

UQAM – INF3080 – Automne 2024

TP # 3 : TRIGGERS ET APPLICATION JAVA
(G2BUqamInc)

A. IDENTIFICATION DES MEMBRES DE L'EQUIPE

- MARIUS GUIMATSIA AKALONG(GUIM27309006)
- AHADJI MBA CHRISTIAN (AHAC82320201)

I. Création des déclencheurs, procédures, et modification des tables

I.1 Configuration et installation de la base de données.

- Étape 1 : Création des tables du schéma physique de la base de données et modifications de la structure pour respecter les exigences du TP3.
Ouvrir une feuille SQL sous oracle et faire un copier/coller du script qui se trouve dans le fichier **02-Creation.sql**
- Étape 2 : Faire des insertions dans la base de données.
Ouvrir une nouvelle feuille SQL sous Oracle et faire un copier/coller du script qui se trouve dans le fichier **02-insert.sql**
- Étape 3 : Créer et tester les déclencheurs et les procédures.
Dans le fichier **01-trigger.sql** il se trouve les scripts de chaque déclencheur et procédure ainsi que son test.

Les lignes suivantes décrivent quelques tests et résultat des déclencheurs et procédures.

Question #1:

Q1_a) Création et test déclencheur T_validerbudget_Insertion_projetphase

- Création du déclencheur T_validerbudget_Insertion_projetphase

```
--Q1_a) Création du trigger T_validerbudget_Insertion_projetphase
create or replace TRIGGER Q1_a
AFTER INSERT ON Projet_phase
DECLARE
    total_budget_phase NUMBER(10, 2);
    budget_projet NUMBER(10, 2);
BEGIN
    -- Récupérer le budget total des phases existantes pour le projet concerné
    SELECT NVL(SUM(budget), 0) INTO total_budget_phase
    FROM Projet_phase
    WHERE id_projet = (SELECT id_projet FROM Projet_phase WHERE ROWID =
    (SELECT MAX(ROWID) FROM Projet_phase));

    -- Récupérer le budget du projet
    SELECT budget INTO budget_projet
    FROM Projet
    WHERE id_projet = (SELECT id_projet FROM Projet_phase WHERE ROWID =
    (SELECT MAX(ROWID) FROM Projet_phase));

    -- Vérifier que la somme des budgets ne dépasse pas le budget du projet
    IF (total_budget_phase > budget_projet) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Echec d''insertion : Le budget total des phases
        (' || total_budget_phase || ') dépasse le budget du projet (' || budget_projet || ').');
    END IF;
END;
```

- Test du déclencheur T_validerbudget_Insertion_projetphase

On tente l'insertion d'une phase qui fera déborder le budget

```
INSERT INTO Projet_phase VALUES (1, 4, 'Finalisation du projet', '2024-12-01', '2024-12-15', 500000.00,0);
```

Sortie de script x Résultat de requête x

Tâche terminée en 0,185 secondes

Erreur commençant à la ligne: 1 de la commande -
 INSERT INTO Projet_phase VALUES (1, 4, 'Finalisation du projet', '2024-12-01', '2024-12-15', 500000.00,0)
 Erreur à la ligne de commande: 1 Colonne: 84
 Rapport d'erreur -
 Erreur SQL : ORA-20001: Echec d'insertion : Le budget total des phases (5500000) dépasse le budget du projet (5000000).

