```java
//RecipeService.java

package recipes.service;

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.*;
import recipes.exception.DbException;
import recipes.dao.*;
import recipes.entity.Recipe;

public class RecipeService {
    private static final String SCHEMA_FILE = "recipe_schema.sql";
    private static final String DATA_FILE = "recipe_data.sql";

    private RecipeDao recipeDao = new RecipeDao();

    public Recipe fetchRecipeById(Integer recipeId) {
        return recipeDao.fetchRecipeById(recipeId)
                .orElseThrow(() -> new NoSuchElementException("Recipe with ID=" + recipeId + "does not exist"));

    }

    public void createAndPopulateTables() {
        loadFromFile(SCHEMA_FILE);
        loadFromFile(DATA_FILE);
    }

    private void loadFromFile(String fileName) {
        String content = readFileContent(fileName);
        List<String> sqlStatements = convertContentToSqlStatements(content);

        recipeDao.executeBatch(sqlStatements);
    }

    private List<String> convertContentToSqlStatements(String content) {
        content = removeComments(content);
        content = replaceWhiteSpaceSequencesWithSingleSpace(content);

        return extractLinesFromContent(content);
    }

    private List<String> extractLinesFromContent(String content) {
        List<String> lines = new LinkedList<>();

        while (!content.isEmpty()) {
            int semicolon = content.indexOf(";");

            if (semicolon == -1) {
                if (!content.isBlank()) {
                    lines.add(content);
                }
                content = "";
            } else {
                lines.add(content.substring(0, semicolon).trim());
                content = content.substring(semicolon + 1);
            }
        }

        return lines;
    }

    private String replaceWhiteSpaceSequencesWithSingleSpace(String content) {
        return content.replaceAll("\\s+", " ");
    }

    private String removeComments(String content) {
        StringBuilder builder = new StringBuilder(content);
        int commentPos = 0;

        while ((commentPos = builder.indexOf("-- ", commentPos)) != -1) {
            int eolPos = builder.indexOf("\n", commentPos + 1);

            if (eolPos == -1) {
```

```
                builder.replace(commentPos, builder.length(), "");
            } else {
                builder.replace(commentPos, eolPos + 1, "");
            }
        }
        return builder.toString();
    }

    private String readFileContent(String fileName) {
        try {
            Path path = Paths.get(getClass().getClassLoader().getResource(fileName).toURI());
            return Files.readString(path);
        } catch (Exception e) {
            throw new DbException(e);
        }
    }

    public Recipe addRecipe(Recipe recipe) {
        return recipeDao.insertRecipe(recipe);

    }

    public List<Recipe> fetchRecipes() {
        return recipeDao.fetchAllRecipes();
    }

}



//Category.java

package recipes.entity;

public class Category {
    private Integer categoryId;
    private String categoryName;

    public String getCategoryName() {
        return categoryName;
    }
    public void setCategoryName(String categoryName) {
        this.categoryName = categoryName;
    }
    public Integer getCategoryId() {
        return categoryId;
    }
    public void setCategoryId(Integer categoryId) {
        this.categoryId = categoryId;
    }
    @Override
    public String toString() {
        return "ID=" + categoryId + ", categoryName=" + categoryName;
    }

}



//Ingredient.java

package recipes.entity;

import java.math.BigDecimal;
import java.util.Objects;

import provided.entity.EntityBase;

public class Ingredient extends EntityBase{
    private Integer ingredientId;
    private Integer recipe_id;
    private Unit unit;
    private String ingredientName;
    private String instruction;
    private Integer ingredientOrder;
```

```
    private BigDecimal amount;

    @Override
    public String toString() {
        StringBuilder b = new StringBuilder();

        b.append("ID =").append(ingredientId).append(": ");
        b.append(toFraction(amount));

        if(Objects.nonNull(unit) && Objects.nonNull(unit.getUnitId())) {
            String singular = unit.getUnitNameSingular();
            String plural = unit.getUnitNamePlural();
            String word = amount.compareTo(BigDecimal.ONE) > 0 ? plural : singular;

            b.append(word).append(" ");
        }
        b.append(ingredientName);
        if(Objects.nonNull(instruction)) {
            b.append(", ").append(instruction);
        }
        return b.toString();
    }
    public Integer getIngregredient_id() {
        return ingredientId;
    }
    public void setIngregredient_id(Integer ingregredient_id) {
        this.ingredientId = ingregredient_id;
    }
    public Integer getRecipe_id() {
        return recipe_id;
    }
    public void setRecipe_id(Integer recipe_id) {
        this.recipe_id = recipe_id;
    }
    public Unit getUnit() {
        return unit;
    }
    public void setUnit(Unit unit) {
        this.unit = unit;
    }
    public String getIngredientName() {
        return ingredientName;
    }
    public void setIngredientName(String ingredientName) {
        this.ingredientName = ingredientName;
    }
    public String getInstruction() {
        return instruction;
    }
    public void setInstruction(String instruction) {
        this.instruction = instruction;
    }
    public Integer getIngredientOrder() {
        return ingredientOrder;
    }
    public void setIngredientOrder(Integer ingredientOrder) {
