161)](https://www.edureka.co/full-stack-web-development?utm_source=blogbanner&utm_campaign=trending_course" \o "Full Stack Web Development Internship Program" \t "_blank)

### [JAVA CERTIFICATION TRAINING COURSE](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Java Certification Training Course" \t "_blank)

### [Java Certification Training Course](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Java Certification Training Course" \t "_blank)

*[Reviews](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Java Certification Training Course" \t "_blank)*

**[4](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Java Certification Training Course" \t "_blank)**[(66096)](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Java Certification Training Course" \t "_blank)

### [NODE.JS CERTIFICATION TRAINING COURSE](https://www.edureka.co/nodejs-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Node.js Certification Training Course" \t "_blank)

### [Node.js Certification Training Course](https://www.edureka.co/nodejs-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Node.js Certification Training Course" \t "_blank)

*[Reviews](https://www.edureka.co/nodejs-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Node.js Certification Training Course" \t "_blank)*

**[5](https://www.edureka.co/nodejs-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Node.js Certification Training Course" \t "_blank)**[(8499)](https://www.edureka.co/nodejs-certification-training?utm_source=blogbanner&utm_campaign=trending_course" \o "Node.js Certification Training Course" \t "_blank)

### [PYTHON SCRIPTING CERTIFICATION TRAINING](https://www.edureka.co/python-scripting?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Scripting Certification Training" \t "_blank)

### [Python Scripting Certification Training](https://www.edureka.co/python-scripting?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Scripting Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/python-scripting?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Scripting Certification Training" \t "_blank)*

**[5](https://www.edureka.co/python-scripting?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Scripting Certification Training" \t "_blank)**[(12779)](https://www.edureka.co/python-scripting?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Scripting Certification Training" \t "_blank)

### [ADVANCED JAVA CERTIFICATION TRAINING](https://www.edureka.co/advanced-java-sp?utm_source=blogbanner&utm_campaign=trending_course" \o "Advanced Java Certification Training" \t "_blank)

### [Advanced Java Certification Training](https://www.edureka.co/advanced-java-sp?utm_source=blogbanner&utm_campaign=trending_course" \o "Advanced Java Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/advanced-java-sp?utm_source=blogbanner&utm_campaign=trending_course" \o "Advanced Java Certification Training" \t "_blank)*

**[5](https://www.edureka.co/advanced-java-sp?utm_source=blogbanner&utm_campaign=trending_course" \o "Advanced Java Certification Training" \t "_blank)**[(5229)](https://www.edureka.co/advanced-java-sp?utm_source=blogbanner&utm_campaign=trending_course" \o "Advanced Java Certification Training" \t "_blank)

### [SPRING FRAMEWORK CERTIFICATION COURSE](https://www.edureka.co/spring-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Spring Framework Certification Course" \t "_blank)

### [Spring Framework Certification Course](https://www.edureka.co/spring-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Spring Framework Certification Course" \t "_blank)

*[Reviews](https://www.edureka.co/spring-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Spring Framework Certification Course" \t "_blank)*

**[5](https://www.edureka.co/spring-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Spring Framework Certification Course" \t "_blank)**[(10178)](https://www.edureka.co/spring-certification-course?utm_source=blogbanner&utm_campaign=trending_course" \o "Spring Framework Certification Course" \t "_blank)

### [PYTHON DJANGO CERTIFICATION TRAINING COURSE](https://www.edureka.co/python-django?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Django Certification Training Course" \t "_blank)

### [Python Django Certification Training Course](https://www.edureka.co/python-django?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Django Certification Training Course" \t "_blank)

*[Reviews](https://www.edureka.co/python-django?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Django Certification Training Course" \t "_blank)*

**[5](https://www.edureka.co/python-django?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Django Certification Training Course" \t "_blank)**[(7472)](https://www.edureka.co/python-django?utm_source=blogbanner&utm_campaign=trending_course" \o "Python Django Certification Training Course" \t "_blank)

### [PHP & MYSQL WITH MVC FRAMEWORKS CERTIFICATION TRAINING](https://www.edureka.co/php-mysql-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)

### [PHP & MySQL with MVC Frameworks Certification Training](https://www.edureka.co/php-mysql-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/php-mysql-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)*

**[5](https://www.edureka.co/php-mysql-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)**[(4447)](https://www.edureka.co/php-mysql-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)

### [MICROSOFT .NET FRAMEWORK CERTIFICATION TRAINING](https://www.edureka.co/microsoft-dotnet-framework-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "Microsoft .NET Framework Certification Training" \t "_blank)

