

[Home](#) › [Interview](#) › [Spring, Hibernate, & Maven Interview Q&A](#) › [Hibernate](#) › 01:

♥♦ 15+ Hibernate basics Q1 – Q7 interview questions & answers

# 01: ♥♦ 15+ Hibernate basics Q1 – Q7 interview questions & answers

Posted on [August 27, 2014](#) by [Arulkumaran Kumaraswamipillai](#)

Hibernate interview questions are as popular as [Spring interview questions & answers](#).

**Q1.** What is an Object-to-Relational Mapping (i.e. ORM) tool?

**A1.** There are several ways to persist data to a database from your Java applications.

**1) Direct JDBC** SQL statements for the CRUD (Create, Read, Uppdate, and Dele) operations. Direct JDBC is more SQL centric. You need to write all the SQL statements. This is covered in detail at [JDBC Overview Interview Questions and Answers](#).

**600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs**

[open all](#) | [close all](#)

[Ice Breaker Interview](#)

[Core Java Interview C](#)

[JEE Interview Q&A \(3](#)

[Pressed for time? Jav](#)

[SQL, XML, UML, JSC](#)

[Hadoop & BigData Int](#)

[Java Architecture Inte](#)

[Scala Interview Q&As](#)

[Spring, Hibernate, & I](#)

[Spring \(18\)](#)

[Hibernate \(13\)](#)

[01: ♥♦ 15+ Hiber](#)

[01b: ♦ 15+ Hiber](#)

[02: Understandir](#)

[03: Identifying ar](#)

[04: Identifying ar](#)

[05: Debugging H](#)

[06: Hibernate Fii](#)

[07: Hibernate mi](#)

[08: Hibernate au](#)

[09: Hibernate en](#)

[10: Spring, Java](#)

```

2  try (
3      Connection connection = ds.getConnection()
4      Statement statement = connection.createStatement()
5      ResultSet rs = statement.executeQuery("SELECT * FROM CUSTOMER")
6  ) {
7      rs.next();
8      final int c = rs.getInt(1);
9  }
10

```

**2) Persistence frameworks** like Spring JDBC Template, iBatis, etc to make things easier to use. Spring JDBC Template is very popular, and provides an abstraction over JDBC.

```

1
2  final int count = new JdbcTemplate(ds).queryForInt("SELECT COUNT(*) FROM CUSTOMER");
3

```

**3) ORM frameworks** like Hibernate, TopLink from Oracle, etc which are more object centric. You create your domain objects in Java and use Hibernate annotations to map your objects to underlying database tables and columns, and ORM tool will do a lot of work like generating the CRUD SQL statements, caching, optimistic locking, etc. These out of the box benefits come at a price of increased complexity. Using ORM tools like Hibernate as a black-box without properly understanding its object life cycle, proxying concepts, caching strategies, lazy loading vs eager fetching, etc can not only cause performance issues, but also lots of frustrations during development.

**Spring-data** with Hibernate: Spring Data unifies the access to different kinds of persistence stores, both relational database systems and NoSQL data.

```

1
2  public interface CustomerRepository extends CrudRepository<Customer, Long> {
3      Long count();
4  }
5

```

**Q2.** Where will you use ORM tools like Hibernate as opposed to using direct Spring JDBC Template?

**A2.**

- 11: Hibernate de
- 12: Hibernate cu
- AngularJS (2)
- Git & SVN (6)
- JMeter (2)
- JSF (2)
- Maven (3)
- Testing & Profiling/Sa
- Other Interview Q&A 1
- Free Java Interview

## 16 Technical Key Areas

[open all](#) | [close all](#)

- Best Practice (6)
- Coding (26)
- Concurrency (6)
- Design Concepts (7)
- Design Patterns (11)
- Exception Handling (3)
- Java Debugging (21)
- Judging Experience I
- Low Latency (7)
- Memory Management
- Performance (13)
- QoS (8)
- Scalability (4)
- SDLC (6)
- Security (13)
- Transaction Managen

## 80+ step by step Java Tutorials

[open all](#) | [close all](#)

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2)

1) If you have complete control of your database schema and if you don't have an extremely high throughput requirement then Hibernate can work quite well when used properly by the developers who know what they are using. Never use it as a black box framework, which can lead to performance issues and frustrations during development.

