

Report - Sistema di Gestione Dati Pazienti con Snowflake

1. Obiettivo del Progetto

Realizzare un sistema centralizzato, sicuro e scalabile su Snowflake per la gestione dei dati clinici e amministrativi di un ospedale, garantendo conformità GDPR, accesso controllato e aggiornamenti automatici dai file CSV su Google Cloud Storage (GCS).

2. Struttura del modello - Health Data Platform

2.1 Tabelle

Entità	Descrizione breve
pazienti	Informazioni anagrafiche dei pazienti
ricoveri	Dati sui ricoveri ospedalieri
visite	Dettagli delle visite effettuate
terapie	Farmaci prescritti ai pazienti
sensori_iot	Dati biometrici raccolti da sensori
utenti	Personale medico, infermieristico e amministrativo
accessi_log	Tracciamento degli accessi ai dati

2.2 Relazioni tra entità

Relazione	Tipo	Spiegazione
ricoveri.ID_Paziente → pazienti.ID_Paziente	1:N	Un paziente può avere più ricoveri
visite.ID_Paziente → pazienti.ID_Paziente	1:N	Un paziente può avere più visite
terapie.ID_Paziente → pazienti.ID_Paziente	1:N	Un paziente può avere più terapie
sensori_iot.ID_Paziente → pazienti.ID_Paziente	1:N	Un paziente può avere più rilevazioni IoT
accessi_log.ID_Utente → utenti.ID_Utente	1:N	Un utente può accedere più volte
accessi_log.ID_Paziente → pazienti.ID_Paziente	1:N	Ogni accesso si riferisce a un paziente

2.3 Chiavi Primarie / Esterne








Tabella	Chiave Primaria	Chiavi Esterne
pazienti	ID_Paziente	-
ricoveri	ID_Ricovero	ID_Paziente
visite	ID_Visita	ID_Paziente
terapie	ID_Terapia	ID_Paziente
sensori_iot	ID_Rilevazione	ID_Paziente
utenti	ID_Utente	-
accessi_log	ID_Accesso	ID_Utente, ID_Paziente

2.4 Sicurezza

- **RBAC (Role-Based Access Control):** Ruoli medico, infermiere, amministrativo
- **Mascheramento:** Codice fiscale mascherato a chi non è medico/infermiere
- **Log accessi:** ogni accesso tracciato in accessi_log


3. Setup Iniziale

- **Creazione dei file CSV sul PC:**

Nome	Ultima modifica	Tipo	Dimensione
 accessi_log.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 pazienti.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 ricoveri.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 sensori_iot.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 terapie.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 utenti.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB
 visite.csv	12/07/2025 11:10	Microsoft Excel Co...	1 KB

- **Creazione Progetto GCP:**

Nuovo progetto

 Nella quota rimangono 10 projects. Richiedi un aumento o elimina dei progetti. [Ulteriori informazioni](#)


[Manage Quotas](#)

Nome progetto *

ID progetto: snowflake-sg-465714. Non può essere modificato in un secondo momento.

[Modifica](#)

Località *

 Nessuna organizzazione [Sfogliala](#)

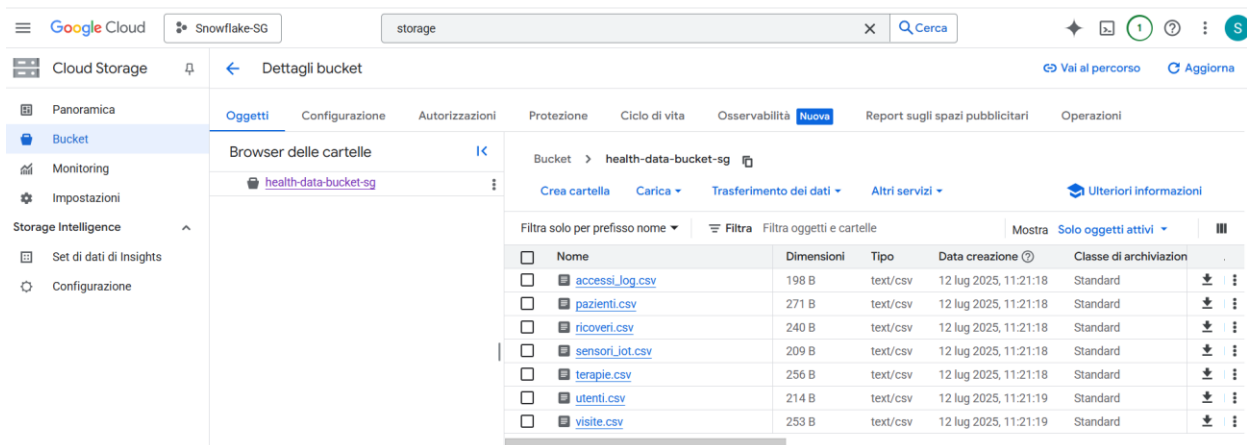
Organizzazione o cartella padre

[Crea](#) [Annulla](#)

