

**S2.04 - Exploitation d’une base de données**

Partie 2 : Interrogation dynamique de la base

Compte-rendu final

ROUYER Johan – TD1 / TP2

GUIHENEUF Mattin – TD1 / TP2

B.U.T Informatique – Semestre 2 (2022 – 2023)

I.U.T de Bayonne et du Pays Basque

Département Informatique

# **Rappel de la problématique**

Peut-on établir un lien entre type de pages fréquentées et localisation des visiteurs ?

# **Solution Informatique**

## **Objectif**

L’objectif de la solution informatique est d’avoir des éléments de réponses à la problématique grâce aux questions mises en place. Chacune d’elles correspond à une requête qui sera exécutée, suite à une saisie de paramètres par l’utilisateur ou non.

## **Organisation**

L’application informatique est séparée en 4 fichiers python :

* Un premier consacré au menu principal
* Un deuxième consacré à la requête n°1
* Un troisième pour la requête n°2
* Un dernier pour la requête n°3

### **Menu Principal**

On affiche tout d’abord le menu principal avec les 3 questions à disposition et un chiffre associé à chacune. On propose à l’utilisateur de saisir le chiffre de son choix. Tant que celui-ci ne correspondra pas à une des trois valeurs possibles, un message d’erreur apparaitra, demandant d’adapter la saisie aux possibilités qui s’offrent à lui. Une fois que la valeur saisi sera possible, l’utilisateur sera redirigé vers la question associé à son choix.

### **Requête 1**

Si le choix de l’utilisateur est la première question, le terminal lui affichera les éléments suivants :

* La question suivie d’une courte description du rôle de la requête
* Le résultat de la requête 1
* Une saisie à réaliser par l’utilisateur.

Celle-ci lui propose de dessiner un graphique à partir du résultat, d’obtenir une courte analyse du résultat ou de retourner au menu principal. Elle fonctionne de la même manière que la précédente saisie, c’est-à-dire que tant que la valeur ne correspondra pas à une possible, un message d’erreur s’affichera.

### **Requête 2**

Si le choix de l’utilisateur est la deuxième question, le terminal lui affichera les éléments suivants :

* La question suivie d’une courte description de la requête
* Une explication du fonctionnement et des informations utiles pour démarrer la requête
* Une saisie pour connaitre la période que souhaite analyser l’utilisateur.

A partir de ceci, le terminal affichera le résultat de la requête. Celle-ci fonctionne grâce à des vues, ci-jointes en annexe, établissant pour chacune des 4 périodes, le nombre d’actions réalisées par région pour chaque type de page puis pour chaque type de page le nombre d’actions maximum.

De la même façon qu’à la requête 2, l’utilisateur peut dessiner un graphique, obtenir une courte analyse ou revenir au menu principal. Il peut également analyser une autre période si il le souhaite.

### **Requête 3**

Si le choix de l’utilisateur est la deuxième question, le terminal lui affichera les éléments suivants :

* La question et une courte description du rôle de la requête
* Une liste régions disponibles à l’analyse, puisque nous avons choisi de garder uniquement celles ayant réalisées au moins 200 actions pour garder les échantillons utiles à l’analyse et pour ne pas surcharger le terminal
* Une saisie de la région que l’utilisateur désire.

Ensuite, une deuxième saisie s’impose à lui, pour obtenir les x premiers types de pages les plus fréquentés. Il devra saisir le nombre qu’il souhaite, devant être supérieur à 0 et inférieur ou égal à 20.

Le résultat de la requête s’affichera et on lui proposera de dessiner un graphique, démarrer une nouvelle analyse ou retourner au menu principal.

## **Annexes**

**Création de vues :**

1. HIVER

Vue 1 : CREATE VIEW ActRegionHiver AS

SELECT P.type\_page, L.Region1, COUNT(ADV.Id\_action) AS nbrActions

FROM Pages P

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON A.idaction = ADV.Id\_action

JOIN Visites V ON V.Id\_visit = ADV.Id\_visit

JOIN Localisations L ON V.Location\_city = L.Id

WHERE ADV.Server\_time LIKE ‘%-12-%’

OR ADV.Server\_time LIKE ‘%-01-%’

OR ADV.Server\_time LIKE ‘%-02-%’

GROUP BY P.Type\_page, L.Region1;

