HMIN122M - Entrepôt de données

Data Warehouse Spotify

Joël BEYA NTUMBA Imrhan Dareine MINKO AMOA
Christophe QUENETTE Teiki RAIHAUTI

Jeudi 8 novembre 2018

Table des matières

1 Analyse

Spotify est un service suédois de streaming musical sous la forme d'un logiciel propriétaire et d'un site Web. Spotify permet une écoute quasi instantanée de fichiers musicaux.

Dans ce mini projet, nous allons nous intéresser aux questions fondamentales que se posent Spotify. Logiquement, ils chercheront à faire du profit et pour cela, ils vont chercher à augmenter la fréquentation de leur application. En effet, pour générer des revenus, ils vont faire payer leurs services de distribution aux utilisateurs. Donc maximiser le nombre d'abonnés va augmenter leur profit.

De plus, les employés de Spotify Technology vont pouvoir générer des revenus grâce aux publicités publiées sur leur application. Ils vont pouvoir observer les moments auquels une publicité est la plus efficace durant l'utilisation de l'application. Maximiser les revenus de la publicité est aussi un point essentiel car plus il y a de trafic, plus l'entreprise va pouvoir augmenter son chiffre d'affaire.

Spotify Technology ne crée pas de musiques mais diffuse celles d'artistes. Il va donc falloir attirer les artistes musicaux en leur proposant des revenus corrects, une application sécurisée limitant au maximum le piratage.

Voici une liste non exhaustive des points à étudier pour Spotify Technology :

- Les musiques que veulent écouter les utilisateurs : les genres musicaux, la période de sortie, catégories, etc.
- Les artistes qui sont les plus recherchés par styles musicaux, par langues, pays, tendances ...
- Le thème des playlists qu'ils proposeront : analyser les écoutes d'une playlist par rapport au genre musical, à une année ou une période, un sentiment ou une émotion, le pays d'origine, les artistes auteurs des titres ...
- L'évolution de l'affluence des artistes suite à une mise à jour concernant le piratage et le respect des droits d'auteurs, ou à une mise à jour facilitant le processus de publication de contenu musical
- L'évolution de l'affluence des artistes (ou de la fréquence de publication de leurs contenus musicaux) lors d'un changement concernant leur rémunération.
- Le prix mensuel qu'est prêt à payer un utilisateur pour écouter de la musique en analysant (par exemple) la part des utilisateurs qui ont souscrit à un abonnement payant en fonction de leur âge, de leur pays d'origine ...
- L'évolution de l'affluence des utilisateurs (artistes et auditeurs) suite à un changement au niveau du design et de l'ergonomie des applications
- Les moyens de promotion et leur impact sur la venue des auditeurs. Par exemple promouvoir les services par des publicités selon divers paramètres tels que : passer des publicités durant les heures des pointes ou durant certains moments propices comme les fins de mois où la population active touchent leurs salaires, les débuts d'années universitaires pour les étudiants ou les débuts de vacances pour les familles ... et observer l'impact sur la fréquentation.

Les points énoncés précédemment nous permettent d'établir une liste des actions qui sont les plus importantes pour Spotify.

2 Traitements

2.1 Actions/opérations importantes pour Spotify

- L'écoute d'une musique (appelée aussi un stream). Seuls les streams durant plus de 30 secondes sont comptabilisées dans le nombre d'écoutes d'une musique.
  L'écoute est une action importante car c'est celle qui permet de quantifier l'intérêt du public pour l'application. Augmenter le nombre d'écoutes est donc le but constamment recherché par l'entreprise.
- L'abonnement à une option : un abonnement à la plateforme pour les utilisateurs avec des offres diverses et sans engagement par exemple:
  - une inscription gratuite;
  - une offre Premium etudiant/abonnement jeune;
  - une offre Famille, avec possibilité d'utiliser à plusieurs un même compte abonné;
  - La résiliation d'un abonnement payant : Spotify ne met pas en place d'engagement pour les abonnements donc un utilisateur peut se désabonner à tout moment
  - ...
- Publication de contenu musical par un artiste : uploader ses musiques sur la plateforme, possibilité de faire de la publicité pour promouvoir sa musique, consulter les statistiques d'écoute ou d'abonnement à sa page.

