



Mini-projet PL/SQL

Réalisé par : @Sarra OTHMANI

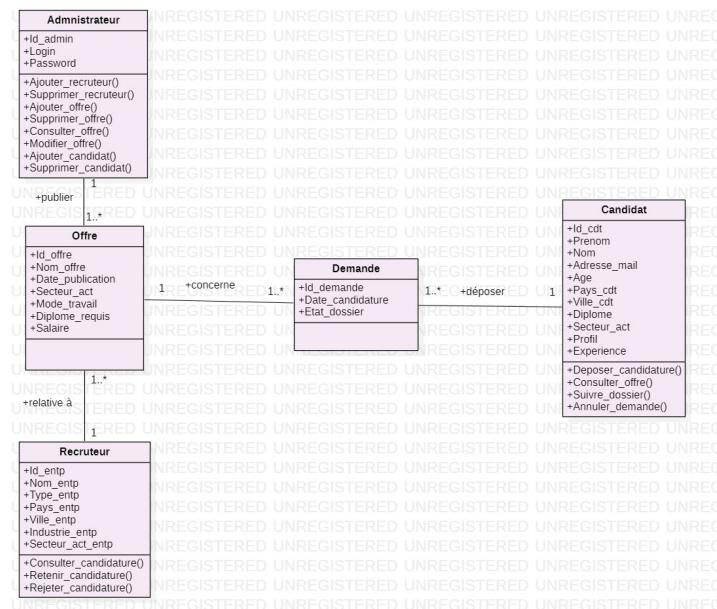
Présentation générale :

La mise en place d'un système de base de données relationnelles d'une application de mise en contact de demandes et d'offres d'emploi.

Le but de cette application est de mettre en contact des CVs de demandeurs d'emploi et les offres d'entreprises. À tout moment, un demandeur d'emploi peut voir toutes les offres en corrélation avec son CV, et un recruteur, tous les CV en corrélation avec son offre.

Diagramme de classe

Outil utilisé : **StarUML**



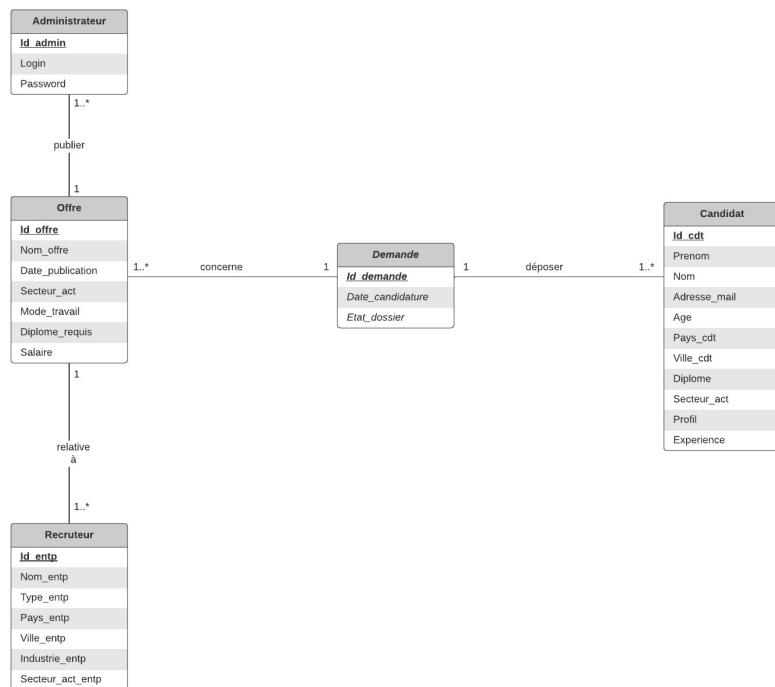
Les cardinalités

- Un candidat peut postuler à **un ou plusieurs** offres d'emploi, c'est à dire, il peut déposer **une ou plusieurs** demandes de candidatures, alors qu'une demande ne peut concerner qu'**un seul** candidat.
- Une demande de candidature ne concerne qu'**une seule** offre d'emploi, alors qu'une unique offre peut avoir **un ou plusieurs** demandes de candidature.
- L'administrateur peut publier et gérer **un ou plusieurs** offres d'emploi, alors qu'une seule offre d'emploi ne peut être gérée que par **un seul** unique administrateur de l'application de recrutement.

