



**UNIVERSITÀ
DEGLI STUDI
DI UDINE**



DATABASE PER RILIEVI NIVOLOGICI

Modelli 2 e 3 AINEVA

[Luca Iacoletti - 111895](#)

Sistemi di Elaborazione delle informazioni

Corso di Laurea Magistrale in Scienze e
Tecnologie per l'Ambiente e il
Territorio

A.A. 2015-2016



**Associazione
Interregionale
di Coordinamento
e Documentazione
per i Problemi Inerenti
alla Neve e alle Valanghe**

Sommario

1. ANALISI DEI REQUISITI	3
a. Descrizione del dominio applicativo	3
b. Glossario dei termini	4
c. Esempi di operazioni	5
2. PROGETTAZIONE CONCETTUALE	7
a. Scelte progettuali	11
3. PROGETTAZIONE LOGICA	12
b. Schema di base di dati	12
c. Vincoli di integrità	16
4. PROGETTAZIONE FISICA	17
a. Domini	17
b. Tabelle	18
c. Funzioni	22
d. Trigger	26
e. Verifica di vincoli d'integrità complessi	27
f. Viste	28
5. POPOLAZIONE DELLA BASE DI DATI	30
a. Dati grezzi	30
b. Raffinamento della base di dati	30
c. Verifica di vincoli d'integrità	33
6. INTERROGAZIONI SQL	36
Riferimenti	44

1. ANALISI DEI REQUISITI

È una fase preliminare che indaga e definisce quali proprietà e funzionalità dovrà avere il sistema informativo: il progettista interagisce con gli utenti del sistema per produrre una descrizione informale, ma completa, dei dati coinvolti (descrizione del dominio applicativo) e delle operazioni che verranno effettuate su di essi. [1]

a. Descrizione del dominio applicativo

Si vuole realizzare una base di dati per gestire i rilievi nivologici (Modelli 2 e 3 AINEVA¹) dell'ufficio valanghe della Regione FVG, interfacciandoli con lo *shapefile* dei Comuni della Regione FVG.

Di ogni rilievo nivologico, identificato da un numero progressivo, si devono memorizzare i seguenti dati: Codice Stazione, Data, Ora, Località, Altitudine, Esposizione, Pendenza, Coordinate, Precipitazioni (WW), Nubi (N), Vento (VQ), Il rilevatore, Temperatura aria (Ta), Altezza manto nevoso (HS), Rugosità superficiale (SF), Note.

Il Modello 2 memorizza i dati relativi alla prova penetrometrica con sonda svizzera ("test della sonda"²) per calcolare l'indice di durezza (R) dei vari strati del manto nevoso.

Il Modello 3 memorizza i dati relativi all'analisi degli strati, alla temperatura della neve ed alla densità degli strati.³

Codice Stazione, Esposizione, Pendenza, Data, Ora, Località, WW, N, VQ, Il rilevatore, Ta e HS devono essere sempre noti.

I rilievi nivologici possono avvenire sempre nelle stesse stazioni (rilievi fissi in campi di rilievo fissi) o in stazioni scelte di volta in volta (rilievi itineranti in stazioni itineranti).

Località, Altitudine, Esposizione, Pendenza e Coordinate dei campi di rilievo fissi non variano nel tempo.

Nei rilievi itineranti, a volte, la prova penetrometrica col "test della sonda" (Modello 2) non viene eseguita, mentre deve essere sempre eseguita nei rilievi fissi.

¹Vedi Figura 1: Modelli 2 e 3 AINEVA

² I dati memorizzati relativi al "test della sonda" del Modello 2 sono: peso massa battente (p) [N], numero di colpi relativo alla serie di battute (n) [adimensionale], altezza di caduta della massa battente (h) [cm], affondamento dei colpi (Δ , differenza tra penetrazione x_i e x_{i-1}) [cm], peso dei tubi sonda (q) [N], penetrazione della sonda (x) [cm]. L'indice di durezza (R) [N] degli strati del manto nevoso viene calcolato indirettamente tramite l'equazione

$$R = \left(\frac{pnh}{\Delta} \right) + p + q$$

³ I dati memorizzati relativi al Modello 3 sono i seguenti.

Analisi degli strati: profondità H (divisa per strati in limite superiore e inferiore), contenuto in acqua liquida (Θ_s), forma dei grani (FF: si possono rilevare due forme di grani, FF1 e FF2), dimensione dei grani (E: si possono rilevare due dimensioni prevalenti, E1 ed E2), "test della mano" (R), eventuale commento.

Temperatura della neve alle diverse profondità: profondità (H), temperatura (°C).

Densità dello strato: profondità di prelievo del campione (H) [cm], campione verticale (Cv)/orizzontale (Co), altezza di riempimento del tubo carotatore (h) [cm], peso neve [N], densità (ρ_s) [kg m⁻³].

Un rilievo può essere effettuato da uno o più rilevatori, di cui va memorizzato almeno il cognome; talvolta non vengono memorizzati dati anagrafici, ma solo il nome della stazione forestale di appartenenza dei rilevatori.

Non si può eseguire più di un rilievo al giorno nella stessa stazione.

Quando i rilievi vengono effettuati dalle Guide Alpine, si tratta esclusivamente di rilievi itineranti (codice stazione ITGA).

I rilievi itineranti non possono avvenire nei campi di rilievo fissi, ma più di un rilievo itinerante può essere svolto nella medesima località di un altro rilievo itinerante.

b. Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Analisi degli strati	Valutazione degli strati del manto nevoso, rilevando tipo di grani e loro dimensione.		Modello 3
Campo di rilievo fisso	Stazione predisposta, a volte delimitata, dedicata appositamente ai rilievi nivologici sempre nello stesso luogo.	Stazione fissa	Stazione
Modello 2	Modulo per la raccolta dei dati relativi alla prova penetrometrica con sonda svizzera, che indaga la durezza degli strati del manto nevoso.		Prova penetrometrica, Rilevatore
Modello 3	Modulo per la raccolta delle caratteristiche stratigrafiche del manto nevoso.		Analisi degli strati, Rilevatore
Prova penetrometrica	Prova di penetrazione del manto nevoso per verificarne la durezza. Può essere eseguita con sonda svizzera o col "test della mano"		Modello 2, Modello 3, Sonda svizzera, Test della mano
Rilevatore	Persona/e che esegue/eseguono il rilievo nivologico.	Tecnico, osservatore nivologico	Modello 2, Modello 3
Rilievo fisso	Rilievo nivologico effettuato in un campo di rilievo fisso.	Rilievo in stazione fissa, rilievo in campo di rilievo fisso	Campo di rilievo fisso

Rilievo itinerante	Rilievo nivologico effettuato in luoghi diversi dai campi di rilievo fissi, spesso in zone non facilmente raggiungibili	Rilievo in stazione itinerante	Stazione itinerante
Rilievo nivologico	Attività di indagine del manto nevoso condotta in ambiente innevato da uno o più rilevatori. Possono essere fissi o itineranti.	Rilievo	
<i>Shapefile</i>	Insieme di file in formato vettoriale (proprietario di ESRI) per sistemi informativi geografici.		
Sonda svizzera	Particolare sonda pesante dotata all'estremità superiore di una massa battente.	Sonda battage	Prova penetrometrica
Stazione	Luogo del rilievo. È contraddistinta da un proprio codice.		
Stazione fissa		Campo di rilievo fisso	Rilievo fisso
Stazione itinerante	Luogo di un rilievo itinerante		Rilievo itinerante
Test della mano	Test speditivo che valuta la resistenza degli strati del manto nevoso ad un pugno chiuso, 4 dita stese, 1 dito steso, matita, lama di coltello		Prova penetrometrica
Test della sonda	Prova penetrometrica con sonda svizzera, effettuata misurando l'affondamento della sonda in seguito ai colpi dati con una massa battente di 1 kg.		Prova penetrometrica, Sonda svizzera

c. Esempi di operazioni

1. Inserisci un nuovo rilievo con tutti i dati misurati (operazione da effettuare in media 3-4 volte a settimana).
2. Calcola l'indice di durezza R del "test della sonda" (ogniqualevolta si inserisce un nuovo rilievo).
3. Assegna ad un rilievo fisso i dati già noti del campo di rilievo fisso relativo (ogniqualevolta si inserisce un nuovo rilievo).
4. Visualizza in ambiente GIS la località di ogni rilievo di cui sono note le coordinate (4 volte al mese).
5. Trova tutte le informazioni relative all'ultimo rilievo effettuato (10 volte a settimana).
6. Calcola una statistica sulle località in cui avvengono più rilievi (1 volta al mese).
7. Trova le informazioni disponibili su tutte le stazioni di rilievo (1 volta al mese).

Figura 1: Modelli 2 e 3 AINEVA

2. PROGETTAZIONE CONCETTUALE

Rappresenta in termini formali le specifiche informali ricavate dall'analisi dei requisiti, indipendentemente dai criteri di rappresentazione propri del DBMS. Usando un modello concettuale dei dati che rappresenta il contenuto informativo, si produce un diagramma concettuale [1]. Il modello concettuale qui usato sarà il modello Entità-Relazione.

7

Analizzando i requisiti è stato prodotto un modello concettuale dei dati (Figura 2) adottando il formalismo proposto dal software *yEd* [2], con aggiunta dei costrutti propri di *ChronoGeoGraph* [3] [4] che consente un'estensione spaziale del modello Entità-Relazione.

Nel diagramma di Figura 2 la partecipazione opzionale di un attributo è indicata con cerchietto O e barra verticale | sull'arco di congiunzione con l'entità, la restrizione di valor non nullo è rappresentata con archi semplici, mentre la restrizione di unicità è indicata dal nome dell'attributo in *corsivo*. Gli attributi derivati hanno contorno tratteggiato, mentre gli attributi identificatori (o semi-identificatori, nel caso di entità deboli) sono scritti in **grassetto sottolineato**. Sono stati utilizzati sostantivi al singolare sia per le entità che per le relazioni (per non conferire direzionalità a queste ultime).

Si crea un'entità centrale RILIEVO, di cui **Numero** è l'attributo identificatore. I dati misurati vengono memorizzati da attributi di RILIEVO oppure mediante relazioni cui RILIEVO partecipa.

SONDA, ANALISI STRATI, DENSITÀ e T_NEVE sono *entità deboli* dipendenti da RILIEVO. La relazione Modello 2 tra SONDA e RILIEVO ha cardinalità (0,N) sull'arco prossimo a RILIEVO, in quanto se il rilievo è itinerante l'esecuzione del "test della sonda" non è obbligatoria.

Quantunque non siano note le coordinate di tutte le stazioni, si è scelto di modellare STAZIONE come entità spaziale di geometria puntuale; per le stazioni sprovviste di coordinate, sarà concesso il valor nullo della geometria.

