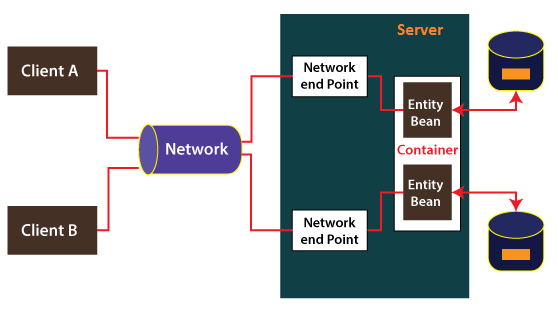
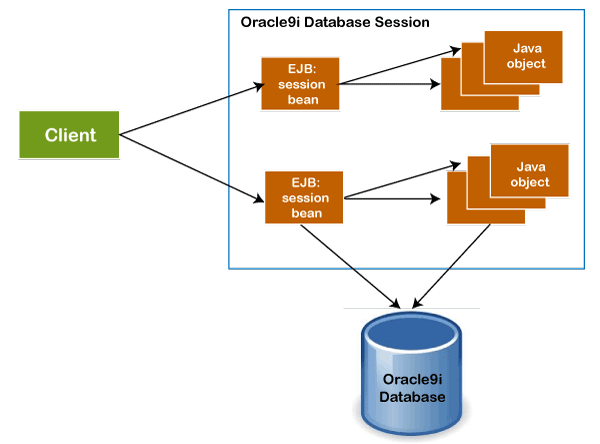
**Entity Bean**

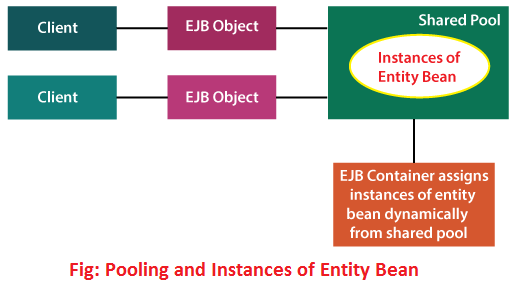
An entity bean is an unpredictable business substance. It models a business substance or models different activities inside a business interaction. It is used to encourage business benefits that include data and calculations on that data. It can deal with various, needy, diligent articles in playing out its essential assignments. A substance bean is a far-off object that oversees steady information and performs complex business rationale. It can be extraordinarily distinguished by an essential key.



It is a far-off object that oversees steady information, performs complex business rationale, possibly utilizes a few ward Java protests, and can be remarkably distinguished by an essential key. It ordinarily coarse-grained determined items since they use steady information put away inside a few fine-grained relentless [Java](https://www.javatpoint.com/java-tutorial) objects. Element beans are diligent on the grounds that they do endure a worker crash or an organization's disappointment.



Every entity bean has a persistence identity that is associated with it. It means that it comprises a unique identity that can be fetched if we have a primary key. The type for the unique key is defined by the bean provider. A client can retrieve the entity bean if the primary has been misplaced. If the bean is not available, the EJB container first, instantiates the bean and then re-populates the data for the client.



The diligence for entity bean information is given both to saving state when the bean is passivated and for improving the state when a failover has detected. It can endure in light of the fact that the information is put away determinedly by the holder in some type of information stockpiling framework, like a data set.

Entity beans persist business data by using the following two methods:

* Bean-Managed Persistence (BMP)
* Container-Managed Persistence (CMP)

Bean Managed Persistence

Bean-managed persistence is more complicated in comparison to container-managed persistence. Because it allows us to write the persistence logic into the bean class, explicitly. In order to write the persistence handling code into the bean class, we must know the type of database that is being used and how fields of the bean class are mapped to the database. Therefore, it provides more flexibility between the database and the bean instance.

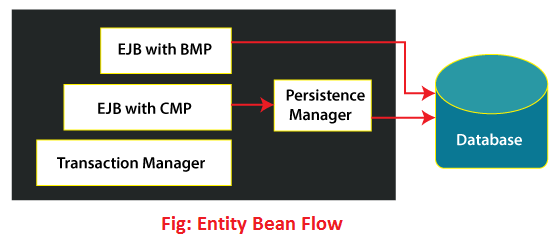
It can be used as an alternative to CMP. If the deployment tools are incompatible for mapping the bean instance's state to the database. The disadvantage of BMP is that more work is required to define the bean and it ties the bean to a specific database type and structure.

Container-Managed Persistence

In **container-managed persistence,** the EJB container transparently and implicitly handles the relationship between the bean and the database. Bean developers focus on the data and the business process. The principal constraint is that the EJB compartment can most likely not produce information base access articulations with the proficiency of a programmer.

Unlike BMP, CMP does not allow us to write database access calls in the methods of the entity bean class. It is because the persistence is handled by the container at run-time. The following two things are required to support the CMP:

* **Mapping:** It denotes that how to map an entity bean to a resource.
* **Runtime Environment:** A CMP runtime environment that uses the mapping information to perform persistence operations on each bean.



Note: Entity bean has been replaced with Java persistence API.