- Un recruteur de l'entreprise représente **une ou plusieurs** offres d'emploi, alors qu'une offre n'est représentée que par **un et un seul** recruteur.
- Un recruteur peut traiter **une ou plusieurs** demandes de candidatures, alors que la demande ne peut être traitée que par **un seul** recruteur en question.

Modèle Conceptuel de données (MCD)

Outil utilisé : *Lucid.app*



Du schéma conceptuel au schéma relationnel (MLD)

Candidat ([Id_cdt](#), Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience, #[Id_demande](#))

Administrateur ([Id_admin](#), Login, Password, #[Id_offre](#))

Recruteur ([Id_entp](#), Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp, #[Id_Offre](#))

Offre ([Id_offre](#), Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire, #[Id_demande](#))

Demande ([Id_demande](#), Date_candidature, Etat_dossier)

Code source :

```

/* Creating a new user and granting privileges */

alter session set "_ORACLE_SCRIPT"=true;
CREATE USER user101 IDENTIFIED BY 101;
GRANT CONNECT, RESOURCE, DBA TO user101;

```

```
/* BASE DE DONNEES */
```

```
/* Drop tables */
```

```
DROP TABLE Candidat;
DROP TABLE Administrateur;
```

```

DROP TABLE Recruteur;
DROP TABLE Offre;
DROP TABLE Demande;

/* Create tables */

/* Table Candidat */

CREATE TABLE Candidat(
Id_cdt CHAR(3),
Prenom VARCHAR2(30),
Nom VARCHAR2(30),
Adresse_mail VARCHAR2(30),
Age NUMBER(2),
Pays_cdt VARCHAR2(30),
Ville_cdt VARCHAR2(30),
Diplome VARCHAR2(30),
Secteur_act_cdt VARCHAR2(30),
Profil VARCHAR2(30),
Experience Number(2),
Id_demande CHAR(3),
CONSTRAINT Pk_cdt PRIMARY KEY(Id_cdt));

/* Table Administrateur */

CREATE TABLE Administrateur(
Id_admin CHAR(3),
Login VARCHAR2(30),
Password VARCHAR2(30),
Id_offre CHAR(3),
CONSTRAINT Pk_admin PRIMARY KEY(Id_admin));

/* Table Recruteur */

CREATE TABLE Recruteur(
Id_entp CHAR(3),
Nom_entp VARCHAR2(30),
Type_entp VARCHAR2(30),
Pays_entp VARCHAR2(30),
Ville_entp VARCHAR2(30),
Industrie_entp VARCHAR2(30),
Secteur_act_entp VARCHAR2(30),
Id_offre CHAR(3),
CONSTRAINT Pk_entp PRIMARY KEY(Id_entp));

/* Table Offre */

CREATE TABLE Offre(
Id_offre CHAR(3),
Nom_offre VARCHAR2(30),
Date_publication DATE,
Secteur_act VARCHAR2(30),
Mode_travail VARCHAR2(30),
Diplome_requis VARCHAR2(30),
Salaire NUMBER(8,3),
Id_demande CHAR(3),
CONSTRAINT Pk_offre PRIMARY KEY(Id_offre));

/* Table Demande */

CREATE TABLE Demande(
Id_demande CHAR(3),
Date_candidature DATE,
Etat_dossier VARCHAR2(30),
CONSTRAINT Pk_demande PRIMARY KEY(Id_demande));

/* Add Relational constraints */

/* ADD relational constraints (foreign keys) */
ALTER TABLE Candidat
ADD CONSTRAINT Fk_cdt FOREIGN KEY (Id_demande) REFERENCES Demande (Id_demande);

ALTER TABLE Administrateur
ADD CONSTRAINT Fk_admin FOREIGN KEY (Id_offre) REFERENCES Offre(Id_offre);

ALTER TABLE Recruteur
ADD CONSTRAINT Fk_entp FOREIGN KEY (Id_offre) REFERENCES Offre(Id_offre);

ALTER TABLE Offre
ADD CONSTRAINT Fk_offre FOREIGN KEY (Id_demande) REFERENCES Demande (Id_demande);

/* Add records */

/* INSERT INTO Candidat */

