

Parte III

9. Progettazione fisica

A) Interrogazioni del carico di lavoro

Linguaggio naturale

1. I nomi degli orti che contengono cetrioli
2. Le piante su cui sono state fatte rilevazioni da un dirigente, referente di una scuola che ha ricevuto almeno un finanziamento
3. I parametri ambientali con $ph > 4$ o temperatura < 10 e umidità compresa fra il 10 e il 90 % maggiore rilevati nel 2022

SQL

1. `select orti.nome from orti join gruppi on orti.nome = gruppi.orto and orti.possessore = gruppi.scuola join specie on gruppi.specie = specie.nome_scientifico where specie.nome_comune = 'cetriolo';`
2. `select rilevazioni.replica, rilevazioni.gruppo from rilevazioni, persone, destinatari_finanziamenti where rilevazioni.persona_rilevazione = persone.mail and persone.ruolo = 'dirigente' and persone.scuola = destinatari_finanziamenti.scuola and persone.referente = true;`
3. `select data_ora_rilevazione, orto, scuola from parametri_ambientali where extract (year from data_ora_rilevazione) = 2022 and (ph >= 3 or temperatura < 10) and (umidita > 10 and umidita < 90);`

B) Progetto fisico

- Indice hash su `specie.nome_comune` (non clusterizzabile con postgresql): ottimizza la ricerca della pianta portando alla scansione di molte meno tuple
- Indice hash su `gruppi.specie` (non clusterizzabile con postgresql): viene usato nel index nested loop per diminuire il numero di tuple confrontate e favorire il pipelining
- Indice hash su `orti.nome` (non clusterizzabile con postgresql): viene usato nel index nested loop per diminuire il numero di tuple confrontate e favorire il pipelining
- Indice hash su `persone.ruolo` (non clusterizzabile con postgresql): viene utilizzato per ottimizzare la ricerca del ruolo sulla tabella di base portando alla scansione di molte meno tuple
- Indice hash su `rilevazioni.persona_rilevazione` (non clusterizzabile con postgresql): viene usato nel index nested loop, sfruttando prima un bitmap index scan, per diminuire il numero di tuple confrontate e favorire il pipelining
- Indice ordinato su `destinatari_finanziamenti.scuola` (clusterizzato): viene usato nel index nested loop, sfruttando prima un bitmap index scan, per diminuire il numero di tuple confrontate e favorire il pipelining
- Indice ordinato su `parametri_ambientali.umidita` (clusterizzato), perché nella terza query il fattore booleano è `umidita > 10` oppure `umidita < 90` quindi verrà sempre usato solo l'indice sulla umidità e le altre condizioni verranno verificate sequenzialmente

Altri indici vengono ignorati nel momento della selezione del piano fisico

C) Tuple inserite e la dimensione in blocchi

Tabelle direttamente utilizzate

NOME	TUPLE	BLOCCHI
Orti	1500	38
Gruppi	1500	38
Specie	1500	79
Rilevazioni	1500	42
Persone	1500	38
Destinatari_finanziamenti	1000	23
Parametri_ambientali	1500	19

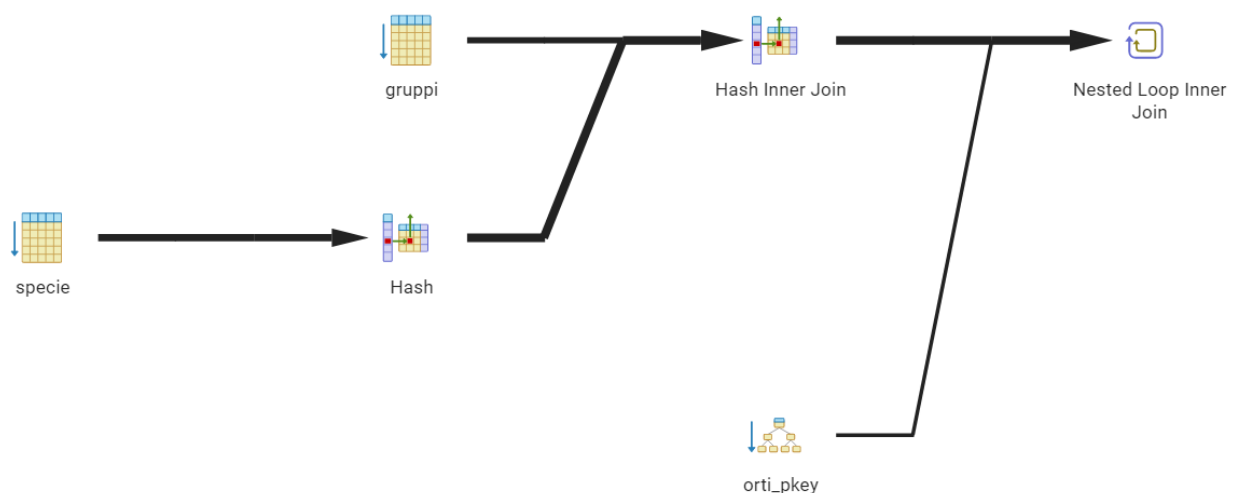
Tabelle riempite per rispettare i vicoli di chiave esterna