2) ORM is a technique of mapping data representation from an object model to a SQL based relational model, and is well suited for "read, modify and write centric applications" and not suited for bulk write centric applications like batch processes with large data sets like 5000 rows or more where data is seldom read. Although this was generally true of many earlier ORM mapping frameworks, most today's ORM tools including Hibernate allow for efficient ways of performing large batch style write operations.

**Q3.** What are the features & strengths of ORM tools?

**A3.** ORM tools/frameworks allow your application to be:

- **Less verbose** (e.g. transparent persistence , Object Oriented query language , transitive persistence, etc)
- **More portable** (i.e. vendor independence due to multi dialect support ).
- **More maintainable** (i.e. transparent persistence, inheritance mapping strategies, automatic dirty checking etc).

Takes care of much of the plumbing like

- 1) Connection establishment & Multi-dialect support.
- 2) SQL code generation under the covers.
- 3) Cache management.
- 4) Optimistic locking.
- 5) Support for auditable fields like CreatedBy, UpdatedBy, CreatedDateTime, and UpdatedDateTime.
- 6) Soft deletes with @SQLDelete annotation.
- 7 Exception handling.

and you can often focus on the functional aspects. You can also leverage the framework's strategies and capabilities to get efficiencies via features such as eager fetching, lazy

- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hlbernat](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

## 100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

## How good are your .....?

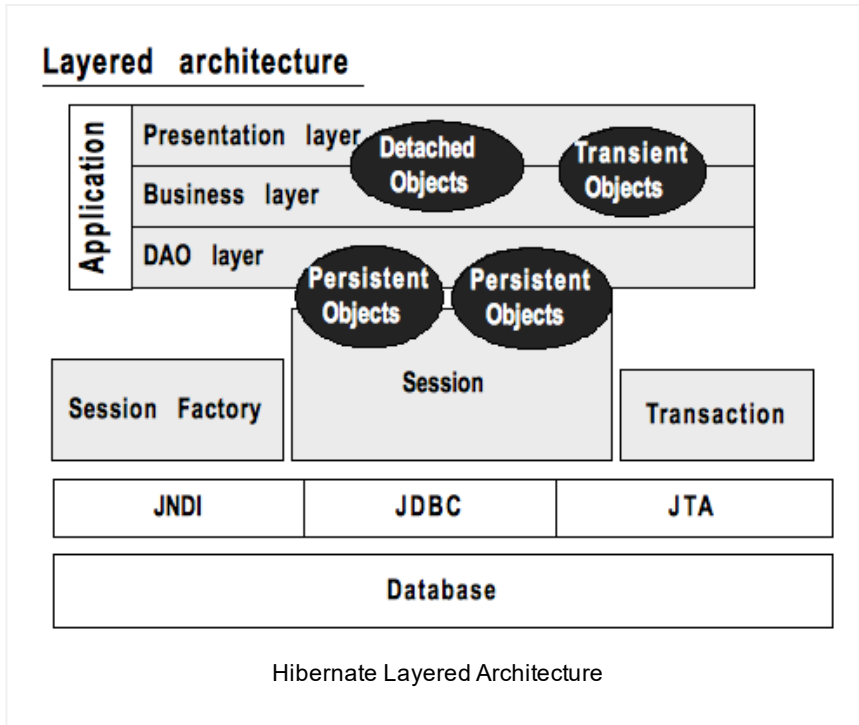
[open all](#) | [close all](#)

- [Career Making Know-](#)
- [Job Hunting & Resum](#)

loading (i.e. using proxy objects), caching strategies, detached objects to be passed to the other layers, dirty checking for optimistic concurrency control, etc.

