Elaborazione e ottimizzazione dei Big Data

Obiettivi di apprendimento

Al termine di questa unità, gli studenti saranno in grado di:

- Spiegare il concetto di Big Data e le sue caratteristiche (volume, velocità, varietà).
- Identificare le principali fonti di Big Data ambientali (satelliti, sensori, archivi aperti).
- Comprendere le principali sfide legate all'archiviazione, alla pulizia e all'elaborazione di dataset di grandi dimensioni.
- Utilizzare strumenti appropriati per elaborare i Big Data (Google BigQuery, Spark, Hadoop, ecc.).
- Ottimizzare le interrogazioni e i flussi di elaborazione per migliorare prestazioni ed efficienza dei costi.
- Visualizzare dataset di grandi dimensioni mediante dashboard e interpretare i risultati in ambito ambientale.
- Progettare ed eseguire un flusso di lavoro scalabile per elaborare, analizzare e visualizzare dataset ambientali.

Introduzione

Le scienze ambientali sono entrate in un'epoca in cui il volume, la complessità e la velocità di produzione dei dati superano ampiamente le capacità per le quali sono stati concepiti i metodi tradizionali di gestione e analisi. Il termine Big Data indica insiemi di dati talmente grandi, dinamici o eterogenei da richiedere strumenti computazionali avanzati, sistemi distribuiti e nuovi approcci analitici per ricavarne informazioni significative. Nelle scienze ambientali, i Big Data non riguardano soltanto le dimensioni: comprendono flussi continui, molteplici formati e dimensioni spaziali e temporali che devono essere analizzate congiuntamente.

I Big Data sono comunemente descritti attraverso alcune caratteristiche fondamentali, spesso denominate le «V»:

- Volume – enormi quantità di dati, da terabyte a petabyte, prodotte quotidianamente da satelliti, reti di sensori e simulazioni.

- Velocità – elevata rapidità di generazione dei dati, come nel caso delle stazioni meteorologiche in tempo reale, dei sistemi di monitoraggio della qualità dell'aria e delle boe oceanografiche.
- Varietà – molteplici tipologie di dati, tra cui misurazioni numeriche, immagini, video, serie temporali, rapporti testuali e livelli geospaziali.
- Veridicità – qualità, incertezza e affidabilità variabili dei dati, aspetti particolarmente importanti nelle misurazioni ambientali.
- Valore – possibilità di ricavare conoscenze utilizzabili per la modellazione climatica, la protezione degli ecosistemi, la prevenzione dei disastri e le decisioni pubbliche.

Nelle scienze ambientali, i Big Data svolgono un ruolo cruciale nella comprensione dei complessi sistemi terrestri, nell'individuazione di tendenze di lungo periodo e nel supporto a decisioni fondate su evidenze per la sostenibilità e la tutela ambientale.

Fonti dei dati ambientali

I Big Data ambientali provengono da una vasta gamma di fonti, ciascuna delle quali offre una prospettiva specifica sui processi naturali e sulle attività umane.

I dati satellitari costituiscono una delle fonti principali. I satelliti per l'osservazione della Terra raccolgono continuamente informazioni sull'uso del suolo, sulla temperatura superficiale dei mari, sulla composizione dell'atmosfera, sullo stato della vegetazione, sulla copertura glaciale e sulla dinamica delle nubi. Si tratta generalmente di dataset ad alta risoluzione, multispettrali e a copertura globale, estremamente preziosi ma anche molto impegnativi dal punto di vista computazionale.

I sensori in situ (misurazioni effettuate direttamente nel luogo di interesse e a contatto con l'oggetto di studio) forniscono misure di riferimento e dati ad alta frequenza. Comprendono stazioni meteorologiche per temperatura, umidità, pressione e precipitazioni; boe oceanografiche per salinità, correnti e clorofilla; sensori della qualità dell'aria per inquinanti quali PM2.5, NO_2 e ozono; sensori fluviali e del suolo per livelli idrici, umidità e nutrienti. Questi sensori spesso funzionano ininterrottamente e generano flussi da elaborare quasi in tempo reale.

I database aperti globali arricchiscono ulteriormente la ricerca ambientale. Enti meteorologici, agenzie spaziali e consorzi internazionali pubblicano dataset ad accesso aperto relativi a clima, biodiversità, oceanografia e scienze atmosferiche. Tali dati consentono di combinare misurazioni

locali e osservazioni su scala globale, favorendo studi comparativi e di lungo periodo. Nel loro insieme, queste fonti costituiscono l'ecosistema di dati complesso e interconnesso che caratterizza i Big Data ambientali.

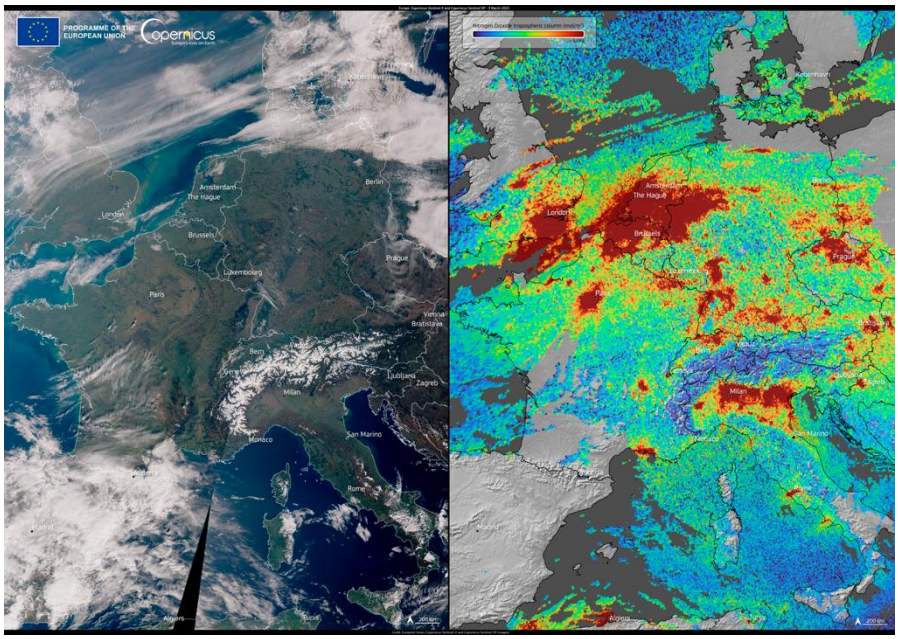


Figura 6.1 Satellite Copernicus Sentinel-5P il 3 marzo 2025, con evidenza delle concentrazioni di biossido di azoto (NO_2).

Il ciclo di vita dei Big Data

La gestione dei Big Data ambientali segue un processo strutturato, detto ciclo di vita dei Big Data (Figura 6.2), articolato in più fasi interconnesse.

Raccolta dei dati: i dati vengono acquisiti da satelliti, sensori, rilevazioni sul campo, simulazioni e database. Occorre considerare frequenza di campionamento, risoluzione spaziale e formati per garantire compatibilità e rilevanza scientifica.

