

Projet SGBD

1-Créer une procédure P_AFF_INC qui affiche la liste des copies ayant l'état 'D' malgré qu'elles sont actuellement prêtées et non retournées (DATER est NULL). Afficher pour chacune des copies le titre du livre, le code de la copie concernée, son état, la date de prêt et la date de retour (qui est forcément NUL).

```
CREATE OR REPLACE PROCEDURE P_AFF_INC
IS
CURSOR cur_ouv IS SELECT o.titre, c.idc, c.etat, p.datep, p.dater from
ouvrage o, copie c, pret p
WHERE o.ido= c.ido AND c.idc=p.idc AND etat='D' AND p.dater IS NULL;
c_record cur_ouv%ROWTYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE((RPAD('TITRE',60,' '))||(RPAD('IDC',11,'
'))||(RPAD('ETAT',11,' '))||(RPAD('DATEP',11,' '))||(RPAD('DATER',11,' ')));
OPEN cur_ouv;
LOOP
FETCH cur_ouv INTO c_record;
EXIT WHEN cur_ouv%NOTFOUND;
DBMS_OUTPUT.PUT_LINE((RPAD(c_record.titre,60,'
'))||(RPAD(c_record.idc,11,' '))||(RPAD(c_record.etat,11,'
'))||(RPAD(c_record.datep,11,' '))||(RPAD(c_record.dater,11,' ')));
END LOOP;
CLOSE cur_ouv;
END P_AFF_INC;
/
show error procedure P_AFF_INC;

SET SERVEROUTPUT ON
BEGIN
P_AFF_INC();
END;
```

0,26499999 secondes

Feuille de calcul Query Builder

```
CREATE OR REPLACE PROCEDURE P_AFF_INC
IS
CURSOR cur_ouv IS SELECT o.titre, c.idc, c.etat, p.datep, p.dater from ouvrage o, copie c, pret p
WHERE o.ido= c.ido AND c.idc=p.idc AND etat='D' AND p.dater IS NULL;
c_record cur_ouv%ROWTYPE;
BEGIN
DBMS_OUTPUT.PUT_LINE((RPAD('TITRE',60,' '))|| (RPAD('IDC',11,' '))|| (RPAD('ETAT',11,' '))|| (RPAD('DATEP',11,' '))|| (RPAD('DATER',11,' ')));
OPEN cur_ouv;
LOOP
FETCH cur_ouv INTO c_record;
EXIT WHEN cur_ouv%NOTFOUND;
DBMS_OUTPUT.PUT_LINE((RPAD(c_record.titre,60,' '))|| (RPAD(c_record.idc,11,' '))|| (RPAD(c_record.etat,11,' '))
|| (RPAD(c_record.datep,11,' '))|| (RPAD(c_record.dater,11,' ')));
END LOOP;
CLOSE cur_ouv;
END P_AFF_INC;
/
show error procedure P_AFF_INC; --Afficher les erreurs du code de la procédure--
SET SERVEROUTPUT ON
BEGIN
P_AFF_INC();
END;
```

Sortie de script x Résultat de requête x

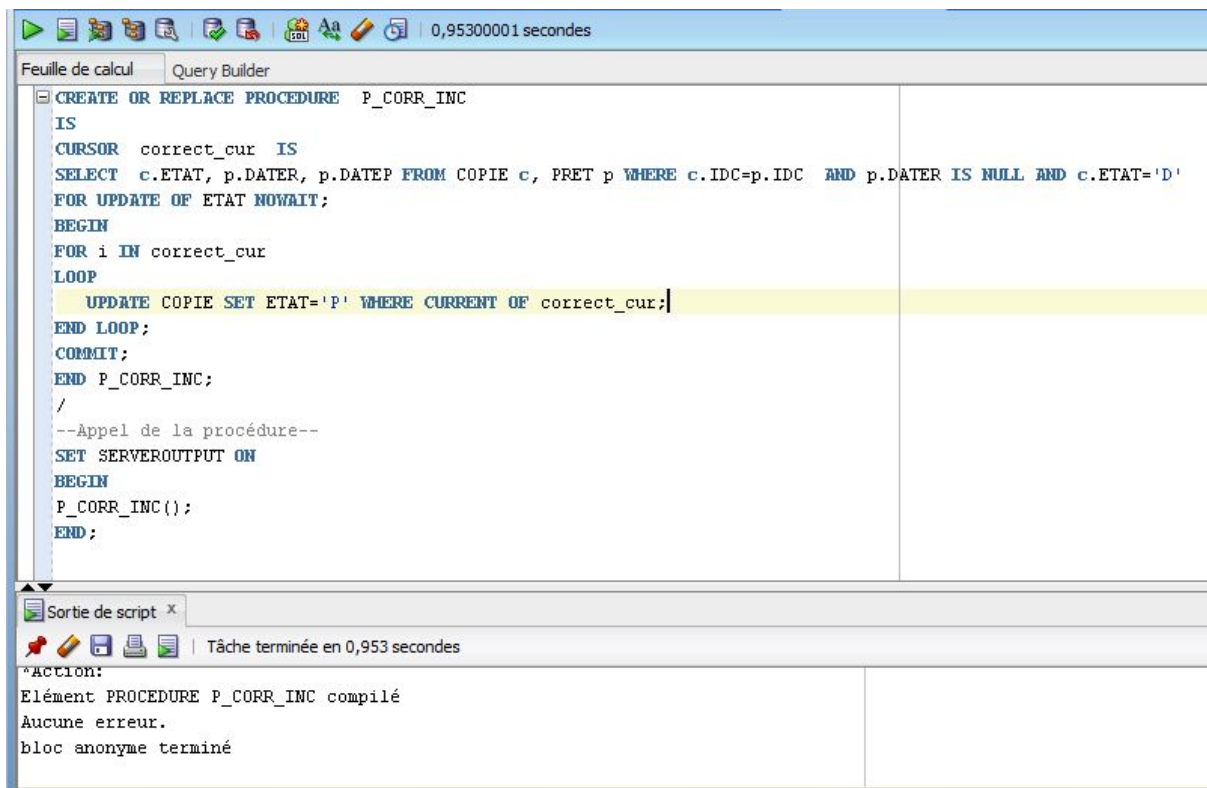
Tâche terminée en 0,265 secondes

Elément PROCEDURE P_AFF_INC compilé

2-Créer une procédure P_CORR_INC qui corrige cette situation incohérente en affectant l'état 'P' aux copies actuellement prêtées et non retournées.

```
CREATE OR REPLACE PROCEDURE P_CORR_INC
IS
CURSOR correct_cur IS
SELECT c.ETAT, p.DATER, p.DATEP FROM COPIE c, PRET p WHERE
c.IDC=p.IDC AND p.DATER IS NULL AND c.ETAT='D'
FOR UPDATE OF ETAT NOWAIT;
BEGIN
FOR i IN correct_cur
LOOP
    UPDATE COPIE SET ETAT='P' WHERE CURRENT OF correct_cur;

END LOOP;
COMMIT;
END P_CORR_INC;
/
```



