

RELAZIONE DI BASI DI DATI

Simone Lazzaro, Fabrizio Sardo, Sonia Spinelli

PARTE III

PROGETTO FISICO E VALIDAZIONE

Carico di lavoro

Il carico di lavoro per la parte di progettazione fisica della base di dati è costituito dalle tre seguenti operazioni:

- **Operazione 1 (SELEZIONE):** *"Determinare tutte le scuole del comune di Genova"*
- **Operazione 2 (SELEZIONE CON CONDIZIONE COMPLESSA):** *"Determinare le rilevazioni comprese tra il 20/09/2021 e il 21/12/2021 dove è stata registrata una temperatura maggiore di 17°C e un tasso di umidità maggiore del 75% "*
- **Operazione 3 (JOIN):** *"Determinare tutte le repliche possedute da scuole del comune di Novara messe a dimora in date successive al 01/01/2023 "*

Traducendo le interrogazioni dal linguaggio naturale in codice SQL:

- **Operazione 1 (SELEZIONE):**

```
SELECT * FROM scuola WHERE comune = 'Genova';
```

- **Operazione 2 (SELEZIONE CON CONDIZIONE COMPLESSA):**

```
SELECT *  
FROM rilevazione  
WHERE  
    dataOraRil BETWEEN '20-09-2021' AND '21-12-2021'  
    AND temperatura > 17 AND umidità > 75;
```

- **Operazione 3 (JOIN):**

```
SELECT siOccupadi.scuola, dataMessaADimora, numeroRep, gruppo,  
siOccupadi.specie  
FROM replica  
    JOIN specie ON replica.specie = nomeScientifico  
    JOIN siOccupadi ON nomeScientifico = siOccupadi.specie  
    JOIN scuola ON siOccupadi.scuola = codiceMeccanografico  
WHERE comune = 'Novara' AND dataMessaADimora > '01-01-2023';
```

Progetto fisico

Per ottimizzare le interrogazioni presenti nel carico di lavoro si intendono creare i seguenti indici:

1. Indice sulla relazione 'scuola' con chiave di ricerca l'attributo 'comune', ordinato, clusterizzato.
2. Indice sulla relazione 'rilevazione' con chiave di ricerca l'attributo 'dataOraRil', ordinato, clusterizzato.
3. Indice sulla relazione 'replica' con chiave di ricerca l'attributo 'dataMessaADimora', ordinato, clusterizzato.

Ognuno dei tre indici è l'unico presente sulla relazione a cui è legato perciò è conveniente che sia clusterizzato; inoltre, poiché PostgreSQL non implementa gli indici hash in modo efficace si è deciso di realizzare solo indici ordinati anche laddove sarebbe invece stato possibile utilizzare indici hash.

Operazione 1

L'indice ordinato clusterizzato $I_{comune}(scuola)$ è stato realizzato per rendere più efficiente la prima operazione.

Il cammino di accesso scelto senza l'implementazione di questo indice sarebbe la scansione sequenziale della relazione 'scuola' con filtro sull'attributo 'comune' (`SEQ_SCAN scuola FILTER comune = 'Genova'`).

Implementando l'indice si crea un nuovo cammino d'accesso (`INDEX_SCAN (I_comune (scuola), comune = 'Genova')`), il quale permette di accedere ad un numero di blocchi minore (nessun blocco "inutile" del file che materializza la relazione 'scuola' viene acceduto).

La clusterizzazione inoltre permette, una volta trovato il primo record che verifica la condizione, di accedere sequenzialmente a tutti e soli i record che compongono il risultato atteso, fino alla violazione del fattore di selezione.

```
CREATE INDEX idx_ord_comune_scuola
ON scuola(comune);

CLUSTER scuola
USING idx_ord_comune_scuola;
```

Operazione 2

Per la seconda operazione si è scelto di realizzare l'indice ordinato clusterizzato $I_{dataOraRil}(rilevazione)$.

