Engineer

BigQuery

Googles datalager för frågor mot mycket stora tabeller

Vad det är

BigQuery är en hanterad tjänst för analys med SQL i Google Cloud: det finns ingen server att sätta upp, tabellerna lagras kolumn för kolumn, och varje databasfråga betalas antingen efter de byte som den läser eller ur beräkningskapacitet som har reserverats i förväg. Tjänsten säljs och söks också efter som Google BigQuery, och dess SQL-dialekt heter GoogleSQL, som följer ANSI-standarden och lägger till egna funktioner.

Designen vilar på att hålla lagring och beräkning isär. Tabellerna ligger i Googles lagringslager, och en fråga lånar beräkningskraft så länge den körs. Eftersom varje kolumn lagras för sig läser en fråga som nämner tre kolumner i en tabell med femtio bara de tre och lämnar resten på disken. Att trimma prestandan ser därför annorlunda ut än på en PostgreSQL-server. I stället för att utforma B-tree-index och dimensionera minnet handlar arbetet om tabellens form, med partitionering och klustring: en tabell kan delas upp efter en kolumn med datum eller tidsstämpel, efter tidpunkten för inläsningen eller efter ett heltalsintervall, och klustras på upp till fyra kolumner så att rader som hör ihop delar lagringsblock.

BigQuery hör till familjen analysdatalager, och de produkter den oftast vägs mot är Snowflake och Amazon Redshift, plus möjligheten att behålla analysen i den transaktionsdatabas som företaget redan kör. Den är byggd för frågor som sveper över miljontals eller miljarder rader och returnerar en handfull summor. De uppslag och uppdateringar av enskilda rader som en applikation gör vid varje förfrågan hör hemma i en databas som PostgreSQL eller SQL Server, och ett vanligt upplägg kör båda, med en datapipeline som för data från den ena till den andra.

Använd begreppet i ett beslut

Använd ”Vad är BigQuery?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Siffror från försäljning, drift och ekonomi samlade på ett ställe, så att två avdelningar slutar komma med två olika svar till samma möte.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Gjorde pipelinen för geodata 50 gånger snabbare — 50x snabbare datapipelines för geodata

Gjorde pipelinen för geodata 50 gånger snabbare Hör av er →

Varför det spelar roll

Beslutet som BigQuery ställer ett team inför är hur det vill betala för analys. Räkningen följer de frågor som folk skriver, inte storleken på en maskin som någon en gång valde, och det skär åt båda hållen.

Med prissättning på begäran (on-demand) debiteras varje fråga för de byte som den bearbetar. Det passar oregelbundet arbete: ett team som kör några tunga rapporter i veckan betalar för de rapporterna och för inget däremellan, utan någon server som står på tomgång över natten. Det sätter också ett pris på en slarvig fråga. En `SELECT *` över ett års sensoravläsningar kostar lika mycket oavsett om analytikern ville ha en rad eller alla. Kapacitetsprissättning ersätter det med slots, enheter av beräkningskraft som köps genom reservationer, vilket gör räkningen till en siffra som är känd i förväg och gör tung, jämn användning billigare att planera för. Ett projekt kan byta mellan de två modellerna, så det första valet går att ompröva när den verkliga användningen syns.

Lagringen har sina egna regler. En tabell eller partition som har lämnats oförändrad i 90 dagar i följd går över till priset för långtidslagring, som är 50 % lägre, utan att någon flyttar den. Varje dataset debiteras antingen efter den logiska, okomprimerade storleken på sina data eller efter den fysiska, komprimerade, och för data som komprimeras väl kan den andra modellen vara den billigare.

Tre saker förtjänar eftertanke innan man skriver på. Platsen för ett dataset, en enskild region eller multiregionen US eller EU, bestäms när det skapas och ligger sedan fast, så den är ett beslut om regelefterlevnad som fattas den första dagen. Dialekten, kommandoradsverktyget `bq` och klientbiblioteken är Googles egna, så frågor flyttar till ett annat analysdatalager med ändringar snarare än oförändrade; själva datan tar sig lätt ut, eftersom tabeller kan exporteras till CSV,

Avro och Parquet. Och frågan om rekrytering är mildare än den ser ut: analytiker som kan SQL är produktiva inom några dagar, medan den ovanligare färdigheten är att läsa en fråga i byte lika väl som i rader.

Så när är BigQuery rätt val? När data kommer i stora volymer som mest läggs till, t.ex. händelser, loggar eller mätaravläsningar, när frågorna som ställs till dem ändras från vecka till vecka, och när teamet helst slipper driva databasservrar för analys. För data som en enda PostgreSQL-server rymmer utan besvär lägger ett analysdatalager till ett andra system, en andra räkning och en datapipeline mellan de två. Skälen för det växer med volymen och med antalet personer som ställer frågor.

