

Orti scolastici

Nuovi requisiti disambiguati

- Si è deciso che ogni sensore associato a una replica venga utilizzato per le eventuali rilevazioni
- Si è deciso che una persona non può avere ruoli multipli, per esempio la stessa persona non può essere docente e dirigente di una scuola (titolare di finanziamento e partecipante al finanziamento non sono dei ruoli)
- In `biomassa_struttura`, si misura solo una tra lunghezza chioma e lunghezza foglia
- In `biomassa_struttura`, si misura solo una tra larghezza chioma e larghezza foglia
- Di base ogni gruppo è identificato come dislocato nell'orto dell'istituto in cui sono state piantate le repliche. Qualora si decidesse di dislocare il gruppo in un altro orto dovrà essere aggiornata la chiave esterna di orto in gruppo
- `id_orto` in replica fa riferimento all'orto dove la replica è stata piantata che può differire dall'`id_orto` del gruppo in cui si trova nel caso in cui quest'ultimo sia stato dislocato in un altro orto

Progetto concettuale

Schema ER

Inserito come ultima pagina per una migliore visibilità.

Entità e attributi

N.B: pk indica primary key, pk__(entità) indica un riferimento esterno alla chiave primaria dell'entità, ak indica una chiave alternativa

- scuola (pk: id)
 - id
 - ciclo
 - tipo
- istituti (pk: codice_meccanografico)
 - codice_meccanografico (char[10])
 - nome
 - via
 - cap
 - comune
 - provincia
- persona (pk: id)
 - id (int autoincrementante)
 - nome
 - cognome

- email
 - ruolo
 - telefono (opzionale)
- finanziamento (pk: pk_istituto)
 - tipo
- classe (pk: id | ak: pk_scuola + nome)
 - id
 - nome
- orto (pk: id | ak: pk_istituti + nome)
 - id
 - nome
 - tipo (false - terra, true - vaso)
 - latitudine
 - longitudine
 - superficie_m2
- gruppo (pk: id)
 - id
 - ok_controllo (bool)
- replica (pk: id)
 - id
 - scopo (true - biomonitoraggio, false - fitobonifica)
 - data_dimora (timestamp)
- sensore (pk: seriale)
 - seriale
 - tipo
- specie (pk: nome_scientifico)
 - nome_scientifico
 - nome_comune
 - esposizione
- rilevazione (pk: id)
 - id
 - data_ins (timestamp)
 - data_ril (timestamp)
- biomassa_struttura (pk: id)
 - id
 - larghezza_chioma
 - lunghezza_chioma
 - peso_fresco_chioma
 - peso_secco_chioma
 - altezza_pianta
 - lunghezza_radice
- alterazione_fioritura_fruttificazione (pk: id)
 - id
 - peso_fresco_radici
 - peso_secco_radici
 - n_fiori

- n_frutti
- altri_danni (pk:id)
 - id
 - n_foglie_danneggiate
 - perc_sup_danneggiata
- parametri_suolo (pk:id)
 - id
 - ph
 - umidita
 - temperatura

Associazioni

- in: scuola (1,N) - persona (0,N), una scuola ha almeno una persona e una persona può essere in più scuole
- riferimento_scuola: scuola (1,N) - persona (0,N), una scuola ha almeno una persona di riferimento e una persona può essere il riferimento per 0 o più scuole
- riferimento_istituto: scuola (1,N) - persona (0,N), un istituto ha almeno una persona di riferimento e una persona può essere il riferimento per 0 o più istituti
- formata_da: scuola (1,N) - classe (1,1), una scuola è formata da una o più classi e una classe appartiene a una determinata scuola
- titolare_di: istituti (0,1) - finanziamento (1,1), un istituto può essere titolare o meno di un finanziamento e un finanziamento è specifico per un istituto
- appartiene: istituti (1,N) - scuola (1,1), un istituto può avere da 1 a N scuole e una scuola appartiene a un solo istituto
- partecipa: persona (0,N) - finanziamento (0,N), una persona può partecipare a 0 o più finanziamenti e ad un finanziamento possono partecipare delle persone
- capo: persona (0,N) - finanziamento (1,1), una persona può essere a capo di più finanziamenti e un finanziamento ha a capo una sola persona
- docente_riferimento: persona (0,N) - classe (1,1), una persona può essere docente di riferimento di 0 o più classi, una classe ha uno e un solo docente di riferimento
- possiede: istituti (1,N) - orto (1,1), un istituto ha almeno un orto e un orto fa riferimento ad un unico istituto
- dotato_di: orto (1,N) - sensore (1,1), un orto ha almeno un sensore e un sensore è specifico di un orto
- piantata: orto (1,N) - replica (1,1), un orto ha 1 o più repliche di una pianta e una replica è piantata in un determinato orto
- di: replica (1,1) - specie (0,N), una replica è di una determinata specie e ad una specie possono appartenere 0 o più repliche
- messa_a_dimora: replica (1,1) - classe (0,N), una replica può essere messa a dimora da una classe e una classe può mettere a dimora 0 o più repliche