Vue 2 : CREATE VIEW MaxActPageHiver AS

SELECT ActRegionHiver.Type\_page, MAX(ActRegionHiver.nbrActions) as nbrActMax

FROM ActRegionHiver

JOIN Pages P ON P.Type\_page = ActRegionHiver.Type\_page

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON ADV.Id\_action = A.idaction

WHERE ADV.Server\_time LIKE ‘%-12-%’

OR ADV.Server\_time LIKE ‘%-01-%’

OR ADV.Server\_time LIKE ‘%-02-%’

GROUP BY ActRegionHiver.Type\_page;

1. PRINTEMPS

Vue 1 : CREATE VIEW ActRegionPrintemps AS

SELECT P.type\_page, L.Region1, COUNT(ADV.Id\_action) AS nbrActions

FROM Pages P

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON A.idaction = ADV.Id\_action

JOIN Visites V ON V.Id\_visit = ADV.Id\_visit

JOIN Localisations L ON V.Location\_city = L.Id

WHERE ADV.Server\_time LIKE ‘%-03-%’

OR ADV.Server\_time LIKE ‘%-04-%’

OR ADV.Server\_time LIKE ‘%-05-%’

GROUP BY P.Type\_page, L.Region1;

Vue 2 : CREATE VIEW MaxActPagePrintemps AS

SELECT ActRegionPrintemps.Type\_page, MAX(ActRegionPrintemps.nbrActions) as nbrActMax

FROM ActRegionPrintemps

JOIN Pages P ON P.Type\_page = ActRegionPrintemps.Type\_page

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON ADV.Id\_action = A.idaction

WHERE ADV.Server\_time LIKE ‘%-03-%’

OR ADV.Server\_time LIKE ‘%-04-%’

OR ADV.Server\_time LIKE ‘%-05-%’

GROUP BY ActRegionPrintemps.Type\_page;

1. ETE

Vue 1 : CREATE VIEW ActRegionEte AS

SELECT P.type\_page, L.Region1, COUNT(ADV.Id\_action) AS nbrActions

FROM Pages P

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON A.idaction = ADV.Id\_action

JOIN Visites V ON V.Id\_visit = ADV.Id\_visit

JOIN Localisations L ON V.Location\_city = L.Id

WHERE ADV.Server\_time LIKE ‘%-06-%’

OR ADV.Server\_time LIKE ‘%-07-%’

OR ADV.Server\_time LIKE ‘%-08-%’

GROUP BY P.Type\_page, L.Region1;

Vue 2 : CREATE VIEW MaxActPageEte AS

SELECT ActRegionEte.Type\_page, MAX(ActRegionEte.nbrActions) as nbrActMax

FROM ActRegionEte

JOIN Pages P ON P.Type\_page = ActRegionEte.Type\_page

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON ADV.Id\_action = A.idaction

WHERE ADV.Server\_time LIKE ‘%-06-%’

OR ADV.Server\_time LIKE ‘%-07-%’

OR ADV.Server\_time LIKE ‘%-08-%’

GROUP BY ActRegionEte.Type\_page;

1. AUTOMNE

Vue 1 : CREATE VIEW ActRegionAutomne AS

SELECT P.type\_page, L.Region1, COUNT(ADV.Id\_action) AS nbrActions

FROM Pages P

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON A.idaction = ADV.Id\_action

JOIN Visites V ON V.Id\_visit = ADV.Id\_visit

JOIN Localisations L ON V.Location\_city = L.Id

WHERE ADV.Server\_time LIKE ‘%-09-%’

OR ADV.Server\_time LIKE ‘%-10-%’

OR ADV.Server\_time LIKE ‘%-11-%’

GROUP BY P.Type\_page, L.Region1;

Vue 2 : CREATE VIEW MaxActPageAutomne AS

SELECT ActRegionAutomne.Type\_page, MAX(ActRegionAutomne.nbrActions) as nbrActMax

FROM ActRegionAutomne

JOIN Pages P ON P.Type\_page = ActRegionAutomne.Type\_page

JOIN Actions A ON P.Id\_page = A.idpage

JOIN ActionsVisites ADV ON ADV.Id\_action = A.idaction

WHERE ADV.Server\_time LIKE ‘%-09-%’

OR ADV.Server\_time LIKE ‘%-10-%’

OR ADV.Server\_time LIKE ‘%-11-%’

GROUP BY ActRegionAutomne.Type\_page;