



Università di Pisa
Corso di laurea in Ingegneria Informatica

Specifiche di progetto per il corso di
Basi di dati
Anno Accademico 2017–2018

Prof. Gigliola Vaglini, Ing. Francesco Pistolesi

Indice

1	Descrizione delle fasi di progettazione	3
1.1	Note sull'esame	4
1.2	Analisi delle specifiche	4
1.3	Progettazione concettuale	4
1.4	Ristrutturazione del diagramma E-R	5
1.5	Individuazione di operazioni sui dati	5
1.6	Analisi delle prestazioni delle operazioni	5
1.7	Introduzione di ridondanze	6
1.8	Progettazione logica	7
1.9	Analisi delle dipendenze funzionali e normalizzazione	7
1.10	Implementazione su DBMS Oracle MySQL	8
2	Specifiche	9
2.1	Visione d'insieme	9
2.2	Area registrazione	9
2.2.1	Utenti	9
2.2.2	Autovetture	10
2.3	Area car sharing	10
2.3.1	Fruibilità dei veicoli	10
2.3.2	Tragitti	11
2.3.3	Strade	11
2.3.4	Tracking dei veicoli	12
2.3.5	Sinistri	13
2.4	Area car pooling	13
2.4.1	Iscrizione e ruoli	13
2.4.2	Utente proponente e tragitti di car pooling	13
2.4.3	Utente fruitore	14
2.5	Area ride sharing on demand	15
2.5.1	Ruolo del proponente	15
2.5.2	Fruitori di sharing e gestione delle chiamate	16
2.5.3	Sharing multiplo	16
2.6	Area social	17
2.6.1	Valutazione del conducente e del passeggero	17
2.7	Area analytics	17
2.7.1	Affidabilità di un utente	17
2.7.2	Tempi di percorrenza e rilevazione delle criticità	17

Capitolo 1

Descrizione delle fasi di progettazione

Si desidera progettare un database relazionale su DBMS Oracle MySQL che gestisca i dati relativi al sistema informativo di una grande azienda multinazionale che si occupa di mobilità intelligente. L'oggetto della progettazione include l'implementazione di alcune funzionalità di back-end per l'analisi dei dati.

Il progetto deve essere svolto in gruppi composti da due studenti. Solo in casi eccezionali si accettano gruppi di tre studenti.

Le specifiche del database da progettare e delle funzionalità da implementare sono fornite nel Capitolo 2. La progettazione del database si articola nelle seguenti fasi:

1. Analisi delle specifiche;
2. Progettazione concettuale e produzione del diagramma entità-relazione;
3. Ristrutturazione del diagramma entità-relazione;
4. Individuazione di operazioni interessanti sui dati;
5. Analisi delle prestazioni delle operazioni individuate;
6. Miglioramento della performance e introduzione di ridondanze;
7. Progettazione logica e produzione del modello logico relazionale;
8. Analisi delle dipendenze funzionali e normalizzazione dello schema;
9. Scrittura di uno script MySQL che crei il database e lo popoli;
10. Implementazione delle funzionalità di analisi dei dati.

Il presente capitolo ha la funzione di spiegare come si svolge l'esame e quali sono i requisiti e lo scopo di ogni fase della progettazione, le tecniche in esse utilizzate, e gli artefatti prodotti.

Come risultato dell'attività di progettazione, oltre al **database** e al **codice** per la sua creazione, popolazione e implementazione delle funzionalità richieste, in sede d'esame deve essere consegnata una **documentazione** nella quale si evidenziano e si giustificano in modo chiaro tutte le scelte effettuate fase per fase, e si documentano le parti che necessitano di essere spiegate in dettaglio.