Q1_b) Création et test déclencheur T_validerbudget_Modification_projetphase

- Q1_b Création du trigger T_validerbudget_Modification_projetphase

```
CREATE OR REPLACE TRIGGER Q1_b
AFTER UPDATE OF budget ON Projet_phase
DECLARE
    total_budget_phase NUMBER(10, 2);
    budget_projet NUMBER(10, 2);
BEGIN
    -- Parcourir chaque ligne de la table Projet_phase pour récupérer les projets

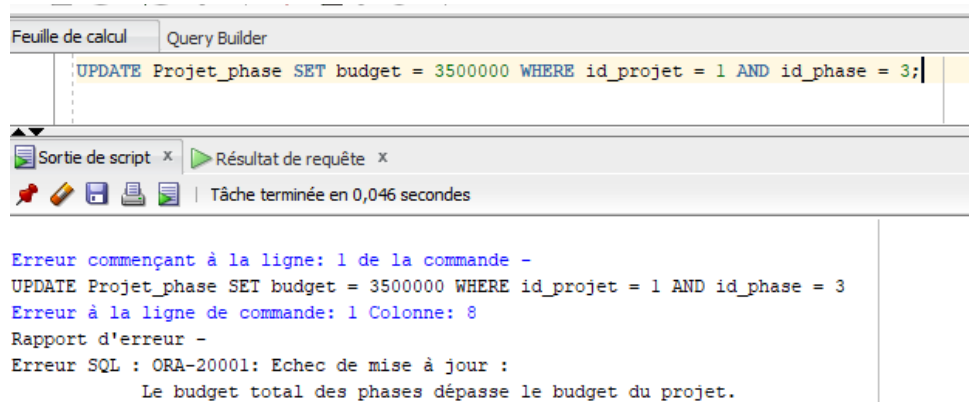
    FOR rec IN (SELECT id_projet
                FROM Projet_phase
                ) LOOP

        -- Pour chaque phase, récupérer le budget du projet
        SELECT budget INTO budget_projet
        FROM Projet
        WHERE id_projet = rec.id_projet;

        -- Calculer le budget total des phases (y compris les phases mises à jour)
        SELECT NVL(SUM(budget), 0)
        INTO total_budget_phase
        FROM Projet_phase
        WHERE id_projet = rec.id_projet;

        -- Vérifier que la somme des budgets des phases ne dépasse pas le budget du projet
        -- Ainsi seule la ligne modifiée sera rejetée
        IF total_budget_phase > budget_projet THEN
            RAISE_APPLICATION_ERROR(-20001, 'Echec de mise à jour :
            Le budget total des phases dépasse le budget du projet.');
```

- Test du déclencheur T_validerbudget_Modification_projetphase



Q2_a) Création et test déclencheur T_validerbudget_Insertion_projet

NOTE: Ce déclencheur ne sera jamais lever car le déclencheur Q1_a fait déjà les vérifications de conformité de budget

```
-- Q2_a) Création du trigger T_validerbudget_Insertion_projetphase
/*NOTE: Ce déclencheur ne sera jamais lever car le déclencheur Q1_a
Fait déjà les vérifications de conformité de budget
*/
CREATE OR REPLACE TRIGGER Q2_a
BEFORE INSERT ON Projet
FOR EACH ROW
DECLARE
    total_budget_phase Projet_phase.budget%TYPE;
BEGIN
    SELECT NVL(SUM(budget),0) INTO total_budget_phase FROM Projet_phase
    WHERE id_projet = :NEW.id_projet ;

    IF( total_budget_phase > :NEW.budget) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Le budget total des phases dépasse le budget du projet.');

```

Q2_b) Création et test déclencheur T_validerbudget_Modification_projet

NOTE: Ce déclencheur ne sera jamais lever car le déclencheur Q1_a fait déjà les vérifications de conformité de budget

```
-- Q2_a) Création du trigger T_validerbudget_Modification_projet
CREATE OR REPLACE TRIGGER T_validerbudget_Modification_projet
BEFORE UPDATE OF budget ON Projet
FOR EACH ROW
DECLARE
    total_budget_phase Projet.budget%TYPE;
BEGIN
    SELECT NVL(SUM(budget),0) INTO total_budget_phase FROM Projet
    WHERE id_projet = :NEW.id_projet;

    IF(total_budget_phase > :NEW.budget) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Echec de mis a jour : Le budget total des phases dépasse le budget du projet.');

```

Q3_a) Modifions la structure de la B.D pour ajouter le champ « dépenses » dans la table « projet_phase »

```
ALTER TABLE Projet_phase ADD depenses NUMBER(10, 2) DEFAULT 0 NOT NULL;
```

