1. What is Hibernate?
Hibernate is a Java ORM framework that maps Java classes to database tables, allowing developers to work with objects instead of SQL.

2. What is ORM?
ORM maps Java objects to database tables and handles CRUD automatically.

3. Advantages of Hibernate?
DB independence, caching, lazy loading, no SQL, automatic schema generation.

4. Components of Hibernate architecture?
Session, SessionFactory, Transaction, Query, Criteria, Cache.

5. States of Hibernate entity?
Transient, Persistent, Detached.

6. What is SessionFactory?
Thread-safe factory for creating Session objects.

7. Difference between get() and load()?
get returns null; load returns proxy and throws exception if not found.

8. What is lazy loading?
Loads associated objects only when accessed.

9. What is cascading?
Parent operations propagate to child entities.

10. What is Hibernate caching?
1st level (session) and 2nd level (SessionFactory).

11. What is HQL?
Object-oriented query language using entity names.

12. Example HQL query?
FROM Employee WHERE salary > :sal

13. HQL vs SQL?
HQL uses entity names; SQL uses tables.

14. Does HQL support joins?
Yes, using JOIN keyword.

15. Pagination in HQL?
setFirstResult() and setMaxResults().

16. What is a named query?
Predefined HQL in entity class.

17. HQL update example?
UPDATE Employee SET salary=:s WHERE id=:id

18. Native SQL in Hibernate?
session.createSQLQuery().

19. What is projection?
Selecting specific fields instead of whole entity.

20. N+1 problem?
Multiple queries for child collections; solved via JOIN FETCH.

21. What is JPA?
Java ORM specification; Hibernate is an implementation.

22. JPA vs Hibernate?
JPA is spec; Hibernate is provider.

23. What is an entity?
POJO annotated with @Entity.

24. Purpose of @Id and @GeneratedValue?
PK and strategy for ID generation.

25. @OneToMany & @ManyToOne?
Defines entity relationships.

26. LAZY vs EAGER?
Lazy loads on access; eager loads immediately.

27. @Transactional purpose?
Defines transaction boundaries.

28. What is JPQL?
JPA query language similar to HQL.

29. HQL vs JPQL?
HQL is Hibernate-specific; JPQL is JPA standard.

30. JPA lifecycle states?
New, Managed, Detached, Removed.

31. Persistence Context?
Environment tracking entity changes.

32. Dirty Checking?
Hibernate auto-detects entity field changes.

33. EntityManager?

JPA interface for CRUD and queries.

34. Fetch Join?
JOIN FETCH to solve N+1 problem.

35. Criteria API?
Type-safe dynamic query building.