

1. Identification du SGBD

Nom: BigQuery

Date parution: 19 Mai 2010

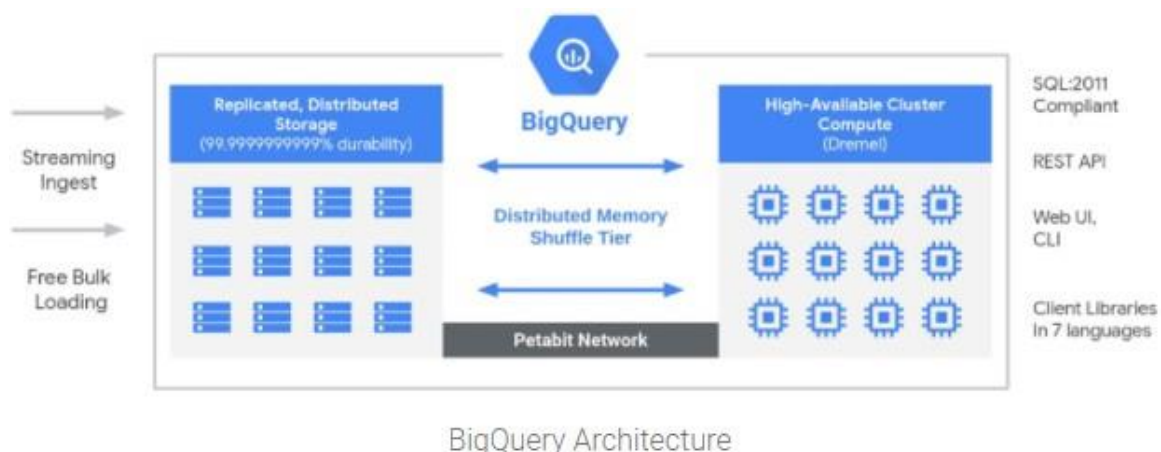
Créateur: Google

Propriétaire: Google

Modèles de données supportées: SQL

2. Architecture fonctionnelles du SGBD choisi

Dessins d'architecture:



Les caches mémoires et leurs rôles:

<https://blog.ippon.fr/2021/10/13/optimiser-les-couts-bigquery/>

Le cache mémoire interne de BigQuery est une fonctionnalité que l'on retrouve dans une grande majorité des DataWarehouse managés et son implémentation varie selon la solution que l'on utilise.

Le cache mémoire de BigQuery est une implémentation très simple, lorsqu'une requête est lancée par un utilisateur, le résultat de la requête va être stocké dans ce cache mémoire pendant 24H. Ainsi si n'importe quel autre utilisateur va lancer une requête identique, BigQuery va comprendre qu'il n'a pas besoin de lire son stockage interne et de refaire toutes les transformations. Il va lire le résultat de la requête stocké dans le cache, puis l'afficher à l'utilisateur, cette opération est extrêmement rapide et n'est pas facturée à l'utilisateur.

Cependant, pour pouvoir bénéficier du cache, plusieurs conditions doivent être réunies :

- 1 - La donnée de la table source n'a pas changée
- 2 - La requête n'utilise pas de fonctions non-déterministe comme NOW()
- 3 - Le cache mémoire est activé sur le résultat de la requête

Gestion des connexions: Pour exécuter une tâche de requête, il faut l'autorisation IAM (Identify and Access Management).

Par défaut, autoriser l'accès à un projet permet également d'accéder aux ensembles de données qu'il contient. L'accès par défaut peut être ignoré pour chaque ensemble de données. Le tableau suivant décrit les accès accordés aux membres des rôles IAM de base.