Q3_b) Création et test du déclencheur T_calcul_depensephase

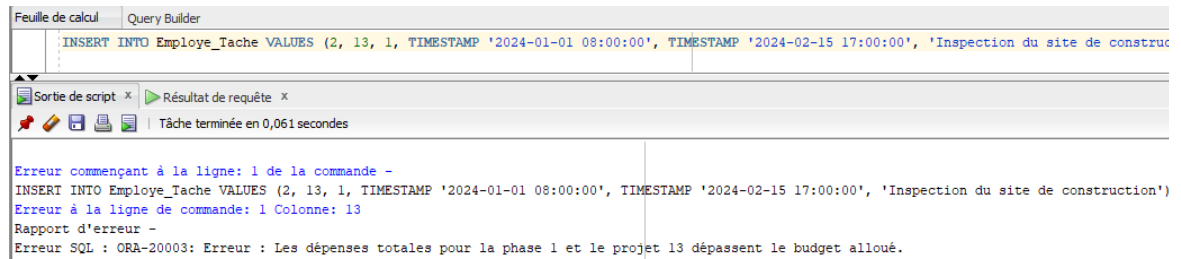
Note : Pour résoudre le problème des tables mutantes, nous allons utiliser un déclencheur au niveau déclaration (statement-level trigger) plutôt qu'un déclencheur au niveau ligne (row-level trigger).

```
CREATE OR REPLACE TRIGGER T_calcul_depensephase_statement
AFTER INSERT OR UPDATE OR DELETE ON Employe_tache
DECLARE
    total_depense_phase NUMBER(10, 2);
    CURSOR c_depenses IS
        SELECT et.id_phase, et.id_projet,
               NVL(SUM(e.salaire * (EXTRACT(HOUR FROM (et.date_heure_fin - et.date_heure_debut)))), 0) AS total_depense
        FROM Employe_tache et
        INNER JOIN Employe e ON et.id_employe = e.id_employe
        GROUP BY et.id_phase, et.id_projet;
BEGIN
    -- Parcours du curseur pour calculer les dépenses totales par phase et projet
    FOR rec IN c_depenses LOOP
        -- Vérification du budget de la phase et du projet
        DECLARE
            budget_phase NUMBER(10, 2);
        BEGIN
            -- Récupérer le budget de la phase
            SELECT budget
            INTO budget_phase
            FROM Projet_phase
            WHERE id_phase = rec.id_phase AND id_projet = rec.id_projet;

            -- Si les dépenses dépassent le budget alors on lève une exception
            IF rec.total_depense > budget_phase THEN
                RAISE_APPLICATION_ERROR(
                    -20003,
                    'Erreur : Les dépenses totales pour la phase ' || rec.id_phase || ' et le projet ' || rec.id_projet ||
                    ' dépassent le budget alloué.'
                );
            END IF;
        END;

        -- Sinon on procède à la mise à jour des dépenses dans la table Projet_phase
        UPDATE Projet_phase
        SET depenses = rec.total_depense
        WHERE id_phase = rec.id_phase AND id_projet = rec.id_projet;
    END LOOP;
END;
```

- Test du déclencheur T_calcul_depensephase



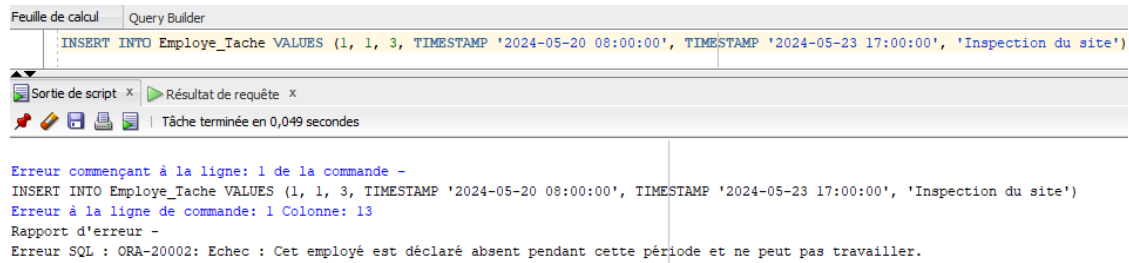
The screenshot shows a SQL Query Builder window with a query: `INSERT INTO Employe_Tache VALUES (2, 13, 1, TIMESTAMP '2024-01-01 08:00:00', TIMESTAMP '2024-02-15 17:00:00', 'Inspection du site de construc`. Below the query, it indicates the task was completed in 0.061 seconds. The error message states:
 Erreur commençant à la ligne: 1 de la commande -
 INSERT INTO Employe_Tache VALUES (2, 13, 1, TIMESTAMP '2024-01-01 08:00:00', TIMESTAMP '2024-02-15 17:00:00', 'Inspection du site de construction')
 Erreur à la ligne de commande: 1 Colonne: 13
 Rapport d'erreur -
 Erreur SQL : ORA-20003: Erreur : Les dépenses totales pour la phase 1 et le projet 13 dépassent le budget alloué.