NOME	TUPLE	BLOCCHI
Classi	1500	33
Scuola	2000	23
Sensori	1500	59
Finanziamenti	2000	70
Piante	1500	40

D) Descrizione piani di esecuzione

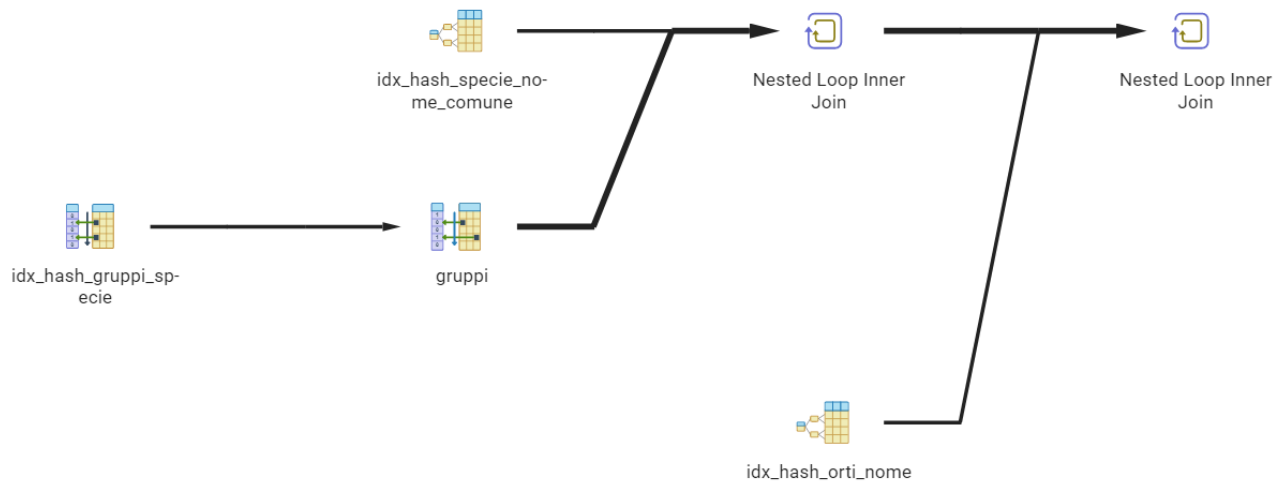
Query 1

Piano prima della creazione dello schema fisico



L'accesso alla relazione di base "specie" permette di estrarre le tuple desiderate (quelle aventi come nome comune "cetriolo") mediante una ricerca sequenziale, che vengono poi joinate (mediante hash join) con la relazione "gruppi". Viene poi eseguita una index nested loop join, usando come relazione inner il join precedente, e come outer l'indice generato automaticamente da PostgreSQL, relativo alla chiave primaria di "orti".

Piano dopo la creazione dello schema fisico



L'accesso alla relazione di base "specie" permette di estrarre le tuple desiderate (quelle aventi nome comune "cetriolo") mediante una ricerca ad indice sull'attributo nome_comune. Le tuple vengono poi joinate (index nested loop) con la relazione "gruppi" (viene utilizzato per l'accesso a "gruppi" una bitmap Index scan). Viene poi eseguita una index nested loop join, usando come relazione inner il join precedente, e come outer l'indice hash sull'attributo nome di "orti" con filtro sull'attributo possessore)

Confronto tempi

Prima:

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Nested Loop Inner Join (cost=98.04..155.14 rows=1 width=11) (actual=...	0.335 ms	0.335 ms	↓ 1	0	1	1
2.	→ Hash Inner Join (cost=97.76..154.71 rows=1 width=17) (actual=... Hash Cond: ((gruppi.specie)::text = (specie.nome_scientifico)::text)	0.005 ms	0.335 ms	↓ 1	0	1	1
3.	→ Seq Scan on gruppi as gruppi (cost=0..53 rows=1500 width=...	0.011 ms	0.011 ms	↑ 1500	1	1500	1
4.	→ Hash (cost=97.75..97.75 rows=1 width=129) (actual=0.32.... Buckets: 1024 Batches: 1 Memory Usage: 8 kB	0.32 ms	0.32 ms	↓ 1	0	1	1
5.	→ Seq Scan on specie as specie (cost=0..97.75 rows=1 ... Filter: ((nome_comune)::text = 'cetriolo'::text) Rows Removed by Filter: 1500	0.32 ms	0.32 ms	↓ 1	0	1	1
6.	→ Index Only Scan using orti_pkey on orti as orti (cost=0.28..0.43 r... Index Cond: ((nome = (gruppi.orti)::text) AND (possessore = (gruppi.sci...	1 ms	1 ms	↓ 1	0	1	0

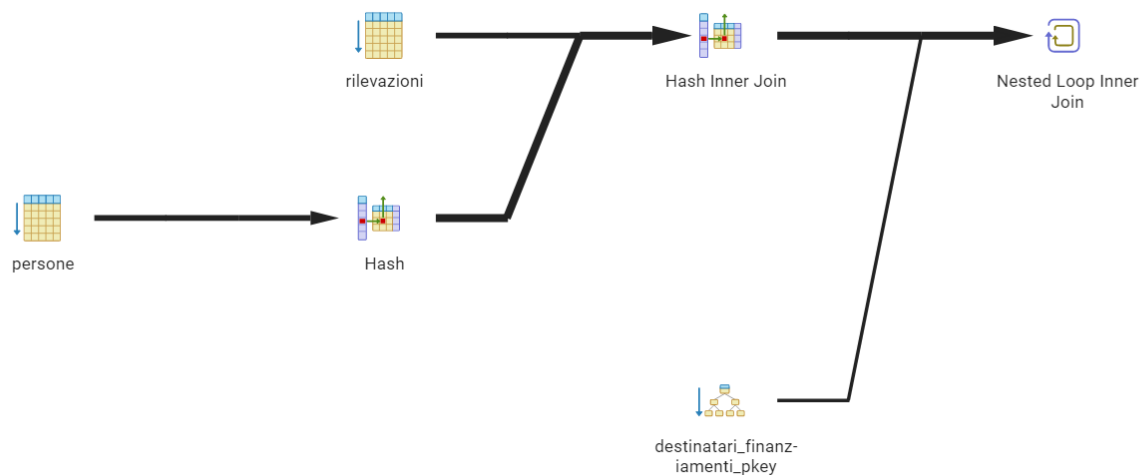
Dopo:

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Nested Loop Inner Join (cost=4.04..26.89 rows=1 width=11) (actual=...	0.002 ms	0.01 ms	↓ 1	0	1	1
2.	→ Nested Loop Inner Join (cost=4.04..26.73 rows=1 width=17) (act...	0.009 ms	0.009 ms	↓ 1	0	1	1
3.	→ Index Scan using idx_hash_specie_nome_comune on specie... Index Cond: ((nome_comune)::text = 'cetriolo'::text)	0.009 ms	0.009 ms	↓ 1	0	1	1
4.	→ Bitmap Heap Scan on gruppi as gruppi (cost=4.04..18.66 ro... Recheck Cond: ((specie)::text = (specie.nome_scientifico)::text) Heap Blocks: exact=0	0	0	↓ 1	0	5	0
5.	→ Bitmap Index Scan using idx_hash_gruppi_specie (cost=... Index Cond: ((specie)::text = (specie.nome_scientifico)::text)	1 ms	1 ms	↓ 1	0	5	0
6.	→ Index Scan using idx_hash_orti_nome on orti as orti (cost=0..0.1... Filter: ((gruppi.scuola)::text = (possessore)::text) Index Cond: ((nome)::text = (gruppi.orti)::text) Rows Removed by Filter: 0	1 ms	1 ms	↓ 1	0	1	0

Si può notare un incremento prestazionale di circa 300 volte, rendendo trascurabile il tempo di esecuzione. Si nota anche una riduzione notevole delle tuple intermedie. (Il fatto che siano così poche le tuple intermedie è corretto, in quanto i dati generati da Datanamic non risultano nel risultato della query)