I fattori booleani presenti sono:

- `dataOraRil BETWEEN '20-09-2021' AND '21-12-2021'`
- `temperatura > 17`
- `umidità > 75`

Si può quindi notare che il fattore con selettività più alta è il primo.

Ci si può aspettare quindi che una selezione sulla relazione 'rilevazione' con condizione `dataOraRil BETWEEN '20-09-2021' AND '21-12-2021'` restituisca un numero relativamente basso di tuple.

Per questo motivo si è deciso di utilizzare l'attributo 'dataOraRil' come chiave di ricerca per

l'indice.

L'indice è ordinato così da rendere più efficienti le ricerche per intervallo.

```
CREATE INDEX idx_ord_dataOraRil_rilevazione
ON rilevazione(dataOraRil);

CLUSTER rilevazione
USING idx_ord_dataOraRil_rilevazione;
```

Operazione 3

Per la terza operazione si è scelto di realizzare l'indice ordinato clusterizzato
I_{dataMessaADimora}(replica).

Si è scelto di non realizzare indici sugli attributi che costituiscono le condizioni di JOIN perché si è notato che, per l'implementazione di questo operatore logico, il sistema sceglie comunque l'HASH_JOIN, che risulta più efficiente del MERGE_JOIN in quanto, in questo caso, l'operatore fisico MERGE_JOIN richiederebbe delle trasformazioni intermedie.

Sono stati fatti dei test: creando indici ordinati clusterizzati su 'specie' per la relazione 'replica' e su 'nomeScientifico' per la relazione 'specie', forzando il sistema a scegliere il MERGE_JOIN con l'opzione `SET enable_hashjoin = off`, si è potuto verificare empiricamente quanto detto.

Quindi per la relazione 'replica' il sistema è portato a scegliere il cammino d'accesso:

```
INDEX_SCAN (IdataMessaADimora (replica), dataMessaADimora > '01-01-2023').
```

Inoltre, poiché l'indice I_{comune}(scuola) è già presente nella base di dati, il cammino d'accesso sulla relazione 'scuola' è: `INDEX_SCAN (Icomune (scuola), comune = 'Novara')`.

Per le altre relazioni il cammino d'accesso è la scansione sequenziale.

```
CREATE INDEX idx_ord_dataMessaADimora_replica
ON replica(dataMessaADimora);

CLUSTER replica
USING idx_ord_dataMessaADimora_replica;
```



Dimensioni delle tabelle

```
SELECT C.relname AS tabella, C.relpages AS numeroBlocchi, C.reltuples AS
numeroTuple
FROM pg_namespace N JOIN pg_class C ON N.oid = C.relnamespace
WHERE N.nspname = 'ortiScolastici' AND relname IN ('classe', 'finanziamento',
'gruppo', 'orto', 'persona', 'replica', 'responsabile', 'rilevazione', 'scuola', 'sensor
e', 'sioccupadi', 'specie');
```

tabella	numeroblocchi	numeropagine
orto	2	101
classe	2	100
sensore	2	200
gruppo	47	5001
replica	219	20001
responsabile	1	50
persona	8	600
finanziamento	2	100
specie	170	20000
sioccupadi	74	10001
scuola	197	20000
rilevazione	453	20000

Descrizione dei piani di esecuzione

OPERAZIONE 1

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
VISUALIZZAZIONE	 <p>scuola</p> <p>Seq Scan on ortiScolasticiLarge.scuola as scuola (cost=0..447 rows=6784 width=48) (actual=0.172..7.283 rows=6784 loops=1) Filter: ((scuola.comune)::text = 'Genova'::text) Rows Removed by Filter = 13216</p>	 <p>ortiScolasticiLarge.idx_ord_comune_scuola</p> <p>Index Scan using idx_ord_comune_scuola on ortiScolasticiLarge.scuola as scuola (cost=0.29..277.01 rows=6784 width=48) (actual=0.06..2.767 rows=6784 loops=1) Index Cond: ((scuola.comune)::text = 'Genova'::text)</p>