Så ser det ut i praktiken

Ett bestånd med över 100 databaser, fördelade på PostgreSQL, Microsoft SQL Server och Google BigQuery, höll mest geografiska data och tidsseriedata: koordinater, sensoravläsningar, loggar och mätvärden i realtid. Databasingenjörens arbete mellan 2022 och 2024 omfattade att lagra de stora tidsseriemängderna i BigQuery, i tabeller som var tidspartitionerade och klustrade, med komprimering och kolumnlagring för att hålla nere lagringskostnaderna och snabba upp skanningarna. Partitionerade och klustrade skanningar i BigQuery, tillsammans med indexering, materialiserade vyer och cachning i hela beståndet, kortade svarstiderna för frågor mot GIS- och tidsseriedata med 40–60 %.

I en datapipeline för geografiska data som hade blivit långsam analyserades först SQL-koden och datamodellerna i PostgreSQL, MS SQL och Google Cloud BigQuery. Analysen blottlade svag indexering, ineffektiva databasfrågor och saknade integritetsvillkor. Varje motor fick sedan det verktyg den har: B-tree- och GiST-index i PostgreSQL, filtrerade index i MS SQL och klustring i BigQuery, tillsammans med normaliserade modeller, partitionering och omskrivna databasfrågor. Grundarens resultatbeskrivning anger att datapipelinen som helhet blev 50 gånger snabbare; den anger inte prestandatestets dataset, hårdvara eller metod för jämförelsen.

Och på data från smarta elmätare i flera regioner togs över 500 analysrapporter fram i ren SQL mot PostgreSQL och BigQuery, i rollen som dataanalytiker, med CTE:er, fönsterfunktioner, underfrågor och geospatiala joins över mätar-ID:n, tidsstämplar, koordinater och miljömetadata. Varje rapport levererades med kommenterad SQL, så att en kollega kunde köra om och granska den, och med

anteckningar om kända datakvalitetsproblem som saknade GPS-koordinater och korrupta mätvärden. Rapporterna blev ett standardunderlag på flera avdelningar och bidrog till 10–15 % effektivare operativ planering.

I alla tre fallen arbetade BigQuery bredvid PostgreSQL snarare än i dess ställe, och varje motor fick det arbete som den gör bäst.

Var det går fel

En fråga med `LIMIT 10` debiteras för hela tabellen. På en tabell utan klustring begränsar `LIMIT` resultatet, inte de byte som läses, och `SELECT *` läser varje kolumn som finns. En snabb titt på datan blir en fullständig skanning. Namnge de kolumner som frågan behöver, och använd konsolens förhandsvisning av tabellen eller `bq head` för en blick på några exempelrader, vilket inte kostar något.

En partitionerad tabell skannas ändå från början till slut. Partitioner hoppas bara över när filtret på partitioneringskolumnen är en konstant som motorn kan utvärdera innan den läser. Att jämföra kolumnen med resultatet av en underfråga, svepa in den i en funktion eller räkna på den skickar frågan genom varje partition. Två skydd hjälper. En tabell kan kräva ett partitionsfilter, så att en fråga utan ett användbart filter stoppas med ett fel i stället för att köras. Och en torrkörning (dry run) rapporterar hur många byte en fråga skulle bearbeta, utan att köra den eller ta betalt för den:

```
bq query --use_legacy_sql=false --dry_run \
  'SELECT meter_id, kwh
   FROM energy.readings
   WHERE reading_date = DATE "2026-10-01"'
```

Klustring läggs till och mängden lästa byte rör sig knappt. Klustring sorterar lagringsblock efter upp till fyra kolumner i ordning, och ordningen spelar roll: en tabell som är klustrad efter orderdatum, land och status är inte organiserad för ett filter på enbart land och status. Tabeller under 64 MB vinner lite på det, och på en klustrad tabell blir antalet byte som faktiskt skannas känt först när frågan är klar. Ordna klustringskolumnerna på det sätt som frågorna filtrerar. Där kostnaden måste ha ett tak innan något körs, lita på partitionering, vars beskärning en torrkörning redan räknar med, och sätt ett högsta antal debiterade byte (maximum bytes billed), så att en fråga över gränsen stoppas innan den debiteras.

En fråga som gör en join mellan två dataset avvisas. Varje dataset ligger på en plats, och en fråga körs på en plats, så varje dataset som den rör måste dela den. Team som låter varje projekt ta sin egen standardplats får tabeller som bara kan förenas efter att en av dem har kopierats över. Välj platsen en gång, för regelefterlevnadens skull och efter var datan kommer ifrån, och skapa varje dataset där med avsikt.

Läs mer

- BigQuery overview — Googles egen beskrivning av tjänsten, dess separata lager för lagring och beräkning och det som sköts åt användaren.
- Introduction to partitioned tables — typerna av partitionering, deras upplösning och deras gränser.
- Query partitioned tables — vilka filter som beskär partitioner och vilka som skickar en fråga genom alla.
- Introduction to clustered tables — hur kolumnordningen styr beskärningen av block, och vad det betyder för kostnadsuppskattningar.
- Estimate and control costs — torrkörningar, högsta antal debiterade byte och de kostnadsfria sätten att förhandsvisa en tabell.
- BigQuery locations — regioner, multiregioner och reglerna för frågor som rör mer än ett dataset.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med ett representativt dataset, frågorna det ska besvara och dagens flaskhals. Vi kan avtala mått på kvalitet, genomströmning och återställning.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är BigQuery?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/bigquery/