

Wikicycle

Sommaire:

1. Introduction
2. Cahier des charges
3. Modèle Entités/Associations
4. Modèle relationnel
5. Questions SQL
6. Questions SQL (Réponses)
7. Vues
8. Triggers
9. Meta Données
10. Insertion des Données

1.Introduction

Nous avons choisi comme sujet de base de données une encyclopédie pour recenser et organiser les informations sur les courses cyclistes, les équipes, les coureurs et les grands événements cyclistes.

Les cyclistes sont identifiés par un identifiant, mais ont aussi un nom, prénom et leur âge. Ils appartiennent à une équipe mais peuvent être transférés dans une nouvelle équipe.

Les équipes sont identifiées par un identifiant et possèdent un budget. Les équipes encore d'actualités sont composées des coureurs mais on peut également trouver les anciens cyclistes ayant appartenu à l'équipe. Une équipe abandonnée n'a plus de coureurs à son actif mais l'historique est toujours disponible.

Une course comme Paris-Roubaix est une épreuve d'un jour indépendante d'une course qu'on retrouve dans un événement.

Les événements comme le Giro d'Italie ou le Tour de France sont des courses à étapes, donc plusieurs courses composent ces grands événements.

Un titre est un succès décerné à un coureur pour une performance à la fin d'un événement ("meilleur sprinteur ou meilleur grimpeur pour le Tour de France 2022). Le titre de la combativité peut être attribué pour un coureur attaquant.

Les titres concernant les vainqueurs d'un événement sont attribués à la fin de celui-ci

2. Cahier des charges

Cycliste :

- Un cycliste doit appartenir à une équipe durant la même année que l'événement/course auquel il veut participer.
- L'équipe doit également s'inscrire pour l'événement pour que les cyclistes de son équipe puissent participer et s'inscrire aux courses
- Un cycliste ne peut appartenir à deux équipes en même temps

Équipe :

- Une équipe doit faire participer au minimum 4 coureurs pour pouvoir s'inscrire, si l'équipe passe en dessous de la barre des 3 coureurs, elle est alors disqualifiée et les temps des cyclistes ne seront pas comptabilisés lors du classement final, les cyclistes disqualifiés doivent être entrés en premier (sinon trigger panic)
- Une équipe sans coureurs pendant 1 an est considérée comme abandonnée et ne peut plus récupérer de nouveaux coureurs

Titre :

- Un titre ne peut être attribué à un coureur disqualifié.
- Un titre ne peut être attribué qu'à un seul coureur (impossible d'avoir 2 meilleurs sprinteurs pour un événement)

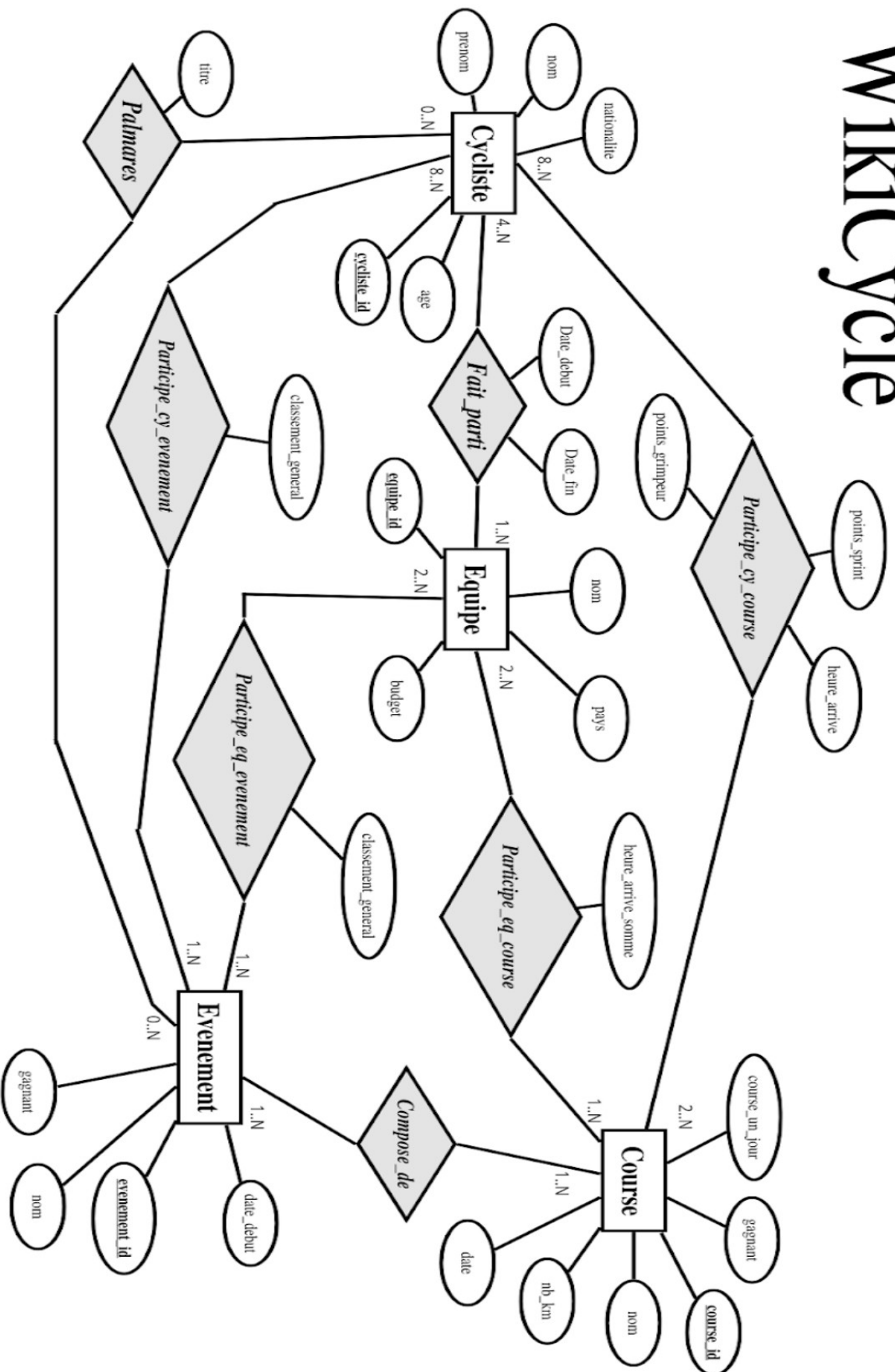
Événement :

- Le classement général se calcule de la façon suivante : les temps additionnés personnels de chaque coureur sont additionnés puis classés dans l'ordre croissant
- Pour le classement par équipe, les temps des 3 meilleurs cyclistes d'une équipe sont additionnés et classés également dans l'ordre croissant.
- Un coureur n'ayant pas participé à toutes les courses (blessure/abandon/équipe disqualifiée) est relégué à la fin du classement de l'événement.
- Les points "sprint" et "grimpeur" permettent de choisir le gagnant du meilleur sprinteur/grimpeur de cet événement
- Le titre de combativité est attribué de façon objective à un coureur selon ses performances (souvent dans le top 5), il doit également avoir fait la totalité de l'événement
- Le titre du vainqueur de l'événement est attribué à la suite du calcul du classement général au premier de celui-ci

Course :

- Si pendant une course hors événement une équipe passe en dessous des 3 coureurs, elle sera disqualifiée pour le classement par équipe mais les coureurs restants garderont leur temps. Une course peut se disputer seule mais ne peut pas alors appartenir à un événement.
- Le temps des cyclistes est trié dans l'ordre croissant pour établir le classement
- Une course d'un jour ne peut pas servir pour une étape d'un événement
- Une course doit être parcourue par 2 équipes minimums donc 8 coureurs minimums. (Les temps seront annulés pour tout le monde est aucun gagnant ne sera choisi).

3. Modèle Entités/Associations



4.Modèle relationnel

Equipe :

- Equipe_id
- Budget
- Pays
- Nom

Cycliste :

- Cycliste_id
- Nom
- Prenom
- Age
- Nationalite

Course :

- Course_id
- Nom
- Date
- Gagnant
- Nb_km

Evénement :

- Evenement_id
- Nom
- Date_debut
- Gagnant

Compose_de:

- Compose_id
- Evenement_id
- Course_id

Participe_eq_evenement:

- Parti_eq_id
- Equipe_id
- Evenement_id
- classement_general

Participe_eq_course:

- Parti_eq_id
- equipe_id
- course_id
- heure_arrive_somme

Participe_cy_evenement:

- parti_cy_id
- cycliste_id
- evenement_id
- classement général

Participe_cy_course:

- parti_cy_id
- cycliste_id
- course_id
- points_sprint
- points_grimpeur
- heure_arrive

Palmares:

- Palmares_id
- titre
- cycliste_id
- evenement_id

Fait_Parti:

- Fait_parti_id
- Date_debut
- Date_fin
- Cycliste_id
- Equipe_id

Clés étrangères (Les clés primaires sont soulignées):

compose_de.evenement_id REFERENCE evenement.evenement_id

compose_de.course_id REFERENCE course.course_id

participe_eq_evenement.equipe_id REFERENCE Equipe.equipe_id

participe_eq_evenement.evenement_id REFERENCE Evenement.evenement_id

participe_eq_course.equipe_id REFERENCE equipe.equipe_id

participe_eq_course.course_id REFERENCE course.course_id

participe_cy_evenement.cycliste_id REFERENCE cycliste.cycliste_id

participe_cy_evenement.evenement_id REFERENCE evenement.evenement_id

participe_cy_course.course_id REFERENCE course.course_id

participe_cy_cycliste.cycliste_id REFERENCE cycliste.cycliste_id

palmares.titre_id REFERENCE titre.titre_id

palmares.cycliste_id REFERENCE cycliste.cycliste_id

participe.course_id REFERENCE course.course_id

fait_parti.cycliste_id REFERENCE cycliste.cycliste_id

fait_parti.equipe_id REFERENCE equipe.equipe_id

course.gagnant REFERENCE cycliste.cycliste_id

evenement.gagnant REFERENCE cycliste.cycliste_id

5. Questions SQL

- 1) Cyclistes ayant remporté le plus de courses.
- 2) Liste des équipes avec leur nombre de cyclistes pour l'année en cours.
- 3) Somme de tous les coureurs ayant réalisé un événement précis (Toutes les éditions du Tour de France).
- 4) Équipes ayant fait remporter le plus de titres à leurs cyclistes.
- 5) Budget moyen des équipes qui remportent un événement.
- 6) Moyenne des âges des cyclistes lorsqu'ils gagnent un titre.
- 7) Temps moyen en heure pour finir une course. Celle de 'Reims - Châlons-en-Champagne' par exemple.
- 8) Cycliste avec le plus de victoires d'un même événement. Pour le Giro d'Italia.
- 9) Cycliste ayant participé à tous les événements de l'année 2016.
- 10) Moyenne du nombre de cyclistes par course.
- 11) Équipe qui a effectué le plus de courses.
- 12) Pour un cycliste donné, trouver l'équipe avec laquelle il a remporté le plus de victoires.
Inverse de la 4)

6. Questions SQL (Réponses)

1) Cyclistes ayant remporté le plus de courses.

```
SELECT cyc.cycliste_id, cyc.nom, cyc.prenom, COUNT(*) as nb_victoires
FROM cycliste cyc, course cou
WHERE cyc.cycliste_id = cou.gagnant
GROUP BY c.cycliste_id
ORDER BY nb_victoires DESC
HAVING COUNT(*) < 0;
```

2) Liste des équipes avec leur nombre de cyclistes pour l'année en cours.

```
SELECT equ_id, equ.nom, COUNT(*)
FROM equipe equi, fait_parti fap
WHERE equi.equipe_id = fap_equipe_id AND '2015-01-02' BETWEEN fap.date_debut
AND fap.date_fin
GROUP BY equi.equipe_id
HAVING COUNT(*) < 0;
```

3) Somme de tous les coureurs ayant réalisé un événement précis (Toutes les éditions du Tour de France).

```
SELECT eve.nom as 'Tour de France', COUNT(*) as cyclistes_participants
FROM Cycliste cycl, participe_cy_evenement pacc, evenement even
WHERE cycl.cycliste_id = pacc.cycliste_id, pacc.evenement_id = even.evenement_id AND
even.nom LIKE 'Tour de France %'
```

4) Équipes ayant fait remporté le plus de titres à leurs cyclistes.

```
SELECT equi.equipe_id, equi.nom, equi.budget, COUNT(*) as 'titre_parmis_cette_equipe'
FROM equipe equi, fait_parti fap, cycliste cycl, palmares palm, evenement even
WHERE equi.equipe_id = fap.equipe_id, cycl.cycliste_id = fap.cycliste_id, cycl.cycliste_id =
palm.cycliste_id, palm.evenement_id = even.evenement_id AND even.date_debut
BETWEEN fap.date_debut AND fap.date_fin
GROUP BY equi.equipe_id
ORDER BY titre_parmis_cette_equipe;
```

5) Budget moyen des équipes qui remportent un événement:

```
SELECT AVG(ecu.budget) as budget_moyen  
FROM ( SELECT MIN(classement_general), ecu.budget  
        FROM equipe equi, participe_eq_evenement pae, evenement even  
        WHERE equi.equipe_id = pae.classement_general, pae.evenement_id =  
        even.evenement_id  
        GROUP BY pae.evenement_id  
      );
```

6) Moyenne des âges des cyclistes lorsqu'ils gagnent un titre.

```
SELECT AVG( EXTRACT(YEAR FROM even.date_debut) - EXTRACT( YEAR FROM  
SYSDATE) - cycl.age) as age_moyen_titre  
FROM cycliste cycl, palmares palm, evenement even  
WHERE cycl.cycliste_id = palm.cycl_id, palm.evenement_id = even.evenement_id;
```

7) Temps moyen en heure pour finir une course. Celle de 'Reims - Châlons-en-Champagne'
par exemple:

```
SELECT AVG( SUM(EXTRACT(HOUR FROM classement)) + SUM(EXTRACT(MINUTE  
FROM classement) / 60 + SUM(EXTRACT(SECOND FROM classement / 3600)  
FROM Course cour, participe_cy_course pacc  
WHERE cour.course_id = pacc.course_id AND cour.nom LIKE 'Reims - Châlons-en-  
Champagne';
```

8) Cycliste avec le plus de victoires d'un même événement. Pour le Giro d'Italia:

```
SELECT cycl.cycliste_id, cycl.nom, cycl.prenom, COUNT(*)  
FROM cycliste cycl, evenement even  
WHERE cycl.cycliste_id = even.cycliste_id AND even.nom LIKE 'Giro d'Italia%'  
GROUP BY cycl.cycliste_id;
```


9) Cycliste ayant participé à tous les événements de l'année 2016:

```
SELECT cycl.cycliste_id, cycl.nom, cycl.prenom
FROM cycliste cycl, participe_cy_evenement pacc, evenement even
WHERE cycl.cycliste_id = pacc.cycliste_id, pacc.evenement_id = even.evenement_id AND
EXTRACT(YEAR FROM even.date_debut) = 2016
GROUP BY cycl.cycliste_id
HAVING COUNT(*) = ( SELECT COUNT(*)
                     FROM evenement
                     WHERE EXTRACT(YEAR FROM evenement.date_debut) = 2016
);
```

10) Moyenne du nombre de cyclistes par course.

```
SELECT AVG(nb_cyclistes) as nb_cyclistes_moyen
FROM ( SELECT cour.course_id, COUNT(*) as nb_cyclistes
      FROM course cour, participe_cy_course pacc
      WHERE cour.course_id = pacc.course_id
      GROUP BY cour.course_id
);
```

11) Équipe qui a effectué le plus de courses.

```
SELECT COUNT(*) as nb_courses
```

```
FROM Participe_eq_course
```

```
WHERE equipe_id = 1;
```

```
SELECT MAX(nb_courses)
```

```
FROM (SELECT
```

```
    FROM equi, participe_eq_course paec
```

```
SELECT equi.equipe_id
```

```
FROM equi, participe_eq_course paec
```

```
WHERE equi.equipe_id = paec.equipe_id
```

```
GROUP BY equi.equipe_id
```

```
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
```

```
    FROM participe_eq_course partiec
```

```
    GROUP BY partiec.equipe_id
```

```
);
```

12) Pour un cycliste donné, trouver l'équipe avec laquelle il a remporté le plus de victoires.
Inverse de la 4)

```
SELECT equi.equipe_id, equi.nom, COUNT(*) as nb_titre_avec_cette_equipe
```

```
FROM equipe equi, fait_parti fap, cycliste cycl, palmares palm, evenement even
```

```
WHERE equi.equipe_id = fap.equipe_id, fap_cycliste_id = cycl.cycliste_id, cycl.cycliste_id  
= palm.cycliste_id, palm.evenement_id = even.evenement_id AND even.date_debut  
BETWEEN fap.date_but AND fap.date_fin
```

```
ORDER BY nb_titre_avec_cette_equipe DESC;
```

7. VUES

```
CREATE USER cycliste IDENTIFIED BY cycliste
CREATE USER manager IDENTIFIED BY manager
CREATE USER organisateur IDENTIFIED BY organisateur
```

1 : Liste des cyclistes avec leur palmares

```
CREATE OR REPLACE VIEW vue_palmares AS
SELECT cycl.cycliste_id, palm.titre
FROM cycliste cycl, palmares palm
WHERE cycl.cycliste_id = palm.cycliste_id;
```

Spectateur/: vues sur cyclistes et leur palmares

cyclistes: select sur table participe_cy_course et participe_cy_evenement correspond

Manager: select sur table équipe, participe_equipe_ev et participe_ev_course

Organisateur: select sur table course, événement et palmares

```
GRANT SELECT ON vue_palmares TO cycliste;
```

2 : Historique des équipes des cyclistes français

```
CREATE OR REPLACE VIEW historique_equipes_cyclistes_fr AS
SELECT cycl.cycliste_id, equi.nom
FROM cycliste cycl, fait_parti, fap, equipe equi
WHERE cycl.cycliste_id = fap.cycliste_id, equi.equipe_id = fap.cycliste_id AND equi.pays
LIKE 'France';
```

```
GRANT SELECT historique_equipes_cyclistes_fr ON TO cycliste;
```

```
GRANT SELECT historique_equipes_cyclistes_fr ON TO organisateur;
```

3: Records: Liste des courses avec son nb de km ainsi que le meilleur temps.

```
CREATE VIEW vue_courses AS
SELECT cour.course_id, MIN(pacc.heure_arrive)
FROM course cour, participe_cy_course pcc
WHERE cour.course_id = pcc.course_id,
GROUP BY cour.course_id;
GRANT SELECT vue_course ON TO organisateur;
```

4 :KM parcourus par chaque équipe en 2018:

```
CREATE VIEW vue_distance_année_equipes AS

SELECT equi.nom, SUM(cour.nb_km)
FROM equipe equi, participe_eq_course pae, course cour
WHERE equi.equipe_id = pae.equipe_id, pae.course_id = cour.course_id and
EXTRACT(YEAR FROM cour.datec = 2018);
GROUP BY equi.nom

GRANT SELECT vue_distance_année_equipes ON TO manager;
```

5: Liste des équipes leur nombres d'adhérents depuis leur création

```
CREATE VIEW vue_adherants_equipes AS
SELECT equi.nom, COUNT(*) as nb_adherants_depuis_creation
FROM equipe equi, fait_parti fap, cycliste cycl
WHERE equi.equipe_id = fap.equipe_id, cycl.cycliste_id = fap.cycliste_id
GROUP BY equi.nom

GRANT SELECT vue_adherants_equipes ON TO manager;
```

8. Triggers

Pour l'insertion des classements des cyclistes et équipes, il faut update le tuple concerné pour activer le trigger qui recalcule le temps pour le classement (général ou non) du cycliste/équipe

-Un cycliste doit appartenir à une équipe durant la même année que la course à laquelle il veut participer ET l'équipe à laquelle il appartient doit également être inscrite

```
CREATE OR REPLACE TRIGGER inscription_course_cycliste
BEFORE INSERT ON Participe_cy_course FOR EACH ROW

DECLARE
temp_date_ev DATE;
check_date INT;
check_equipe_inscrite INT;

BEGIN
SELECT date INTO temp_date
FROM course
WHERE course_id = :new.course_id;

SELECT COUNT(*) INTO check_date
FROM fait_parti
```

```
WHERE cycliste_id = :new.cycliste_id AND temp_date BETWEEN date_debut AND date_fin;
```

```
IF check_date = 0 THEN  
    RAISE_APPLICATION_ERROR(-20001, 'Le cycliste n'a pas d'équipe pour  
    participer à cette course);  
END IF;
```

```
SELECT count(*) INTO check_equipe_inscrite  
FROM fait_parti faip, equipe equ, participe_eq_course parec  
WHERE faip.cycliste_id = :new.cycliste_id, faip.equipe_id = equ.equipe_id, equ.equipe_id =  
parec.equipe_id AND parec.course_id = :new.course_id;
```

```
IF check_equipe_inscrite = 0 THEN  
    RAISE_APPLICATION_ERROR(-20002, 'L'équipe du cycliste n'est pas inscrite à la  
    course);  
END IF;  
END;
```

-Un cycliste doit appartenir à une équipe durant la même année que l'événement auquel il veut participer ET l'équipe à laquelle il appartient doit également être inscrite

```
CREATE or REPLACE TRIGGER inscription_evenement_cycliste  
BEFORE INSERT ON Participe_cy_evenement FOR EACH ROW
```

```
DECLARE  
temp_date_ev DATE;  
check_date INT;  
check_equipe_inscrite INT;
```

```
BEGIN  
SELECT date into temp_date  
FROM evenement  
WHERE evenement_id = :new.evenement_id
```

```
SELECT COUNT(*) INTO check_date  
FROM fait_parti  
WHERE cycliste_id = :new.cycliste_id AND temp_date BETWEEN date_debut AND date_fin;
```

```
IF check_date = 0 THEN  
    RAISE_APPLICATION_ERROR(-20003, 'Le cycliste n'a pas d'équipe pour  
    participer à cet événement);  
END IF;
```

```
SELECT COUNT(*) INTO check_equipe_inscrite  
FROM fait_parti faip, equipe equ, participe_eq_evenement paree
```

```
WHERE faip.cycliste_id = :new.cycliste_id, faip.equipe_id = equ.equipe_id, equ.equipe_id =  
paree.equipe_id AND patee.evenement_id = :new.evenement_id;
```

```
IF check_equipe_inscrite = 0 THEN  
    RAISE_APPLICATION_ERROR(-20004, 'L'équipe du cycliste n'est pas inscrite à  
    l'événement);  
END IF;  
END;
```

-Un cycliste ne peut appartenir à deux équipes en même temps (vérifie qu'il n'existe pas déjà une inscription pour le cycliste concerné avec des dates qui chevauchent celles de la nouvelle inscription).
De plus, une équipe sans coureurs pendant 1 an est considérée comme abandonnée et ne peut plus récupérer de nouveaux coureurs.

```
CREATE TRIGGER conflict_team  
BEFORE INSERT on fait_parti FOR EACH ROW
```

```
DECLARE  
date_fin_avant_insertion DATE;  
verif INT;
```

```
BEGIN
```

```
SELECT MAX(fp.date_fin) INTO date_fin_avant_insertion  
FROM fait_parti faip  
WHERE faip.equipe_id = :new.equipe_id;  
IF date_fin_avant_insertion < :new.date_debut - INTERVAL 1 YEAR THEN  
    RAISE_APPLICATION_ERROR(-20006, 'Cette équipe a été abandonnée et  
    ne peut plus récupérer de coureurs.');
```

```
END IF;
```

```
SELECT COUNT(*) INTO verif  
FROM fait_parti  
WHERE cycliste_id = :new.cycliste_id AND ( :new.date_debut BETWEEN  
date_debut AND date_fin OR :new.date_fin BETWEEN date_debut AND date_fin)
```

```
IF verif THEN  
    RAISE_ERROR_APPLICATION(-20005, 'Les dates inscrites entrent en  
    conflit avec l'historique du cycliste')  
END IF;  
END;
```

-Un titre ne peut être attribué qu'à un seul coureur (impossible d'avoir 2 meilleurs sprinteurs pour un même événement)

```
CREATE OR REPLACE TRIGGER titre_conflict
BEFORE INSERT OR UPDATE ON palmares FOR EACH ROW

BEGIN
IF (SELECT COUNT(*)
    FROM palmares pal
    WHERE titre LIKE CONCAT(:new.titre, '%') AND pal.evenement_id = :new.evenement_id;
) THEN
    RAISE_ERROR_APPLICATION(-20007, 'Titre déjà attribué à un coureur');
END IF;
END;
```

-Si moins de 3 coureurs dans l'équipe, disqualifié le coureur lors de l'insertion de son temps

```
CREATE OR REPLACE TRIGGER disqualify_cyclist
BEFORE UPDATE ON participe_cy_course FOR EACH ROW

DECLARE
temp_date DATE,
compt_cy INT,

BEGIN
SELECT date_c into temp_date
FROM course
WHERE course_id = :new.course_id;

SELECT COUNT(*) INTO compt_cy
FROM cycliste cyc, fait_parti fap, equipe equ
WHERE cyc.cycliste_id = fap.cycliste_id and fap.equipe_id = equ.equipe_id and fap.classement IS
NOT NULL AND equ.equipe_id = (SELECT equipe_id
                                FROM equipe, fait_parti, cycliste
                                WHERE equipe.equipe_id = fait_parti.equipe_id and
                                fait_parti.cycliste_id = cycliste.cycliste_id
                                AND temp_date BETWEEN fait_parti.date_debut and
                                fait_parti.date_fin);

IF compt_cy < 4 THEN
    :new.classement = NULL;
    DBMS_OUTPUT.put_line('Cycliste ' || :new.cycliste_id || ' disqualifié. ');
END IF;
END;
```

-Vérifie si bon gagnant est inséré dans la table Événement

```
CREATE OR REPLACE TRIGGER winner_event  
BEFORE UPDATE on Evenement FOR EACH ROW
```

```
DECLARE
```

```
temp_winner int;
```

```
BEGIN
```

```
IF :new.cycliste_id != (SELECT parcc.cycliste_id into temp_winner  
                        FROM Participe_cy_course parcc, Course cou, Compose_de comd  
                        WHERE comd.evenement_id = :old.evenement_id, comd.course_id =  
                        cou.course_id, cou.course_id = parcc.course_id  
                        GROUP BY parcc.cycliste_id  
                        HAVING SUM(parcc.classement) = (  
                                SELECT( MIN(SUM(particc.classement))  
                                FROM Participe_cy_course particc, course cour,  
                                compose_de compd  
                                WHERE compd.course_id = old.evenement_id,  
                                compd.course_id = cour.course_id,  
                                cour.course_id = particc.course_id  
                                GROUP BY particc.cycliste_id  
                                ));
```

```
THEN
```

```
    :new.cycliste_id := temp_winner;
```

```
    DMBS_OUTPUT.put_line('Nouveau gagnant changé et enregistré');
```

```
END IF;
```

```
END;
```

-Vérifie si bon gagnant attribué dans la table course

```
CREATE OR REPLACE TRIGGER winner_race  
BEFORE UPDATE on Course FOR EACH ROW
```

```
DECLARE
```

```
temp_winner int;
```

```
BEGIN
```

```
IF :new.cycliste_id != (SELECT parcc.cycliste_id into temp_winner  
                        FROM Participe_cy_course parcc, Course cou  
                        WHERE cou.course_id = old.course_id AND  
                        parcc.classement = ( SELECT MIN(particc.classement)  
                        FROM Participe_cy_course partcc  
                        WHERE partcc.course_id = :old.course_id)
```

```
THEN
```

```
    :new.gagnant := temp_winner;
```

```
    DMBS_OUTPUT.put_line('Gagnant rectifié et inséré');
```

```
END IF;
```

```
END;
```


-Impossibilité de donner un titre à un coureur qui a été disqualifié (sert aussi pour check gagnant evenement, meilleur sprinteur/grimpeur).

De plus, impossible de donner un titre déjà attribué pour le même événement

```
CREATE OR REPLACE TRIGGER title_check
BEFORE UPDATE on Palmares FOR EACH ROW
```

```
DECLARE
actual_winner int;
```

```
BEGIN
IF ( (SELECT COUNT(*)
      FROM course cou, compose_de comd, participe_cy_course parcc
      WHERE comd.evenement_id = :new.evenement_id, cou.course_id =
      comd.course_id, parcc.cycliste_id = :new.cycliste_id AND parcc.classement
      NOT LIKE NULL) !=
      (SELECT COUNT(*)
      FROM course cour, compose_de com_de
      WHERE com_de.evenement_id = :new.evenement_id AND
      cour.course_id = com_de.course_id) )
THEN
      RAISE_APPLICATION_ERROR(-20008, 'Le coureur a été disqualifié,
      impossible de lui donner un titre')
END IF;
```

```
IF (SELECT COUNT(*)
    FROM palmares pal
    WHERE titre LIKE CONCAT(:new.titre, '%') AND pal.evenement_id = :new.evenement_id;
) THEN
      RAISE_ERROR_APPLICATION(-20007, 'Titre déjà attribué à un coureur');
END IF;
```

```
IF :new.titre LIKE 'gagnant %' THEN
      IF :new.titre != ( SELECT eve.gagnant into actual_winner
                        FROM Evenement eve
                        WHERE eve.evenement_id = old.evenement_id)
      THEN
            :new.titre := actual_winner;
            DMBS_OUTPUT.put_line(Titre affecté et inséré à l'actuel vainqueur de
            l'événement');
      END IF;
```

```
IF :new.titre LIKE 'meilleur sprinteur %' AND
      :new.cycliste_id != (SELECT pcc.cycliste_id into actual_winner
                        FROM compose_de comd, course cou, Participe_cy_course
                        parcc
```

```

WHERE comd.evenement_id = :new.evenement_id,
comd.course_id = cou.course_id, cou.course_id =
parcc.course_id
GROUP BY parcc.cycliste_id
HAVING SUM(points_sprint) =
        (SELECT MAX(SUM(points_sprint)
        FROM Participe_cy_course particc,
        course cour, compose_de compod
        WHERE compo.evenement_id =
        :new.evenement_id, compod.course_id =
        cour.course_id, cour.course_id =
        particc.course_id
        GROUP BY particc.cycliste_id
        );

THEN
    :new.cycliste_id = actual_winner;
    DMBS_OUTPUT.put_line('Meilleur sprinteur corrigé et inséré pour l'événement
    correspondant');

END IF;

IF :new.titre LIKE 'meilleur grimpeur%' AND
    :new.cycliste_id != (SELECT pcc.cycliste_id into actual_winner
        FROM compose_de comd, course cou, Participe_cy_course
        parcc
        WHERE comd.evenement_id = :new.evenement_id,
        comd.course_id = cou.course_id, cou.course_id =
        parcc.course_id
        GROUP BY parcc.cycliste_id
        HAVING SUM(points_grimpeur) =
            (SELECT MAX(SUM(points_grimpeur)
            FROM Participe_cy_course particc,
            course cour, compose_de compod
            WHERE compo.evenement_id =
            :new.evenement_id, compod.course_id =
            cour.course_id, cour.course_id =
            particc.course_id
            GROUP BY particc.cycliste_id
            );

THEN
    :new.cycliste_id = actual_winner;
    DMBS_OUTPUT.put_line('Meilleur grimpeur corrigé et inséré pour l'événement
    correspondant');

END IF;
END;

```

-Pas possible de rajouter une course d'un jour dans un événement

```
CREATE OR REPLACE TRIGGER check_one_day_race  
BEFORE INSERT ON Compose_de
```

```
BEGIN
```

```
IF (SELECT course_un_jour  
    FROM course  
    WHERE course_id = :new.course_id) THEN
```

```
    RAISE_ERROR_APPLICATION(-20009, 'Impossible de rajouter une course d'un  
    jour à un événement');
```

```
END IF;
```

```
END
```

- Il faut 2 équipes et 8 cyclistes minimum ainsi que 4 cyclistes par équipes pour ajouter le résultat à un événement, le trigger recalcule le temps de ses 3 meilleurs cyclistes qui servira pour le classement

```
CREATE OR REPLACE TRIGGER team_race_result  
BEFORE ALTER ON Participe_eq_course ON EACH ROW
```

```
DECLARE
```

```
temps_total DATE
```

```
BEGIN
```

```
IF ( ( (SELECT COUNT(*)  
    FROM Participe_eq_course parec  
    WHERE parec.course_id = :new.course_id) < 2) OR  
    ( (SELECT COUNT(*)  
    FROM Participe_cy_course parcc  
    WHERE parcc.course_id = :new.course_id) < 8) ) )
```

```
THEN
```

```
    :new.classement_general = NULL;
```

```
    DBMS_OUTPUT.put_line('Pas assez d'équipe ou de coureurs, temps  
    annulé.');
```

```
END IF;
```

```
SELECT SUM(parcc.classement) into temps_total
```

```
FROM Participe_cy_course parcc, course cou
```

```
WHERE parcc.course_id = :new.course_id, parcc.course_id = cou.course_id AND  
parcc.equipe_id = :old.equipe_id
```

```
ORDER BY parcc.classement
```

```
FETCH FIRST 3 ROWS ONLY;
```

```
:new.classement_general := temps_total
```

```
END;
```

-Vérifie que l'équipe est éligible pour le classement général d'un événement, recalcule également le temps de ses 3 meilleurs cycliste sur l'ensemble des courses de l'événement

```
CREATE OR REPLACE TRIGGER team_event_result
BEFORE UPDATE ON Participe_eq_evenement FOR EACH ROW

DECLARE
actual_sum_time DATE;

BEGIN
IF ( (SELECT COUNT(*)
      FROM Participe_eq_course parec, Course cou, Compose_de comd
      WHERE parec.equipe_id = :old.equipe_id, comd.evenement_id = :old.evenement_id,
      comd.course_id = cou.course_id, parec.course_id = cou.course_id AND
      parec.classement IS NOT NULL) !=
      (SELECT COUNT(*)
      FROM Compose_de comd
      WHERE comd.evenement_id = :old.evenement_id)
THEN
    :new.classement_general := NULL;
    DBMS_OUTPUT.put_line('Equipe disqualifiée car disqualifiée dans une des courses
    de l'événement);
END IF;

actual_sum_time := (SELECT SUM(parec.classement)
                    FROM Participe_eq_course, course cou, compose_de comd
                    WHERE parec.course_id = cou.course_id, cou.course_id =
                    comd.course_id AND comd.evenement_id = old.evenement.id);

END;
```

-Vérifie que le temps d'un cycliste est éligible pour le classement général(si il a réalisé toute les courses, recalcule également son temps général sur l'événement

```
CREATE OR REPLACE TRIGGER cyclist_event_result
BEFORE UPDATE ON Participe_cy_evenement FOR EACH ROW

DECLARE
actual_sum_time TIMESTAMP

BEGIN
IF ( (SELECT COUNT(*)
      FROM Participe_cy_course parcc, Course cou, Compose_de comd
      WHERE comd.evenement_id = :old.evenement_id, comd.course_id = cou.course_id,
      cou.course_id = parcc.course_id, parcc.cycliste_id = old.cycliste_id AND parcc.classement
      IS NOT NULL) != ( SELECT COUNT(*)
      FROM Compose_de comd
```

```

WHERE comd.evenement_id = :old.evenement_id)
THEN
    :new.classement_general := NULL;
    DBMS_OUTPUT.put_line('Coureur disqualifié car il n'a pas effectué toute les
courses');

END IF;

actual_sum_tim := ( SELECT SUM(parcc.classement)
                     FROM Participe_cy_course parcc, Course cou, Compose_de comd
                     WHERE parcc.course_id = cou.course_id, cou.course_id =
                     comd.course_id, comd.evenement_id = old.evenement_id AND
                     parcc.cycliste_id = old.cycliste_id );

END;
```

9. META DONNEES

```

(SELECT * FROM dba_constraints) as liste_ora_constraints
SELECT table_name,constraint_name, constraint_type
FROM dba_constraints
ORDER BY table_name, constraint_type.
```

```

SELECT table_owner, trigger_name, trigger_type as liste_ora_triggers
FROM dba_triggers
ORDER BY table_owner
```

```

SELECT CONSTRAINTS_NAME, CONSTRAIN_TYPE
FROM DBA_constraints
WHERE DBA_Constrants LIKE 'CHECK'
```

10.INSERTION DES DONNEES

```

CREATE TABLE Equipe(
equipe_id int PRIMARY KEY,
budget NUMBER(10),
pays varchar(30),
nom varchar(30)
);
```

```

CREATE TABLE Cycliste(
cycliste_id int PRIMARY KEY,
nom varchar(30),
prenom varchar(30),
age NUMBER(2),
```

```
nationalite varchar(30),  
CHECK (age >= 18)  
);
```

```
CREATE TABLE Course(  
course_id int PRIMARY KEY,  
nom varchar(50),  
dateC TIMESTAMP,  
gagnant int,  
nb_km NUMBER(4,1),  
course_un_jour NUMBER(1),  
FOREIGN KEY(gagnant) REFERENCES Cycliste(cycliste_id)  
);
```

```
CREATE TABLE Evenement(  
evenement_id int PRIMARY KEY,  
nom varchar(30),  
date_debut date,  
gagnant int,  
FOREIGN KEY (gagnant) REFERENCES Cycliste(cycliste_id)  
);
```

```
CREATE TABLE Compose_de(  
compose_id int PRIMARY KEY,  
evenement_id int NOT NULL,  
course_id int NOT NULL,  
FOREIGN KEY (course_id) REFERENCES Course(course_id),  
FOREIGN KEY (evenement_id) REFERENCES Evenement(evenement_id)  
);
```

```
CREATE TABLE Participe_eq_evenement(  
parti_eq_id NUMBER(3) PRIMARY KEY,  
equipe_id NUMBER(3) NOT NULL,  
evenement_id NUMBER(3) NOT NULL,  
classement_general TIMESTAMP,  
FOREIGN KEY (Equipe_id) REFERENCES Equipe(Equipe_id),  
FOREIGN KEY (Evenement_id) REFERENCES Evenement(Evenement_id)  
);
```

```
CREATE TABLE Participe_eq_course(  
parti_eq_id int PRIMARY KEY,  
equipe_id int NOT NULL,  
course_id int NOT NULL,  
heure_arrive_somme TIMESTAMP,  
FOREIGN KEY (equipe_id) REFERENCES Equipe(equipe_id),
```

```
FOREIGN KEY (course_id) REFERENCES Course(course_id)
);
```

```
CREATE TABLE Participe_cy_evenement(
parti_cy_id int PRIMARY KEY,
cycliste_id int NOT NULL ,
evenement_id int NOT NULL,
classement_general TIMESTAMP,
FOREIGN KEY (cycliste_id) REFERENCES Cycliste(cycliste_id),
FOREIGN KEY (evenement_id) REFERENCES Evenement(evenement_id)
);
```

```
CREATE TABLE Participe_cy_course(
parti_cy_id int PRIMARY KEY,
cycliste_id int NOT NULL,
course_id int NOT NULL,
points_sprint NUMBER(2),
points_grimpeur NUMBER(2),
heure_arrive TIMESTAMP,
FOREIGN KEY (cycliste_id) REFERENCES Cycliste(cycliste_id),
FOREIGN KEY (course_id) REFERENCES Course(course_id)
);
```

```
CREATE TABLE Palmares(
palmares_id int PRIMARY KEY,
titre varchar(30),
cycliste_id int NOT NULL,
evenement_id int NOT NULL,
FOREIGN KEY (cycliste_id) REFERENCES Cycliste(cycliste_id),
FOREIGN KEY (evenement_id) REFERENCES Evenement(evenement_id)
);
```

```
CREATE TABLE Fait_Parti(
fait_parti int PRIMARY KEY,
Date_debut date,
Date_fin date,
Cycliste_id int NOT NULL,
Equipe_id int NOT NULL,
FOREIGN KEY (Cycliste_id) REFERENCES Cycliste(Cycliste_id),
FOREIGN KEY (Equipe_id) REFERENCES Equipe(Equipe_id)
);
```

```
compose_de: 1..99
cycliste: 100.199
equipe: 200.299
Course: 300..399
even: 400..499
```

par_cy_cr:500.599
par_equi_cr: 600..699
palm: 700..799
party_cy_ev; 800..899
party_eq;ev: 900..999
fait_par: 1000..1099

INSERT DE LA TABLE CYCLISTE AVEC SQL LOADER

LOAD DATA

INFILE 'wikicycle.dat'
APPEND
INTO TABLE cycliste
FIELDS terminated by ',' OPTIONALLY ENCLOSED BY ''''
(cycliste_id,
nom, prenom,
age,
nationalite)
BEGIN DATA

101, 'Dupont', 'Paul', 25, 'France'
102, 'Durand', 'Marie', 30, 'France'
103, 'Smith', 'John', 35, 'USA'
104, 'Johnson', 'Jane', 22, 'USA'
105, 'Lopez', 'Mario', 24, 'Espagne'
106, 'Garcia', 'Sara', 28, 'Espagne'
107, 'Wang', 'Ying', 21, 'Chine'
108, 'Chen', 'Qiang', 27, 'Chine'
109, 'Kim', 'Jung', 23, 'Corée du Sud'
110, 'Martin', 'Paul', 25, 'France'
111, 'Rodriguez', 'Carlos', 28, 'Espagne'
112, 'Perez', 'Sofia', 25, 'Espagne'
113, 'Tsukamoto', 'Takashi', 24, 'Japon'
114, 'Suzuki', 'Akiko', 27, 'Japon'
115, 'Nguyen', 'Linh', 22, 'Vietnam'
116, 'Tran', 'Phuong', 29, 'Vietnam'
117, 'Kumar', 'Rajesh', 23, 'Inde'
118, 'Sharma', 'Priya', 26, 'Inde'
119, 'Mohammed', 'Ali', 25, 'Egypte'
120, 'Hassan', 'Fatima', 28, 'Egypte'
121, 'Kowalski', 'Jan', 24, 'Pologne'
122, 'Nowak', 'Agata', 26, 'Pologne'
123, 'Garcia', 'Miguel', 22, 'Mexico'
124, 'Rodriguez', 'Ana', 29, 'Mexico'
125, 'Müller', 'Hans', 23, 'Allemagne'
126, 'Schmidt', 'Katja', 25, 'Allemagne'
127, 'Chen', 'Tian', 24, 'Taiwan'

128, 'Lin', 'Mei', 28, 'Taiwan'
129, 'Kim', 'Jin-soo', 22, 'Corée du Sud'
130, 'Park', 'Hye-rin', 29, 'Corée du Sud'

INSERT INTO Equipe VALUES (201, 1000000, 'France', 'Team Sky');

INSERT INTO Equipe VALUES (202, 2000000, 'Italie', 'Bora Hansgrohe');
INSERT INTO Equipe VALUES (203, 3000000, 'Espagne', 'Movistar');
INSERT INTO Equipe VALUES (204, 4000000, 'Belgique', 'Quick Step Floors');
INSERT INTO Equipe VALUES (205, 5000000, 'Allemagne', 'Sunweb');
INSERT INTO Equipe VALUES (206, 6000000, 'Australie', 'Mitchelton-Scott');
INSERT INTO Equipe VALUES (207, 7000000, 'Irlande', 'Trek-Segafredo');
INSERT INTO Equipe VALUES (208, 8000000, 'Etats-Unis', 'EF Education First');
INSERT INTO Equipe VALUES (209, 9000000, 'Nouvelle-Zélande', 'Jumbo-Visma');
INSERT INTO Equipe VALUES (210, 10000000, 'Royaume-Uni', 'Ineos Grenadiers');
INSERT INTO Equipe VALUES (211, 12000000, 'Italie', 'Astana');
INSERT INTO Equipe VALUES (212, 13000000, 'Espagne', 'Bahrain-McLaren');
INSERT INTO Equipe VALUES (213, 14000000, 'Belgique', 'BMC Racing Team');
INSERT INTO Equipe VALUES (214, 15000000, 'Allemagne', 'Cofidis');
INSERT INTO Equipe VALUES (215, 16000000, 'Australie', 'Deceuninck-Quick Step');
INSERT INTO Equipe VALUES (216, 17000000, 'Irlande', 'Groupama-FDJ');
INSERT INTO Equipe VALUES (217, 18000000, 'Etats-Unis', 'Lotto Soudal');
INSERT INTO Equipe VALUES (218, 19000000, 'Nouvelle-Zélande', 'UAE Team Emirates');
INSERT INTO Equipe VALUES (219, 20000000, 'Royaume-Uni', 'Team Ineos');
INSERT INTO Equipe VALUES (220, 21000000, 'France', 'Team Sunweb');
INSERT INTO Equipe VALUES (221, 22000000, 'Italie', 'AG2R La Mondiale');
INSERT INTO Equipe VALUES (222, 23000000, 'Espagne', 'Caja Rural-Seguros RGA');
INSERT INTO Equipe VALUES (223, 24000000, 'Belgique', 'Cannondale Pro Cycling Team');
INSERT INTO Equipe VALUES (224, 25000000, 'Allemagne', 'Dimension Data');
INSERT INTO Equipe VALUES (225, 26000000, 'Australie', 'EF Education First-Drpac');
INSERT INTO Equipe VALUES (226, 27000000, 'Irlande', 'Fortuneo-Samsic');
INSERT INTO Equipe VALUES (227, 28000000, 'Etats-Unis', 'Katusha-Alpecin');
INSERT INTO Equipe VALUES (228, 29000000, 'Nouvelle-Zélande', 'LottoNL-Jumbo');
INSERT INTO Equipe VALUES (229, 30000000, 'Royaume-Uni', 'Movistar Team');
INSERT INTO Equipe VALUES (230, 31000000, 'France', 'Team Jumbo-Visma');

INSERT INTO Cycliste VALUES (101, 'Dupont', 'Paul', 25, 'France');
INSERT INTO Cycliste VALUES (102, 'Durand', 'Marie', 30, 'France');
INSERT INTO Cycliste VALUES (103, 'Smith', 'John', 35, 'USA');
INSERT INTO Cycliste VALUES (104, 'Johnson', 'Jane', 22, 'USA');
INSERT INTO Cycliste VALUES (105, 'Lopez', 'Mario', 24, 'Espagne');
INSERT INTO Cycliste VALUES (106, 'Garcia', 'Sara', 28, 'Espagne');
INSERT INTO Cycliste VALUES (107, 'Wang', 'Ying', 21, 'Chine');
INSERT INTO Cycliste VALUES (108, 'Chen', 'Qiang', 27, 'Chine');
INSERT INTO Cycliste VALUES (109, 'Kim', 'Jung', 23, 'Corée du Sud');
INSERT INTO Cycliste VALUES (110, 'Martin', 'Paul', 25, 'France');
INSERT INTO Cycliste VALUES (111, 'Rodriguez', 'Carlos', 28, 'Espagne');

```

INSERT INTO Cycliste VALUES (112, 'Perez', 'Sofia', 25, 'Espagne');
INSERT INTO Cycliste VALUES (113, 'Tsukamoto', 'Takashi', 24, 'Japon');
INSERT INTO Cycliste VALUES (114, 'Suzuki', 'Akiko', 27, 'Japon');
INSERT INTO Cycliste VALUES (115, 'Nguyen', 'Linh', 22, 'Vietnam');
INSERT INTO Cycliste VALUES (116, 'Tran', 'Phuong', 29, 'Vietnam');
INSERT INTO Cycliste VALUES (117, 'Kumar', 'Rajesh', 23, 'Inde');
INSERT INTO Cycliste VALUES (118, 'Sharma', 'Priya', 26, 'Inde');
INSERT INTO Cycliste VALUES (119, 'Mohammed', 'Ali', 25, 'Egypte');
INSERT INTO Cycliste VALUES (120, 'Hassan', 'Fatima', 28, 'Egypte');
INSERT INTO Cycliste VALUES (121, 'Kowalski', 'Jan', 24, 'Pologne');
INSERT INTO Cycliste VALUES (122, 'Nowak', 'Agata', 26, 'Pologne');
INSERT INTO Cycliste VALUES (123, 'Garcia', 'Miguel', 22, 'Mexique');
INSERT INTO Cycliste VALUES (124, 'Rodriguez', 'Ana', 29, 'Mexique');
INSERT INTO Cycliste VALUES (125, 'Müller', 'Hans', 23, 'Allemagne');
INSERT INTO Cycliste VALUES (126, 'Schmidt', 'Katja', 25, 'Allemagne');
INSERT INTO Cycliste VALUES (127, 'Chen', 'Tian', 24, 'Taiwan');
INSERT INTO Cycliste VALUES (128, 'Lin', 'Mei', 28, 'Taiwan');
INSERT INTO Cycliste VALUES (129, 'Kim', 'Jin-soo', 22, 'Corée du Sud');
INSERT INTO Cycliste VALUES (130, 'Park', 'Hye-rin', 29, 'Corée du Sud');

```

```

INSERT INTO Course VALUES (301, 'Etape 1 - Brest - Landerneau',
TO_TIMESTAMP('2022-06-26 14:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 201,0);
INSERT INTO Course VALUES (302, 'Etape 2 - Perros-Guirec - Mûr-de-Bretagne',
TO_TIMESTAMP('2022-06-27 16:45:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 181,0);
INSERT INTO Course VALUES (303, 'Etape 3 - Lorient - Pontivy',
TO_TIMESTAMP('2022-06-28 10:20:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 187,0);
INSERT INTO Course VALUES (304, 'Etape 4 - Redon - Fougères',
TO_TIMESTAMP('2022-06-29 12:30:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 192,0);
INSERT INTO Course VALUES (305, 'Etape 5 - Carhaix - Cap Fréhel',
TO_TIMESTAMP('2022-06-30 18:15:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 207,0);
INSERT INTO Course VALUES (306, 'Etape 1: Turin - Novara', TO_TIMESTAMP('2022-
05-08 09:25:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 153.0,0);
INSERT INTO Course VALUES (307, 'Etape 2: Stupinigi - Novara',
TO_TIMESTAMP('2022-05-09 11:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 180.0,0);
INSERT INTO Course VALUES (308, 'Etape 3: Abbiategrosso - Sestola',
TO_TIMESTAMP('2022-05-10 13:50:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 186.0,0);
INSERT INTO Course VALUES (309, 'Etape 4: Tuscany Firenze - Bagno di Romagna',
TO_TIMESTAMP('2022-05-11 15:40:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 214.0,0);
INSERT INTO Course VALUES (310, 'Etape 5: Modena - Cattolica',
TO_TIMESTAMP('2022-05-12 17:30:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 166.0,0);
INSERT INTO Course VALUES (311, 'Paris Tour', TO_TIMESTAMP('2022-07-01 19:20:00',
'YYYY-MM-DD HH24:MI:SS'), 101, 200,1);
INSERT INTO Course VALUES (312, 'Etape 1: Augsburg - Plzen', TO_TIMESTAMP('2022-
06-25 09:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 173.2,0);
INSERT INTO Course VALUES (313, 'Etape 2: Cheb - Wurzburg', TO_TIMESTAMP('2022-
06-26 11:45:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 196.6,0);

```

```

INSERT INTO Course VALUES (314, 'Etape 3: Schweinfurt - Kronach',
TO_TIMESTAMP('2022-06-27 13:50:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 159.8,0);
INSERT INTO Course VALUES (315, 'Etape 4: Erfurt - Eisleben', TO_TIMESTAMP('2022-
06-28 15:40:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 169.6,0);
INSERT INTO Course VALUES (316, 'Etape 5: Magdeburg - Cottbus',
TO_TIMESTAMP('2022-06-29 17:30:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 214.4,0);
INSERT INTO Course VALUES (317, 'Etape 6: Cottbus - Zittau', TO_TIMESTAMP('2022-
06-30 19:20:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 231.2,0);
INSERT INTO Course VALUES (318, 'Etape 7: Görlitz - Dresden', TO_TIMESTAMP('2022-
07-01 21:10:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 152.4,0);
INSERT INTO Course VALUES (319, 'Etape 8: Dresden - Chemnitz',
TO_TIMESTAMP('2022-07-02 09:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 182.8,0);
INSERT INTO Course VALUES (320, 'Etape 9: Chemnitz - Pöhl', TO_TIMESTAMP('2022-
07-03 11:45:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 208.0,0);
INSERT INTO Course VALUES (321, 'Etape 10: Hof - Bayreuth', TO_TIMESTAMP('2022-
07-04 13:50:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 150.0,0);
INSERT INTO Course VALUES (322, 'Rouen - Le Havre', TO_TIMESTAMP('2022-06-25
09:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 173.2,0);
INSERT INTO Course VALUES (323, 'Lille - Dunkirk', TO_TIMESTAMP('2022-06-26
11:45:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 196.6,0);
INSERT INTO Course VALUES (324, 'Metz - Nancy', TO_TIMESTAMP('2022-06-27
13:50:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 159.8,0);
INSERT INTO Course VALUES (325, 'Reims - Châlons-en-Champagne',
TO_TIMESTAMP('2022-06-28 15:40:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 169.6,0);
INSERT INTO Course VALUES (326, 'Tours - Poitiers', TO_TIMESTAMP('2022-06-29
17:30:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 214.4,0);
INSERT INTO Course VALUES (327, 'Limoges - Bordeaux', TO_TIMESTAMP('2022-06-30
19:20:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 231.2,0);
INSERT INTO Course VALUES (328, 'Toulouse - Montpellier', TO_TIMESTAMP('2022-07-
01 21:10:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 152.4,0); INSERT INTO Course
VALUES (329, 'Marseille - Nice', TO_TIMESTAMP('2022-07-02 09:35:00', 'YYYY-MM-
DD HH24:MI:SS'), NULL, 182.8,0);
INSERT INTO Course VALUES (330, 'Ville - Nice', TO_TIMESTAMP('2022-07-02
09:35:00', 'YYYY-MM-DD HH24:MI:SS'), NULL, 182.8,0);

```

```

INSERT INTO Evenement VALUES (401, 'Tour de France', TO_DATE('2022-06-26','YYYY-
MM-DD'), NULL);
INSERT INTO Evenement VALUES (402, 'Giro d'Italia', TO_DATE('2022-05-01','YYYY-
MM-DD'), NULL);
INSERT INTO Evenement VALUES (403, 'Tour d'Allemagne', TO_DATE('2022-06-
25','YYYY-MM-DD'), NULL);

```

INSERT INTO Evenement VALUES (404, 'Double-course1', TO_DATE('2022-06-26','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (405, 'Double-course2', TO_DATE('2022-06-27','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (406, 'Double-course3', TO_DATE('2022-06-28','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (407, 'Double-course4', TO_DATE('2022-06-29','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (408, 'Double-course5', TO_DATE('2022-06-30','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (409, 'Double-course6', TO_DATE('2022-06-28','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (410, 'Double-course7', TO_DATE('2022-06-29','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (411, 'Double-course8', TO_DATE('2022-06-30','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (412, 'Double-course9', TO_DATE('2022-07-01','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (413, 'Double-course10', TO_DATE('2022-07-02','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (414, 'Double-course11', TO_DATE('2022-07-03','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (415, 'Double-course12', TO_DATE('2022-07-04','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (416, 'Double-course13', TO_DATE('2022-07-05','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (417, 'Double-course14-1', TO_DATE('2022-07-06','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (418, 'Double-course14-2', TO_DATE('2022-07-06','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (419, 'Double-course15', TO_DATE('2022-07-07','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (420, 'Double-course16', TO_DATE('2022-07-08','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (421, 'Double-course17', TO_DATE('2022-07-09','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (422, 'Double-course18', TO_DATE('2022-07-10','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (423, 'Double-course19', TO_DATE('2022-07-11','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (424, 'Double-course20', TO_DATE('2022-07-12','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (425, 'Double-course21', TO_DATE('2022-07-13','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (426, 'Double-course22', TO_DATE('2022-07-14','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (427, 'Double-course23', TO_DATE('2022-07-15','YYYY-MM-DD'), NULL);

```
INSERT INTO Evenement VALUES (428, 'Double-course24', TO_DATE('2022-07-16','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (429, 'Double-course25', TO_DATE('2022-07-17','YYYY-MM-DD'), NULL);
INSERT INTO Evenement VALUES (430, 'Double-course26', TO_DATE('2022-07-18','YYYY-MM-DD'), NULL);
```

```
INSERT INTO Compose_de VALUES (1, 401, 301);
INSERT INTO Compose_de VALUES (2, 401, 302);
INSERT INTO Compose_de VALUES (3, 401, 303);
INSERT INTO Compose_de VALUES (4, 401, 304);
INSERT INTO Compose_de VALUES (5, 401, 305);
INSERT INTO Compose_de VALUES (6, 402, 306);
INSERT INTO Compose_de VALUES (7, 402, 307);
INSERT INTO Compose_de VALUES (8, 402, 308);
INSERT INTO Compose_de VALUES (9, 402, 309);
INSERT INTO Compose_de VALUES (10, 402, 310);
INSERT INTO Compose_de VALUES (11, 403, 312);
INSERT INTO Compose_de VALUES (12, 403, 313);
INSERT INTO Compose_de VALUES (13, 403, 314);
INSERT INTO Compose_de VALUES (14, 403, 315);
INSERT INTO Compose_de VALUES (15, 403, 316);
INSERT INTO Compose_de VALUES (16, 403, 317);
INSERT INTO Compose_de VALUES (17, 403, 318);
INSERT INTO Compose_de VALUES (18, 403, 319);
INSERT INTO Compose_de VALUES (19, 403, 320);
INSERT INTO Compose_de VALUES (20, 403, 320);
INSERT INTO Compose_de VALUES (21, 404, 321);
INSERT INTO Compose_de VALUES (22, 404, 322);
INSERT INTO Compose_de VALUES (23, 405, 323);
INSERT INTO Compose_de VALUES (24, 405, 324);
INSERT INTO Compose_de VALUES (25, 406, 325);
INSERT INTO Compose_de VALUES (26, 406, 326);
INSERT INTO Compose_de VALUES (27, 407, 327);
INSERT INTO Compose_de VALUES (28, 407, 328);
INSERT INTO Compose_de VALUES (29, 408, 329);
INSERT INTO Compose_de VALUES (30, 408, 330);
```

```
INSERT INTO Participe_eq_evenement VALUES (901, 201, 401, NULL);
INSERT INTO Participe_eq_evenement VALUES (902, 202, 402, NULL);
INSERT INTO Participe_eq_evenement VALUES (903, 203, 403, NULL);
INSERT INTO Participe_eq_evenement VALUES (904, 204, 404, NULL);
INSERT INTO Participe_eq_evenement VALUES (905, 205, 405, NULL);
INSERT INTO Participe_eq_evenement VALUES (906, 206, 406, NULL);
INSERT INTO Participe_eq_evenement VALUES (907, 207, 407, NULL);
INSERT INTO Participe_eq_evenement VALUES (908, 208, 408, NULL);
INSERT INTO Participe_eq_evenement VALUES (909, 209, 409, NULL);
INSERT INTO Participe_eq_evenement VALUES (910, 210, 410, NULL);
```


INSERT INTO Participe_eq_evenement VALUES (911, 211, 411, NULL);
INSERT INTO Participe_eq_evenement VALUES (912, 212, 412, NULL);
INSERT INTO Participe_eq_evenement VALUES (913, 213, 413, NULL);
INSERT INTO Participe_eq_evenement VALUES (914, 214, 414, NULL);
INSERT INTO Participe_eq_evenement VALUES (915, 215, 415, NULL);
INSERT INTO Participe_eq_evenement VALUES (916, 216, 416, NULL);
INSERT INTO Participe_eq_evenement VALUES (917, 217, 417, NULL);
INSERT INTO Participe_eq_evenement VALUES (918, 218, 418, NULL);
INSERT INTO Participe_eq_evenement VALUES (919, 219, 419, NULL);
INSERT INTO Participe_eq_evenement VALUES (920, 220, 420, NULL);
INSERT INTO Participe_eq_evenement VALUES (921, 221, 421, NULL);
INSERT INTO Participe_eq_evenement VALUES (922, 222, 422, NULL);
INSERT INTO Participe_eq_evenement VALUES (923, 223, 423, NULL);
INSERT INTO Participe_eq_evenement VALUES (924, 224, 424, NULL);
INSERT INTO Participe_eq_evenement VALUES (925, 225, 425, NULL);
INSERT INTO Participe_eq_evenement VALUES (926, 226, 426, NULL);
INSERT INTO Participe_eq_evenement VALUES (927, 227, 427, NULL);
INSERT INTO Participe_eq_evenement VALUES (928, 228, 428, NULL);
INSERT INTO Participe_eq_evenement VALUES (929, 229, 429, NULL);
INSERT INTO Participe_eq_evenement VALUES (930, 230, 430, NULL);

INSERT INTO Participe_eq_course VALUES (600, 201, 301);
INSERT INTO Participe_eq_course VALUES (601, 201, 302);
INSERT INTO Participe_eq_course VALUES (602, 201, 303);
INSERT INTO Participe_eq_course VALUES (603, 201, 304);
INSERT INTO Participe_eq_course VALUES (604, 201, 305);
INSERT INTO Participe_eq_course VALUES (605, 201, 306);
INSERT INTO Participe_eq_course VALUES (606, 201, 307);
INSERT INTO Participe_eq_course VALUES (607, 201, 308);
INSERT INTO Participe_eq_course VALUES (608, 201, 309);
INSERT INTO Participe_eq_course VALUES (609, 201, 310);
INSERT INTO Participe_eq_course VALUES (610, 202, 301);
INSERT INTO Participe_eq_course VALUES (611, 202, 302);
INSERT INTO Participe_eq_course VALUES (612, 202, 303);
INSERT INTO Participe_eq_course VALUES (613, 202, 304);
INSERT INTO Participe_eq_course VALUES (614, 202, 305);
INSERT INTO Participe_eq_course VALUES (615, 202, 306);
INSERT INTO Participe_eq_course VALUES (616, 202, 307);
INSERT INTO Participe_eq_course VALUES (617, 202, 308);
INSERT INTO Participe_eq_course VALUES (618, 202, 309);
INSERT INTO Participe_eq_course VALUES (619, 202, 310);

INSERT INTO Participe_cy_evenement VALUES (800, 101, 401, NULL);
INSERT INTO Participe_cy_evenement VALUES (801, 102, 402, NULL);
INSERT INTO Participe_cy_evenement VALUES (802, 103, 403, NULL);
INSERT INTO Participe_cy_evenement VALUES (803, 104, 404, NULL);

```

INSERT INTO Participe_cy_evenement VALUES (804, 105, 405, NULL);
INSERT INTO Participe_cy_evenement VALUES (805, 106, 406, NULL);
INSERT INTO Participe_cy_evenement VALUES (806, 107, 407, NULL);
INSERT INTO Participe_cy_evenement VALUES (807, 108, 408, NULL);
INSERT INTO Participe_cy_evenement VALUES (808, 109, 409, NULL);
INSERT INTO Participe_cy_evenement VALUES (809, 110, 410, NULL);
INSERT INTO Participe_cy_evenement VALUES (810, 111, 411, NULL);
INSERT INTO Participe_cy_evenement VALUES (811, 112, 412, NULL);
INSERT INTO Participe_cy_evenement VALUES (812, 113, 413, NULL);
INSERT INTO Participe_cy_evenement VALUES (813, 114, 414, NULL);
INSERT INTO Participe_cy_evenement VALUES (814, 115, 415, NULL);
INSERT INTO Participe_cy_evenement VALUES (815, 116, 416, NULL);
INSERT INTO Participe_cy_evenement VALUES (816, 117, 402, NULL);
INSERT INTO Participe_cy_evenement VALUES (817, 118, 402, NULL);
INSERT INTO Participe_cy_evenement VALUES (818, 119, 402, NULL);
INSERT INTO Participe_cy_evenement VALUES (819, 120, 402, NULL);

```

```

INSERT INTO Participe_cy_course VALUES (500, 101, 301, 10, 10, NULL);
INSERT INTO Participe_cy_course VALUES (501, 102, 301, 9, 5, NULL);
INSERT INTO Participe_cy_course VALUES (502, 103, 301, 8, 7, NULL);
INSERT INTO Participe_cy_course VALUES (503, 104, 301, 7, 9, NULL);
INSERT INTO Participe_cy_course VALUES (504, 105, 301, 6, 10, NULL);
INSERT INTO Participe_cy_course VALUES (505, 106, 301, 5, 6, NULL);
INSERT INTO Participe_cy_course VALUES (506, 107, 301, 4, 8, NULL);
INSERT INTO Participe_cy_course VALUES (507, 108, 301, 3, 10, NULL);
INSERT INTO Participe_cy_course VALUES (508, 109, 301, 2, 5, NULL);
INSERT INTO Participe_cy_course VALUES (509, 110, 301, 1, 7, NULL);

```

```

INSERT INTO Palmares VALUES (700, 'Meilleur Sprinteur', 110, 401);
INSERT INTO Palmares VALUES (701, 'Meilleur Grimpeur', 101, 401);
INSERT INTO Palmares VALUES (702, 'Meilleur Jeune Coureur', 102, 401);
INSERT INTO Palmares VALUES (703, 'Coureur le Plus Offensif', 103, 401);
INSERT INTO Palmares VALUES (704, 'Meilleure Equipe', 104, 401);
INSERT INTO Palmares VALUES (705, 'Meilleur Sprinteur', 105, 402);
INSERT INTO Palmares VALUES (706, 'Meilleur Grimpeur', 106, 402);
INSERT INTO Palmares VALUES (707, 'Meilleur Jeune Coureur', 107, 402);
INSERT INTO Palmares VALUES (708, 'Coureur le Plus Offensif', 108, 402);
INSERT INTO Palmares VALUES (709, 'Meilleure Equipe', 109, 402);