3-Créer une procédure P_AFF_PRETS qui affiche les prêts remis et non remis d'un étudiant dont on entre le CIN en paramètre. Afficher le code de l'étudiant, son CIN, le code de la copie, le titre de l'ouvrage, la date de prêt et la date de remise.

```

SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE P_AFF_PRETS(ncin IN ETUDIANT.CIN%TYPE)
IS
  CURSOR c_pret(ncin VARCHAR2) IS SELECT e.IDE, e.CIN, p.DATEP, p.DATER,
o.TITRE, c.IDC
  FROM OUVRAGE o, COPIE c, ETUDIANT e, PRET p WHERE
  ncin=e.CIN AND c.IDC=p.IDC AND e.IDE=p.IDE AND c.IDO=o.IDO;
  r_pret c_pret%ROWTYPE;
BEGIN
  OPEN c_pret(ncin);
LOOP
  FETCH c_pret INTO r_pret;
  EXIT WHEN c_pret%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('LE NUMERO DE L ETUDIANT EST:' || r_pret.IDE||' CIN:'
||r_pret.CIN ||' LE CODE COPIE :'||r_pret.IDC|| 'LE TITRE est '||r_pret.TITRE||'Datepret
:'||r_pret.DATEP||'dateretour est '||
r_pret.DATER );
END LOOP ;
CLOSE c_pret;
END P_AFF_PRETS;
/
show error procedure P_AFF_PRETS;
execute P_AFF_PRETS('66665335');

```

The screenshot shows a SQL Developer window with the following components:

- Top Bar:** Displays icons for various tools and a timer showing "0,266 secondes".
- Feuille de calcul / Query Builder:** The main area containing the SQL script for the procedure P_AFF_PRETS. The script is as follows:


```

SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE P_AFF_PRETS(ncin IN ETUDIANT.CIN%TYPE)
IS
  CURSOR c_pret(ncin VARCHAR2) IS SELECT e.IDE, e.CIN, p.DATEP, p.DATER, o.TITRE, c.IDC
  FROM OUVRAGE o, COPIE c, ETUDIANT e, PRET p WHERE
  ncin=e.CIN AND c.IDC=p.IDC AND e.IDE=p.IDE AND c.IDO=o.IDO;
  r_pret c_pret%ROWTYPE;
BEGIN
  OPEN c_pret(ncin);
LOOP
  FETCH c_pret INTO r_pret;
  EXIT WHEN c_pret%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('LE NUMERO DE L ETUDIANT EST:' || r_pret.IDE||' CIN:'
||r_pret.CIN ||' LE CODE COPIE :'||r_pret.IDC|| 'LE TITRE est '||r_pret.TITRE||'Datepret
:'||r_pret.DATEP||'dateretour est '||
r_pret.DATER );
END LOOP ;
CLOSE c_pret;
END P_AFF_PRETS;
/
show error procedure P_AFF_PRETS;

```
- Bottom Bar:** Contains tabs for "Sortie de script" and "Résultat de requête". Below these, it shows the execution status: "Tâche terminée en 0,266 secondes".
- Output Area:** Displays the results of the procedure execution:


```

bloc anonyme terminé
Élément PROCEDURE P_AFF_PRETS compilé
Aucune erreur.

```

4-Créer un trigger T_MAJ_P qui se déclenche avant l'insertion d'une ligne dans PRÊT. Ce trigger doit mettre à jour la colonne ETAT de la copie à prêter à 'P'. Il doit aussi lire le nom de l'utilisateur qui a lancé la requête INSERT, la chercher dans la table BIBLIO et affecter son identifiant au champ IDB de la ligne à insérer.

```
CREATE OR REPLACE TRIGGER T_MAJ_P
BEFORE INSERT ON PRET
FOR EACH ROW
```

```
DECLARE
```

```
v_user VARCHAR2(30);
```

```
v_idb VARCHAR2(10);
```

```
v_idc VARCHAR2(10);
```

```
BEGIN
```

```
IF INSERTING THEN
```

```
SELECT :NEW.IDC INTO v_idc
```

```
FROM PRET ;
```

```
UPDATE COPIE
```

```
SET ETAT='P'
```

```
WHERE COPIE.IDC LIKE v_idc;
```

```
SELECT USER INTO v_user
```

```
FROM DUAL;
```

```
SELECT IDB INTO v_idb
```

```
FROM BIBLIO
```

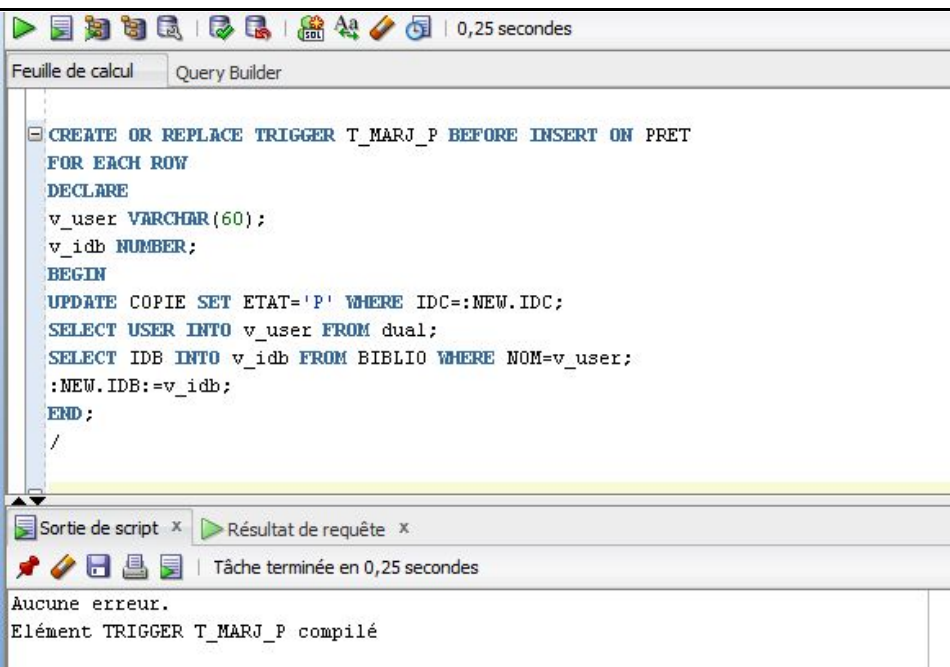
```
WHERE BIBLIO.NOM LIKE v_user;
```

```
UPDATE PRET SET IDB=v_idb;
```

```
END IF;
```

