JPA Mapping II

"Code with Passion!"

Agenda

- Entity Inheritance relationship
- Single table strategy
- Joined strategy
- Which one to use

Entity Inheritance

- Entities can have inheritance relationship
  - They are POJO's
- Three inheritance mapping strategies (mapping entity inheritance to database tables)
  - Single table
  - Joined subclass
  - Table per class (rarely used)
- Use annotation *@Inheritance(..)*

Single Table Strategy

Single Table Strategy

- All the classes in a hierarchy are mapped to a single table
- Annotation to the parent Entity
  - @Inheritance(strategy=InheritanceType.SINGLE_TABLE)
- Root table has a discriminator column whose value identifies the specific subclass to which the instance represented by row belongs
  - @DiscriminatorColumn(columnDefinition="MYDTYPE")

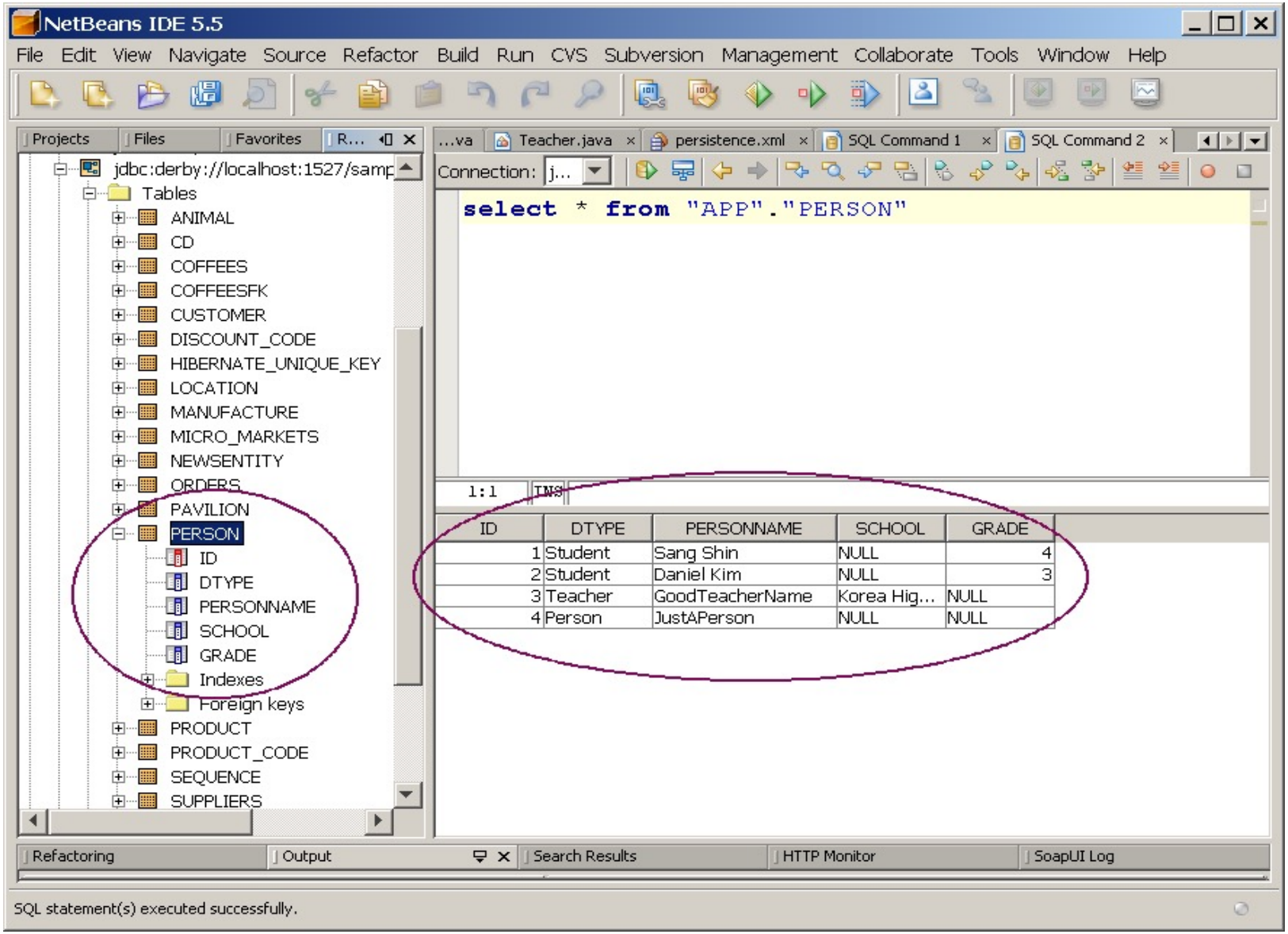
Single Table Strategy Example

```
// Parent Entity
@Entity
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(columnDefinition="MYDTYPE")
public class Person implements Serializable {...}

// Child Entity
@Entity
public class Student extends Person {...}

// Child Entity
@Entity
public class Teacher extends Person {...}
```

Single Table Strategy Example



Lab:

Exercise 1: Single Table Inheritance
4324_jpa_mapping2.zip

Joined Strategy

Joined Strategy

- One table for each class in the hierarchy
  - A parent class is represented by a single common table
  - Each child class is represented by a separate table that contains fields specific to the child class
  - Foreign key relationship exists between parent common table and subclass tables
- Annotation to the parent Entity
  - @Inheritance(strategy=InheritanceType.JOINED)

Joined Table Strategy Example

```
// Parent Entity
@Entity
@Inheritance(strategy=InheritanceType.JOINED)
@DiscriminatorColumn(columnDefinition="MYDTYPE")
public class Person implements Serializable {...}

// Child Entity
@Entity
public class Student extends Person {...}

// Child Entity
@Entity
public class Teacher extends Person {...}
```

Which One to Use?

So Which One Should You Use?

- SINGLE_TABLE or JOIN_TABLE?

SINGLE_TABLE

- Advantages
  - Offers best performance even for in the deep hierarchy since single select may suffice
- Disadvantages
  - Changes to members of the hierarchy require column to be altered, added or removed from the table

JOIN_TABLE

- Advantages
  - Does not require complex changes to the schema when a single parent class is modified
  - Works well with shallow hierarchy
- Disadvantages
  - Can result in poor performance – as hierarchy grows, the number of joins required to construct a leaf class also grows

Code with Passion!