Test Business Analyst

Adopte - BI

2024

Consignes

Le fichier à rendre devra être sous format PDF.
Les tables user et purchase, permettant de répondre aux questions, sont disponibles dans une base de donnée BigQuery nommée tests via le lien : https://console.cloud.google.com/bigquery?project=aum-bi.Vouspouvezaccéderàlabasededonnéegrâceaumail Gmail que vous avez transmis. Vous pouvez requêter les tables pour vous aider à répondre aux questions.

Question 1 :
Décrivez brièvement les tables purchase et user.

La table de purchase enregistre les informations relatives aux achats effectués par les utilisateurs. Elle est composée de 4 colonnes et de 3946 lignes. Les colonnes sont :

- *account_id* en nombre qui représentent l' identifiant unique de l'utilisateur ayant réalisé l'achat ;
- *product* représenté sous forme textuelle et correspond au type de produit acheté ;
- *date* de l'achat, enregistrée sous forme numérique ;
- *amount* représenté sous forme numérique,
-

La table user contient les informations de profil des utilisateurs. Elle est composée de 3 colonnes et de 4578 lignes. Les colonnes sont :

- *accont_id* en nombre qui représentent l' identifiant unique de l'utilisateur ayant réalisé l'achat ;
- *birth_date* sous forme de date, permettant d'estimer l'âge ;
- *zip_code* sous forme numérique représentant le code postal de l'utilisateur.

Pour les questions suivantes, écrivez une rêquete SQL (pouvant s'exécuter sur BigQuery) permettant d'y répondre.
N'hésitezpasécrirevotreprocessusderéflexionetàêtrecritiqueavecvosrequêtes.

Question 2 :
Quels sont les IDs, le nombre et la somme des montants des achats des 5 plus gros consommateurs ?

```sql
SELECT account_id, count(*) as nombre_achats, sum(amount) as total_amount
from aum-bi.tests.purchase

group by account_id
order by total_amount desc
```

```
limit 5;
```

Ligne	account_id	nombre_achats	total_amount
1	22463	3	16997
2	23696	2	15998
3	20102	2	15998
4	20800	4	14996
5	20898	4	14996

Pour identifier les cinq plus gros consommateurs, j'ai conçu une requête SQL qui extrait les identifiants (account_id), le nombre total d'achats (nombre_achats) et la somme des montants (total_amount) pour chaque utilisateur dans la table des achats.

Question 3 :
Quel est l'âge moyen des 5 plus gros consommateurs au 1er novembre 2024?

```
With top5 as (SELECT account_id, count(*) as nombre_achats, sum(amount) as
total_amount
from aum-bi.tests.purchase

group by account_id
order by total_amount desc
limit 5)

SELECT avg(date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', birth_date), YEAR)) as
age_moyen
FROM aum-bi.tests.user as u
JOIN top5
ON u.account_id = top5.account_id;
```

Ligne	age_moyen
1	56.33333333333...

Pour déterminer l'âge moyen des cinq plus gros consommateurs au 1er novembre 2024 il faut d'abord utiliser une sous-requête (top5) pour sélectionner les cinq utilisateurs ayant le plus gros montant cumulé d'achats. Cela se fait en regroupant par account_id, en comptant le nombre d'achats et en totalisant les montants, puis en triant par total_amount de manière décroissante
Il faut ensuite faire une jointure entre *top5* et la table des utilisateurs. J'ai calculé l'âge moyen en années des cinq plus gros consommateurs en fonction de leur date de naissance, en utilisant date_diff entre le 1er novembre 2024 et birth_date.
L'âge moyen des 5 plus gros consommateurs au 1er novembre 2024 est de 56 ans.

Question 4 :
Quel est la somme des montants des 3 premiers achats des utilisateurs ayant fait au moins 3 achats ?
Cette extraction devra contenir les colonnes suivantes : account_id et total_amount.

```
With achat as (SELECT account_id, amount, row_number() over (partition by account_id
order by date asc) as achat_place
from aum-bi.tests.purchase)

Select account_id, sum(amount) as total_amount
from achat
where achat_place <= 3
group by account_id
having count(*) >= 3
```

Ligne	account_id	total_amount
1	20898	13997
2	23075	4997
3	23251	2997
4	24100	2997
5	24320	4997
6	21616	2997

Pour calculer la somme des montants des trois premiers achats pour les utilisateurs ayant effectué au moins trois achats, j'ai d'abord créé une sous-requête (achat) qui attribue un numéro d'ordre (achat_place) à chaque achat de chaque utilisateur, en utilisant ROW_NUMBER() partitionné par account_id et ordonné par date croissante (date). Cela permet de numéroter les achats du plus ancien au plus récent pour chaque utilisateur.
Ensuite, j'ai sélectionné les account_id et calculé la somme (total_amount) des montants pour les trois premiers achats (achat_place <= 3). La clause HAVING COUNT(*) >= 3 garantit que seuls les utilisateurs ayant effectué au moins trois achats sont inclus dans les résultats.

Question 5 :
Posez une question que vous trouvez pertinente et répondez-y grâce aux données.

La question est : Pour chaque produit, quelle tranche d'âge l'achète le plus.

```
WITH user_goupe_age as (select u.account_id, case
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) < 25
THEN '-25'
```

```
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) between
26 and 35 THEN '26-35'
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) between
36 and 45 THEN '36-45'
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) between
46 and 55 THEN '46-55'
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) between
56 and 65 THEN '56-65'
when date_diff(date '2024-11-01' , parse_date('%Y-%m-%d', u.birth_date), YEAR) > 65
THEN '66+'
else 'unknown'
end as groupe_age
from aum-bi.tests.user as u ),

age_groupe_purchase as (select p.product, uga.groupe_age, count(p.product) as
produit_total, row_number() over (partition by p.product order by count(product) desc)
as rang
from user_goupe_age as uga
join aum-bi.tests.purchase as p
on uga.account_id = p.account_id
group by p.product, uga.groupe_age)

SELECT product, groupe_age
from age_groupe_purchase
where rang = 1
order by product;
```

Ligne	product	groupe_age
1	product1	26-35
2	product2	46-55
3	product3	46-55

Pour déterminer la tranche d'âge qui achète le plus chaque produit au 1er novembre 2024, j'ai d'abord créé une sous-requête (user_goupe_age) pour assigner une tranche d'âge (groupe_age) aux utilisateurs selon leur date de naissance (birth_date). En fonction de l'écart d'années entre le 1er novembre 2024 et leur date de naissance, les utilisateurs sont classés dans des tranches d'âge prédéfinies (par exemple, '-25', '26-35', etc.).
Dans une deuxième sous-requête (age_groupe_purchase), j'ai joint les données de user_goupe_age avec celles de la table des achats (purchase). Cette requête calcule le nombre d'achats (produit_total) pour chaque produit et chaque tranche d'âge, puis utilise row_number() pour attribuer un rang aux tranches d'âge en fonction du nombre d'achats par produit (le plus élevé au rang 1).

Enfin, j'ai sélectionné pour chaque produit la tranche d’âge ayant le rang 1 (c’est-à-dire celle avec le plus d’achats).