COMUNE è un'entità spaziale di geometria poligonale ed è in relazione topologica con Stazione di rilievo.

Rispetto alle richieste di cui al paragrafo 1, vengono effettuate le seguenti modifiche:

- DENSITÀ non ha gli attributi Campione e Altezza riempimento, in quanto non strettamente necessari al calcolo della densità e raramente memorizzati (numericamente, il valore della densità coincide col doppio del valore del peso del campione di neve prelevato).
- Essendo STAZIONE un'entità spaziale, non è necessario creare un attributo "coordinate". L'informazione spaziale sarà contenuta nella geometria.
- Vengono aggiunti degli identificatori (**Id** a STAZIONE e **Id** a RILEVATORE) per disporre di attributi facilmente convertibile in chiavi primarie in fase di progettazione logica.
- Si rilassano le cardinalità minime della relazione tra RILIEVO e RILEVATORE: in caso di assenza dei nomi dei rilevatori, si farà riferimento all'attributo Competenza di STAZIONE.

In Tabella 1 sono riportate tutte le entità e le relazioni presenti nel diagramma; in *Tabella 2* sono riportati vincoli e derivazioni non esplicitabili nello schema concettuale e pertanto espressi in linguaggio naturale.

Figura 2: Diagramma concettuale

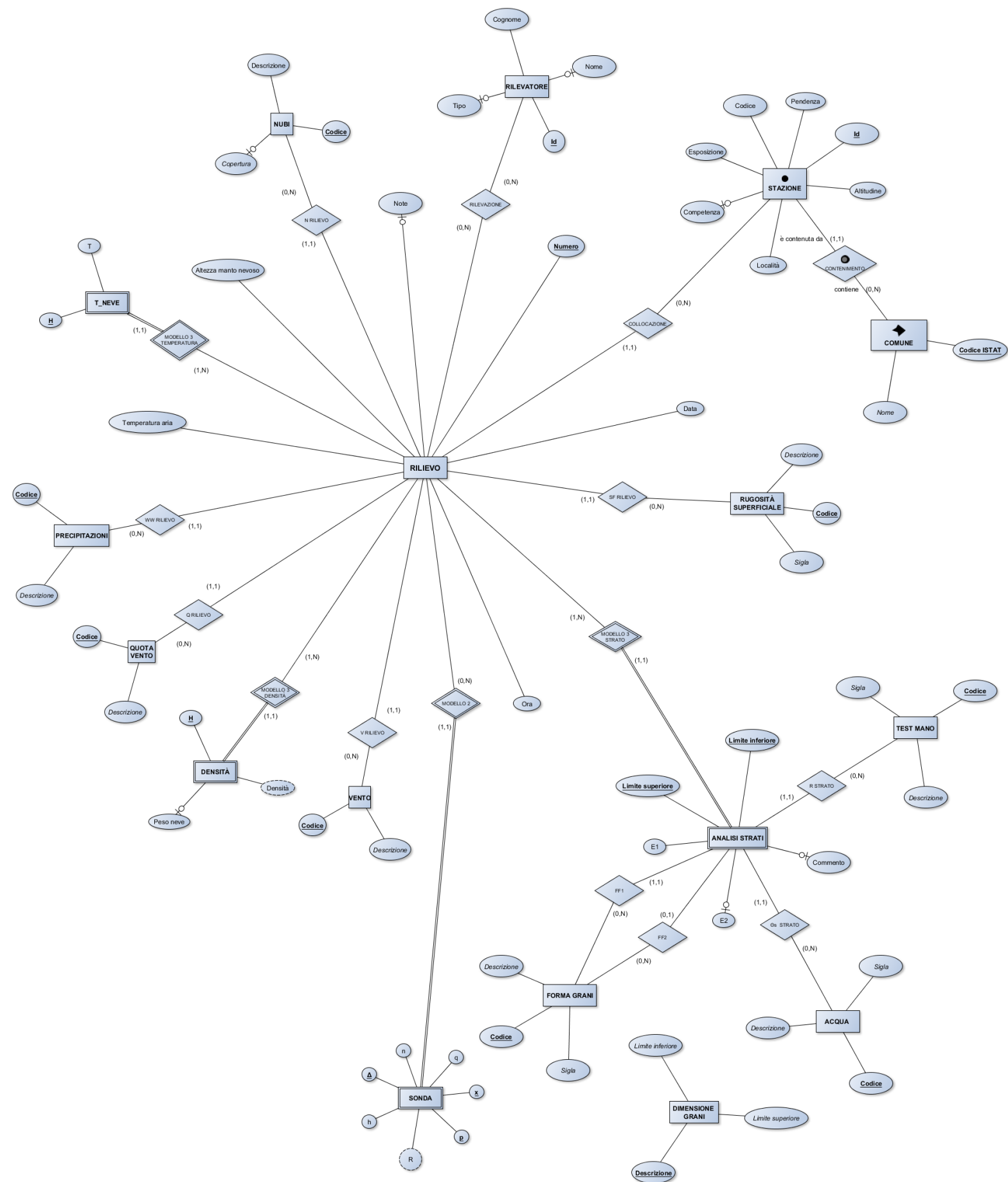


Tabella 1 : Dizionario dei dati per lo schema concettuale

Entità	Descrizione	Attributi	Identificatori
Acqua	Didascalia sui valori di Θ_s nell'analisi degli strati	Codice, Sigla, Descrizione	Codice
Analisi strati	Analisi cristallografica di uno strato del manto nevoso	Limite superiore, Limite inferiore, E1, E2, R, Commento	Limite superiore, Limite inferiore
Comune		Codice ISTAT, Nome	Codice ISTAT
Densità	Densità di uno strato del manto nevoso	H, Peso neve, Densità	H
Dimensione grani	Didascalia sui valori di E nell'analisi degli strati	Descrizione, Limite superiore, Limite inferiore	Descrizione
Forma grani	Didascalia sui valori di FF nell'analisi degli strati	Codice, Sigla, Descrizione	Codice
Nubi	Nuvolosità al momento del rilievo	Codice, Copertura, Descrizione	Codice
Precipitazioni	Condizioni meteo al momento del rilievo	Codice, Descrizione	Codice
Quota vento	Quota ed esposizione del vento	Codice, Descrizione	Codice
Rilevatore	Colui che effettua il rilievo	Id, Nome, Cognome, Tipo	Id
Rilievo	Singolo rilievo nivologico	Numero, Data, Ora, Temperatura aria, Altezza manto nevoso, Note	Numero
Sonda	Test della sonda	q, p, n, h, x, Δ , R	p, x, Δ
Stazione	Luogo di un rilievo	Id, Codice, Località, Altitudine, Esposizione, Pendenza, Competenza	Id
T_neve	Temperatura della neve alle diverse profondità	H, T	H
Test mano	Didascalia sui valori di R nell'analisi degli strati	Codice, Sigla, Descrizione	Codice
Vento	Ventosità	Codice, Descrizione	Codice

Relazione	Descrizione	Entità coinvolte	Note
Collocazione	Associa un rilievo al luogo dove è stato eseguito	Rilievo (1,1), Stazione (0,N)	
Contenimento	Associa una stazione di rilievo al comune su cui insiste	Stazione (1,1), Comune (0,N)	Relazione topologica
FF1	Associa una forma prevalente di grani ad uno strato	Analisi strati (1,1), Forma grani (0,N)	
FF2	Associa una forma secondaria di cristalli ad uno strato	Analisi strati (0,1), Forma grani (0,N)	
Modello 2	Associa un test della sonda al proprio rilievo	Rilievo (0,N), Sonda (1,1)	
Modello 3 densità	Associa le densità degli strati del Modello 3 al proprio rilievo	Rilievo (1,N), Densità (1,1)	
Modello 3 strato	Associa l'analisi stratigrafica del Modello 3 al proprio rilievo	Rilievo (1,N), Analisi Strati (1,1)	
Modello 3 temperatura	Associa la temperatura degli strati del Modello 3 al proprio rilievo	Rilievo (1,N), T_neve (1,1)	
N Rilievo	Associa un rilievo alla copertura nuvolosa	Rilievo (1,1), Nubi (0,N)	
Q Rilievo	Associa un rilievo alla quota ed esposizione del vento	Rilievo (1,1), Quota vento (0,N)	
R strato	Associa un valore di durezza al test della mano ad uno strato	Analisi strati (1,1), Test mano (0,N)	
Rilevazione	Associa i nomi dei rilevatori ai rilievi che hanno eseguito	Rilievo (0,N), Rilevatore (0,N)	
V Rilievo	Associa un rilievo alla ventosità	Rilievo (1,1), Vento (0,N)	
WW Rilievo	Associa un rilievo alle	Rilievo (1,1), WW (0,N)	

	condizioni meteo		
O ₃ strato	Associa un valore di contenuto di acqua liquida ad uno strato	Analisi strati (1,1), Acqua (0,N)	

Tabella 2: “Regole aziendali” per lo schema concettuale

Regole di vincolo
RV1) I valori di Sonda devono essere sempre memorizzati se il rilievo avviene in una stazione fissa.
RV2) Un rilevatore che è Guida alpina deve effettuare solo rilievi itineranti.
RV3) Rilievi nella stessa stazione devono avvenire in date diverse.
RV4) I rilievi itineranti devono avvenire in località diverse da quelle dei campi di rilievo fissi e viceversa.
RV5) Le stazioni fisse devono avere codici tutti diversi
Regole di derivazione
RD1) L'indice di durezza R del test della sonda si ottiene dai parametri misurati tramite un'equazione ⁴ .
RD2) Il valore numerico della densità della neve si ottiene moltiplicando il peso della neve per due.

a. Scelte progettuali

Si è scelto di non creare due entità distinte per le stazioni fisse ed itineranti per non duplicare gli stessi attributi su due entità diverse. L'unica entità Stazione ha un Id progressivo come attributo identificatore; nella traduzione nel modello relazionale, l'Id di Stazione sarà chiave esterna di Rilievo. Creando due entità distinte per stazioni fisse ed itineranti si perderebbe in efficienza; la relazione tra Rilievo e Stazione itinerante sarebbe del tipo uno a molti con partecipazione opzionale per entrambe le entità e, senza la creazione di un nuovo Id artificiale, nel modello relazionale la chiave primaria di Stazione itinerante sarebbe composta da tutti i suoi attributi: questo vorrebbe dire che tutti gli attributi di Stazione itinerante dovrebbero essere chiavi esterne in Rilievo (con appesantimento di quest'ultimo) oppure dovrebbero tutti figurare in uno schema di relazione creato allo scopo.