Archiviazione dei dati: date le dimensioni e la complessità dei dataset ambientali, l'archiviazione si basa spesso su sistemi distribuiti, piattaforme cloud e database specializzati per serie temporali e dati geospaziali. Un'archiviazione efficiente è fondamentale per la scalabilità e l'accessibilità nel lungo periodo.

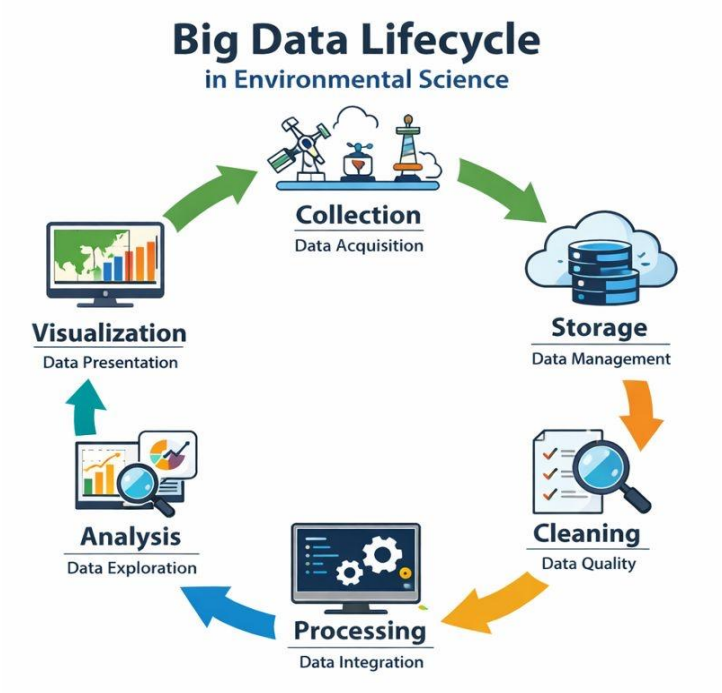
Pulizia e preelaborazione dei dati: i dati ambientali presentano spesso valori mancanti, rumore, valori anomali o misurazioni incoerenti. La pulizia comprende rilevamento degli errori, completamento delle lacune, standardizzazione delle unità e allineamento temporale o spaziale. Questa fase è essenziale, perché una scarsa qualità può condurre a conclusioni fuorvianti.

Elaborazione dei dati: questa fase trasforma i dati grezzi in forme strutturate e analizzabili, mediante aggregazione, filtraggio, ricampionamento e integrazione di diversi dataset. Nei contesti Big Data l'elaborazione è spesso parallelizzata per gestire grandi volumi in modo efficiente.

Analisi dei dati: i metodi spaziano dall'analisi statistica e dall'individuazione delle tendenze al machine learning e alla modellazione predittiva. Nelle scienze ambientali l'analisi serve a individuare schemi ricorrenti, valutare rischi e simulare scenari futuri.

Visualizzazione dei dati: questa fase traduce dataset complessi in mappe, grafici, dashboard e animazioni intuitive. Una visualizzazione efficace è essenziale per comunicare i risultati a ricercatori, decisori e pubblico, soprattutto in presenza di fenomeni spaziali e temporali.

Figura 6.2 Ciclo di vita dei Big Data



Principali sfide dei Big Data ambientali

Nonostante il loro potenziale, i Big Data ambientali pongono sfide importanti. L'eterogeneità costituisce un problema centrale: i dataset differiscono per formato, risoluzione, scala e incertezza. Combinare immagini satellitari, letture dei sensori e serie storiche richiede un'attenta armonizzazione.

La velocità rappresenta una sfida quando i dati arrivano continuamente e devono essere elaborati in tempo reale, per esempio durante eventi meteorologici estremi o allerte per inquinamento. Anche il volume può superare le capacità dei sistemi informatici tradizionali. Serie climatiche

pluridecennali e immagini satellitari ad alta risoluzione richiedono infrastrutture scalabili e algoritmi efficienti.

La qualità dei dati e l'incertezza sono particolarmente critiche nelle applicazioni ambientali. Guasti dei sensori, errori di calibrazione e dati mancanti sono comuni; l'incertezza deve essere gestita esplicitamente per garantire conclusioni scientifiche affidabili.

Formati dei Big Data e modalità di archiviazione

Uno dei formati più semplici e diffusi è CSV (Comma-Separated Values). I file CSV memorizzano dati in testo semplice: ogni riga rappresenta un record e le colonne sono separate da virgole. Grazie alla semplicità, sono facili da generare, ispezionare e scambiare tra piattaforme. Sono utilizzati per dati tabellari, quali osservazioni meteorologiche, misurazioni dell'inquinamento e riepiloghi statistici. Tuttavia, non dispongono di supporto integrato ai metadati, rappresentano male le strutture gerarchiche e diventano inefficienti con dataset molto grandi.

JSON (JavaScript Object Notation) è un formato semistrutturato che rappresenta dati gerarchici e annidati. È particolarmente adatto allo scambio di dati tramite API web, dispositivi IoT e servizi cloud. Nei sistemi di monitoraggio ambientale, le reti di sensori trasmettono spesso dati JSON in oggetti contenenti misurazioni, marcature temporali e identificativi dei dispositivi. Pur essendo leggibile e flessibile, JSON è meno efficiente dei formati binari per quantità enormi di dati, perché più verboso.

Per superare i limiti di prestazioni e scalabilità, i moderni sistemi Big Data adottano spesso formati binari ad archiviazione colonnare, come Parquet. Apache Parquet è ottimizzato per i carichi analitici e viene usato in sistemi distribuiti come Apache Spark e nei data warehouse cloud. Memorizzando i dati per colonna anziché per riga, consente interrogazioni più rapide, migliore compressione e costi di archiviazione inferiori. È particolarmente efficace per grandi archivi storici, come dati climatici e misurazioni ambientali di lungo periodo.

Nelle discipline scientifiche, soprattutto climatologia e oceanografia, NetCDF (Network Common Data Form) è un formato molto diffuso. È progettato specificamente per memorizzare dati multidimensionali, quali temperatura, salinità o pressione atmosferica nello spazio e nel tempo. A differenza dei formati generici, NetCDF include metadati ricchi che descrivono variabili, unità e sistemi di coordinate, favorendo la riproducibilità della ricerca.

