

GLOSSARY

Term	Meaning
@AttributeOverride	Annotation that allows you to override the columns to which the embedded class's properties are mapped.

@AttributeOverrides	Annotation encapsulating multiple @AttributeOverride annotations.
@CollectionId	Annotation to generate a primary key for the table corresponding to the attribute that specifies the collection.
@Column	Annotation whose name property is used to specify a different name for the

	corresponding table column
@DiscriminatorColumn	Annotation that specifies the discriminator column for the SINGLE_TABLE and JOINED inheritance mapping strategies.
@DiscriminatorValue	Annotation that specifies the value of the discriminator column for entities of the given type.
@ElementCollection	Annotation which

	defines a collection of objects of a basic type or embeddable class.
@Embeddable	Annotation for a value type which indicates that its object will be stored as an intrinsic part of the Entity object and its persistent attributes will have corresponding columns in the table corresponding to the entity class.

@EmbeddedId	Annotation that denotes a composite primary key that is an embeddable class.
@Entity	Annotation used to specify that the class that is annotated with it is an entity i.e. a persistent class.
@GeneratedValue	Annotation which can be used to annotate an attribute that is annotated with @Id annotation to automatically generate

	values for the primary key.
@Id	Annotation that declares the identifier attribute of an entity class.
@Inheritance	Annotation that specifies the inheritance strategy that must be used for an entity class hierarchy. It is specified on the entity class that is the root

	of the entity class hierarchy.
@JoinColumn	Annotation whose name property is used to specify the name of the foreign key and is used as the value of the joinColumns property of the @JoinTable annotation.
@JoinTable	Annotation that makes it possible to assign our own names to the table corresponding to the

	collection and the foreign key that will be generated, when @ElementCollection is used to define an object collection.
@Lob	Annotation to indicate that the particular attribute must be persisted as a Large object.
@ManyToMany	Annotation which defines a many-valued association

	with many-to-one multiplicity
@ManyToOne	Annotation which defines a single-valued association to another entity class that has many-to-one multiplicity.
@NamedNativeQueries	Annotation used to encapsulate multiple @NamedNativeQuery annotations
@NamedNativeQuery	Annotation that is

	used to specify a named native query.
@NamedQueries	Annotation used to encapsulate multiple @NamedQuery annotations
@NamedQuery	Annotation that is used to specify a named query.
@OneToMany	Annotation which defines a many-valued association with one-to-many multiplicity.

@PrimaryKeyJoinColumn	Annotation that is used to join the table corresponding to a subclass in the JOINED mapping strategy to the table of its superclass. It specifies a primary key column that is used as a foreign key to join another table.
@SelectBeforeUpdate	Annotation that is used to decide whether to select the entity's current state from

	the database when determining whether to perform an update operation while reattaching detached objects.
@Table	Annotation that is used to specify a name for a table that will be created due to the usage of @Entity annotation
@Temporal	Annotation for attributes of type Date to describe expected

	precision (only date, only time or both date and time) in the database.
@Transient	Annotation to indicate that the particular attribute should not be persisted.
Abstraction	A feature of Object-Oriented Programming Paradigm. It means that a class is an abstract unit. If a task that is performed

	involves an object of the class, a message must be sent to the object asking it to execute the desired method.
Aggregate Function	Function that performs calculations on a set of values and returns a single value.
AUTO	Primary key generation strategy which indicates that Hibernate must select

	the strategy for the particular database
Caching	A facility in Hibernate that stores objects in memory and makes it unnecessary for the application to access the database repeatedly for the objects.
Cascading	Concept of persisting many objects in a collection through the invocation of a single method instead

	of persisting each object in a collection separately.
Composite Primary Key	A primary key of a table that is composed of more than 1 column.
Configuration	A class whose object allows the application to specify the properties when a session factory is being created.
Criteria	Interface used to retrieve entities by

	composing Criterion objects.
Criterion	Interface which is an object-oriented representation of a query criterion.
CRUD operations	The create, read, update and delete operations.
Detached State	State of an object which was in a persistent state and then its session was closed. Any change that is made to an

	object in the detached state is not reflected in the database.
Dynamic Binding	A technique in which the code to be executed in response to a method invocation is determined only at runtime.
Eager Fetch	A type of fetch in which a collection is fetched at the same time as the object in which this collection

	is encapsulated, is fetched.
Encapsulation	A feature of Object-Oriented Programming Paradigm. It means that the properties and the methods are grouped together in a class. The outside world can interact with the data stored in the variables representing the properties only

	through the methods of that class.
Entity type	An independent type that usually represents a real-world entity, contains attributes and has a meaning on its own.
Example	A class in the org.hibernate.criterion package which assists in the implementation of the Query By Example concept.
from clause	HQL clause that