```

```

INSERT INTO Candidat (Id_cdt, Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience)
VALUES ('1', 'Carol', 'Anderson', 'Carol.anderson@outlook.com', 23, 'United States', 'San Francisco', 'Bachelor', 'Dentistry', 'Dentis
INSERT INTO Candidat (Id_cdt, Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience)
VALUES ('2', 'Linda', 'Rahhal', 'Linda.Rahhal@gmail.com', 25, 'Lebanon', 'Beyrouth', 'Masters', 'Finance and insurance', 'Finance and
INSERT INTO Candidat (Id_cdt, Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience)
VALUES ('14', 'Zeineb', 'Essefi', 'Zeineb.essefi98@live.fr', 25, 'Tunisia', 'Tunis', 'Bachelor', 'Advertising and marketing', 'Digital
INSERT INTO Candidat (Id_cdt, Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience)
VALUES ('15', 'Giovanna', 'Terracciano', 'Giovanna.terracciano@gmail.com', 26, 'Italy', 'Naples', 'Bachelor', 'Consulting', 'HR consul
INSERT INTO Candidat (Id_cdt, Prenom, Nom, Adresse_email, Age, Pays_cdt, Ville_cdt, Diplome, Secteur_act_cdt, Profil, Experience)
VALUES ('20', 'Antoine', 'Dupont', 'Antoine_dupont566@gmail.com', 28, 'France', 'Paris', 'Masters', 'Information Systems', 'Data Analy

/* INSERT INTO Administrateur */
INSERT INTO Administrateur (Id_admin, Login, Password)
VALUES ('3', 'ADMIN101', 'ADMIN0000');

/* INSERT INTO Recruteur */
INSERT INTO Recruteur (Id_entp, Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp)
VALUES ('1', 'Coca Cola', 'Public', 'United States', 'Atlanta, Georgia', 'Manufacturing', 'Food manufacturing');
INSERT INTO Recruteur (Id_entp, Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp)
VALUES ('2', 'McKinsey and Company', 'Public', 'France', 'Paris', 'Consulting', 'Business consultancy');
INSERT INTO Recruteur (Id_entp, Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp)
VALUES ('9', 'Tri Creek Family Industry', 'Private', 'United States', 'Indiana', 'Healthcare', 'Food manufacturing');
INSERT INTO Recruteur (Id_entp, Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp)
VALUES ('17', 'The Walt Disney Company', 'Private', 'Spain', 'Madrid', 'Entertainment and Media', 'Media production');
INSERT INTO Recruteur (Id_entp, Nom_entp, Type_entp, Pays_entp, Ville_entp, Industrie_entp, Secteur_act_entp)
VALUES ('18', 'Al GHAZI BUSINESS SERVICES', 'Private', 'United Arab Emirates', 'Abu Dhabi', 'Consulting', 'Business services');

/* INSERT INTO Offre */
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ('1', 'Business Intelligence manager', '01/12/2022', 'Information Systems', 'Full time', 'Bachelor', 108.397);
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ('102', 'Data scientist', '02/11/2022', 'Information Systems', 'Full time', 'Masters', 228.355);
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ('9', 'Dental Assistant', '19/11/2022', 'Dentistry', 'Part time', 'Masters', 401.000);
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ('17', 'Business Intelligence manager', '02/12/2022', 'Payroll compliance management', 'Full time', 'Bachelor', 125.900);
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ('18', 'Manager HR', '30/11/2022', 'Consulting', 'Full time', 'Bachelor', 320.250);

/* INSERT INTO Demande */
INSERT INTO Demande (Id_demande, Date_candidature, Etat_dossier)
VALUES ('25', '08/12/2022', 'en cours de traitement');
INSERT INTO Demande (Id_demande, Date_candidature, Etat_dossier)
VALUES ('34', '30/11/2022', 'Retenu');
INSERT INTO Demande (Id_demande, Date_candidature, Etat_dossier)
VALUES ('88', '06/12/2022', 'Retenu');
INSERT INTO Demande (Id_demande, Date_candidature, Etat_dossier)
VALUES ('91', '03/12/2022', 'Rejeté');
INSERT INTO Demande (Id_demande, Date_candidature, Etat_dossier)
VALUES ('87', '30/11/2022', 'Retenu');

COMMIT;

```

```

/* REQUETES PL/SQL BASIQUES */

SET SERVEROUTPUT ON;
SELECT * FROM Candidat;
SELECT * FROM Recruteur;
SELECT * FROM Offre;
SELECT * FROM Demande;
SELECT * FROM USER_OBJECTS;