**Q4.** Can you explain the Hibernate architecture?

**A4.** The Hibernate architecture is layered to keep you isolated from having to know the underlying APIs. Configuration, SessionFactory, Session, Transaction, Query, and Criteria are some of the Hibernate objects.



**Q5.** What is a SessionFactory object? Is it a thread-safe object?

**A5.** **SessionFactory** is Hibernate's concept of a single datastore and is threadsafe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database. A SessionFactory is usually only built once at startup. SessionFactory should be wrapped in some kind of singleton so that it can be easily accessed & reused in an application code.

```
1 SessionFactory sessionFactory = new Configuration
```

**Q6.** What is a Session object? Can you share a session object between different threads?

**A6. Session** is a light weight and a non-threadsafe object. So, you cannot share it between threads. Session represents a single unit-of-work with the database. Sessions are opened by a SessionFactory and then are closed when all work is complete. Session is the primary interface for the persistence service. A session obtains a database connection lazily (i.e. only when required). To avoid creating too many sessions, a ThreadLocal object can be used to have a session per thread. In JPA 2.0, you use an **EntityManager** instead of a Session.

**Q7.** What is the link between EntityManagerFactory/EntityManager and SessionFactory/Session?

**A7. EntityManagerFactory** and **EntityManager** are **JPA** (java Persistence API) specific. **SessionFactory** and **Session** are **hibernate**-specific.

**Q8.** Why favor **JPA APIs** like EntityManagerFactory and EntityManager over **Hibernate APIs** like SessionFactory & Session?

**A8.**

**1)** JPA API is a standard and evolving whereas Hibernate API is proprietary. So, use “javax.persistence.\*” annotations as opposed to “org.hibernate.annotations.\*”. Standard APIs in theory has better portability than proprietary APIs. This does not mean that you can easily port your application to use other ORM tools like TopLink or EclipseLink.

**2)** JPA API has better semantics than Hibernate as it has rectified some of the the early mistakes of Hibernate API.

**3)** The EntityManager invokes the hibernate session under the hood, and if you need some specific features that are not available in the EntityManager, you can obtain the session by calling:

```
1  
2 Session session = entityManager.unwrap(Session.cl  
3
```

OR

```
1  
2 Session session = (Session) entityManager.getDele  
3
```

4) EntityManagerFactory approach allows us to use callback method annotations like @PrePersist, @PostPersist, and @PreUpdate with no extra configuration. Using similar callbacks SessionFactory requires extra effort.

**Q8 – Q15** 15+ Hibernate basics Q8 – Q15 interview questions & answers

## More Hibernate Interview Questions & Answers

1. [Hibernate mistakes](#)
2. [Hibernate cache First & second level interview questions and answers](#)
3. [Hibernate automatic dirty checking](#)
4. [Hibernate entities with auditable, soft delete & optimistic locking fields](#)
5. [Understanding Hibernate proxy objects](#)
6. [Hibernate custom data type](#)
7. [8 JPA interview questions and answers](#)

## Popular Member Posts

♦ 11 Spring boot interview questions & answers

904 views

## ♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

816 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

427 views

18 Java scenarios based interview Questions and Answers

408 views

♦ 7 Java debugging interview questions & answers

323 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

311 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

303 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views

001B: ♦ Java architecture & design concepts interview questions & answers

209 views

30

Like

Share

Tweet

↑

submit

↓

reddit

2

G+1

Share

51

Bio

Latest Posts



### Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java



interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



#### About [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ ♦ 15 Java old I/O and NIO (i.e. New I/O) interview Q&As

JTA interview Q&A ▶

**Posted in** Hibernate, Hibernate Job Interview Essentials

**Tags:** Java/JEE FAQs, JEE FAQs, Popular frameworks

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)  
 ☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

### Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

## Prepare to succeed



☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.