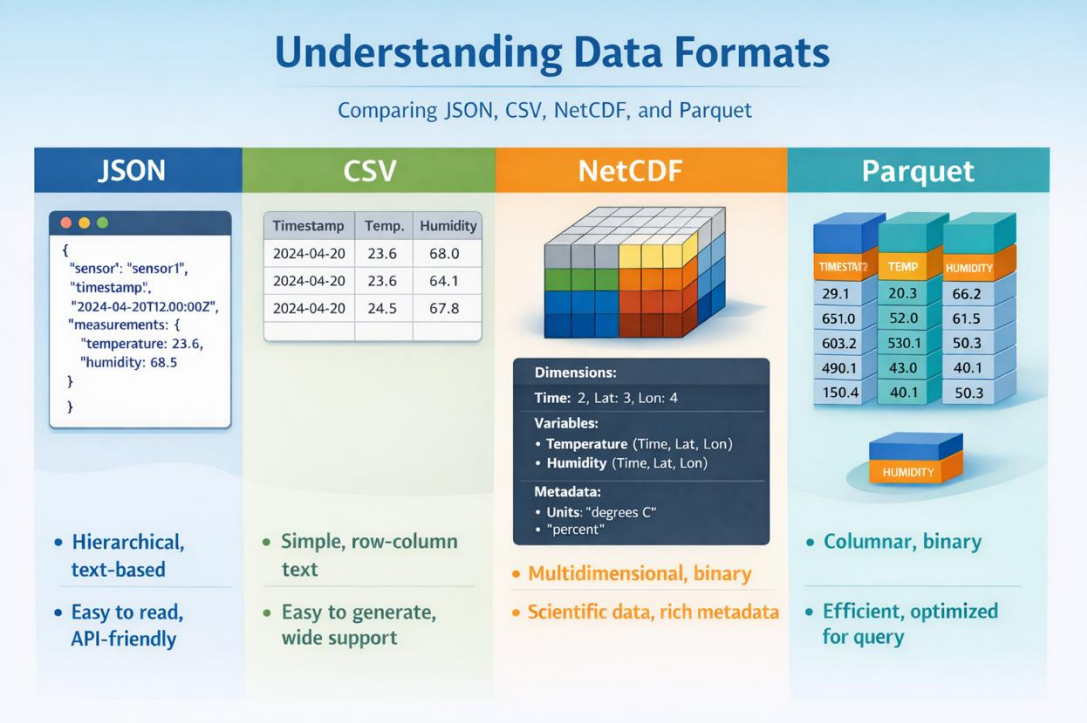


Figura 6.3 Comprendere i formati dei dati: JSON, CSV, NetCDF e Parquet

Oltre ai formati (Figura 6.3), i sistemi Big Data dipendono da diverse modalità di archiviazione. Quella locale tradizionale, su computer personali o server di laboratorio, è semplice da gestire ma limitata per scalabilità e affidabilità. Con l'aumentare delle dimensioni e della complessità, non è più sufficiente per analisi collaborative e su vasta scala.

L'elaborazione dei Big Data richiede ecosistemi software coordinati, capaci di gestire archiviazione, calcolo e allocazione delle risorse su molte macchine. Tra i primi e più influenti vi è Apache Hadoop, che ha introdotto l'archiviazione distribuita (HDFS) insieme al calcolo distribuito tramite il modello MapReduce. Hadoop ha consentito di elaborare dataset prima troppo grandi per una singola macchina.

Sebbene Hadoop abbia posto le basi dei Big Data, framework più recenti come Apache Spark si sono ampiamente diffusi grazie alle migliori prestazioni e alla flessibilità. Spark supporta il calcolo in memoria, che accelera notevolmente l'elaborazione, e fornisce API di alto livello per analisi dei dati, machine learning ed elaborazione dei grafi. In ambito ambientale viene spesso usato per grandi dataset di sensori o archivi di immagini satellitari.

Negli ultimi anni le piattaforme cloud hanno trasformato il panorama dei Big Data offrendo servizi gestiti che evitano di amministrare manualmente l'infrastruttura. Google Cloud Platform (GCP, Figura 6.4), Amazon Web Services (AWS) e Microsoft Azure offrono ecosistemi integrati per archiviazione, elaborazione, analisi e visualizzazione. I ricercatori possono così concentrarsi sui dati anziché sull'amministrazione dei sistemi.

Gli ecosistemi cloud sono particolarmente interessanti per ambiti interdisciplinari come le scienze ambientali, dove i volumi di dati sono elevati e la domanda di calcolo varia nel tempo. Supportano la scalabilità elastica, ossia l'aumento o la diminuzione dinamici delle risorse in funzione del carico, ottimizzando prestazioni e costi.

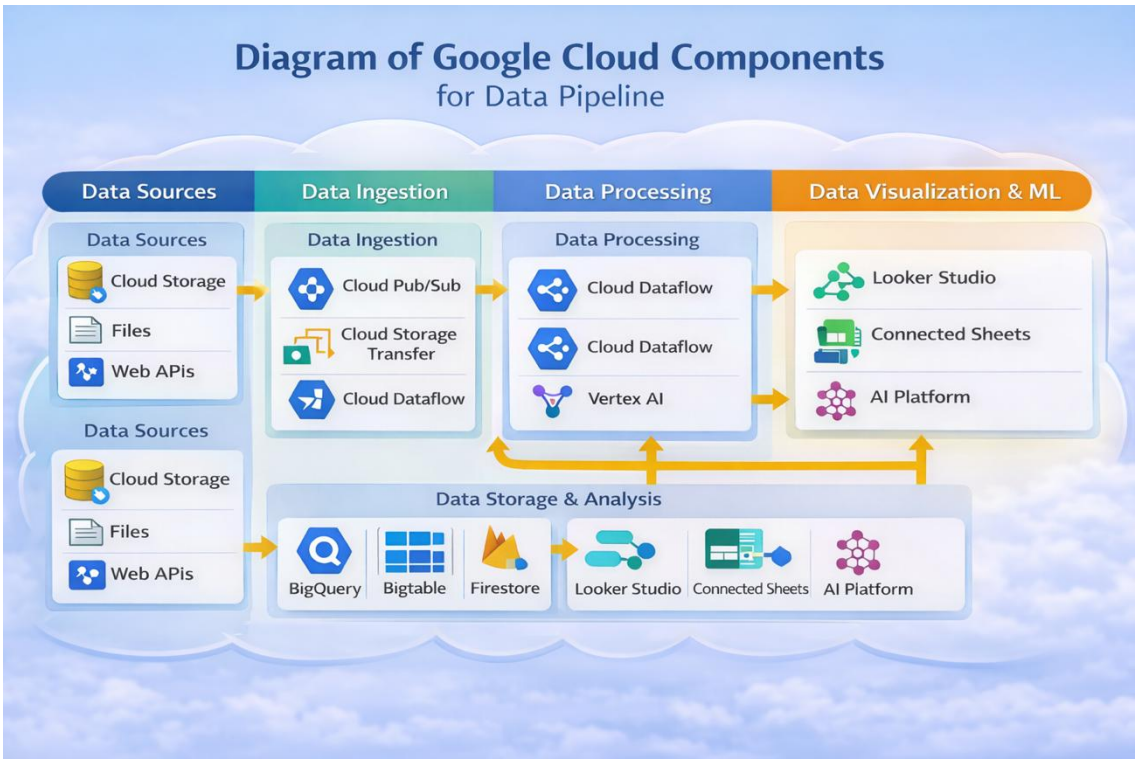


Figura 6.4 Componenti di Google Cloud

Preelaborazione e pulizia dei dati su larga scala

I dati grezzi raramente sono pronti per l'analisi. In pratica, i Big Data contengono spesso valori mancanti, incoerenze, duplicati ed errori di misurazione. Preelaborazione e pulizia sono dunque fasi fondamentali in ogni flusso Big Data, soprattutto per dati ambientali e flussi IoT.