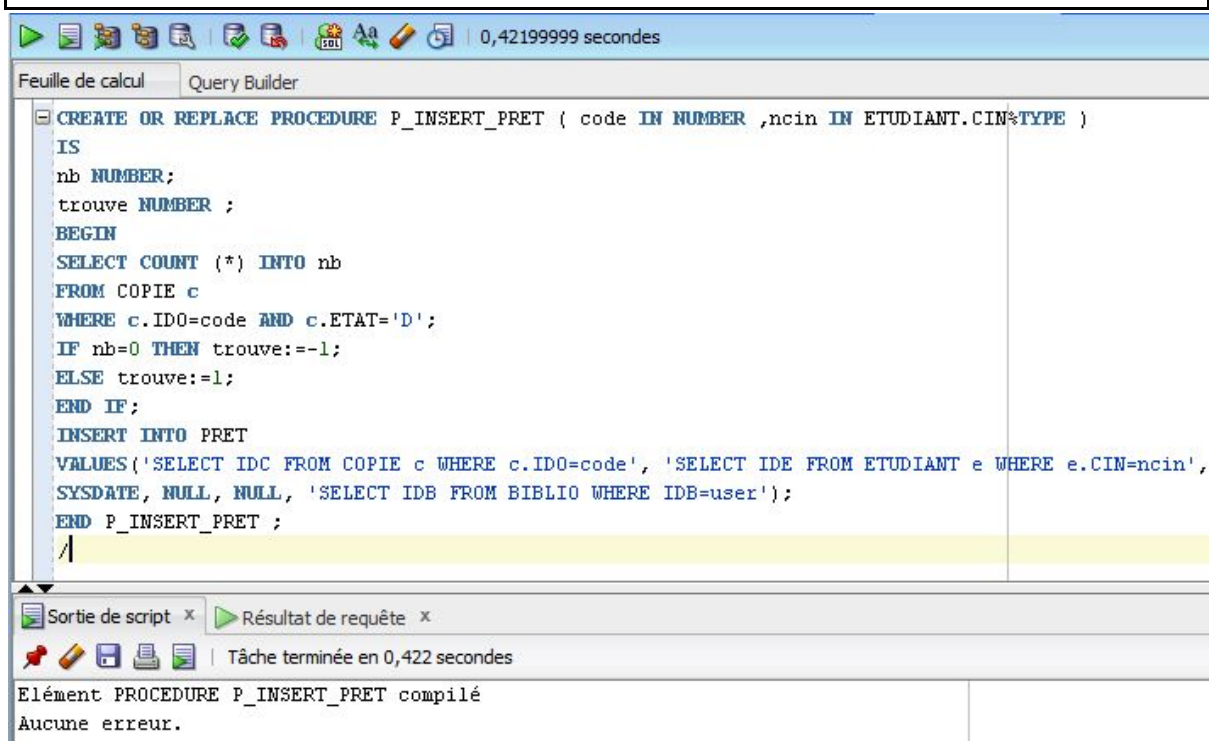
```
END;
```

```
/
```



5-Créer une procédure P_INSERT_PRET qui enregistre le prêt d'un ouvrage de code donné à un étudiant de CIN donné avec comme date de prêt la date système. La procédure doit trouver une copie disponible de cet ouvrage et ensuite insérer une ligne dans PRET en utilisant le code de la copie, le code de l'étudiant et la date système comme date de prêt. Les autres champs doivent être NULL. La procédure retourne dans un paramètre la valeur 1 si le prêt a été effectué, et -1 si la procédure n'a pas trouvé de copie disponible pour l'ouvrage en question.

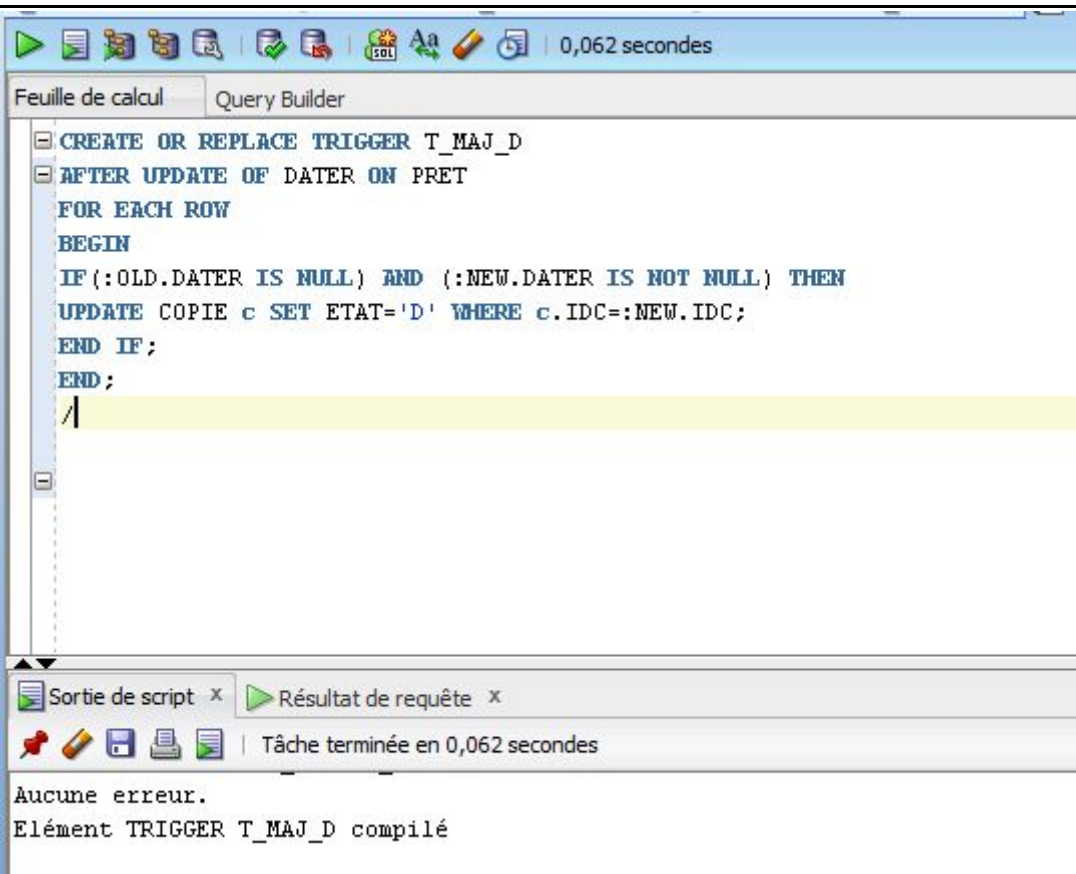
```
CREATE OR REPLACE PROCEDURE P_INSERT_PRET ( code IN NUMBER ,ncin IN
NUMBER )
IS
nb NUMBER;
trouve NUMBER ;
BEGIN
SELECT COUNT (*) INTO nb
FROM COPIE c
WHERE c.IDO=code AND c.ETAT='D';
IF nb=0 THEN trouve:=-1;
ELSE trouve:=1;
END IF;
INSERT INTO PRET
VALUES('SELECT IDC FROM COPIE c WHERE c.IDO=code', 'SELECT IDE FROM
ETUDIANT e WHERE e.CIN=ncin', SYSDATE, NULL, NULL, 'SELECT IDB FROM BIBLIO
WHERE IDB=user');
END P_INSERT_PRET ;
/
show error PROCEDURE P_INSERT_PRET;
```



