**Introduction to HQL and JPQL**

* **HQL (Hibernate Query Language)** is an **object-oriented query language** developed as part of the Hibernate framework.
* **JPQL (Java Persistence Query Language)** is defined by the **Java Persistence API (JPA)** specification and is used to interact with databases in an object-oriented way.

**Key Similarities:**

* Both **HQL and JPQL** are similar in syntax to **SQL**, but they work with **Java objects** and **entity classes** instead of directly with database tables.
* They support **SELECT**, **UPDATE**, and **DELETE** operations.
* Both allow queries to be written using the **entity names and attributes**, not table names or columns.

**Key Differences:**

| **Feature** | **HQL** | **JPQL** |
| --- | --- | --- |
| Ownership | Part of Hibernate | Part of JPA specification |
| Syntax Base | Object-oriented SQL-like | Subset of HQL |
| INSERT Operation | Supported | **Not supported** |
| Native SQL Conversion | Hibernate handles SQL conversion | JPA providers (e.g., Hibernate) do it |

* **JPQL is a subset of HQL**, meaning **every JPQL query is a valid HQL query**, but **not all HQL queries are valid JPQL**.
* **INSERT operations** are allowed in **HQL** but **not in JPQL**.

**Example Syntax:**

// JPQL / HQL SELECT example

Query q = entityManager.createQuery("SELECT e FROM Employee e WHERE e.salary > :minSal");

// HQL INSERT example (not allowed in JPQL)

Query q = session.createQuery("INSERT INTO Employee(name, salary) SELECT e.name, e.salary FROM OldEmployee e");