OpenRefine è uno strumento open source molto diffuso per la pulizia e trasformazione interattiva dei dati. Permette di esplorare dataset, individuare anomalie, standardizzare valori e correggere

errori senza competenze di programmazione. È utile in particolare per dati ottenuti da archivi pubblici o indagini sul campo.

Trifacta Wrangler, successivamente noto come Google Cloud Dataprep, estende tali capacità a dataset di scala cloud. Fornisce un'interfaccia visuale basata su regole per definire trasformazioni eseguibili su larga scala, utile soprattutto per grandi dataset in Google Cloud Storage o BigQuery.

Per chi programma, Pandas, libreria Python, offre potenti funzioni di manipolazione dei dati. Viene spesso usata con notebook Jupyter e Google Colab per pulire, filtrare e trasformare i dataset prima dell'elaborazione su larga scala. Adatta soprattutto a dataset di dimensioni medie, è importante per l'analisi esplorativa e la prototipazione.

Concetti di elaborazione distribuita e parallela

Al centro dell'analisi dei Big Data vi sono l'elaborazione distribuita e quella parallela. Anziché elaborare i dati in sequenza su un'unica macchina, i sistemi li suddividono in parti più piccole processabili contemporaneamente su più nodi.

MapReduce (Figura 6.5) permette di elaborare dataset molto grandi suddividendo il lavoro in numerosi piccoli compiti eseguibili simultaneamente. È diventato diffuso perché consente di gestire dati troppo grandi per un singolo computer senza richiedere una macchina eccezionalmente potente. Si utilizzano molte macchine comuni (o istanze cloud), ripartendo il dataset tra loro. Questo è utile quando i dati sono distribuiti tra file o server, come log, letture dei sensori, immagini satellitari e archivi ambientali storici.

Per comprendere MapReduce è utile distinguere il ruolo delle due funzioni. Map trasforma l'input grezzo in informazioni intermedie facilmente combinabili; non conclude il calcolo, ma produce elementi strutturati (coppie chiave-valore). Reduce combina poi gli elementi intermedi con la stessa chiave per ottenere il risultato finale. Molti compiti Big Data seguono questo schema: conteggio, somma, raggruppamento, medie, massimi e minimi, oppure statistiche per località.

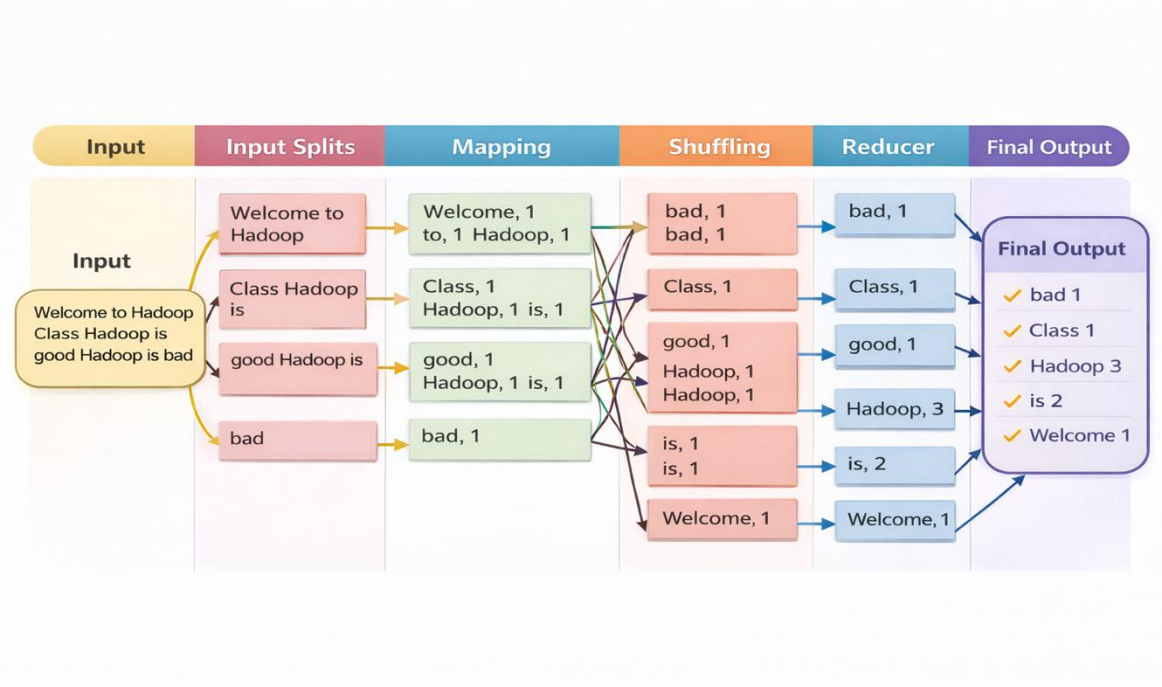


Figura 6.5 Funzionamento di MapReduce

La fase di «shuffle» (talvolta considerata una fase del sistema anziché definita dall'utente) è essenziale perché raccoglie insieme tutti gli elementi con la stessa chiave. Spesso è anche una delle fasi più costose, perché richiede il trasferimento di dati tra macchine attraverso la rete. Se Map produce chiavi come «station_id» o «city», il sistema deve inoltrare tutti i record della stessa stazione allo stesso reducer. Poiché la comunicazione di rete è più lenta delle operazioni in memoria, una buona progettazione cerca di ridurre i dati intermedi e bilanciare le chiavi evitando di sovraccaricare un singolo reducer.

Un grande vantaggio di MapReduce è la tolleranza ai guasti. Nei sistemi distribuiti di grandi dimensioni è normale che alcune macchine si guastino. Framework come Hadoop riassegnano automaticamente a un altro nodo il lavoro di un processo interrotto, consentendo di continuare l'elaborazione. Questo è importante nelle applicazioni ambientali, in cui i flussi possono durare ore: l'affidabilità conta quanto la velocità.

Nelle analisi ambientali e della qualità dell'aria, il ragionamento tipico di MapReduce ricorre spesso anche senza scrivere codice MapReduce esplicito. Si immagini una serie pluriennale di misure PM2.5 da molte stazioni. Map può emettere (station_id, pm25_value) per ogni misura; Reduce può calcolare medie per stazione, massimi giornalieri o il numero di giorni che superano una soglia.

Oggi molti sistemi moderni (come Spark, BigQuery e le pipeline Dataflow) non richiedono di implementare manualmente MapReduce; il modello concettuale resta però utile. Spiega perché gli strumenti Big Data danno importanza a operazioni come raggruppamento, aggregazione e join distribuite: trasformare i dati in categorie significative, riunire gli elementi correlati e sintetizzarli in risultati interpretabili.

Indicizzazione, ottimizzazione delle query e gestione energetica

Con l'aumentare delle dimensioni dei dataset, diventa sempre più importante accedere ai dati in modo efficiente. L'indicizzazione migliora le prestazioni delle interrogazioni permettendo di trovare rapidamente i dati rilevanti senza scandire l'intero dataset. Nei data warehouse cloud come Google BigQuery, strategie di indicizzazione e partizionamento riducono in modo significativo tempi e costi delle query.