### [Microsoft .NET Framework Certification Training](https://www.edureka.co/microsoft-dotnet-framework-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "Microsoft .NET Framework Certification Training" \t "_blank)

*[Reviews](https://www.edureka.co/microsoft-dotnet-framework-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "Microsoft .NET Framework Certification Training" \t "_blank)*

**[5](https://www.edureka.co/microsoft-dotnet-framework-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "Microsoft .NET Framework Certification Training" \t "_blank)**[(5132)](https://www.edureka.co/microsoft-dotnet-framework-self-paced?utm_source=blogbanner&utm_campaign=trending_course" \o "Microsoft .NET Framework Certification Training" \t "_blank)

Next

***Note:*** This is only useful for queries that are run frequently with the same parameters.

**Q39. What is the benefit of Native SQL query support in Hibernate?**

Hibernate provides an option to execute Native SQL queries through the use of the [SQLQuery](https://www.edureka.co/blog/insert-query-sql/)object. For normal scenarios, it is however not the recommended approach because you might lose other benefits like Association and Hibernate first-level caching.

Native SQL Query comes handy when you want to execute database-specific queries that are not supported by Hibernate API such query hints or the Connect keyword in Oracle Database.

**Q40. What is Named SQL Query?**

Hibernate provides another important feature called Named Query using which you can define at a central location and use them anywhere in the code.

You can create named queries for both HQL as well as for Native SQL. These Named Queries can be defined in Hibernate mapping files with the help of JPA annotations @NamedQuery and @NamedNativeQuery.

**Q41. When do you use merge() and update() in Hibernate?**

This is one of the tricky Hibernate Interview Questions asked.

update(): If you are sure that the Hibernate Session does not contain an already persistent instance with the same id .  
*merge():*  Helps in merging your modifications at any time without considering the state of the Session.

**Q42. Difference between get() vs load() method in Hibernate?**

This is one of the most frequently asked Hibernate Interview Questions. The key difference between the get() and load() method is:

load(): It will throw an exception if an object with an ID passed to them is not found.  
get():  Will return null.

load(): It can return proxy without hitting the database unless required.  
get(): It always goes to the database.

So sometimes using load() can be faster than the get() method.

**Q43. Difference between the first and second level cache in Hibernate?**

The first-level cache is maintained at Session level while the second level cache is maintained at a SessionFactory level and is shared by all sessions.

**Q44. Difference between Session and SessionFactory in Hibernate?**

This is yet another popular Hibernate Interview Question asked.

* *A Session* is a single-threaded, short-lived object. It provides the first-level cache.
* SessionFactory is immutable and shared by all Session. It also lives until the Hibernate is running. It also provides the second-level cache.

**Q45. Difference between save() and saveOrUpdate() method of Hibernate?**

Even though save() and saveOrUpdate() method is used to store an object into Database, the key difference between them is that save() can only **Insert** records but saveOrUpdate() can either Insert or Update records.

**Q46. Difference between sorted and ordered collection in Hibernate?**

sorted collection sort the data in JVM’s heap memory using Java’s collection framework sorting methods. The ordered collection is sorted using order by clause in the database itself.

***Note:*** A sorted collection is more suited for small dataset but for a large dataset, it’s better to use ordered collection to avoid

**Q47. Difference between the transient, persistent and detached state in Hibernate?**

**Transient state:** New objects are created in the Java program but are not associated with any Hibernate Session.

**Persistent state:** An object which is associated with a Hibernate session is called Persistent object. While an object which was earlier associated with Hibernate session but currently it’s not associate is known as a detached object. You can call save() or persist() method to store those object into the database and bring them into the Persistent state.

[[](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=batches)](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=batches" \t "_blank)

### [Java Certification Training Course](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=batches" \t "_blank)

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/java-j2ee-training-course?utm_source=blogbanner&utm_campaign=batches" \t "_blank)

**Detached state**: You can re-attach a detached object to Hibernate sessions by calling either update() or saveOrUpdate() method.

**Q48. Difference between managed associations and Hibernate associations?**

**Managed associations:** Relate to container management persistence and are bi-directional.

**Hibernate Associations:** These associations are unidirectional.

V

Database

# Indexing in Databases:

## What is Indexing?

**Indexing** is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of [pointers](https://www.guru99.com/c-pointers.html) for holding the address of the disk block where that specific key value stored.