- `soggetta_a`: replica (0,N) - rilevazione (1,1), una replica è soggetta a zero o più rilevazioni e una rilevazione fa riferimento a una determinata replica
- `associata`: replica (1,N) - sensore (1,N), un sensore viene utilizzato per una o più repliche e una replica può essere associata a 1 o più sensori
- `responsabile_inserimento_classe`: classe (0,N) - rilevazione (0,1), una classe può essere responsabile (di inserimento) di 0 o più rilevazioni, e una rilevazione ha come responsabile di inserimento (di classe) al più una classe
- `responsabile_inserimento_persona`: persona (0,N) - rilevazione (0,1), una persona può essere responsabile (di inserimento) di 0 o più rilevazioni, e una rilevazione ha come responsabile di inserimento (di persona) al più una persona
- `responsabile_rilevazione_classe`: classe (0,N) - rilevazione (0,1), una classe può essere responsabile (di rilevazione) di 0 o più rilevazioni, e una rilevazione ha come responsabile di rilevazione (di classe) al più una classe
- `responsabile_rilevazione_persona`: persona (0,N) - rilevazione (0,1), una persona può essere responsabile (di rilevazione) di 0 o più rilevazioni, e una rilevazione ha come responsabile di rilevazione (di persona) al più una persona
- `fa_parte_di`: replica (1,1) - gruppo (0,20), una replica fa parte di un solo gruppo e a un gruppo possono fare parte massimo 20 repliche
- `controlla`: gruppo (0,1) - gruppo (0,1), un gruppo può controllare un altro gruppo e un gruppo può essere controllato da un altro gruppo
- `dislocato`: gruppo (1,1) - orto (1,N), un gruppo può essere dislocato in un solo orto e un orto può ospitare 1 o più gruppi
- `ha_bs`: `biomassa_struttura` (1,1) - rilevazione (0,1), i dati di `biomassa_struttura` sono relativi ad una e una sola rilevazione e una rilevazione può contenere dati di `biomassa_struttura`
- `ha_aff`: `alterazione_fioritura_fruttificazione` (1,1) - rilevazione (0,1), i dati di `alterazione_fioritura_fruttificazione` sono relativi ad una e una sola rilevazione e una rilevazione può contenere dati di `alterazione_fioritura_fruttificazione`
- `ha_ad`: `altri_danni` (1,1) - rilevazione (0,1), i dati di `altri_danni` sono relativi ad una e una sola rilevazione e una rilevazione può contenere dati di `altri_danni`
- `ha_ps`: `parametri_suolo` (1,1) - rilevazione (0,1), i dati di `parametri_suolo` sono relativi ad una e una sola rilevazione e una rilevazione può contenere dati di `parametri_suolo`

Vincoli

- Ogni rilevazione ha un solo responsabile di inserimento (o una persona o una classe)
- Ogni rilevazione ha un solo responsabile di rilevazione (o una persona o una classe)
- Il gruppo può controllare un altro gruppo solo se è un gruppo di controllo (attributo `controllo` è `true`)

- In caso di biomonitoraggio, il numero di repliche di un gruppo di controllo deve essere lo stesso del numero di repliche per monitorare lo stress ambientale
- Una classe o una persona possono fare rilevazioni solo su repliche il cui gruppo è dislocato in orti del proprio istituto
- Un gruppo può controllare solo se non è controllato
- Quando viene tolta o aggiunta una replica in un gruppo controllato o controllato il loro collegamento viene rimosso poichè il numero di repliche non sarebbe più lo stesso per entrambi
- Una classe può mettere a dimora delle repliche solo in orti dell'istituto in cui risiede scuola di cui ne fa parte
- Una persona può essere docente di una classe solo se appartiene alla scuola in cui si trova la classe
- Il sensore associato alla replica deve fare parte dell'orto in cui è dislocato il gruppo che contiene tale replica
- Quando una replica viene piantata deve appartenere ad un gruppo dislocato nello stesso orto dove viene piantata

Riepilogo in forma tabellare

Entità

nome	attributi	identificatori
scuola	id, ciclo, tipo	id
istituti	codice_meccanografico (char[10]), nome, via, cap, comune ,provincia	codice_meccanografico
persona	id (int autoincrementante) , nome, cognome, email, ruolo, telefono (opzionale)	id
finanziamenti	id, tipo	id
classe	id, nome	id
orto	id, nome, tipo (bool), latitudine, longitudine, superficie	id
gruppo	id, controllo (bool)	id
sensore	seriale, tipo	seriale
replica	id, scopo (bool), data_dimora (timestamp)	id
specie	nome_scientifico, nome_comune, esposizione	nome_scientifico
rilevazione	id, data_ins (timestamp), data_ril (timestamp)	id
biomassa_studio	id, larghezza_chioma_foglia, lunghezza_chioma_foglia, peso_fresco_chioma_foglia, peso_secco_chioma_foglia, altezza_pianta, lunghezza_radice	id
alterazione_fioritura	id, peso_fresco_radice, peso_secco_radice, n_fiori, n_frutti	id
altri_danni	id, n_foglie_danneggiate, perc_sup_danneggiata	id
parametri_suolo	id, ph, umidità, temperatura	id