L'ottimizzazione delle query consiste nella scelta di piani di esecuzione efficienti. I sistemi Big Data moderni ottimizzano automaticamente le interrogazioni limitando gli spostamenti dei dati, sfruttando il parallelismo e utilizzando meccanismi di cache. Questo è particolarmente importante nel cloud, dove le risorse computazionali vengono fatturate in base all'utilizzo.

Oltre alle prestazioni, la gestione della potenza e dell'energia è diventata fondamentale. I grandi data center consumano quantità significative di energia, favorendo lo sviluppo di hardware efficiente e algoritmi di schedulazione. I fornitori cloud investono sempre di più nelle energie rinnovabili e nell'allocazione energeticamente consapevole delle risorse, rendendo le soluzioni Big Data cloud potenzialmente più sostenibili delle infrastrutture locali tradizionali.

Caso di studio: elaborazione e analisi della qualità dell'aria nel cloud con strumenti Google

In questo caso di studio viene scelto come fonte principale il dataset OpenAQ, disponibile tramite Google Cloud Marketplace/BigQuery. Contiene osservazioni sulla qualità dell'aria su scala mondiale (ad esempio PM2.5, PM10, NO_2, O_3), raccolte da vari fornitori e da migliaia di punti di monitoraggio. I record includono generalmente metadati spazio-temporali (Paese, città, stazione e spesso coordinate), inquinante misurato, valore, unità e marcatura temporale. Queste caratteristiche lo rendono adatto ad analisi ambientali su vasta scala, come le tendenze delle serie temporali, l'individuazione di aree critiche e la produzione di dashboard con strumenti cloud quali BigQuery e Looker Studio. Più in generale, OpenAQ mira ad acquisire, aggregare, armonizzare e diffondere dati interoperabili, storici e quasi in tempo reale, migliorandone utilizzo e confrontabilità.

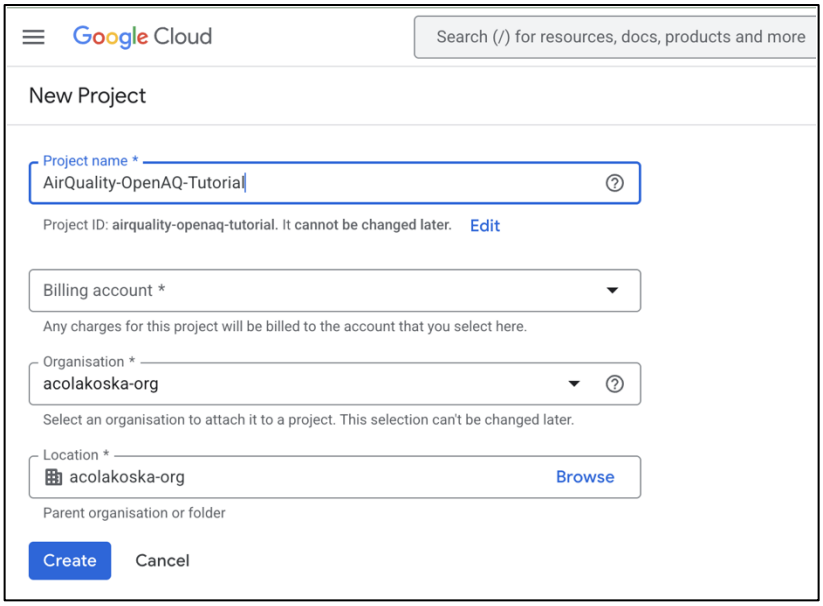


Figura 6.6 Creazione di un progetto Google Cloud

Per iniziare creiamo un progetto Google Cloud dedicato (Figura 6.6), in modo da tenere organizzate e separate le risorse dell'esercitazione sulla qualità dell'aria. Nella Google Cloud Console utilizziamo il selettore dei progetti nella barra superiore e scegliamo New Project. Assegniamo un nome chiaro, per esempio AirQuality-OpenAQ-Tutorial, così da riconoscerlo facilmente quando cambiamo progetto, gestiamo i permessi o controlliamo la fatturazione. Dopo

aver inserito il nome (e, se richiesto, organizzazione o posizione), facciamo clic su Create per inizializzare l'ambiente in cui gestiremo dataset, query e dashboard BigQuery.

Accediamo poi a BigQuery tramite Google Cloud Console. Nella barra di ricerca scriviamo «BigQuery» e apriamo il servizio. Una volta caricato, verifichiamo che sia selezionato il progetto corretto: così tutti i dataset creati, le query eseguite e le viste salvate saranno associati al progetto previsto.

Assicuriamoci che l'API di BigQuery sia abilitata (Figura 6.7). In genere viene attivata automaticamente al primo accesso. In presenza di errori di accesso o relativi all'API, occorre abilitarla manualmente: apriamo APIs & Services → Enabled APIs & services, controlliamo che BigQuery API sia presente e, in caso contrario, selezioniamola e facciamo clic su Enable. Il progetto potrà così eseguire query, creare viste e altre operazioni BigQuery.

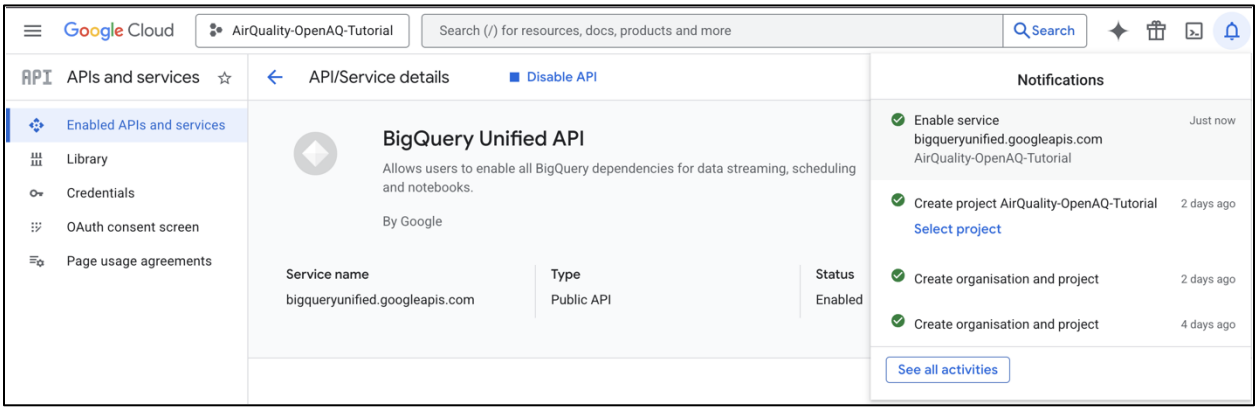


Figura 6.7 Abilitazione dell'API unificata di BigQuery

Dopo aver abilitato l'API, aggiungiamo il dataset OpenAQ. Il metodo più rapido consiste nell'utilizzare Google Cloud Marketplace: scegliamo Add Data e poi Public datasets per aprire il Marketplace. Cerchiamo OpenAQ, selezioniamo il risultato e scegliamo View dataset. In questo modo il dataset è disponibile nel progetto corrente senza trasferimenti dei dati né costi aggiuntivi di archiviazione (Figura 6.8).

