Rendu final

Réaliseé par : Noura Aoujil

exercice1

creation des tableaux

Quick SQL

Clear | Settings | Download | Save as Script | Save Model

Shorthand

Samples | Generate SQL

```sql
CREATE TABLE PILOTE (
    NUMPIL INT PRIMARY KEY,
    NOMPIL VARCHAR(50),
    ADR VARCHAR(50),
    SAL INT
);

CREATE TABLE AVION (
    NUMAV INT PRIMARY KEY,
    NOMAV VARCHAR(50),
    CAPACITE INT,
    LOC VARCHAR(50)
);

CREATE TABLE VOL (
    NUMVOL INT PRIMARY KEY,
    NUMPIL INT,
    NUMAV INT,
```

Output

Copy to Worksheet

```oracle
-- create tables
create table create_table_pilote (
    numpil_int_primary_key          varchar2(4000),
    nompil                          varchar2(50),
    adr                             varchar2(50),
    sal                             integer
)
;

create table create_table_avion (
    numav_int_primary_key           varchar2(4000),
    nomav                           varchar2(50),
    capacite                        integer,
    loc                             varchar2(50)
)
;

create table create_table_vol (
```

exercice1

Insertion des Données

SQL Worksheet

Clear Find Actions Save Run

```sql
-- Insert into PILOTE
INSERT INTO PILOTE (NUMPIL, NOMPIL, ADR, SAL) VALUES (1, 'Jean Dupont', 'Er-Rachidia', 50000);
INSERT INTO PILOTE (NUMPIL, NOMPIL, ADR, SAL) VALUES (2, 'Alice Martin', 'Casablanca', 60000);
INSERT INTO PILOTE (NUMPIL, NOMPIL, ADR, SAL) VALUES (3, 'Marc Durand', 'Er-Rachidia', 55000);

-- Insert into AVION
INSERT INTO AVION (NUMAV, NOMAV, CAPACITE, LOC) VALUES (1, 'Airbus A320', 150, 'Casablanca');
INSERT INTO AVION (NUMAV, NOMAV, CAPACITE, LOC) VALUES (2, 'Boeing 737', 180, 'Er-Rachidia');
INSERT INTO AVION (NUMAV, NOMAV, CAPACITE, LOC) VALUES (3, 'Cessna 172', 4, 'Rabat');

-- Insert into VOL
INSERT INTO VOL (NUMVOL, NUMPIL, NUMAV, VILLE_DEP, VILLE_ARR, H_DEP, H_ARR) VALUES
(1, 1, 1, 'Er-Rachidia', 'Casablanca',
```

```
1 row(s) inserted.

1 row(s) inserted.
```

exercice1

1. Afficher les pilotes d’Er-Rachidia triés par salaire croissant :

SQL Worksheet

Clear Find Actions Save Run

```
DECLARE
    CURSOR c_pilotes IS
        SELECT * FROM PILOTE WHERE ADR = 'Er-Rachidia' ORDER BY SAL;
BEGIN
    FOR r_pilote IN c_pilotes LOOP
        DBMS_OUTPUT.PUT_LINE('NUMPIL: ' || r_pilote.NUMPIL || ', NOMPIL: ' || r_pilote.NOMPIL || ', SAL: ' || r_pilote.SAL)
    END LOOP;
END;
/
```

```
Statement processed.
NUMPIL: 1, NOMPIL: Jean Dupont, SAL: 50000
NUMPIL: 3, NOMPIL: Marc Durand, SAL: 55000
```

exercice1

Afficher les informations des pilotes en service :