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
TEMPI	Total query runtime: 125 msec Exclusive timing: 5.498 ms Inclusive timing: 5.498 ms	Total query runtime: 66 msec Exclusive timing: 1.376 ms Inclusive timing: 1.376 ms
DESCRIZIONE	Il sistema ha scelto SEQ_SCAN con filtro comune = 'Genova' che ha rimosso 13216 righe.	Il sistema ha scelto INDEX_SCAN con cammino d'accesso (idx_ord_comune_scuola, comune = 'Genova')

La creazione dell'indice ha garantito un'ottimizzazione della query infatti, come si può vedere dalla tabella, l'introduzione dell'indice ha permesso una riduzione significativa dei tempi e dei costi (circa la metà).

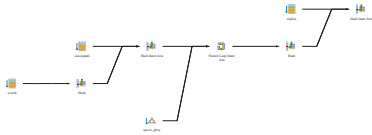
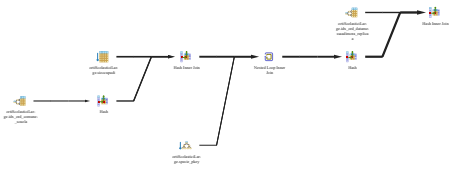
OPERAZIONE 2

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
VISUALIZZAZIONE	 <p>Bitmap Heap Scan on ortiScolasticiLarge.rilevazione as rilevazione (cost=10.75..410.81 rows=27 width=134) (actual=0.151..0.411 rows=32 loops=1) Filter: ((rilevazione.temperatura > '17'::numeric) AND (rilevazione."umidità" > '75'::numeric)) Rows Removed by Filter: 222 Recheck Cond: ((rilevazione.dataorartil >= '2021- 09-20 00:00:00'::timestamp without time zone) AND (rilevazione.dataorartil <= '2021- 12-21 00:00:00'::timestamp without time zone)) Heap Blocks: exact=64</p> <p>Bitmap Index Scan using rilevazione_pkey (cost=0..10.75 rows=246 width=0) (actual=0.075..0.075 rows=254 loops=1) Index Cond: ((rilevazione.dataorartil >= '2021- 09-20 00:00:00'::timestamp without time zone) AND (rilevazione.dataorartil <= '2021- 12-21 00:00:00'::timestamp without time zone))</p>	 <p>ortiScolasticiLar- ge.idx_ord_dataor- artil_rilevazione</p> <p>Index Scan using idx_ord_dataorartil_rilevazione on prova3.rilevazione as rilevazione (cost=0.29..19.44 rows=27 width=134) (actual=0.031..0.258 rows=32 loops=1) Filter: ((rilevazione.temperatura > '17'::numeric) AND (rilevazione."umidità" > '75'::numeric)) Index Cond: ((rilevazione.dataorartil >= '2021-09-20 00:00:00'::timestamp without time zone) AND (rilevazione.dataorartil <= '2021-12-21 00:00:00'::timestamp without time zone)) Rows Removed by Filter: 222</p>
TEMPI	<p>Total query runtime: 117 msec Exclusive timing: 0.426 ms Inclusive timing: 0.508 ms</p>	<p>Total query runtime: 68 msec Exclusive timing: 0.106 ms Inclusive timing: 0.106 ms</p>

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
DESCRIZIONE	Il sistema ha scelto la combinazione di Bitmap Heap Scan e Bitmap Index Scan con filtri temperatura > 17 e umidità > 75.	Il sistema ha scelto INDEX_SCAN con cammino d'accesso (idx_ord_dataOraRil_rilevazione, dataOraRil >= '2021-09-20' AND dataOraRil <= '2021-12-21') e filtri temperatura > 17 e umidità > 75

Anche in questo caso i tempi e i costi si sono notevolmente ridotti in seguito alla creazione dell'indice.

