Paris School of Business

Data-Rouj United

Créer une base de données relationnelles

Thomas Dahrouj - Nadir Boudjerida - Jean-louis Hu

Analyse et exploitation d’une Base de Données Relationnelle

Résumé et Mots-clés

Nous avons conçu une base de données relationnelle sur PostgreSQL pour analyser les comportements des utilisateurs d'une plateforme web. Cette base intègre des données variées, telles que les visiteurs, sessions, clics, termes de recherche et systèmes d'exploitation. Les tables ont été créées à l'aide de requêtes SQL et d'outils graphiques, avec des relations définies par des clés primaires et étrangères. Un nettoyage approfondi a permis d'améliorer la qualité des données, en corrigeant des anomalies comme les doublons et les incohérences temporelles. Ces données offrent un potentiel analytique important pour segmenter les utilisateurs, étudier les tendances et optimiser les stratégies marketing.

Mots-clés : PostgreSQL, données relationnelles, comportement utilisateur, analyse web, qualité des données, segmentation.

I - Introduction

Nous avons travaillé sur une base de données relationnelle pour analyser les comportements des utilisateurs sur une plateforme web. PostgreSQL a été utilisé pour gérer la structure relationnelle des données. Les jeux de données ont été importés, nettoyés, et organisés dans une base relationnelle. Des relations ont été établies entre les tables en utilisant des clés primaires et étrangères. Un processus en plusieurs étapes a été suivi : analyse des données initiales, définition des relations, construction des tables, et exécution de requêtes exploratoires.

II - Description des données

Cette analyse met en évidence les anomalies rencontrées dans les bases de données et propose des solutions pour améliorer leur qualité et leur cohérence.

Dans la table owa_visitor, on constate que certains champs essentiels, comme la localisation ou le type de dispositif, sont parfois absents. De plus, des identifiants de visiteurs sont dupliqués ou mal formatés, et certains utilisateurs enregistrent un nombre de sessions excessivement élevé, suggérant une activité automatisée (comme celle de bots).

Pour la table owa_session, plusieurs sessions sont associées à des identifiants de visiteurs inexistants. D’autres anomalies concernent des durées négatives ou irréalistes, causées par des erreurs dans les timestamps, ainsi que des doublons, ce qui complique l’analyse.

La table owa_click montre que de nombreux clics ne sont pas rattachés à des sessions valides, et certains horodatages ne correspondent pas aux plages horaires des sessions concernées. On trouve également des champs référents mal renseignés, pointant vers des documents inexistants.

Dans owa_search_term_dim, on observe des termes de recherche peu pertinents, comme des champs vides ou des termes génériques, ainsi que des occurrences disproportionnées qui pourraient refléter des biais dans la collecte.

Les URL référentes de la table owa_referer comportent souvent des erreurs de format, et une partie des enregistrements manque d'identifiants, limitant l'analyse des sources. Dans la table owa_action_fact, des actions ne sont pas associées à des sessions valides, et des incohérences temporelles, comme des clics avant le début des sessions, sont fréquentes.

Les tables owa_os et owa_ua contiennent de nombreuses valeurs génériques comme "Unknown OS" ou "Unknown Browser", ainsi que des doublons inutiles. La table owa_site, de son côté, souffre de données non standardisées ou incomplètes, avec des identifiants et des noms de site parfois vides ou erronés.

Les localisations géographiques enregistrées dans la table owa_location_dim sont parfois invalides ou redondantes. La table owa_queue_item inclut de nombreux éléments sans lien avec des données exploitables, ainsi que des informations obsolètes. Enfin, la table owa_document présente des documents non référencés dans les clics ou actions et des titres peu significatifs, comme "Document".

III - Construction de la base de données

Dans PgAdmin, il est possible de créer des tables en utilisant deux méthodes principales : l'écriture de requêtes SQL et l'utilisation de l'interface graphique. Ces deux approches permettent de structurer vos données tout en garantissant l'intégrité relationnelle. Voici une présentation détaillée de ces deux méthodes.

Faites un clic droit sur le dossier Table de votre base de données dans l'arborescence. Sélectionnez Create > Query Tool pour ouvrir l'éditeur SQL.

Voici un exemple de script pour créer une table appelée owa_id :

```
CREATE TABLE IF NOT EXISTS public.owa_id (
        id BIGINT,
        session_id BIGINT,
        nom VARCHAR(255),
        yyyymmdd INT
);
```

Ce script crée une table avec quatre colonnes : id (type BIGINT), session_id (type BIGINT), nom (type VARCHAR(255)), et yyyymmdd (type INT).

Pour garantir l'intégrité des données, ajoutez des contraintes directement dans le script SQL:

```
ALTER TABLE public.owa_id
```

ADD CONSTRAINT owa_id_pk PRIMARY KEY (id),
ADD CONSTRAINT session_fk FOREIGN KEY (session_id)
REFERENCES session (id)
ON UPDATE NO ACTION
ON DELETE NO ACTION;

Ces contraintes définissent la colonne id comme clé primaire unique et établissent une relation entre la colonne session_id de la table owa_id et la colonne id de la table session.

Une fois votre script complété, exécutez-le dans l'éditeur SQL. Actualisez la vue de l'explorateur dans PgAdmin. La table owa_id apparaîtra avec ses colonnes et contraintes bien définies.

Pour importer des données dans la table nouvellement créée, accédez à l'onglet Import Data via un clic droit sur la table owa_id. Utilisez l'option Upload pour charger un fichier local (CSV, par exemple). Sélectionnez et configurez les options d'importation pour que les données soient correctement insérées. Faites un clic droit sur le dossier Table de votre base de données. Sélectionnez Create > Table pour ouvrir l'éditeur graphique de création de table.

Dans l'onglet principal, saisissez le nom de la table (par exemple, owa_id). Ajoutez les colonnes une par une en spécifiant leurs types : id (type BIGINT, défini comme clé primaire), session_id (type BIGINT), nom (type VARCHAR(255)), et yyyymmdd (type INT).

Rendez-vous dans l'onglet Constraints. Ajoutez une contrainte Primary Key sur la colonne id. Configurez une contrainte Foreign Key pour relier la colonne session_id à la colonne id de la table session. Spécifiez les actions sur mise à jour et suppression (ON UPDATE NO ACTION, ON DELETE NO ACTION).

Une fois toutes les colonnes et contraintes configurées, cliquez sur Save. La table owa_id sera visible dans l'explorateur avec sa structure et ses relations correctement établies. Comme avec la méthode SQL, vous pouvez importer des données dans cette table via l'option Import Data accessible par clic droit sur la table.

La méthode SQL offre une flexibilité et un contrôle avancé sur la création des tables, ce qui est idéal pour les utilisateurs familiers avec le langage SQL. En revanche, l'interface graphique de PgAdmin est particulièrement utile pour les débutants ou pour les scénarios où une approche visuelle est préférable. Dans les deux cas, il est possible de définir des colonnes, d'ajouter des contraintes et d'importer des données avec précision. En combinant ces deux méthodes, vous pouvez ajuster vos processus en fonction des besoins spécifiques et de votre niveau de maîtrise de l'outil.

IV - Informations sur la Fiabilité, la Qualité et le Potentiel des Données

L'intégrité et la fiabilité des données ont été des priorités essentielles dans la conception de notre base relationnelle. Diverses vérifications ont été effectuées pour garantir la solidité et la cohérence des relations entre les tables. Chaque clé étrangère, telle que visitor_id ou session_id, a été systématiquement contrôlée afin de s'assurer qu'elle corresponde à une clé primaire valide dans la table parentale. Cette analyse a permis d'identifier des anomalies, comme des sessions sans visiteurs associés, qui ont ensuite été exclues ou corrigées. Une attention particulière a été portée à la cohérence temporelle en examinant les timestamps des tables owa_session et owa_click. Des incohérences, telles que des clics enregistrés avant le début de la session associée, ont été détectées, corrigées ou filtrées. Les doublons dans les tables critiques, notamment owa_visitor et owa_action_fact, ont été identifiés et supprimés pour éviter d'introduire des biais dans les analyses.

Pour améliorer la qualité des données et les rendre pleinement exploitables, divers nettoyages et transformations ont été appliqués. Les valeurs manquantes ont été traitées de manière stratégique : les enregistrements incomplets dans les champs critiques, tels que visitor_id ou session_id, ont été supprimés, tandis que des valeurs par défaut comme "Non défini" ou "Inconnu" ont été attribuées aux champs non critiques pour conserver les enregistrements potentiellement utiles. Les formats de données textuelles, comme les URL dans owa_referer ou les noms de documents dans owa_document, ont été uniformisés pour éviter les doublons causés par des variations mineures. Par ailleurs, des anomalies, telles que des sessions ayant des durées négatives ou des volumes de clics exceptionnellement élevés pour un seul visiteur, ont été identifiées comme des valeurs aberrantes. Ces données ont été marquées pour un examen approfondi ou exclu des analyses afin de maintenir la fiabilité des résultats.

Les données collectées offrent de nombreuses possibilités pour analyser les comportements des utilisateurs et améliorer les stratégies de la plateforme. Elles permettent de mener des analyses descriptives telles que la distribution des visiteurs par pays ou par type de dispositif, grâce aux informations contenues dans owa_location_dim et owa_os. Les termes de recherche les plus fréquents, disponibles dans owa_search_term_dim, fournissent des insights précieux sur les intentions des utilisateurs. Des analyses comportementales avancées sont également envisageables, comme l'étude des tendances horaires à partir des timestamps des clics et des sessions pour déterminer les moments de la journée où l'activité est la plus intense. En combinant les informations issues des tables owa_visitor, owa_os et owa_ua, il devient possible de segmenter les visiteurs selon leurs comportements et dispositifs. L'intégration des données de owa_referer, owa_click et owa_action_fact permet également d'évaluer quelles campagnes marketing ou contenus génèrent le plus d'engagements.

ANNEXES :