Q4) Création d'un déclencheur T_valider_tache permettant de vérifier qu'un employé ne peut pas travailler s'il était absent.

```
CREATE OR REPLACE TRIGGER T_valider_tache
BEFORE INSERT OR UPDATE ON Employe_Tache
FOR EACH ROW
DECLARE
    nb_absences NUMBER;
BEGIN
    -- Vérifier si l'employé est absent pendant la période de la tâche
    SELECT COUNT(*) INTO nb_absences
    FROM Absence_employe ae
    WHERE ae.id_employe = :NEW.id_employe
    -- Un conflit se produit lorsque les intervalles d'absence
    -- et de presence au travail se chevauchent ou coïncident.
    AND (
        -- Vérifier si la tâche est comprise dans une période d'absence
        (:NEW.date_heure_debut BETWEEN ae.date_debut_absence AND ae.date_fin_absence)
        OR
        (:NEW.date_heure_fin BETWEEN ae.date_debut_absence AND ae.date_fin_absence)
        OR
        -- Vérifier si la période d'absence englobe complètement la tâche
        (ae.date_debut_absence <= :NEW.date_heure_debut
        AND ae.date_fin_absence >= :NEW.date_heure_fin)
    );

    -- Si nb_absences > 0, lever une exception
    IF nb_absences > 0 THEN
        RAISE_APPLICATION_ERROR(
            -20002,
            'Echec : Cet employé est déclaré absent pendant cette période et ne peut pas travailler.'
        );
    END IF;
END;
```

- Test du déclencheur T_valider_tache



The screenshot shows a SQL query execution window. The query is: `INSERT INTO Employe_Tache VALUES (1, 1, 3, TIMESTAMP '2024-05-20 08:00:00', TIMESTAMP '2024-05-23 17:00:00', 'Inspection du site')`. The execution failed with the following error message:
 Erreur commençant à la ligne: 1 de la commande -
 INSERT INTO Employe_Tache VALUES (1, 1, 3, TIMESTAMP '2024-05-20 08:00:00', TIMESTAMP '2024-05-23 17:00:00', 'Inspection du site')
 Erreur à la ligne de commande: 1 Colonne: 13
 Rapport d'erreur -
 Erreur SQL : ORA-20002: Echec : Cet employé est déclaré absent pendant cette période et ne peut pas travailler.

Q5) Modifions la structure de la B.D pour refléter les règles d'affaires suivantes

Q5_a) Identifier le ou les directeurs d'un employé dans la compagnie

```
ALTER TABLE Employe ADD id_directeur NUMBER(10);

ALTER TABLE Employe
ADD CONSTRAINT fk_employe_directeur FOREIGN KEY (id_directeur) REFERENCES Employe(id_employe);
```

Q5_b) Garder un historique des paiements des employés

```
CREATE TABLE Historique_Paiement (
  id_paiement NUMBER(10) NOT NULL,
  id_employe NUMBER(10) NOT NULL,
  date_paiement DATE NOT NULL,
  montant NUMBER(10, 2) NOT NULL,
  mode_paiement VARCHAR2(20) NOT NULL,
  CONSTRAINT pk_historique_paiement PRIMARY KEY (id_paiement),
  CONSTRAINT fk_historique_paiement_employe FOREIGN KEY (id_employe) REFERENCES Employe(id_employe)
);
```