1.1 Note sull'esame

Il giorno dell'esame, deve essere consegnata una copia rilegata della documentazione, con pagine numerate e provvista di indice. Devono inoltre essere consegnate due copie del diagramma E-R, una non ristrutturata e una ristrutturata, non rilegate alla documentazione. Il diagramma E-R deve essere ordinato e chiaramente leggibile, deve essere stampato su un foglio di dimensione appropriata (tipicamente A2 o formati più grandi) e il font utilizzato non dovrà essere più piccolo di 12 pt in ogni parte del diagramma.

L'esame si svolge sotto forma di colloquio orale in cui entrambi gli studenti che formano il gruppo sono chiamati a rispondere a domande su tutto il progetto e ad eventuali domande relative ai concetti teorici sui quali esso si basa. Qualora parti della progettazione siano svolte in autonomia dai componenti del gruppo, in sede d'esame non è accettabile che uno dei due componenti non sappia rispondere a domande relative a una parte di progettazione svolta dall'altro. Gli studenti devono inoltre dimostrare capacità critica, e difendere le scelte effettuate motivandone opportunamente le ragioni. Sarà inoltre oggetto di valutazione il livello di chiarezza espositiva.

Il voto della discussione del progetto è espresso in trentesimi ed è lo stesso per entrambi gli studenti che compongono il gruppo.

1.2 Analisi delle specifiche

In questa fase preliminare devono essere lette e analizzate nel dettaglio le specifiche fornite nel Capitolo 2. Lo scopo dell'analisi delle specifiche è quello di capire le funzionalità per le quali si progetta il database, al fine di dotare ognuna di esse del supporto per i dati di cui necessita.

1.3 Progettazione concettuale

Scegliere una strategia di progettazione concettuale fra quelle viste a lezione (top-down, bottom-up, inside-out...), con la quale produrre incrementalmente il diagramma entità-relazione. Si consiglia di focalizzarsi su una piccola parte della specifica, effettuare la progettazione concettuale di essa e produrre la relativa parte di diagramma E-R. Poi leggere un'altra piccola parte di specifica, effettuarne la progettazione concettuale, produrre la relativa parte di diagramma E-R, e così via. Ricontrollare sempre che le parti di diagramma E-R rispecchino tutti i requisiti della specifica. Unire progressivamente le varie parti del diagramma E-R. Il diagramma E-R deve contenere: nomi di entità e associazioni; attributi di entità e associazioni; identificatori primari delle entità; cardinalità delle associazioni; eventuali generalizzazioni, attributi composti e/o multivalore. La notazione da utilizzare per il diagramma entità-relazione è quella vista a lezione, presente sia nelle slide che nel libro di testo consigliato. Non sono accettate notazioni alternative (ad esempio UML, Crow's Foot, etc.).

Esistono vari tool per la realizzazione (più o meno) assistita di diagrammi entità-relazione. Tali tool sono reperibili sul web, taluni con licenza freeware o shareware, altri a pagamento. Alcuni esempi sono Dia, [Draw.io](#), [OmniGraffle](#) [solo macOS], [ConceptDraw](#) [solo macOS], [Microsoft Visio](#), [Adobe Illustrator](#).

I tool sottolineati sono stati utilizzati in modo preferenziale dagli studenti del nostro corso di laurea in edizioni precedenti del corso.

Questa fase produce come risultato il diagramma E-R.

1.4 Ristrutturazione del diagramma E-R

La ristrutturazione del diagramma entità-relazione prevede che siano eliminate le generalizzazioni, gli attributi composti/multivalore. Né le generalizzazioni né gli attributi composti/multivalore sono infatti direttamente traducibili nel modello logico relazionale. Le generalizzazioni devono essere espresse mediante opportune traduzioni alternative (accorpamenti, introduzione di associazioni...). Gli eventuali attributi composti/multivalore presenti devono essere espressi nel diagramma entità-relazione in modo tale da poter essere tradotti nel modello logico relazionale, come appreso durante il corso.

Questa fase produce come risultato il diagramma E-R ristrutturato, traducibile nel modello logico relazionale.