        this.ingredientOrder = ingredientOrder;
    }
    public BigDecimal getAmount() {
        return amount;
    }
    public void setAmount(BigDecimal amount) {
        this.amount = amount;
    }

}



//Recipe.java

package recipes.entity;

import java.time.*;
import java.time.format.DateTimeFormatter;
```

```java
import java.util.*;

public class Recipe {
    private Integer recipeId;
    private String recipeName;
    private String notes;
    private Integer numServings;
    private LocalTime prepTime;
    private LocalTime cookTime;
    private LocalDateTime createdAt;

    private List<Ingredient> ingredients= new LinkedList<>();
    private List<Step> steps = new LinkedList<>();
    private List<Category> categories = new LinkedList<>();

    @Override
    public String toString() {
        DateTimeFormatter fmt = DateTimeFormatter.ofPattern("dd-MMM-yyyy HH:mm");
        String createTime = Objects.nonNull(createdAt) ? fmt.format(createdAt) : "(null)";
        String recipe = " ";

        recipe += "\n ID = " + recipeId;
        recipe += "\n Recipe Name = " + recipeName;
        recipe += "\n Notes = " + notes;
        recipe += "\n Number of Servings = " + numServings;
        recipe += "\n Prep Time = " + prepTime;
        recipe += "\n Cook Time = " + cookTime;
        recipe += "\n Created At = " + createTime;

        recipe += "\n Ingredients:";

        for (Ingredient ingredient : ingredients) {
            recipe += "\n       " + ingredient;
        }

        recipe += "\n   Steps";

        for (Step step : steps) {
            recipe += "\n       " + step;
        }

        recipe += "\n   Categories";

        for(Category category : categories) {
            recipe+= "\n        "+ category;
        }

        return recipe;
    }

    public Integer recipeId() {
        return recipeId;
    }
    public Integer getRecipeId() {
        return recipeId;
    }

    public void setRecipeId(Integer recipeId) {
        this.recipeId = recipeId;
    }

    public void recipeId(Integer recipe_id) {
        this.recipeId = recipe_id;
    }
    public String getRecipeName() {
        return recipeName;
    }
    public void setRecipeName(String recipeName) {
        this.recipeName = recipeName;
    }
    public String getNotes() {
        return notes;
    }
    public void setNotes(String notes) {
        this.notes = notes;
```

```
    }
    public Integer getNumServings() {
        return numServings;
    }
    public void setNumServings(Integer numServings) {
        this.numServings = numServings;
    }
    public LocalTime getPrepTime() {
        return prepTime;
    }
    public void setPrepTime(LocalTime prepTime) {
        this.prepTime = prepTime;
    }
    public LocalTime getCookTime() {
        return cookTime;
    }
    public void setCookTime(LocalTime cookTime) {
        this.cookTime = cookTime;
    }
    public LocalDateTime getCreatedAt() {
        return createdAt;
    }
    public void setCreatedAt(LocalDateTime createdAt) {
        this.createdAt = createdAt;
    }
    public List<Ingredient> getIngredients() {
        return ingredients;
    }

    public List<Step> getSteps() {
        return steps;
    }
    public List<Category> getCategories() {
        return categories;
    }


}



//Step.java

package recipes.entity;

public class Step {
    private Integer stepId;
    private Integer recipeId;
    private Integer stepOrder;
    private String stepText;

    @Override
    public String toString() {
        return "ID=" + stepId + ", stepText=" + stepText;
    }

    public Integer getStepId() {
        return stepId;
    }

    public void setStepId(Integer stepId) {
        this.stepId = stepId;
    }

    public Integer getRecipeId() {
        return recipeId;
    }

    public void setRecipeId(Integer recipeId) {
        this.recipeId = recipeId;
    }

    public Integer getStepOrder() {
        return stepOrder;
    }
```

```
    public void setStepOrder(Integer stepOrder) {
        this.stepOrder = stepOrder;
    }

    public String getStepText() {
        return stepText;
    }

    public void setStepText(String stepText) {
        this.stepText = stepText;
    }
}
```

```
//Unit.java

package recipes.entity;

public class Unit {
    public Integer getUnitId() {
        return unitId;
    }
    public void setUnitId(Integer unitId) {
        this.unitId = unitId;
    }
    public String getUnitNameSingular() {
        return unitNameSingular;
    }
    public void setUnitNameSingular(String unitNameSingular) {
        this.unitNameSingular = unitNameSingular;
    }
    public String getUnitNamePlural() {
        return unitNamePlural;
    }
    public void setUnitNamePlural(String unitNamePlural) {
        this.unitNamePlural = unitNamePlural;
    }
    private Integer unitId;
    private String unitNameSingular;
    private String unitNamePlural;
    @Override
    public String toString() {
        return "unit [unitId=" + unitId + ", unitNameSingular=" + unitNameSingular + ", unitNamePlural="
                + unitNamePlural + "]";
    }

}
```

```
//DbException
package recipes.exception;

@SuppressWarnings("serial")
public class DbException extends RuntimeException {

    public DbException() {
    }

    public DbException(String message) {
        super(message);
    }

    public DbException(Throwable cause) {
        super(cause);
    }

    public DbException(String message, Throwable cause) {
        super(message, cause);
    }

}
```

```
//REFERENCES
//YOUTUBE
//>>>>https://youtu.be/4-ZwFtUjFbc
//GITHUB
//>>>>https://github.com/fmd5045/Week07-11SQLRecipe/tree/main
```