- Procédure permettant de générer l'historique de paiement de chaque employé

```
-- Création d'un itérateur
CREATE SEQUENCE HISTORIQUE_PAIEMENT_SEQ START WITH 1 INCREMENT BY 1;

-- Procédure permettant de remplir la table historique de paiement generer_historique_paiement
CREATE OR REPLACE PROCEDURE generer_historique_paiement
IS
    CURSOR taches_employes IS
        SELECT et.id_employe, e.salaire, et.date_heure_fin,
            (EXTRACT(HOUR FROM (et.date_heure_fin - et.date_heure_debut)) * e.salaire) AS montant
        FROM Employe_tache et
        INNER JOIN Employe e ON et.id_employe = e.id_employe;

    -- Variables pour stocker les données du curseur
    v_id_employe Employe_Tache.id_employe%TYPE;
    v_date_paiement Employe_Tache.date_heure_fin%TYPE;
    v_salaire Employe.salaire%TYPE;
    v_montant NUMBER(10, 2);
BEGIN
    FOR rec IN taches_employes LOOP
        -- Assigner les valeurs récupérées
        v_id_employe := rec.id_employe;
        v_date_paiement := rec.date_heure_fin;
        v_montant := rec.montant;

        INSERT INTO Historique_Paiement
            (id_paiement, id_employe, date_paiement, montant, mode_paiement)

        VALUES (
            HISTORIQUE_PAIEMENT_SEQ.NEXTVAL, -- Utilisation d'une séquence pour générer l'ID
            v_id_employe,
            TRUNC(v_date_paiement), -- Date du paiement basée sur la fin de la tâche
            v_montant,
            'Virement Bancaire' -- Mode de paiement par défaut
        );
    END LOOP;

    COMMIT;
END;

-- Exécution de la procédure generer_historique_paiement
BEGIN
    generer_historique_paiement;
END;
```


Q6) Créer une procédure (P-afficher-vip) permettant d'afficher le nom et prénom des employés qui ont encaissé plus que 30000\$ entre le 1er et le 30 novembre 2024

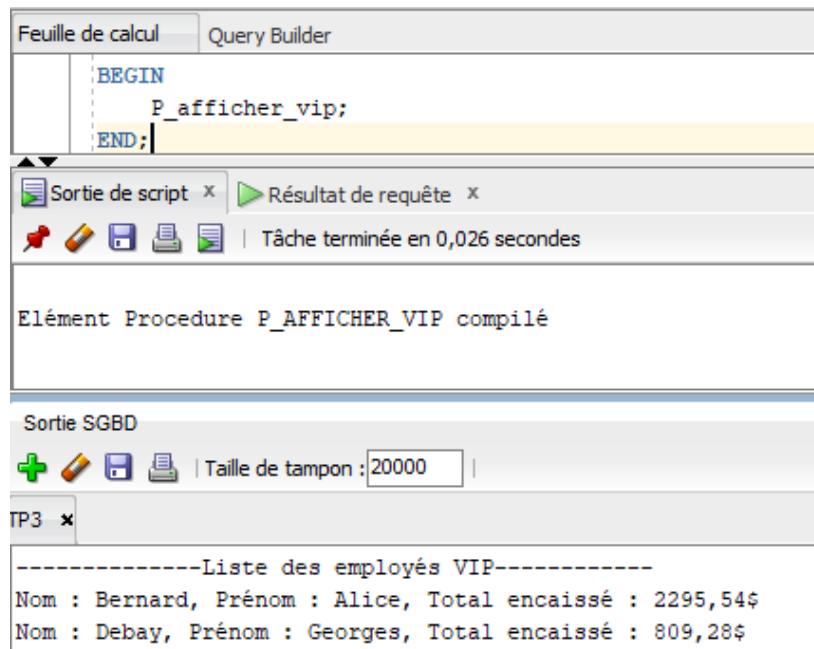
Attention !!! Pour des raisons de tests vu que les montants des paiements des employés sont faibles la valeur 30000\$ sera modifié a 300 \$

```
CREATE OR REPLACE PROCEDURE P_afficher_vip
IS
    -- Déclare un curseur pour récupérer les employés VIP
    CURSOR liste_employes_vip IS
        SELECT e.nom, e.prenom, SUM(hp.montant) AS total_encaisse
        FROM Employe e
        INNER JOIN Historique_Paiement hp ON e.id_employe = hp.id_employe
        WHERE hp.date_paiement BETWEEN TO_DATE('2024-11-01', 'YYYY-MM-DD')
        AND TO_DATE('2024-11-30', 'YYYY-MM-DD')
        GROUP BY e.nom, e.prenom
        HAVING SUM(hp.montant) > 30000;

    -- Déclare une variable pour compter le nombre d'enregistrements retournés
    nb_employes NUMBER := 0;
BEGIN
    -- Boucle pour parcourir les enregistrements du curseur
    DBMS_OUTPUT.PUT_LINE('-----Liste des employés VIP----- ');
    FOR rec IN liste_employes_vip LOOP
        DBMS_OUTPUT.PUT_LINE('Nom : ' || rec.nom || ', Prénom : ' || rec.prenom || ', Total encaissé : ' || rec.total_encaisse || '$');
        nb_employes := nb_employes + 1;
    END LOOP;

    -- Vérifie si aucun employé VIP n'a été trouvé
    IF nb_employes = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Aucun client VIP trouvé. ');
    END IF;
END;
```