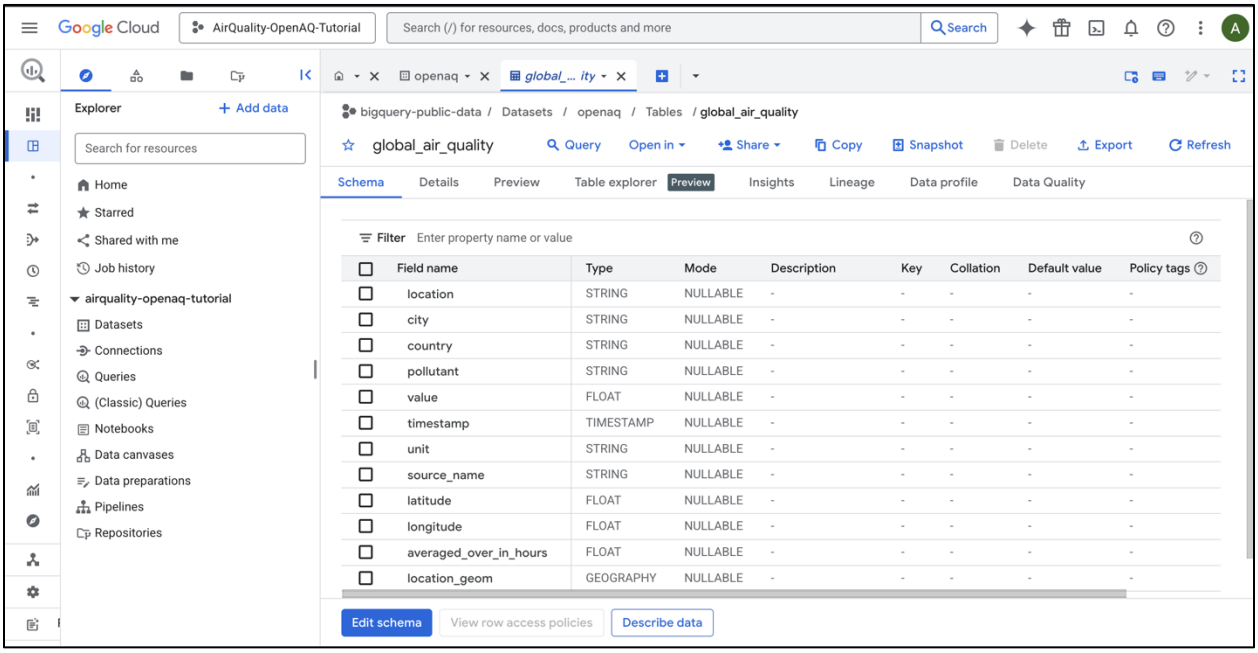


Figura 6.8 Esplorazione del dataset in Google BigQuery

Nel pannello centrale è selezionata la vista Schema, che mostra l'elenco strutturato dei campi (colonne) della tabella global_air_quality. Le intestazioni includono Field name, Type e Mode; tra gli attributi della qualità dell'aria figurano location, city, country, pollutant, value e timestamp, insieme a unit e source_name. Sono inoltre presenti latitude, longitude e il campo spaziale location_geom di tipo GEOGRAPHY, che indica la possibilità di svolgere analisi cartografiche e spaziali oltre alle consuete interrogazioni sulle serie temporali.

Per analizzare in modo riproducibile i dati OpenAQ e predisporli per una dashboard, sviluppiamo diverse query BigQuery da salvare come viste. La prima aggrega le misure PM2.5 in una serie giornaliera, per valutare l'andamento dell'inquinamento nel tempo. La seconda individua gli episodi più critici ordinando i giorni per concentrazione media di PM2.5. La terza calcola i livelli medi nei diversi siti, consentendo di confrontarli e individuare aree persistentemente più inquinate. La quarta seleziona per ogni località l'osservazione PM2.5 più recente, con coordinate geografiche valide, per creare un livello cartografico. Poiché il dataset BigQuery è una fotografia storica (con timestamp massimo nel 2022), gli intervalli temporali sono calcolati rispetto alla data più recente contenuta nei dati, garantendo risultati coerenti anche senza aggiornamenti quasi in tempo reale. Nel loro insieme, le viste costituiscono la base analitica da collegare a Looker Studio per dashboard interattive e report trasparenti (Figura 6.9).

1) Andamento giornaliero del PM2.5 negli ultimi 90 giorni rispetto a max_ts

```
WITH t AS (
 SELECT MAX(timestamp) AS max_ts
 FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
 DATE(g.timestamp) AS day,
 AVG(g.value) AS avg_pm25
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
 AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 90 DAY)
GROUP BY day
ORDER BY day;
```

Salva come vista: Run → SAVE → Save view → dataset openaq_dashboard → nome v_pm25_daily_trend

2) I 10 giorni peggiori (PM2.5 più elevato) nell'ultimo anno rispetto a max_ts

```
WITH t AS (
 SELECT MAX(timestamp) AS max_ts
 FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
 DATE(g.timestamp) AS day,
 AVG(g.value) AS avg_pm25,
 COUNT(*) AS measurements
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
 AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 365 DAY)
GROUP BY day
ORDER BY avg_pm25 DESC
LIMIT 10;
```

Salva come vista: v_pm25_worst_days

3) Le 15 località con PM2.5 medio più alto (ultimi 90 giorni rispetto a max_ts)

```
WITH t AS (
```

```
  SELECT MAX(timestamp) AS max_ts
  FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
  COALESCE(g.country, 'Unknown') AS country,
  COALESCE(g.city, 'Unknown') AS city,
  g.location,
  AVG(g.value) AS avg_pm25,
  COUNT(*) AS measurements
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
  AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 90 DAY)
GROUP BY country, city, location
HAVING measurements >= 50
ORDER BY avg_pm25 DESC
LIMIT 15;
```

Salva come vista: v_pm25_top_locations

4) Ultima misurazione PM2.5 per località (ultimi 14 giorni rispetto a max_ts)

```
WITH t AS (
  SELECT MAX(timestamp) AS max_ts
  FROM `bigquery-public-data.openaq.global_air_quality`
),
latest AS (
  SELECT
    g.location,
    g.city,
    g.country,
    g.latitude,
    g.longitude,
    g.value,
    g.unit,
    g.timestamp,
```

```
  ROW_NUMBER() OVER (PARTITION BY g.location ORDER BY g.timestamp DESC) AS rn
 FROM `bigquery-public-data.openaq.global_air_quality` g
 CROSS JOIN t
 WHERE g.pollutant = 'pm25'
  AND g.latitude IS NOT NULL
  AND g.longitude IS NOT NULL
  AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 14 DAY)
)
SELECT
 location,
 city,
 country,
 latitude,
 longitude,
 value AS latest_pm25,
 unit,
 timestamp AS latest_time
FROM latest
WHERE rn = 1;
```

Ora possiamo osservare i risultati (viste) nella Figura 6.9.