OPERAZIONE 3

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
VISUALIZZAZIONE	 <p>1) Hash Inner Join (cost=692.63..1163.28 rows=2 width=40) (actual=5.79..7.618 rows=822 loops=1) Hash Cond: ((replica.specie)::text = (specie.nomescientifico)::text)</p> <p>2) -> Seq Scan on ortiScolasticiLarge.replica as replica (cost=0..469.01 rows=432 width=21) (actual=0.03..1.835 rows=437 loops=1) Filter: (replica.datamessaadimora > '2023-01-01'::date) Rows Removed by Filter: 19564</p> <p>3) -> Hash (cost=691.74..691.74 rows=71 width=43) (actual=5.608..5.608 rows=65 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 13 kB</p> <p>4) -> Nested Loop Inner Join (cost=449.06..691.74 rows=71 width=43) (actual=3.213..5.565 rows=65 loops=1)</p> <p>5) -> Hash Inner Join (cost=448.77..661 rows=71 width=28) (actual=3.155..4.941 rows=65 loops=1) Hash Cond: (sioccupadi.scuola = scuola.codicemeccanografico)</p> <p>6) -> Seq Scan on ortiScolasticiLarge.sioccupadi as sioccupadi (cost=0..174.01 rows=10001 width=28) (actual=0.008..0.952 rows=10001 loops=1)</p> <p>7) -> Hash (cost=447..447 rows=142 width=11) (actual=2.994..2.994 rows=142 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 14 kB</p>	 <p>1) Hash Inner Join (cost=257.69..282.89 rows=2 width=40) (actual=3.253..3.675 rows=822 loops=1) Hash Cond: ((replica.specie)::text = (specie.nomescientifico)::text)</p> <p>2) -> Index Scan using idx_ord_datamessaadimora_replica on ortiScolasticiLarge.replica as replica (cost=0.29..23.85 rows=432 width=21) (actual=0.182..0.37 rows=437 loops=1) Index Cond: (replica.datamessaadimora > '2023-01-01'::date)</p> <p>3) -> Hash (cost=256.51..256.51 rows=71 width=43) (actual=3.037..3.037 rows=65 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 13 kB</p> <p>4) -> Nested Loop Inner Join (cost=13.83..256.51 rows=71 width=43) (actual=0.527..3 rows=65 loops=1)</p> <p>5) -> Hash Inner Join (cost=13.55..225.77 rows=71 width=28) (actual=0.451..2.369 rows=65 loops=1) Hash Cond: (sioccupadi.scuola = scuola.codicemeccanografico)</p> <p>6) -> Seq Scan on ortiScolasticiLarge.sioccupadi as sioccupadi (cost=0..174.01 rows=10001 width=28) (actual=0.056..0.909 rows=10001 loops=1)</p> <p>7) -> Hash (cost=11.77..11.77 rows=142 width=11) (actual=0.203..0.203 rows=142 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 14 kB</p> <p>8) -> Index Scan using idx_ord_comune_scuola on ortiScolasticiLarge.scuola as scuola (cost=0.29..11.77 rows=142 width=11)</p>

	Prima della creazione dello schema fisico	Dopo la creazione dello schema fisico
	<p>Usage: 14 kB</p> <p>8) -> Seq Scan on ortiScolasticiLarge.scuola as scuola (cost=0..447 rows=142 width=11) (actual=0.008..2.902 rows=142 loops=1) Filter: ((scuola.comune)::text = 'Novara'::text) Rows Removed by Filter: 19858</p> <p>9) -> Index Only Scan using specie_pkey on ortiScolasticiLarge.specie as specie (cost=0.29..0.42 rows=1 width=15) (actual=0.009..0.009 rows=1 loops=65) Index Cond: (specie.nomescientifico = (sioccupadi.specie)::text)</p>	<p>(actual=0.081..0.162 rows=142 loops=1) Index Cond: ((scuola.comune)::text = 'Novara'::text)</p> <p>9) -> Index Only Scan using specie_pkey on ortiScolasticiLarge.specie as specie (cost=0.29..0.42 rows=1 width=15) (actual=0.008..0.009 rows=1 loops=65) Index Cond: (specie.nomescientifico = (sioccupadi.specie)::text)</p>
TEMPI	<p>Total query runtime: 79 msec Exclusive timing: 0.176 ms Inclusive timing: 7.618 ms</p>	<p>Total query runtime: 72 msec Exclusive timing: 0.268 ms Inclusive timing: 3.675 ms</p>
DESCRIZIONE	<p>Il piano scelto dal sistema è una sequenza di HASH INNER JOIN, SEQ_SCAN (con filtri dataMessaADimora > '2023-01- 01' e comune = 'Novara'), HASH e NESTED LOOP JOIN</p>	<p>Il piano scelto dal sistema è una sequenza di HASH INNER JOIN, HASH, SEQ_SCAN, NESTED LOOP JOIN e INDEX SCAN con cammini d'accesso (idx_ord_datamessaadimora_replica, datamessaadimora > '2023-01-01') e (idx_ord_comune_scuola, comune = 'Novara')</p>