2.2 Trois traitements possibles permettant d'aider à la prise de décision pour chaque action/opération

- Pour l'écoute d'une musique :
  1. Combien de secondes en moyenne une musique a été écoutée
  2. Le pays d'origine des auditeurs
  3. Les musiques les plus écoutées pour une période donnée
- Pour l'abonnement d'un utilisateur :
  1. Les publicités qui l'ont incité soit à s'abonner de façon gratuite, soit à passer à une option payante
  2. Le rapport entre les abonnements classiques et les abonnements bénéficiant d'une offre promotionnelle
  3. A quel moment de l'année il y a le plus de résiliations
- Pour la publication de contenu musical :
  1. Combien de fois un artiste à t-il consulté les statistiques (nombre d'écoutes, pourcentage de gain ou perte d'abonnés ...) durant une période
  2. A quelle période y a-t-il le plus de publications
  3. Nombre d'écoutes le jour de la publication d'un contenu musical

2.3 Actions par ordre d'importance ou de rentabilité potentielle.

1. **L'écoute d'une musique** est la tâche la plus importante. En effet, c'est l'activité principale de Spotify et c'est celle qui va attirer les utilisateurs à souscrire un abonnement.
2. **L'abonnement** est le deuxième point car c'est la principale source de revenus pour Spotify.
3. Ensuite, **le nombre d'artistes présents** est important car Spotify va avoir la capacité de répondre à un maximum de recherches d'artistes. Cela leur permet d'être la source musicale dans le plus de cas possibles et donc faire en sorte que l'utilisateur ne quitte pas Spotify pour une application tierce (comme YouTube ou encore SoundCloud).

La conception de l'entrepôt de données dépendra des traitements indiqués ci-dessus.

3 Conception

3.1 Identifiez les deux actions les plus importantes à analyser.

- **L'écoute d'une musique** est une des actions les plus importantes. En effet, c'est l'activité principale et donc nous l'utiliserons comme l'action la plus prioritaire.
- **L'abonnement à une option** est aussi une action d'importance primordiale étant donné que c'est (dans le cas d'un abonnement payant) l'action rémunératrice pour l'entreprise.

3.2 Pour chaque action, concevez un data-mart indépendant

- **Les écoutes sur Spotify (les streams)**

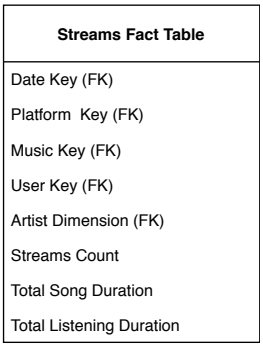

Figure 1: Table de fait d'une écoute Spotify

Faits :

- Le nombre de streams (mesure additive)
- La durée totale d'écoute (mesure semi-additive)
- La durée totale des musiques écoutées (mesure semi-additive)

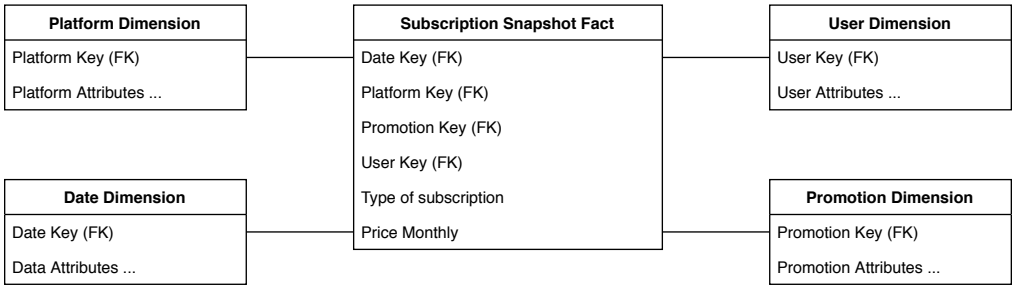

Figure 2: Table de fait d'abonnement à Spotify (Inscription)

- **L'abonnement à Spotify**

  Granularités :

  - Transaction unitaire d'abonnement (souscription à d'abonnement payant)
  - Snapshot des abonnement mensuels payant
  - Record-update d'un type d'abonnement mensuel payant (par exemple un abonnement famille)

  Faits:

  - Le nombre d'abonnements par mois (mesure additive)

3.3 Cinq dimensions et une dizaine d'attributs minimum pour chaque action

Choix de conception :