6-Créer un trigger T_MAJ_D qui se déclenche après la modification du champ 'DATER' de la table PRET si l'ancienne valeur est NULL et que la nouvelle n'est pas NULL

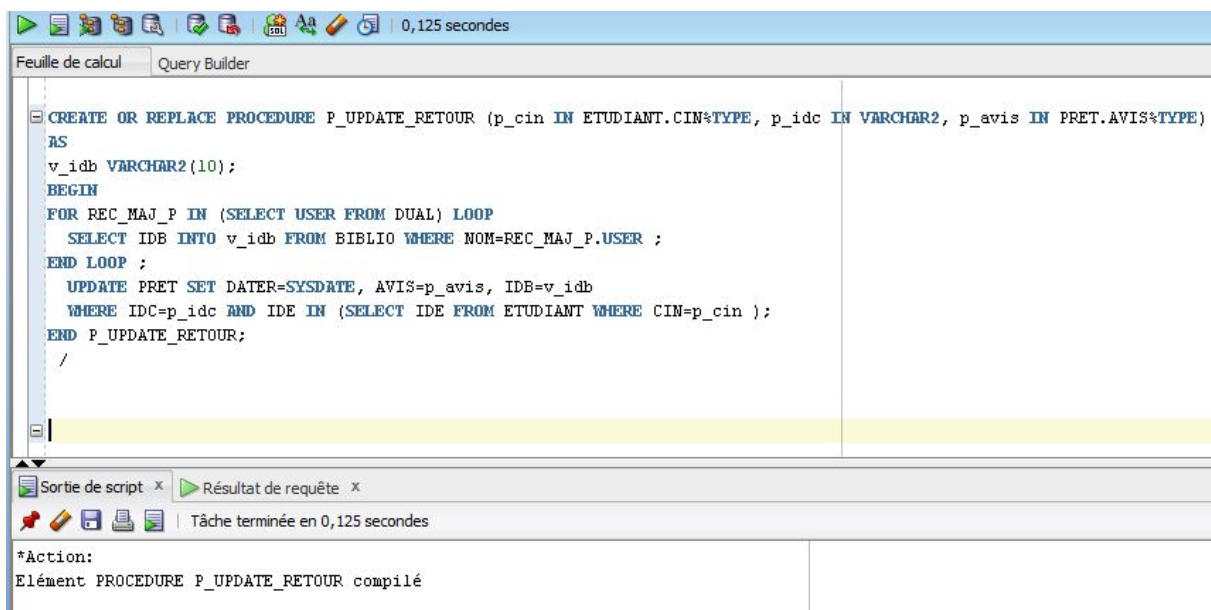
(ici, nous sommes dans le cas d'une modification d'une date de retour de NULL à une valeur non nulle, et donc nous sommes dans le cas d'un retour de prêt). Le trigger doit mettre à jour la copie concernée par ce prêt en modifiant son état à 'D'.

```
CREATE OR REPLACE TRIGGER T_MAJ_D
AFTER UPDATE OF DATER ON PRET
FOR EACH ROW
BEGIN
IF(:OLD.DATER IS NULL) AND (:NEW.DATER IS NOT NULL) THEN
UPDATE COPIE c SET ETAT='D' WHERE c.IDC=:NEW.IDC;
END IF;
END;
/
```



-7-Créer une procédure P_UPDATE_RETOUR qui affecte à un prêt la date de retour, l'avis ('LIKE' ou 'DISLIKE') de l'étudiant par rapport à l'ouvrage emprunté, et le code du bibliothécaire qui effectue le retour de prêt (lire le nom d'utilisateur, et le chercher dans la table BIBLIO). Le prêt est identifié par le CIN de l'étudiant et le code de la copie prêtée.

```
CREATE OR REPLACE PROCEDURE P_UPDATE_RETOUR (p_cin IN VARCHAR2,
p_idc IN VARCHAR2, p_avis IN VARCHAR2)
AS
    v_idb VARCHAR2(10);
BEGIN
    FOR REC_MAJ_P IN (SELECT USER FROM DUAL ) LOOP
        SELECT IDB INTO v_idb FROM BIBLIO WHERE
NOM=REC_MAJ_P.USER ;
    END LOOP ;
    UPDATE PRET SET DATER=SYSDATE, AVIS=p_avis, IDB=v_idb
    WHERE IDC=p_idc AND IDE IN (SELECT IDE FROM ETUDIANT
WHERE CIN=p_cin );
    END P_UPDATE_RETOUR;
/
```



8-Créer une procédure P_MAJ_LIKES qui affecte aux champs LIKES et DISLIKES de la table ouvrage le nombre de LIKES et de DISLIKES reçus pour chaque ouvrage.

```
SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE P_MAJ_LIKES
IS
  CURSOR ouv_cur IS SELECT * FROM OUVRAGE;
  CURSOR like_cur(p_ido varchar2) IS SELECT p.AVIS , p.IDC , c.IDO FROM PRET p ,
  COPIE c , OUVRAGE o WHERE p.IDC=c.IDC AND o.IDO=c.IDO AND o.IDO=p_ido
  AND AVIS='LIKE' ;
  CURSOR dislike_cur(p_ido varchar2) IS SELECT p.AVIS, p.IDC FROM PRET p , COPIE
  c ,
  OUVRAGE o WHERE p.IDC=c.IDC AND o.IDO=c.IDO AND o.IDO=p_ido AND
  AVIS='DISLIKE' ;
  v_like NUMBER:=0 ;
  v_dislike NUMBER:=0;
BEGIN
  FOR ouv_rec IN ouv_cur LOOP
    --DBMS_OUTPUT.PUT_LINE('IDO '|| ouv_rec.ido );

    FOR like_rec IN like_cur(ouv_rec.IDO) LOOP
      v_like:=v_like+1;
    END LOOP;
    UPDATE OUVRAGE SET LIKES=v_like WHERE ouv_rec.IDO=IDO;
    --DBMS_OUTPUT.PUT_LINE('like= '|| v_like);
    v_like:=0 ;
    v_dislike:=0;
  END LOOP;

  FOR ouv_rec IN ouv_cur LOOP
    FOR dislike_rec IN dislike_cur(ouv_rec.IDO) LOOP
      v_dislike:=v_dislike+1;
    END LOOP;
    UPDATE OUVRAGE SET DISLIKES=v_dislike where ouv_rec.IDO=IDO;
    v_like:=0 ;
    v_dislike:=0;
  END LOOP;
END;
/
```

