Enzo BROUXEL

13/10/2023

SAE-S5-IRIT

Groupe 2

Compte-rendu d'activité SAE IRIT

Polyglot rewriting engine

1. Introduction

Le projet vise à créer une interface graphique pour un prototype d'interrogation de polystore, un système de gestion de données intégrant divers types de stockages. Le but est d'améliorer la visualisation des requêtes et d'étendre le prototype pour réécrire des sous-requêtes, simplifiant ainsi l'accès aux données stockées de manière hétérogène dans ces systèmes. Lien du github du projet : https://github.com/AurelienSP/SAE-S5-IRIT-G2.

2. Avancement du projet

Aujourd'hui, vendredi 13 octobre 2023, notre projet est très bien avancé par rapport à nos objectifs. Nous avons eu une entrevue avec le client ces derniers jours pour qu'il puisse vérifier la qualité de notre produit, cet entretien s'est bien déroulé et le client était satisfait de la version qui lui a été proposée.

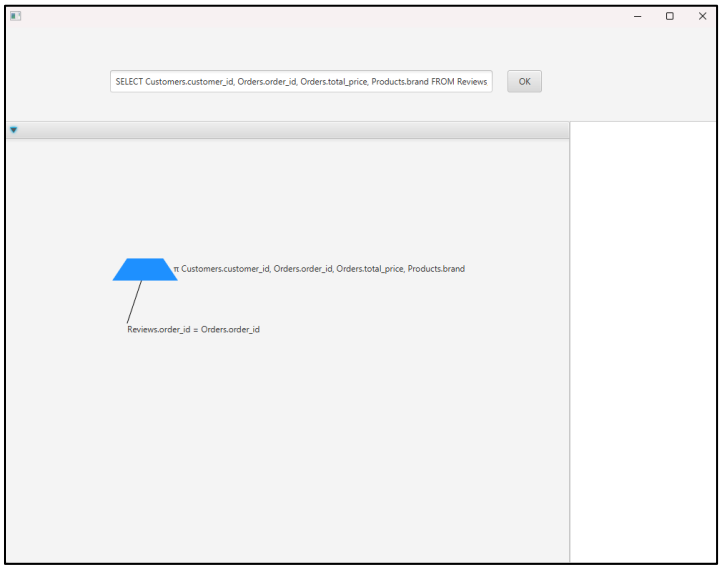

Figure 1 : Etat de l'application avant notre arrivé

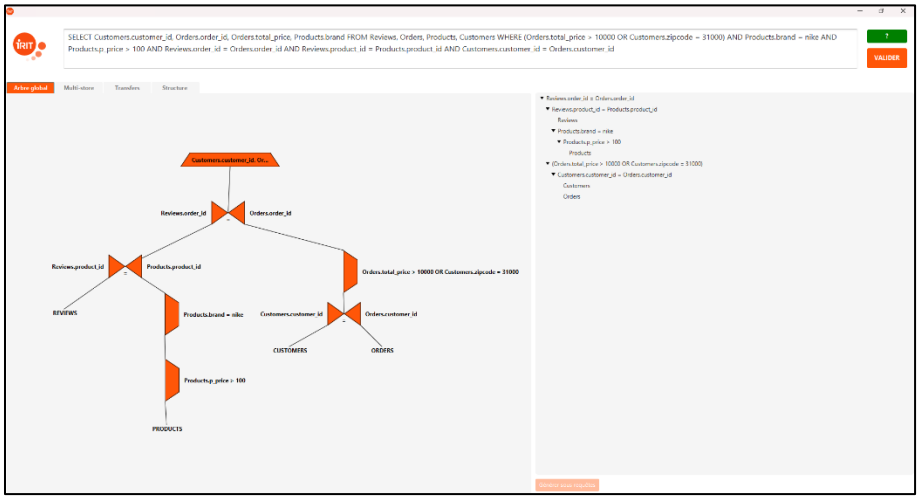

Figure 2 : Etat de l'application le 13/1

3. Mes issues

a. Amélioration du visuel des arbres algébriques

Ma première tâche dans le projet a été de retravailler complétement le visuel des arbres algébrique générés par notre application. Le but était de rendre ces arbres plus lisibles et compréhensibles par l'utilisateur. Pour arriver à ce résultat, il a fallu travailler sur deux points principaux, les nœuds et les liaisons entre ces nœuds.

Avant mon travail sur le design des arbres, les icônes des nœuds (ex : projection, sélection, jointure, etc.) étaient codées en dur dans le code java. J'ai donc transféré ce code dans des fichier FXML pouvant être utilisé avec SceneBuilder (un éditeur graphique). Cela rend donc l'aspect graphique des arbres plus facilement modifiable.