```

```

/* CURSEURS */

SET SERVEROUTPUT ON;

/* Curseur paramétré C1 : Trier les offres selon le salaire */
DECLARE CURSOR C1 (v_sal Offre.Salaire%TYPE) IS
SELECT Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis
FROM Offre
WHERE Salaire > v_sal;
v_Id Offre.Id_offre%TYPE;
v_Nom Offre.Nom_offre%TYPE;
v_Date Offre.Date_publication%TYPE;
v_Secteur Offre.Secteur_act%TYPE;
v_Mode Offre.Mode_travail%TYPE;
v_Diplome Offre.Diplome_requis%TYPE;
BEGIN
OPEN C1 (300); /* Les offres dont le salaire dépasse 300 */
LOOP
FETCH C1 INTO v_Id, v_Nom, v_Date, v_Secteur, v_Mode, v_Diplome;
EXIT WHEN C1%NOTFOUND;

```

```

DBMS_OUTPUT.PUT_LINE('Id offre : ' || v_Id);
DBMS_OUTPUT.PUT_LINE('Intitulé offre: ' ||v_Nom);
DBMS_OUTPUT.PUT_LINE('Date publication: '||v_Date);
DBMS_OUTPUT.PUT_LINE('Secteur activité: '||v_Secteur);
DBMS_OUTPUT.PUT_LINE('Mode de travail: '||v_Mode);
DBMS_OUTPUT.PUT_LINE('Diplome requis : '||v_Diplome);
DBMS_OUTPUT.PUT_LINE(' ');
END LOOP;
CLOSE C1;
END;
/

/* Curseur paramétré C2 : Trier les offres selon le secteur d'activité */
DECLARE CURSOR C2 (v_act Candidat.Secteur_act_cdt%TYPE) IS
SELECT Id_offre
FROM Offre
WHERE v_act = Secteur_act;
v_offre Offre.Id_offre%TYPE;
BEGIN
OPEN C2 ('Information Systems'); /* Test avec le secteur d'activité Information Systems */
LOOP
FETCH C2 INTO v_offre;
EXIT WHEN C2%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_offre);
END LOOP;
CLOSE C2;
END;
/

/* Curseur simple C3 */
/* Trier offre par date de publication et diplôme */
DECLARE CURSOR C3 IS
SELECT Nom_offre, Date_publication
FROM Offre
WHERE Salaire>200
AND
Diplome_requis='Masters';
Nom Offre.Nom_offre%TYPE;
DatePub Offre.Date_publication%TYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE('Noms offres :      Date publication :');
OPEN C3;
LOOP
FETCH C3 INTO Nom, DatePub;
EXIT WHEN C3%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(Nom||'      '||DatePub);
END LOOP;
CLOSE C3;
END;
/

/* Curseur C4 */
/* Trier offre par date */
DECLARE CURSOR C4 IS
SELECT Nom_offre, Diplome_requis, Salaire
FROM Offre
ORDER BY Date_publication;
Nom Offre.Nom_offre%TYPE;
Diplome Offre.Diplome_requis%TYPE;
Sal Offre.Salaire%TYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE('Les offres triées par date :');
OPEN C4;
LOOP
FETCH C4 INTO Nom, Diplome, Sal;
EXIT WHEN C4%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(Nom || ' ' || Diplome || ' ' || Sal);
END LOOP;
CLOSE C4;
END;
/

```

Curseurs utilisés :