In ogni caso, il carattere delle stazioni itineranti fa sì che non ci siano due tuple identiche: a parità degli altri attributi, infatti, ogni tupla di stazione itinerante varia per almeno un altro attributo (ad esempio la pendenza o le coordinate). Per questo motivo è sufficiente adottare un Id di nuova creazione e creare un'unica entità Stazione.

Si è scelto di adottare l'entità Rilevatore per i soli nomi di persone fisiche. Molto spesso infatti cognome e nome dei rilevatori non vengono memorizzati, ed al loro posto viene riportato solo il nome della stazione forestale di appartenenza dei rilevatori (es. CFR Pontebba). Ma questa è un'informazione ridondante che può essere ricavata dall'attributo *competenza* di Stazione.

⁴ Si rimanda al paragrafo 1.a

3. PROGETTAZIONE LOGICA

Traduce lo schema concettuale ottenuto nella fase precedente in uno schema logico, in termini del modello di rappresentazione adottato dal DMBS (in questo caso, nel modello relazionale). [1]

Vengono effettuate alcune abbreviazioni e semplificazioni nei termini, facilmente intuibili, rispetto a quelli adottati nello schema concettuale.

Di seguito si riporta lo schema della base di dati, composto dagli schemi di relazione completi di chiavi esterne e restrizioni d'unicità e valor non nullo quando necessarie. Accanto allo schema di relazione si riporta la semantica associata.

12

b. Schema di base di dati

ANALISISTRATI (Rilievo, Limite superiore, Limite inferiore, Θ_s , FF1, FF2, E1, E2, R, Commento)

CE: Rilievo \rightarrow RILIEVO (Numero)

$\Theta_s \rightarrow$ ACQUA (Codice)

R \rightarrow TESTMANO (Codice)

FF1 \rightarrow DIMENSIONEGRANI (Codice)

FF2 \rightarrow DIMENSIONEGRANI (Codice)

VNN: $\{\Theta_s, FF1, R\}$

Nel rilievo *Rilievo*, lo strato di manto nevoso compreso tra *Limite superiore* cm e *Limite inferiore* cm ha un contenuto di acqua di codice Θ_s , una forma di grani prevalente *FF1* di diametro *E1* mm (e un'eventuale forma di grani minoritaria *FF2* di *E2* mm); lo strato ha resistenza *R* al test della mano ed è stato aggiunto il commento "*Commento*" come informazione aggiuntiva.

ACQUA (Codice, Sigla, Descrizione)

UNI: {Sigla, Descrizione}

VNN: {Sigla, Descrizione}

La neve di uno strato del manto nevoso è *Descrizione* in quanto a contenuto di acqua liquida, ed è caratterizzata dalla sigla *Sigla* e dal codice *Codice*.

COMUNE (Codice ISTAT, Nome, Geometria)

UNI: {Nome, Geometria}

VNN: {Nome, Geometria}

Il comune di codice *Codice ISTAT* si chiama *Nome* ed ha geometria *Geometria*.

DENSITÀ (Rilievo, H, Peso, Densità)

CE: Rilievo → RILIEVO (Numero)

VNN: {Densità}

Nel rilievo *Rilievo*, lo strato di manto nevoso a profondità *H* pesa *Peso* N ed ha densità *Densità* kg m⁻³.

13

DIMENSIONEGRANI (Descrizione, Min, Max)

UNI: {Min, Max}

VNN: {Min, Max}

La neve di uno strato del manto nevoso è *Descrizione* in quanto a dimensione dei grani, ed il loro diametro è compreso tra *Min* mm e *Max* mm.

FORMAGRANI (Codice, Sigla, Descrizione)

UNI: {Sigla, Descrizione}

VNN: {Sigla, Descrizione}

La neve di uno strato del manto nevoso è *Descrizione* in quanto a forma dei grani, cui corrisponde la sigla *Sigla* ed il codice *Codice*.

NUBI (Codice, Copertura, Descrizione)

UNI: {Copertura}

VNN: {Descrizione}

La nuvolosità durante un rilievo è *Descrizione*, cui corrisponde una copertura del cielo (in termini di frazione di cielo visibile) di *Copertura* ed un codice *Codice*.

PRECIPITAZIONI (Codice, Descrizione)

UNI: {Descrizione}

VNN: {Descrizione}

Durante un rilievo sono stati osservati *Descrizione* come fenomeni meteorologici cui corrisponde il codice *Codice*.

QUOTAVENTO (Codice, Descrizione)

UNI: {Descrizione}

VNN: {Descrizione}

Durante un rilievo è stata rilevata un'attività eolica *Descrizione* per quanto riguarda la quota, cui corrisponde il codice *Codice*.

RILEVATORE (Id, Cognome, Nome, Tipo)

VNN: {Cognome}

Il rilevatore identificato col numero *Id* ha cognome *Cognome* (e nome *Nome*, se noto) ed è *Tipo* come figura professionale.

RILEVAZIONE (Rilievo, Rilevatore)

CE: Rilievo → RILIEVO (Numero)

Rilevatore → RILEVATORE (Id)

Il rilievo nivologico *Rilievo* è stato eseguito dal rilevatore *Rilevatore*.

RILIEVO (Numero, Stazione, Data, Ora, Precipitazioni, Nubi, Vento, Quota vento, Temperatura aria, Altezza manto nevoso, Rugosità superficiale, Note)

CE: Stazione → STAZIONE (Id)

Precipitazioni → PRECIPITAZIONI (Codice)

Nubi → NUBI (Codice)

Vento → VENTO (Codice)

Quota vento → QUOTAVENTO (Codice)

Rugosità superficiale → RUGOSITÀSUPERFICIALE (Codice)

VNN: {Stazione, Data, Ora, Precipitazioni, Nubi, Temperatura aria, Altezza manto nevoso, Rugosità superficiale}

Un rilievo nivologico di numero *Numero* è stato eseguito nella stazione *Stazione* nel giorno *Data* alle ore *Ora*. Al momento del rilievo le condizioni meteo in codice erano *Precipitazioni*, *Nubi*, *Vento*, *Quota vento*. È stata misurata una temperatura dell'aria di *Temperatura aria* °C, il manto nevoso misurava *Altezza manto nevoso* cm ed è stato aggiunto il commento *Note*.

RUGOSITÀ SUPERFICIALE (Codice, Sigla, Descrizione)

UNI: {Sigla, Descrizione}

VNN: {Sigla, Descrizione}

La superficie della neve è *Descrizione*, cui corrisponde la sigla *Sigla* ed il codice *Codice*.

SONDA (Rilievo, *q*, *p*, *n*, *h*, *x*, Δ , *R*)

CE: Rilievo → RILIEVO (Numero)

VNN: {*q*, *n*, *h*, *R*}

Per una singola serie di colpi del test della sonda relativa al rilievo *Rilievo*, il peso dei tubi sonda è *q* N, la massa battente pesa *p* N ed è stata fatta cadere da un'altezza di *h* cm, affondando di Δ cm rispetto alla serie precedente; il numero di colpi dato è *n* e la sonda è penetrata *x* cm nella neve. Lo strato relativo ad una serie di colpi ha indice di durezza calcolato *R* N.

STAZIONE (Id, Codice, Località, Altitudine, Esposizione, Pendenza, Competenza, Comune, Geometria)

CE: Comune → COMUNE (Codice ISTAT)

VNN: {Codice, Località, Altitudine, Esposizione, Pendenza, Comune}

Vincolo topologico: la geometria di una stazione dev'essere contenuta all'interno della geometria del Comune di appartenenza.

La stazione di rilievo con identificatore *Id* e codice *Codice* si trova a *Altitudine* m s.l.m., è esposta a *Esposizione*, ha inclinazione *Pendenza* °, è sotto la competenza della stazione forestale *Competenza*, si trova nel Comune di codice *Comune* ed ha geometria *Geometria*.

TEST MANO (Codice, Sigla, Descrizione)

UNI: {Sigla, Descrizione}

VNN: {Sigla, Descrizione}

Uno strato del manto nevoso ha resistenza *Codice* al test della mano;

TNEVE (Rilievo, H, T)

CE: Rilievo → RILIEVO (Numero)

VNN: {T}

Nel rilievo *Rilievo* la neve alla profondità *H* cm misura *T* °C.

VENTO (Codice, Descrizione)

UNI: {Descrizione}

VNN: {Descrizione}

Durante un rilievo è stata rilevata l'attività eolica *Descrizione* di codice *Codice*.

c. Vincoli di integrità

Oltre alle “regole aziendali” della progettazione concettuale riportate in *Tabella 2*, la base di dati dovrà soddisfare questi vincoli di integrità:

1. Ad ogni occorrenza di RILIEVO deve corrispondere almeno un'occorrenza di TNEVE.
2. Ad ogni occorrenza di RILIEVO deve corrispondere almeno un'occorrenza di DENSITÀ.
3. Ad ogni occorrenza di RILIEVO deve corrispondere almeno un'occorrenza di ANALISISTRATI.

Tali vincoli sono rappresentati da cardinalità minime 1 nel diagramma concettuale, ma non sono esprimibili nello schema della base di dati.

4. PROGETTAZIONE FISICA

È la realizzazione in senso stretto della base di dati. Completa lo schema logico specificando i parametri fisici di memorizzazione di dati. Basandosi sul modello fisico dei dati proprio del DBMS, viene prodotto uno schema fisico dei dati.

La base di dati è stata implementata usando PostgreSQL 9.4 tramite il tool pgAdmin III.

Vengono effettuate alcune abbreviazioni ai nomi delle tabelle e degli attributi rispetto allo schema della base di dati. Si riportano le modifiche non intuibili: i nuovi nomi richiamano le sigle che si leggono in Figura 1: Modelli 2 e 3 AINEVA.