Rôle de base	Fonctions
Viewer	<ul style="list-style-type: none">• Peut démarrer une tâche dans le projet. Des rôles supplémentaires pour les ensembles de données peuvent être nécessaires en fonction du type de tâche.• Peut répertorier et obtenir toutes les tâches, et mettre à jour celles qui ont été démarrées pour le projet.• Si vous créez un ensemble de données dans un projet pour lequel des lecteurs sont définis, BigQuery attribue à ces utilisateurs le rôle prédéfini bigquery.dataViewer dans le nouvel ensemble de données.
Editor	<ul style="list-style-type: none">• Compétences identiques à celles de Viewer, plus les suivantes :<ul style="list-style-type: none">• Peut créer un ensemble de données dans le projet.• Si vous créez un ensemble de données dans un projet pour lequel des éditeurs sont définis, BigQuery attribue à ces utilisateurs le rôle prédéfini bigquery.dataEditor dans le nouvel ensemble de données.
Owner	<ul style="list-style-type: none">• Compétences identiques à celles de Editor, plus les suivantes :<ul style="list-style-type: none">• Peut révoquer ou modifier n'importe quel rôle de projet• Peut répertorier tous les ensembles de données du projet.• Peut supprimer n'importe quel ensemble de données du projet.• Peut répertorier et obtenir toutes les tâches exécutées dans le projet, y compris celles qui sont exécutées par d'autres utilisateurs du projet.• Si vous créez un ensemble de données, BigQuery attribue à tous les propriétaires de projets le rôle prédéfini bigquery.dataOwner dans le nouvel ensemble de données. <p>Exception : Lorsqu'un utilisateur exécute une requête, un ensemble de données anonyme est créé pour stocker la table des résultats mis en cache. Seul l'utilisateur qui exécute la requête peut avoir l'accès OWNER à l'ensemble de données anonyme.</p>

Processus d'exécution des requêtes:

Pour exécuter une [tâche](#) de requête, vous devez disposer de l'autorisation IAM (Identity and Access Management) `bigquery.jobs.create` pour le projet qui exécute la tâche de requête.

Chacun des rôles IAM prédéfinis suivants inclut les autorisations dont vous avez besoin pour exécuter une tâche de requête :

- roles/bigquery.admin
- roles/bigquery.jobUser
- roles/bigquery.user

En outre, vous avez besoin de l'autorisation IAM bigquery.tables.getData pour toutes les tables ou vues auxquelles la requête fait référence.

Chacun des rôles IAM prédéfinis suivants inclut les autorisations dont vous avez besoin pour interroger une table :

- roles/bigquery.dataEditor
- roles/bigquery.dataOwner
- roles/bigquery.dataViewer

Pour en savoir plus sur les rôles IAM dans BigQuery, consultez la page [Rôles prédéfinis et autorisations](#).

Exécuter des requêtes interactives

Par défaut, BigQuery exécute des tâches de requête interactives, ce qui signifie que la requête est exécutée dès que possible. Les requêtes interactives sont comptabilisées dans [votre limite de débit simultanée et votre limite quotidienne](#).

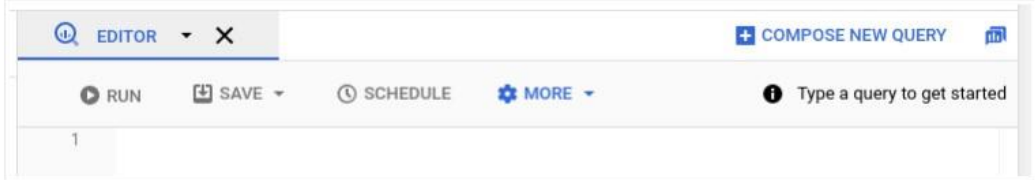
Pour exécuter une requête interactive qui écrit dans une table temporaire, procédez comme suit :

Console bq API C# Go Java Node.js PHP Python Ruby

1. Dans Cloud Console, ouvrez la page "BigQuery".

Accéder à BigQuery

2. Cliquez sur **Saisir une nouvelle requête**.



3. Saisissez une requête SQL BigQuery valide dans la zone de texte de l'**éditeur de requête**.

4. (Facultatif) Pour modifier l'emplacement de traitement des données, cliquez sur **Plus**, puis sur **Paramètres de requête**. Dans le champ **Zone de traitement**, cliquez sur **Sélection automatique** et choisissez l'**emplacement** de vos données. Cliquez ensuite sur **Enregistrer** pour mettre à jour les paramètres de la requête.

5. Cliquez sur **Run** (Exécuter).

Cette action crée une tâche de requête qui écrit les résultats dans une table temporaire.

BigQuery propose également des requêtes par lot. Il met en file d'attente chaque requête par lot en votre nom et en lance une dès que des ressources inactives sont disponibles dans le pool de ressources partagées de BigQuery. Cela prend généralement quelques minutes. Si BigQuery n'a pas lancé la requête dans les 24 heures, il redéfinit la priorité de la tâche sur interactive.