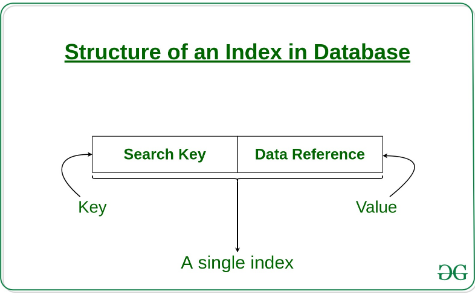
An index –

* Takes a search key as input
* Efficiently returns a collection of matching records.

# It is a data structure technique which is used to quickly locate and access the data in a database

Indexes are created using a few database columns.

* The first column is the **Search key** that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly.   
  *Note: The data may or may not be stored in sorted order.*
* The second column is the **Data Reference** or **Pointer** which contains a set of pointers ho

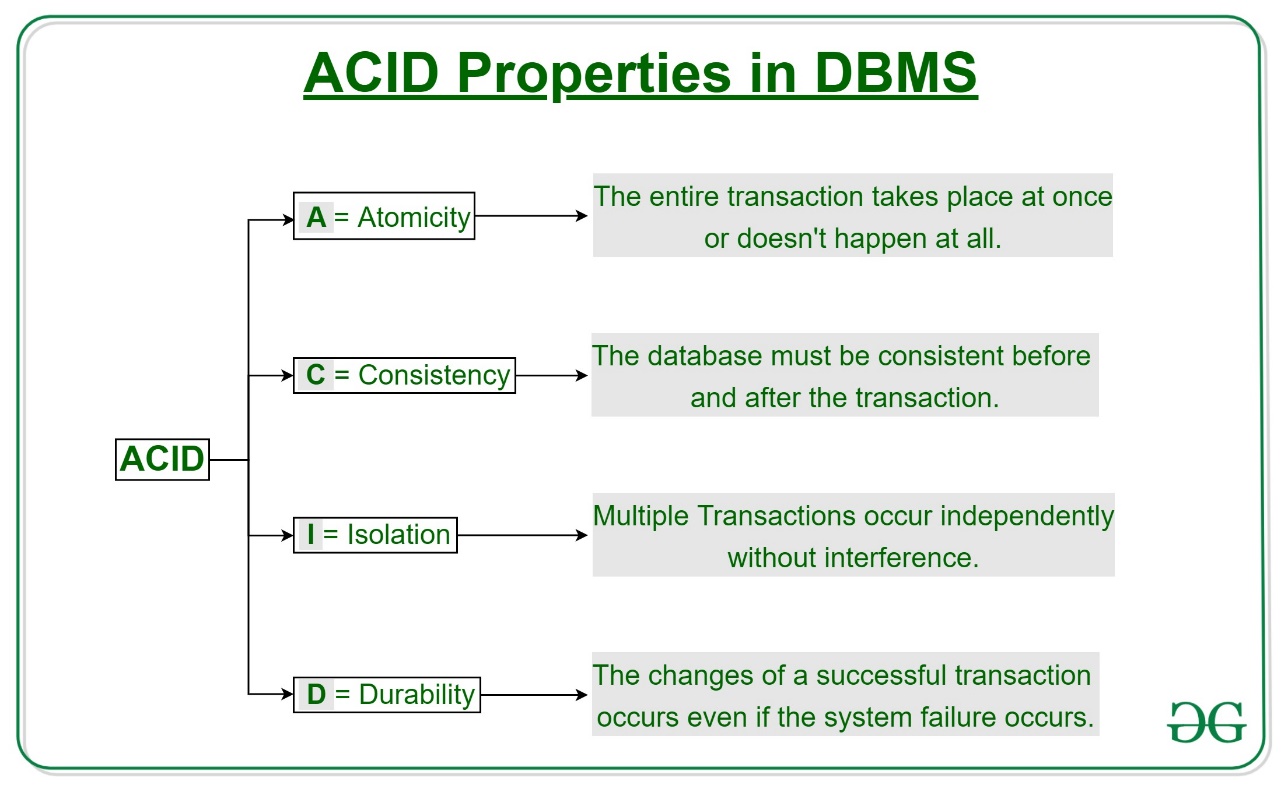


The indexing has various attributes:

* **Access Types**: This refers to the type of access such as value based search, range access, etc.
* **Access Time**: It refers to the time needed to find particular data element or set of elements.
* **Insertion Time**: It refers to the time taken to find the appropriate space and insert a new data.
* **Deletion Time**: Time taken to find an item and delete it as well as update the index structure.
* **Space Overhead**: It refers to the additional space required by the index

# ACID Properties in DBMS

ACID Properties in SQL Server ensure Data Integrity during a transaction. The ACID is an acronym for Atomicity, Consistency, Isolation, and Durability.