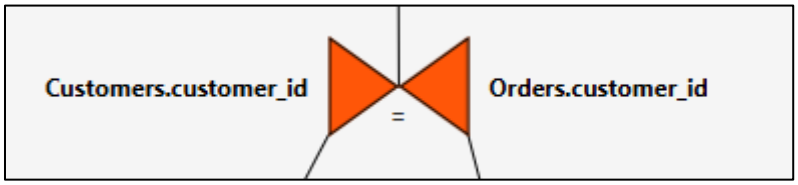

Figure 3 : Jointure résultat graphique

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.shape.*?>

<?import java.net.URL?>
<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/17.0.2-ea"
xmlns:fx="http://javafx.com/fxml/1">
   <Group fx:id="jointureGroup" layoutX="300.0" layoutY="170.0">
      <stylesheets>
         <URL value="@../../css/styles.css" />
      </stylesheets>
      <Polygon fx:id="jointurePolygon" styleClass="color-polygon" points="-
10.0, 0.0, -10.0, 50.0, 60.0, 0.0, 60.0, 50.0" />
      <Label fx:id="jointureEqualSign" layoutX="21.0" layoutY="30.0"
text="="/>
      <Label fx:id="jointureLeftLabel" alignment="CENTER_RIGHT"
contentDisplay="RIGHT" layoutX="-25.0" layoutY="17.0"
            text="left" textAlignment="RIGHT"/>
      <Label fx:id="jointureRightLabel" layoutX="67.0" layoutY="17.0"
text="right"/>
   </Group>
</AnchorPane>
```

Figure 4 : Fichier FXML pour l'icône de la jointure

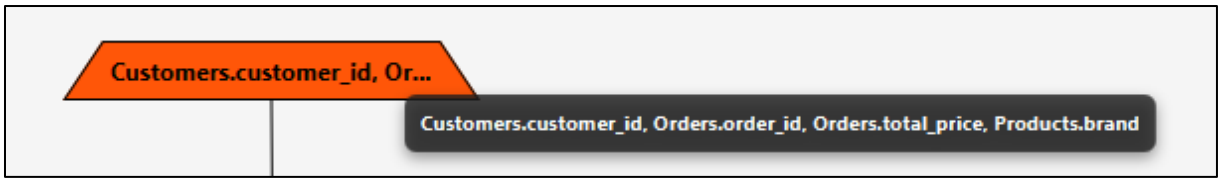

Figure 5 : Projection résultat graphique avec tooltip

Après avoir redesigné les icônes, je me suis penché les liaisons entre les nœuds. Avant la modification, les nœuds étaient mal reliés entre eux, les lignes se plaçaient aux coordonnées (0, 0) du nœud. À présent, les nœuds sont reliés proprement comme vu sur les figures ci-dessus.

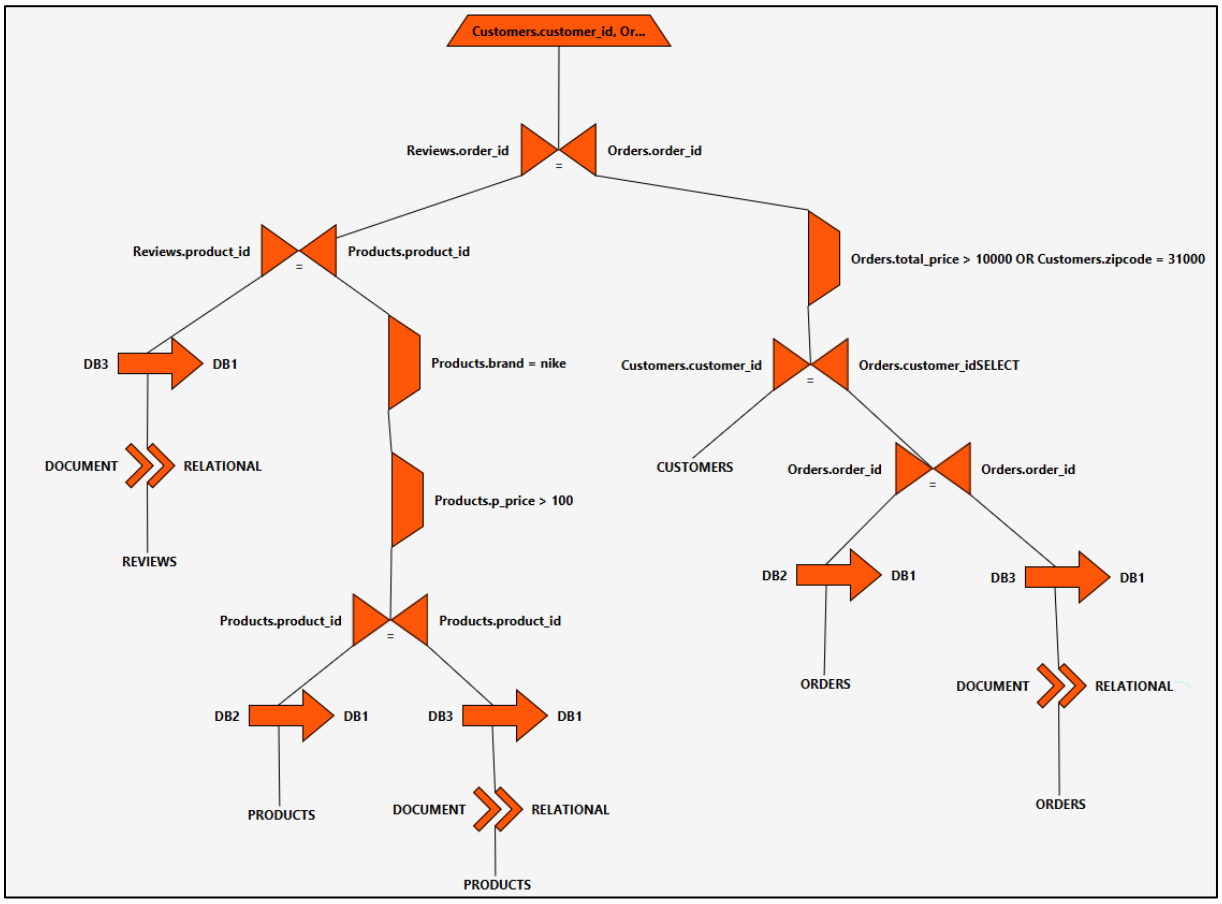

Figure 6 : Représentation complète d'un arbre

Commit associés à cette issue : 555a97c, 555a97c…6f02d6a, 6f02d6a…a949a3a, a949a3a…7b4b178 et 7b830d2…c70ee77

b. Améliorations IHM mineures

La deuxième issue à laquelle j'ai été affecté avait pour but d'améliorer le visuel global de l'application Java FX. Cette issue avait pour objectif plusieurs petites tâches (ex : ajout d'une fenêtre d'aide, ergonomie, etc.).

Lors de notre rencontre avec le client, celui-ci nous a demandé l'ajout d'une fenêtre d'aide pour mieux comprendre à quoi correspondes les icônes dans les arbres algébriques.

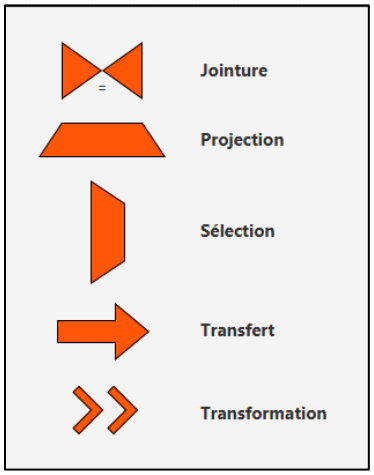

Figure 7 : Fenêtre d'aide