Associazioni

nome	entità_collegate
in	scuola (1:N) - persona (0:N)
riferimento_scuola	scuola (1:N) - persona (0:N)
riferimento_istituto	istituto (1:N) - persona (0:N)
formata_da	scuola (1:N) - classe (1,1)
titolare_di	istituti (0,1) - finanziamento (1,1)
appartiene	istituti (1,N) - scuola (1,1)
partecipa	persona (0,N) - finanziamento (0,N)
capo	persona (0,N) - finanziamento (1,1)
docente_riferimento	persona (0,N) - classe (1,1)
possiede	istituti (1,N) - orto (1,1)
dotato_di	orto (1,N) - sensore (1,1)
piantata	orto (1,N) - replica (1,1)
di	replica (1,1) - specie (0,N)
messa_a_dimora	rreplica (1,1) - classe (0,N)
soggetta_a	replica (0,N) - rilevazione (1,1)
associata	replica (1,N) - sensore (1,N)
responsabile_inserimento_classe	classe (0,N) - rilevazione (0,1)
responsabile_inserimento_persona	persona (0,N) - rilevazione (0,1)
responsabile_rilevazione_classe	classe (0,N) - rilevazione (0,1)
responsabile_rilevazione_persona	persona (0,N) - rilevazione (0,1)
fa_parte_di	replica (1,1) - gruppo (0,20)
controlla	gruppo (0,1) - gruppo (0,1)
dislocato	gruppo (1,1) - orto (1,N)
ha_bs	biomassa_struttura (1,1) - rilevazione (0,1)
ha_aff	alterazione_fioritura_fruttificazione (1,1) - rilevazione (0,1)
ha_ad	altri_danni (1,1) - rilevazione (0,1)
ha_ps	parametri_suolo (1,1) - rilevazione (0,1)

Script per il popolamento del DB

Per popolare il database abbiamo scelto di realizzare uno script python. Il codice è diviso in vari file:

- connection.py : si occupa di stabilire la connessione al db
- create_schema.py : legge il file create_schema.sql e lo esegue creando le tabelle (se si vogliono ricreare le tabelle passare -reset)
- models.py: effettua un automapping delle tabelle create in classi python tramite l'ORM di sqlalchemy
- populate.py: effettua una popolazione base del db

- `extra_populate.py`: continua la popolazione del db fino a quando ogni tabella non ha almeno 8000 bytes (dimensione pagina) * 10 -> requisito per la parte fisica
- `custom_data`: contiene delle liste di dati che abbiamo generato per mettere dei valori sensati in alcuni attributi

I dati sono generati randomicamente cercando di rappresentare il più logicamente possibile l'attributo. Questo è possibile tramite Faker che mette a disposizione dei metodi per generare nomi, cognomi, società, email, etc. . . Per altri dati come i nomi dei finanziamenti, le piante, le scuole e i sensori abbiamo chiesto a chatgpt di generarceli, aiutandoci con un piccolo script in javascript per continuare la generazione della risposta che si può trovar nella cartella js. Infine per i comuni italiani e le provincie associate abbiamo realizzato una funzione in utils che crea automaticamente un file python con la lista di tuple (comune, provincia) di tutti i comuni italiani parsati dal sito del governo.

Per la gestione delle dipendenze di python abbiamo utilizzato poetry. Poetry crea un virtual environment python per installare le dipendenze del progetto (isolate rispetto a quelle del sistema).

Di seguito sono elencati i comandi in ordine da eseguire per testare il progetto:

1. installare poetry (seguire la documentazione ufficiale oppure utilizzare `pip install poetry`)
2. attivare la shell nell'ambiente virtuale: `poetry shell`
3. creare il file `.env` nella project root e inserire il proprio DB_URL: per esempio `DB_URL=postgresql://postgres:postgres@localhost:5432/orti_scolastici`
4. creare le tabelle: `python3 src/py/create_tables.py`
5. popolare il database:
 1. `python3 src/py/populate.py`
 2. `python3 src/py/extra_populate.py`

In alternativa si può: - utilizzare un dump del nostro db già popolato realizzato con `pg_dump -U postgres -d orti_scolastici -W -F p --clean --inserts -f sql/inserts.sql` ripristinandolo con `psql -U postgres -d orti_scolastici -W -f sql/inserts.sql` - collegarsi ad un'istanza di postgres online che abbiamo già popolato al seguente DB_URL: `postgresql://gyxbhqpb:BoRkDUhN25VDy9AMggrb0kBmAOhICp9A@trumpet.db.elephantsql.com/gyxbhqpb`

Normalizzazione

Tutte le relazioni sono in BCFN.