**Some important points:**

| **Property** | **Responsibility for maintaining properties** |
| --- | --- |
| Atomicity | Transaction Manager |
| Consistency | Application programmer |
| Isolation | Concurrency Control Manager |
| Durability | Recovery Manager |

**1. Relational Database :**   
RDBMS stands for Relational Database Management Systems. It is most popular database. In it, data is store in the form of row that is in the form of tuple. It contain numbers of table and data can be easily accessed because data is store in the table. This Model was proposed by E.F. Codd.

**2. NoSQL :**   
NoSQL Database stands for a non-SQL database. NoSQL database doesn’t use table to store the data like relational database. It is used for storing and fetching the data in database and generally used to store the large amount of data. It supports query language and provides better performance.

**Difference between Relational database and NoSQL :**

|  |  |
| --- | --- |
| Relational Database | NoSQL |
| It is used to handle data coming in low velocity. | It is used to handle data coming in high velocity. |
| It gives only read scalability. | It gives both read and write scalability. |
| It manages structured data. | It manages all type of data. |
| Data arrives from one or few locations. | Data arrives from many locations. |
| It supports complex transactions. | It supports simple transactions. |
| It has single point of failure. | No single point of failure. |
| It handles data in less volume. | It handles data in high volume. |
| Transactions written in one location. | Transactions written in many locations. |
| support ACID properties compliance | doesn’t support ACID properties |
| Its difficult to make changes in database once it is defined | Enables easy and frequent changes to database |
| schema  is mandatory to store the data | schema design is not required |
| Deployed in vertical fashion. | Deployed in Horizontal fashion. |

### Features of relational databases

* They work with structured data.
* Relationships in the system have constraints, which promotes a high level of data integrity.
* There are limitless indexing capabilities, which results in faster query response times.
* They are excellent at keeping data transactions secure.
* They provide the ability to write complex SQL queries for data analysis and reporting.
* Their models can ensure and enforce business rules at the data layer adding a level of data integrity not found in a non-relational database.
* They are table and row oriented.
* They Use SQL (structured query language) for shaping and manipulating data, which is very powerful.
* SQL database examples: [MySql](https://www.pluralsight.com/paths/mysql), [Oracle](https://www.pluralsight.com/courses/oracle-database-12c-fundamentals), Sqlite, Postgres and MS-SQL. NoSQL database examples: MongoDB, [BigTable](https://www.pluralsight.com/courses/google-bigtable-architecting-big-data-solutions), Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb.
* SQL databases are best fit for heavy duty transactional type applications.

### Features of non-relational databases

* They have the ability to store large amounts of data with little structure.
* They provide scalability and flexibility to meet changing business requirements.
* They provide schema-free or schema-on-read options.
* They have the ability to capture all types of data “Big Data” including unstructured data.
* They are document oriented.
* NoSQL or non-relational databases examples: MongoDB, Apache Cassandra, Redis, Couchbase and Apache HBase.
* They are best for Rapid Application Development. NoSQL is the best selection for flexible data storage with little to no structure limitations.
* They provide flexible data model with the ability to easily store and combine data of any structure without the need to modify a schema.

SOLID

They all serve the same purpose:

"To create understandable, readable, and testable code that many developers can collaboratively work on."

Let's look at each principle one by one. Following the SOLID acronym, they are:

* The **S**ingle Responsibility Principle
* The **O**pen-Closed Principle
* The **L**iskov Substitution Principle
* The **I**nterface Segregation Principle
* The **D**ependency Inversion Principle

## The Single Responsibility Principle

The Single Responsibility Principle states that **a class should do one thing and therefore it should have only a single reason to change**.

## Open-Closed Principle

The Open-Closed Principle requires that **classes should be open for extension and closed to modification.**

## Liskov Substitution Principle

The Liskov Substitution Principle states that subclasses should be substitutable for their base classes.

This means that, given that class B is a subclass of class A, we should be able to pass an object of class B to any method that expects an object of class A and the method should not give any weird output in that case

## Interface Segregation Principle

Segregation means keeping things separated, and the Interface Segregation Principle is about separating the interfaces.

The principle states that many client-specific interfaces are better than one general-purpose interface. Clients should not be forced to implement a function they do no need.

## Dependency Inversion Principle

The Dependency Inversion principle states that our classes should depend upon interfaces or abstract classes instead of concrete classes and functions.

In his [article](https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf)(2000), Uncle Bob summarizes this principle as follows:

"If the OCP states the goal of OO architecture, the DIP states the primary mechanism".

These two principles are indeed related and we have applied this pattern before while we were discussing the Open-Closed Principle.