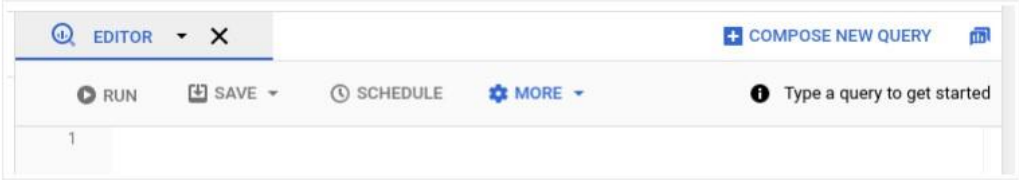
Les requêtes par lot ne sont pas comptabilisées dans votre limite de débit simultané, ce qui peut faciliter le lancement de plusieurs requêtes à la fois. Elles utilisent les mêmes ressources que les requêtes interactives (à la demande). Si vous avez opté pour un [tarif forfaitaire](#), les requêtes par lot et les requêtes interactives partagent les emplacements attribués.

Console bq API Go Java Node.js Python

1. Dans Cloud Console, ouvrez la page "BigQuery".

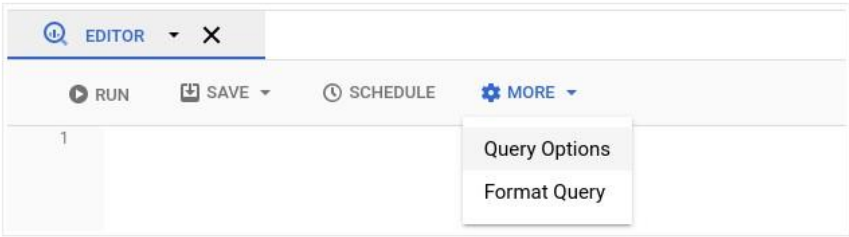
[Accéder à BigQuery](#)

2. Cliquez sur le bouton **Saisir une nouvelle requête**.

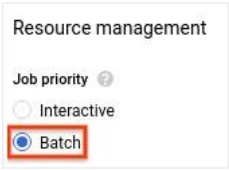


3. Saisissez une requête SQL valide dans la zone de texte de l'**éditeur de requête**.

4. Cliquez sur le bouton **More** (Plus), puis sur **Query settings** (Paramètres de la requête).



5. Sélectionnez l'option **Batch** (Lot) dans la section **Job priority** (Priorité de la tâche).



6. (Facultatif) Dans le champ **Zone de traitement**, cliquez sur **Non spécifiée** et sélectionnez l'**emplacement** de vos données.

7. Cliquez sur **Enregistrer** pour mettre à jour les paramètres de la requête.

8. Cliquez sur **Run** (Exécuter).

3. Dictionnaire de données Big Query

Un ensemble de données est ajouté au niveau de chaque vue Analytics pour laquelle l'intégration BigQuery est activée. Son nom correspond à l'ID de la vue correspondante.

Dans chaque ensemble de données, un tableau est importé pour chaque jour d'exportation. Ces tableaux quotidiens utilisent le format "ga_sessions_AAAAMMJJ".

Les données intrajournalières sont importées au moins trois fois par jour. Les tableaux intrajournaliers utilisent le format "ga_sessions_intraday_AAAAMMJJ". Au cours d'une même journée, chaque importation de données intrajournalières remplace l'importation précédente dans le même tableau.

À la fin de l'importation quotidienne, le tableau intrajournalier de la journée précédente est supprimé. Il n'existe pas de tableau intrajournalier pour la date du jour tant qu'une première importation de données intrajournalières n'a pas eu lieu. En cas d'échec d'écriture d'un tableau intrajournalier, le tableau du jour précédent est conservé.

Les données de la date du jour ne sont pas définitives tant que l'importation quotidienne n'est pas terminée. Vous constaterez peut-être quelques différences entre les données intrajournalières et les données quotidiennes reposant sur des sessions d'utilisateurs actifs qui ont eu lieu en partie durant la dernière importation intrajournalière.