- **C1** : Un curseur paramétré qui permet de trier les offres d'emploi selon le critère salaire.
- **C2** : Un curseur paramétré qui permet de trier les offres d'emploi selon le secteur d'activité.
- **C3** : Un curseur simple qui permet de trier les offres d'emploi selon deux critères : Salaire et Diplôme. Ce curseur permet d'afficher les offres d'emploi, dont le salaire dépasse 200 et dont le diplôme requis est Masters.
- **C4** : Un curseur simple qui permet de trier les offres d'emploi selon la date de publication.

```

/* EXCEPTIONS */

/* Une exception rejeter_candidature est déclenchée si la demande du candidat est rejetée, permettant d'insérer */
/* les différentes informations du dossier en question (dossier rejeté) dans une table SQL Rejet */
DROP TABLE Rejet;
CREATE TABLE Rejet(Id char(3), Datecand Date, Etat varchar2(30));
DECLARE CURSOR C2 IS SELECT Id_demande, Date_candidature, Etat_dossier
FROM Demande
WHERE Etat_dossier ='Rejete';
v_Id Demande.Id_demande%TYPE;
v_date Demande.Date_candidature%TYPE;
v_etat Demande.Etat_dossier%TYPE;
rejeter_candidature EXCEPTION;
BEGIN
OPEN C2;
LOOP
FETCH C2 INTO v_Id, v_date, v_etat;
EXIT WHEN C2%NOTFOUND;
IF v_etat = 'Rejete' THEN RAISE rejeter_candidature;
END IF;
END LOOP;
CLOSE C2;
EXCEPTION
WHEN rejeter_candidature THEN INSERT INTO Rejet VALUES(v_Id, v_date, v_etat);
END;
/
SELECT * FROM Rejet;

```

Les exceptions :

- rejeter_candidature** : Une exception déclenchée si le dossier de candidature est rejeté. Si la condition est vérifiée, on insère les différentes informations du dossier en question dans une table SQL 'Rejet'.

```

/* PROCEDURES ET FONCTIONS */

SET SERVEROUTPUT ON;

/* Fonction filtrer_mode */
CREATE OR REPLACE FUNCTION filtrer_mode (v_mode Offre.Mode_travail%TYPE)
RETURN NUMBER IS
v_filtre Offre%ROWTYPE;
Nombre_offre INTEGER;
BEGIN
SELECT COUNT(*)
INTO Nombre_offre
FROM Offre
WHERE mode_travail = v_mode;
RETURN (Nombre_offre);
END filtrer_mode;
/

/* Execution de la fonction filtrer_mode */
VARIABLE M1 NUMBER;
EXECUTE :M1:=filtrer_mode('Part time');
PRINT M1;

/* Execution de la fonction filtrer_mode */
VARIABLE M2 NUMBER;
EXECUTE :M2:=filtrer_mode('Full time');
PRINT M2;

/* Fonction etat_candidature */
/* Fonction stockée etat_candidature qui renvoie le nombre de dossiers de candidature quelque soit l'état donnée */
CREATE OR REPLACE FUNCTION etat_candidature(V_etat Demande.Etat_dossier%TYPE)
RETURN NUMBER IS
nb_cdt INTEGER;
BEGIN
SELECT COUNT(*)
INTO nb_cdt
FROM Demande
WHERE Etat_dossier=V_etat;
RETURN (nb_cdt);
END etat_candidature;
/
/* Execution de la fonction etat_candidature (dans le cas d'une candidature retenue) */
VARIABLE nb NUMBER;
EXECUTE :nb:=etat_candidature('Retenu');
PRINT nb;

/* Execution de la fonction etat_candidature (dans le cas d'une candidature en cours de traitement) */
VARIABLE nb2 NUMBER;
EXECUTE :nb2:=etat_candidature('en cours de traitement');