```
public void start(Stage primaryStage) throws Exception {
    Parent root =
FXMLLoader.load(Objects.requireNonNull(getClass().getResource("/fxml/helpPo
pup.fxml")));

    Scene scene = new Scene(root);

    Stage popupStage = new Stage();
    popupStage.initModality(Modality.WINDOW_MODAL);
    popupStage.setScene(scene);
    primaryStage.addEventFilter(MouseEvent.MOUSE_PRESSED, event -> {
        if
(!popupStage.getScene().getRoot().getLayoutBounds().contains(event.getScene
X(), event.getSceneY())) {
            popupStage.close();
        }
    });
    popupStage.setResizable(false);
    popupStage.getIcons().add(new
Image(String.valueOf(getClass().getResource("/img/help.png"))));
    popupStage.show();
}
```

Figure 8 : Code de la fenêtre d'aide

Le code ci-dessus, est celui qui permet la création d'une nouvelle fenêtre pour communiquer des informations d'aide à l'utilisateur de l'application. Le code est assez basique dans l'ensemble pour du Java FX, récupération du Template FXML, création d'un nouveau Stage et configuration de ce dernier. Cependant, j'ai ajouté à cela une fonctionnalité assez

intéressante qui est de pouvoir cliquer à l'extérieur de la fenêtre pour pouvoir la fermer (code entouré en rouge).

Commit associés à cette issue : c70ee77…2f70d04, 003e164…3d893c4, 3d893c4…70e25b2, 70e25b2…a1e470a, ffca5cd…ec0e798

4. Problèmes

Le principal problème que j'ai rencontré, a été de m'informer et de comprendre le projet à notre arrivé pour aider les initiaux. En effet, le projet de l'IRIT est assez complexe à première vue et le code de base n'était pas très compréhensible.

Il y a eu aussi à mon avis un manque de communications entre alternants et initiaux sur ce projet, un manque de motivation du côté de ces derniers était perceptible.

5. Conclusion

En conclusion, le projet d'interface graphique a considérablement avancé jusqu'à aujourd'hui.

Néanmoins, des défis liés à la complexité initiale du projet et à la communication au sein de l'équipe ont été rencontrés. Malgré cela, l'équipe a progressé de manière significative vers les objectifs du projet.