Nom du champ	Type de données	Description
<code>clientId</code>	CHAÎNE	Version non hachée de l'ID client pour un utilisateur donné, associé à une visite/session donnée.
<code>fullVisitorId</code>	CHAÎNE	ID de visiteur unique
<code>visitorId</code>	VALEUR NULLE	Ce champ a été abandonné. Utilisez plutôt "fullVisitorId".
<code>userId</code>	CHAÎNE	User ID remplacé envoyé à Analytics.
<code>visitNumber</code>	NOMBRE ENTIER	Numéro de session associé à cet utilisateur. S'il s'agit de la première session, la valeur utilisée est 1.
<code>visitId</code>	NOMBRE ENTIER	Identifiant défini pour cette session. Il fait partie de la valeur généralement stockée sous la forme du cookie <code>_utmb</code> . Il est propre à cet utilisateur. Pour obtenir un identifiant parfaitement unique, combinez les valeurs

		des champs fullVisitorId et visitId.
<code>visitStartTime</code>	NOMBRE ENTIER	Horodatage (spécifié au format POSIX).
<code>date</code>	CHAÎNE	Date de la session au format AAAAMMJJ.
<code>totals</code>	ENREGISTREMENT	Cette section contient les valeurs cumulées de la session.
<code>totals.bounces</code>	NOMBRE ENTIER	Nombre total de rebonds (pour information). Pour une session avec rebond, la valeur est de 1. Sinon, elle est nulle.
<code>totals.hits</code>	NOMBRE ENTIER	Nombre total d'appels durant la session.
<code>totals.newVisits</code>	NOMBRE ENTIER	Nombre total de nouveaux utilisateurs durant la session (pour information). S'il s'agit de la première visite, la valeur est égale à 1. Sinon, elle est nulle.
<code>totals.pageviews</code>	NOMBRE ENTIER	Nombre total de pages vues au cours de la session.
<code>totals.screenviews</code>	NOMBRE ENTIER	Nombre total de vues de l'écran au cours de la session.

<code>totals.sessionQualityDim</code>	NOMBRE ENTIER	Estimation indiquant dans quelle mesure une session spécifique était proche de générer une transaction. Ce chiffre est calculé pour chaque session. Il est compris entre 1 et 100. Une valeur proche de 1 indique une faible qualité de session ou une session qui n'était pas en passe de générer une transaction, tandis qu'une valeur proche de 100 indique une qualité de session élevée ou une session qui était tout près de générer une transaction. Une valeur de 0 indique que la qualité de la session n'est pas calculée pour la plage de dates sélectionnée.
---------------------------------------	---------------	--

<code>totals.timeOnScreen</code>	NOMBRE ENTIER	Durée totale à l'écran (en secondes).
----------------------------------	---------------	---------------------------------------

<code>totals.timeOnSite</code>	NOMBRE ENTIER	Durée totale de la session (en secondes).
--------------------------------	---------------	---

<code>totals.totalTransactionRevenue</code>	NOMBRE ENTIER	Chiffre d'affaires total des transactions. Il correspond à la valeur transmise à Analytics, multipliée par 10 ⁶ (par exemple, la valeur 2,40 est restituée sous la forme "2 400 000").
---	---------------	---

<code>totals.transactionRevenue</code>	NOMBRE ENTIER	Ce champ a été abandonné. Utilisez plutôt "totals.totalTransactionRevenue" (voir ci-dessus).
--	---------------	--

<code>totals.transactions</code>	NOMBRE ENTIER	Nombre total de transactions e-commerce au cours de la session.
----------------------------------	---------------	---

<code>totals.UniqueScreenViews</code>	NOMBRE ENTIER	Nombre total de vues uniques de l'écran au cours de la session.
<code>totals.visits</code>	NOMBRE ENTIER	Nombre de sessions (pour information). Cette valeur correspond à 1 pour les sessions comprenant des événements d'interaction. Elle est nulle en l'absence d'interactions.
<code>trafficSource</code>	ENREGISTREMENT	Cette section contient des informations sur la source de trafic ayant généré la session.
<code>trafficSource.adContent</code>	CHAÎNE	Contenu de l'annonce associée à la source de trafic. Il peut être défini par le paramètre d'URL <code>utm_content</code> .
<code>trafficSource.adwordsClickInfo</code>	ENREGISTREMENT	Cette section contient les informations sur les clics Google Ads éventuellement associés à cette session. Analytics utilise le modèle "Dernier clic non direct".
<code>trafficSource.adwordsClickInfo.adGroupId</code>	NOMBRE ENTIER	ID du groupe d'annonces Google.
<code>trafficSource.adwordsClickInfo.adNetworkType</code>	CHAÎNE	Type de réseau. Adopte l'une des valeurs suivantes : {"Google Search", "Content", "Search partners", "Ad Exchange", "Yahoo Japan Search", "Yahoo Japan AFS", "unknown"}
<code>trafficSource.adwordsClickInfo.adCampaignId</code>	NOMBRE ENTIER	ID de la campagne Google Ads.

```
adwordsClickInfo.campaignId
```

```
trafficSource.
```