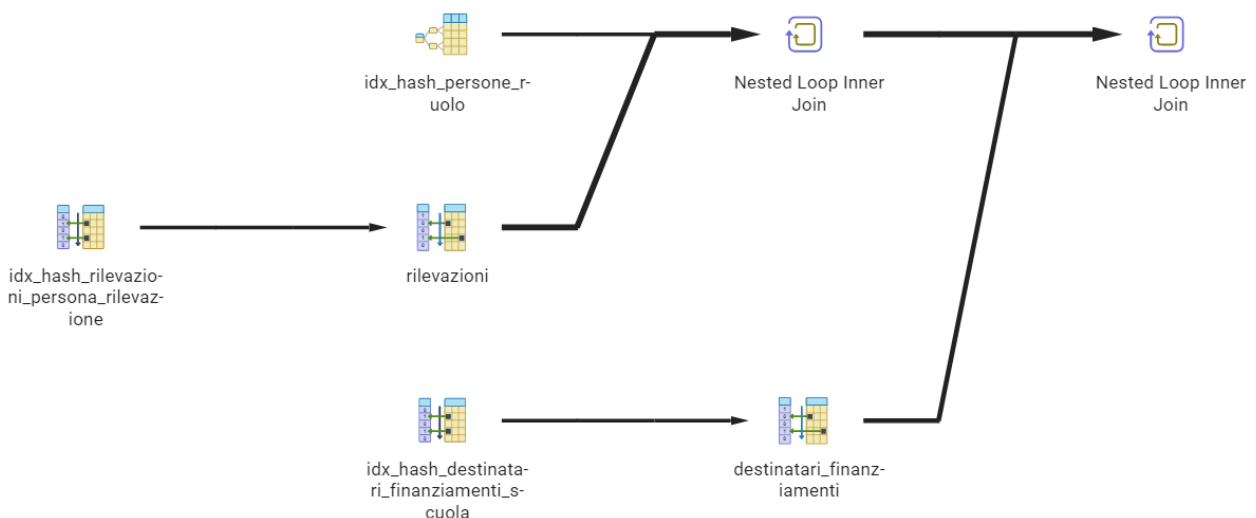
Query 2

Piano prima della creazione dello schema fisico



L'accesso alla relazione di base "persone" permette di estrarre le tuple desiderate (quelle aventi come ruolo "dirigente" e che sono referenti) mediante una ricerca sequenziale con filtro, che vengono poi joinate (mediante hash join) con la relazione "rilevazioni". Viene poi eseguita una index nested loop join, usando come relazione inner il join precedente, e come outer l'indice generato automaticamente da PostgreSQL, relativo alla chiave primaria di "destinatari_finanziamenti".

Piano dopo la creazione dello schema fisico



L'accesso alla relazione di base "persone" permette di estrarre le tuple desiderate (quelle aventi come ruolo "dirigente" e che sono referenti) mediante una ricerca ad indice sull'attributo ruolo, con filtro su referente.

Le tuple vengono poi joinate (index nested loop) con la relazione “rilevazioni” (viene utilizzato per l’accesso a “rilevazioni” una bitmap Index scan). Viene poi eseguita una index nested loop join, usando come relazione inner il join precedente, e come outer l’indice di “destinatari_finanziamenti” (anche qui viene utilizzato per l’accesso a “scuole” una bitmap Index scan)

Confronto tempi

Prima:

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Nested Loop Inner Join (cost=57.04..122.07 rows=8 width=10) (actu...	0.002 ms	0.178 ms	↓ 1	0	8	1
2.	→ Hash Inner Join (cost=56.76..117.71 rows=1 width=16) (actual=... Hash Cond: ((rilevazioni.persona_rilevazione)::text = (persone.mail)::tex	0.003 ms	0.177 ms	↓ 1	0	1	1
3.	→ Seq Scan on rilevazioni as rilevazioni (cost=0..57 rows=150...	0.009 ms	0.009 ms	↑ 1500	1	1500	1
4.	→ Hash (cost=56.75..56.75 rows=1 width=134) (actual=0.165... Buckets: 1024 Batches: 1 Memory Usage: 8 kB	0.165 ms	0.165 ms	↓ 1	0	1	1
5.	→ Seq Scan on persone as persone (cost=0..56.75 rows=... Filter: ((ruolo)::text = 'dirigente')::text) Rows Removed by Filter: 1500	0.165 ms	0.165 ms	↓ 1	0	1	1
6.	→ Index Only Scan using destinatari_finanziamenti_pkey on destin... Index Cond: (scuola = (persone.scuola)::text)	1 ms	1 ms	↓ 1	0	3	0