La creazione degli indici idx_ord_datamessaadimora_replica e idx_ord_comune_scuola garantiscono migliori prestazioni in termini di costi mentre, per quanto riguarda i tempi non sembra esserci una differenza significativa.

Questo può essere dovuto al fatto che, nonostante indici permettano di accedere ad un numero minore di blocchi per le relazioni 'replica' e 'scuola', ciò ha poca influenza nelle operazioni di JOIN.

CONTROLLO DELL'ACCESSO

Politica di accesso

Come indicato nella parte di progettazione logica si è pensato di considerare 'scuola' ed 'istituto' sinonimi; per questo motivo i ruoli di 'referente di istituto' e di 'referente della scuola' sono stati accorpati.

Tabella\Ruolo	Gestore del progetto	Referente della scuola	Insegnante	Studente
Persona	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	Nessun privilegio	Nessun privilegio	Nessun privilegio
Scuola	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	Nessun privilegio	Nessun privilegio	Nessun privilegio
Classe	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (tutti senza grant option)	Nessun privilegio	Nessun privilegio
Specie	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT (con grant option)	SELECT (con grant option)	SELECT (senza grant option)
Orto	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (con grant option)	SELECT (senza grant option)
Sensore	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (con grant option)	SELECT (senza grant option)

Tabella\Ruolo	Gestore del progetto	Referente della scuola	Insegnante	Studente
Gruppo	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (con grant option)	SELECT (senza grant option)
Replica	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (senza grant option)
Responsabile	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT (senza grant option)	Nessun privilegio	Nessun privilegio
Rilevazione	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (senza grant option)
Finanziamento	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	Nessun privilegio	Nessun privilegio	Nessun privilegio
SiOccupaNi	SELECT, INSERT, UPDATE, DELETE (tutti con grant option)	SELECT, INSERT (solo SELECT con grant option)	SELECT (senza grant option)	Nessun privilegio

Si è scelto di seguire il principio del minimo privilegio: sono stati dati i privilegi minimi che consentono ad ogni ruolo di svolgere le proprie attività.

La gerarchia dei ruoli è così costituita:

Studente ≤ Insegnante ≤ Referente della scuola ≤ Gestore globale del progetto

L'aggiunta/modifica/cancellazione di una persona, di una scuola, di un finanziamento e di un responsabile sono privilegi concessi solo al gestore globale del progetto. Per motivi di privacy anche la stampa dei contenuti di quelle tabelle è riservata al gestore, tranne per quanto riguarda la tabella 'Responsabile', dal momento che il referente della scuola ha bisogno di conoscere i codici dei responsabili della sua scuola.

Il referente della scuola può aggiungere e stampare le classi, gli orti, i sensori, i gruppi, le repliche

e le rilevazioni. Può inoltre scegliere di quali specie si occupa la sua scuola.

Tutti i ruoli possono visualizzare le specie scelte dal gestore del progetto.

Un insegnante può anche aggiungere repliche e rilevazioni in quanto potrebbe essere il docente di riferimento per l'iniziativa di una classe.