 NOMBRE ENTIER ID Google Ads.

```
adwordsClickInfo.creativeId
```

```
trafficSource.
```

 NOMBRE ENTIER Il s'agit de l'identifiant du critère de ciblage.

```
adwordsClickInfo.criteriaId
```

```
trafficSource.
```

 CHAÎNE Il s'agit de la chaîne descriptive du critère de ciblage.

```
adwordsClickInfo.criteriaParameters
```

```
trafficSource.
```

 NOMBRE ENTIER Numéro client Google Ads.

```
adwordsClickInfo.customerId
```

4. Organisation physique du SGBD

BigQuery stocke les données en interne dans un format de colonne propriétaire appelé Capacitor, qui présente de nombreux avantages pour les charges de travail de l'entrepôt de données. BigQuery s'appuie sur un format propriétaire, car il peut évoluer en même temps que le moteur de requête qui tire parti d'une connaissance approfondie de la structure des données pour optimiser l'exécution des requêtes. BigQuery détermine le nombre optimal de fragments physiques et leur encodage à l'aide de modèles d'accès aux requêtes.

Les données sont stockées sur le système de fichiers distribué de Google, appelé Colossus, qui garantit la durabilité en exploitant l'encodage d'effacement pour stocker des fragments redondants des données sur plusieurs disques physiques. Les données sont dupliquées dans plusieurs centres de données.

On peut également exécuter des requêtes BigQuery sur des données extérieures à l'espace de stockage BigQuery, telles que des données stockées dans Cloud Storage, Google Drive ou Bigtable, à l'aide de sources de données fédérées. Cependant, ces sources n'étant pas optimisées pour les opérations BigQuery, elles risquent de ne

pas fonctionner aussi bien que les données stockées dans l'emplacement de stockage BigQuery.

<https://cloud.google.com/architecture/bigquery-data-warehouse>

5. Organisation logique des données

Dans BigQuery, chaque donnée se retrouve stockée au sein de tables qui se retrouvent elles-mêmes au sein d'un dataset qui appartient à un projet. Il s'agit du GCP (Google Cloud Platform) qui est responsable de définir les droits pour les utilisateurs qui pourront exploiter BigQuery mais aussi les règles générales d'exploitation (mise en place de quota par exemple).

Le dataset regroupe un ensemble de données cohérent qui sera réparti dans différentes tables ou vues. C'est au niveau du dataset que nous devons fournir les droits d'accès aux données aux utilisateurs.

BigQuery est composé de deux services, un service de stockage et un service de requêtage.

<https://medium.com/codeshake/bigquery-les-bases-e6a689739b0c>

Elias Partie 1: 6,7,8

6. Gestion de la concurrence d'accès

Notion de transaction

<https://cloud.google.com/bigquery/docs/reference/standard-sql/transactions>

BigQuery accepte les transactions contenant plusieurs instructions dans les scripts. Une transaction à instructions multiples vous permet d'effectuer des opérations de mutation, telles que l'insertion ou la suppression de lignes, sur une ou plusieurs tables, et de valider ou d'annuler les modifications de manière atomique.

Les transactions multi-instructions sont utilisées notamment pour les opérations suivantes :

- Effectuer des mutations LMD sur plusieurs tables en une seule transaction. Les tables peuvent couvrir plusieurs ensembles de données ou projets.
- Effectuer des mutations sur une seule table en plusieurs étapes, en fonction de calculs intermédiaires.

Les transactions garantissent des propriétés ACID et sont compatibles avec l'isolation d'instantané. Au cours d'une transaction, toutes les lectures renvoient un instantané cohérent des tables référencées dans la transaction. Si une instruction dans une transaction modifie une table, les modifications sont visibles pour les instructions suivantes au sein de la même transaction.