Dopo:

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	→ Nested Loop Inner Join (cost=8.34..37 rows=8 width=10) (actual=0.0...	0.001 ms	0.006 ms	↓ 1	0	8	1
2.	→ Nested Loop Inner Join (cost=4.04..26.99 rows=1 width=16) (act...	0.005 ms	0.005 ms	↓ 1	0	1	1
3.	→ Index Scan using idx_hash_persone_ruolo on persone as pe... Index Cond: ((ruolo)::text = 'dirigente')::text)	0.005 ms	0.005 ms	↓ 1	0	1	1
4.	→ Bitmap Heap Scan on rilevazioni as rilevazioni (cost=4.04..1... Recheck Cond: ((persona_rilevazione)::text = (persone.mail)::text) Heap Blocks: exact=0	0	0	↓ 1	0	5	0
5.	→ Bitmap Index Scan using idx_hash_rilevazioni_persona... Index Cond: ((persona_rilevazione)::text = (persone.mail)::text)	1 ms	1 ms	↓ 1	0	5	0
6.	→ Bitmap Heap Scan on destinatari_finanziamenti as destinatari_fi... Recheck Cond: ((scuola)::text = (persone.scuola)::text) Heap Blocks: exact=0	0	0	↓ 1	0	3	0
7.	→ Bitmap Index Scan using idx_hash_destinatari_finanziament... Index Cond: ((scuola)::text = (persone.scuola)::text)	1 ms	1 ms	↓ 1	0	3	0

Si può notare un incremento prestazionale di circa 30 volte, rendendo quasi trascurabile il tempo di esecuzione. Si nota anche una riduzione notevole delle tuple intermedie. (Il fatto che siano così poche le tuple intermedie è corretto, in quanto i dati generati da Datanamic non risultano nel risultato della query)

Query 3

Piano prima della creazione dello schema fisico



parametri_ambientali

L'accesso alla relazione di base "parametri_ambientali" permette di estrarre le tuple desiderate mediante una scansione sequenziale su tutti gli attributi.

Piano dopo la creazione dello schema fisico



idx_ord_parametri_ambientali



parametri_ambientali

L'accesso alla relazione di base "parametri_ambientali" permette di estrarre le tuple desiderate mediante una scansione sull'indice per il fattore booleano e una successiva scansione sequenziale sul risultato per tutti gli attributi.

Confronto tempi

Prima:

1.	→ Seq Scan on parametri_ambientali as p (cost=0..56.5 rows=2 width=2...) Filter: ((umidita > '10'::numeric) AND (umidita < '90'::numeric) AND ((ph >= '3'::numeric) OR (temperatura < '10'::numeric))) Rows Removed by Filter: 1006	0.291 ms	0.291 ms	↓ 247	494	2	1
----	---	----------	----------	-------	-----	---	---

Dopo:

1.	→ Bitmap Heap Scan on parametri_ambientali as p (cost=18.08..51.58 rows=2 width=2...) Filter: (((ph >= '3'::numeric) OR (temperatura < '10'::numeric)) AND (date_part('year', (umidita > '10'::numeric)) < '90'::numeric)) Rows Removed by Filter: 87 Recheck Cond: ((umidita > '10'::numeric) AND (umidita < '90'::numeric)) Heap Blocks: exact=8	0.147 ms	0.221 ms	↓ 247	494	2	1
2.	→ Bitmap Index Scan using idx_ord_parametri_ambientali (cost=0..0.00 rows=2 width=0...) Index Cond: ((umidita > '10'::numeric) AND (umidita < '90'::numeric))	0.075 ms	0.075 ms	↓ 1.01	581	580	1

Si può notare un incremento prestazionale ridotto, ma comprensibile in quanto il fattore di selettività del valore booleano in questo caso è molto alto, ne deriva che la successiva scansione sequenziale dovrà scorrere comunque un elevatissimo numero di tuple.