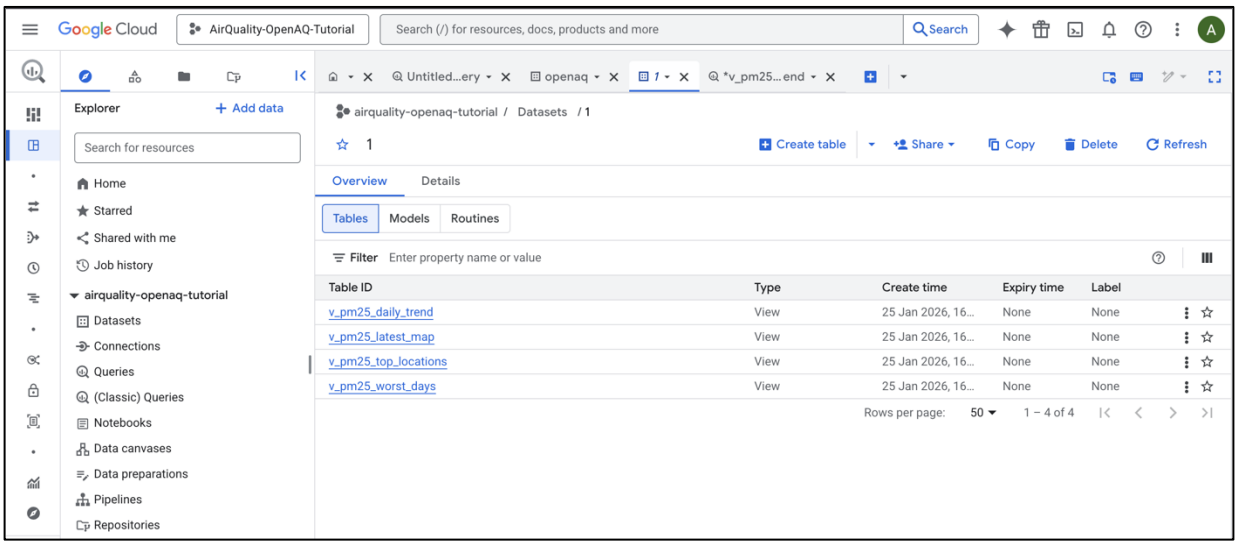


Figura 6.9 Quattro viste: risultati delle interrogazioni SQL

Successivamente utilizziamo Looker Studio per costruire una dashboard interattiva sulla qualità dell'aria, collegandola direttamente alle quattro viste BigQuery derivate dal dataset OpenAQ. Ogni

vista è una componente analitica riutilizzabile: andamento giornaliero del PM2.5, eventi estremi (giorni peggiori), classifiche delle località e livello cartografico delle osservazioni più recenti. Separare preparazione e visualizzazione migliora trasparenza e riproducibilità. Il risultato è una dashboard integrata con grafici, tabelle e mappe, condivisibile per la comunicazione dei risultati ambientali.

Per iniziare, apriamo Looker Studio e scegliamo Create → Report. Se richiesto, effettuiamo l'accesso con l'account Google e concediamo i permessi necessari per utilizzare le risorse Google Cloud. Nel report vuoto facciamo clic su Add data e selezioniamo BigQuery come connettore. Nel navigatore scegliamo il progetto corretto (ad esempio AirQuality-OpenAQ-Tutorial), il dataset openaq_dashboard e la vista v_pm25_daily_trend. Infine, facciamo clic su Add (e confermiamo Add to report, se richiesto) per utilizzare la vista dell'andamento giornaliero come prima fonte della dashboard (Figura 6.10).

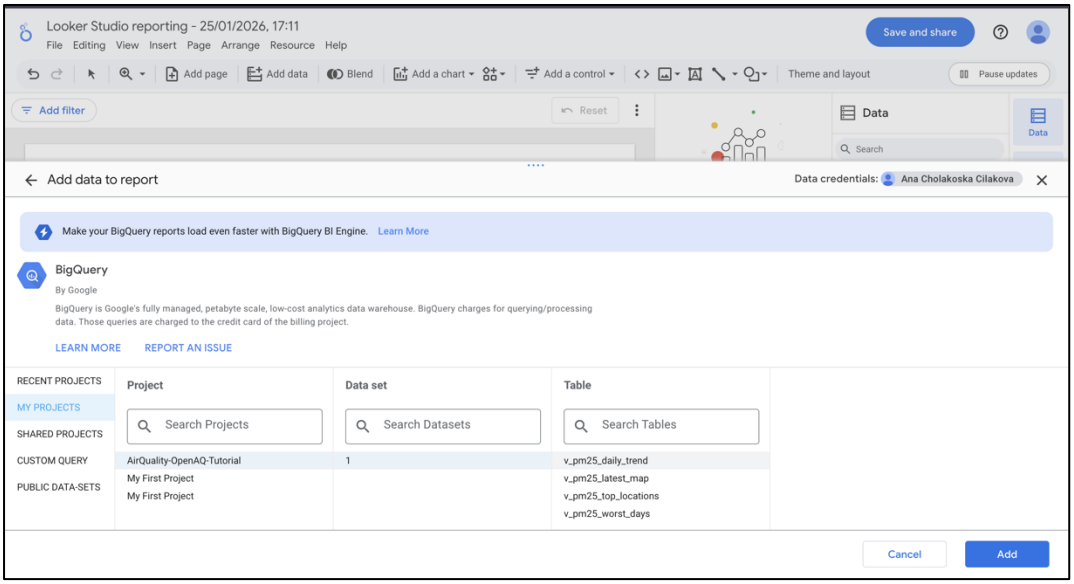


Figura 6.10 Aggiunta dei dati al report

Dopo aver aggiunto v_pm25_daily_trend come fonte, creiamo un grafico di serie temporale (Time Series) per rappresentare la variazione dell'inquinamento nel tempo. Inseriamo il grafico nel report e impostiamo il campo data (day) come dimensione principale e il PM2.5 medio (avg_pm25) come metrica. Questa visualizzazione riassume le oscillazioni quotidiane e le tendenze di più lungo periodo, aiutando a riconoscere aumenti persistenti, diminuzioni o picchi episodici eventualmente legati al meteo o alle emissioni.

Usando la vista v_pm25_top_locations, aggiungiamo un grafico a barre (Bar Chart) per confrontare le concentrazioni medie di PM2.5 tra i siti di monitoraggio. Impostiamo location (o city, secondo l'aggregazione desiderata) come dimensione e avg_pm25 come metrica, ordinando le barre in modo decrescente per evidenziare i siti più esposti. Il grafico permette di individuare rapidamente i punti critici e può essere accompagnato dal campo measurements per contestualizzare l'affidabilità della stima.

Per mettere in evidenza gli episodi di inquinamento estremo, inseriamo una tabella basata sulla vista v_pm25_worst_days. Configuriamo day come dimensione e avg_pm25 (e, se disponibile, measurements) come metriche, ordinando i valori medi di PM2.5 in ordine decrescente. La tabella consente di riportare valori esatti e confrontare i giorni con i picchi più alti: è utile per sintetizzare i periodi di rischio ambientale in un formato conciso e verificabile.