Tabella 3: modifiche ai nomi

Nome schema relazionale	Nome tabella
Acqua	th
Analisi strati	strato
Dimensione grani	e
Forma grani	ff
Nubi	n
Precipitazioni	ww
Quota vento	q
Rugosità superficiale	sf
Test mano	r_mano
Vento	v

Il codice per la creazione della base di dati si può consultare dal backup; di seguito si riportano solo alcuni casi rilevanti di domini, tabelle, funzioni, trigger e viste creati.

a. Domini

Vengono creati alcuni domini per i valori che verranno impiegati nelle tabelle.

dom_e

i possibili valori assumibili in mm dalla dimensione dei grani

```
CREATE DOMAIN dom_e AS real  
CONSTRAINT e_valido CHECK (VALUE between 0.1 and 15.0);
```

dom_q_sonda

I possibili valori assumibili dal peso [N] dei tubi sonda

```
CREATE DOMAIN dom_q_sonda AS integer
    CONSTRAINT q_valido CHECK (VALUE in (10, 15, 20, 30, 40, 50));
```

dom_r

I possibili valori assumibili dal codice del test della mano

```
CREATE DOMAIN dom_r AS character(1)
    CONSTRAINT dom_r_check CHECK (VALUE in ('1', '2', '3', '4', '5', '6'));
```

dom_th

I possibili valori assumibili dal codice del contenuto in acqua liquida

```
CREATE DOMAIN dom_th AS character(1)
    CONSTRAINT dom_th_check CHECK (VALUE in ('1', '2', '3', '4', '5'));
```

b. Tabelle

densita

```
CREATE TABLE densita(
    rilievo smallint NOT NULL,
    h smallint NOT NULL,
    peso smallint,
    dens smallint NOT NULL,
    CONSTRAINT densita_pkey PRIMARY KEY (rilievo, h),
    CONSTRAINT densita_id_fkey FOREIGN KEY (rilievo)
        REFERENCES rilievo (num) ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT densita_densita_check CHECK (dens = (2 * peso)));
```

La chiave primaria è costituita dalla coppia di `rilievo` e `h`: ciò indica che in un rilievo ad una certa profondità del manto nevoso ci può essere una e una sola misura di densità.

Si noti l'ultimo vincolo `densita_densita_check`: questo sta ad indicare che se `peso` e `dens` sono entrambi noti, quest'ultimo deve coincidere numericamente col doppio del valore del peso. In tal modo è soddisfatta la RD2 di Tabella 2 e si evitano inserimenti errati nei valori di densità.

sonda

```
CREATE TABLE sonda(
    rilievo smallint NOT NULL,
    q dom_q_sonda NOT NULL,
    p integer NOT NULL,
    n integer NOT NULL,
    h integer NOT NULL,
    x integer NOT NULL,
    delta integer NOT NULL,
    r integer NOT NULL,
    CONSTRAINT sonda_pkey PRIMARY KEY (rilievo, p, x, delta),
    CONSTRAINT sonda_id_fkey FOREIGN KEY (rilievo)
        REFERENCES rilievo (num) ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT sonda_check CHECK (delta <= x));
```

La chiave primaria è formata da quattro attributi.

Il vincolo `sonda_check` verifica che l'affondamento della sonda rispetto alla serie di colpi precedente (`delta`) non superi la penetrazione della sonda nella neve (`x`).

stazione

```
CREATE TABLE stazione(
    id smallint NOT NULL,
    cod character(4) NOT NULL,
    loc text NOT NULL,
    alt smallint NOT NULL,
    exp character varying(2) NOT NULL,
    pend smallint NOT NULL,
    comp text,
    comune character(6),
    geom geometry(Point,3004),
    CONSTRAINT stazione_pkey PRIMARY KEY (id),
```

```

CONSTRAINT stazione_comune_fkey FOREIGN KEY (comune)
    REFERENCES comune (cod) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE NO ACTION,
CONSTRAINT stazione_alt_check CHECK (alt >= 600 AND alt <= 2800),
CONSTRAINT stazione_pend_check CHECK (pend >= 0 AND pend <= 45));

```

L'attributo `geom` memorizza geometrie di punti con coordinate nel sistema di riferimento di codice EPSG:3004, cioè Gauss-Boaga fuso est.

Gli ultimi due vincoli non sono stati richiesti esplicitamente nei precedenti paragrafi, tuttavia è sensato crearli per evitare di memorizzare valori anomali: il primo verifica che l'altitudine di una stazione sia compresa tra 600 m e 2800 m (essendo il punto più elevato della Regione non più alto di tale quota), il secondo che la pendenza non assuma valori negativi e che non oltrepassi i 45° (inclinazione considerata estrema).

Strato

```

CREATE TABLE strato(
    rilievo smallint NOT NULL,
    h1 integer NOT NULL,
    h2 integer NOT NULL,
    th dom_th NOT NULL,
    ff1 character varying(2) NOT NULL,
    ff2 character varying(2),
    e1 dom_e,
    e2 dom_e,
    r dom_r NOT NULL,
    comm text,
    CONSTRAINT strato_pkey PRIMARY KEY (rilievo, h1, h2),
    CONSTRAINT ff1_fograni_fkey FOREIGN KEY (ff1)
        REFERENCES ff (cod)
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT ff2_fograni_fkey FOREIGN KEY (ff2)
        REFERENCES ff (cod)
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT r_r_mano_fkey FOREIGN KEY (r)

```

```

REFERENCES r_mano (cod)
ON UPDATE CASCADE ON DELETE NO ACTION,
CONSTRAINT strato_rilievo_fkey FOREIGN KEY (rilievo)
REFERENCES rilievo (num)
ON UPDATE CASCADE ON DELETE NO ACTION,
CONSTRAINT th_acqua_fkey FOREIGN KEY (th)
REFERENCES th (cod)
ON UPDATE CASCADE ON DELETE NO ACTION,
CONSTRAINT strato_check CHECK (h2 <= h1),
CONSTRAINT strato_ff1ff2_check CHECK (ff1 <> ff2 OR e1 <> e2));

```

Si noti l'uso di diversi domini originali.

La chiave primaria è costituita dal numero del rilievo `rilievo` e dai limiti superiore `h1` ed inferiore `h2` dello strato analizzato.

Nei vincoli `ff1_fograni_fkey` e `ff2_fograni_fkey` è volutamente lasciata l'istruzione `ON UPDATE NO ACTION`, in quanto il cambiamento di codice nella forma dei grani nella tabella `ff` è raro e delicato (avviene nel caso di una revisione scientifica internazionale della classificazione dei cristalli di neve): un suo aggiornamento potrebbe comportare degli errori.

Il vincolo `strato_check` verifica che il limite inferiore dello strato sia minore o uguale al limite superiore (è uguale nel caso in cui si valuti la superficie del manto nevoso, di spessore convenzionalmente infinitesimo).

Il vincolo `strato_ff1ff2_check` verifica che, se in uno strato sono presenti due forme di cristalli uguali, queste differiscano per la dimensione dei grani e viceversa (cioè che eventuali grani di dimensione identica abbiano forme diverse).

t_neve

```

CREATE TABLE t_neve(
    rilievo smallint NOT NULL,
    h smallint NOT NULL,
    t real NOT NULL,
    CONSTRAINT t_neve_pkey PRIMARY KEY (rilievo, h),
    CONSTRAINT t_neve_id_fkey FOREIGN KEY (rilievo)
    REFERENCES public.rilievo (num)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT t_neve_t_check CHECK (t <= 0.5));

```

L'ultimo vincolo impone che non vengano memorizzate temperature della neve maggiori di 0.5 °C.

c. Funzioni

22

compute_r(dom_q_sonda, integer, integer, integer, integer, integer)

```
CREATE OR REPLACE FUNCTION compute_r(q dom_q_sonda, p integer, n integer,
h integer, x integer, delta integer)

    RETURNS integer AS

$$

    declare r integer;

    begin

        if delta = 0 then

            r := p + q;

        else

            r := (p * n * h)/delta + p + q;

        end if;

        return r;

    end;

$$

LANGUAGE plpgsql;
```

Questa funzione non fa altro che applicare la formula di cui a pagina 3 e soddisfare la RD1 di Tabella 2. Vengono passati i parametri del test della sonda e la funzione calcola l'indice di durezza R. Si noti l'istruzione if... else che garantisce l'assenza di errori di calcolo nel caso in cui il valore di `delta` sia 0⁵.

Successivamente, con la funzione trigger `f_compute_r()`⁶, si permette il calcolo dell'indice di durezza nella tabella r.

strato()

```
CREATE OR REPLACE FUNCTION strato()

    RETURNS trigger AS
```

⁵ Questo accade quando, appoggiando la massa battente prima di iniziare le serie di colpi, non c'è affondamento della sonda nella neve. Non essendoci ancora state serie di colpi, in questo caso il valore di `delta` dovrebbe essere a rigore NULL; semplificando e per evitare di rilassare il vincolo di valor non nullo sull'attributo `delta`, si assume che valga 0 in questo caso.

⁶ Si veda la base di dati per il codice.

```

$BODY$
begin
    perform * from rilievo
        where num not in
            (select distinct rilievo
             from strato);
    if found
    then
        raise notice 'Ricordati di inserire la stratigrafia';
        return new;
    else
        return new;
    end if;
end;
$BODY$

LANGUAGE plpgsql;

```

Questa funzione trigger verifica che per ogni rilievo sia inserita l'analisi stratigrafica: se la funzione trova un rilievo il cui attributo `num` non è presente in `rilievo` di `strato`, l'inserimento di un nuovo rilievo è concesso ma viene stampata una notifica di promemoria.

Le funzioni `t_neve()` e `densita()` eseguono il medesimo controllo.

In questo modo si soddisfano i vincoli di integrità espressi in linguaggio naturale di cui a pagina 16.

La funzione `staz_sonda()` ha un'utilità simile e serve a soddisfare la RV1 di Tabella 2 (durante ogni rilievo in stazione fissa si deve eseguire il "test della sonda"). Si veda la base di dati per il codice.

staz_cod()

```

CREATE OR REPLACE FUNCTION staz_cod()
    RETURNS trigger AS
$BODY$
begin
    perform * from stazione
        where cod = new.cod
            and cod <> 'ITGA'

```

```

        and cod <> 'ITIN';
    if found
    then
        raise exception 'Stai inserendo il codice di una stazione fissa già
esistente';
    else
        return new;
    end if;
end;
$BODY$
LANGUAGE plpgsql;

```

Visto che non è possibile imporre il vincolo d'unicità all'attributo `cod` di `stazione` (potendosi ripetere ITIN e ITGA) con questa funzione si impedisce l'inserimento di due stazioni fisse con lo stesso codice.