10. Controllo dell'accesso

	Insegnate	Gestore globale del progetto	Referente della scuola	Studente
Scuola	-	select, update, insert	select	-
Finanziamenti	-	select, update, insert	select	-
Persone	-	select, update, insert	select, update, insert	-
Classi	select	select, update, insert	select, update, insert	-
Orti	select	select, update, insert	select, update, insert	select
Parametri_ambientali	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Sensori	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Specie	select	select, update, insert	select	select
Piante	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Gruppi	select, update, insert	select, update, insert	select, update, insert	select
Rilevazioni	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Misure	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Nomi_misure	select	select, update, insert	select	select
Destinatari_finanziamenti	-	select, update, insert	select	-
Monitoraggi	select, update, insert	select, update, insert	select, update, insert	select, update, insert
Orti_condivisi	select	select, update, insert	select, update, insert	select

Gerarchia

Gestore globale del progetto > Referente della scuola > Insegnate > Studente

Giustifiche

Insegnate

Sarebbe meglio poter utilizzare viste che limitino l'accesso da parte dell'insegnate alle sole componenti delle tabelle relative alle proprie classi/scuole.

L'insegnate non può accedere ad informazioni relative ad altre scuole, o altre persone. Invece può accedere ad informazioni relative alle sue classi (qui sarebbe meglio utilizzare una vista), può accedere all'informazioni sugli orti. I parametri ambientali, i sensori e i monitoraggi sono modificabili (anche qui servirebbe una vista per limitare l'accesso e la modifica dei sensori e informazioni ambientali della propria classe). Può accedere ai dati relativi alle specie (in realtà basterebbe una vista che limiti l'accesso alle tre specie di quella scuola). Le piante, i gruppi, le rilevazioni e le misure sono accessibili e modificabili (anche qui servirebbe una vista per impedire l'accesso e la modifica dell'informazioni relative ad altre classi/scuole). Infine, nomi misure sono liberamente accessibili, mentre orti condivisi richiederebbero una vista per limitare l'accesso agli orti condivisi con la propria scuola.

Gestore globale del progetto

Essendo l'amministratore del progetto esso ha completo accesso a tutte le tabelle, anche a tabelle relative a misure, piante, gruppi ecc... perché anche se gestite da insegnanti e studenti devono poter essere modificate da un ente superiore in caso di necessità.

Referente della scuola

Sarebbe meglio poter utilizzare viste che limitino l'accesso da parte del referente alle sole componenti delle tabelle relative alla propria scuola.

Il referente può accedere ad informazioni relative alle scuole (anche se meglio solo alla sua scuola). Può accedere e modificare persone e classi interne alla scuola (sarebbe consigliato l'utilizzo di una vista), per assegnare gli insegnanti alle classi per esempio. Il referente può anche accedere e modificare informazioni sugli orti (sempre della scuola). I parametri ambientali, i sensori e i monitoraggi sono modificabili (anche qui servirebbe una vista per limitare l'accesso e la modifica dei sensori e informazioni ambientali della propria scuola). Può accedere ai dati relativi alle specie. Le piante, i gruppi, le rilevazioni e le misure sono accessibili e modificabili (anche qui servirebbe una vista per impedire l'accesso e la modifica dell'informazioni relative ad altre scuole). Anche i finanziamenti e i destinatari finanziamenti sono consultabili (meglio con vista per limitare l'accesso ai finanziamenti relativi alla propria scuola). Infine, nomi misure sono liberamente accessibili, mentre orti condivisi richiederebbero una vista per limitare l'accesso agli orti condivisi con la propria scuola, ma permettendo la condivisione dei propri orti.

Studente

Sarebbe meglio poter utilizzare viste che limitino l'accesso da parte degli studenti alle sole componenti delle tabelle relative alle proprie classi.

Lo studente non può accedere ad informazioni relative ad altre scuole, altre persone o altre classi; tuttavia, può accedere all'informazioni sugli orti (meglio attraverso una vista per limitare gli orti visibili a quelli della propria scuola). I parametri ambientali, i sensori e i monitoraggi sono modificabili (anche qui servirebbe una vista per limitare l'accesso e la modifica dei sensori e informazioni ambientali della propria classe). Può accedere ai dati relativi alle specie (in realtà basterebbe una vista che limiti l'accesso alle tre specie di quella scuola). Le piante, i gruppi, le rilevazioni e le misure sono accessibili e modificabili (anche qui servirebbe una vista per impedire l'accesso e la modifica dell'informazioni relative ad altre classi/scuole). Infine, nomi

misure sono liberamente accessibili, mentre orti condivisi richiederebbero una vista per limitare l'accesso agli orti condivisi con la propria scuola.