Dimension temps:

Stocker l'information de la période dans une journée était indispensable pour les streams. En effet elle est nécessaire pour pouvoir analyser les écoutes des utilisateurs en fonction des moments de la journée.
Nous avons décidé d'ajouter une dimension Temps au datamart au lieu d'ajouter des attributs à la dimension Date déjà existante. Cela nous aurait négatif notamment pour des raisons de performances : une journée contient 24 heures, ainsi une fois crée, le nombre de tuples de la dimension Temps sera 86 400 (60 secondes x 60 minutes x 24 heures). Si tous les attributs étaient rassemblés dans la dimension Date, la quantité de tuples pour une année serait 31 536 000 (60 secondes x 60 minutes x 24 heures x 365 jours) au lieu de 365 tuples.

Écoute :

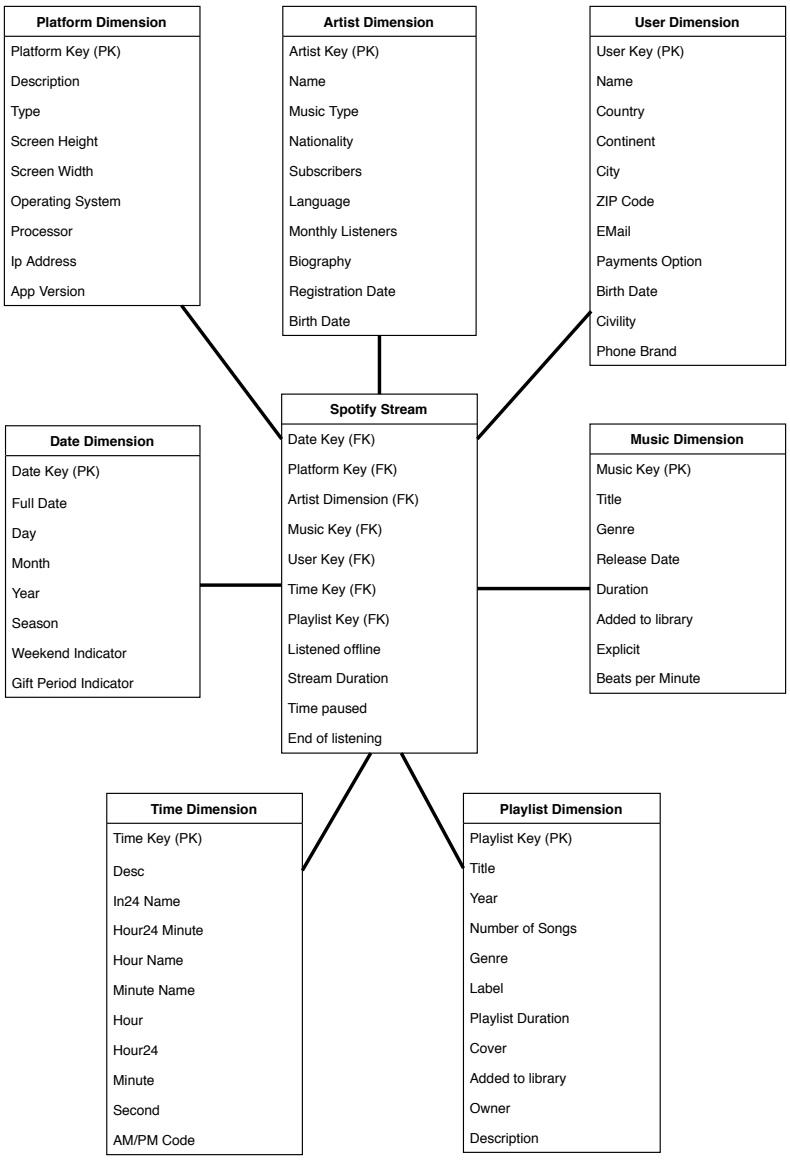

Figure 3: Dimension des écoutes

Abonnement :

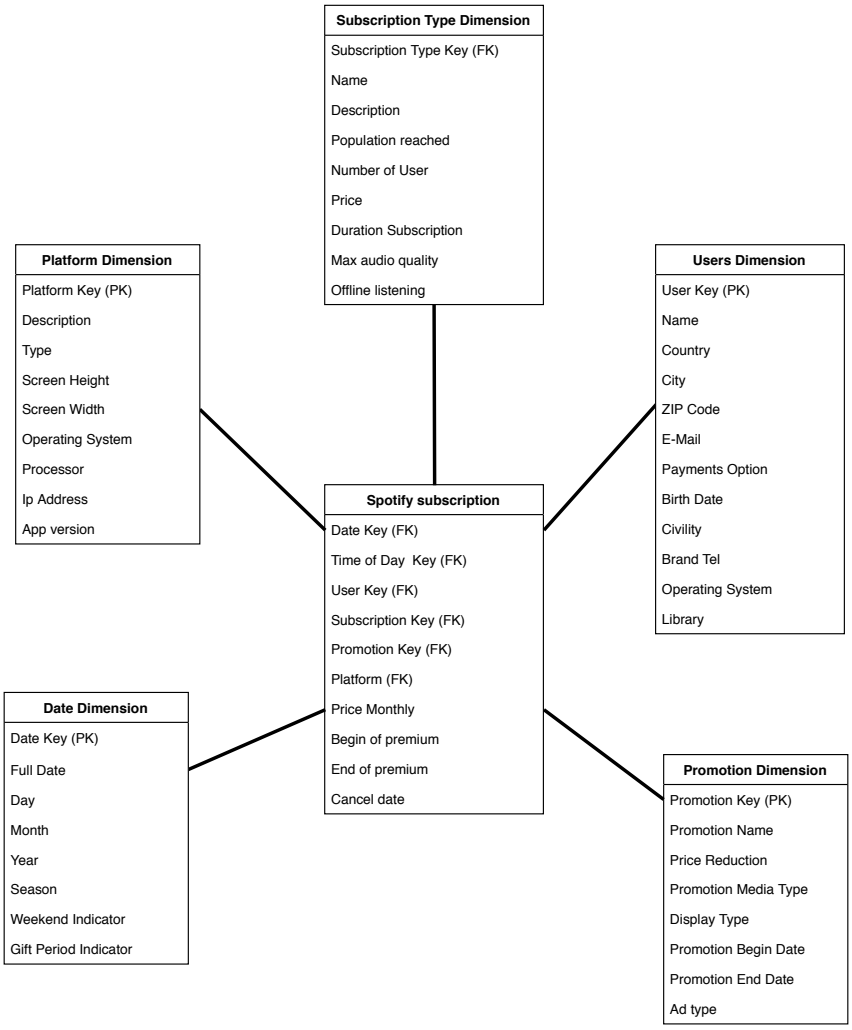

Figure 4: Dimension des abonnements

3.4 Est-il possible de répondre aux traitements que vous avez indiqué avec le modèle que vous avez mis en place ?

Chaque traitement que nous avons décrit est facilement retrouvable par des requêtes `SELECT` concernant nos 2 principales actions qui sont l'écoute et la souscription d'un abonnement.

Par exemple, pour la requête: "Combien de secondes en moyenne une musique a été écoutée", il suffira de faire :

```
SELECT AVG(total_listening_duration)
FROM streams_fact_table, music
WHERE streams_fact_table.music_id = music.music_id
GROUP BY music_id;
```

3.5 Instance de l'entrepôt de données

User	Artist	Music	Playlist	Platform	Season	Day Part Segment	Listened offline	Stream Duration (s)	Time Paused (s)	Stream count	Total song duration (s)	Total Listening Duration (s)
Ford Ferguson	John Coltrane	Naima	Giant Steps (Deluxe Edition)	PC	Autumn	Morning	YES	00:03:30	00:00:15	4000000	840000000	2,88E+15
Mario Price	Ed Sheeran	Shape of You	÷ (Deluxe)	Mobile	Autumn	Evening	NO	00:02:30	00:00:00	25000000	3,75E+09	1,13E+17
Didier Drogba	John Williams	Hedwig's Theme	Harry Potter and The Sorcerer's Stone	Web	Autumn	Morning	NO	00:03:00	00:00:00	10000000	1,8E+09	1,80E+16

User	Type of subscription	Platform	Promotion Name	Season	Day Part Segment	Price Monthly (€)	Begin of Premium	End of Premium	Cancel Date
John Doe	Student	Mobile	Discover Promotion	Winter	Morning	4,99	13/01/2018	13/04/2018	NULL
Emma Sting	Normal	Mobile	NON	Summer	Afternoon	9,99	10/07/2015	10/04/2018	NULL
Arya Stark	Free	PC	NON	Summer	Morning	0	NULL	NULL	10/10/2017

3.6 Estimation de la taille des tables de l'entrepôt sur 12 mois

Pour faire une estimation de la taille des tables de l'entrepôt de données, nous allons nous intéresser au nombre d'écoutes et d'abonnements (gratuit ou non) sur une année. Pour cela, nous utiliserons les chiffres communiqués par Spotify.
L'entreprise estime pouvoir dépasser les 200 millions d'utilisateurs avant 2019. En sachant qu'ils avaient 160 millions d'utilisateurs début 2018, ils connaîtront une augmentation de 40 millions d'utilisateurs en un an ce qui correspond à 40 millions de lignes dans l'entrepôt.

Nous pouvons estimer le nombre d'écoutes sur une année simplement avec un calcul :

- Environ 180 millions d'utilisateurs actifs en moyenne sur un an
- 15 musiques écoutées en moyenne par utilisateur et par jour
- En considérant une année de 365 jours

On obtient environ mille milliards d'écoutes sur toute une année.
C'est une taille qui semble raisonnable pour une entreprise de cette ampleur car en effet, une ligne dans la table de faits des écoutes Spotify et des inscriptions prend environ 50 octets. Cela fait environ 50 To de données à stocker sur 12 mois dans l'entrepôt.

4 Implémentation

4.1 Requêtes analytiques

- Connaître le temps moyen d'écoute d'une musique
- Connaître les types de musique les plus écoutés suivant les régions du globe
- Connaître le nombre d'utilisateurs écoutant un certain genre de playlist suivant l'heure de la journée
- Connaître les musiques les plus écoutées sur les dernières 24 heures
- Connaître les genres musicaux préférés de chaque utilisateur grâce à ces derniers streams pour proposer des playlist similaires
- Le prix moyen dépensé par mois par les utilisateurs écoutant un grande quantité de musique (plus de 30) par jour
- Analyser quelles sont les promotions les plus efficaces grâce au nombre d'abonnements suivant une promotion donnée
- Analyser les promotions menant le plus souvent à un abonnement premium suivant l'âge d'un utilisateur
- Connaître les types d'abonnements souscrits le plus fréquemment en fonction du type de plateforme utilisé
- Analyser le nombre de souscription à une offre Premium à l'approche des périodes de fête

4.2 Donnez l'ensemble des vues matérialisées permettant de répondre à l'ensemble des requêtes

```
-- Connaître le temps moyen d'écoute d'une musique
DROP MATERIALIZED VIEW MV_AVG_LTM;
CREATE MATERIALIZED VIEW MV_AVG_LTM
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT TITLE, AVG(STREAM_DURATION) FROM SPOTIFY_STREAM, MUSIC_DIM
WHERE SPOTIFY_STREAM.MUSIC_KEY = MUSIC_DIM.MUSIC_KEY
GROUP BY TITLE;

-- Connaître les types de musique les plus écoutés suivant la région du globe
DROP MATERIALIZED VIEW MV_LMTR;
CREATE MATERIALIZED VIEW MV_LMTR
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT CONTINENT, COUNTRY, GENRE, TITLE, SUM(STREAM_DURATION) AS "streams_count"
FROM USER_DIM, MUSIC_DIM, SPOTIFY_STREAM
WHERE USER_DIM.USER_KEY = SPOTIFY_STREAM.USER_KEY
AND MUSIC_DIM.MUSIC_KEY = SPOTIFY_STREAM.MUSIC_KEY
GROUP BY CONTINENT, COUNTRY, GENRE, TITLE;

-- Connaître le nombre d'utilisateurs écoutant un certain genre
--de playlist suivant l'heure de la journée
DROP MATERIALIZED VIEW MV_ULKT;
CREATE MATERIALIZED VIEW MV_ULKT
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT TITLE, GENRE, PLAYLIST_DIM.YEAR, MONTH, DAY, COUNT(SPOTIFY_STREAM.USER_KEY)
FROM DATE_DIM, SPOTIFY_STREAM, USER_DIM, PLAYLIST_DIM
WHERE DATE_DIM.DATE_KEY = SPOTIFY_STREAM.DATE_KEY
AND USER_DIM.USER_KEY = SPOTIFY_STREAM.USER_KEY
AND PLAYLIST_DIM.PLAYLIST_KEY = SPOTIFY_STREAM.PLAYLIST_KEY
GROUP BY GENRE, TITLE, PLAYLIST_DIM.YEAR, MONTH, DAY;

-- Connaître les premiers musiques les plus écoutées sur les dernières 24 heures
```

```sql
DROP MATERIALIZED VIEW MV_MLM;
CREATE MATERIALIZED VIEW MV_MLM
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
DISABLE QUERY REWRITE
AS
SELECT TITLE, COUNT(STREAM_DURATION)
FROM SPOTIFY_STREAM, TIME_DIM, MUSIC_DIM
WHERE TIME_DIM.TIME_KEY = SPOTIFY_STREAM.DATE_KEY
AND MUSIC_DIM.MUSIC_KEY = SPOTIFY_STREAM.MUSIC_KEY
AND TIME_HOUR BETWEEN EXTRACT(HOUR FROM CURRENT_TIMESTAMP)
AND (EXTRACT(HOUR FROM CURRENT_TIMESTAMP) - 1)
GROUP BY TITLE;
-- --
-- -- -- Connaître les genres musicaux préférés de chaque utilisateur
--grâce à ces derniers streams pour proposer des playlist similaires
DROP MATERIALIZED VIEW MV_UMGP;
CREATE MATERIALIZED VIEW MV_UMGP
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
DISABLE QUERY REWRITE
AS
SELECT NAME, EMAIL, PLAYLIST_DIM.TITLE, PLAYLIST_DIM.GENRE, SUM(STREAM_DURATION)
FROM USER_DIM, SPOTIFY_STREAM, DATE_DIM, PLAYLIST_DIM
WHERE USER_DIM.USER_KEY = SPOTIFY_STREAM.USER_KEY
AND DATE_DIM.DATE_KEY = SPOTIFY_STREAM.DATE_KEY
AND PLAYLIST_DIM.PLAYLIST_KEY = SPOTIFY_STREAM.PLAYLIST_KEY
AND DAY BETWEEN EXTRACT(DAY FROM CURRENT_DATE) AND (EXTRACT(DAY FROM CURRENT_DATE) - 7)
GROUP BY NAME, EMAIL, TITLE, GENRE;
--
-- -- Analyser quelles sont les promotions les plus efficaces grâce au
--nombre d'abonnements suivant une promotion donnée
DROP MATERIALIZED VIEW MV_VOEP;
CREATE MATERIALIZED VIEW MV_VOEP
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT PROMOTION_NAME, SUM(NUMBER_OF_USER)
FROM USER_DIM, SPOTIFY_SUBSCRIPTION_DIM, PROMOTION_DIM, SUBSCRIPTION_TYPE_DIM
WHERE USER_DIM.USER_KEY = SPOTIFY_SUBSCRIPTION_DIM.USER_KEY
AND PROMOTION_DIM.PROMOTION_KEY = SPOTIFY_SUBSCRIPTION_DIM.PROMOTION_KEY
AND SUBSCRIPTION_TYPE_DIM.SUBSCRIPTION_TYPE_KEY = SPOTIFY_SUBSCRIPTION_DIM.SUBSCRIPTION_TYPE_KEY
```

```
GROUP BY PROMOTION_NAME;
--
-- -- Analyser les promotions menant le plus souvent à un
--abonnement premium suivant l'âge d'un utilisateur
DROP MATERIALIZED VIEW MV_PRUA;
CREATE MATERIALIZED VIEW MV_PRUA
BUILD IMMEDIATE
REFRESH COMPLETE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT BIRTH_DATE, PROMOTION_NAME, COUNT(SPOTIFY_SUBSCRIPTION_DIM.SUBSCRIPTION_TYPE_KEY)
FROM PROMOTION_DIM, SPOTIFY_SUBSCRIPTION_DIM, USER_DIM
WHERE PROMOTION_DIM.PROMOTION_KEY = SPOTIFY_SUBSCRIPTION_DIM.PROMOTION_KEY
AND SPOTIFY_SUBSCRIPTION_DIM.USER_KEY = USER_DIM.USER_KEY
GROUP BY BIRTH_DATE, PROMOTION_NAME;
```