Parts, Suppliers and Shipments — are defined as ordinary Java classes.

The serialized form of the key and value objects is stored directly in the database.

An important point is that instances of these classes are passed and returned by value, not by reference, when they are stored and retrieved from the database.

An important characteristic of these classes is that they are independent of the database. Therefore, they may be easily used in other contexts.

The database environment manages resources (for example, memory, locks and transactions) for any number of databases. A single environment instance is normally used for all databases.

The home directory is the location of the environment's log files that store all database information.

The class descriptions are stored in the catalog rather than storing them redundantly in each database record. A single class catalog per environment must be opened whenever serialized objects will be stored in the database.

Stored collections use bindings transparently to convert the records to objects when they are retrieved from the collection, and to convert the objects to records when they are stored in the collection.

The SampleViews class is used to create the bindings and collections.

A [StoredSortedMap](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/collections/StoredMap.html) field is used for each database. The StoredSortedMap class implements the standard Java [Map](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Map.html) interface, which has methods for obtaining a [Set](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Set.html) of keys, a [Collection](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Collection.html) of values, or a [Set](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Set.html) of [Map.Entry](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Map.Entry.html) key/value pairs. Because databases contain key/value pairs, any Berkeley DB database may be represented as a Java map.

The addSuppliers(), addParts() and addShipments() methods add objects to the Suppliers, Parts and Shipments stores. The [Map](http://download.oracle.com/javase/1.5.0/docs/api/java/util/Map.html) for each store is obtained from the SampleViews object.

The PrintDatabase.doWork() method calls printEntries() to print the map entries for each database store

A foreign key index is a secondary key index that also provides integrity constraints.

* A primary database may be associated with one or more secondary indices. A secondary index is always associated with exactly one primary database.
* For a secondary index, a [SecondaryKeyCreator](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/SecondaryKeyCreator.html) must be implemented by the application to extract the index key from the record of its associated primary database.
* A primary database is represented by a [Database](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/Database.html) object and a secondary index is represented by a [SecondaryDatabase](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/SecondaryDatabase.html) object. The [SecondaryDatabase](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/SecondaryDatabase.html) class extends the [Database](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/Database.html) class.
* When a [SecondaryDatabase](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/SecondaryDatabase.html) is created it is associated with a primary [Database](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/Database.html) object and a [SecondaryKeyCreator](https://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/SecondaryKeyCreator.html).