

VSLD

"Java Standard Library Documentation"

Elaborato di progetto per il corso di Basi di Dati
A.A. 2023/2024

Marco Buda - 0001071464
marco.buda3@studio.unibo.it

Analisi dei requisiti

Una piccola casa di sviluppo di nome Sus Microsistemi sta sviluppando un linguaggio di programmazione chiamato Vava, ideato per essere simile al popolare Java, ma più severo nell'applicazione delle norme della programmazione ad oggetti. Insieme alla realizzazione di un compilatore, i programmatori sono attualmente impegnati nell'implementazione di una libreria base per il linguaggio. Per facilitare questo processo, si è deciso di realizzare un sistema per la gestione della documentazione, formato da una base di dati e da un sito web di interfacciamento ad essa. Il sistema dovrà memorizzare anche gli account dei dipendenti e i relativi permessi di modifica della documentazione.

Intervista

Il codice in linguaggio Vava è strutturato in spazi, equivalenti ai package Java. All'interno di questi spazi è possibile collocare interfacce, implementazioni ed enumerazioni. Sia interfacce che implementazioni possono estendere altre interfacce; le restanti forme di ereditarietà sono rese possibili dalla composizione. Le interfacce possono contenere metodi astratti, metodi statici e costanti. Le implementazioni possono contenere campi, metodi, metodi statici e costanti.

Vincoli sulla visibilità:

- Le interfacce e le enumerazioni possono essere pubbliche (Java public) o private (Java package-private), mentre le implementazioni sono necessariamente private (Java package-private).
- I metodi astratti sono necessariamente pubblici (Java public).
- Le costanti, i metodi e i metodi statici, possono essere pubblici (Java public) o privati (Java private).
- I campi sono necessariamente privati (Java private).
- I valori delle enumerazioni sono necessariamente pubblici (Java public).

A ciascuno degli elementi descritti finora, nonché ai parametri delle funzioni, può essere associata della documentazione, composta da un blocco di testo, che può contenere riferimenti ad altri elementi, e da informazioni aggiuntive quali gli autori e la data di ultima modifica.

Altre caratteristiche del linguaggio Vava:

- Non si intende supportare il polimorfismo.
- I costruttori sono sostituiti dal pattern della programmazione a oggetti *Static Factory Method*.

Descrizione del sito:

In una pagina apposita tutti gli utenti, anche non registrati, potranno visualizzare la documentazione delle interfacce pubbliche e delle enumerazioni pubbliche, nonché dei loro membri pubblici.

Per mezzo di un'altra pagina, ad uso esclusivo della casa di sviluppo, sarà possibile visualizzare anche la documentazione degli elementi privati e si potranno aggiungere, modificare o eliminare parti della documentazione, in base ai permessi dei singoli utenti registrati, i quali si dividono in tre categorie:

- Visualizzatori: non possono modificare la documentazione;
- Sviluppatori: possono modificare la documentazione all'interno degli spazi a loro assegnati;
- Amministratori: possono creare nuovi utenti, gestirne i permessi e, in caso di necessità, modificare l'intera documentazione.

Per ciascun tipo di utente si vogliono memorizzare le informazioni di base, ossia nome, cognome, indirizzo email aziendale, password (opportunamente offuscata) e numero badge.

Estrazione dei concetti principali

Termine	Descrizione	Sinonimi
Elemento	Termine generico per riferirsi a spazi, tipi, membri e parametri	
Spazio	Contenitore per l'organizzazione del codice	Package
Tipo	Termine generico per riferirsi a interfacce, implementazioni ed enumerazioni	
Oggetto	Termine generico per riferirsi a interfacce e implementazioni	
Interfaccia	Tipo astratto	
Implementazione	Tipo concreto e istanziabile	Classe
Enumerazione	Tipo che assume uno dei valori stabiliti	
Membro	Termine generico per riferirsi a funzioni, campi, costanti e valori	
Funzione	Termine generico per riferirsi a metodi, metodi astratti e metodi statici	

Metodo astratto	Funzione definita all'interno di un'interfaccia, da realizzare nelle implementazioni	
Metodo statico	Funzione non associata ad una specifica istanza	
Metodo	Funzione implementata	
Parametro	Variabile passata ad una funzione	Argomento
Costante	Membro che funge da variabile non modificabile e non è associato ad una specifica istanza	
Campo	Variabile modificabile associata ad un'istanza	
Blocco	Corpo del testo della documentazione associata ad un elemento	
Riferimento	All'interno di un blocco, porzione di testo che rimanda ad un altro elemento	
Utente	Account personale di un dipendente della casa di sviluppo	
Visualizzatore	Utente senza permessi di modifica	
Sviluppatore	Utente con permessi limitati agli spazi assegnati	
Amministratore	Utente con permessi massimi	

A seguito di un'analisi preliminare e conseguenti interviste supplementari sono emersi i seguenti requisiti aggiuntivi e precisazioni:

- Nonostante l'apparenza, gli spazi non seguono una gerarchia.
- Il nome dei tipi deve essere univoco all'interno degli spazi.
- Un oggetto può ereditare solo da interfacce visibili, ossia interfacce che siano pubbliche oppure collocate nello stesso spazio dell'oggetto in questione.
- Un'interfaccia pubblica non può estenderne una privata.
- Un'interfaccia non può estendere se stessa. In generale non sono ammesse ereditarietà cicliche.
- Il nome dei membri deve essere univoco all'interno dei tipi, considerando anche eventuali metodi ereditati.

- I campi, le costanti e i parametri hanno un tipo associato, che ne determina il dominio, mentre i valori delle enumerazioni sono considerati essi stessi dati.
- Le funzioni definiscono sempre un tipo restituito, eventualmente Void.
- Il nome dei parametri deve essere univoco all'interno delle funzioni.
- L'ordine dei parametri è rilevante.
- Una funzione pubblica appartenente ad un tipo pubblico richiede che il tipo restituito e i tipi dei suoi parametri siano anch'essi pubblici.
- Solo gli sviluppatori possono essere segnati come autori di un blocco di documentazione.
- Il sito necessita di pagine aggiuntive per l'accesso e la gestione del sistema. In particolare è richiesta una pagina di login, una per la modifica della password, una per la gestione e l'inserimento degli utenti (ad uso esclusivo degli amministratori) e infine una per l'assegnazione degli spazi agli sviluppatori (anch'essa riservata agli amministratori).

Riassunte di seguito sono le principali azioni che deve consentire il sito:

1. Mostrare la visuale pubblica dell'elenco degli spazi
2. Mostrare la visuale pubblica di uno spazio
3. Mostrare la visuale pubblica di un tipo
4. Mostrare la visuale interna dell'elenco degli spazi
5. Mostrare la visuale interna di uno spazio
6. Mostrare la visuale interna di un tipo, elencando i membri in ordine alfabetico
7. Mostrare la visuale interna di un tipo, ordinando i membri in base al numero di riferimenti
8. Mostrare la visuale interna di un membro
9. Mostrare la visuale interna di un parametro
10. Inserire uno spazio
11. Rimuovere uno spazio
12. Inserire un tipo
13. Rimuovere un tipo
14. Aggiungere un'interfaccia da estendere / implementare
15. Rimuovere un'interfaccia estesa / implementata
16. Inserire un membro
17. Rimuovere un membro
18. Inserire un parametro
19. Rimuovere un parametro
20. Aggiungere un blocco di documentazione

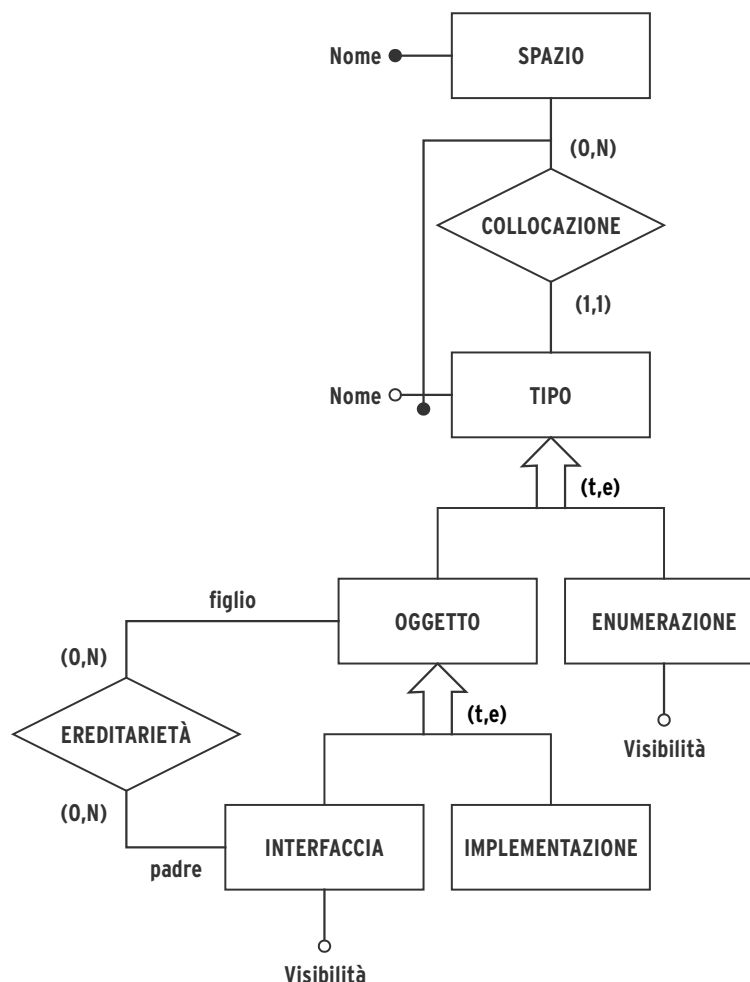
21. Modificare la documentazione di un elemento
22. Aggiungere un autore
23. Rimuovere un autore
24. Visualizzare l'elenco degli utenti
25. Inserire un utente
26. Rimuovere un utente
27. Visualizzare l'elenco degli spazi e degli sviluppatori
28. Selezionato uno spazio, visualizzare gli sviluppatori assegnati e quelli assegnabili
29. Selezionato uno sviluppatore, visualizzare gli spazi assegnati e quelli assegnabili
30. Assegnare uno spazio ad uno sviluppatore
31. Rimuovere un'assegnazione
32. Accedere come utente
33. Modificare la password

Progettazione concettuale

Schemi parziali

1.

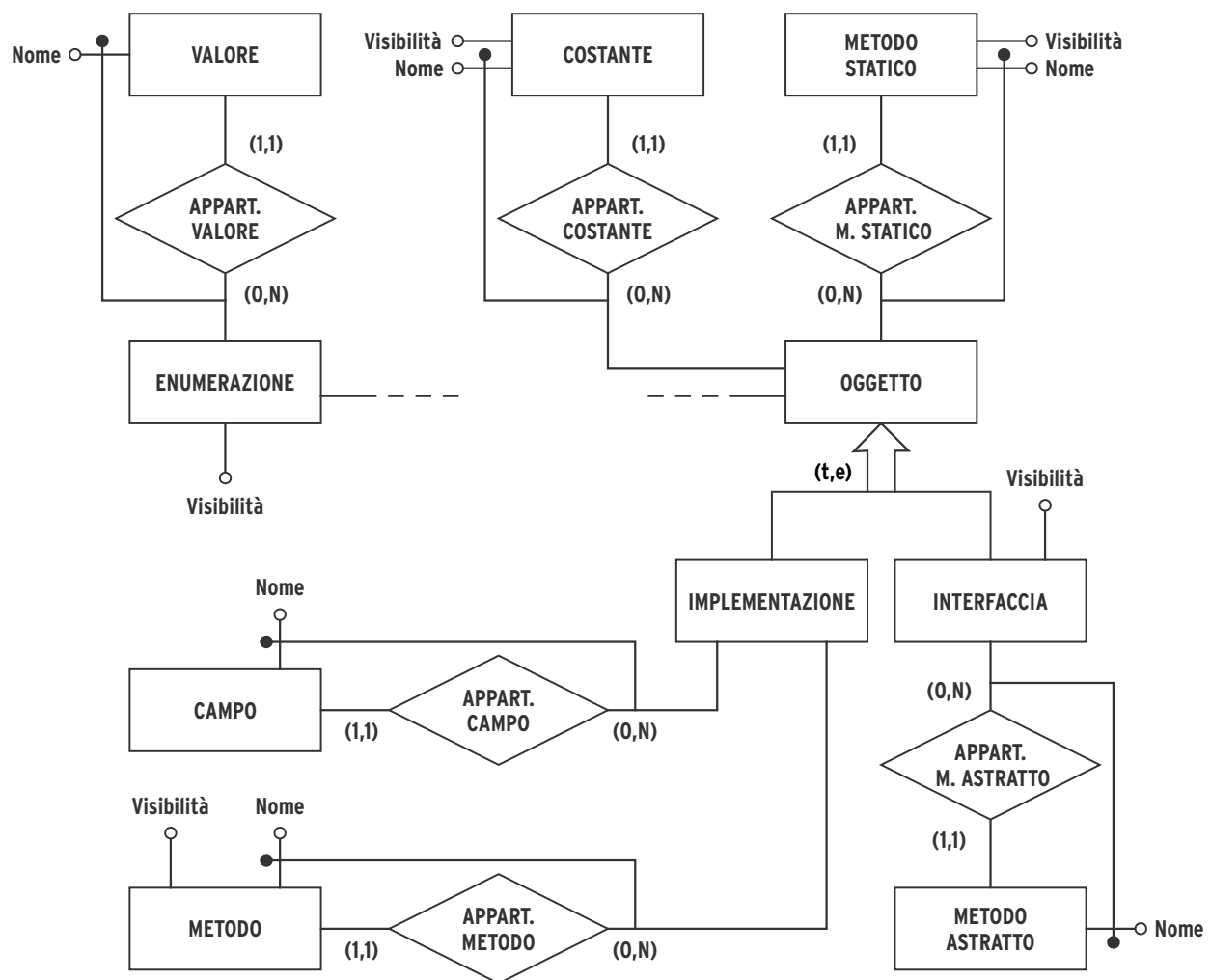
In base alle informazioni raccolte si decide di introdurre l'entità OGGETTO come generalizzazione delle entità INTERFACCIA e IMPLEMENTAZIONE, in aggiunta all'entità TIPO che raccoglie le entità OGGETTO ed ENUMERAZIONE. Il concetto di ereditarietà è dunque realizzato con un'associazione asimmetrica tra un OGGETTO figlio e un'INTERFACCIA padre. Come descritto in fase di analisi, un TIPO è identificato dal proprio Nome insieme allo SPAZIO in cui è collocato.



2.

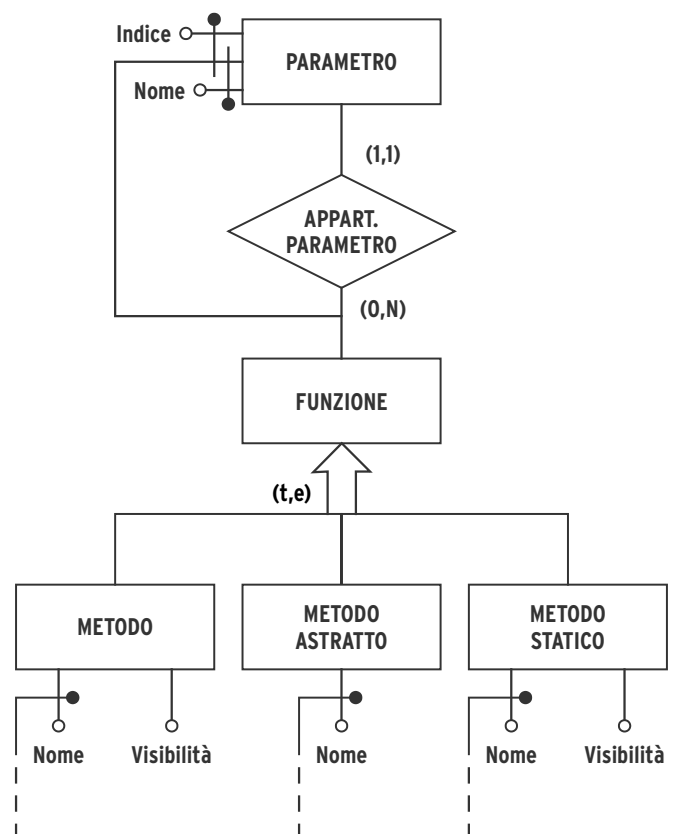
Si decide di creare un'entità distinta per ogni categoria di membro, in modo da poter esplicitare quali categorie di membro possano essere contenute in quali categorie di TIPO.

Tutti i membri sono identificati dal proprio Nome e dal TIPO a cui appartengono.



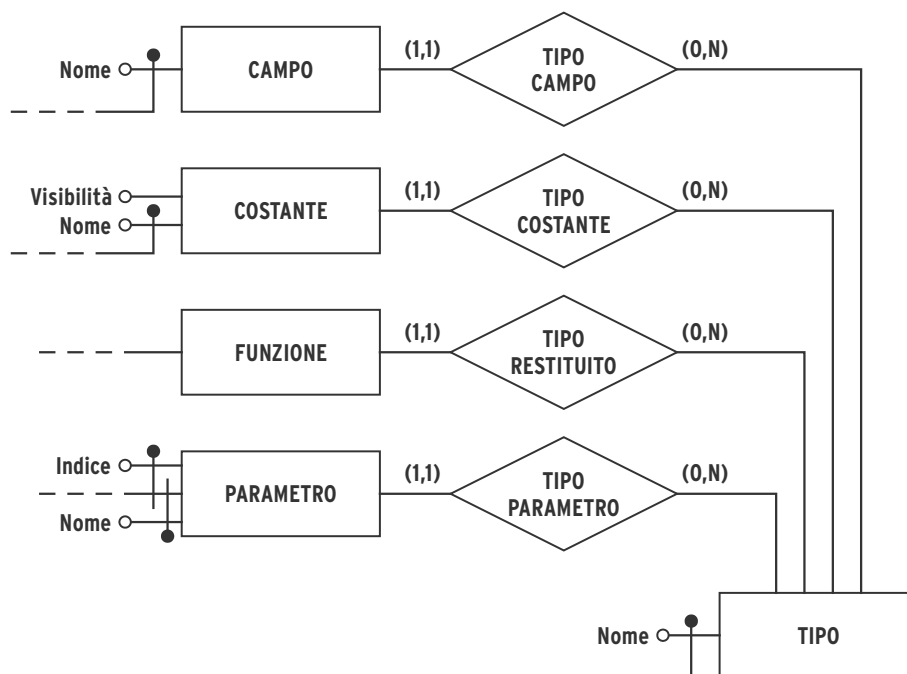
3.

Si decide però di introdurre l'entità **FUNZIONE**, in modo da raccogliere assieme le categorie di membro che possono avere dei parametri. Ogni **PARAMETRO** è identificato dal proprio **Nome** e dalla **FUNZIONE** a cui appartiene, ma anche dal proprio **Indice** di ordinamento all'interno della **FUNZIONE** stessa.



4.

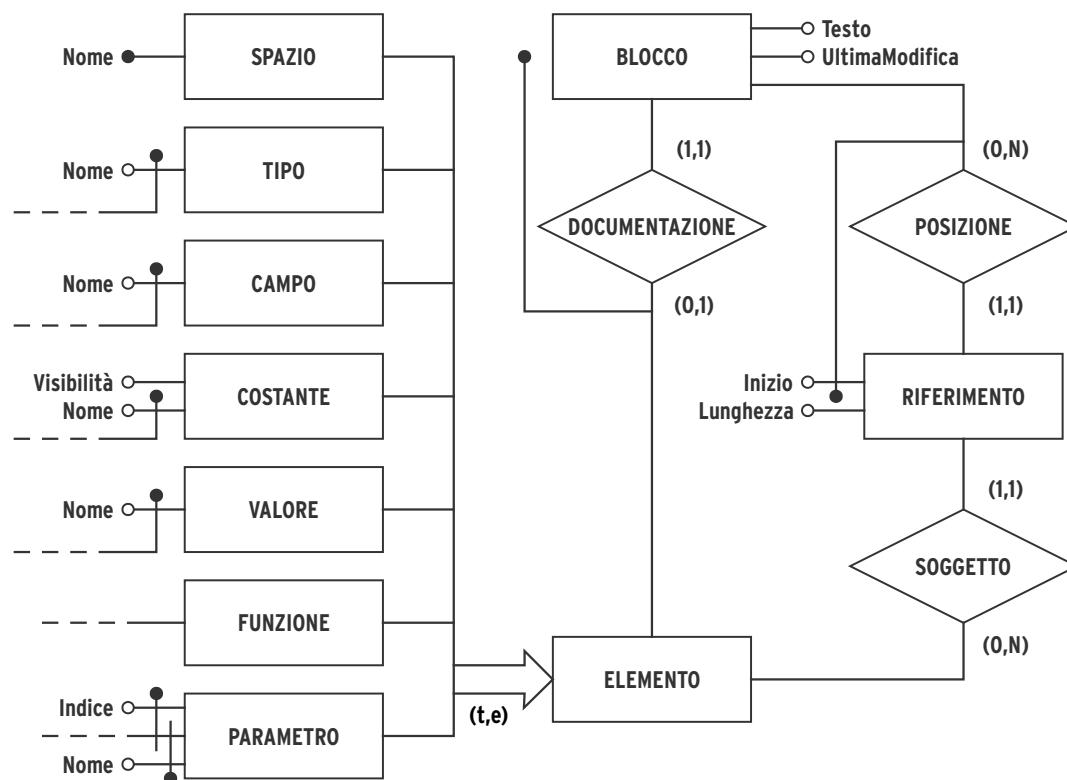
Ogni istanza delle entità CAMPO, COSTANTE e PARAMETRO dovrà essere associata al TIPO che ne costituisce il dominio. Analogamente, ogni FUNZIONE dovrà specificare il TIPO restituito.



5.

Si introduce l'entità ELEMENTO per contenere tutte le istanze che possono essere documentate. Questo permetterà di associare ad ogni ELEMENTO un BLOCCO di documentazione, il quale sarà identificato dall'ELEMENTO stesso. Una porzione di testo potrà essere arricchita con un RIFERIMENTO, del quale verrà memorizzata la posizione iniziale e la lunghezza, insieme all'ELEMENTO a cui si vuole rimandare.

Il compito di verificare che i vari riferimenti non si sovrappongano è lasciato alla logica dell'applicazione, mentre a livello concettuale si sceglie di identificare un RIFERIMENTO per mezzo del blocco di appartenenza e della posizione d'Inizio.



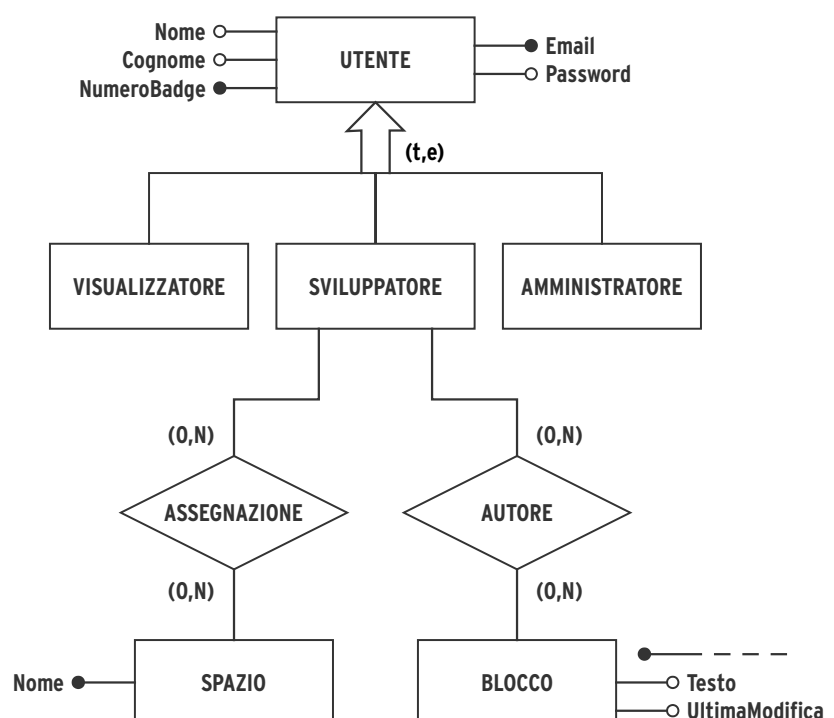
6.

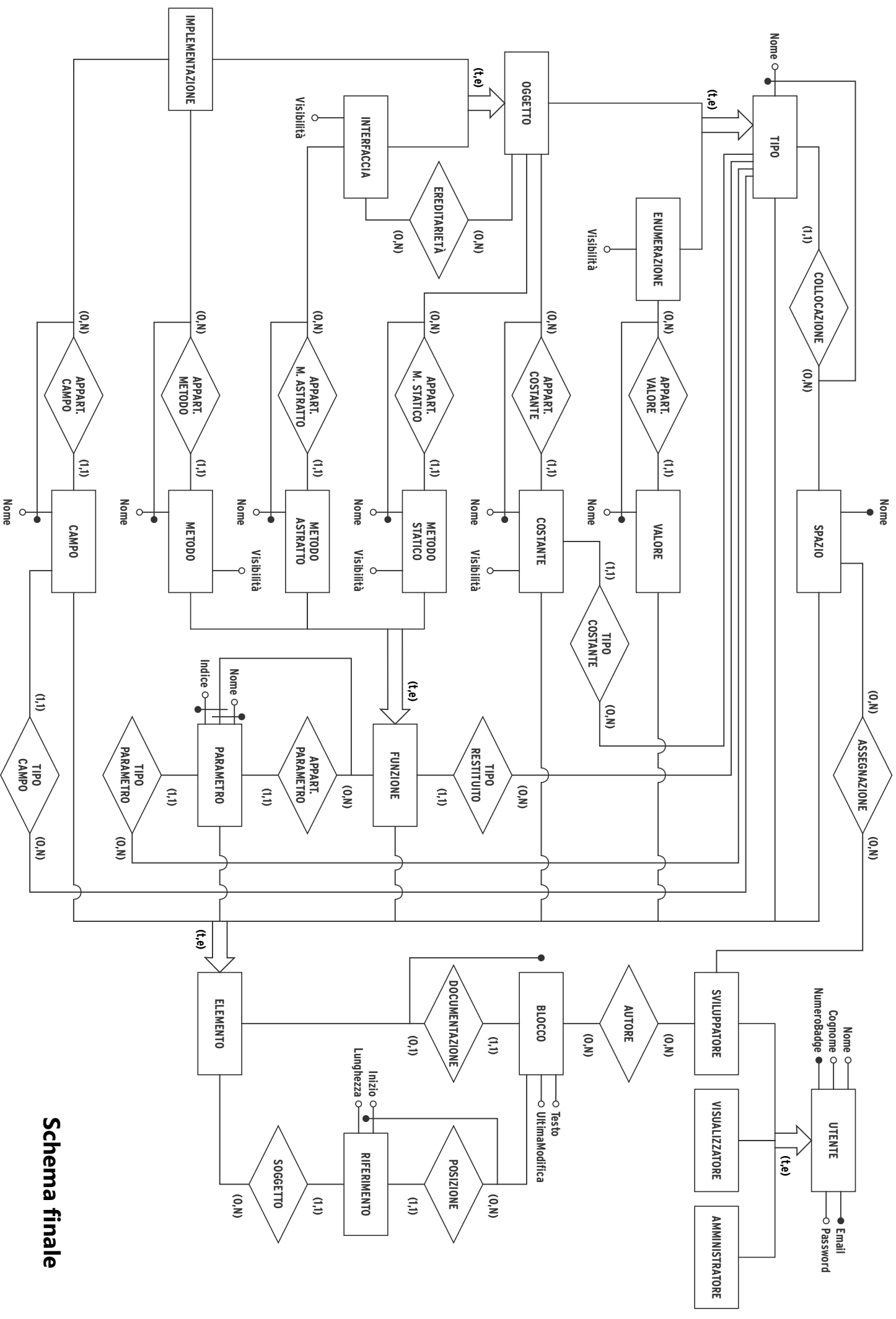
Per ogni UTENTE sono memorizzate le informazioni richieste.

Si deduce dai requisiti che sia l'indirizzo email aziendale che il numero badge identifichino univocamente un UTENTE.

Fra le tipologie di UTENTE specificate, solo quelli appartenenti all'entità SVILUPPATORE possono essere assegnati agli spazi.

Similmente gli autori elencati nei vari di blocchi di documentazione possono essere solo sviluppatori.





Schema finale

Progettazione logica

Stima del volume dei dati

Concetto	Costrutto	Volume
SPAZIO	E	50
INTERFACCIA	E	2.000
IMPLEMENTAZIONE	E	3.000
ENUMERAZIONE	E	500
COLLOCAZIONE	R	5.500
EREDITARIETÀ	R	5.000
VALORE	E	2.500
APPARTENENZA VALORE	R	2.500
COSTANTE	E	3.000
APPARTENENZA COSTANTE	R	3.000
TIPO COSTANTE	R	3.000
CAMPO	E	15.000
APPARTENENZA CAMPO	R	15.000
TIPO CAMPO	R	15.000
METODO STATICO	E	10.000
APPARTENENZA METODO STATICO	R	10.000
METODO ASTRATTO	E	15.000
APPARTENENZA METODO ASTRATTO	R	15.000
METODO	E	20.000
APPARTENENZA METODO	R	20.000
TIPO RESTITUITO	R	45.000
PARAMETRO	E	150.000
APPARTENENZA PARAMETRO	R	150.000
TIPO PARAMETRO	R	150.000
BLOCCO	E	200.000
DOCUMENTAZIONE	R	200.000

RIFERIMENTO	E	250.000
POSIZIONE	R	250.000
SOGGETTO	R	250.000
VISUALIZZATORE	E	10
SVILUPPATORE	E	20
AMMINISTRATORE	E	5
ASSEGNAZIONE	R	40
AUTORE	R	5.000

Stima della frequenza delle operazioni principali

Inizialmente si esaminano le azioni descritte in fase di analisi.

n°	Azione	Frequenza
1	Visuale pubblica dell'elenco degli spazi	1000 al giorno
2	Visuale pubblica di uno spazio	1500 al giorno
3	Visuale pubblica di un tipo	3000 al giorno
4	Visuale interna dell'elenco degli spazi	100 al giorno
5	Visuale interna di uno spazio	200 al giorno
6	Visuale interna tipo, ordine alfabetico	400 al giorno
7	Visuale interna tipo, ordinando per n. di riferimenti	100 al giorno
8	Visuale interna di un membro	800 al giorno
9	Visuale interna di un parametro	1200 al giorno
10	Inserire uno spazio	3 all'anno
11	Rimuovere uno spazio	1 all'anno
12	Inserire un tipo	10 al mese
13	Rimuovere un tipo	5 al mese
14	Aggiungere interfaccia da estendere / implementare	2 al giorno
15	Rimuovere interfaccia estesa / implementata	10 al mese
16	Inserire un membro	100 al giorno
17	Rimuovere un membro	5 al giorno

18	Inserire un parametro	300 al giorno
19	Rimuovere un parametro	15 al giorno
20	Aggiungere un blocco di documentazione	800 al giorno
21	Modificare la documentazione di un elemento	1200 al giorno
22	Aggiungere un autore	20 al mese
23	Rimuovere un autore	1 al mese
24	Visualizzare l'elenco degli utenti	10 al giorno
25	Inserire un utente	3 all'anno
26	Rimuovere un utente	1 all'anno
27	Elenco degli spazi e degli sviluppatori	4 al giorno
28	Sviluppatori assegnati e assegnabili	2 al giorno
29	Spazi assegnati e assegnabili	2 al giorno
30	Assegnare uno spazio ad uno sviluppatore	15 al mese
31	Rimuovere un'assegnazione	10 al mese
32	Accedere come utente	150 al giorno
33	Modificare la password	70 all'anno

Trattandosi di un sito web, queste azioni non sono atomiche (ad esempio inserire o rimuovere un utente provocherà l'aggiornamento della pagina, il che richiederà di accedere nuovamente all'elenco degli utenti) perciò si decide di definire alcune operazioni elementari.

OP1	Visualizzazione della documentazione di un elemento
OP2	Elenco degli spazi
OP3	Verifica dell'esistenza di uno spazio
OP4	Elenco dei tipi pubblici in uno spazio
OP5	Verifica dell'esistenza e della visibilità di un tipo
OP6	Elenco delle interfacce estese da un tipo
OP7	Elenco delle interfacce pubbliche derivate da un'interfaccia
OP8	Elenco dei membri pubblici di un tipo
OP9	Elenco dei metodi ereditati da un tipo

OP10	Elenco dei parametri di una funzione
OP11	Elenco degli autori di un elemento
OP12	Elenco degli sviluppatori che non sono già autori di un elemento
OP13	Aggiunta di un autore
OP14	Rimozione di un autore
OP15	Inserimento di un blocco vuoto di documentazione
OP16	Modifica della documentazione di un elemento
OP17	Elenco dei riferimenti ad un elemento
OP18	Inserimento di uno spazio
OP19	Rimozione di uno spazio
OP20	Verifica dei permessi di modifica all'interno di uno spazio
OP21	Elenco dei tipi in uno spazio
OP22	Inserimento di un tipo
OP23	Rimozione di un tipo
OP24	Accesso ai dettagli di un tipo
OP25	Elenco dei tipi derivati da un'interfaccia
OP26	Aggiunta di un'interfaccia da estendere / implementare
OP27	Rimozione di un'interfaccia estesa / implementata
OP28	Elenco dei membri di un tipo in ordine alfabetico
OP29	Elenco dei membri ordinati per numero di riferimenti
OP30	Inserimento di un membro
OP31	Rimozione di un membro
OP32	Accesso ai dettagli di un membro
OP33	Inserimento di un parametro
OP34	Rimozione di un parametro
OP35	Accesso ai dettagli di un parametro
OP36	Elenco degli utenti
OP37	Inserimento di un utente
OP38	Rimozione di un utente
OP39	Elenco degli spazi e degli sviluppatori

OP40	Elenco degli sviluppatori assegnati e assegnabili
OP41	Elenco degli spazi assegnati e assegnabili
OP42	Assegnazione di uno spazio ad uno sviluppatore
OP43	Rimozione di un'assegnazione
OP44	Accesso come utente
OP45	Modifica di una password

Si esaminano ora le operazioni necessarie per svolgere ciascuna delle azioni richieste. Le quantità numeriche riportate sono stime basate sull'analisi del volume dei dati.

Si utilizzerà l'abbreviazione DOC per indicare la seguente espressione:

$$\text{DOC} = \text{OP1} + 0,9 \cdot (\text{OP11} + 0,2 \cdot \text{OP12}) + \text{OP17}$$

Nota: circa il 90% degli elementi sono documentati e circa il 20% degli utenti sono amministratori.

Azione	Operazioni coinvolte
A1	$\text{OP2} + 50 \cdot \text{OP1}$
A2	$\text{OP3} + \text{OP4} + 35 \cdot \text{OP1}$ In media uno spazio contiene circa 34 tipi pubblici
A3	$\text{OP5} + 0,8 \cdot (\text{OP6} + \text{OP7}) + \text{OP8} + \text{OP9} + 10 \cdot \text{OP10} + 45 \cdot \text{OP1}$ Circa l'80% dei tipi pubblici sono interfacce, in media un tipo pubblico ha circa 10 funzioni pubbliche e in media una funzione ha circa 3,5 parametri
A4	OP2
A5	$\text{OP3} + 0,6 \cdot \text{OP20} + \text{OP21} + \text{DOC}$ Circa il 60% degli utenti sono sviluppatori
A6	$0,6 \cdot \text{OP20} + \text{OP24} + 0,9 \cdot \text{OP6} + 0,4 \cdot \text{OP25} + \text{OP28} + \text{DOC}$ Circa il 90% dei tipi sono oggetti, circa il 40% dei tipi sono interfacce
A7	$0,6 \cdot \text{OP20} + \text{OP24} + 0,9 \cdot \text{OP6} + 0,4 \cdot \text{OP25} + \text{OP29} + \text{DOC}$
A8	$0,6 \cdot \text{OP20} + \text{OP24} + \text{OP32} + 0,7 \cdot \text{OP10} + \text{DOC}$ Circa il 70% dei membri sono funzioni
A9	$0,6 \cdot \text{OP20} + \text{OP35} + \text{DOC}$

A10	OP18 + A4
A11	OP19 + A4
A12	OP22 + A5
A13	OP23 + A5
A14	OP26 + 0,8*A6 + 0,2*A7 Circa l'80% delle volte l'utente sceglie di utilizzare l'ordine alfabetico
A15	OP27 + 0,8*A6 + 0,2*A7
A16	OP30 + 0,8*A6 + 0,2*A7
A17	OP31 + 0,8*A6 + 0,2*A7
A18	OP33 + A8
A19	OP34 + A8
A20	OP15 + 0,3*A8 + 0,7*A9 Circa il 30% degli elementi sono membri e circa il 70% sono parametri, mentre spazi e tipi risultano trascurabili
A21	OP16 + 0,3*A8 + 0,7*A9
A22	OP13 + 0,3*A8 + 0,7*A9
A23	OP14 + 0,3*A8 + 0,7*A9
A24	OP36
A25	OP37 + A24
A26	OP38 + A24
A27	OP39
A28	OP40
A29	OP41
A30	OP42 + 0,5*A28 + 0,5*A29 Si prevede che vengano utilizzate con la stessa frequenza le funzioni di selezione di uno spazio e di selezione di uno sviluppatore
A31	OP43 + 0,5*A28 + 0,5*A29
A32	OP44 + A4
A33	OP45 + A4

Da queste osservazioni si può stimare la frequenza con cui verranno invocate le operazioni di base.

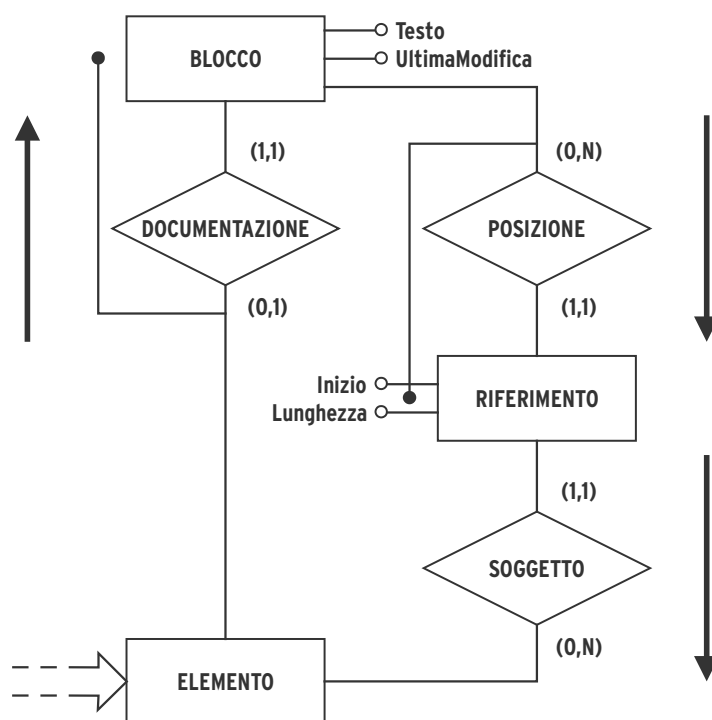
Operazione	Frequenza	Operazione	Frequenza
OP1	250.000 al giorno	OP24	2300 al giorno
OP2	1200 al giorno	OP25	250 al giorno
OP3	1700 al giorno	OP26	2 al giorno
OP4	1500 al giorno	OP27	10 al mese
OP5	3000 al giorno	OP28	500 al giorno
OP6	3000 al giorno	OP29	100 al giorno
OP7	2400 al giorno	OP30	100 al giorno
OP8	3000 al giorno	OP31	5 al giorno
OP9	3000 al giorno	OP32	1700 al giorno
OP10	30.000 al giorno	OP33	300 al giorno
OP11	4600 al giorno	OP34	15 al giorno
OP12	1000 al giorno	OP35	2600 al giorno
OP13	20 al mese	OP36	10 al giorno
OP14	1 al mese	OP37	3 all'anno
OP15	800 al giorno	OP38	1 all'anno
OP16	1200 al giorno	OP39	4 al giorno
OP17	5000 al giorno	OP40	2/3 al giorno
OP18	3 all'anno	OP41	2/3 al giorno
OP19	1 all'anno	OP42	15 al mese
OP20	1600 al giorno	OP43	10 al mese
OP21	200 al giorno	OP44	150 al giorno
OP22	10 al mese	OP45	70 all'anno
OP23	5 al mese		

Schemi di navigazione e tabelle degli accessi

Si illustrano gli schemi di navigazione e le tabelle degli accessi delle operazioni più complesse, utilizzando la convenzione 1 S = 2 L.

OP1 (Visualizzazione della documentazione di un elemento)

Dato un elemento è necessario verificare se esiste documentazione associata. Circa il 90% delle volte esiste, per cui si procede leggendo il blocco. Si trova poi che in media un blocco contiene circa 1,25 riferimenti. Infine si conclude l'operazione leggendo gli elementi a cui rimandano i riferimenti.



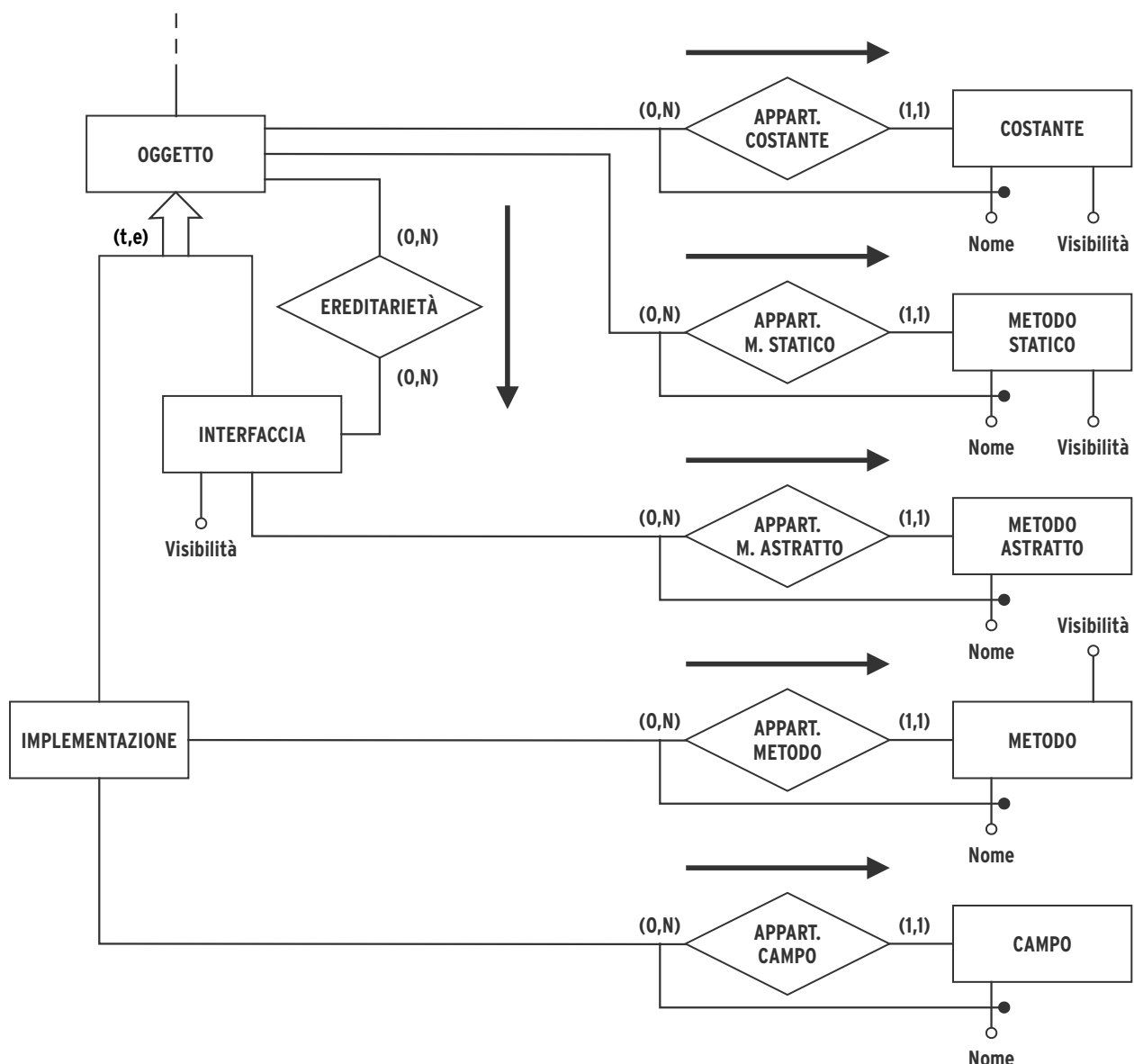
Concetto	Costrutto	Accessi	Tipo
ELEMENTO	E	1	L
DOCUMENTAZIONE	R	0,9	L
BLOCCO	E	0,9	L
POSIZIONE	R	1,125	L
RIFERIMENTO	E	1,125	L
SOGGETTO	R	1,125	L
ELEMENTO	E	1,125	L
Accessi totali: 7,3 L → ~1.800.000 al giorno			

OP26 (Aggiunta di un'interfaccia da estendere / implementare)

Il primo passo è selezionare l'oggetto attuale e l'interfaccia da aggiungere. Se l'oggetto attuale è anch'esso un'interfaccia (circa il 40% lo sono), è necessario verificare che non si creino ereditarietà cicliche. Si ipotizza per semplicità che in media questo controllo comporti 2 accessi ai costrutti EREDITARIETÀ e INTERFACCIA.

Si procede poi con il verificare che non ci siano collisioni fra i nomi dei membri dell'oggetto attuale e i nomi dei metodi che verrebbero ereditati dall'interfaccia. Si utilizza il fatto che in media un oggetto contiene 3 campi, 3 metodi astratti, 2 metodi statici e 0,6 costanti. Inoltre i dati suggeriscono che un oggetto contenga in media 4 metodi e che un'interfaccia contenga in media 7,5 metodi astratti, ma si sceglie di incrementare questi valori del 50% per stimare l'impatto dei metodi ereditati.

Infine si procede con l'inserimento dell'ereditarietà.



Concetto	Costrutto	Accessi	Tipo
OGGETTO	E	1	L
INTERFACCIA	E	1	L
EREDITARIETÀ	R	0,8	L
INTERFACCIA	E	0,8	L
APPARTENENZA CAMPO	R	3	L
CAMPO	E	3	L
APPARTENENZA METODO ASTRATTO	R	3	L
METODO ASTRATTO	E	3	L
APPARTENENZA METODO STATICO	R	2	L
METODO STATICO	E	2	L
APPARTENENZA COSTANTE	R	0,6	L
COSTANTE	E	0,6	L
APPARTENENZA METODO	R	6	L
METODO	E	6	L
APPARTENENZA METODO ASTRATTO	R	11	L
METODO ASTRATTO	E	11	L
EREDITARIETÀ	R	1	S
Accessi totali: 55 L + 1 S → ~115 al giorno			

Raffinamento dello schema

Eliminazione delle gerarchie

Poiché lo schema concettuale di partenza fa ampio uso di generalizzazioni, questa fase richiederà modifiche sostanziali all'intera struttura.

Per le entità INTERFACCIA, IMPLEMENTAZIONE ed ENUMERAZIONE si sceglie un collasso verso l'alto nell'entità TIPO, per via del fatto che quasi sempre si accede a tutte e tre le categorie contemporaneamente attraverso l'entità padre.

Per evitare di complicare eccessivamente lo schema si ipotizza che nella fase di progettazione concettuale fosse stata introdotta una gerarchia che riunisse tutti i membri all'interno di un'entità unica. In effetti ciò sarebbe stato possibile e anche corretto a livello intuitivo, ma non avrebbe portato alcun vantaggio pratico. A questo punto, invece, diventa vantaggioso optare per un collasso verso l'alto nell'entità MEMBRO, per motivi analoghi alla generalizzazione dei tipi. Ciò comporta che anche le associazioni APPARTENENZA VALORE, APPARTENENZA METODO, ... vengano riunite in un'unica associazione APPARTENENZA MEMBRO, così come le associazioni TIPO VALORE, TIPO METODO, ..., che verranno riunite nell'associazione TIPO MEMBRO.

Lo svantaggio di questo approccio sta nel fatto che gran parte dei vincoli che erano precedentemente espressi dalla struttura stessa dello schema, dovranno essere implementati nel codice dell'applicazione.

Procedendo oltre, si sceglie invece di non riunire ulteriormente i concetti all'interno dell'entità ELEMENTO, poiché in questo caso gli accessi alle varie tipologie sono ben distinti. Si procede dunque con un collasso verso il basso, replicando l'associazione DOCUMENTAZIONE per ciascuna delle entità SPAZIO, TIPO, MEMBRO e PARAMETRO.

Per quanto riguarda gli utenti si sceglie ancora un collasso verso l'alto, poiché anche in questo caso risulta più comodo accedere a tutte le tipologie in contemporanea in fase di gestione.

Scelta delle chiavi primarie

Le ambiguità principali si trovano nelle entità PARAMETRO, per la quale si sceglie di usare il Nome come chiave primaria, e UTENTE, per la quale si sceglie l'indirizzo Email, in entrambi i casi perché questi identificatori sono ritenuti più descrittivi.

Per quanto riguarda l'entità BLOCCO si introduce un Codice identificativo, per ovviare al problema dell'eliminazione dell'entità ELEMENTO.

Eliminazione di identificatori esterni

- Nell'entità TIPO si sostituisce l'identificatore esterno importando il Nome dello SPAZIO, rinominato semplicemente in Spazio.
- Nell'entità MEMBRO si sostituisce l'identificatore esterno importando il Nome e lo Spazio del TIPO di appartenenza, rinominati in TipoPadre e SpazioPadre.
- Nell'entità PARAMETRO si sostituisce l'identificatore esterno importando il Nome, il TipoPadre e lo SpazioPadre del MEMBRO di appartenenza, rinominati in MembroFunzione, TipoFunzione e SpazioFunzione.
- Nell'entità RIFERIMENTO si sostituisce l'identificatore esterno importando il Codice del BLOCCO, rinominato in Blocco.

Si eliminano le associazioni *uno a molti* banali:

- COLLOCAZIONE, APPARTENENZA MEMBRO, APPARTENENZA PARAMETRO e POSIZIONE per mezzo degli attributi descritti sopra.
- TIPO MEMBRO, importando nell'entità MEMBRO il Nome e lo Spazio del TIPO, rinominati in Tipo e Spazio.
- TIPO PARAMETRO, importando nell'entità PARAMETRO il Nome e lo Spazio del TIPO, rinominati in Tipo e Spazio.

Si reificano le associazioni *molti a molti*:

- EREDITARIETÀ, importando il Nome e lo Spazio del TIPO padre, rinominati in TipoPadre e SpazioPadre, e il Nome e lo Spazio del TIPO figlio, rinominati in TipoFiglio e SpazioFiglio.
- ASSEGNAZIONE, importando il Nome dello SPAZIO, rinominato in Spazio, e l'indirizzo Email dell'UTENTE, rinominato in Sviluppatore.
- AUTORE, importando il Codice del BLOCCO, rinominato in Blocco, e l'indirizzo Email dell'UTENTE, rinominato in Sviluppatore.

Richiedono particolare attenzione le associazioni DOCUMENTAZIONE e SOGGETTO, per le quali si sceglie un approccio insolito ma efficace. Non si tratterebbe nemmeno di associazioni singole, bensì di associazioni ripetute per ciascuna delle entità SPAZIO, TIPO, MEMBRO e PARAMETRO. Tuttavia si decide di usare a proprio vantaggio il fatto che queste ultime sono legate da una catena di composizioni, il che permette di limitare la complessità del modello logico. Sia nell'entità BLOCCO che nell'entità RIFERIMENTO si introducono gli attributi Spazio, Tipo, Membro e Parametro, richiedendo che:

- Quando si necessita di una chiave esterna che punti ad uno SPAZIO, venga riempito l'attributo Spazio e vengano lasciati vuoti gli altri tre.
- Quando si necessita di una chiave esterna che punti ad un TIPO, vengano riempiti gli attributi Spazio e Tipo, lasciando vuoti gli altri due, in modo da importare il Nome e lo Spazio che identificano il TIPO desiderato.
- Così via per membri e parametri.

Analisi delle ridondanze

Sorge spontaneo chiedersi qualora sia vantaggioso inserire per ciascun membro un attributo che tenga traccia del numero di riferimenti ad esso. Si analizzano pertanto le operazioni che sarebbero coinvolte da questa modifica.

OP29 (Elenco dei membri ordinati per numero di riferimenti)

In questo caso la modifica permetterebbe di risparmiare l'accesso all'entità RIFERIMENTO.

Con ridondanza:

Concetto	Costrutto	Accessi	Tipo
TIPO	E	1	L
APPARTENENZA MEMBRO	R	12	L
MEMBRO	E	12	L
Accessi totali: 25 L → ~2500 al giorno			

(Si è sfruttato il fatto che in media un tipo ha in media 12 membri)

Senza ridondanza:

Concetto	Costrutto	Accessi	Tipo
TIPO	E	1	L
APPARTENENZA MEMBRO	R	12	L
MEMBRO	E	12	L
SOGGETTO	R	14	L
RIFERIMENTO	E	14	L
Accessi totali: 53 L → ~5300 al giorno			

(Si è sfruttato il fatto che in media ci sono 1,1 riferimenti ad un elemento)

OP16 (Modifica della documentazione di un elemento)

In questo caso, invece, la ridondanza introduce la necessità di aggiornare il valore dell'attributo, qualora si verificano determinate condizioni.

Si ipotizzi che ogni volta che viene modificato un blocco di documentazione vengano apportate in media 3 modifiche ai riferimenti al suo interno, sotto forma di aggiunte, rimozioni, o sostituzioni di un elemento soggetto. Poiché circa il 30% degli elementi sono membri, otteniamo una frequenza approssimativa di 1000 modifiche al numero di riferimenti a un membro al giorno. Ciascuna di queste modifiche introduce i seguenti accessi aggiuntivi:

Concetto	Costrutto	Accessi	Tipo
SOGGETTO	R	1	L
ELEMENTO	E	1	L
ELEMENTO	E	1	S
Accessi totali: 2 L + 1 S \rightarrow ~4000 al giorno			

Confrontando i risultati ottenuti dalle due operazioni, si trova che l'introduzione della ridondanza comporterebbe 1200 accessi aggiuntivi al giorno.

Per questo motivo si sceglie di scartare l'idea.

Traduzione di entità e associazioni in relazioni

UTENTI(Email, NumeroBadge, Nome, Cognome, Password, Ruolo)

SPAZI(Nome)

ASSEGNAZIONI(Spazio, Sviluppatore)

FK: Spazio REFERENCES Spazi

FK: Sviluppatore REFERENCES Utenti

TIPI(Nome, Spazio, Categoria, Visibilità)

FK: Spazio REFERENCES Spazi

EREDITARIETÀ(TipoPadre, SpazioPadre, TipoFiglio, SpazioFiglio)

FK: TipoPadre, SpazioPadre REFERENCES Tipi

FK: TipoFiglio, SpazioFiglio REFERENCES Tipi

MEMBRI(Nome, TipoPadre, SpazioPadre, Categoria, Visibilità, Tipo, Spazio)

FK: TipoPadre, SpazioPadre REFERENCES Tipi

FK: Tipo, Spazio REFERENCES Tipi

PARAMETRI(Nome, MembroFunzione, TipoFunzione, SpazioFunzione, Indice, Tipo, Spazio)

FK: MembroFunzione, TipoFunzione, SpazioFunzione REFERENCES Membri

FK: Tipo, Spazio REFERENCES Tipi

UNIQUE(Indice, MembroFunzione, TipoFunzione, SpazioFunzione)

BLOCCHI(Codice, Testo, Spazio, Tipo*, Membro*, Parametro*, UltimaModifica)

FK: Spazio REFERENCES Spazi

FK: Spazio, Tipo REFERENCES Tipi

FK: Spazio, Tipo, Membro REFERENCES Membri

FK: Spazio, Tipo, Membro, Parametro REFERENCES Parametri

AUTORI(Blocco, Sviluppatore)

FK: Blocco REFERENCES Blocchi

FK: Sviluppatore REFERENCES Utenti

RIFERIMENTI(Blocco, Inizio, Lunghezza, Spazio, Tipo*, Membro*, Parametro*)

FK: Blocco REFERENCES Blocchi

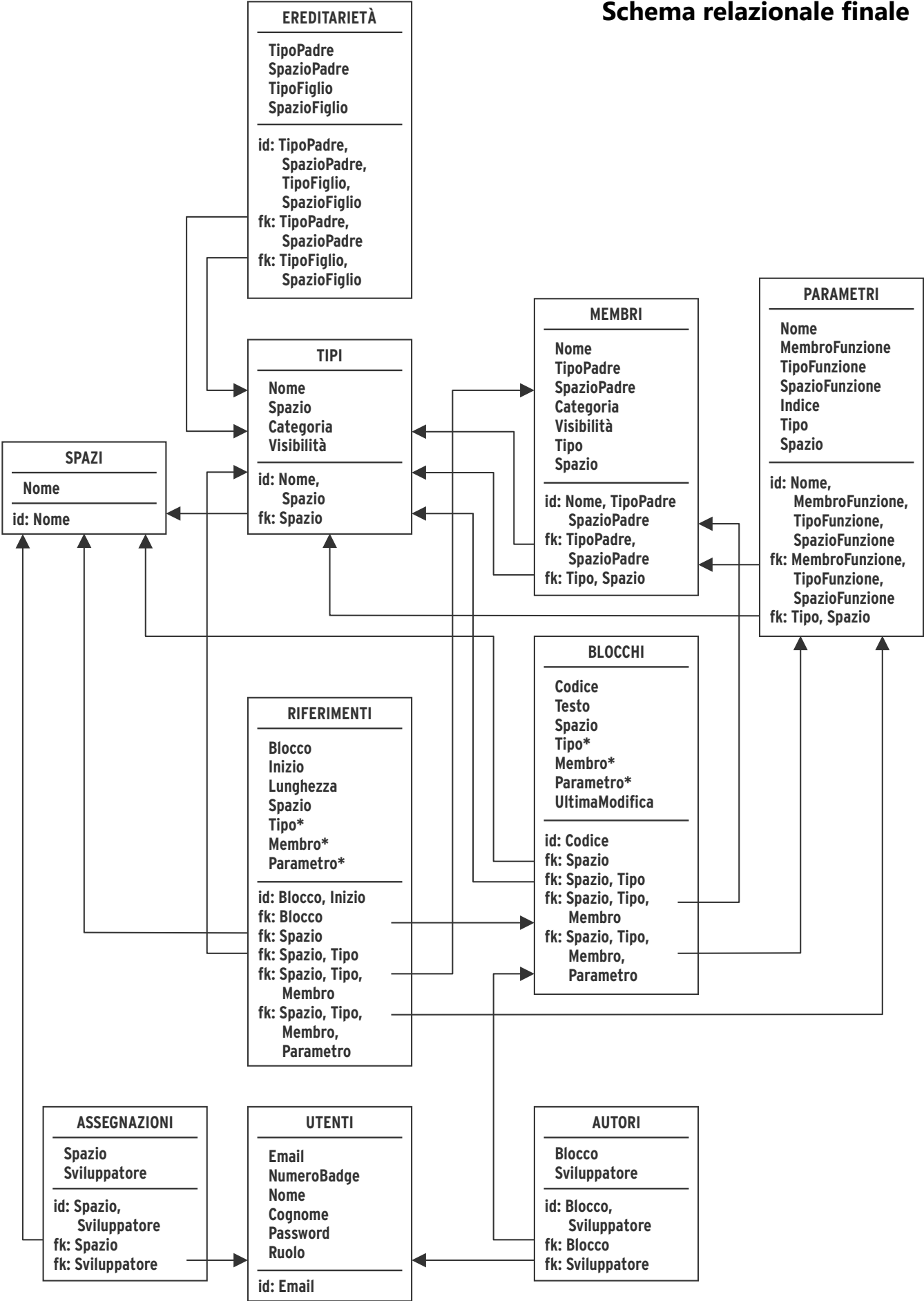
FK: Spazio REFERENCES Spazi

FK: Spazio, Tipo REFERENCES Tipi

FK: Spazio, Tipo, Membro REFERENCES Membri

FK: Spazio, Tipo, Membro, Parametro REFERENCES Parametri

Schema relazionale finale



Traduzione delle operazioni in query SQL

Viste utilizzate:

```
CREATE VIEW Avi AS
WITH RECURSIVE AviTmp(TipoFiglio, SpazioFiglio, TipoAvo, SpazioAvo)
AS (
    SELECT TipoFiglio, SpazioFiglio, TipoPadre, SpazioPadre
        FROM Ereditarietà
    UNION ALL
    SELECT E.TipoFiglio, E.SpazioFiglio, A.TipoAvo, A.SpazioAvo
        FROM Ereditarietà AS E, AviTmp AS A
        WHERE E.SpazioPadre = A.SpazioFiglio
        AND E.TipoPadre = A.TipoFiglio
)
SELECT *
    FROM AviTmp;
```

```
CREATE VIEW MetodiEreditati AS
SELECT *
    FROM Avi, Membri
    WHERE SpazioPadre = SpazioAvo
    AND TipoPadre = TipoAvo
    AND Categoria = 'MetodoAstratto';
```

OP1 (Visualizzazione della documentazione di un elemento)

```
SELECT Codice, Testo
    FROM Blocchi
    WHERE Spazio = ? AND Tipo = ? AND Membro = ? AND Parametro = ?;
```

Una volta verificata l'esistenza:

```
SELECT Inizio, Lunghezza, Spazio, Tipo, Membro, Parametro
    FROM Riferimenti
    WHERE Blocco = ?;
```

OP2 (Elenco degli spazi)

```
SELECT Nome FROM Spazi;
```

OP3 (Verifica dell'esistenza di uno spazio)

```
SELECT * FROM Spazi WHERE Nome = ?;
```

OP4 (Elenco dei tipi pubblici in uno spazio)

```
SELECT Nome, Categoria FROM Tipi
    WHERE Spazio = ? AND Pubblico = TRUE;
```

OP5 (Verifica dell'esistenza e della visibilità di un tipo)

```
SELECT Categoria FROM Tipi
WHERE Spazio = ? AND Nome = ?
AND Pubblico = TRUE;
```

OP6 (Elenco delle interfacce estese da un tipo)

```
SELECT TipoPadre, SpazioPadre FROM Ereditarietà
WHERE SpazioFiglio = ? AND TipoFiglio = ?;
```

OP7 (Elenco delle interfacce pubbliche derivate da un'interfaccia)

```
SELECT TipoFiglio, SpazioFiglio FROM Ereditarietà, Tipi
WHERE SpazioPadre = ? AND TipoPadre = ?
AND Spazio = SpazioFiglio AND Nome = TipoFiglio
AND Pubblico = TRUE;
```

OP8 (Elenco dei membri pubblici di un tipo)

```
SELECT Nome, Categoria, Tipo, Spazio FROM Membri
WHERE SpazioPadre = ? AND TipoPadre = ?
AND Pubblico = TRUE;
```

OP9 (Elenco dei metodi ereditati da un tipo)

```
SELECT Nome, TipoAvo, SpazioAvo, Tipo, Spazio FROM MetodiEreditati
WHERE SpazioFiglio = ? AND TipoFiglio = ?;
```

OP10 (Elenco dei parametri di una funzione)

```
SELECT Nome, Spazio, Tipo FROM Parametri
WHERE SpazioFunzione = ? AND TipoFunzione = ?
AND MembroFunzione = ?
ORDER BY Indice;
```

OP11 (Elenco degli autori di un elemento)

```
SELECT Sviluppatore FROM Autori WHERE Blocco = ?;
```

OP12 (Elenco degli sviluppatori che non sono già autori di un elemento)

```
SELECT Email FROM Utenti
WHERE Ruolo = 'Sviluppatore'
AND Email NOT IN
(SELECT Sviluppatore FROM Autori WHERE Blocco = ?);
```

OP13 (Aggiunta di un autore)

```
INSERT INTO Autori VALUES (?, ?);
```

OP14 (Rimozione di un autore)

```
DELETE FROM Autori  
WHERE Blocco = ? AND Sviluppatore = ?;
```

OP15 (Inserimento di un blocco vuoto di documentazione)

```
INSERT INTO Blocchi(Spazio, Tipo, Membro, Parametro)  
VALUES (?, ?, ?, ?);
```

OP16 (Modifica della documentazione di un elemento)

```
UPDATE Blocchi  
SET Testo = ?, UltimaModifica = CURRENT_DATE  
WHERE Codice = ?;  
DELETE FROM Riferimenti  
WHERE Blocco = ?;
```

Per ogni riferimento:

```
INSERT INTO Riferimenti VALUES (?, ?, ?, ?, ?, ?, ?);
```

OP17 (Elenco dei riferimenti ad un elemento)

```
SELECT B.Testo, R.Inizio, R.Lunghezza,  
       B.Spazio, B.Tipo, B.Membro, B.Parametro  
FROM Riferimenti AS R, Blocchi AS B  
WHERE B.Codice = R.Blocco  
AND R.Spazio = ? AND R.Tipo = ?  
AND R.Membro = ? AND R.Parametro = ?;
```

OP18 (Inserimento di uno spazio)

```
INSERT INTO Spazi VALUES (?);
```

OP19 (Rimozione di uno spazio)

```
DELETE FROM Spazi WHERE Nome = ?;
```

OP20 (Verifica dei permessi di modifica all'interno di uno spazio)

```
SELECT * FROM Assegnazioni  
WHERE Spazio = ? AND Sviluppatore = ?;
```

OP21 (Elenco dei tipi in uno spazio)

```
SELECT Nome, Categoria, Pubblico FROM Tipi  
WHERE Spazio = ?;
```

OP22 (Inserimento di un tipo)

```
INSERT INTO Tipi VALUES (?, ?, ?, ?);
```

OP23 (Rimozione di un tipo)

```
DELETE FROM Tipi WHERE Spazio = ? AND Nome = ?;
```

OP24 (Accesso ai dettagli di un tipo)

```
SELECT Categoria, Pubblico FROM Tipi  
WHERE Spazio = ? AND Nome = ?;
```

OP25 (Elenco dei tipi derivati da un'interfaccia)

```
SELECT TipoFiglio, SpazioFiglio FROM Ereditarietà  
WHERE SpazioPadre = ? AND TipoPadre = ?;
```

OP26 (Aggiunta di un'interfaccia da estendere / implementare)

```
SELECT Categoria, Pubblico FROM Tipi  
WHERE Spazio = ? AND Nome = ?;
```

Verifica dell'assenza di ereditarietà cicliche:

```
SELECT * FROM Avi  
WHERE SpazioAvo = ? AND TipoAvo = ?  
AND SpazioFiglio = ? AND TipoFiglio = ?;
```

Verifica dell'assenza di collisioni con i nomi dei metodi ereditati:

```
WITH TuttiMembri(Nome, Tipo, Spazio) AS (  
    SELECT Nome, TipoPadre, SpazioPadre FROM Membri  
    UNION  
    SELECT Nome, TipoFiglio, SpazioFiglio FROM MetodiEreditati  
)  
SELECT Nome FROM TuttiMembri  
WHERE Spazio = ? AND Tipo = ?  
AND Nome IN (  
    SELECT Nome FROM MetodiEreditati  
    WHERE SpazioFiglio = ? AND TipoFiglio = ?  
    UNION  
    SELECT Nome FROM Membri  
    WHERE SpazioPadre = ? AND TipoPadre = ?  
    AND Categoria = 'MetodoAstratto'  
);
```

Infine inserimento:

```
INSERT INTO Ereditarietà VALUES (?, ?, ?, ?);
```

OP27 (Rimozione di un'interfaccia estesa / implementata)

```
DELETE FROM Ereditarietà  
WHERE SpazioFiglio = ? AND TipoFiglio = ?  
AND SpazioPadre = ? AND TipoPadre = ?;
```

OP28 (Elenco dei membri di un tipo in ordine alfabetico)

```
SELECT Nome, Categoria, Pubblico, Tipo, Spazio FROM Membri
WHERE SpazioPadre = ? AND TipoPadre = ?
ORDER BY Nome;
```

OP29 (Elenco dei membri ordinati per numero di riferimenti)

```
SELECT M.Nome, M.Categoria, M.Pubblico, M.Tipo, M.Spazio,
COUNT(R.Blocco) AS Numero
FROM Membri AS M LEFT JOIN Riferimenti AS R
ON R.Spazio = M.SpazioPadre
AND R.Tipo = M.TipoPadre
AND R.Membro = M.Nome
AND R.Parametro IS NULL
WHERE M.SpazioPadre = ?
AND M.TipoPadre = ?
GROUP BY M.Nome
ORDER BY Numero DESC, M.Nome;
```

OP30 (Inserimento di un membro)

```
SELECT * FROM MetodiEreditati
WHERE SpazioFiglio = ? AND TipoFiglio = ? AND Nome = ?;
```

Verifica dell'esistenza del tipo e della visibilità:

```
SELECT Pubblico FROM Tipi
WHERE Spazio = ? AND Nome = ?;
```

Inserimento:

```
INSERT INTO Membri VALUES (?, ?, ?, ?, ?, ?, ?);
```

OP31 (Rimozione di un membro)

```
DELETE FROM Membri
WHERE SpazioPadre = ? AND TipoPadre = ? AND Nome = ?;
```

OP32 (Accesso ai dettagli di un membro)

```
SELECT Categoria, Pubblico
FROM Membri
WHERE SpazioPadre = ? AND TipoPadre = ? AND Nome = ?;
```

OP33 (Inserimento di un parametro)

```
SELECT Pubblico FROM Tipi
WHERE Spazio = ? AND Nome = ?;
SELECT MAX(Indice) AS Massimo FROM Parametri
WHERE SpazioFunzione = ? AND TipoFunzione = ?;
INSERT INTO Parametri VALUES (?, ?, ?, ?, ?, ?, ?);
```


OP34 (Rimozione di un parametro)

```
DELETE FROM Parametri
WHERE SpazioFunzione = ? AND TipoFunzione = ?
AND MembroFunzione = ? AND Nome = ?;
```

OP35 (Accesso ai dettagli di un parametro)

```
SELECT * FROM Parametri
WHERE SpazioFunzione = ? AND TipoFunzione = ?
AND MembroFunzione = ? AND Nome = ?;
```

OP36 (Elenco degli utenti)

```
SELECT Email, NumeroBadge, Nome, Cognome, Ruolo FROM Utenti;
```

OP37 (Inserimento di un utente)

```
INSERT INTO Utenti VALUES (?, ?, ?, ?, ?, ?);
```

OP38 (Rimozione di un utente)

```
DELETE FROM Utenti WHERE Email = ?;
```

OP39 (Elenco degli spazi e degli sviluppatori)

Elenco spazi:

```
SELECT Nome, COUNT(Sviluppatore) AS Numero
FROM Spazi LEFT JOIN Assegnazioni
ON Spazio = Nome
GROUP BY Nome
ORDER BY Numero DESC;
```

Elenco sviluppatori:

```
SELECT Nome, Cognome, Email, COUNT(Spazio) AS Numero
FROM Utenti LEFT JOIN Assegnazioni
ON Sviluppatore = Email
WHERE Ruolo = 'Sviluppatore'
GROUP BY Email
ORDER BY Numero DESC;
```

OP40 (Elenco degli sviluppatori assegnati e assegnabili)

Elenco sviluppatori assegnati allo spazio:

```
SELECT Nome, Cognome, Email, COUNT(Spazio) AS Numero
FROM Utenti LEFT JOIN Assegnazioni
ON Sviluppatore = Email
WHERE Ruolo = 'Sviluppatore'
AND Email IN
    (SELECT Sviluppatore FROM Assegnazioni WHERE Spazio = ?)
GROUP BY Email
ORDER BY Numero DESC;
```

Elenco sviluppatori non assegnati allo spazio:

```
SELECT Nome, Cognome, Email, COUNT(Spazio) AS Numero
FROM Utenti LEFT JOIN Assegnazioni
ON Sviluppatore = Email
WHERE Ruolo = 'Sviluppatore'
AND Email NOT IN
    (SELECT Sviluppatore FROM Assegnazioni WHERE Spazio = ?)
GROUP BY Email
ORDER BY Numero;
```

OP41 (Elenco degli spazi assegnati e assegnabili)

Elenco spazi assegnati allo sviluppatore:

```
SELECT Nome, COUNT(Sviluppatore) AS Numero
FROM Spazi LEFT JOIN Assegnazioni
ON Spazio = Nome
WHERE Nome IN
    (SELECT Spazio FROM Assegnazioni WHERE Sviluppatore = ?)
GROUP BY Nome
ORDER BY Numero DESC;
```

Elenco spazi non assegnati allo sviluppatore:

```
SELECT Nome, COUNT(Sviluppatore) AS Numero
FROM Spazi LEFT JOIN Assegnazioni
ON Spazio = Nome
WHERE Nome NOT IN
    (SELECT Spazio FROM Assegnazioni WHERE Sviluppatore = ?)
GROUP BY Nome
ORDER BY Numero;
```

OP42 (Assegnazione di uno spazio ad uno sviluppatore)

```
INSERT INTO Assegnazioni VALUES (?, ?);
```

OP43 (Rimozione di un'assegnazione)

```
DELETE FROM Assegnazioni  
WHERE Spazio = ? AND Sviluppatore = ?;
```

OP44 (Accesso come utente)

```
SELECT Nome, Cognome, Password, Ruolo  
FROM Utenti WHERE Email = ?;
```

OP45 (Modifica di una password)

```
UPDATE Utenti SET Password = ?  
WHERE Email = ?;
```

Progettazione dell'applicazione

Descrizione dell'architettura

Il sito web per la consultazione e la gestione della documentazione è stato realizzato in linguaggio PHP, utilizzando l'estensione mysqli per comunicare con il database locale. L'interfaccia utente è stata progettata per essere funzionale e per rispecchiare i vincoli posti sui dati. Tuttavia si è scelto di implementare anche nel codice lato server tutti i controlli sui comandi in entrata, in modo da garantire la robustezza del database.

Sono state create sei pagine per consentire i principali tipi di utilizzo del sistema, ovvero la visuale pubblica della documentazione, la visuale interna per la sua modifica, una pagina per la gestione degli utenti, una per l'assegnazione degli spazi, un'altra per l'accesso dei dipendenti e infine una per la modifica della password.

Si mostrano di seguito alcune schermate significative, in negativo e in bianco e nero per facilitarne la stampa.

Visuale pubblica di un'interfaccia:

VSLD

[Accedi alla visuale interna](#)

[/](#) [vava.collections](#) / List

intf List

Rappresenta una collezione ordinata che ammette elementi ripetuti.

Interfacce estese: Collection.

Elenco membri:

- static method List array_list ()
Restituisce una lista vuota, basata sull'utilizzo di un vettore.
- static method List array_list_of (Array elements)
Crea e restituisce una nuova lista, basata su un vettore, che contenga gli elementi specificati.

Parametri:

- elements** Gli elementi da inserire nella nuova lista.

- abstract method Any get (Integer index)
Restituisce l'elemento alla posizione specificata.
Se la posizione non è valida, viene lanciato un errore.

Parametri:

- index** La posizione dell'elemento da ottenere.

Editor della documentazione:

Documentazione:

Anteprima:

Rappresenta una collezione ordinata che ammette elementi ripetuti.

Modifica:

Rappresenta una collezione ordinata che ammette elementi ripetuti.

Annulla

Salva

Rimuovi

Gestisci riferimenti:

Seleziona una porzione di testo per continuare.

Pulisci

oppure

(specifica uno spazio)

(specifica un tipo)

(specifica un membro)

(specifica un parametro)

+

Aggiungi autore:

(seleziona)

+

Ultima modifica: 2024-09-11.

Riferimenti:

- "...i sono liste." in [java.collections/Collection](#)
- "...ta una lista implem..." in [java.collections/ArrayList](#)

Pagina di gestione degli utenti:

VSLD

[Indietro](#)

Inserisci nuovo utente

Nome:

...

Cognome:

...

Numero badge:

00000

Indirizzo email aziendale:

(genera automaticamente)

Ruolo:

Visualizzatore

▼

Inserisci

Elenco utenti:

Nome	Cognome	Ruolo	Email	Badge	
Marco	Buda	Amministratore	marco.buda@sus.com	00000	🗑️
Mario	Rossi	Sviluppatore	mario.rossi@sus.com	00001	🗑️

In questa versione, all'inserimento di un nuovo utente, l'amministratore può vedere la password temporanea per il dipendente. In un ambiente reale sarebbe necessario adottare un processo più sicuro.