- Test de la procédure P-afficher-vip



Feuille de calcul Query Builder

```
BEGIN
    P_afficher_vip;
END;
```

Sortie de script x Résultat de requête x

Tâche terminée en 0,026 secondes

Elément Procedure P_AFFICHER_VIP compilé

Sortie SGBD

TP3 x

```
-----Liste des employés VIP-----
Nom : Bernard, Prénom : Alice, Total encaissé : 2295,54$
Nom : Debay, Prénom : Georges, Total encaissé : 809,28$
```

II. Créer une application Java

Pour installer et exécuter l'application java (AppJava)

- Ouvrir le dossier du projet avec IntelliJ IDEA
- Lancer l'exécution du programme depuis IntelliJ IDEA
- Dans le fichier AppJava se trouve le code source de l'application.
Il faudra définir un nom utilisateur et mot de passe correspondant à la base de données que vous avez créée plus haut.

Les paramètres sont :

DB_URL = "url_de_connexion_bd"

DB_USERNAME = "username_de_la_bd";

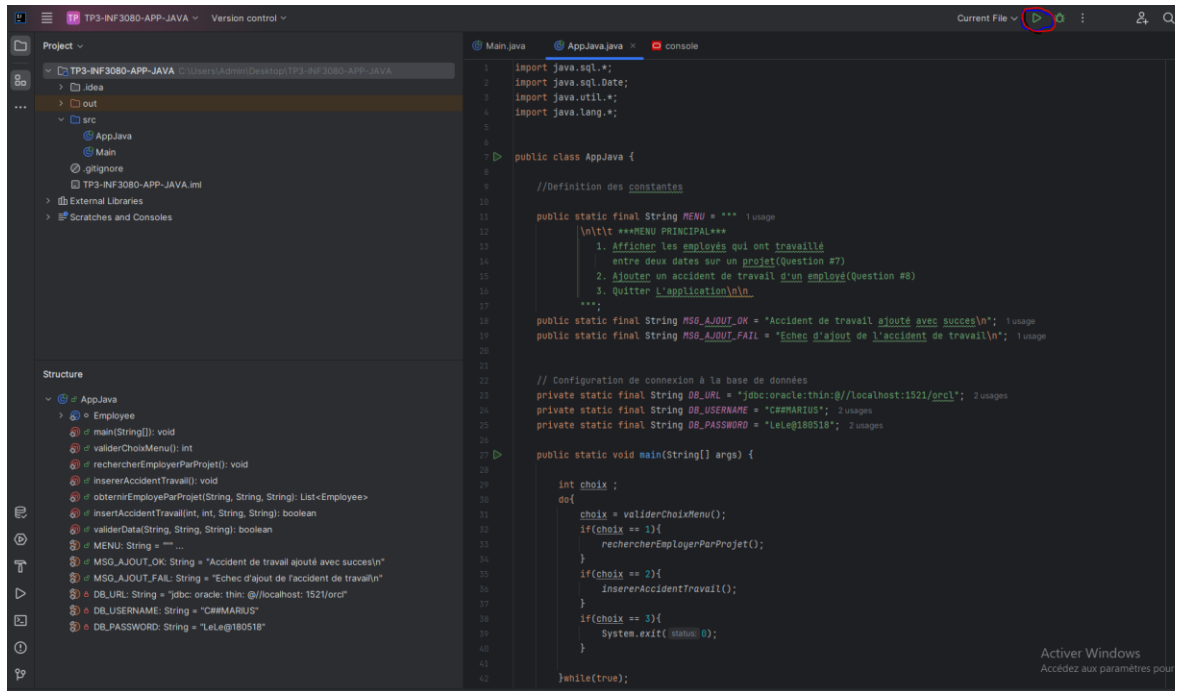
DB_PASSWORD = "votre_mot_de_passe"

