Bases de données - TP 2

Corrigé Code de partage avec Capytale : ac17-8534288

Il s'agit d'un extrait de sujet Ecricome (2024) : extrait d'une partie de l'énoncé concernant les bases de données et extrait de la partie de l'annexe correspondante.
Pour la séance, un échantillon de données a été créé pour permettre de tester les requêtes.

Partie IV

Un fabricant d'ordinateurs souhaite publier des données statistiques sur la durée de vie de ses appareils fabriqués à partir de l'an 2000. Dans une base de données, on dispose d'une table `ordinateurs` contenant des informations sur tous les ordinateurs produits par le fabricant. Cette table possède les attributs (ou colonnes) suivants.
- `id` (de type `INTEGER`) : le numéro d'identification de l'ordinateur.
- `annee_fabrication` (de type `INTEGER`) : l'année de fabrication de l'ordinateur.
- `adresse_ip` (de type `INTEGER`) : l'adresse IP associée à l'ordinateur.
- `annee_panne` (de type `INTEGER`) : l'année où l'ordinateur a cessé de fonctionner, valant -1 si l'ordinateur est encore en état de marche.

Dans les questions qui suivent, en plus des commandes SQL au programme, on pourra utiliser les fonctions présentées dans l'**Annexe B** en fin de sujet.

9. (a) Ecrire une requête SQL permettant de déterminer le nombre total d'ordinateurs produits par le fabricant.

   Il suffit de compter le nombre total d'enregistrements de la table `ordinateur`. Ce qu'on fait avec la requête :

   ```
   SELECT COUNT(*) FROM ordinateur
   ```

   (b) Ecrire une requête SQL permettant de déterminer le nombre d'ordinateurs ayant cessé de fonctionner exactement un an après leur production.

   Il faut maintenant ajouter une condition en ne gardant que les enregistrements qui vérifient la condition voulue :

   ```
   SELECT COUNT(*) FROM ordinateur WHERE annee_panne = annee_fabrication +1
   -- ou
   select count(*) from ordinateurs where annee_panne-annee_fabrication=1
   ```

   (c) Dans cette question uniquement, on suppose que la durée de vie en années d'un ordinateur est une variable aléatoire de loi géométrique, de paramètre p inconnu.
   Expliquer de quelle manière le résultat des requêtes écrites dans les questions 9.(a) et 9.(b) peut être utilisé pour estimer le paramètre p

   En calculant la fréquence du nombre d'ordinateurs qui tombent en panne au bout de un an d'utilisation (c'est-à-dire le quotient du nombre calculé en **9.b.** par celui calculé en **9.a.**), on a donc une valeur approchée de $P(X = 1) = p$ (par la loi faible des grands nombres) si X est la variable aléatoire égale à la durée de vie d'un ordinateur et car $X \hookrightarrow \mathscr{G}(p)$

10. Un attribut `duree_vie`, de type `INTEGER`, a été ajouté à la table `ordinateur`. Aux champs de l'attribut `duree_vie` a été affectée la valeur -1
    Ecrire une requête SQL permettant de modifier la table `ordinateur` en affectant, pour chaque ordinateur, sa durée de vie à l'attribut `duree_vie`. Dans le cas des ordinateurs qui sont encore en état de marche, on ne modifiera pas la valeur -1 déjà affectée.

On utilise la requête `UPDATE` avec une condition pour ne modifier que les lignes où les ordinateurs sont tombés en panne. Plus précisément :

```
UPDATE ordinateur
SET duree_vie = annee_panne - annee_fabrication WHERE annee_panne <>-1
```

11. Dans cette question, on cherche à déterminer s'il est raisonnable de représenter la durée de vie d'un ordinateur par une variable aléatoire de loi géométrique d'un certain paramètre p que l'on cherchera à approcher.