```

```

PRINT nb2;

/* Execution de la fonction etat_candidature (dans le cas d'une candidature rejetee *)
VARIABLE nb3 NUMBER;
EXECUTE :nb3:=etat_candidature('Rejete');
PRINT nb3;

/* Procedures de suppression */

/* Procédure Remove_cdt */
CREATE OR REPLACE PROCEDURE Remove_cdt(v_cdt in Candidat.Id_cdt%TYPE) IS
BEGIN
DELETE FROM Candidat
WHERE Id_cdt = v_cdt;
COMMIT;
END Remove_cdt;
/

/* Procédure Remove_entp */
CREATE OR REPLACE PROCEDURE Remove_entp(v_entp in Recruteur.Id_entp%TYPE) IS
BEGIN
DELETE FROM Recruteur
WHERE Id_entp = v_entp;
COMMIT;
END Remove_entp;
/

/* Procédure Remove_offre */
CREATE OR REPLACE PROCEDURE Remove_offre(v_offre in Offre.Id_offre%TYPE) IS
BEGIN
DELETE FROM Offre
WHERE v_offre = Id_offre;
COMMIT;
END Remove_offre;
/

/* Procédure experience_cdt */
CREATE OR REPLACE PROCEDURE experience_cdt(v_Id Candidat.Id_cdt%TYPE) IS
v_exp Candidat.Experience%TYPE;
BEGIN
SELECT Experience
INTO v_exp
FROM Candidat
WHERE v_Id = Id_cdt;
IF v_exp>0 THEN DBMS_OUTPUT.PUT_LINE('Le candidat qui a pour id'||v_Id||' a de l expérence.');
ELSE
DBMS_OUTPUT.PUT_LINE('pas d experience');
END IF;
END experience_cdt;
/

/* Execution */
EXECUTE experience_cdt('1');
EXECUTE experience_cdt('20');

/* Procédure retenir_candidature */
/* Procédure qui permet d'insérer le dossier de candidature choisi dans une table de sauvegarde */
DROP TABLE Confirmation;
CREATE TABLE Confirmation(Id_dossier char(3), date_dossier date, etat_dossier varchar2(30));

CREATE OR REPLACE PROCEDURE retenir_candidature(v_id Demande.Id_demande%TYPE) IS
dossier Demande%ROWTYPE;
BEGIN
INSERT INTO Confirmation VALUES (dossier.Id_demande, dossier.Date_candidature, dossier.Etat_dossier);
END retenir_candidature;
/

/* Procédure stockée update_salaire pour mettre à jours le salaire */
CREATE OR REPLACE PROCEDURE update_salaire(Id IN Offre.Id_offre%TYPE, pourcentage IN NUMBER) IS
v_salaire Offre.Salaire%TYPE;
BEGIN
SELECT Salaire
INTO v_salaire
FROM Offre
WHERE Id=Id_offre;
DBMS_OUTPUT.PUT_LINE('Le salaire avant la mise à jour est: '||v_salaire);
UPDATE Offre
SET salaire = salaire*pourcentage
WHERE Id=Id_offre;
SELECT Salaire
INTO v_salaire

```

```

FROM Offre
WHERE Id=Id_offre;
DBMS_OUTPUT.PUT_LINE('Le salaire avant la mise à jour est: '||v_salaire);
END update_salaire;
/

```

Procédures et fonctions utilisées :

- **Remove_cdt** : une procédure stockée pour retirer un candidat dont l'Id est spécifié.
- **Remove_entp** : une procédure stockée pour retirer un recruteur dont l'Id est spécifié.
- **Remove_offre** : une procédure stockée pour retirer une offre d'emploi dont l'Id est spécifié.
- **experience_cdt** : une procédure stockée qui permet de vérifier si un candidat donné a de l'expérience ou pas.
- **retenir_candidature** : une procédure stockée qui permet d'insérer le dossier de candidature donné dans une table de sauvegarde.
- **update_salaire** : une procédure stockée pour mettre à jour le salaire de l'offre d'emploi proposée, selon un pourcentage donnée.
- **etat_candidature** : une fonction stockée qui renvoie le nombre de candidatures, qu'elles soient retenues, en cours de traitement ou rejetées.
- **filtrer_offre** : une fonction stockée qui renvoie le nombre d'offres d'emploi, dont le mode de travail est spécifié (soit Full-time, soit Part-time)

```

/* TRIGGERS */

/* TRIGGER salary_update */
CREATE OR REPLACE TRIGGER salary_update
BEFORE UPDATE OF Salaire
ON Offre
FOR EACH ROW
BEGIN
IF((:OLD.Salaire/:NEW.Salaire) >= 2.5) THEN
RAISE_APPLICATION_ERROR(-20002, 'Pourcentage de mise à jour élevé');
END IF;
END;
/

/* TRIGGER dateUpdate */
CREATE OR REPLACE TRIGGER dateUpdate
BEFORE INSERT OR UPDATE OF Date_publication
ON Offre
FOR EACH ROW
BEGIN
IF (:NEW.Date_publication > sysdate) THEN
RAISE_APPLICATION_ERROR(-20000, 'La date indiquée est supérieure à la date du système');
END IF;
END;
/

/* TEST DU TRIGGER dateUpdate */
INSERT INTO Offre (Id_offre, Nom_offre, Date_publication, Secteur_act, Mode_travail, Diplome_requis, Salaire)
VALUES ( '11', 'Business analyst manager', '27/02/2023', 'Information Systems', 'Full time', 'Bachelor', 108.397);