	corresponds to the selection of all columns of all records from a table in SQL.
group by clause	HQL clause that groups aggregate values returned by a query by any property of the returned class.
Hibernate Query Language (HQL)	An Object-Oriented query language that operates on persistent objects and their properties.
IDENTITY	Primary key

	generation strategy which indicates that Hibernate must generate primary key values by using an identity column
Implementation-hiding	A feature of Object-Oriented Programming Paradigm. It means that the manner in which data is stored in a property or a task is implemented inside a

	method is hidden from the outside world.
Inheritance	A concept in Object-Oriented Programming Paradigm in which a class inherits the features of another class.
InheritanceType. JOINED	Value of the strategy property of the @Inheritance annotation in which a table is generated for each class in the

	inheritance structure , with each table corresponding to the subclasses storing the attributes of the subclasses only and not the inherited attributes.
InheritanceType. TABLE_PER_CLASS	Value of the strategy property of the @Inheritance annotation in which a table is generated for each class in the inheritance structure

InheritanceType.SINGLE_TABLE	Value of the strategy property of the @Inheritance annotation in which a single table is generated for all classes in the inheritance structure.
InheritanceType	Enum that defines the inheritance strategy options. It has three constants namely JOINED, SINGLE_TABLE and TABLE_PER_CLASS.

Java Persistence API	A Java specification for accessing, persisting, and managing data between Java objects/ classes and a relational database.
Lazy Fetch	A type of fetch in which a collection is fetched only when it is accessed through its getter method in the session.
like clause	HQL clause that is used along with the where

	clause to filter results.
Method Overloading	A term that is used when many methods of a class have the same name but different signatures. A method signature is the number and/or type of input parameters in the method.
Method Overriding	The redefinition of a method of a class in the subclasses of that class.

Name placeholders	Placeholders for parameter values used in a query based on the names of the parameters in the query.
Named native query	Named query specified by using native SQL.
Named query	HQL query that is uniquely named and invoked by using the name.
Natural key	A unique key that is formed of attributes

	that already exist in the real world.
Object-Oriented Programming Paradigm	A paradigm that is based on the concept of an object which is an instance of a class.
OneToOne	Annotation which defines a single-valued association to another entity that has one-to-one multiplicity.
order by clause	HQL clause that is used to sort the query result in ascending or descending order

	based on specific columns.
Pagination	Process of dividing a large number of records into multiple parts and retrieving and displaying the parts.
Parameter Binding	A technique of passing values to a query by using placeholders in place of actual values and then setting the values.
Persistent State	State of an object when

	it has a corresponding representation in the database and an identifier value. When an object is in the persistent state, it is within the scope of a Hibernate session.
Polymorphism	It means multiple forms. Both Method Overriding and Method Overloading are types of polymorphism.
Positional placeholders	Placeholders for

	parameter values used in a query based on their position sequence in the query.
Projections	Concept by which only a subset of the attributes of entity/entities that are queried with Criteria are queried. Projections are also used to specify aggregation functions.
Projections	Class in the <code>org.hibernate.criterion</code>

	package having many static methods.
Query	An interface that is an Object-Oriented representation of Hibernate query.
Query By Example	A technique of retrieving objects in which an Example object can be created and all objects that are similar to this example object can be retrieved.
Query Cache	Hibernate cache that is used to cache the

	results of a query.
Relational Database	A database that is based on the relational model of data. It stores information in tables.
Relational Database Management System	A software system that is used to maintain relational databases.
Restrictions	Class containing many methods that return a Criterion object.
Second level Cache	Hibernate cache responsible for caching objects across sessions

select clause	HQL clause that is used to select multiple properties of an Entity class
SEQUENCE	Primary key generation strategy which indicates that Hibernate must generate primary key values by using a database sequence.
Session	Main interface between a Java application and Hibernate.

Session Cache	Hibernate cache, enabled by default, that caches objects in the current session.
SessionFactory	Interface used to create Session instance.
SQL injection	Process of injecting malicious code in SQL statements through web pages.
Surrogate key	Attribute created by the developer that appears as an

	additional column in the generated table. It can also be used when no attribute of the entity can uniquely identify it.
TABLE	Primary key generation strategy which indicates that Hibernate must generate primary key values by using a database table
Transaction	Interface that allows the application to

	define units of work.
Transient state	State of an object when it has been created by using the new operator and it has not yet been associated with a Hibernate session. When an object is in the transient state, no record corresponding to this object exists in the corresponding database table.
Value type	A dependent type

	containing attributes, having no meaning on its own and acquiring a meaning only when it is associated with an Entity type.
where clause	HQL clause that is used to select specific objects.