- **Creazione Bucket:**

- **Nome bucket:** health-data-bucket-sg
- **Posizione:** europe-west1
- **Default storage class:** Standard
- **Controllo accessi:** Uniforme

- **Caricamento file CSV**



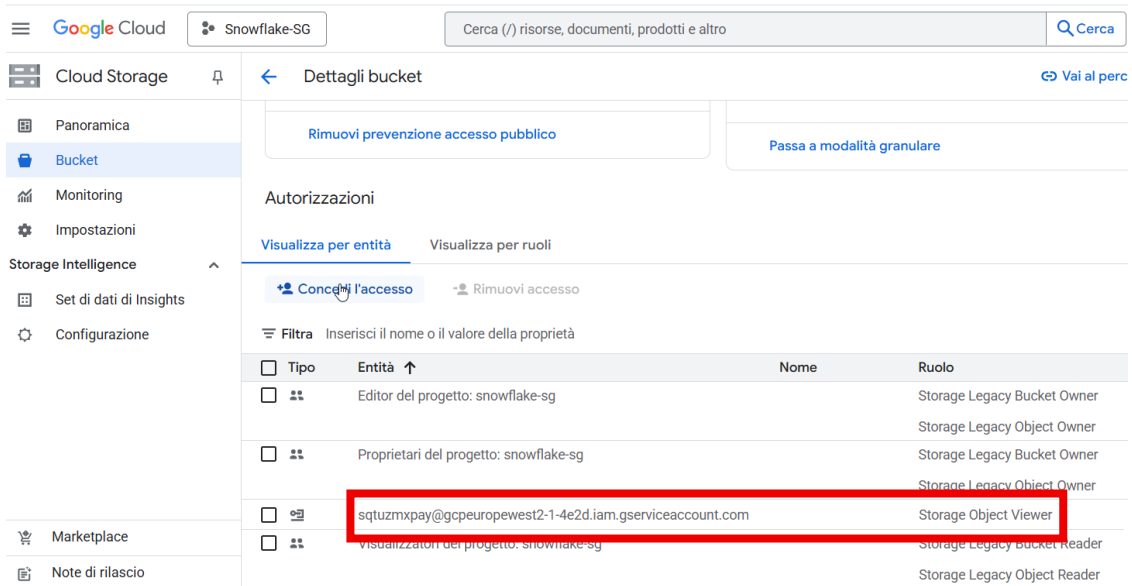
- **Creazione Integrazione con GCS:**

```
CREATE OR REPLACE STORAGE INTEGRATION gcs_integrazione
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = GCS
  ENABLED = TRUE
  STORAGE_ALLOWED_LOCATIONS = ('gcs://health-data-bucket-sg');

DESC INTEGRATION gcs_integrazione;
```

	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDER	String	GCS	
3	STORAGE_ALLOWED_LOCATIONS	List	gcs://health-data-bucket-sg	[]
4	STORAGE_BLOCKED_LOCATIONS	List		[]
5	STORAGE_GCP_SERVICE_ACCOUNT	String	sqtuzmxpay@gcpeuropewest2-1-4e2d.iam	
6	COMMENT	String		

- **Aggiunta autorizzazione accesso bucket GCS:**



- **Database e Schema:**

```
CREATE OR REPLACE DATABASE health_data;
USE DATABASE health_data;
```

```
CREATE OR REPLACE SCHEMA raw_data;
USE SCHEMA raw_data;
```

- **Stage Esterno:**

```
CREATE OR REPLACE STAGE health_data_stage
  URL = 'gcs://health-data-bucket-sg'
  STORAGE_INTEGRATION = gcs_integrazione
  FILE_FORMAT = (TYPE = CSV FIELD_OPTIONALLY_ENCLOSED_BY = '"' SKIP_HEADER = 1);
```

4. Creazione Tabelle Cliniche e Amministrative

Sono state create 7 tabelle per rappresentare:

- Dati anagrafici dei pazienti (*pazienti*)
- Ricoveri ospedalieri (*ricoveri*)
- Visite mediche (*visite*)
- Terapie (*terapie*)
- Dati sensori IoT (*sensori_iot*)
- Utenti ospedalieri (*utenti*)
- Log di accesso ai dati (*accessi_log*)

Per ogni tabella è stato effettuato il caricamento tramite:

```
CREATE OR REPLACE TABLE <nome_tabella> (
  <nome_colonna> <tipo_di_dato>,
  ...
);
```

5. Caricamento Dati da GCS

Per ogni tabella è stato effettuato il caricamento tramite:

```
COPY INTO <nome_tabella>
FROM @health_data_stage/<nome_file>.csv
FILE_FORMAT = (TYPE = CSV FIELD_OPTIONALLY_ENCLOSED_BY = '"' SKIP_HEADER = 1);
```

6. Sicurezza - Ruoli e Permessi RBAC

- **Ruoli creati:**

- *medico*
- *infermiera*
- *amministrativo*

- **Permessi assegnati:**

```
-- MEDICO
```

```
GRANT SELECT ON TABLE pazienti TO ROLE medico;
GRANT SELECT ON TABLE ricoveri TO ROLE medico;
GRANT SELECT ON TABLE visite TO ROLE medico;
GRANT SELECT ON TABLE terapie TO ROLE medico;
```

```
GRANT SELECT ON TABLE sensori_iot TO ROLE medico;
```

```
-- INFERMIERA
```

```
GRANT SELECT ON TABLE pazienti TO ROLE infermiera;
```

```
GRANT SELECT ON TABLE ricoveri TO ROLE infermiera;
```

```
-- AMMINISTRATIVO
```

```
GRANT SELECT ON TABLE utenti TO ROLE amministrativo;
```

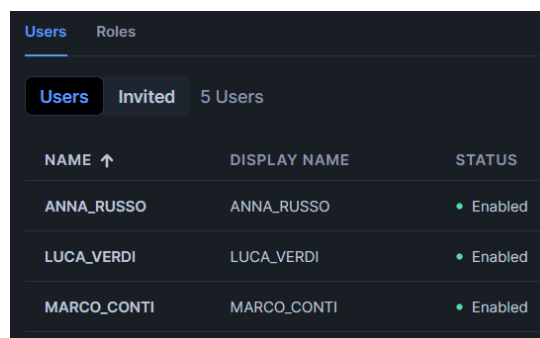
```
GRANT SELECT ON TABLE accessi_log TO ROLE amministrativo;
```

7. Creazione Utenti Snowflake

```
CREATE USER luca_verdi ... DEFAULT_ROLE = medico;
```

```
CREATE USER anna_russo ... DEFAULT_ROLE = infermiera;
```

```
CREATE USER marco_conti ... DEFAULT_ROLE = amministrativo;
```

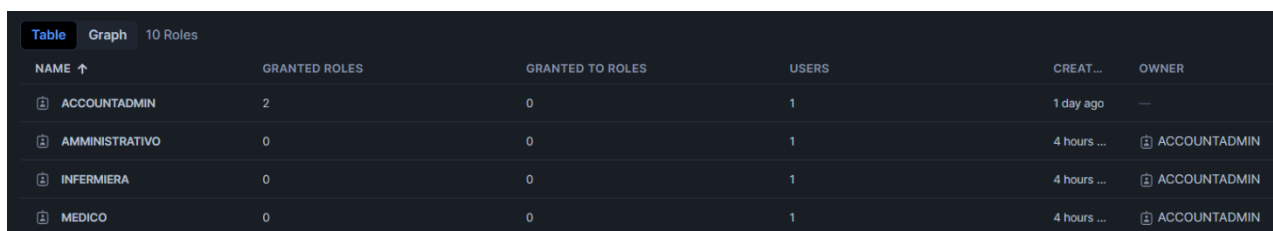


NAME ↑	DISPLAY NAME	STATUS
ANNA_RUSSO	ANNA_RUSSO	• Enabled
LUCA_VERDI	LUCA_VERDI	• Enabled
MARCO_CONTI	MARCO_CONTI	• Enabled

```
GRANT ROLE medico TO USER luca_verdi;
```

```
GRANT ROLE infermiera TO USER anna_russo;
```

```
GRANT ROLE amministrativo TO USER marco_conti;
```