Per l'interpretazione spaziale, creiamo infine una mappa Google Maps usando la vista v_pm25_latest_map. Inseriamo la mappa e assegniamo latitude e longitude ai campi geografici, utilizzando latest_pm25 come metrica per rappresentare l'intensità delle osservazioni recenti, ad esempio tramite dimensione o intensità dei marcatori. Informazioni contestuali come location, city e latest_time nei tooltip permettono di esaminare i singoli punti, combinando posizione e valori quantitativi per mostrare dove si registrano le concentrazioni più elevate. Tutti i grafici sono visibili nella Figura 6.11.



Figura 6.11 Dashboard completa

Facoltativamente possiamo ottimizzare la quarta query per ridurre il carico computazionale e i costi, mantenendo lo stesso obiettivo analitico: recuperare l'ultima misurazione PM2.5 per ogni località. Innanzitutto, sostituiamo l'espressione di tabella comune e il filtro successivo (WHERE rn = 1) con QUALIFY, che applica il filtro della funzione finestra direttamente nella stessa scansione, semplificando l'esecuzione. Aggiungiamo poi controlli espliciti timestamp IS NOT NULL per escludere record non validi. Inoltre, definiamo gli ultimi 14 giorni rispetto al timestamp massimo presente nel dataset (MAX(timestamp)), anziché rispetto alla data corrente: è essenziale per snapshot storici come OpenAQ e consente di ottenere risultati non vuoti. Infine, selezioniamo soltanto i campi necessari per la mappa e il report (località, coordinate, valore, unità e tempo), limitando la lettura delle colonne e accelerando l'aggiornamento della dashboard. Il codice completo è riportato di seguito:

```
SELECT
 location,
 city,
 country,
```

```
 latitude,
 longitude,
 value AS latest_pm25,
 unit,
 timestamp AS latest_time
FROM `bigquery-public-data.openaq.global_air_quality`
WHERE pollutant = 'pm25'
 AND timestamp IS NOT NULL
 AND latitude IS NOT NULL
 AND longitude IS NOT NULL
 -- Use dataset's latest timestamp window if your snapshot is old:
 AND timestamp >= TIMESTAMP_SUB(
  (SELECT MAX(timestamp) FROM `bigquery-public-data.openaq.global_air_quality` WHERE timestamp IS NOT NULL),
  INTERVAL 14 DAY
 )
QUALIFY ROW_NUMBER() OVER (PARTITION BY location ORDER BY timestamp DESC) = 1;
```

Esercitazione pratica

Mini-progetto: scegliere un dataset ambientale (ad esempio qualità dell'aria, ambiente marino o biodiversità). Pulire, elaborare, interrogare e visualizzare i risultati usando gli strumenti Google Cloud. Presentare il flusso di lavoro e i risultati in una breve esposizione (5–10 minuti).

Verifica dell'apprendimento

1. Quale opzione descrive correttamente le tre V dei Big Data?

A) Volume = accuratezza dei dati, Velocità = dimensione dei dati, Varietà = velocità dei dati

B) Volume = quantità di dati, Velocità = rapidità di generazione/elaborazione, Varietà = differenti tipi e fonti di dati

C) Volume = numero di utenti, Velocità = velocità di Internet, Varietà = numero di server

D) Volume = costo di archiviazione, Velocità = costo delle query, Varietà = numero di dashboard

2. Quale associazione è corretta?

A) CSV = non strutturato, JSON = strutturato, immagini = semistrutturate

B) CSV = strutturato, JSON = semistrutturato, immagini/video = non strutturati

C) CSV = semistrutturato, JSON = non strutturato, immagini = strutturate

D) CSV = non strutturato, JSON = non strutturato, immagini = strutturate

3. Occorre analizzare rapidamente grandi log di eventi (con colonne timestamp, device_id, PM2.5, temperatura) archiviati come oggetti nel cloud. Qual è la scelta migliore?

A) Parquet/ORC (archiviazione colonnare)

B) File di testo semplice (.txt)

C) Immagini PNG

D) XML con elementi profondamente annidati

4. Quale opzione associa correttamente ciascun servizio alla sua funzione?

A) BigQuery = archiviazione di oggetti, Cloud Storage = analisi SQL, Dataflow = visualizzazione

B) Cloud Storage = archiviazione di oggetti, BigQuery = data warehouse analitico serverless, Dataflow = elaborazione distribuita dei dati (ETL)

C) BigQuery = coda di messaggi, Pub/Sub = dashboard, Looker = acquisizione dei dati

D) Dataflow = archiviazione di file, Cloud Storage = addestramento ML, BigQuery = gestione delle identità

5. Quale operazione riduce generalmente costi e tempi delle query in un data warehouse cloud come BigQuery, per dati di serie temporali?

A) Utilizzare SELECT * per non escludere colonne

B) Partizionare per data/ora e organizzare in cluster secondo chiavi di filtro comuni (ad esempio station_id/device_id)

C) Memorizzare tutto in un'unica grande colonna JSON per semplificare lo schema

D) Disabilitare la cache e ripetere le query per «riscaldare» il sistema

Riferimenti bibliografici e approfondimenti

[1] BigQuery dataset pubblici (panoramica e accesso). Disponibile: https://docs.cloud.google.com/bigquery/public-data, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/bigquery/public-data

[2] BigQuery: Introduzione all'ottimizzazione delle prestazioni delle query. Disponibile: https://docs.cloud.google.com/bigquery/docs/best-practices-performance-overview, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/bigquery/docs/best-practices-performance-overview

[3] BigQuery: Ottimizzare il calcolo delle query (buone pratiche). Disponibile: https://docs.cloud.google.com/bigquery/docs/best-practices-performance-compute, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/bigquery/docs/best-practices-performance-compute

[4] Looker Studio: Collegamento a Google BigQuery. Disponibile: https://docs.cloud.google.com/looker/docs/studio/connect-to-google-bigquery, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/looker/docs/studio/connect-to-google-bigquery

[5] Looker Studio Documentazione. Disponibile: https://docs.cloud.google.com/looker/docs/studio, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/looker/docs/studio

[6] Looker Studio guida introduttiva . Disponibile: https://docs.cloud.google.com/looker/docs/studio/quick-start-guide, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/looker/docs/studio/quick-start-guide

[7] Cloud Billing: Budget e avvisi di spesa. Disponibile: https://docs.cloud.google.com/billing/docs/how-to/budgets, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/billing/docs/how-to/budgets

[8] Cloud Billing portale della documentazione. Disponibile: https://docs.cloud.google.com/billing/docs, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/billing/docs

[9] BigQuery condivisione (già Analytics Hub): introduzione. Disponibile: https://docs.cloud.google.com/bigquery/docs/analytics-hub-introduction, Consultato il: 15 gennaio 2026https://docs.cloud.google.com/bigquery/docs/analytics-hub-introduction

[10] BigQuery spiegazione dell’architettura (Google Cloud Blog). Disponibile: https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview, Consultato il: 15 gennaio 2026https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview