



Università degli Studi di Salerno
Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica
Classe LM-18

Boscombe

Un sistema informativo per un'agenzia investigativa

A cura di:
Troisi Vito
Matricola 0522501735

Corso di Basi di Dati II
Anno Accademico 2023/24

Repository GitHub del progetto:
<https://github.com/v-troisi5/boscombe>

Indice dei contenuti

1	Descrizione del miniworld	3
1.1	Sommario	3
1.2	Specifica (informale) del miniworld	3
1.3	Glossario	5
2	Progettazione concettuale	7
2.1	Dizionario dei dati	7
2.1.1	Note su entità ed attributi	10
2.2	Schema concettuale	10
2.3	Note sullo schema concettuale	12
3	Insieme campione di operazioni	12
4	Progettazione logica	14
4.1	Ristrutturazione dello schema concettuale	14
4.1.1	Eliminazione delle specializzazioni	14
4.1.2	Eliminazione di attributi non atomici	15
4.1.3	Selezione degli identificatori primari definitivi	16
4.2	Mapping in schema logico e conclusioni	18
5	Normalizzazione dello schema logico	19
5.1	Prima Forma Normale (1NF)	20
5.2	Seconda Forma Normale (2NF)	20
5.3	Terza Forma Normale (3NF)	21
5.4	Forma Normale di Boyce-Codd (BCNF)	22
5.5	Quarta Forma Normale (4NF)	22
6	Note sull'implementazione dell'applicativo Boscombe	24
6.1	Struttura dell'applicativo e tecnologie utilizzate	24
6.2	Operazioni campione	25
6.3	Scelte di modellazione NoSQL	27
7	Conclusioni	29

1 Descrizione del miniworld

1.1 Sommario

Un'agenzia investigativa vuole dotarsi di un sistema informativo per la gestione e la manipolazione delle informazioni relative agli incarichi che riceve, agli investigatori dipendenti ed alle personalità estranee di cui essa si avvale. Il sistema informativo, implementato mediante infrastrutture informatiche, dovrà inoltre tenere traccia delle sedi dell'agenzia e dell'archivio delle prove rinvenute dagli investigatori ed esaminate dagli esperti, nonché di alcune informazioni relative ai clienti dell'agenzia.

È inoltre desiderato un supporto applicativo mediante il quale gli utenti finali del Database System siano in grado di sfruttare le funzionalità progettate e manipolare le informazioni descritte nella specifica seguente. Si precisa che gli utilizzatori finali del sistema, dipendenti dell'agenzia, non sono costituiti esclusivamente da utenti sofisticati/esperti, dunque è richiesto che l'applicazione in questione sia intuitiva in apprendimento e utilizzo.

A seguito della diffusione di questionari ed attuazione di interviste per l'*elicitation* di requisiti, è ottenuta la specifica di cui alla sezione seguente.

1.2 Specifica (informale) del miniworld

L'agenzia investigativa Brett & Co. ha deciso di dotarsi di un sistema informativo da gestire attraverso l'utilizzo di opportuni supporti informatici, nonché di un intuitivo programma applicativo affinché gli impiegati dell'agenzia, preposti a tale scopo ma non necessariamente esperti, possano interagire col suddetto sistema.

L'agenzia si occupa principalmente dell'espletamento di incarichi, mansioni di vario tipo richieste da clienti pubblici e privati, siano esse aziende o persone fisiche. Di ogni incarico si vuole tenere traccia del tipo di incarico di cui si tratta (e.g. sorveglianza, collaborazione in indagine ufficiale) e del suo costo orario, richiesto al cliente perché l'agenzia consideri lo svolgimento della mansione. Si vuole inoltre tener traccia della data in cui ogni incarico è stato richiesto: nessun cliente può richiedere più di un incarico nel medesimo giorno e ogni cliente viene registrato dall'agenzia solo al momento della richiesta di un incarico. Dei clienti, si vogliono memorizzare esclusivamente un nominativo e un recapito relativo al cliente stesso.

Un incarico è espletato da un investigatore, opportunamente identificato da una matricola assegnata dall'agenzia al momento dell'assunzione. Il formato della matricola prevede 9 caratteri alfanumerici. Di ogni investigatore si vuole inoltre tener traccia del suo nominativo, della data di nascita e del codice fiscale. A ciascun investigatore può venire assegnato un mentore, che prende in carico diversi investigatori cadetti. Dei mentori. Naturalmente, i mentorati non sono a loro volta mentori.

La Brett & Co. è una piccola agenzia, ma è dislocata sul territorio in più sedi, alla quale afferiscono un determinato numero di investigatori. L'afferenza rappresenta un rapporto di collaborazione più che di appartenenza tra un investigatore e una sede, e pertanto, ciascun investigatore può essere coinvolto anche nelle azioni di più sedi dell'agenzia. Di ciascuna sede, si vuole memorizzare un nome, che la identifica, e un indirizzo, con particolare rilevanza su via e numero civico. Seppur opzionalmente, è possibile memorizzare anche la data di

insediamento degli investigatori nelle sedi. Ciascuna sede può disporre di un dato numero di esperti, i quali sono identificati con un codice univoco nell'ambito della sede stessa. Gli esperti si occupano di analizzare scientificamente le prove rinvenute, mediante l'applicazione del metodo forense sulla prova stessa. Di ogni esperto si vuole tener traccia del nome, del cognome e soprattutto della professione. Quest'ultima rappresenta anche il ramo delle scienze forensi (e.g. medicina, informatica, psicologia, chimica...) in cui l'esperto svolge le sue funzioni. Ciascun esperto effettua dei test sulle prove: l'agenzia non è interessata ai dettagli tecnici, ma si occupa comunque di tracciare il test principale con cui una prova rinvenuta è stata analizzata. L'agenzia indica esplicitamente che la professione dell'esperto è vitale: essa deve necessariamente corrispondere ad una valida occupazione nel campo delle indagini forensi.

Di ciascuna prova, esaminata da uno ed un solo esperto e relativa ad uno ed un solo incarico, è necessario memorizzare un nome identificativo (non è dato sapere il formato di tale nome, né se utilizzi degli standard, dunque si assumeranno caratteri alfanumerici), un codice di protocollo costituito da esattamente 5 caratteri alfanumerici, oltre a data e luogo di ottenimento. È esplicitamente richiesto che il protocollo corrisponda effettivamente ad uno dei protocolli standard correntemente in uso dall'agenzia. Le prove sono principalmente prove scientifiche, con attinenza stretta alle varie branche delle scienze forensi. Di esse, è necessario memorizzare la tipologia di traccia rinvenuta (ovvero il tipo di prova, e.g. "traccia ematica"), e l'eventuale indicazione sulla volatilità della prova in questione. Per altre prove, quelle in formato di documenti, è necessario memorizzare la categoria di documento. Si noti che alcune prove in forma di documentazione potrebbero essere trattate come prove scientifiche nel caso in cui, su di esse, siano rinvenute tracce particolari o risultino soggetti a volatilità.

L'agenzia si avvale poi di informatori, i quali, confidenzialmente o meno, offrono sostegno alle indagini in vario modo. Degli informatori, si vuole tenere traccia del codice fiscale, di nome e cognome, della paga relativa ad ogni collaborazione e di uno o più contatti con cui è possibile reperire l'informatore stesso. Gli informatori si suddividono in "occasional" e consulenti. I primi non sono professionisti del loro settore e forniscono semplice assistenza in un caso a cui gli investigatori lavorano. Tipicamente, queste persone hanno informazioni relative ad un determinato incarico (o almeno affermano di averne!). Di essi, si vuole memorizzare una breve descrizione. Data la frequente inaffidabilità di questi ultimi, l'agenzia comunica di non aver bisogno di più di informatori consulenti, che occasionali.

I consulenti sono coloro che operano in una data area di competenza, di cui è necessario tenere traccia, e che aiutano gli investigatori dell'agenzia con le loro abilità e conoscenze. Alcuni consulenti poi, sono effettivi professionisti del settore e dispongono di un titolo di studio riconosciuto, che è necessario memorizzare.

In chiusura, l'agenzia si raccomanda di porre particolare rilevanza sulla collaborazione tra uno o più informatori ed uno o più investigatori, relativamente ad un dato numero di incarichi a cui essi partecipano. È da notare che ad ogni incarico partecipano diversi informatori.

Questo chiude la descrizione informale della realtà da modellare. Prima di passare al dizionario dei dati, viene fornito un glossario dei termini più "tecnici" della realtà descritta, i

quali saranno utilizzati per far riferimento ai concetti espressi nella specifica.

1.3 Glossario

Di seguito, è fornita una descrizione - in linguaggio naturale - dei termini tecnici legati alla realtà di interesse, l'agenzia investigativa (Brett & Co.), al fine di facilitarne la comprensione e l'uso all'interno del presente documento.

- **Brett & Co.:** Il nome dell'agenzia che ha richiesto la progettazione di un sistema informatico.
- **Incarico:** Una mansione richiesta da un cliente e presa in carico dall'agenzia.
- **Cliente:** Il richiedente di una determinata mansione (incarico). Può essere pubblico o privato, persona fisica o giuridica.
- **Investigatore:** I dipendenti dell'agenzia che svolgono le indagini, in ufficio e sul campo.
- **Mentore:** Un investigatore, tipicamente con esperienza anche se non è esplicitato, che prende in carico i nuovi investigatori per introdurli al mestiere.
- **Mentorato:** Un investigatore, tipicamente "cadetto", preso in carico da un mentore.
- **Matricola:** L'identificatore numerico di un investigatore. È univoco rispetto all'intera agenzia.
- **Sede:** Una delle suddivisioni amministrative dell'agenzia, con sede legale ad un dato indirizzo.
- **Afferenza:** La cooperazione di un investigatore con una sede.
- **Collaborazione:** La cooperazione professionale di investigatori e informatori in relazione ad un dato incarico.
- **Esperto:** Un professionista in una branca delle scienze forensi che analizza le prove.
- **Forense:** Appellativo fornito alle scienze che operano nel campo dell'indagine criminale.
- **Prova:** Un elemento rilevante per un caso.
- **Protocollo:** L'identificativo del processo metodologico interno all'agenzia con cui una prova viene ricevuta, analizzata e depositata.
- **Prova scientifica:** Una prova analizzabile mediante metodologie strettamente forensi.
- **Traccia:** L'insieme e la tipologia di segni, più o meno visibili, che una prova scientifica riporta.
- **Prova documento:** Una prova in forma di documentazione cartacea.
- **Informatore:** Un collaboratore dell'agenzia, che fornisce informazioni agli investigatori durante gli incarichi.

- **Informatore occasionale:** Un informatore non di professione, solitamente un privato cittadino che risponde ad una richiesta di informazioni emessa dall'agenzia o dalle forze dell'ordine.
- **Informatore consulente:** Un informatore abile nel proprio campo e con una data esperienza in esso.
- **Informatore consulente professionista:** Un consulente lavoratore nel proprio campo, con un titolo di studio riconosciuto.

Proseguendo con la progettazione concettuale, si passerà alla descrizione del dizionario dei dati seguito dallo schema concettuale sviluppato con i costrutti del modello EER.

2 Progettazione concettuale

2.1 Dizionario dei dati

Di seguito, si vuole riportare una raccolta comprensiva delle entità individuate nella specifica di cui sopra. I sostantivi particolarmente rilevanti costituiscono, in modo audacemente chiaro, le svariate entità del problema. Per questa prima sottofase di analisi ci occuperemo esclusivamente delle entità e degli attributi che è possibile individuare mediante la specifica. Sarà poi posta particolare rilevanza sugli attributi che, potenzialmente, consentono di identificare univocamente le entità. Per ogni entità, sono riportati il nome, una breve descrizione, una lista degli attributi delle entità, con il relativo dominio, descrizione ed eventuali particolarità, e i possibili identificatori. Si noti che l'identificatore primario definitivo sarà stabilito in fase di progettazione logica.

Per quanto riguarda le entità deboli individuate, l'identificatore sarà una composizione della chiave del possessore di identificazione e della chiave parziale dell'entità stessa (tale composizione è indicata nella tavola con la notazione “+”).

Per quanto concerne gli attributi derivabili, viene indicato l'identificativo della relativa regola di derivazione, espressa nella sezione seguente.

Investigatore	
Matricola	9 caratteri alfanumerici
Nominativo	Stringa alfanumerica
Data di nascita	Data
Codice Fiscale	16 caratteri alfanumerici in formato standard
Identificatori: Matricola, Codice Fiscale	

Mentore
Tutti gli attributi di Investigatore
Identificatori: Matricola, Codice Fiscale

Informatore	
Codice Fiscale	16 caratteri alfanumerici in formato standard
Nome	Stringa alfanumerica
Cognome	Stringa alfanumerica
Paga	Floating point - esprime la paga per incarico
Contatti	(Multivalore) Stringa alfanumerica
Identificatori: Codice Fiscale	

Occasionale	
Tutti gli attributi di Informatore	
Descrizione	Stringa alfanumerica
Identificatori: Codice Fiscale	

Consulente	
Tutti gli attributi di Informatore	
Area competenza	Stringa alfanumerica
Identificatori: Codice Fiscale	

Professionista	
Tutti gli attributi di Consulente	
Titolo studio	Stringa alfanumerica
Identificatori: Codice Fiscale	

Sede	
Nome	Stringa alfanumerica
Indirizzo	Composto da Via (stringa) e Civico (intero)
Identificatori: Nome	

Esperto	
Codice Esperto	Stringa alfanumerica relativa alla sede
Nome	Stringa alfanumerica
Cognome	Stringa alfanumerica
Professione	Stringa alfanumerica
Identificatori: Sede.Nome + Codice Esperto	

Prova	
Nome identificativo	Stringa alfanumerica univoca
Protocollo	5 caratteri alfanumerici
Data ottenimento	Data
Luogo ottenimento	Stringa alfanumerica
Identificatori: Nome identificativo	

Scientifica	
Tutti gli attributi di Prova	
Tipologia Traccia	Stringa alfanumerica
Volatilità	Booleano
Identificatori: Nome identificativo	

Documento	
Tutti gli attributi di Prova	
Categoria	Stringa alfanumerica
Identificatori: Nome identificativo	

Cliente	
Codice Cliente	Stringa numerica univoca
Nominativo	Stringa alfanumerica
Recapito	Stringa alfanumerica
Identificatori: Codice Cliente	

Incarico	
Data	Data
Tipo	Stringa alfanumerica
Costo orario	Float
Identificatori: Cliente.Codice Cliente + Data	

2.1.1 Note su entità ed attributi

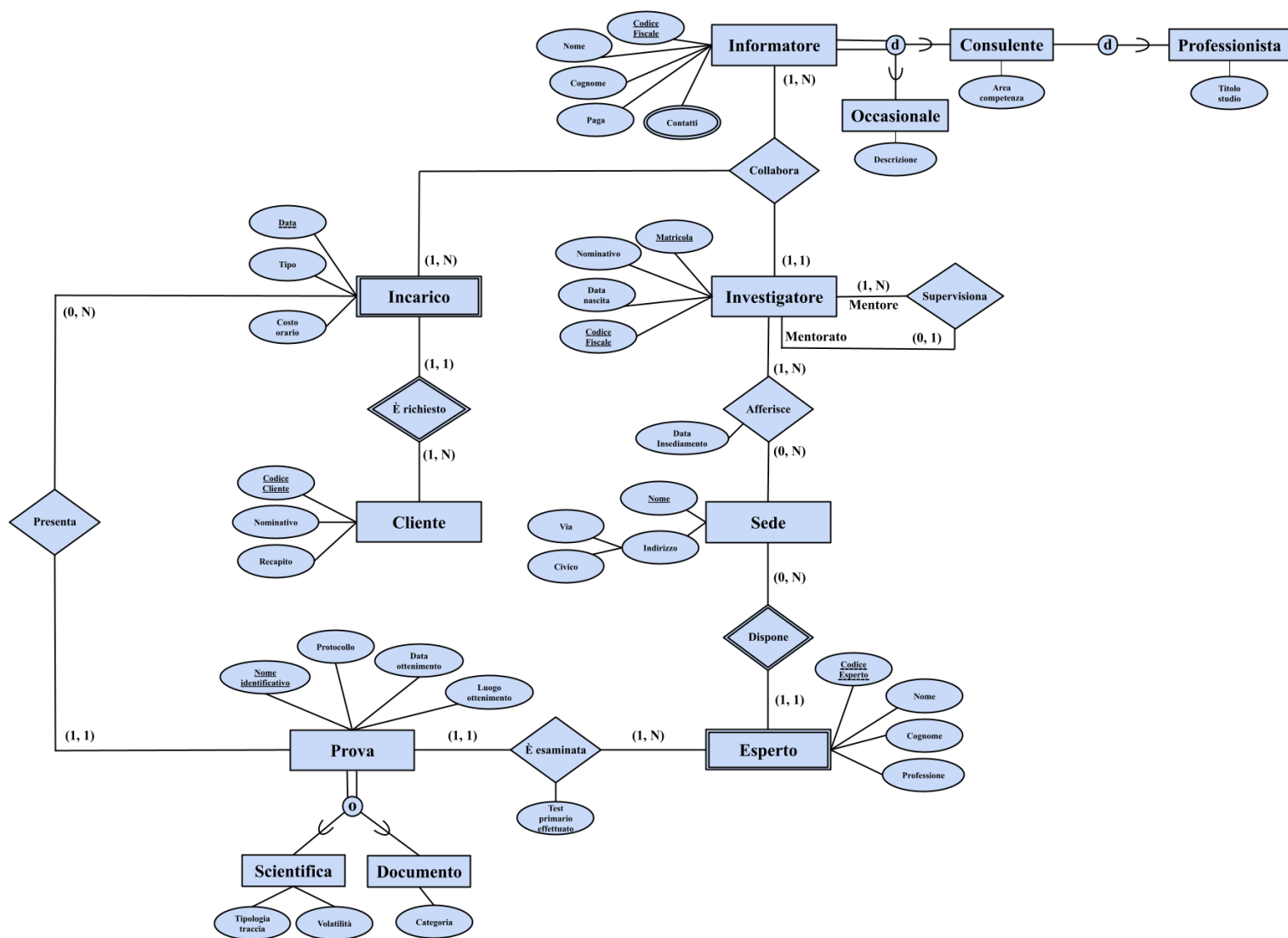
Si noti che l'entità Mentore è, in particolare, un Investigatore. In quanto parte di quella che è stata identificata come una relazione ricorsiva (come trasparirà di seguito), essa non sarà presente nello schema definitivo.

Si osservi inoltre che l'entità Cliente manca di un identificatore individuato dalla specifica. Pertanto, si è scelto di aggiungere un attributo chiave arbitrario: Codice Cliente. Le entità precedentemente descritte possono essere raffigurate secondo la notazione Navathe (alla pagina seguente), la stessa che sarà utilizzata per lo schema definitivo.

Dalla specifica seguono, in aggiunta alle entità proposte, diverse relazioni, di cui sono descritti attributi ed entità partecipanti, con i relativi vincoli di cardinalità/partecipazione, insieme con le citate entità nello schema concettuale nella sezione che segue.

2.2 Schema concettuale

Analizziamo dunque lo schema concettuale (alla pagina seguente) sviluppato a partire dalle tavole di cui ai punti precedenti. Seguiranno poi una serie di note su quest'ultimo e la specifica formale delle regole di derivazione citate durante la descrizione degli attributi:



Regole di lettura e note generali

Si precisa che lo schema concettuale di cui sopra, è stato costruito in maniera tale da permettere, per quanto possibile, il rispetto delle regole di lettura occidentali convenzionali (lettura da sinistra a destra e dall'alto in basso). Ad esempio: Sede dispone (di) Esperto.

Inoltre, per quanto concerne le seguenti partecipazioni parziali:

- Sede (0,N) in Afferisce
- Sede (0,N) in Dispone

Si assume che le entità dei tipi indicati, all'atto dell'inserimento nella base di dati, possano non partecipare, seppure sia una condizione temporanea, ad alcuna istanza delle relazioni individuate. (E.g Una sede appena inserita è priva di investigatori. Ciò ha senso anche in previsione della progettazione logica, e dello stabilimento dei vincoli di integrità referenziale tra le entità partecipanti). Diverso è il caso di Incarico (0,N) in Prevede, in quanto un incarico espletato dall'agenzia può non avere affatto prove ad esso associate.

2.3 Note sullo schema concettuale

Altri elementi degni di nota:

- **La relazione ricorsiva Supervisiona**

Come annunciato in precedenza, la relazione in questione fa di fatto riferimento a due investigatori, seppur essi partecipino con ruoli diversi. Pertanto la relazione è stata trattata come ricorsiva, le cui entità partecipano secondo i ruoli previsti.

- **L'albero di specializzazioni di radice Informatore**

L'entità Informatore prevede specializzazioni, a loro volta specializzate, in accordo con le direttive della specifica. Tale albero sarà opportunamente ristrutturato in sede di progettazione logica.

- **La relazione ternaria Collabora**

La relazione ternaria Collabora può essere oggetto di reificazione. Tuttavia, al fine di esemplificare la gestione della *dipendenza multivalore* che essa genera, per ora, si sceglie di non ristrutturare tale componente dello schema.

3 Insieme campione di operazioni

Si propone ivi di seguito, un insieme di tipiche operazioni fattibili sulla base di dati oggetto di progettazione, e delle quali l'agenzia Brett & Co. è interessata a verificare il funzionamento, in quanto operazioni piuttosto frequenti.

In ogni caso, si noti che queste operazioni NON costituiscono l'insieme complessivo delle

richieste/modifiche operabili sul database. In fase di sviluppo dell'applicazione, verranno proposte ulteriori operazioni, tra cui interrogazioni più complesse.

- **Operazione 1:** Inserire un nuovo informatore consulente con i suoi contatti.
- **Operazione 2:** Elimina i contatti relativi agli informatori con paga minore di 100 che hanno collaborato nel corso di qualche incarico con uno o più investigatori afferenti ad una sede, dato un identificativo di quest'ultima.
- **Operazione 3:** Elimina la sede di tutti gli investigatori che hanno collaborato con un informatore consulente specializzato in Criminologia e con almeno un contatto.
- **Operazione 4:** Recupera gli esperti afferenti a una data sede in cui sono state analizzate prove massimamente volatili.
- **Operazione 5:** Recupera le sedi (e i relativi esperti) a cui afferiscono i mentori degli investigatori che hanno svolto incarichi di un dato tipo e richiesti da un cliente fissato.
- **Operazione 6:** Recupera i dettagli delle sedi a cui, prima di una data fornita, sono afferiti investigatori con due o più incarichi registrati e che hanno lavorato ad incarichi che coinvolgono prove su cui non è stato effettuato alcun test primario.
- **Operazione 7:** Recupera i nominativi degli informatori consulenti limitandosi ai professionisti (ovvero quelli con titolo di studio non nullo), che hanno collaborato con almeno un investigatore, tra quelli che hanno affrontato un incarico con un costo orario maggiore di uno fissato.
- **Operazione 8:** Recupera gli incarichi di un cliente fissato raggruppandoli per data e computano la somma dei costi degli incarichi suddetti. Ordinare i risultati per data.

Si procede con la ristrutturazione dello schema concettuale.

4 Progettazione logica

Si prosegue con la fase di progettazione logica, che si apre con la ristrutturazione dello schema concettuale appena osservato, per consentirne un mapping efficiente verso un opportuno modello logico.

4.1 Ristrutturazione dello schema concettuale

4.1.1 Eliminazione delle specializzazioni

In questa fase, è bene occuparsi dell'eliminazione di specializzazioni e generalizzazioni varie, in quanto non direttamente rappresentabili con l'ausilio del data model relazionale. Le gerarchie individuate nello schema concettuale sono le seguenti:

1. Specializzazione dell'entità Consulente in Professionista
2. Specializzazione dell'entità Informatore in Consulente e Occasionale
3. Specializzazione dell'entità Prova in Scientifica e Documento

Si noti che le prime due specializzazioni DEVONO essere affrontate in tale ordine, in quanto sono parte di un albero gerarchico.

1. Specializzazione dell'entità Consulente in Professionista

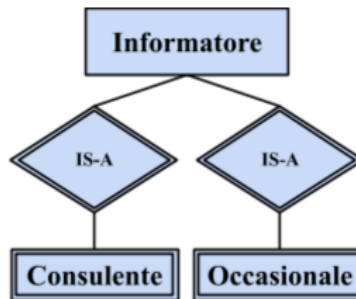
Osserviamo come, le operazioni indicate effettuino poca se non alcuna distinzione tra consulenti e i consulenti Professionisti. Inoltre, la generalizzazione in questione è parziale e l'entità Professionista aggiunge un unico attributo, non avvalorato per tutti i consulenti. Una buona strategia dunque, potrebbe essere quella di accorpare l'entità figlia Professionista all'entità padre Consulente. Lo schema risultante è il seguente:



Si noti che l'attributo Titolo studio è ora facoltativo.

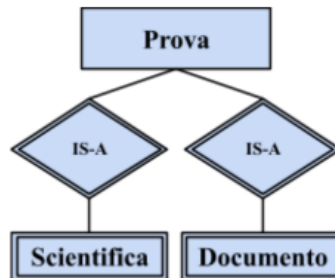
2. Specializzazione dell'entità Informatore in Consulente e Occasionale

La generalizzazione in questione è totale, ma data la presenza di un attributo multivalore in Informatore, portare l'entità padre nelle figlie non sembra una soluzione valida. Considerata la soluzione adottata per risolvere la specializzazione precedente, applicare l'opposto è altrettanto sconsigliato data la potenziale sovrabbondanza di attributi nulli. Dunque, riformuliamo la specializzazione in questione tramutandola in un'associazione del tipo:



3. Specializzazione dell'entità Prova in Scientifica e Documento

Accorpare il padre nelle figlie è una scelta immediata, almeno fintantoché non si considera il considerevole volume atteso dell'entità Prova. Dunque, ci comportiamo come nel caso precedente e tramutiamo la specializzazione in un'associazione del tipo:



La cui cardinalità non è indicata, in quanto, per assunzione è uno-a-molti, trattandosi di un'identificazione di un'entità debole.

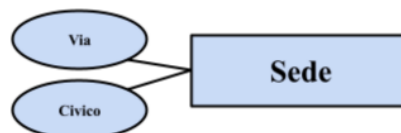
4.1.2 Eliminazione di attributi non atomici

Gli elementi per cui si individua una necessità di ristrutturazione sono i seguenti:

- L'attributo composto Indirizzo dell'entità Sede
- L'attributo multivalore Contatti dell'entità Informatore

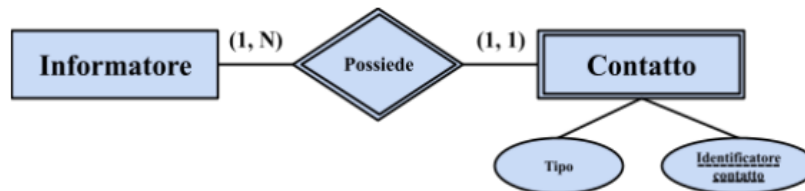
Attributo composto: Indirizzo - Entità Sede

La soluzione in questo caso è piuttosto banale: è possibile partizionare l'attributo negli attributi semplici dai quali è costituito:



Attributo multivalore: Contatti - Entità Informatore

L'attributo multivalore è trasposto in una nuova entità. Tale entità, Contatto, costituirà un'entità debole il cui possessore sarà proprio Informatore. In fase di mapping ovviamente, ci assicureremo di porre un vincolo di integrità referenziale che, dall'entità Contatto, referenzia il suo possessore. Si propone inoltre, di aggiungere un attributo Tipo che disambigua i tipi di contatti:

**4.1.3 Selezione degli identificatori primari definitivi**

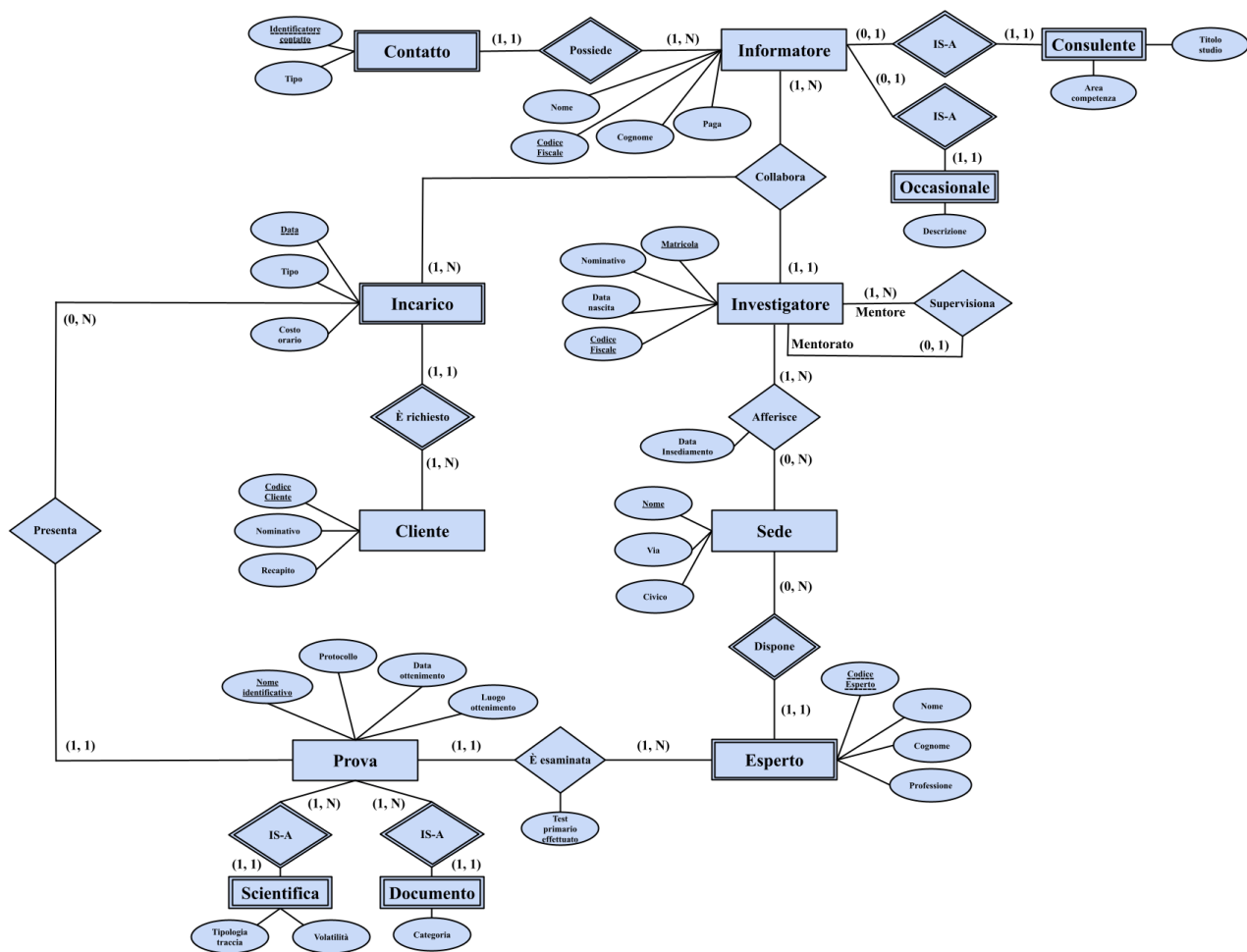
Per la maggioranza delle entità possiamo semplicemente riaffermare la chiave designata in fase di progettazione concettuale. L'entità Investigatore presenta 2 potenziali attributi chiave (candidati): Matricola e Codice Fiscale. La specifica indica che la Matricola è costituita da 9 caratteri esclusivamente numerici. In relazione alla velocità di accesso a tale indice e pertinenza al miniworld, è dunque selezionata la Matricola:

Entità	Chiave
Investigatore	Matricola

Infine, si elencano le chiavi delle entità deboli in funzione della chiave del possessore:

Entità	Chiave (possessore)	Chiave (parziale)
Incarico	Cliente.Codice Cliente	Data
Esperto	Sede.Nome	Codice Esperto
Consulente	Informatore.Codice Fiscale	-
Occasionale	Informatore.Codice Fiscale	-
Scientifica	Prova.Nome Identificativo	-
Documento	Prova.Nome Identificativo	-
Telefono	Informatore.Codice Fiscale	Identificatore Contatto

Lo schema concettuale ristrutturato è dunque il seguente (alla pagina successiva), ed esso sarà oggetto del mapping descritto nella prossima sezione.



4.2 Mapping in schema logico e conclusioni

Alla luce di quanto evidenziato nelle fasi precedenti, è ora possibile passare alla fase di mapping. Le entità e le associazioni saranno tradotte in relazioni considerabili dal modello relazionale. Tendenzialmente, ciascuna entità, eventualmente debole, sarà mappata in una relazione, rispettando i vincoli canonici del modello relazionale.

Schema logico:

- Investigatori(Matricola, Nominativo, Data_nascita, Codice_Fiscale)
- Sedi(Nome, Via, Civico)
- Esperti(Codice_Esperto, Sedi.Nome↑, Nome, Cognome, Professione)
- Clienti(Codice_Cliente, Nominativo, Recapito)
- Incarichi(Clienti.Codice_Cliente↑, Data, Tipo, Costo_orario)
- Prove(Nome_identificativo, Protocollo, Data_ottenimento, Luogo_ottenimento, Esperti.Sedi.Nome↑, Esperti.Codice_Esperto↑, Test_primario_effettuato, Incarichi.Clienti.Codice_Cliente↑, Incarichi.Data↑)
- Scientifiche(Prove.Nome_identificativo↑, Tipologia_traccia, Volatilità)
- Documenti(Prove.Nome_identificativo↑, Categoria)
- Informatori(Codice_Fiscale, Nome, Cognome, Paga)
- Consulenti(Informatori.Codice_Fiscale↑, Area_competenza, Titolo_studio*)
- Occasionali(Informatori.Codice_Fiscale↑, Descrizione)
- Contatti(Informatori.Codice_Fiscale↑, Identificatore_contatto, Tipo)
- SUPERVISIONI(Investigatori(Mentorati).Matricola↑, Investigatori(Mentori).Matricola↑)
- AFFERENZE(Investigatori.Matricola↑, Sedi.Nome↑, Data_insedimento*)
- COLLABORAZIONI(Incarichi.Cliente.Codice_Cliente↑, Incarichi.Data↑, Investigatori.Matricola↑, Informatori.Codice_Fiscale↑)

Si noti che in tale fase rientra un'indicazione finora non direttamente considerabile. L'attributo:

- Data_insedimento della relazione AFFERENZE

è stato marchiato come opzionale in accordo con la specifica.

5 Normalizzazione dello schema logico

Nella fase immediatamente precedente, abbiamo ristrutturato lo schema concettuale uscente dalla fase omonima; successivamente, è stato effettuato il mapping dallo schema suddetto ad uno logico-relazionale.

Si procederà ora, ad osservare la normalizzazione di tale schema, al fine di preparare quest'ultimo all'implementazione su DBMS relazionale. Saranno analizzate le forme normali - prima, seconda, terza forma normale, oltre alla forma normale di Boyce-Codd e alla quarta forma normale, rispettivamente codificate con 1NF, 2NF, 3NF, BCNF, 4NF - analisi che sarà effettuata osservando scrupolosamente la definizione formale delle stesse.

Gli elementi centrali che costituiscono l'input della fase di normalizzazione sono ovviamente le dipendenze funzionali. Si procede di seguito, a definire l'insieme delle medesime individuabile dallo schema logico prodotto in precedenza. Per praticità, la relazione alla quale gli attributi appartengono, viene indicata un'unica volta nel lato sinistro delle dipendenze - a meno di ambiguità sul nome degli attributi stessi.

Sia F l'insieme delle dipendenze funzionali:

$$F = \{$$

Investigatori.Matricola \rightarrow Nominativo, Data_nascita, Codice_Fiscale

Sedi.Nome \rightarrow Via, Civico

Esperti.Codice_Esperto, Esperti.Sedi.Nome \rightarrow Esperti.Nome, Cognome, Professione

Prove.Nome_identificativo \rightarrow Protocollo, Data_ottenimento, Luogo_ottenimento, Esperto.Sede.Nome \uparrow , Esperto.Codice Esperto \uparrow , Test_primario_effettuato

Scientifiche.Prove.Nome_identificativo \rightarrow Tipologia_traccia, Volatilità

Documenti.Prove.Nome_identificativo \rightarrow Categoria

Clienti.Codice_Cliente \rightarrow Nominativo, Recapito

Incarichi.Clienti.Codice_Cliente, Incarichi.Data \rightarrow Tipo, Costo_orario

Informatori.Codice_Fiscale \rightarrow Nome, Cognome, Paga

Consulenti.Informatori.Codice_Fiscale \rightarrow Area_competenza, Titolo_studio

Occasionali.Informatori.Codice_Fiscale \rightarrow Descrizione

Contatti.Informatori.Codice_Fiscale, Contatti.Identificatore_contatto \rightarrow Tipo

Supervisioni.Mentorati.Matricola \rightarrow Supervisioni.Mentori.Matricola

Afferenze.Investigatori.Matricola \rightarrow Sedi.Nome, Data.insediamento

Contatti.Identificatore_contatto \rightarrow Tipo

Scientifiche.Prove.Tipologia_Traccia \rightarrow Volatilità

Incarichi.Tipo \rightarrow Costo orario

}

Sono inoltre presenti le dipendenze multivalore:

- Collaborazioni.Investigatori.Matricola $\rightarrow\rightarrow$ Informatori.Codice_Fiscale
- Collaborazioni.Investigatori.Matricola $\rightarrow\rightarrow$ Incarichi.Cliente.Codice_Cliente, Incarichi.Data

Ovvero:

Collaborazioni.Investigatori.Matricola $\rightarrow\rightarrow$ Informatori.Codice_Fiscale | Incarichi.Cliente.Codice_Cliente, Incarichi.Data

Con tali premesse, si procede con la normalizzazione.

5.1 Prima Forma Normale (1NF)

La fase di ristrutturazione affrontata in precedenza contribuisce significativamente al rispetto della 1NF, la quale specifica che, per ogni relazione appartenente schema di database, questa NON deve prevedere attributi composti e/o multivalore. Ovvero, ciascun attributo, in ciascuna relazione deve risultare atomico e indivisibile.

Dunque, ne concludiamo che l'utilizzo del data model relazionale e il conseguente impiego delle tecniche di mapping, garantisce il rispetto dei vincoli imposti dalla 1NF.

5.2 Seconda Forma Normale (2NF)

La definizione formale della 2NF impone, oltre al rispetto dei canoni della 1NF, il divieto alla comparsa di dipendenze parziali dalla chiave. Dunque, non devono esistere attributi non primi dipendenti da un sottoinsieme proprio di attributi della chiave della relazione di appartenenza.

La relazione Incarichi presenta, effettivamente una dipendenza funzionale degna di nota. Ciononostante, Incarichi.Tipo non è un attributo primo, dunque attualmente, tale considerazione non riguarda la normalizzazione. D'altro canto, dall'insieme F precedentemente individuato si osserva che il tipo di Contatto è funzionalmente dipendente dall'identificatore del contatto stesso, il quale, si rammenta, rappresenta il contatto effettivo (i.e. l'indirizzo e-mail, il numero di telefono). Ovvero, la dipendenza funzionale:

Contatti.Identificatore_contatto \rightarrow Tipo

è parziale in quanto la parte destra dipende da una parte e non da tutta la chiave della relazione Contatti. Trasponiamo dunque l'attributo "problematico" Contatti.Tipo in una nuova relazione, propagando il frammento di chiave da cui dipende - ovvero Contatti.Identificatore_Contatto. Nella relazione originaria, saranno conservati gli attributi rimanenti, ovvero ambedue gli attributi che costituiscono l'identificatore primario di Contatti.

Definiamo la nuova relazione Tipi_Contatti come segue:

Tipi_Contatti(Identificatore_contatto, Tipo)

e riformattiamo la precedente relazione Contatti:

Contatti(Informatori.Codice_Fiscale↑, Tipi_Contatti.Identificatore_contatto↑)

in accordo con quanto osservato nel paragrafo immediatamente precedente. Osserviamo inoltre che nella relazione Contatti è mantenuto un riferimento esterno alla nuova relazione individuata.

5.3 Terza Forma Normale (3NF)

Si ricordi che una relazione è in 3NF se e solo se essa si trova in 2NF e NON figurano dipendenze transitive dalla chiave primaria della relazione stessa.

Esaminando la chiusura dell'insieme delle dipendenze funzionali individuate in precedenza, è possibile notare la presenza di alcune dipendenze transitive, presenti nelle relazioni:

- Incarichi
- Scientifiche

che devono essere opportunamente normalizzate, al fine di rafforzare la semantica di ciascuna.

La relazione Incarichi presenta la seguente FD transitiva:

Incarichi.Clienti.Codice_Cliente, Incarichi.Data \rightarrow Tipo \rightarrow Costo_orario

in quanto il costo orario di un incarico dipende, in accordo con la specifica, dal tipo di incarico da svolgere. L'algoritmo di normalizzazione in 3NF prevede di spostare gli attributi transitivamente dipendenti dalla chiave - in questo caso, Incarichi.Costo_orario - in una nuova relazione, propagando con essi l'attributo non primo da cui dipendono - ovvero Incarichi.Tipo.

Dunque, modelliamo la nuova relazione Tipi_Incarichi:

Tipi_Incarichi(Tipo, Costo_orario)

e la relazione originaria Incarichi diverrà:

Incarichi(Cliente.Codice_Cliente↑, Data, Tipi_Incarichi.Tipo↑)

Osserviamo che, nella nuova relazione Tipi_Incarichi l'attributo Tipo svolge la funzione di identificatore primario, al quale è mantenuto un riferimento esterno nella relazione Incarichi.

In maniera analoga, esaminiamo la dipendenza funzionale:

$$\text{Scientifiche.Prove.Nome_identificativo} \rightarrow \text{Tipologia_traccia} \rightarrow \text{Volatilità}$$

in quanto la volatilità di una determinata prova dipende dalla tipologia in cui la prova stessa ricade.

In accordo con quanto detto in precedenza, concepiamo una nuova relazione *Tipologie_Prove_Scientifiche* in cui l'attributo transitivamente dipendente viene trasposto, e l'attributo da cui dipende svolge la funzione di chiave primaria:

$$\text{Tipologie_Prove_Scientifiche}(\underline{\text{Tipologia_traccia}}, \text{Volatilità})$$

mentre la relazione *Scientifiche* viene ristrutturata come segue:

$$\text{Scientifiche}(\underline{\text{Prove.Nome_identificativo}}, \text{Tipologie_Prove_Scientifiche.Tipologia_traccia})$$

5.4 Forma Normale di Boyce-Codd (BCNF)

La definizione formale della BCNF impone che la parte sinistra di ogni FD di ciascuna relazione sia necessariamente superchiave nella relazione considerata. Nel database così com'è stato previsto finora, le dipendenze funzionali scaturiscono esclusivamente da chiavi primarie o da sovrainsiemi contenenti la chiave primaria. In entrambi i casi, l'insieme considerato è per definizione una superchiave e dunque lo schema di database stesso rispetta i vincoli della BCNF.

5.5 Quarta Forma Normale (4NF)

La 4NF prevede che in ciascuna MVD non banale, la parte sinistra sia una superchiave della relazione di riferimento. Nel caso della relazione *Collaborazioni*, è evidente che ciò non corrisponde al vero, in quanto, ad esempio, la MVD:

$$\text{Collaborazioni.Investigatori.Matricola} \twoheadrightarrow \text{Informatori.Codice_Fiscale}$$

non è banale in quanto le due parti (sinistra e destra) sono disgiunte, gli attributi di *Collaborazione.Incarichi.** non partecipano alla MVD e *Collaborazioni.Investigatori.Matricola* non è una superchiave di *Collaborazioni*.

Normalizziamo *Collaborazioni* applicando l'algoritmo opportuno. Abbiamo già selezionato una MVD della forma $X \twoheadrightarrow Y$ che viola la 4NF dunque non resta che decomporre la relazione in:

$$\text{Collaborazioni} - Y \text{ e } X \cup Y$$

ovvero:

$$\text{Collaborazioni} - \text{Informatori.Codice_Fiscale} \text{ e } \text{Investigatori.Matricola} \cup \text{Informatori.Codice_Fiscale}$$

Ribattezzando *Collaborazioni*, i.e. la relazione rimanente, in *Incarichi_Investigatori*, per chiarezza, abbiamo dunque decomposto *Collaborazioni* in due nuove relazioni:

$$\text{Incarichi_Investigatori}(\underline{\text{Investigatori.Matricola}}, \underline{\text{Incarichi.Cliente.Codice_Cliente}}, \underline{\text{Incarichi.Data}})$$

e

Informatori_Investigatori(Investigatori.Matricola↑, Informatori.Codice_Fiscale↑)

Si noti che l'unica MVD presente in ciascuna è banale in quanto vi partecipano tutti gli attributi della relazione corrispondente.

Questo chiude la fase di normalizzazione effettuata a chiusura della fase di progettazione logica. Si noti comunque che, nel passaggio all'implementazione NoSQL, potrebbe essere effettuata una denormalizzazione implicita o esplicita per facilitare il recupero dei dati o il loro aggiornamento (i.e. le operazioni CRUD).

Schema logico normalizzato:

- Investigatori(Matricola, Nominativo, Data_nascita, Codice_Fiscale)
- Sedi(Nome, Via, Civico)
- Esperti(Codice_Esperto, Sedi.Nome↑, Nome, Cognome, Professione)
- Clienti(Codice_Cliente, Nominativo, Recapito)
- Tipi_Incarichi(Tipo, Costo_orario)
- Incarichi(Clienti.Codice_Cliente↑, Data, Tipi_Incarichi.Tipo↑)
- Prove(Nome_identificativo, Protocollo, Data_ottenimento, Luogo_ottenimento, Esperti.Sedi.Nome↑, Esperti.Codice_Esperto↑, Test_primario_effettuato, Incarichi.Clienti.Codice_Cliente↑, Incarichi.Data↑)
- Tipologie_Prove_Scientifiche(Tipologia_traccia, Volatilità)
- Scientifiche(Prove.Nome_identificativo↑, Tipologie_Prove_Scientifiche.Tipologia_traccia↑)
- Documenti(Prove.Nome_identificativo↑, Categoria)
- Informatori(Codice_Fiscale, Nome, Cognome, Paga)
- Consulenti(Informatori.Codice_Fiscale↑, Area_competenza, Titolo_studio*)
- Occasionali(Informatori.Codice_Fiscale↑, Descrizione)
- Tipi_Contatti(Identificatore_contatto, Tipo)
- Contatti(Informatori.Codice_Fiscale↑, Tipi_Contatti.Identificatore_contatto↑)
- Supervisioni(Investigatori(Mentorati).Matricola↑, Investigatori(Mentori).Matricola↑)
- Afferenze(Investigatori.Matricola↑, Sedi.Nome↑, Data_insediamento*)
- Incarichi_Investigatori(Investigatori.Matricola↑, Incarichi.Cliente.Codice_Cliente↑, Incarichi.Data↑)
- Informatori_Investigatori(Investigatori.Matricola↑, Informatori.Codice_Fiscale↑)

6 Note sull'implementazione dell'applicativo Boscombe

6.1 Struttura dell'applicativo e tecnologie utilizzate

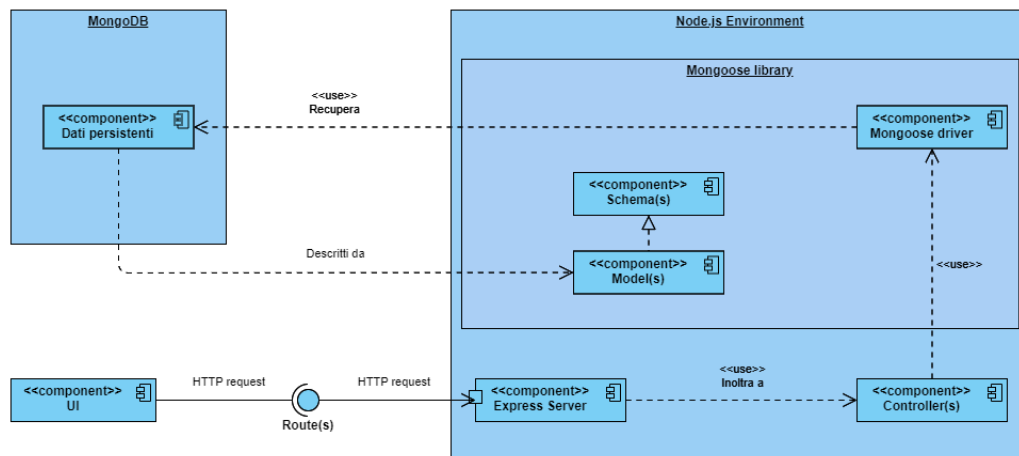
Implementiamo adesso l'applicativo per effettuare i vari tipi di query, da quelle elementari, alle operazioni più complesse. Si supponga di voler convertire lo schema appena normalizzato in un modello non relazionale per prepararlo all'implementazione su un DBMS NoSQL, quale MongoDB.

Il back-end sarà costituito da un server implementato in **Node.js + Express**, mediante le note astrazioni di Route, Controller e Model. I modelli descriveranno i dati persistenti archiviati nel database, il quale, come preannunciato, sarà gestito in logica non relazionale mediante il DBMS **MongoDB**. Le scelte effettuate consentono di lavorare con tecnologie ben consolidate, oltre che spesso utilizzate cooperativamente in stack di progettazione dedicati: Node.js + MongoDB, ad esempio, è una scelta comune, per via della presenza di driver e librerie dedicate che consentono di definire il modello dei dati ed effettuare operazioni CRUD su questi ultimi. Un esempio è dato dalla libreria **Mongoose**, della quale ci si avvarrà per le funzionalità descritte. Per il front-end, non si prevede di utilizzare alcuna tecnologia particolare, avendo l'interfaccia scopi dimostrativi ai fini dell'attività progettuale. La UI (intesa in senso lato di interfaccia + tecnologie di scripting lato client) si limiterà ad accedere alle routes definite nell'ambiente Node.js.

Concentrandoci ora sulla modellazione dei dati, la libreria Mongoose consente di definire degli schemi a partire dai quali sarà generato il data model su cui i dati persistenti baseranno la propria struttura. Gli schemi consentono di specificare il nome delle collection da definire, il tipo dei campi dei documenti con eventuali vincoli (e.g. unicità, obbligatorietà) oltre che array di documenti per l'annidamento e riferimenti ad altre collection. Mongoose consente, in sede di effettuazione della query di “popolare” i riferimenti, sostituendo l'ID fornito da MongoDB con l'intero documento a cui l'ID si riferisce. Ciò consente di effettuare query complesse sfruttando sia la logica di annidamento che quella dei riferimenti, lasciando alla libreria il compito di recuperare fisicamente i dati a partire da un campo di tipo “ref” (*reference*).

E' bene notare che tutto ciò che viene implementato lato back-end è, a tutti gli effetti un modulo Node.js ed eventuali risultati di elaborazione possono essere esportati per l'utilizzo in altri moduli. I Model gestiti da Mongoose non fanno eccezione: una volta costruito ed esportato un Model, i Controller, ovvero i componenti che si occupano di implementare la logica di business delle chiamate all'API possono accedervi, popolare i riferimenti ed effettuare operazioni CRUD.

Una visione più formale della struttura dell'applicativo è osservabile nel component diagram di seguito:



6.2 Operazioni campione

Per comodità, si riportano di seguito le operazioni campione precedentemente descritte. Si noti comunque che, in aggiunta a queste ultime, una API di base verrà implementata per ogni collection elementare (e.g. recupero degli investigatori, delle sedi, dei clienti...).

- **Operazione 1:** Inserire un nuovo informatore consulente con i suoi contatti.
- **Operazione 2:** Elimina i contatti relativi agli informatori con paga minore di 100 che hanno collaborato nel corso di qualche incarico con uno o più investigatori afferenti ad una sede, dato un identificativo di quest'ultima.
- **Operazione 3:** Elimina la sede di tutti gli investigatori che hanno collaborato con un informatore consulente specializzato in Criminologia e con almeno un contatto.
- **Operazione 4:** Recupera gli esperti afferenti a una data sede in cui sono state analizzate prove massimamente volatili.
- **Operazione 5:** Recupera le sedi (e i relativi esperti) a cui afferiscono i mentori degli investigatori che hanno svolto incarichi di un dato tipo e richiesti da un cliente fissato.
- **Operazione 6:** Recupera i dettagli delle sedi a cui, prima di una data fornita, sono afferiti investigatori con due o più incarichi registrati e che hanno lavorato ad incarichi che coinvolgono prove su cui non è stato effettuato alcun test primario.
- **Operazione 7:** Recupera i nominativi degli informatori consulenti limitandosi ai professionisti (ovvero quelli con titolo di studio non nullo), che hanno collaborato con almeno un investigatore, tra quelli che hanno affrontato un incarico con un costo orario maggiore di uno fissato.
- **Operazione 8:** Recupera gli incarichi di un cliente fissato raggruppandoli per data e computano la somma dei costi degli incarichi suddetti. Ordinare i risultati per data.

Al fine di effettuare le scelte circa il mapping da relazionale a non relazionale (il quale è fortemente basato sulle operazioni da effettuare), analizziamo il flusso di ciascuna operazione e formuliamo delle ipotesi:

- **Operazione 1:** L'inserimento di un documento consulente comporta un secondo inserimento di un documento informatore se le due collection vengono mantenute separate. In tal caso, Consulente dovrà disporre di un ID che referencia il proprio documento nella collection Informatore. Inoltre, anche un'eventuale operazione di cancellazione deve premurarsi di cancellare sia il documento Consulente che il corrispondente documento Informatore.
- **Operazione 2:** Con riferimento allo schema logico, per tale operazione è necessario accedere a Informatori, Incarichi e Investigatori, oltre che alle Afferenze di questi ultimi alle Sedi. Se i contatti fossero mantenuti come array annidato nell'Informatore che li possiede, eliminarli significherebbe semplicemente svuotare l'array.
- **Operazione 3:** Per tale operazione, è necessario accedere alle Afferenze, agli Investigatori e agli Informatori (eventualmente Consulenti). Ancora una volta, se i contatti fossero mantenuti come array annidato nell'Informatore che li possiede, verificare una cardinalità di n significherebbe semplicemente verificare l'esistenza dell' $n - 1$ -esimo elemento.
- **Operazione 4:** Rispetto alle altre, quest'operazione è più semplice e richiede di accedere a Prove, Sedi ed Esperti, i quali potrebbero essere mantenuti annidati nella propria sede di riferimento (dovendo comunque selezionare una sede precisa). Ciò implicherebbe la necessità di mantenere l'ID della sede dell'esperto che analizza una prova, nel documento Prova stesso.
- **Operazione 5:** E' necessario accedere a Sedi, Esperti, Investigatori, Incarichi e Clienti. Si noti che per com'è formulata l'operazione, è possibile ipotizzare di mantenere un array di incarichi annidato nell'Investigatore che li esegue. Inoltre, tale operazione non contraddice la scelta precedente di mantenere gli esperti come un array annidato nella loro Sede di riferimento. Inoltre, potremmo dover mantenere l'ID del cliente che richiede un incarico nel documento di quest'ultimo. Per quanto riguarda i mentori degli investigatori, invece, Mongoose consente riferimenti ricorsivi in un Model, dunque la cosa non pone alcun problema.
- **Operazione 6:** E' necessario accedere a Sedi, Afferenze, Investigatori, Incarichi e Prove. Tale operazione parrebbe confermare la scelta di mantenere gli incarichi annidati negli investigatori.
- **Operazione 7:** Tale operazione accede a Incarichi, Investigatori e Informatori (Consulenti), oltre che alle collaborazioni tra questi ultimi (la relazione Informatori_Investigatori nel modello logico).
- **Operazione 8:** L'operazione accede ad Incarichi e Clienti, coinvolgendo raggruppamento e ordinamento. Ciò parrebbe confermare a mantenere in ogni incarico l'ID del cliente che lo richiede, ma anche di fare embedding degli incarichi nel cliente che li richiede.

Questo conclude l'analisi delle operazioni campione. Alla luce di quanto appreso, possiamo ora affrontare il mapping in logica NoSQL delle nostre relazioni.

6.3 Scelte di modellazione NoSQL

Approccio generale: il contenuto di una collection e la struttura di un documento dipende da cosa devo leggere/scrivere più spesso. Si noti che il mapping dal relazionale al non relazionale (scopo di questo esperimento) fa perdere vincoli e dunque saranno utilizzati diversi campi *reference* e meno *embedding*, il che fa ricadere sul programmatore la necessità di implementare la consistenza a seguito di operazioni di aggiornamento. Si noti infine che gli ID delle relazioni vengono ascritti agli ObjectID utilizzati da MongoDB, pena l'impossibilità di implementare in modo semplice e robusto i riferimenti. Analizziamo per i vari cluster di relazioni dello schema logico, le scelte effettuate passando al NoSQL:

- **Cliente, Incarico, TipoIncarico**

Ogni cliente mantiene una collection di incarichi richiesti ed ogni incarico mantiene un riferimento al cliente. Questa ridondanza consente di implementare ottimamente sia l'operazione 5 che l'operazione 8.

Porre TipoIncarico in una collection differente porterebbe a significativi rallentamenti in lettura per le operazioni che richiedono di verificare il tipo di un incarico. Dal momento che Incarico è la collezione che in uno stato di operatività tipico viene sia letta che scritta molto frequentemente (gli incarichi solo la principale mansione dell'agenzia) conviene denormalizzare e lasciare a chi inserisce i dati il soddisfacimento della FD Incarico.Tipo → Incarico.CostoOrario. Tipo e CostoOrario, tornano ad essere parte della sola collection Incarico.

Il concetto di entità debole (Incarico) è costruito con un indice multiplo specificato su ClienteID + Incarico.data.

- **Investigatore, Esperto, Sede, Afferenze:**

Come discusso in precedenza, sembra una buona idea quella di fare embedding degli esperti nella relativa sede.

E' poco sensato fare embedding degli investigatori nella sede, non solo perché la relazione tra le due è M:N, ma in quanto la collection Investigatore partecipa a molte query. Inoltre, dal momento che faccio embedding di Esperti in Sede, per recuperare gli investigatori, dovrei coinvolgere anche Sedi ed Esperti, recuperando molti dati non necessari. Se non vi fosse l'attributo DataInsediamento, potremmo fare embedding delle sedi negli investigatori. In presenza di quest'ultimo tuttavia, è d'obbligo costruire una nuova collection Afferenze e mantenere un riferimento a Sede ed Investigatore.

- **Investigatore, Incarichi_Investigatori:**

La scelta in questo caso è semplice: ogni investigatore mantiene un array embedded di incarichi che ha affrontato.

- **Investigatore, Supervisiona:**

Come ribadito, Mongoose supporta riferimenti ricorsivi ai documenti, dunque Supervisiona si traduce in un riferimento di Investigatore alla collection Investigatore stessa.

- **Prove, Scientifiche, Tipologie_Prove_Scientifiche, Documenti:**

Per quanto concerne Scientifiche e Tipologie_Prove_Scientifiche, in analogia a Incarico e

TipoIncarico, si denormalizzano le relazioni riportando l'attributo Tipo in Scientifiche e lasciare a chi inserisce i dati il soddisfacimento della FD Scientifiche.TipologiaTraccia \rightarrow Scientifiche.Volatilità. TipologiaTraccia e Volatilità, tornano ad essere parte della sola collection Prove.

Come modellare l'associazione IS-A tra Prove e Scientifiche e tra Prove e Documenti? Si osservi che:

- Le operazioni richiedono di recuperare le prove, indipendentemente dal loro tipo
- L'agenzia investigativa, per quanto abbia varie sedi, non è di grandi dimensioni: anche se gestisse (per assurdo) 100,000,000 di prove senza eliminarne alcuna o spostarla su un dispositivo adibito al backup di informazioni archiviate (e.g. nastri, dischi di backup), lo spazio sprecato sarebbe: Taglia di un campo stringa medio $*1.5 * 5,000,000 = 32 \text{ bit} * 1.5 * 100.000.000 = 6 * 10^6 \text{ byte} = \text{ca. } 600 \text{ Mb}$, trascurabile rispetto allo spazio effettivamente richiesto dai dati. (Nota che 100.000.000 di prove corrisponde all'inserimento una prova al secondo per poco più di 3 anni!)

Alla luce di tali considerazioni è possibile denormalizzare ulteriormente e porre gli attributi: Documento.Categoria e Scientifica.TipologiaTraccia, Scientifica.Volatilità, nella collection Prova per facilitare il retrieval delle informazioni.

Nota: sarebbe possibile condurre ulteriori interviste con gli interessati all'agenzia per utilizzare gli attributi con una semantica differente con Categoria che discrimina tra Scientifiche e Documenti e TipologiaTraccia che denota il tipo di prova (il tipo di traccia per le prove scientifiche e il tipo di documento per i documenti).

- **Contatti, TipiContatti, Informatori:**

In analogia ai casi precedenti, si denormalizzano Contatti, TipiContatti riportando un attributo "Tipo" in Contatti. Si effettua embedding di Contatti in Informatori.

- **Informatori, Occasionali, Consulenti:**

Una collection Consulenti deve referenziare la collection "padre" Informatori, in quanto alcune operazioni coinvolgono esclusivamente i Consulenti.

L'attributo Descrizione, unico esclusivo della collection Occasionali è posto in Informatori. L'idea è che, dal momento che la specializzazione di Informatori è disgiunta, se un informatore non è memorizzato come document nella collection Consulenti allora è un informatore occasionale.

- **Investigatore, Informatori_Investigatori:**

Dal momento che potrebbero dover essere recuperati gli Informatori, ma non gli Investigatori, ha senso costruire una collection che mantiene i riferimenti alle collection Investigatore e Informatore.

Nota: per praticità Informatori_Investigatori è stata rinominata in Collaborazioni.

Ciò conclude la modellazione come effettuata. Il tutto è stato implementato, così come dettagliato, mediante schemi e modelli Mongoose. L'implementazione dell'API descritta in precedenza, consente infine di recuperare i dati come modellati e visualizzarli in tabelle, oltre che di eseguire le operazioni campione elencate.

7 Conclusioni

L'esperimento di modellazione di un miniworld volto a costruire una base di dati relazionale e successivamente mapparla per un'implementazione non relazionale costituisce anche la prima esperienza con i DBMS non relazionali dello scrivente. E' stato estremamente stimolante osservare la modellazione dei dati con un nuovo paradigma e lo è stato ancor di più dover effettuare le scelte e prendere le decisioni più opportune per convertire dati relazionali strutturati per utilizzarli in logica NoSQL.

La presente documentazione ed il progetto Boscombe sono stati sviluppati in accordo con la proposta di progetto sottoposta in sede di preparazione all'attività di progettazione, relativamente al Corso di Basi di Dati II - Corso di Laurea Magistrale in Informatica - A.A. 2023/24, mantenuto dalla Chiar.ma Prof.ssa Genoveffa Tortora e dal Dott. Luigi Di Biasi. Sulla pagina GitHub del progetto, raggiungibile dal frontespizio, in aggiunta alla presente ed all'implementazione, è possibile trovare anche i dati mock utilizzati durante lo sviluppo, oltre che lo script Node.js da lanciare per effettuare il caricamento di questi ultimi (il tutto nella directory *load/*). I dati sono stati generati mediante l'applicativo Mockaroo.