Une fois les paramètres de connexion saisis vous serez connecter a la base de données et pourrez désormais utiliser l'application.

- Un menu principal s'affichera présentant les fonctionnalités de l'application
- Choisir celle que vous voulez tester et suivez les étapes. (voir capture ci-dessous)

Q7) Interroger votre base de données afin d'afficher le nom et prénom des employés qui ont travaillé entre deux dates sur un projet (infos en paramètres)

- Lancer l'exécution de l'application



- Faire un choix de menu pour la Question #7 choisir 1

```
***MENU PRINCIPAL***
1. Afficher les employés qui ont travaillé
   entre deux dates sur un projet(Question #7)
2. Ajouter un accident de travail d'un employé(Question #8)
3. Obtenir la liste des conflits horaire (Question #9)
4. Quitter L'application

Entrez votre choix :1
```

- Entrez les informations sur le projet

Par exemple

Entrez le nom du projet : Réhabilitation Route XYZ

Entrez la date de début (YYYY-MM-DD) : 2024-03-01

Entrez la date de fin (YYYY-MM-DD) : 2024-10-01

```
Entrez votre choix :1
Entrez le nom du projet : Réhabilitation Route XYZ
Entrez la date de début (YYYY-MM-DD) : 2024-03-01
Entrez la date de fin (YYYY-MM-DD) : 2024-10-01

Liste des employés ayant travaillé sur le projet : Réhabilitation Route XYZ
entre 2024-03-01 et 2024-10-01
- Nom : Bernard, Prénom : Alice

- Nom : Durand, Prénom : Paul
```

Q8) Créer un formulaire permettant d'ajouter un accident de travail d'un employé

- Faire un choix de menu pour la Question #8 choisir 2

Puis saisir les informations nécessaires pour ajouter un accident de travail

Exemple

Entrez votre choix : 2

Saisir l'identifiant du projet(id_projet) : 1

Saisir l'identifiant de l'employe(id_employe) : 7

Saisir la date de l'accident(YYYY-MM-DD) : 2024-11-25

Saisir la description de l'accident : Torsion du genou en sortant d'un tunnel

```
***MENU PRINCIPAL***
1. Afficher les employés qui ont travaillé
   entre deux dates sur un projet(Question #7)
2. Ajouter un accident de travail d'un employé(Question #8)
3. Obtenir la liste des conflits horaire (Question #9)
4. Quitter L'application

Entrez votre choix :2
Saisir l'identifiant du projet(id_projet) : 1
Saisir l'identifiant de l'employe(id_employe) : 7
Saisir la date de l'accident(YYYY-MM-DD) : 2024-11-25
Saisir la description de l'accident : Torsion du genou en sortant d'un tunnel
```

Q9) Créer un formulaire de votre choix impliquant au moins 3 tables

- Créer une fonctionnalité permettant d'afficher les noms et prénoms des employés qui ont déclaré des heures de travail malgré qu'ils soient absent.

Pour obtenir la liste des employés ayant déclaré des heures de travail alors qu'ils étaient absents il faut choisir l'option 3.

NB : Ce conflit existe parce que le déclencheur valider_tache(T_valider_tache) n'était pas encore implémenté.

```
***MENU PRINCIPAL***
1. Afficher les employés qui ont travaillé
   entre deux dates sur un projet(Question #7)
2. Ajouter un accident de travail d'un employé(Question #8)
3. Obtenir la liste des conflits présence (Question #9)
4. Quitter L'application

Entrez votre choix :3

Liste des employés ayant déclaré etre entrain de travailler alors qu'ils étaient absents
- Nom : Tremblay, Prénom : Natasha
```