SQL Worksheet

Clear Find Actions Save Run

```
DECLARE
    CURSOR c_pilotes_en_service IS
        SELECT DISTINCT p.*
        FROM PILOTE p
        JOIN VOL v ON p.NUMPIL = v.NUMPIL;

    r_pilote c_pilotes_en_service%ROWTYPE; -- Declare the record variable to hold the cursor's row data

BEGIN
    OPEN c_pilotes_en_service;
    LOOP
        FETCH c_pilotes_en_service INTO r_pilote;
        EXIT WHEN c_pilotes_en_service%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('NUMPIL: ' || r_pilote.NUMPIL || ', NOMPIL: ' || r_pilote.NOMPIL || ', SAL: ' || r_pilote.SAL);
    END LOOP;
    CLOSE c_pilotes_en_service;
END;
/
```

```
Statement processed.
```

exercice1

Afficher les informations des avions avec capacité ≥ 50 :

SQL Worksheet

Clear Find Actions Save Run

```sql
DECLARE
    v_avion AVION%ROWTYPE;
    e_avion_not_found EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_avion_not_found, -20001);
BEGIN
    BEGIN
        FOR r_avion IN (SELECT * FROM AVION WHERE CAPACITE >= 50) LOOP
            DBMS_OUTPUT.PUT_LINE('NUMAV: ' || r_avion.NUMAV || ', NOMAV: ' || r_avion.NOMAV || ', LOC: ' || r_avion.LOC);
        END LOOP;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_avion_not_found;
    END;
EXCEPTION
    WHEN e_avion_not_found THEN
        DBMS_OUTPUT.PUT_LINE('Aucun avion trouvé avec une capacité >= 50.');
END;
/
```

```
Statement processed.
NUMAV: 1, NOMAV: Airbus A320, LOC: Casablanca
NUMAV: 2, NOMAV: Boeing 737, LOC: Er-Rachidia
```

exercice1

4-Afficher les pilotes effectuant un vol au départ de leur ville de résidence :

SQL Worksheet

Clear Find Actions Save Run

```
DECLARE
    CURSOR c_pilotes_vol IS
        SELECT DISTINCT p.NUMPIL, p.NOMPIL
        FROM PILOTE p
        JOIN VOL v ON p.NUMPIL = v.NUMPIL
        WHERE p.ADR = v.VILLE_DEP;
BEGIN
    FOR r_pilote IN c_pilotes_vol LOOP
        DBMS_OUTPUT.PUT_LINE('NUMPIL: ' || r_pilote.NUMPIL || ', NOMPIL: ' || r_pilote.NOMPIL);
    END LOOP;
END;
/
```

Statement processed.

exercice1

5-Insérer un vol et gérer les exceptions :

SQL Worksheet

Clear Find Actions Save Run

```sql
CREATE OR REPLACE TRIGGER TRGR_UPDATE_TRACK_VOLS
AFTER INSERT ON VOL
FOR EACH ROW
BEGIN
    UPDATE TRACK_VOLS
    SET NB_VOLS = NB_VOLS + 1
    WHERE NUMPIL = :NEW.NUMPIL;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001, 'Error updating track_vols: ' || SQLERRM);
END;
/
DECLARE
    v_numvol INT := 101;
    v_numav INT := 1; -- Replace with the actual input value
    v_numpil INT := 1; -- Replace with the actual input value
    e_vol_exists EXCEPTION;
    e_pilote_not_found EXCEPTION;
    e_avion_not_found EXCEPTION;
    v_exists INT;
    v_pilot_exists INT;
```

```
Trigger created.
```

exercice1

5-Insérer un vol et gérer les exceptions :

SQL Worksheet

Clear Find Actions Save Run

```sql
BEGIN
    -- Vérification si le vol existe déjà
    BEGIN
        SELECT 1 INTO v_exists
        FROM VOL
        WHERE NUMPIL = v_numpil AND NUMAV = v_numav AND VILLE_DEP = 'Er-Rachidia' AND VILLE_ARR = 'Marrakech'
              AND H_DEP = TO_TIMESTAMP('2024-06-03 18:00:00', 'YYYY-MM-DD HH24:MI:SS')
              AND H_ARR = TO_TIMESTAMP('2024-06-03 19:10:00', 'YYYY-MM-DD HH24:MI:SS');
        RAISE e_vol_exists;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            NULL; -- Le vol n'existe pas encore, on peut continuer
    END;

    -- Vérification de l'existence du pilote
    BEGIN
        SELECT NUMPIL INTO v_pilot_exists FROM PILOTE WHERE NUMPIL = v_numpil;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_pilote_not_found;
    END;

```

```
Trigger created.
```

exercice1

5-Insérer un vol et gérer les exceptions :

SQL Worksheet

Clear Find Actions Save Run

```
    BEGIN
        SELECT NUMAV INTO v_avion_exists FROM AVION WHERE NUMAV = v_numav;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_avion_not_found;
    END;

    -- Insertion du vol
    INSERT INTO VOL (NUMVOL, NUMPIL, NUMAV, VILLE_DEP, VILLE_ARR, H_DEP, H_ARR) VALUES
    (v_numvol, v_numpil, v_numav, 'Er-Rachidia', 'Marrakech', TO_TIMESTAMP('2024-06-03 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-06-03 19:10:00', 'YYYY-MM-DD HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE('Vol inséré avec succès.');

EXCEPTION
    WHEN e_vol_exists THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : le vol existe déjà.');
    WHEN e_pilote_not_found THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : le numéro de pilote nexiste pas.');
    WHEN e_avion_not_found THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : le numéro davion nexiste pas.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erreur inconnue : ' || SQLERRM);
END;
```

exercice1

5-Insérer un vol et gérer les exceptions :

```
Trigger created.


Statement processed.
Vol inséré avec succès.
```

exercice2

Procédure profilePilote :

SQL Worksheet

Clear Find Actions Save Run

```
CREATE OR REPLACE PROCEDURE profilePilote(p_numpil IN INT) IS
    v_pilote PILOTE%ROWTYPE;
    e_pilote_not_found EXCEPTION;
BEGIN
    BEGIN
        SELECT * INTO v_pilote FROM PILOTE WHERE NUMPIL = p_numpil;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE e_pilote_not_found;
    END;

    DBMS_OUTPUT.PUT_LINE('NUMPIL: ' || v_pilote.NUMPIL || ', NOMPIL: ' || v_pilote.NOMPIL || ', ADR: ' || v_pilote.ADR || ', SAL: ' || v_pilote.SAL);

EXCEPTION
    WHEN e_pilote_not_found THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : le pilote nexiste pas.');
```

```
Procedure created.
```

exercice2

Fonction pilotesErrLibres :

SQL Worksheet

Clear Find Actions Save Run

```sql
CREATE OR REPLACE FUNCTION pilotesErrLibres RETURN INT IS
    v_count INT;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM PILOTE p
    WHERE p.ADR = 'Er-Rachidia' AND p.NUMPIL NOT IN (SELECT NUMPIL FROM VOL);
    RETURN v_count;
END;
/
```

```
Function created.
```

exercice2

Déclencheur trgr_update_track_vols :

SQL Worksheet

Clear Find Actions Save Run

```sql
CREATE OR REPLACE FUNCTION pilotesErrLibres RETURN INT IS
    v_count INT;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM PILOTE p
    WHERE p.ADR = 'Er-Rachidia' AND p.NUMPIL NOT IN (SELECT NUMPIL FROM VOL);
    RETURN v_count;
END;
/
```

Function created.

exercice2

4-Fonction tab_pilotes :

SQL Worksheet

Clear Find Actions Save Run

```sql
CREATE OR REPLACE FUNCTION tab_pilotes(fp_sal IN INT) RETURN SYS.ODCINUMBERLIST IS
    v_pilote_tab SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST();
    v_index INT := 0;
    CURSOR c_pilotes(p_salaire INT) IS
        SELECT NUMPIL FROM PILOTE WHERE SAL >= p_salaire;
BEGIN
    FOR r_pilote IN c_pilotes(fp_sal) LOOP
        v_index := v_index + 1;
        v_pilote_tab.EXTEND;
        v_pilote_tab(v_index) := r_pilote.NUMPIL;
    END LOOP;
    RETURN v_pilote_tab;
END;
/
```

Function created.