NAME ↑	GRANTED ROLES	GRANTED TO ROLES	USERS	CREAT...	OWNER
ACCOUNTADMIN	2	0	1	1 day ago	—
AMMINISTRATIVO	0	0	1	4 hours ...	ACCOUNTADMIN
INFERMIERA	0	0	1	4 hours ...	ACCOUNTADMIN
MEDICO	0	0	1	4 hours ...	ACCOUNTADMIN

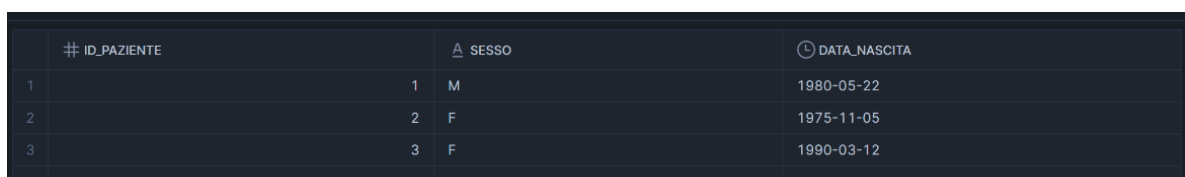
8. Viste

- Vista Anonimizzata per GDPR:**

```
CREATE OR REPLACE VIEW vista_anonimizzata_pazienti AS
```

```
SELECT ID_Paziente, Sesso, Data_Nascita
```

```
FROM pazienti;
```



	ID_PAZIENTE	SESSO	DATA_NASCITA
1	1	M	1980-05-22
2	2	F	1975-11-05
3	3	F	1990-03-12

- **Vista con JOIN per report visite:**

```
CREATE OR REPLACE VIEW vista_visite_completa AS
SELECT v.ID_Visita, p.Nome, p.Cognome, v.Data_Visita, v.Tipo_Visita, v.Medico
FROM visite v
JOIN pazienti p ON v.ID_Paziente = p.ID_Paziente;
```

	# ID_VISITA	A NOME	A COGNOME	🕒 DATA_VISITA	A TIPO_VISITA	A MEDICO
1	501	Mario	Rossi	2024-03-15	Controllo	Dr. Bianchi
2	502	Laura	Bianchi	2024-04-10	Visita di controllo post-ricove	Dr. Verdi
3	503	Giulia	Neri	2024-06-05	Neurologica	Dr. Neri

9. Mascheramento Dinamico (GDPR)

```
CREATE OR REPLACE MASKING POLICY mask_codice_fiscale AS (val STRING)
RETURNS STRING ->
CASE WHEN CURRENT_ROLE() IN ('medico', 'infermiera') THEN val ELSE '*****' END;

ALTER TABLE pazienti MODIFY COLUMN Codice_Fiscale SET MASKING POLICY mask_codice_fiscale;
```

	# ID_PAZIENTE	A NOME	A COGNOME	🕒 DATA_NASCITA	A SESSO	A CODICE_FISCALE	A INDIRIZZO
1	1	Mario	Rossi	1980-05-22	M	*****	Via Roma 10, Milano
2	2	Laura	Bianchi	1975-11-05	F	*****	Via Verdi 22, Torino
3	3	Giulia	Neri	1990-03-12	F	*****	Via Dante 3, Bologna

10. Task di Aggiornamento Automatico

- **Task principale orario:**

```
CREATE OR REPLACE TASK task_aggiorna_dati_gcs ... SCHEDULE = '1 HOUR' ...
BEGIN
COPY INTO ... -- tutte le tabelle
END;
ALTER TASK task_aggiorna_dati_gcs RESUME;
```

- **Task analitico per report ricoveri:**

```
CREATE TABLE report_ricoveri (...);
CREATE OR REPLACE TASK aggiorna_report_ricoveri SCHEDULE = 'USING CRON 0 * * * * UTC' AS
INSERT INTO report_ricoveri SELECT CURRENT_TIMESTAMP, COUNT(*) FROM ricoveri;
ALTER TASK aggiorna_report_ricoveri RESUME;
```

11. Conclusioni

Questo sistema cerca di soddisfare tutti i requisiti funzionali e normativi per una moderna gestione dei dati sanitari:

- Centralizzazione, automazione e protezione dei dati con accesso sicuro in base ai ruoli
- Conformità con GDPR e mascheramento dinamico
- Supporto alla business intelligence tramite viste e report periodici

HealthDataPro è ora in grado di offrire un'infrastruttura scalabile, efficiente e sicura per le esigenze ospedaliere moderne.