(a) Expliquer comment le résultat de la requête suivante permet d'obtenir une valeur approchée de p

```
SELECT AVG(duree_vie) FROM ordinateurs
```

Cette requête renvoie la moyenne des valeurs de l'attribut `duree_vie`. D'après la loi faible des grands nombres, la moyenne empirique d'un échantillon d'une variable aléatoire renvoie une valeur approchée de l'espérance de la variable échantillonnée. Si la durée de vie est une loi géométrique de paramètre p, son espérance vaut $\frac{1}{p}$. Il faut donc prendre l'inverse de la valeur renvoyée par la requête pour une valeur approchée de p

Remarque : cette requête comporte un biais (erreur ou oubli de l'énoncé), puisqu'elle intègre dans la moyenne des valeurs égales à -1 qui n'ont pas de sens en termes de durée de vie. Il aurait fallu la compléter avec `where duree_vie<> -1`.

(b) La base de données compte au total 10 000 ordinateurs. On exécute les requêtes suivantes :

```
SELECT COUNT(*)/10000 FROM ordinateurs WHERE duree_vie = 1 ;
SELECT COUNT(*)/10000 FROM ordinateurs WHERE duree_vie = 2 ;
```

⋮ ⋮ ⋮

```
SELECT COUNT(*)/10000 FROM ordinateurs WHERE duree_vie = 24 ;
```

En utilisant les résultats de la question 8, expliquer de quelle manière les données de la table ordinateur peuvent être exploitées pour déterminer s'il est raisonnable de représenter la durée de vie d'un ordinateur par une variable aléatoire de loi géométrique.

Précision : à la question 8, on cherchait à déterminer la loi d'une variable aléatoire en fonction du comportement de la suite de ses valeurs $P(X = k)$, quel est-il pour une loi géométrique ?

Chacune des requêtes (pour k de 1 à 24)

```
SELECT COUNT(*)/10000 FROM ordinateur WHERE duree_vie= k
```

renvoie la proportion des appareils avec une durée de vie de k années, qui devrait donner une *estimation* de la probabilité qu'un appareil ait une durée de vie de k années.
Si toutes les valeurs calculées sont les termes d'une suite géométrique, il est raisonnable d'utiliser une loi géométrique pour la durée de vie. Pour vérifier cela, on peut calculer les quotients des valeurs successives qui doivent être constants (comme le garantit la question **8.a.**).
Rappel : la suite des valeurs d'une loi géométrique est une suite géométrique,
$P(X = k + 1) = pq^k = qpq^{k-1} = qP(X = k)$

Remarque : ce jeu de données comporte plusieurs biais. Outre le fait de garder les valeurs égales à -1, les ordinateurs à longue durée de vie sont sous-représentés, notamment dans les dernières années, puisqu'ils n'ont pas encore « eu le temps » de tomber en panne.

Annexe B - Commandes SQL

La fonction COUNT()

La fonction d'agrégation `COUNT()` permet de connaitre le nombre d'enregistrements d'une table, vérifiant éventuellement une certaine condition.
Nous donnons ci-dessous plusieurs exemples d'utilisation de la fonction `COUNT()`, en considérant une table nommée `ma_table` comportant deux colonnes `colonne_1` et `colonne_2`.

- La requête suivante renvoie le nombre total d'enregistrements dans `ma_table` :

```
SELECT COUNT(*) FROM ma_table
```

- La requête suivante renvoie le nombre d'enregistrements de `ma_table` vérifiant la condition `cond` :

```
SELECT COUNT(*) FROM ma_table
WHERE cond
```

- La requête suivante renvoie le nombre d'enregistrements de `ma_table` pour lesquels la valeur de `colonne_2` n'est pas vide :

```
SELECT COUNT(colonne_2) FROM ma_table
```

La fonction d'agrégation AVG()

La fonction `AVG()` permet de calculer la moyenne des valeurs d'une colonne dans une table.
Par exemple, si on considère la table nommée `table` contenant les enregistrements suivants :

colonne_1	colonne_2	colonne_3	colonne_4
1	69	Lyon	4
2	31	Toulouse	8
3	54	Nancy	5
4	64	Saint-Jean-de-Luz	17
5	44	Nantes	6

alors la requête suivante

```
SELECT AVG(colonne_4)
    FROM table
    WHERE colonne_1<=3
```

affiche la moyenne des valeurs de `colonne_4` des trois premiers enregistrements : **5.6667** c'est-à-dire $\frac{4+8+5}{3}$