```

```

INSERT INTO Fait_Partir VALUES (1000, TO_DATE('2008-03-05','YYYY-MM-DD'),
TO_DATE('2012-03-06','YYYY-MM-DD'), 101, 227);
INSERT INTO Fait_Partir VALUES (1001, TO_DATE('2008-01-14','YYYY-MM-DD'),
TO_DATE('2013-01-17','YYYY-MM-DD'), 102, 225);
INSERT INTO Fait_Partir VALUES (1002, TO_DATE('2007-03-06','YYYY-MM-DD'),
TO_DATE('2012-03-03','YYYY-MM-DD'), 103, 223);
INSERT INTO Fait_Partir VALUES (1003, TO_DATE('2006-07-02','YYYY-MM-DD'),
TO_DATE('2012-07-06','YYYY-MM-DD'), 104, 201);

```

INSERT INTO Fait_Partir VALUES (1004, TO_DATE('2008-03-04','YYYY-MM-DD'),
 TO_DATE('2011-03-02','YYYY-MM-DD'), 108, 205);
 INSERT INTO Fait_Partir VALUES (1005, TO_DATE('2010-03-08','YYYY-MM-DD'),
 TO_DATE('2015-03-04','YYYY-MM-DD'), 110, 209);
 INSERT INTO Fait_Partir VALUES (1006, TO_DATE('2013-01-10','YYYY-MM-DD'),
 TO_DATE('2015-03-04','YYYY-MM-DD'), 112, 204);
 INSERT INTO Fait_Partir VALUES (1007, TO_DATE('2012-03-02','YYYY-MM-DD'),
 TO_DATE('2015-03-04','YYYY-MM-DD'), 113, 205);
 INSERT INTO Fait_Partir VALUES (1008, TO_DATE('2015-04-08','YYYY-MM-DD'),
 TO_DATE('2015-03-04','YYYY-MM-DD'), 109, 215);
 INSERT INTO Fait_Partir VALUES (1009, TO_DATE('2010-09-02','YYYY-MM-DD'),
 TO_DATE('2022-07-04','YYYY-MM-DD'), 114, 223);
 INSERT INTO Fait_Partir VALUES (1010, TO_DATE('2011-03-05','YYYY-MM-DD'),
 TO_DATE('2020-03-06','YYYY-MM-DD'), 116, 216);
 INSERT INTO Fait_Partir VALUES (1011, TO_DATE('2012-01-16','YYYY-MM-DD'),
 TO_DATE('2022-01-02','YYYY-MM-DD'), 117, 214);
 INSERT INTO Fait_Partir VALUES (1012, TO_DATE('2016-03-07','YYYY-MM-DD'),
 TO_DATE('2019-09-06','YYYY-MM-DD'), 120, 202);
 INSERT INTO Fait_Partir VALUES (1013, TO_DATE('2012-07-09','YYYY-MM-DD'),
 TO_DATE('2020-01-11','YYYY-MM-DD'), 122, 228);
 INSERT INTO Fait_Partir VALUES (1014, TO_DATE('2011-03-02','YYYY-MM-DD'),
 TO_DATE('2021-03-16','YYYY-MM-DD'), 123, 201);
 INSERT INTO Fait_Partir VALUES (1015, TO_DATE('2012-03-05','YYYY-MM-DD'),
 TO_DATE('2021-07-03','YYYY-MM-DD'), 124, 203);
 INSERT INTO Fait_Partir VALUES (1016, TO_DATE('2015-01-16','YYYY-MM-DD'),
 TO_DATE('2021-01-01','YYYY-MM-DD'), 127, 224);
 INSERT INTO Fait_Partir VALUES (1017, TO_DATE('2013-03-02','YYYY-MM-DD'),
 TO_DATE('2021-03-16','YYYY-MM-DD'), 128, 230);
 INSERT INTO Fait_Partir VALUES (1018, TO_DATE('2014-04-06','YYYY-MM-DD'),
 TO_DATE('2022-04-07','YYYY-MM-DD'), 129, 218);
 INSERT INTO Fait_Partir VALUES (1019, TO_DATE('2014-09-05','YYYY-MM-DD'),
 TO_DATE('2022-09-19','YYYY-MM-DD'), 130, 222);

INSERT INTO Fait_Partir VALUES (1020, TO_DATE('2012-03-06','YYYY-MM-DD'),
 TO_DATE('2020-01-05','YYYY-MM-DD'), 101, 202);
 INSERT INTO Fait_Partir VALUES (1021, TO_DATE('2013-01-17','YYYY-MM-DD'),
 TO_DATE('2019-02-21','YYYY-MM-DD'), 102, 204);
 INSERT INTO Fait_Partir VALUES (1022, TO_DATE('2012-03-03','YYYY-MM-DD'),
 TO_DATE('2016-03-22','YYYY-MM-DD'), 103, 206);
 INSERT INTO Fait_Partir VALUES (1023, TO_DATE('2012-07-06','YYYY-MM-DD'),
 TO_DATE('2013-04-17','YYYY-MM-DD'), 104, 208);
 INSERT INTO Fait_Partir VALUES (1024, TO_DATE('2014-03-07','YYYY-MM-DD'),
 TO_DATE('2016-05-05','YYYY-MM-DD'), 105, 210);
 INSERT INTO Fait_Partir VALUES (1025, TO_DATE('2011-03-02','YYYY-MM-DD'),
 TO_DATE('2013-06-14','YYYY-MM-DD'), 106, 212);
 INSERT INTO Fait_Partir VALUES (1026, TO_DATE('2013-01-16','YYYY-MM-DD'),
 TO_DATE('2016-07-12','YYYY-MM-DD'), 107, 214);

INSERT INTO Fait_Partir VALUES (1027, TO_DATE('2011-03-02','YYYY-MM-DD'),
TO_DATE('2015-08-04','YYYY-MM-DD'), 108, 216);
INSERT INTO Fait_Partir VALUES (1028, TO_DATE('2012-04-03','YYYY-MM-DD'),
TO_DATE('2014-09-02','YYYY-MM-DD'), 109, 218);
INSERT INTO Fait_Partir VALUES (1029, TO_DATE('2010-09-07','YYYY-MM-DD'),
TO_DATE('2019-10-01','YYYY-MM-DD'), 110, 220);

INSERT INTO Fait_Partir VALUES (1030, TO_DATE('2022-03-07','YYYY-MM-DD'),
CURRENT_DATE, 111, 222);
INSERT INTO Fait_Partir VALUES (1031, TO_DATE('2017-01-16','YYYY-MM-DD'),
CURRENT_DATE, 112, 224);
INSERT INTO Fait_Partir VALUES (1032, TO_DATE('2019-03-02','YYYY-MM-DD'),
CURRENT_DATE, 113, 226);
INSERT INTO Fait_Partir VALUES (1033, TO_DATE('2022-07-04','YYYY-MM-DD'),
CURRENT_DATE, 114, 228);
INSERT INTO Fait_Partir VALUES (1034, TO_DATE('2017-03-05','YYYY-MM-DD'),
CURRENT_DATE, 115, 230);
INSERT INTO Fait_Partir VALUES (1035, TO_DATE('2020-03-06','YYYY-MM-DD'),
CURRENT_DATE, 116, 201);
INSERT INTO Fait_Partir VALUES (1036, TO_DATE('2022-01-02','YYYY-MM-DD'),
CURRENT_DATE, 117, 203);
INSERT INTO Fait_Partir VALUES (1037, TO_DATE('2021-03-07','YYYY-MM-DD'),
CURRENT_DATE, 118, 205);
INSERT INTO Fait_Partir VALUES (1038, TO_DATE('2018-04-11','YYYY-MM-DD'),
CURRENT_DATE, 119, 207);
INSERT INTO Fait_Partir VALUES (1039, TO_DATE('2019-09-06','YYYY-MM-DD'),
CURRENT_DATE, 120, 209);
INSERT INTO Fait_Partir VALUES (1040, TO_DATE('2021-03-04','YYYY-MM-DD'),
CURRENT_DATE, 121, 211);
INSERT INTO Fait_Partir VALUES (1041, TO_DATE('2020-01-11','YYYY-MM-DD'),
CURRENT_DATE, 122, 213);
INSERT INTO Fait_Partir VALUES (1042, TO_DATE('2021-03-16','YYYY-MM-DD'),
CURRENT_DATE, 123, 215);
INSERT INTO Fait_Partir VALUES (1043, TO_DATE('2021-07-03','YYYY-MM-DD'),
CURRENT_DATE, 124, 217);
INSERT INTO Fait_Partir VALUES (1044, TO_DATE('2020-03-19','YYYY-MM-DD'),
CURRENT_DATE, 125, 219);
INSERT INTO Fait_Partir VALUES (1045, TO_DATE('2022-03-04','YYYY-MM-DD'),
CURRENT_DATE, 126, 221);
INSERT INTO Fait_Partir VALUES (1046, TO_DATE('2021-01-01','YYYY-MM-DD'),
CURRENT_DATE, 127, 223);
INSERT INTO Fait_Partir VALUES (1047, TO_DATE('2021-03-16','YYYY-MM-DD'),
CURRENT_DATE, 128, 225);
INSERT INTO Fait_Partir VALUES (1048, TO_DATE('2022-04-07','YYYY-MM-DD'),
CURRENT_DATE, 129, 227);
INSERT INTO Fait_Partir VALUES (1049, TO_DATE('2022-09-19','YYYY-MM-DD'),
CURRENT_DATE, 130, 229);