Bases de données (1)

octobre 2022

tdm 5

Exercice 1 :
Pour cet exercice on reprend la base de données « course cycliste » utilisée lors des premières séances.
Conservez dans un fichier `.sql` une copie de toutes les commandes SQL demandées.
La course se déroule en plusieurs étapes. Chaque étape possède un numéro (entier ≥ 1) et un nom.
Q 1 . Créez la table `etapes` **en écrivant la commande SQL** nécesssaire (la clé primaire sera le numéro)
Une fois la table créée, peuplez-la, via une commande SQL, de quelques étapes dont vous choisirez les noms.
Q 2 . Une table `temps` contient les temps relevés à l'arrivée de chaque coureur à chaque étape.
Cette table aura donc 4 attributs : `dossard`, `etape` (numéro d'étape), `chrono` (temps de type `interval`), `id` (identifiant du relevé).
Pour l'identifiant (qui sera la clé primaire), vous utiliserez le mécanisme de génération incrémentale (« `serial` »).
Q 3 .

- deux contraintes d'intégrité référentielle devraient être ajoutées à la table `temps`. Lesquelles ?
- ajoutez ces contraintes par 2 requêtes SQL.
  NB : pour ajouter une contrainte non prévue à la création de la table on utlise la commande `alter table` :
  `alter table` *nomDeTable* `add` *contrainte*

Q 4 . Les étapes doivent toutes être parcourues en moins de 6 heures. Ajoutez une contrainte empêchant la saisie d'un chrono négatif ou de plus de 6 heures.
Vérifiez ce qui se passe quand l'on tente d'ajouter dans la table temps un chrono non conforme à la contrainte.

Le type interval

Ce type permet de représenter un intervalle de temps (donc également une durée). Comme les autres types temps, il est saisi sous forme d'une chaîne de caractère convertie en `interval`. De nombreuses formes sont autorisées pour les chaînes. Par exemple `'7 days', '5h', '5:00:00'`. L'extraction d'un composant est réalisée comme pour les autres types de temps, par `extract` ou `date_part`
D'autre part des opérateurs sont autorisés pour les temps. Par exemple on peut ajouter une date et une durée `'2020-09-27 23:53:00'::timestamp + '8 minutes'::interval` vaut `2020-09-28 00:01:00`
Mreci de consulter la documentation :
`https://docs.postgresql.fr/11/datatype-datetime.html`
`https://docs.postgresql.fr/11/functions-datetime.html`

Q 5 . Quelle requête SQL permet d'insérer un tuple dans la table `temps` ? (vous choisirez quel tuple).
Vous prendrez garde de ne pas fixer vous-même la valeur de l'identifiant du relevé afin de laisser la génération incrémentale se réaliser.
Insérez ainsi quelques tuples (de votre choix).
Q 6 .

1. vous allez commencer par créer une copie des données de chacune des tables `etapes` et `temps` par la commande :
   `create table` *tableACreer* `as select * from` *tableACopier*
   NB : ceci ne copie pas les contraintes de la table d'origine, donc ne définit pas de clé primaire. Ce n'est pas très important ici car la table est uniquement destinée à une copie « de sauvegarde ».
2. on souhaite maintenant vider les tables `etapes` et `temps`.
   Tentez d'abord la commande `delete from etapes` Que se passe-t-il ? pourquoi ?

3. videz alors la table `temps` (sans détruire la table elle-même), puis insérez un nouveau tuple dans cette même table. Quelle valeur a été générée pour son identifiant ?
4. videz une nouvelle fois la table `temps` puis reinitialiser le générateur par la commande `alter sequence temps_id_seq restart`
5. videz la table étapes
6. restaurez les données des tables `etapes` et `temps` en utilisant les valeurs sauvegardées.

Q 7 .

Construisez la requête SQL permettant d'obtenir le tableau d'arrivée pour l'étape 1, c'est à dire la liste des coureurs classés du premier au dernier arrivé, avecpour chacun son numéro de dossard, son chrono et son rang de classement.
NB : le rang de classement est obtenu par l'expression `rank() over(order by chrono)`
Enregistrez cette requête comme une vue. Faites de même pour l'étape 2

Q 8 .

Obtenez par une requête select une relation avec les attributs : dossard, nom du coureur, nom de l'équipe, chrono à l'étape 1, classement à l'étape 1, chrono à l'étape 2, classement à l'étape 2
Les coureurs qui n'ont pas terminé une étape n'ont donc pas de chrono ni de classement. Ils apparaissent quand même dans la table, avec les attributs correspondant non définis.

Q 9 .

Complétez la requête précédente pour obtenir en plus l'attribut chrono total des 2 étapes.
Remarquez que l'opérateur `+` travaille de façon logique : si l'une des valeurs est indéfinie le résultat l'est aussi. C'est exactement ce que l'on souhaite ici.