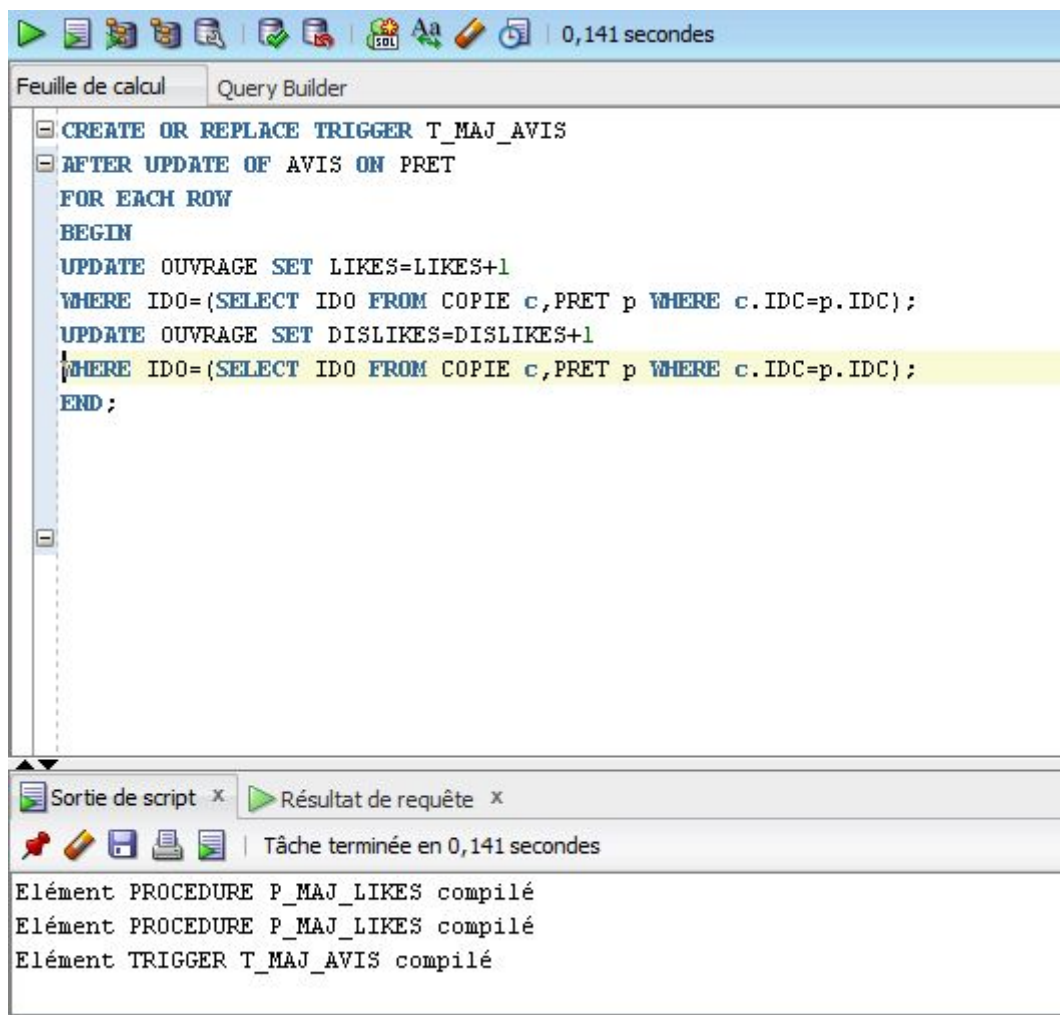
0,016 secondes

Feuille de calcul Query Builder

```
CREATE OR REPLACE PROCEDURE P_MAJ_LIKES
IS
  CURSOR ouv_cur IS SELECT * FROM OUVRAGE;
  CURSOR like_cur(p_ido varchar2) IS SELECT p.AVIS , p.IDC , c.IDO FROM PRET p , COPIE c , OUVRAGE o
  WHERE p.IDC=c.IDC AND o.IDO=c.IDO AND o.IDO=p_ido AND AVIS='LIKE';
  CURSOR dislike_cur(p_ido varchar2) IS SELECT p.AVIS, p.IDC FROM PRET p , COPIE c, OUVRAGE o
  WHERE p.IDC=c.IDC AND o.IDO=c.IDO AND o.IDO=p_ido AND AVIS='DISLIKE' ;
  v_like NUMBER:=0 ;
  v_dislike NUMBER:=0;
BEGIN
  FOR ouv_rec IN ouv_cur LOOP
    --DBMS_OUTPUT.PUT_LINE('IDO '|| ouv_rec.ido );
    FOR like_rec IN like_cur(ouv_rec.IDO) LOOP
      v_like:=v_like+1;
    END LOOP;
    UPDATE OUVRAGE SET LIKES=v_like WHERE ouv_rec.IDO=IDO;
    --DBMS_OUTPUT.PUT_LINE('like= '|| v_like);
    v_like:=0 ;
    v_dislike:=0;
  END LOOP;
  FOR ouv_rec IN ouv_cur LOOP
    FOR dislike_rec IN dislike_cur(ouv_rec.IDO) LOOP
      v_dislike:=v_dislike+1;
    END LOOP;
    UPDATE OUVRAGE SET DISLIKES=v_dislike where ouv_rec.IDO=IDO;
    v_like:=0 ;
  END LOOP;
  v_dislike:=0;
  END LOOP;
END;
```

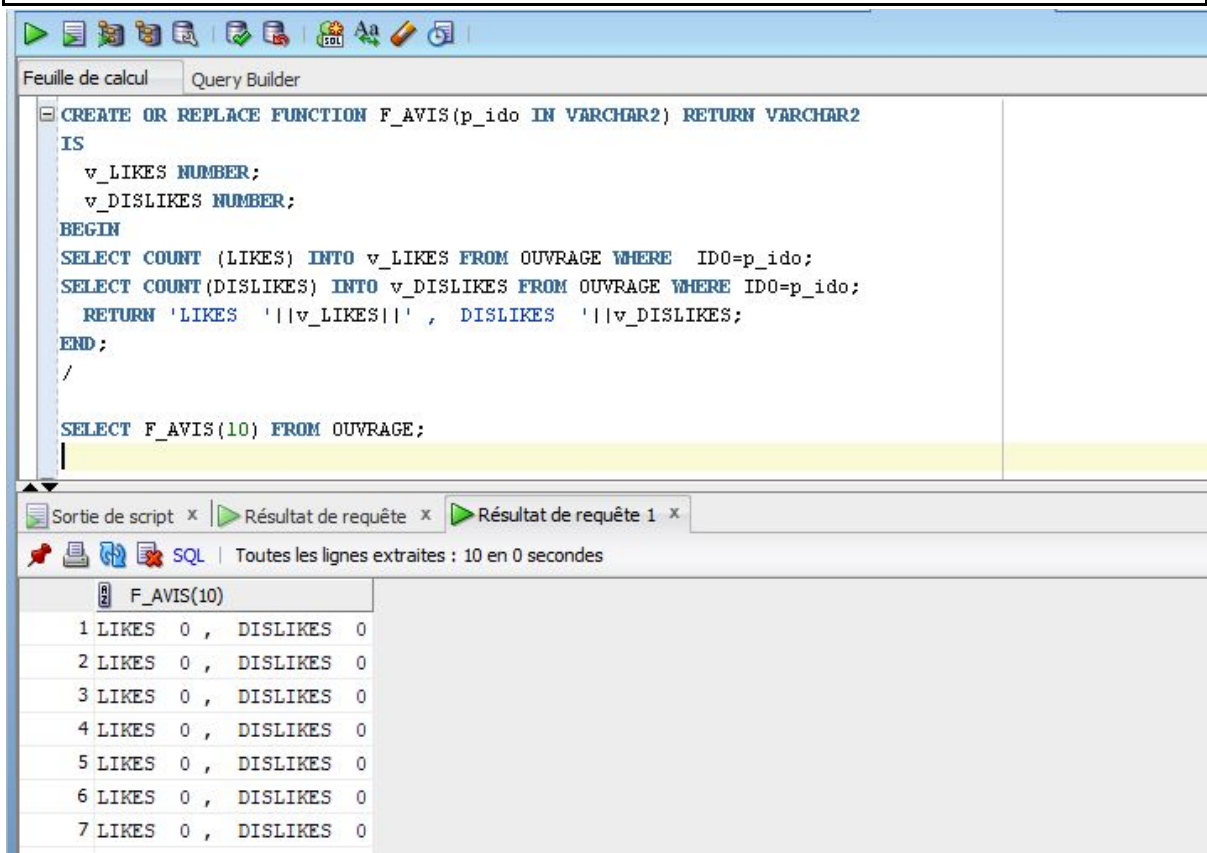
9-Créer un trigger T_MAJ_AVIS qui à chaque UPDATE de la colonne AVIS de la table PRET d'incrémenter les colonnes LIKES ou DISLIKES de l'ouvrage concerné en l'incrémentant.

```
CREATE OR REPLACE TRIGGER T_MAJ_AVIS
AFTER UPDATE OF AVIS ON PRET
FOR EACH ROW
BEGIN
UPDATE OUVRAGE SET LIKES=LIKES+1 WHERE IDO=(SELECT IDO FROM COPIE
c,PRET p WHERE c.IDC=p.IDC);
UPDATE OUVRAGE SET DISLIKES=DISLIKES+1 WHERE IDO=(SELECT IDO FROM
COPIE c,PRET p WHERE c.IDC=p.IDC);
END;
```



-10-Créer une fonction F_AVIS qui retourne une chaîne de caractères renseignant le nombre de LIKES et de DISLIKES d'un ouvrage donné sous cette forme 'LIKES 12, DISLIKES 3'.

```
CREATE OR REPLACE FUNCTION F_AVIS(IDO_pret IN VARCHAR2) RETURN
VARCHAR2
IS
    v_LIKES NUMBER:=0;
    v_DISLIKES NUMBER:=0;
BEGIN
SELECT COUNT(LIKES) INTO v_LIKES FROM OUVRAGE WHERE IDO=IDO_pret;
SELECT COUNT(DISLIKES) INTO v_DISLIKES FROM OUVRAGE WHERE
IDO=IDO_pret;
    RETURN 'LIKES'||v_LIKES||',DISLIKES'||v_DISLIKES;
END;
/
```



The screenshot shows a SQL development environment with a query editor and a results window. The query editor contains the following SQL code:

```
CREATE OR REPLACE FUNCTION F_AVIS(p_ido IN VARCHAR2) RETURN VARCHAR2
IS
    v_LIKES NUMBER;
    v_DISLIKES NUMBER;
BEGIN
SELECT COUNT (LIKES) INTO v_LIKES FROM OUVRAGE WHERE IDO=p_ido;
SELECT COUNT (DISLIKES) INTO v_DISLIKES FROM OUVRAGE WHERE IDO=p_ido;
    RETURN 'LIKES ' ||v_LIKES||' , DISLIKES ' ||v_DISLIKES;
END;
/

SELECT F_AVIS(10) FROM OUVRAGE;
```

The results window shows the output of the query. The title bar indicates 'Résultat de requête 1'. The status bar shows 'Toutes les lignes extraites : 10 en 0 secondes'. The results are displayed in a table with the following data:

	F_AVIS(10)
1	LIKES 0 , DISLIKES 0
2	LIKES 0 , DISLIKES 0
3	LIKES 0 , DISLIKES 0
4	LIKES 0 , DISLIKES 0
5	LIKES 0 , DISLIKES 0
6	LIKES 0 , DISLIKES 0
7	LIKES 0 , DISLIKES 0

11-Créer une vue V_CTO qui retourne pour chaque cursus, pour chaque theme, et pour chaque ouvrage, le nombre de prêts effectués. La vue est définie sur les colonnes CURSUS, THEME, OUVRAGE, NOMBRE.

```
CREATE OR REPLACE VIEW V_CTO AS SELECT o.IDO, o.THEME, e.CURSUS,
count(c.IDC) nb_pret
FROM PRET p, COPIE c, OUVRAGE o, ETUDIANT e
WHERE o.IDO=c.IDO AND c.IDC=p.IDC AND e.IDE=p.IDE AND c.ETAT='P' GROUP BY e.CURSUS, o.THEME, o.IDO;

SELECT * FROM V_CTO;
```

The screenshot shows a database query builder interface. The top section displays the SQL code for creating a view V_CTO and querying it. The bottom section shows the results of the query, which is a table with 9 rows and 4 columns: IDO, THEME, CURSUS, and NB_PRET.

