

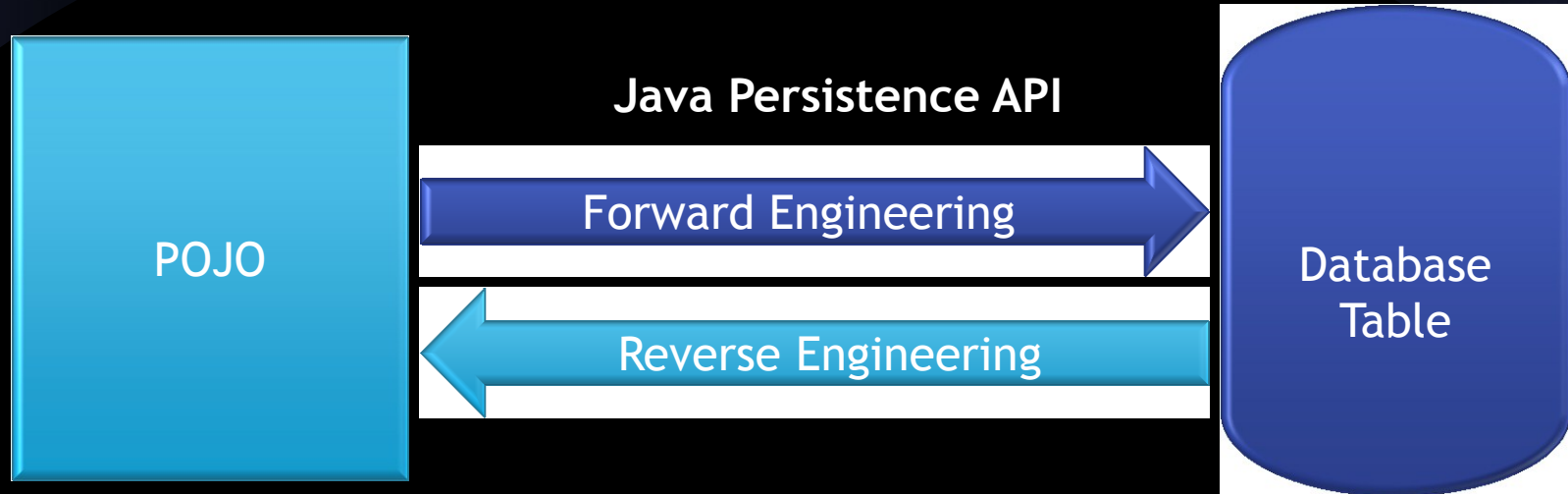
# \*Advanced Java Programming

Java Persistence API part.1

# Agenda

- \* O/R Mapping
- \* Primary entity annotations
- \* EntityManager
- \* Entity Relationships
- \* Practice

# O/R Mapping



# Primary Entity Annotations

- \* `@Entity`. Mark class as an entity
- \* `@Id`. Define primary key
- \* `@EmbeddedId`. Define composite key
- \* `@Table(name="TABLE_NAME")`. Define table name for entity class.
- \* `@Column`. Define column property.
- \* `@Transient`. Ignored by persistence framework.
- \* `@GeneratedValue`, `@SequenceGenerator`. Autopopulate column with sequence generator.

# EntityManager

- \* Maintains a cache of instances within a transactional context (persistence context)
- \* We can acquire EntityManager instance using :
  - Dependency Injection
  - EntityManagerFactory
  - JNDI Lookup
- \* Operations : `persist()`, `merge()`, `remove()`, `find()`, `createNamedQuery()`, `createQuery()`
- \* Persistence unit is declared in *persistence.xml*

# Entity Relationships

- \* @OneToOne. One to One is represented by a single-value entity reference at one or both ends of the relationship
- \* @OneToMany. This annotation is added to a *Collection* relationship field.
- \* @ManyToOne. Indicating that is an entity is part of a *Collection*
- \* @ManyToMany. This annotation is assigned to a *Collection* relationship field to indicate the target entity also has a *Collection* of the source entity type.
- \* Lazy vs Eager Binding
- \* Cascade (ALL, PERSIST, MERGE, REMOVE, REFRESH)

# Practice

- \* You can get entity beans by reverse engineering with Netbeans IDE.
- \* Demo in JPA1Demo.zip

# References

\*Beginning EJB 3 Application Development, Apress