-- Creazione dei ruoli

```
CREATE ROLE GestoreProgetto;  
CREATE ROLE ReferenteScuola;  
CREATE ROLE Insegnante;  
CREATE ROLE Studente;
```

-- Creazione della gerarchia

```
GRANT Studente TO Insegnante;  
GRANT Insegnante TO ReferenteScuola;  
GRANT ReferenteScuola TO GestoreProgetto;
```

-- Privilegi GestoreProgetto

```
GRANT ALL PRIVILEGES ON SCHEMA "ortiScolasticiLarge" TO GestoreProgetto WITH  
GRANT OPTION;  
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA "ortiScolasticiLarge" TO  
GestoreProgetto WITH GRANT OPTION;
```

-- Privilegi Studente

```
GRANT SELECT ON TABLE Specie TO Studente;  
GRANT SELECT ON TABLE Orto TO Studente;  
GRANT SELECT ON TABLE Sensore TO Studente;  
GRANT SELECT ON TABLE Gruppo TO Studente;  
GRANT SELECT ON TABLE Replica TO Studente;  
GRANT SELECT ON TABLE Rilevazione TO Studente;
```

-- Privilegi Insegnante

```
GRANT SELECT ON TABLE Specie TO Insegnante WITH GRANT OPTION;  
GRANT SELECT ON TABLE Orto TO Insegnante WITH GRANT OPTION;  
GRANT SELECT ON TABLE Sensore TO Insegnante WITH GRANT OPTION;  
GRANT SELECT ON TABLE Gruppo TO Insegnante WITH GRANT OPTION;  
GRANT SELECT ON TABLE Replica TO Insegnante WITH GRANT OPTION;  
GRANT INSERT ON TABLE Replica TO Insegnante;  
GRANT SELECT ON TABLE Rilevazione TO Insegnante WITH GRANT OPTION;  
GRANT INSERT ON TABLE Rilevazione TO Insegnante;  
GRANT SELECT ON TABLE SiOccupadi TO Insegnante;
```

-- Privilegi ReferenteScuola

```
GRANT SELECT, INSERT ON TABLE Classe TO ReferenteScuola;  
GRANT INSERT ON TABLE Orto TO ReferenteScuola;  
GRANT INSERT ON TABLE Sensore TO ReferenteScuola;  
GRANT INSERT ON TABLE Gruppo TO ReferenteScuola;  
GRANT INSERT ON TABLE Replica TO ReferenteScuola WITH GRANT OPTION;  
GRANT SELECT ON TABLE Responsabile TO ReferenteScuola;  
GRANT INSERT ON TABLE Rilevazione TO ReferenteScuola WITH GRANT OPTION;  
GRANT SELECT ON TABLE SiOccupadi TO ReferenteScuola WITH GRANT OPTION;
```

```
GRANT INSERT ON TABLE SiOccupaNi TO ReferenteScuola;
```

Assegnazione dei ruoli

Sono stati creati cinque utenti a cui sono stati assegnati i ruoli nel modo seguente:

```
CREATE USER alice PASSWORD 'alice';
CREATE USER bob PASSWORD 'bob';
CREATE USER charlie PASSWORD 'charlie';
CREATE USER dave PASSWORD 'dave';
CREATE USER eve PASSWORD 'eve';

GRANT USAGE ON SCHEMA "ortiscolasticiLarge" TO alice WITH GRANT OPTION;
GRANT USAGE ON SCHEMA "ortiscolasticiLarge" TO bob WITH GRANT OPTION;
GRANT USAGE ON SCHEMA "ortiscolasticiLarge" TO charlie;
GRANT USAGE ON SCHEMA "ortiscolasticiLarge" TO dave;
GRANT USAGE ON SCHEMA "ortiscolasticiLarge" TO eve;

GRANT GestoreProgetto TO alice;
GRANT ReferenteScuola TO bob;
GRANT Insegnante TO charlie;
GRANT Insegnante TO dave;
GRANT Studente TO eve;
```