Support des propriétés ACID

Comme cité plus haut, les transactions garantissent des propriétés ACID et sont compatibles avec l'isolation d'instantané.

“All table modifications in BigQuery, including DML operations, queries with destination tables, **and load jobs are ACID-compliant**. Therefore, modifying a table doesn't require any downtime.”

Technique de gestion de la concurrence d'accès

- **Consulter les informations de chronologie**

Les informations de chronologie peuvent permettre de savoir si une requête donnée est lente par elle-même ou à cause des effets d'autres requêtes en concurrence pour les mêmes ressources.

- **Utiliser des réservations d'emplacements**

Les réservations vous permettent d'attribuer un nombre d'emplacements dédiés d'un engagement à une charge de travail. Par exemple, vous ne souhaitez peut-être pas qu'une charge de travail de production soit en concurrence avec les charges de travail de test pour les emplacements. Vous pouvez créer une réservation nommée prod et lui attribuer vos charges de travail de production.

La sérialisation

Il est possible de télécharger les données de table à l'aide du format de données Arrow ou Avro et de désérialiser les données en objets de ligne.

7. Gestion des transactions distribuées

Architecture

L'architecture technique

BigQuery est une implémentation externe de Dremel. Pour comprendre comment cela fonctionne, il faut connaître les deux technologies de base de Dremel.

7.1 L'architecture en arbre

Ce type d'architecture est utilisé pour dispatcher les requêtes et agréger les résultats à travers des milliers de machines en quelques secondes,

7.2 La base de données orientée colonne

Dremel utilise BigTable et ColumnIO, des bases de données orientées colonne. Il stocke ensuite les données sous forme de colonnes. Grâce à cela, on ne charge que les colonnes dont on a vraiment besoin. Dremel stocke les enregistrements par colonne sur des volumes de stockage différents, alors que les bases de données traditionnelles, stockent normalement l'ensemble des données sur un seul volume. Cela permet un taux de compression très élevé.

Positionnement par rapport aux 12 règles de Date

Méthode pour garantir l'atomicité, la cohérence et la durabilité

Les données sont stockées sur le système de fichiers distribué de Google, appelé Colossus, qui garantit la durabilité en exploitant l'encodage d'effacement pour stocker des fragments redondants des données sur plusieurs disques physiques. De plus, les données sont dupliquées dans plusieurs centres de données.

<https://cloud.google.com/bigquery/docs/availability>

Traitement des pannes

8. Gestion de la reprise sur panne

Les techniques d'annulation

La journalisation

Surveiller et auditer

Vous pouvez surveiller BigQuery à l'aide du Monitoring, qui définit divers graphiques et alertes en fonction des métriques BigQuery. Par exemple, vous pouvez surveiller le débit du système à l'aide de la métrique Durée de la requête ou afficher les tendances de la demande de requête en fonction de la métrique Emplacements alloués. Vous pouvez utiliser la métrique Emplacements disponibles lorsque vous devez planifier à l'avance des requêtes exigeantes. Pour surveiller de manière proactive l'intégrité du système, vous pouvez créer des alertes basées sur des seuils que vous définissez. Monitoring fournit un portail Web en libre-service.

BigQuery crée automatiquement des journaux d'audit sur les actions de l'utilisateur. Vous pouvez exporter les journaux d'audit vers un autre ensemble de données BigQuery, par lot ou sous forme de flux de données, puis visualiser les journaux à l'aide de votre outil d'analyse préféré. Pour plus d'informations, consultez la page Analyser des journaux d'audit à l'aide de BigQuery.

BigQuery fournit également des vues en lecture seule que vous pouvez utiliser pour accéder aux métadonnées de vos ressources BigQuery, telles que des ensembles de données, des tables et des tâches. Ces vues peuvent être utilisées à diverses fins, telles que le suivi de la date d'expiration de votre table ou la compréhension de l'utilisation des emplacements de vos requêtes.

Les sauvegardes

Sauvegarde et récupération

La gestion de la sauvegarde et de la disponibilité a toujours été une tâche complexe et coûteuse pour les administrateurs de base de données. Le besoin de licences et de matériel supplémentaires peut augmenter considérablement les coûts. BigQuery se charge de la sauvegarde et de la reprise après sinistre au niveau du service. En conservant un historique complet des modifications des tables sur sept jours, BigQuery vous permet d'interroger un instantané de vos données à un moment précis à l'aide de l'une des méthodes suivantes : décorateurs de table ou de `SYSTEM_TIME AS OF` dans la clause `FROM`. Vous pouvez facilement annuler les modifications sans avoir à demander une récupération à partir des sauvegardes. (Lorsqu'une table est explicitement supprimée, son historique est vidé après sept jours.) De plus, la commande `cp` offre des instantanés de table dans la région.