	IDO	THEME	CURSUS	NB_PRET
1	O10	Gestion	E-BUSINESS	2
2	O2	Informatique	VIC	1
3	O4	Informatique	E-BUSINESS	1
4	O6	Informatique	DSSD	1
5	O10	Gestion	VIC	1
6	O5	Informatique	E-BUSINESS	2
7	O7	Gestion	DSSD	1
8	O9	Gestion	E-BUSINESS	1
9	O6	Informatique	E-BUSINESS	2

12-Afficher tous les objets BDs qui font partie du schéma de BIBLIO (tables, procédures, fonctions, vues, triggers et séquences) à partir du dictionnaire de données. Afficher aussi les utilisateurs.

```
SET SERVEROUTPUT ON
DECLARE
CURSOR cur_obj IS SELECT OBJECT_NAME, OBJECT_TYPE FROM user_objects;
BEGIN
DBMS_OUTPUT.PUT_LINE('*****OBJETS*****');
FOR REC IN cur_obj
LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(REC.OBJECT_NAME,70,' ')||
' ||REC.OBJECT_TYPE);
END LOOP;
END;
```

0,02 secondes

Feuille de calcul Query Builder

```
SET SERVEROUTPUT ON
DECLARE
CURSOR cur_obj IS SELECT OBJECT_NAME, OBJECT_TYPE FROM user_objects;
BEGIN
DBMS_OUTPUT.PUT_LINE('*****OBJETS*****');
FOR REC IN cur_obj
LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(REC.OBJECT_NAME,70,' ')||
' ||REC.OBJECT_TYPE);
END LOOP;
END;
```

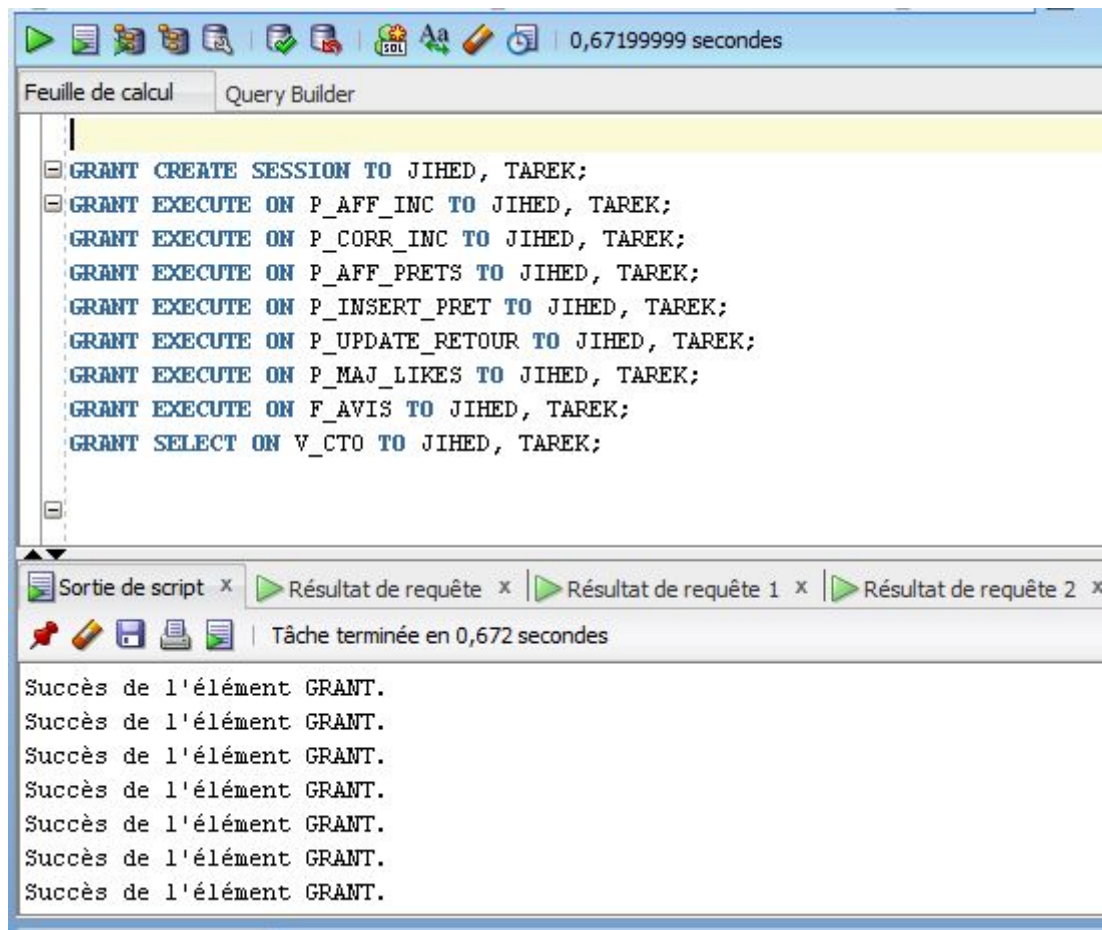
Sortie de script x Résultat de requête x Résultat de requête 1 x Résultat de requête 2 x

Tâche terminée en 0,02 secondes

P_AFF_PRETS	PROCEDURE
P_CORR_INC	PROCEDURE
P_MAJ_LIKES	PROCEDURE
T_MAJ_P	TRIGGER
P_INSERT_PRET	PROCEDURE
P_UPDATE_RETOUT	PROCEDURE
T_MAJ_AVIS	TRIGGER
F_AVIS	FUNCTION
V_CTO	VIEW
T_MAJ_D	TRIGGER