È così soddisfatta la RV5 di Tabella 2.

staz_data()

```

CREATE OR REPLACE FUNCTION staz_data()
    RETURNS trigger AS
$BODY$
begin
    perform * from rilievo
        where staz = new.staz
            and data = new.data;
    if found
    then
        raise exception 'Una stazione non può avere più di un rilievo al
giorno. Controlla la data.';
    else
        return new;
    end if;
end;
$BODY$
LANGUAGE plpgsql;

```


La funzione sopra impedisce l'inserimento di un rilievo al giorno nella medesima stazione e soddisfa la RV3 di Tabella 2.

staz_ok()

25

```
CREATE OR REPLACE FUNCTION staz_ok()
  RETURNS trigger AS
$BODY$
begin
  perform *
  from (select * from stazione
        where cod = 'ITGA'
          and loc in (select loc from stazione
                     where cod <> 'ITGA' and cod <> 'ITIN')
        union
        select * from stazione
        where cod = 'ITIN'
          and loc in (select loc from stazione
                     where cod <> 'ITGA' and cod <> 'ITIN'))
  as RilieviFissi;

  if found
  then
    raise exception 'Codice e località non compatibili!';
  else
    return new;
  end if;
end;
$BODY$

LANGUAGE plpgsql;
```

Questa funzione verifica che non ci siano stazioni itineranti (codice ITIN o ITGA) che siano nelle stesse località di stazioni fisse e viceversa. In tal modo si soddisfa la RV4 di Tabella 2.

stazga()

```
CREATE OR REPLACE FUNCTION stazga()
  RETURNS trigger AS
$BODY$
begin
  perform * from
  (select s.cod, r.tipo
   from ((rilevatore r join rilevazione ri on r.id=ri.rilevatore)
        right join rilievo ril on ril.num=ri.rilievo)
        join stazione s on s.id=ril.staz
        where r.tipo = 'Guida alpina')
  as rilGGAA
  where cod <> 'ITGA';
  if found
  then
    raise exception 'Una guida alpina deve fare rilievi in stazioni con
codice ITGA!';
  else
    return new;
  end if;
end;
$BODY$
LANGUAGE plpgsql;
```

La funzione non permette che una guida alpina effettui rilievi in stazioni diverse da quelle con codice ITGA. Viene così soddisfatta la RV2 di Tabella 2.

d. Trigger

Le tabelle in cui sono stati inseriti trigger sono rilevazione, rilievo, sonda e stazione: si rimanda alla base di dati per i codici relativi. Di seguito si riporta un caso significativo.

control_update_stazdata

```
CREATE TRIGGER control_update_stazdata
  BEFORE UPDATE
  ON rilievo
  FOR EACH ROW
  WHEN (((old.staz IS DISTINCT FROM new.staz) OR (old.data IS DISTINCT
FROM new.data)))
  EXECUTE PROCEDURE staz_data();
```

27

Trigger della tabella `rilievo`. Si noti la condizione `when`: il trigger viene attivato solo se vengono modificati il numero della stazione o la data.

e. Verifica di vincoli d'integrità complessi

Grazie alla creazione dei trigger vengono soddisfatti i vincoli di integrità individuati in Tabella 2: *“Regole aziendali” per lo schema concettuale* di pagina 11 ed al paragrafo c di pagina 16.

A rigore, la RV1 di Tabella 2 e i vincoli di pagina 16 sono solo parzialmente soddisfatti: infatti i trigger `control_insert_sonda`, `control_insert_densita`, `control_insert_strato`, `control_insert_tneve` ed i rispettivi trigger del tipo *“control_update”* della tabella `rilievo`, lasciano effettuare inserimenti ed aggiornamenti anche se i vincoli vengono violati, limitandosi a stampare una notifica di promemoria.

Per il controllo di questi casi, si potranno formulare le seguenti interrogazioni booleane per verificare che i vincoli non vengano violati: se dovesse essere altrimenti, si procederà all’inserimento dei dati mancanti (test della sonda, misure di densità, analisi stratigrafica, temperatura della neve).

RV1) I valori di Sonda devono essere sempre memorizzati se il rilievo avviene in una stazione fissa

```
select not exists
(select * from rilievo, stazione
 where num not in (select rilievo from sonda)
  and rilievo.staz = stazione.id
  and stazione.cod <> 'ITGA'
  and stazione.cod <> 'ITIN');
```

L’interrogazione dà `true` se tutti gli attributi `num` dei rilievi fissi (cioè con codice stazione diverso da `ITIN` o `ITGA`) sono presenti nella tabella `sonda`, `false` altrimenti.

Ad ogni occorrenza di RILIEVO deve corrispondere almeno un'occorrenza di DENSITÀ.

```
select not exists
(select *from rilievo
where num not in (select distinct rilievo from densita));
```

28

L'interrogazione dà true se tutti gli attributi num di rilievo sono presenti nella tabella densita, false altrimenti.

Gli analoghi vincoli di t_neve e strato si possono verificare sostituendo il nome delle relative tabelle a densita nell'interrogazione sopra.

A questo punto tutti i vincoli d'integrità individuati in precedenza possono essere soddisfatti.

f. Viste

Vengono create alcune viste per facilitare l'ideale interfacciarsi dell'utenza con la base di dati. Si rimanda al backup per il codice utilizzato. Di seguito si riporta un esempio significativo

intestazione

```
CREATE OR REPLACE VIEW public.intestazione AS
SELECT r.num AS numero_rilievo,
       s.cod AS codice_stazione,
       r.data,
       r.ora,
       s.loc AS "località",
       s.alt AS altitudine,
       s.exp AS esposizione,
       s.pend AS pendenza,
       st_astext(s.geom) AS coordinate,
       r.ww AS precipitazioni,
       r.n AS nubi,
       r.v AS vento,
       r.q AS quota_vento,
```

```
ri.nome AS nome_rilevatore,  
ri.cognome AS cognome_rilevatore,  
s.comp AS stazione_forestale_di_competenza,  
r.ta AS temperatura_aria,  
r.hs AS altezza_manto_nevoso,  
r.sf AS "rugosità_superficiale",  
r.note  
FROM stazione s  
      JOIN rilievo r ON s.id = r.staz  
      FULL JOIN (rilevazione ril  
      JOIN rilevatore ri ON ril.rilevatore = ri.id) ON ril.rilievo = r.num  
ORDER BY r.data, r.ora;
```

Questa vista riflette precisamente l'intestazione dei modelli 2 e 3 AINEVA che si può vedere in Figura 1 e può facilitare la consultazione dei rilievi senza dover ogni volta ricorrere ad operazioni di join per, ad esempio, ricavare i nomi dei rilevatori oppure la località in cui è stato eseguito un rilievo.

Vengono aggiunte delle etichette ai nomi degli attributi per una comprensione immediata degli stessi.

5. POPOLAMENTO DELLA BASE DI DATI

Per popolare il database creato sono stati usati i dati dei rilievi nivologici di dicembre 2014 e gennaio 2015 forniti dalla Struttura stabile centrale per l'attività di prevenzione del rischio da valanga (Servizio foreste e corpo forestale) della Regione Autonoma Friuli Venezia Giulia

a. Dati grezzi

I dati grezzi erano dei fogli di calcolo estratti da un database Microsoft Access in cui erano memorizzate le informazioni sui rilievi. I dati delle stazioni itineranti erano incorporati a quelli dei rilievi, motivo per cui sono stati scorporate le informazioni proprie delle stazioni, creando un foglio di calcolo ad hoc.

Le coordinate delle varie stazioni si trovavano sparse su più fogli di calcolo, spesso memorizzate in sistemi di riferimento differenti (EPSG:3004, EPSG:4326, EPSG:23033, ...) ed in ogni caso come semplice testo: le coordinate sono state così uniformate al SR EPSG:3004 (Gauss-Boaga fuso est) tramite un tool convertitore online [5].

I fogli di calcolo dalla prova penetrometrica col test della sonda e dell'analisi stratigrafica riflettevano la struttura delle tabelle `sonda` e `strato` della base di dati, rispettivamente.

Nomi e cognomi dei rilevatori erano memorizzati in un'unica colonna del foglio di calcolo "Rilievi.xlsx": è stato dunque creato un foglio ad hoc con nome e cognome dei rilevatori, aggiungendo un id progressivo ed il tipo di figura professionale. È stato poi creato un foglio che associa il numero di un rilievo ai rilevatori che l'hanno eseguito.

Sono stati usati gli *shapefile* dei Comuni FVG 2016 [6] e delle Valanghe Rilevate [7], entrambi scaricati nel SR di progetto (EPSG:3004), ed importati nella base di dati grazie al DB Manager di QGIS 2.12.2-Lyon. Lo *shapefile* delle valanghe servirà unicamente per una selezione spaziale descritta nel prossimo paragrafo.

Eseguito qualche adattamento nella formattazione e nell'ortografia, i fogli di calcolo sono stati convertiti in file *Comma Separated Values* (*.csv), uno per ognuna delle tabelle presenti nel database, ed importati nella base di dati tramite pgAdmin.

b. Raffinamento della base di dati

Le coordinate delle stazioni sono state inserite nell'attributo `geom` della tabella `stazione`. Ad esempio, per inserire la geometria della stazione con id 29 l'istruzione è la seguente:

```
update stazione
set geom = st_geomfromtext
('POINT(2403168 5136014)',3004)
where id = 29;
```

L'appartenenza di una stazione ad un comune non era nota nei dati grezzi: per inserire il codice ISTAT di un comune su cui una stazione insiste nell'attributo `comune` di `stazione` è stata formulata l'interrogazione

```
update stazione
set comune = ( select c.cod
                from comune c
                where st_contains (c.geom, stazione.geom));
```

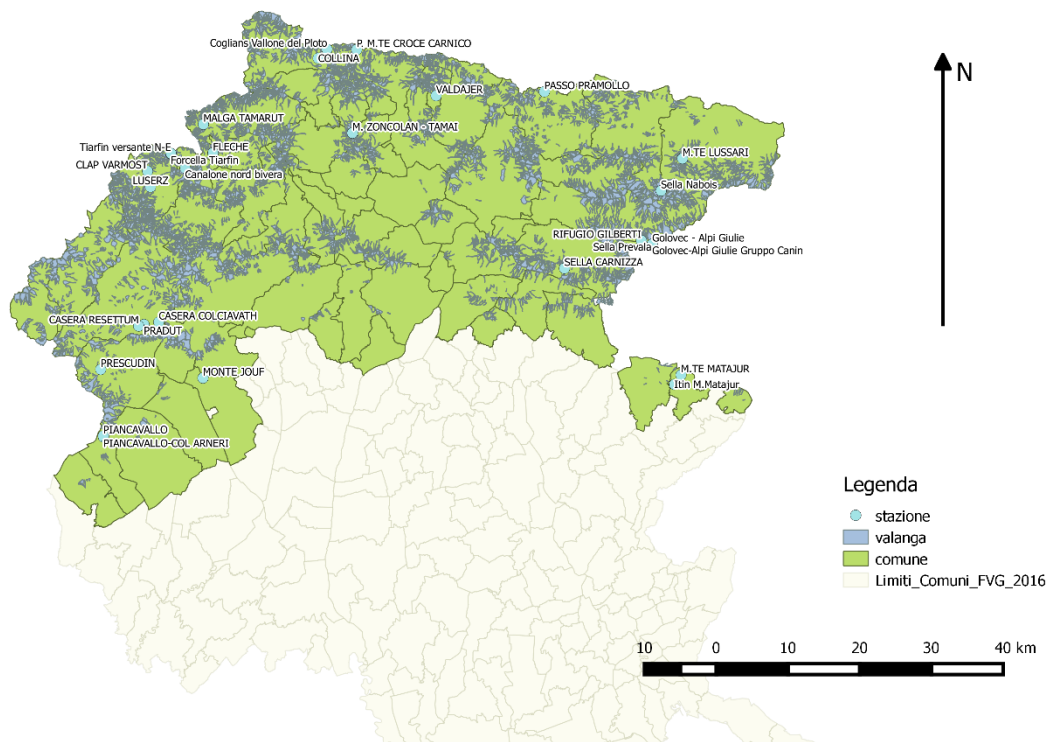
In questa maniera il vincolo per cui la geometria di una stazione dev'essere contenuta all'interno del comune di appartenenza viene automaticamente rispettato.

Per le stazioni di cui non erano note le geometrie e di cui si poteva risalire con certezza al comune su cui insistevano, i codici ISTAT sono stati inseriti manualmente con comandi del tipo

```
update stazione
set comune = '030117'
where id = 26;
```

Per consentire operazioni computazionalmente più efficienti, si è scelto di eliminare dallo shapefile dei comuni le geometrie su cui non sono presenti stazioni di rilievo o valanghe. Per far ciò è stata formulata la seguente interrogazione ottenendo uno shapefile dei comuni "montani" (Figura 3).

```
delete from comune
where cod not in
(select distinct c.cod
 from comune c, stazione s
 where st_within (s.geom, c.geom)
 union
 select distinct c.cod
 from comune c, valanga v
 where st_intersects (v.geom, c.geom));
```



Durante il popolamento della tabella `rilievo` ci si accorge che per alcuni rilievi mancano i dati relativi al vento ed alla quota del vento (attributi `v` e `q`). Per non perdere le altre informazioni che quei rilievi forniscono, si decide di eliminare il vincolo di valor non nullo per quei due attributi.

```
ALTER TABLE rilievo
  ALTER v DROP NOT NULL;
```

```
ALTER TABLE rilievo
  ALTER q DROP NOT NULL;
```

32

Similmente, durante la popolazione di `strato` si vede che alcuni attributi `e1` sono nulli. Si decide e si opera come nel caso precedente, consentendo il valor nullo.

Sempre in `strato`, grazie alla verifica operata dal vincolo `strato_ff1ff2_check`, ci si accorge di dati ridondanti: si sceglie di eliminarli coi comandi

```
UPDATE strato
  SET ff2 = NULL
  WHERE ff1 = ff2
  AND e1=e2;
```

```
UPDATE strato
  SET ff2 = NULL
  WHERE ff1 = ff2
  AND e2 is NULL;
```

In `stazione` si evince come rilievi molto vicini tra di loro presentino nomi di località differenti (cfr. Figura 4). Si sceglie di uniformare i nomi delle località di queste ed altre stazioni itineranti e spostare le altre informazioni nell'attributo `note` di `rilievo`.

Prima di tutto vediamo quali sono i nomi delle località di stazioni itineranti

```
select distinct loc
from stazione
where cod = 'ITIN' or cod = 'ITGA'
group by loc
order by loc
```

Successivamente si può procedere all'aggiornamento

```
update stazione set loc = 'Golovec' where loc = 'Golovec - Alpi Giulie'
or loc = 'Golovec-Alpi Giulie Gruppo Canin';
```

```
update rilievo set note = 'Canalone nord Bivera' where staz = 36;
```

```
update stazione set loc = 'Bivera' where loc = 'Canalone nord bivera';
```

```
update rilievo set note = 'Vallone del Ploto' where staz = 28;
```

```
update stazione set loc = 'Coglians' where loc = 'Coglians Vallone del
Ploto';
```



```

update rilievo set note = 'Rilievo effettuato presso la centralina' where
staz = 38;

update stazione set loc = 'Forcella Baldass' where loc = 'Forcella
Baldass - Centralina';

update stazione set loc = 'Matajur' where loc = 'Itin M.Matajur';

update rilievo set note = 'Forcella Tiarfin' where staz = 32;

update stazione set loc = 'Tiarfin' where loc = 'Forcella Tiarfin';

update rilievo set note = 'Tiarfin versante NE' where staz = 33;

update stazione set loc = 'Tiarfin' where loc = 'Tiarfin versante N-E';

```

33

c. Verifica di vincoli d'integrità

Per come sono stati definiti i trigger che invocano le funzioni `sonda()`, `densita()`, `t_neve()` e `strato()`, l'inserimento di rilievi carenti delle prove relative al Modello 3 AINEVA o di rilievi fissi senza Modello 2 è permesso, limitandosi il sistema ad una notifica. Verifichiamo questi vincoli con interrogazioni booleane:

```

select not exists
(select * from rilievo, stazione
 where num not in (select rilievo from sonda)
   and rilievo.staz = stazione.id
   and stazione.cod <> 'ITGA'
   and stazione.cod <> 'ITIN');

```

Il risultato è `true`, dunque tutti i rilievi in stazione fissa sono provvisti del test della sonda. La RV1 di Tabella 2 è rispettata.

Verifichiamo che in ogni rilievo sia stata misurata la densità della neve:

```

select not exists
(select * from rilievo
 where num not in (select distinct rilievo from densita));

```

Il risultato è `false`: eseguendo la query interna si scopre che il rilievo numero 20 non ha densità della neve. Facendo uso delle viste create si può scoprire il motivo.

```

select * from intestazione
where numero_rilievo not in
  (select distinct rilievo from densita);

select * from analisi_strati where numero_rilievo = 20;

```

L'altezza del manto nevoso di quel rilievo era di soli 16 cm e lo strato centrale era composto da una crosta da fusione e rigelo, rendendo difficoltosa l'operazione di carotaggio per misurare la densità. Inoltre gli altri strati sono eterogenei, dunque una misura della densità non avrebbe fornito informazioni significative sullo stato dell'intero manto nevoso.

Il vincolo non è rispettato, ma per questo rilievo si tollera la mancanza di misure di densità per non perdere le altre informazioni.

Con query analoghe si può verificare infatti che tutti i rilievi dispongono di analisi stratigrafica e di temperature della neve.

Verifichiamo con un'interrogazione booleana che tutte le geometrie di `stazione` (quando non nulle) siano contenute all'interno delle geometrie di `comune`⁷

```
select not exists (
    select distinct s.loc, s.cod, s.geom
    from stazione s
    where s.geom is not null

    except

    select distinct s.loc, s.cod, s.geom
    from stazione s, comune c
    where st_within (s.geom, c.geom)
);
```

Il risultato della query è `false`: ciò vuol dire che alcune geometrie non si trovano all'interno delle geometrie di `comune`. Eseguendo la query interna si può verificare che tre rilievi itineranti sono stati eseguiti fuori dalle geometrie di `comune`.

Visualizzandole in QGIS (Figura 4) ci si accorge della posizione: due rilievi sono stati eseguiti nei pressi del Monte Tiarfin, ma in territorio veneto, essendo presumibilmente i rilevatori saliti da Sella Razzo. Un altro rilievo è stato eseguito a Sella Prevala, dunque è facile aver sconfinato in territorio sloveno. Questo è perfettamente comprensibile e casi come questi vengono tollerati; diverso sarebbe stato avere stazioni lontane diverse decine di chilometri dai confini regionali, segno di un errore nelle coordinate.

⁷ Si noti che aver inserito in `stazione` il codice ISTAT dei comuni su cui le stazioni insistono tramite una query spaziale non esclude che alcune geometrie non siano contenute all'interno della Regione, per errore nelle coordinate od in caso di rilievo effettuato fuori dai confini regionali.

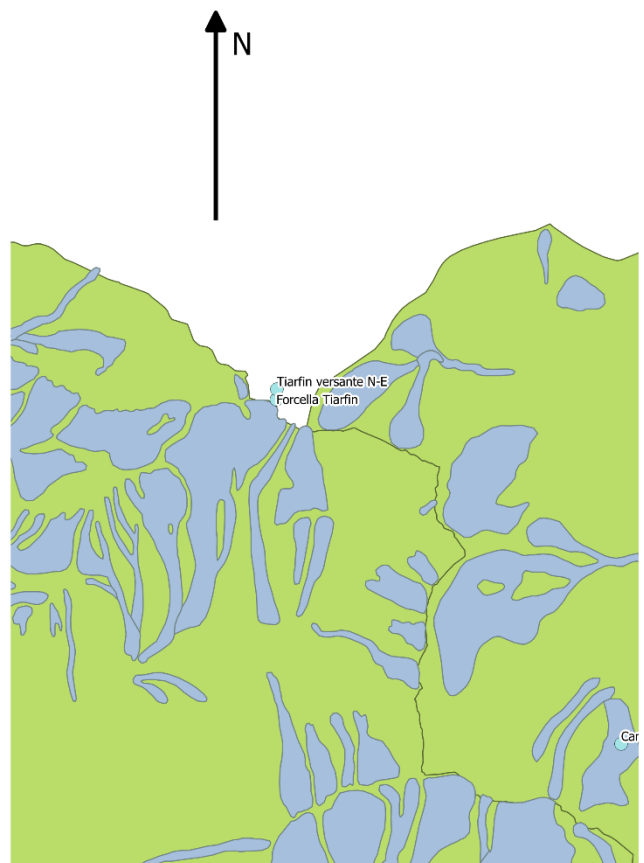
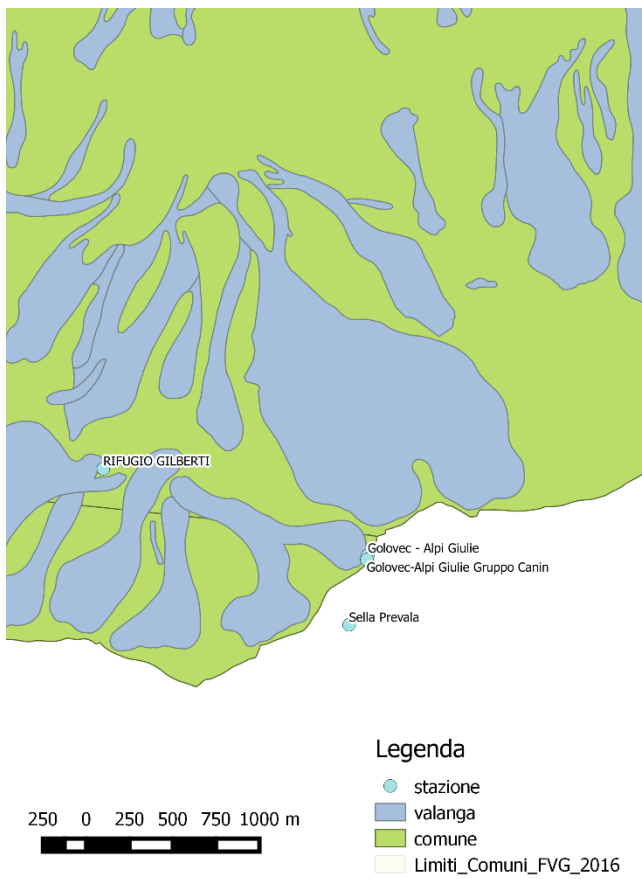


Figura 4: Rilievi fuori regione. Scala 1:25000

6. INTERROGAZIONI SQL

Si propone ora qualche esempio di interrogazione sulla base di dati rilevante nel contesto di questo sistema informativo.

Proviamo a soddisfare innanzitutto le richieste del paragrafo 1.c Esempi di operazioni

1. Inserisci un nuovo rilievo con tutti i dati misurati

Per prima cosa verifichiamo il numero dell'ultimo rilievo

```
select max(num) from rilievo;
```

Successivamente inseriamo tutti i dati del rilievo eseguito il 4 febbraio 2016

```
insert into rilievo
values
(30, (select id from stazione where cod = '7RGI'), '2016-02-04',
'12:00:00', '00', '7', '2', '5', -4.8, 86, 'Rilievo per corso
aggiornamento AINEVA 2a', '5');
```

```
insert into rilevazione
values
(30, (select id from rilevatore where cognome = 'Iacolettig')),
(30, (select id from rilevatore where cognome = 'Lizzero'));
```

```
insert into sonda values
(30, 10, 0, 0, 0, 40,0),
(30, 10, 10, 0, 0, 40,0),
(30, 10, 10, 7, 10, 42,2),
(30, 10, 10, 3, 15, 43,1),
(30, 10, 10, 3, 25, 44,1),
(30, 10, 10, 3, 40, 45,1),
(30, 10, 10, 10, 50, 50,5),
(30, 10, 10, 20, 40, 60,10),
(30, 10, 10, 10, 50, 65,5),
(30, 10, 10, 8, 50, 69,4),
(30, 10, 10, 10, 35, 75,6),
(30, 10, 10, 3, 35, 78,3),
(30, 10, 10, 1, 30, 79,1),
(30, 10, 10, 1, 20, 80,1),
(30, 10, 10, 1, 20, 85,5),
(30, 10, 10, 3, 10, 86,1);
```

```
insert into strato values
(30, 80, 72, 1, '2a', '3b', 1, 1, '3'),
(30, 72, 70, 1, '3b', '1f', 1, 1.5, '3'),
(30, 50, 48, 1, '8r', null, null, null, '5'),
(30, 48, 45, 1, '6a', '6b', null, null, '4'),
(30, 45, 6, 1, '6b', null, null, null, '3'),
(30, 6, 0, 1, '5a', '5x', 3, null, '2');
```

```
insert into t_neve values
```

```
(30, 85, -7),
(30, 80, -5.5),
(30, 70, -4.8),
(30, 60, -3.4),
(30, 50, -2),
(30, 40, -1.5),
(30, 30, -1.4),
(30, 20, -1.3),
(30, 10, -1.3),
(30, 0, -1.5);
```

```
insert into densita values
(30, 80, 65, 130),
(30, 40, 200, 400);
```

- 2. Calcola l'indice di durezza R del "test della sonda".**
- 3. Assegna ad un rilievo fisso i dati già noti del campo di rilievo fisso relativo.**

Per come è costruita la base di dati, queste richieste sono automaticamente soddisfatte ogniqualvolta vengono inseriti nuovi rilievi. Il calcolo di R avviene grazie ad un trigger, l'assegnamento dei dati già noti della stazione ad un rilievo fisso è dovuto al vincolo d'integrità referenziale tra `rilievo` e `stazione`.

- 4. Visualizza in ambiente GIS la località di ogni rilievo di cui sono note le coordinate.**

È sufficiente interfacciarsi con un software GIS e visualizzare le geometrie disponibili di `stazione`. Un esempio in Figura 3 a pagina 31.

- 5. Trova tutte le informazioni relative all'ultimo rilievo effettuato.**

Basta interrogare la vista `intestazione` e le tabelle `strato`, `sonda`, `densita`, `t_neve` chiedendo che il rilievo sia avvenuto nella data e l'ora più recenti.

```
select *
from intestazione
where data >= (select max(data) from rilievo)
      and ora >= (select max(ora) from rilievo);

select *
from strato
where rilievo in
      (select num
       from rilievo
       where data >= (select max(data) from rilievo)
            and ora >= (select max(ora) from rilievo));
```

Per le restanti tabelle, basta sostituire il loro nome a `strato` nell'ultima query.

Vista la frequenza di operazioni come queste, si può creare una vista sui rilievi più recenti (evitando però di inserire l'ora nella `where`, in quanto possono esserci diversi rilievi nello stesso giorno in stazioni diverse)

6. Calcola una statistica sulle località in cui avvengono più rilievi.

Creiamo una vista sulle località ed il numero di rilievi là effettuati

```
create view staz_nril
as
(select località, count(*)
from (select distinct (numero_rilievo), località from intestazione) as
postgres
group by località
order by count(*) desc);
```

38

Dopodiché selezioniamo le località col massimo numero di rilievi

```
select distinct s1.località, s1.count from
staz_nril as s1, staz_nril as s2
where not exists (select from staz_nril s2 where s2.count > s1.count)
order by s1.località;
```

ovvero, usando una funzione aggregata

```
select località, count
from staz_nril
where count = (select max(count) from staz_nril)
order by località;
```

7. Trova le informazioni disponibili su tutte le stazioni di rilievo

Basta semplicemente il comando `select * from stazione;`
Vediamo ora altri esempi di interrogazioni.

8. Trova le informazioni sulle stazioni con località identica

```
select distinct s1.*
from stazione s1, stazione s2
where s1.id <> s2.id
and s1.loc = s2.loc
order by s1.loc;
```

ovvero

```
select distinct s1.*
from stazione s1
where loc in (select loc from stazione group by loc having count (*)>1)
order by s1.loc;
```

9. Trova le coppie di stazioni distanti tra di loro meno di 2000 m e calcola l'effettiva distanza in km.

```

select s1.id, s1.loc, s1.alt, s2.id, s2.loc, s2.alt, st_distance(s1.geom,
s2.geom)/1000 as distanza_km
  from stazione s1, stazione s2
  where s1.id < s2.id
        and st_distance(s1.geom, s2.geom) is not null
        and st_distance(s1.geom, s2.geom)/1000 < 2
/*and s1.cod <> 'ITIN'
   and s2.cod <> 'ITIN'
   and s1.cod <> 'ITGA'
   and s2.cod <> 'ITGA'*/
order by distanza_km;

```

Eseguendo anche il commento si ottiene il confronto per le sole stazioni fisse.

10. Trova le stratigrafie (complete di descrizione qualitativa della forma e dimensione dei grani e del test della mano) dei rilievi eseguiti in una località che inizia per "Golo"

```

select i.località, a.*, e1.descr as descrizione_forma_grani_prevalenti,
e2.descr as descrizione_forma_grani_secondari
  from analisi_strati a
 left join e e1
    on a.dimensione_grani_prevalenti between e1.min and e1.max
 left join e e2
    on a.dimensione_grani_secondari between e2.min and e2.max
 join intestazione i on i.numero_rilievo = a.numero_rilievo
  where a.numero_rilievo in
        (select numero_rilievo
          from intestazione
          where località like 'Golo%')

```

Questa interrogazione fa uso di due viste. Senza di esse, sarebbero necessarie query molto più complesse.

11. Trova tutti i test della sonda in stazione fissa, visualizzando codice stazione e località

```

select s.*, r.staz, st.cod, st.loc
  from sonda s, rilievo r, stazione st
  where s.rilievo = r.num
        and r.staz = st.id
        and st.cod <> 'ITGA'
        and st.cod <> 'ITIN';

```

12. Trova le intestazioni dei rilievi durante i quali non è stato eseguito il test della sonda

```

select *
  from intestazione
  where numero_rilievo not in
        (select rilievo from sonda);

```

13. Trova i rilevatori che non hanno effettuato rilievi

```
select *  
from rilevatore  
where id not in (select rilevatore  
                 from rilevazione);
```

14. Trova il comune di appartenenza delle stazioni che si trovano entro un raggio di 15 km dal rifugio Gilberti, calcolando la distanza.

40

```
select s.id, s.cod, s.loc, c.nome,  
st_distance(s.geom, (select geom from stazione where loc like '%ILB%'))  
as distanza_dal_Gilberti  
from comune c, stazione s  
where st_contains (c.geom, s.geom)  
and s.geom in  
      (select geom from stazione  
       where st_dwithin ((select geom from stazione  
                           where loc like '%ILB%'),geom,15000))  
order by distanza_dal_Gilberti;
```

Si può visualizzare il risultato in una mappa, sottoponendo questa interrogazione al DB Manager di QGIS (ricordandosi di selezionare anche le geometrie delle stazioni e dei comuni) e caricando i nuovi layer.

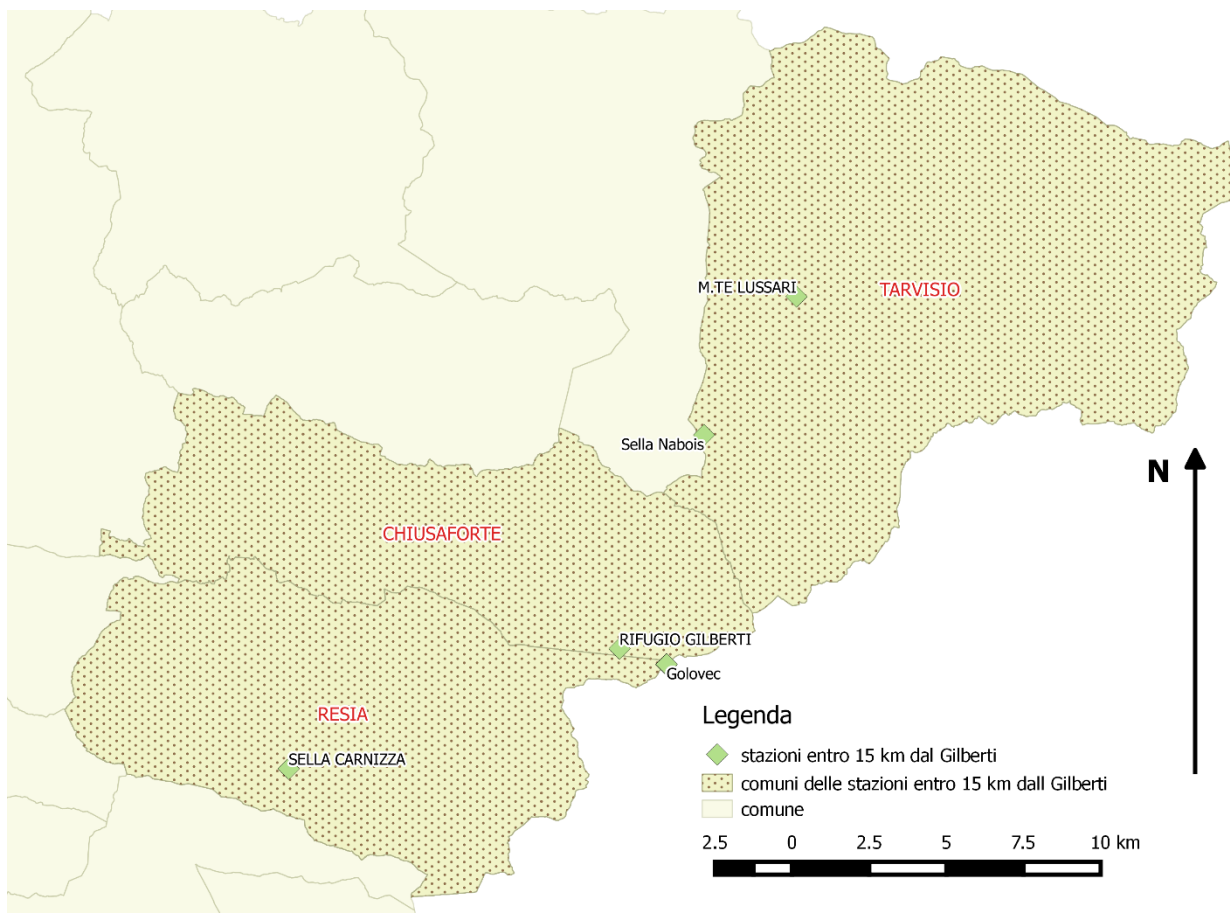


Figura 5: Stazioni entro 15 km dal rifugio Gilberti e comuni di appartenenza

15. Trova i rilievi senza nome dei rilevatori e fornisci la stazione forestale di competenza, se nota

```
select    numero_rilievo,    codice_stazione,    data,    ora,    località,
stazione_forestale_di_competenza
from intestazione
where cognome_rilevatore is null;
```

41

16. Trova i rilievi con altezza del manto nevoso rilevata superiore al metro, ordinando in ordine decrescente

```
select numero_rilievo, codice_stazione, data, ora, località, altitudine,
altezza_manto_nevoso
from intestazione
where altezza_manto_nevoso > 100
order by altezza_manto_nevoso desc;
```

17. Trova la quota del rilievo con massima altezza del manto nevoso misurata

```
select distinct altitudine, altezza_manto_nevoso
from intestazione
where altezza_manto_nevoso in
(select max(altezza_manto_nevoso) from intestazione);
```

18. Trova la quota media dei rilievi itineranti che sono avvenuti nella stessa località

```
select loc as località, avg(alt) as altitudine_media
from rilievo r join stazione s on r.staz = s.id
where s.cod = 'ITIN' or s.cod = 'ITGA'
group by s.loc
having count (s.loc) > 1
order by s.loc;
```

19. Trova il numero del rilievo, l'altezza del manto nevoso, la profondità degli strati più duri (con R massimo) ed il valore di R per i rilievi effettuati da guide alpine

Per prima cosa conviene trovare gli R massimi per ogni test della sonda

```
select rilievo, max(r) from sonda
group by rilievo
order by rilievo;
```

Successivamente usiamo la query sopra come tabella di join e condizione di selezione

```
select    distinct    i.numero_rilievo,    i.altezza_manto_nevoso,    s.x    as
profondità_r_max, r.max as R
from intestazione i join sonda s on i.numero_rilievo = s.rilievo
join (select rilievo, max(r) from sonda
```

```

        group by rilievo) as r
        on r.rilievo = i.numero_rilievo
where s.r = (select max(r) from sonda
            where sonda.rilievo = s.rilievo
            group by rilievo)
and codice_stazione = 'ITGA'
order by i.numero_rilievo;

```

Non facendo uso di viste si devono aggiungere alcuni join. In questo caso ci si riferisce alle guide alpine come figura professionale in senso stretto, senza ricorrere al codice della stazione

42

```

select distinct r.num as numero_rilievo, r.hs as altezza_manto_nevoso,
s.x as profondità_r_max, rs.max as R
from rilievo r join rilevazione ri on r.num = ri.rilievo
join rilevatore ril on ri.rilevatore = ril.id
join sonda s on r.num = s.rilievo
join (select rilievo, max(r) from sonda
      group by rilievo) as rs
      on rs.rilievo = r.num
where s.r = (select max(r) from sonda
            where sonda.rilievo = s.rilievo
            group by rilievo)
and ril.tipo = 'Guida alpina'
order by r.num;

```

20. Trova le due stazioni più lontane tra loro e la distanza in chilometri

```

select s1.cod, s1.loc, s2.cod, s2.loc, st_distance(s1.geom, s2.geom)/1000
as distanza_max
from stazione s1, stazione s2
where st_distance(s1.geom, s2.geom) =
      (select max (st_distance(s1.geom, s2.geom))
       from stazione s1, stazione s2)
and s1.cod < s2.cod

```

Le stazioni più lontane tra loro sono di Col Arneri (Piancavallo) e del Monte Lussari, distanti 89,6 km. Selezionando anche le geometrie si può visualizzare il risultato in QGIS caricando dei nuovi layer

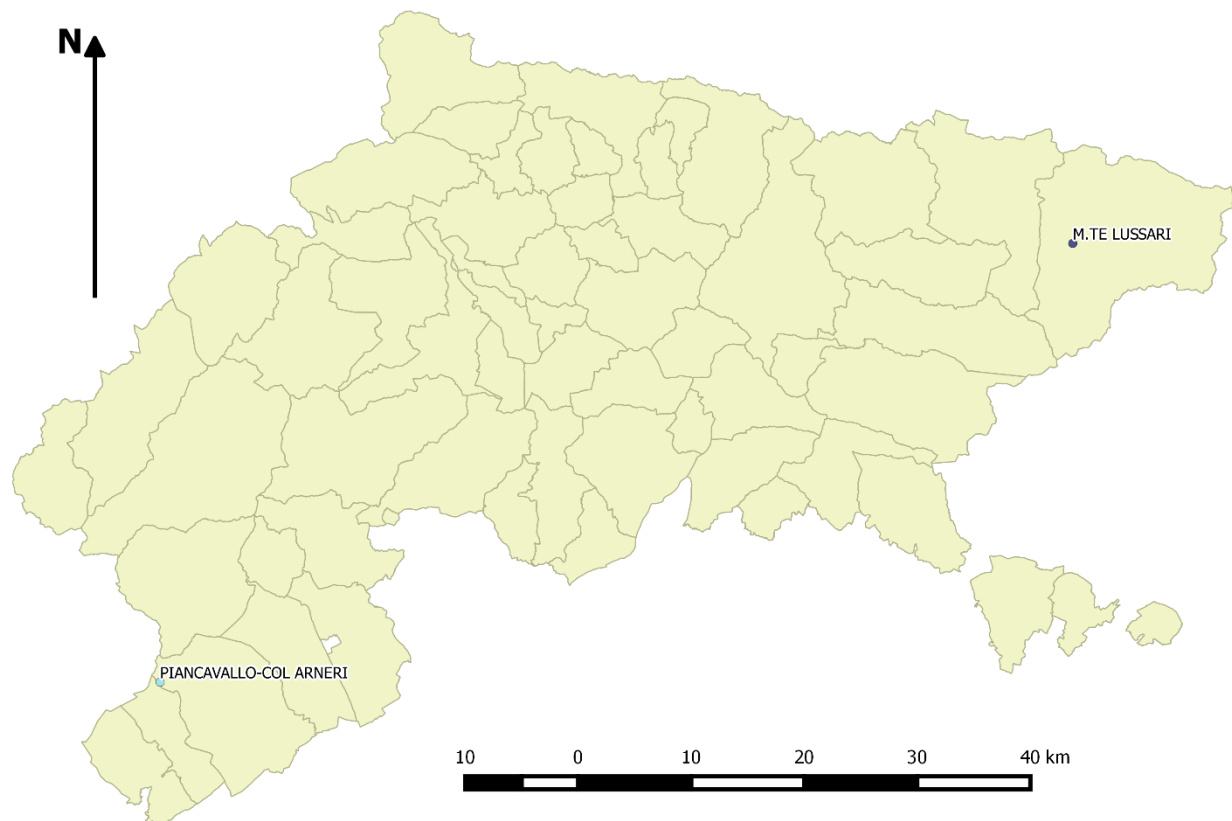


Figura 6: Stazioni alla massima distanza

21. Trova le informazioni sugli strati più deboli (resistenza “pugno” al test della mano), esclusi i primi 5 cm di manto dalla superficie.

```
select r.hs altezza_manto_nevoso, s.*
from analisi_strati s, rilievo r
where s.test_della_mano = 'Pugno'
      and r.num = s.numero_rilievo
      and r.hs > s.limite_superiore + 5
order by numero_rilievo;
```

Riferimenti

- [1] P. Atzeni, Stefano Ceri, Piero Fraternali, Stefano Paraboschi e Riccardo Torlone, Basi di dati, Milano, Italia: McGraw-Hill, 2013, p. 496.
- [2] yWorks - the diagramming company, «yEd - Graph Editor,» yWorks - the diagramming company, [Online]. Available: <https://www.yworks.com/products/yed>. [Consultato il giorno 23 Gennaio 2016].
- [3] D. Gubiani, ChronoGeoGraph: a Development Framework for Spatio-Temporal Databases, 2007.
- [4] D. Gubiani e A. Montanari, A conceptual spatial model supporting topologically-consistent multiple representations, 2008.
- [5] «UltraSoft3D - Conversione tra sistemi di coordinate,» [Online]. Available: http://www.ultrasoft3d.net/Conversione_Coordinate.aspx. [Consultato il giorno 09 02 2016].
- [6] R. A. F. V. Giulia, «IRDATfvg,» [Online]. Available: http://irdat.regione.fvg.it/Distributore/download?idDset=10716&idFmt=383&type=lcr&path=SIT/Shape/Limiti_Comuni_FVG_2016_sh.zip&refSys=3004. [Consultato il giorno 09 02 2016].
- [7] R. A. F. V. Giulia, «IRDATfvg,» [Online]. Available: http://irdat.regione.fvg.it/Distributore/download?idDset=1212&idFmt=383&type=wfs&path=ZONE_RIS C:VIOLA_VALANGHE&refSys=3004. [Consultato il giorno 09 02 2016].