Les ensembles de données BigQuery peuvent être régionaux ou multirégionaux. Pour les ensembles de données régionaux, par exemple un ensemble de données situé dans la région `us-central1`, aucune copie de l'ensemble de données n'est conservée en dehors de la région. Si vous considérez que l'absence de sauvegardes en dehors d'une région est risquée pour votre entreprise, vous pouvez créer et planifier des copies interrégionales à l'aide du service de transfert de données BigQuery. Pour les ensembles de données multirégionaux situés dans de grandes zones géographiques telles que l'Europe (UE), une copie est automatiquement stockée dans une autre région Google Cloud.

Si une région échoue, certaines données récentes risquent d'être perdues. Pour en savoir plus, consultez la documentation BigQuery sur la disponibilité et la durabilité.

Gestion de la reprise à chaud
Gestion de la reprise à froid

9. Techniques d'indexation

BigQuery étant un service entièrement géré, l'équipe d'ingénieurs de BigQuery s'occupe des mises à jour et de la maintenance. Les mises à niveau ne nécessitent généralement pas de temps d'arrêt et ne nuisent pas aux performances du système.

De nombreux systèmes conventionnels requièrent des processus de vide consommant beaucoup de ressources afin de s'exécuter à différents intervalles pour remanier et trier les blocs de données, et récupérer de l'espace. BigQuery n'intègre aucun équivalent à la gestion de vide ou d'index, car le moteur de stockage gère et

optimise en permanence la manière dont les données sont stockées et dupliquées. De plus, étant donné que BigQuery n'utilise pas d'index sur les tables, vous n'avez pas besoin de reconstruire les index.

10. Optimisation de requêtes

Réduire l'analyse des données

BigQuery n'est pas compatible avec les index, et ne les utilise donc pas. Chaque fois qu'il exécute une requête, il analyse la colonne complète. Les coûts liés aux requêtes et aux performances de BigQuery étant basés sur la quantité de données analysées au cours d'une requête, nous vous recommandons de concevoir vos requêtes de sorte qu'elles ne référencent que les colonnes pertinentes. Lorsque vous utilisez des tables partitionnées, seules les partitions pertinentes doivent être analysées. Vous pouvez éviter les analyses indésirables en utilisant des filtres de partition basés sur la colonne de partition. Si nous avons des requêtes qui filtrent fréquemment sur des colonnes particulières, il est envisagé de mettre en cluster la table. Si vous devez fréquemment exécuter une requête d'agrégation pour un traitement ultérieur, envisagez de matérialiser la requête. Cette approche réduit les exigences de calcul ainsi que la quantité de données à analyser.

Réduire les besoins en calcul

Nous vous recommandons d'éviter d'utiliser des fonctions JavaScript définies par l'utilisateur. Lorsque cela est possible, utilisez plutôt des fonctions SQL définies par l'utilisateur. Une autre façon d'accélérer les requêtes consiste à utiliser des agrégations approximatives, telles que `APPROX_COUNT_DISTINCT` au lieu de `COUNT(DISTINCT)`.

Améliorer les performances des jointures

Les entreprises doivent souvent joindre plusieurs tables, en particulier lorsque les entrepôts de données possèdent un schéma en étoile ou en flocon. Une table de faits est généralement plus grande qu'une table de dimension. Dans le schéma en flocon, comme les dimensions sont normalisées, vous pouvez avoir des tables de dimensions encore plus petites. Il est recommandé de commencer par la table de faits sur la gauche et de la joindre aux tables de dimensions plus petites sur la droite, par taille décroissante. Lorsqu'une grande table constitue le côté gauche de la jointure JOIN et une petite le côté droit de la jointure JOIN, une jointure de diffusion

est créée. Celle-ci envoie toutes les données de la plus petite table à chaque emplacement qui traite la plus grande table.

Salim 1-1 1-2 1-5

Thomas 1-9 1-10