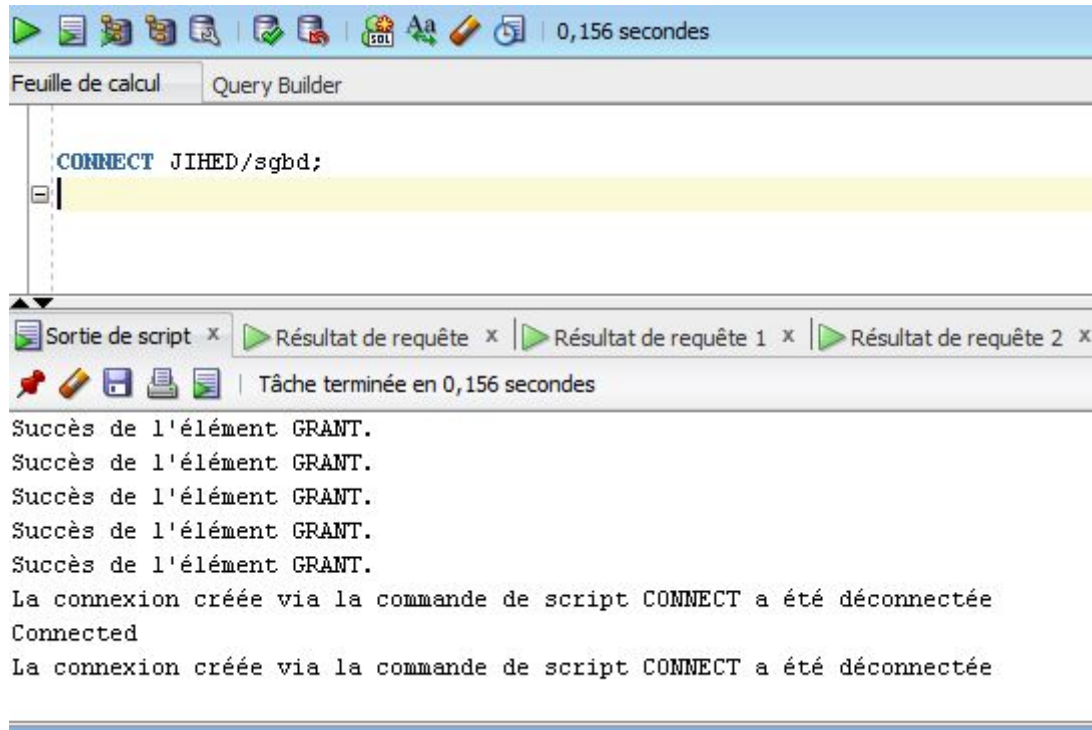
13-Donner à JIHED et TAREK les privilèges de connexion, d'exécution sur les procédures et fonctions créées, et d'accès à la vue V_CTO.

```
GRANT CREATE SESSION TO JIHED, TAREK;  
GRANT EXECUTE ON P_AFF_INC TO JIHED, TAREK;  
GRANT EXECUTE ON P_CORR_INC TO JIHED, TAREK;  
GRANT EXECUTE ON P_AFF_PRETS TO JIHED, TAREK;  
GRANT EXECUTE ON P_INSERT_PRET TO JIHED, TAREK;  
GRANT EXECUTE ON P_UPDATE_RETOUTR TO JIHED, TAREK;  
GRANT EXECUTE ON P_MAJ_LIKES TO JIHED, TAREK;  
GRANT EXECUTE ON F_AVIS TO JIHED, TAREK;  
GRANT SELECT ON V_CTO TO JIHED, TAREK;
```



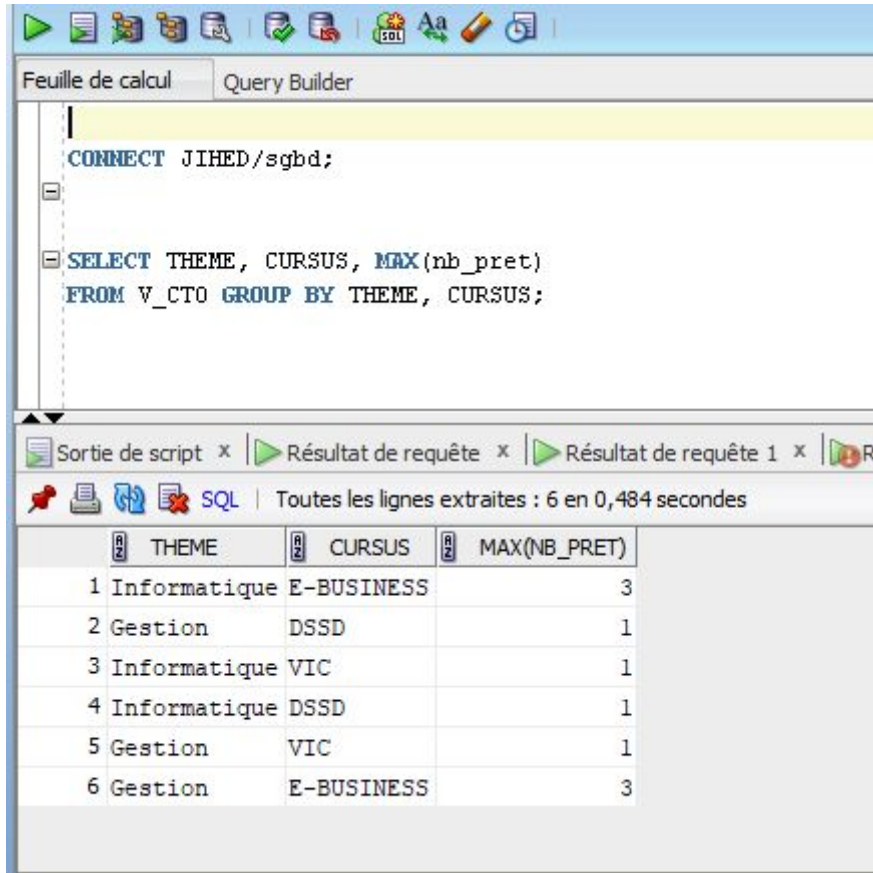
14-Connectez-vous sous JIHED et testez toutes les procédures et fonctions créées.

CONNECT JIHED/sghd;



15-Toujours sous JIHED, écrire une requête SQL qui affiche pour chaque cursus et chaque thème l'ouvrage ayant eu le plus grand nombre de prêts. utiliser la vue V_CTO.

```
SELECT CURSUS, THEME FROM V_CTO WHERE NOMBRE=MAX(nb_pret)
GROUP BY CURSUS,THEME;
```



The screenshot shows a software interface with a top toolbar, a 'Feuille de calcul' (Worksheet) tab, and a 'Query Builder' tab. The query builder contains the following SQL code:

```
CONNECT JIHED/sgbd;

SELECT THEME, CURSUS, MAX(nb_pret)
FROM V_CTO GROUP BY THEME, CURSUS;
```

Below the query builder, there is a results pane with tabs for 'Sortie de script', 'Résultat de requête', and 'Résultat de requête 1'. The 'Résultat de requête' tab is active, displaying a table with 6 rows and 3 columns: 'THEME', 'CURSUS', and 'MAX(NB_PRET)'. The status bar indicates 'Toutes les lignes extraites : 6 en 0,484 secondes'.

	THEME	CURSUS	MAX(NB_PRET)
1	Informatique	E-BUSINESS	3
2	Gestion	DSSD	1
3	Informatique	VIC	1
4	Informatique	DSSD	1
5	Gestion	VIC	1
6	Gestion	E-BUSINESS	3