```

```

/* PACKAGE Emploi */

/* SPECIFICATION */
CREATE OR REPLACE PACKAGE Emploi AS

FUNCTION filtrer_mode (v_mode Offre.Mode_travail%TYPE) RETURN NUMBER;
FUNCTION etat_candidature(V_etat Demande.Etat_dossier%TYPE) RETURN NUMBER;
PROCEDURE Remove_cdt(v_cdt in Candidat.Id_cdt%TYPE);
PROCEDURE Remove_entp(v_entp in Recruteur.Id_entp%TYPE);
PROCEDURE Remove_offre(v_offre in Offre.Id_offre%TYPE);
PROCEDURE experience_cdt(v_Id Candidat.Id_cdt%TYPE);
PROCEDURE retenir_candidature(v_id Demande.Id_demande%TYPE);
PROCEDURE update_salaire(Id IN Offre.Id_offre%TYPE, pourcentage IN NUMBER);
END Emploi;
/

/* BODY */
CREATE OR REPLACE PACKAGE BODY Emploi AS

```

```

FUNCTION filtrer_mode (v_mode Offre.Mode_travail%TYPE)
RETURN NUMBER IS
v_filtre Offre%ROWTYPE;
Nombre_offre INTEGER;
BEGIN
SELECT COUNT(*)
INTO Nombre_offre
FROM Offre
WHERE mode_travail = v_mode;
RETURN (Nombre_offre);
END filtrer_mode;

FUNCTION etat_candidature(V_etat Demande.Etat_dossier%TYPE)
RETURN NUMBER IS
nb_cdt INTEGER;
BEGIN
SELECT COUNT(*)
INTO nb_cdt
FROM Demande
WHERE Etat_dossier=V_etat;
RETURN (nb_cdt);
END etat_candidature;

PROCEDURE Remove_cdt(v_cdt in Candidat.Id_cdt%TYPE) IS
BEGIN
DELETE FROM Candidat
WHERE Id_cdt = v_cdt;
COMMIT;
END Remove_cdt;

PROCEDURE Remove_entp(v_entp in Recruteur.Id_entp%TYPE) IS
BEGIN
DELETE FROM Recruteur
WHERE Id_entp = v_entp;
COMMIT;
END Remove_entp;

PROCEDURE Remove_offre(v_offre in Offre.Id_offre%TYPE) IS
BEGIN
DELETE FROM Offre
WHERE v_offre = Id_offre;
COMMIT;
END Remove_offre;

PROCEDURE experience_cdt(v_Id Candidat.Id_cdt%TYPE) IS
v_exp Candidat.Experience%TYPE;
BEGIN
SELECT Experience
INTO v_exp
FROM Candidat
WHERE v_Id = Id_cdt;
IF v_exp>0 THEN DBMS_OUTPUT.PUT_LINE('Le candidat qui a pour id '||v_Id || 'a de l expérience.');
ELSE
DBMS_OUTPUT.PUT_LINE('pas d experience');
END IF;
END experience_cdt;

PROCEDURE retenir_candidature(v_id Demande.Id_demande%TYPE) IS
dossier Demande%ROWTYPE;
BEGIN
INSERT INTO Confirmation VALUES (dossier.Id_demande, dossier.Date_candidature, dossier.Etat_dossier);
END retenir_candidature;

PROCEDURE update_salaire(Id IN Offre.Id_offre%TYPE, pourcentage IN NUMBER) IS
v_salaire Offre.Salaire%TYPE;
BEGIN
SELECT Salaire
INTO v_salaire
FROM Offre
WHERE Id=Id_offre;
DBMS_OUTPUT.PUT_LINE('Le salaire avant la mise à jour est: '||v_salaire);
UPDATE Offre
SET Salaire = Salaire*pourcentage
WHERE Id=Id_offre;
SELECT Salaire
INTO v_salaire
FROM Offre
WHERE Id=Id_offre;
DBMS_OUTPUT.PUT_LINE('Le salaire avant la mise à jour est: '||v_salaire);
END update_salaire;

END Emploi;
/

```

Conclusion :

La mise en contact de cette application de recrutement peut être faite différemment et automatiquement, en faisant la correspondance entre le secteur d'activité du candidat d'une part et de l'offre d'autre part. Cette amélioration possible est considérée comme solution, qui peut être implémentée comme une première phase ou un premier critère d'acceptation du dossier de candidature.