1.5 Individuazione di operazioni sui dati

Devono essere individuate almeno 8 operazioni significative da effettuare sui dati che devono essere implementate in linguaggio MySQL. Le operazioni individuate possono essere sia query di selezione, che query di inserimento, modifica o cancellazione. Con il termine "significative" si fa riferimento a quelle particolari operazioni che contribuiscono in maniera apprezzabile a determinare le prestazioni del database durante il normale carico applicativo¹. Le prestazioni di queste operazioni devono essere ottimizzate se si desidera una buona performance di tutto il database perché la stragrande maggioranza del carico applicativo sarà determinata da esse. E se esse sono inefficienti, la performance generale sarà pessima.

1.6 Analisi delle prestazioni delle operazioni

Per analizzare le prestazioni delle operazioni significative individuate al paragrafo 1.5 si devono dapprima compilare le tavole dei volumi con le quali stimare la mole di dati coinvolta in ciascuna entità o relazione, in termini di occorrenze. Per effettuare un'analisi delle prestazioni che sia sufficientemente indicativa, le stime devono essere ovviamente fatte con giudizio. Dopodiché, deve essere effettuata una stima sulla frequenza giornaliera di esecuzione di ciascuna operazione significativa.

Conseguentemente, deve essere compilata la tavola degli accessi relativa a ogni operazione, ottenendo così una stima delle operazioni elementari (cioè accessi in lettura o scrittura a entità e relazioni) necessarie all'esecuzione. Il numero di accessi a entità e relazioni dipende ovviamente anche dalle stime presenti nelle tavole dei volumi. Ogni riga della tavola degli accessi deve contenere, nell'ordine, i seguenti campi: numero di operazioni elementari eseguite; tipo di

¹Solitamente, per la legge di Pareto, il 20% delle operazioni è responsabile dell'80% del carico.

operazione elementare (lettura o scrittura); tipologia di costrutto coinvolto (E o R); nome del costrutto; breve descrizione.

Nella documentazione che sarà oggetto di discussione in sede d'esame, per ogni operazione devono essere riportati nell'ordine: una descrizione chiara dell'operazione, dell'input (ciò che è noto a priori ed è considerato come dato) e dell'output (ciò che si desidera ottenere); la porzione del diagramma E-R interessata, necessaria a comprendere i cammini di join effettuati dall'operazione; la porzione della tavola dei volumi interessata; la tavola degli accessi. Si consiglia di studiare approfonditamente questa parte della progettazione prima di cimentarsi nella compilazione delle tavole degli accessi.

1.7 Introduzione di ridondanze

Dall'analisi delle prestazioni di ciascuna operazione significativa può emergere che alcune di esse potrebbero trarre beneficio dall'introduzione di ridondanze. Ogniquale volta si introduce una ridondanza per una operazione, per tale operazione deve essere compilata anche la tavola degli accessi che mostra il numero di operazioni elementari (dette accessi) eseguite in presenza della ridondanza. Deve infine essere presa una decisione sul mantenere o no la ridondanza introdotta, in base al risparmio di operazioni elementari che essa comporta. Si faccia attenzione che, scelta un'operazione, una ridondanza ne comporta naturalmente un alleggerimento del carico (riduce drasticamente le operazioni elementari), ma d'altra parte la ridondanza deve essere inevitabilmente mantenuta aggiornata per essere utilizzata.

Più in dettaglio, una ridondanza deve essere aggiornata, mediante una operazione di aggiornamento (o refresh), ogniquale volta viene eseguita una modifica (update, insert o delete) su una tabella sulla quale la ridondanza è basata. Al fine di decidere se mantenere o meno una ridondanza, deve essere studiata anche l'operazione di aggiornamento della stessa, la sua frequenza, la sua modalità (immediate, deferred, on demand) e la sua complessità in termini di operazioni elementari. Ciò permette di calcolare un rapporto costo-beneficio, dove il beneficio è il risparmio di operazioni elementari che la ridondanza comporta per l'operazione della quale si è deciso di migliorare le prestazioni, mentre il costo (quindi lo svantaggio) è dovuto all'introduzione dell'operazione di aggiornamento che mantiene coerente la ridondanza, in termini di operazioni elementari. Si noti che in assenza della ridondanza l'operazione di aggiornamento della stessa non esiste. Il carico computazionale che si introduce per mantenere aggiornata la ridondanza deve quindi essere motivato dal beneficio che comporta la sua presenza. In buona sostanza, scelta un'operazione target T di cui si intende migliorare la performance, si deve dapprima stimare la sua frequenza giornaliera f^T e si deve compilare la tavola degli accessi per calcolare il numero di operazioni elementari o^T necessari alla sua esecuzione. L'operazione target comporterà un numero di operazioni elementari giornaliere pari a $n^T = f^T \cdot o^T$. L'operazione che usa la ridondanza eseguirà un numero di operazioni elementari o_{RID}^T e comporterà quindi un numero di operazioni elementari giornaliere $n_{RID}^T = f^T \cdot o_{RID}^T$, dove solitamente $o_{RID}^T \ll o^T$. Dal confronto della tavola degli accessi delle due operazioni si ottiene il numero di operazioni elementari risparmiate in lettura dalla presenza della ridondanza $\Delta_{read} = n^T - n_{RID}^T$.

A questo punto occorre compilare la tavola degli accessi dell'operazione A di aggiornamento della ridondanza. Questa operazione avrà una frequenza giornaliera g^A e comporterà un numero di accessi pari a o^A . Il numero di accessi giornalieri sarà quindi $n^A = g^A \cdot o^A$. La ridondanza è effettivamente conveniente se il numero di accessi effettuati in presenza di ridondanza (quelli dell'operazione target sommati a quelli dell'operazione di aggiornamento) è inferiore al numero di accessi effettuati dall'operazione target in assenza di ridondanza. Formalmente, si mantiene la ridondanza se $n_{RID}^T + n^A < n^T$, vale a dire $n^A < n^T - n_{RID}^T$. Questo significa che la ridondanza deve essere mantenuta se il numero di accessi n^A introdotti dall'operazione che la mantiene aggiornata sono inferiori al risparmio di accessi Δ_{read} ottenuto dall'operazione target facendo uso della ridondanza.

Per tutte le operazioni che coinvolgono ridondanze, la documentazione di progetto deve contenere l'analisi costo-beneficio descritta precedentemente, e le motivazioni che spingono al mantenimento o meno di una ridondanza introdotta.

Nella versione finale del database, è richiesta la presenza di almeno due ridondanze e, per ciascuna ridondanza, deve essere fornita un'operazione di lettura e una di scrittura che impatti con essa.

1.8 Progettazione logica

Il diagramma entità-relazione ristrutturato deve essere tradotto nel modello logico relazionale, producendo così lo schema del database. In questa fase devono essere scelte le traduzioni più appropriate, qualora un costrutto concettuale sia traducibile in modi diversi.

Le scelte effettuate devono essere opportunamente motivate nella documentazione. Devono inoltre essere specificati chiaramente tutti i vincoli di integrità referenziale necessari al corretto impiego del database.

Eventuali vincoli di integrità generici devono essere implementati mediante trigger MySQL. Qualora non ve ne siano, è richiesta l'introduzione di almeno 2 vincoli di integrità generici.

1.9 Analisi delle dipendenze funzionali e normalizzazione

Per ciascuna relazione (tabella) individuata, devono essere ricercate tutte le dipendenze funzionali non banali. Escluse le relazioni che coinvolgono ridondanze, è richiesto che il database progettato sia in forma normale di Boyce-Codd a questo livello della progettazione.

Qualora la base di dati non lo sia (per relazioni che non coinvolgono ridondanze volute) occorre effettuare una normalizzazione verso la forma normale di Boyce-Codd, tramite opportune decomposizioni delle tabelle che violano tale forma normale.

1.10 Implementazione su DBMS Oracle MySQL

Deve essere realizzato uno script MySQL per creare il database e popolare ogni sua tabella con un numero di record sufficienti a eseguire le funzionalità da implementare e poterne mostrarne un output sensato durante la prova orale. Durante l'esame sarà infatti richiesta l'esecuzione di una o più funzionalità e sarà richiesto ai componenti del gruppo di spiegare il codice che le implementa e descrivere l'output. Potranno essere fatte domande relative alle modalità di implementazione.

Lo script deve contenere anche tutti i vincoli di integrità referenziale, i trigger che gestiscono i principali vincoli di integrità generici e business rule, e gli event e le stored procedure che realizzano le funzionalità lato server descritte nei vari paragrafi del Capitolo 2.

Capitolo 2

Specifiche

2.1 Visione d'insieme

Il database che si desidera progettare ha lo scopo di memorizzare e gestire efficientemente i dati a supporto delle funzionalità del sistema informativo di una grande impresa che si occupa di mobilità intelligente.

In ottica di smart mobility, i clienti dell'azienda possono iscriversi al sito web dell'azienda e interagirvi per offrire o usufruire di servizi di:

- *car sharing*, cioè noleggio di un'automobile di proprietà di terze parti¹;
- *car pooling*, vale a dire uso condiviso di veicoli privati tra due o più persone che devono percorrere un itinerario comune, o parte di esso, senza finalità di lucro (come per esempio fa BlaBlaCar);
- *ride sharing*, ovvero sia un'attività su richiesta per il trasporto di terzi da parte di un privato con un'automobile di proprietà, con o senza finalità di lucro².

Le principali aree tematiche in cui sono suddivisi i dati del database da progettare sono illustrate nei prossimi paragrafi.

2.2 Area registrazione

2.2.1 Utenti

Gli utenti interessati ai servizi dell'azienda si iscrivono accedendo al sito web e fornendo le loro informazioni di anagrafica. In particolare, ogni utente deve fornire il suo codice fiscale, nome, cognome, indirizzo, numero di telefono. L'utente è così memorizzato nel database, assieme alla data in cui si è iscritto. Contestualmente all'iscrizione, l'utente deve inoltre fornire le informazioni relative a un documento di riconoscimento in corso di validità, le quali includono la tipologia e il numero del documento, la scadenza e l'ente che lo ha rilasciato. Il

¹In Italia, solo le aziende possono attualmente concedere un'auto a noleggio, mentre in altri paesi possono farlo anche i privati.

²Si realizza una finalità di lucro quando un soggetto ottiene un guadagno da un'attività.

sistema crea quindi un nuovo account per l'utente e lo memorizza. L'account è caratterizzato da un nome utente, una password, e una domanda di riserva con la relativa risposta per recuperare la password, in caso di smarrimento.

All'utente è anche associato uno stato. Lo *stato* dell'utente è inizialmente *inattivo*, e deve essere attivato previa verifica dell'identità e della validità delle informazioni inserite.

L'utente attivo ha inoltre un *ruolo*, dipendentemente dal quale può offrire servizi di condivisione o può usufruirne. Nel primo caso, l'utente ha il ruolo di *proponente*; nel secondo, ha il ruolo di *fruitore*. Un utente può avere entrambi i ruoli.

2.2.2 Autovetture

Ogni autovettura offerta da un utente proponente per un servizio deve essere preventivamente registrata. Ogni *autovettura*³ è caratterizzata da un numero di targa, una casa produttrice, un modello, una cilindrata, un numero di posti, una tipologia di alimentazione, l'anno di immatricolazione, un consumo⁴ di carburante medio urbano, extraurbano e misto, una capacità del serbatoio di carburante e una velocità massima raggiungibile.

Un'autovettura ha inoltre un costo operativo e un costo di usura per ogni chilometro percorso. Questi costi aumentano se aumenta il peso del veicolo, quindi se ci sono più passeggeri a bordo.

L'autovettura è poi caratterizzata da alcuni optional. Gli optional includono, ad esempio, i tavolini per i sedili posteriori, la connettività, il tetto in vetro, la dimensione del bagagliaio, un valore medio di rumore a bordo in dB, e così via. Sulla base degli optional e delle sue caratteristiche, un'autovettura è associata a un livello di comfort da una a cinque stelle.

2.3 Area car sharing

Il car sharing è una forma di condivisione di un veicolo di proprietà assimilabile al noleggio fra privati. Questa sezione descrive in dettaglio il servizio e i dati necessari per la memorizzazione delle relative informazioni.

2.3.1 Fruibilità dei veicoli

In ottica di car sharing, un utente proponente specifica la fruibilità delle autovetture registrate, cioè i giorni e le fasce orarie in cui ciascuna sua autovettura può essere noleggiata. Ogni autovettura ha una *disponibilità*, i cui valori sono *disponibile* oppure *noleggiata*. Le autovetture nello stato *disponibile* sono situate presso il domicilio del proponente.

A seguito dell'iscrizione dell'utente proponente al servizio di car sharing, viene installata presso il suo domicilio una torretta provvista di sensoristica che riesce a rilevare lo stato del veicolo. Lo *stato* del veicolo è composto dal numero di chilometri percorsi e dalla quantità di carburante presente nel serbatoio.

³Nel seguito, le autovetture sono anche indicate con i termini *vettura* o *veicolo*.

⁴Il consumo di un'autovettura è tipicamente espresso in litri di carburante necessari per percorrere 100 km.

I noleggi possono essere prenotati mediante il sito web. Una *prenotazione di noleggio* è caratterizzata dall'identificativo dell'utente fruitore, dal numero di targa del veicolo che intende noleggiare, da una data di inizio e da una data di fine noleggio. Le prenotazioni di noleggio possono o meno essere accettate da parte dell'utente proponente. Se la prenotazione è accettata, il noleggio avrà luogo. In ogni caso, i dati relativi alla prenotazione di noleggio rimangono memorizzati.

Quando un utente fruitore noleggia un'autovettura, ne acquisisce il diritto di uso esclusivo per un determinato periodo di tempo pari alla durata del noleggio. Gli utenti fruitori hanno l'obbligo di restituire al proponente l'autovettura noleggiata con una quantità di carburante nel serbatoio più vicina possibile a quella trovata nel serbatoio al momento del noleggio. Ciò significa che il sistema non permette di effettuare la restituzione se questa condizione non è soddisfatta. A livello database, un trigger deve quindi impedire che questo si verifichi. All'atto della restituzione del veicolo, questo torna nuovamente disponibile per il noleggio, compatibilmente con la fruibilità espressa dal proprietario.

2.3.2 Tragitti

Un *tragitto* inizia all'atto dell'accensione del veicolo e termina al momento della sosta. Un tragitto è costituito da una posizione (geografica) di partenza, una posizione di arrivo e dalla strada percorsa. Una posizione è caratterizzata da un indirizzo (solitamente quello geograficamente più prossimo), dalle coordinate di latitudine e longitudine, e dal numero di chilometro della strada presso il quale la posizione geografica si trova. Se la strada percorsa durante il tragitto non è la stessa, il tragitto è composto da un insieme di strade percorse lungo le varie tratte del tragitto, per quanti chilometri sono state percorse, da quale punto a quale punto, in quale giorno e in quale fascia oraria.

I tragitti percorsi dalle vetture noleggate sono costantemente memorizzati nel database come sarà spiegato nel paragrafo 2.3.4. Per rendere possibile la memorizzazione delle informazioni relative ai tragitti di un noleggio, il database deve poter memorizzare i dati relativi alle strade percorse, caratterizzate come spiegato in dettaglio nel prossimo paragrafo.

2.3.3 Strade

Una strada è caratterizzata da un codice che ne indica la tipologia secondo la classificazione amministrativa. Secondo questa classificazione, la *tipologia* di una strada può essere statale (sigla *SS*), regionale (sigla *SR*), provinciale (sigla *SP*), comunale (sigla *SC*) o vicinale (sigla *SV*).

Una strada è poi caratterizzata da un identificatore numerico. Talvolta, oltre al numero, può esserci anche una delle seguenti categorizzazioni: *dir* (per diramazione), *var* (per variante), *racc* (per raccordo), *radd* (per raddoppio), oppure un suffisso numerale come *bis*, *ter* o *quater*. Una strada può inoltre avere un nome. Ci sono invece strade che hanno solo il nome e non sono classificate secondo la classificazione amministrativa. È il caso delle strade comunali situate nei centri abitati.

Una strada ha sempre una lunghezza misurata in chilometri. Ad ogni chilometro di una strada, è associata una coppia di coordinate geografiche composta da latitudine e longitudine. Una strada è anche caratterizzata un numero di

carreggiate, da un numero di corsie per carreggiata e da un numero di sensi di marcia. Se il senso di marcia è uno, si parla di strada a senso unico, altrimenti si parla di strada a doppio senso.

Una strada è infine caratterizzata da uno o più limiti di velocità. Il limite di velocità può infatti variare dipendentemente dal chilometro della strada considerato. Per esempio, dal chilometro 55 al chilometro 80 della strada SS1 (Aurelia) può essere presente il limite di velocità di 50 km/h, mentre dal chilometro 81 al chilometro 200 può essere in vigore il limite di velocità di 70 km/h. I limiti di velocità dei vari tratti di una strada devono essere opportunamente memorizzati nel database.

Le strade seguono inoltre un'ulteriore classificazione, detta tecnica, che le divide in autostrade, strade urbane e strade extraurbane. Le strade extraurbane possono essere secondarie o principali.

Le *strade urbane* e le *strade extraurbane secondarie* possono contenere incroci. Un incrocio è il punto in cui due o più strade entrano in comunicazione. Attraverso gli incroci, un veicolo ha quindi la possibilità di cambiare la strada che sta percorrendo, continuando il proprio tragitto su un'altra, naturalmente se il senso di marcia di quest'ultima glielo permette. Un incrocio è ubicato presso un determinato chilometro di una strada, la quale si incontra con un'altra strada presso un determinato chilometro di quest'ultima. Se le strade che si incontrano in un incrocio sono più di due, il precedente modo di ubicare l'incrocio sulla mappa stradale resta valido, purché opportunamente adattato.

Le *strade extraurbane principali* e le strade extraurbane secondarie con più di una corsia per almeno un senso di marcia non hanno incroci: un esempio di questo tipo di strada è la Fi-Pi-Li. Su questi tipi di strade ci si immette tramite corsie di immissione e se ne esce percorrendo una corsia di uscita. Le corsie di immissione e le corsie di uscita sono situate a un determinato chilometro di una strada che funge da imbocco o uscita, rispettivamente. Un raccordo mette in comunicazione strade che non possiedono incroci.

Le *autostrade* sono un'altra categoria di strade prive di incroci, e sono contrassegnate dalla sigla A seguita da un numero. Esistono anche autostrade con doppia numerazione (ad esempio autostrada A8/A26). Le autostrade prevedono il pagamento di un pedaggio chilometrico a fronte della loro percorrenza. Autostrade diverse possono prevedere tariffe di pedaggio diverse. Il pedaggio può anche variare da un tratto all'altro della stessa autostrada.

2.3.4 Tracking dei veicoli

Durante il corso del noleggio, l'autovettura è costantemente tracciata tramite un'attività di tracking, cioè i vari tragitti percorsi sono memorizzati nel database. Nel database è inoltre memorizzata la posizione del veicolo durante i tragitti, a intervalli di tempo regolari. Questo intervallo di tempo è lo stesso per tutti i veicoli. La posizione è memorizzata includendo la targa del veicolo, il timestamp, l'identificativo della strada e il chilometro presso il quale il veicolo si trova, e le coordinate geografiche di latitudine e longitudine. Le informazioni sulla posizione dei veicoli sono mantenute in un'area del database caratterizzata da alti standard di sicurezza, in quanto soggetti a privacy.

Tutti i dati relativi ai tragitti percorsi durante un noleggio sono utili per esempio in caso di sinistro o nel caso il veicolo noleggiato subisca atti vandalici. I dati relativi alla posizione attuale dei veicoli noleggiati è anche utile ai fini

di individuare, per esempio, i tragitti più veloci da un luogo a un altro in determinate fasce orarie d'interesse. Ma anche per cercare di capire la situazione del traffico in tempo reale nelle strade attualmente percorse dalle macchine noleggiate.

Il database ha un'area dove sono memorizzati i tempi medi di percorrenza di strade e parti di strade. Quando questi tempi medi aumentano, significa che la viabilità è compromessa, per esempio a causa di un incidente o di una coda.

2.3.5 Sinistri

Nel caso in cui il veicolo noleggiato subisca/provochi un sinistro, il database deve memorizzare i dati del sinistro, ovverosia il modello, la casa automobilistica e il numero di targa del veicolo o dei veicoli coinvolti nel sinistro⁵, l'orario del sinistro, la posizione⁶, i dati relativi al fruitore e la dinamica. La dinamica del sinistro è un testo.

I sinistri che un utente commette e quelli in cui si trova coinvolto contribuiscono a determinare l'affidabilità dell'utente sulla quale gli utenti proponenti baseranno l'accettazione o meno delle sue future prenotazioni di noleggio (si veda il paragrafo 2.7).

2.4 Area car pooling

Il car pooling è un altro servizio che l'azienda mette a disposizione, e consiste nel condividere un veicolo di proprietà con altre persone, su tragitti o parti di tragitto. Lo scopo del car pooling è diminuire i costi di utilizzo di un veicolo, distribuendone la quota parte fra più persone che sarebbero diversamente costrette a utilizzare il proprio veicolo sullo stesso tragitto che altre persone sono comunque costrette a percorrere nella stessa fascia oraria. Il car pooling promuove la socializzazione ed è anche (e soprattutto) un modo per ridurre considerevolmente le vetture in circolo, diminuendo così le emissioni e la probabilità di code. BlaBlaCar è un esempio di questo tipo di servizio.

2.4.1 Iscrizione e ruoli

Per poter usufruire del servizio di car pooling, gli utenti si iscrivono al servizio e specificano se desiderano condividere il loro veicolo, oppure se desiderano solo essere passeggeri. Nel primo caso avranno il ruolo di proponente, nell'altro di fruitore, come descritto nel paragrafo 2.2.1.

2.4.2 Utente proponente e tragitti di car pooling

Ciascun utente proponente deve specificare le caratteristiche del veicolo col quale intende erogare il servizio di car pooling, così come accade per il car sharing (si veda il paragrafo 2.2.2). Tutte queste informazioni sono salvate all'interno del database.

⁵I veicoli coinvolti escludono il veicolo condotto dal fruitore. Questi dati sono infatti recuperabili dalle informazioni di noleggio.

⁶La posizione è intesa come chilometro di una strada e/o coordinate geografiche

L'utente utilizza il servizio in modi diversi dipendentemente dal ruolo (proponente o fruitore). L'utente proponente deve creare nuovi pool, cioè tragitti relativamente ai quali intende mettere a disposizione il suo veicolo per il servizio di car pooling. Per fare ciò, l'utente proponente si collega alla sua area riservata del portale web e inserisce i dati di un nuovo pool. Per inserire un pool, l'utente specifica il tragitto che percorrerà, il giorno di partenza e l'orario di partenza. Può essere necessario specificare anche il giorno di arrivo, qualora questo sia diverso dal giorno di partenza.

Se l'utente proponente esprime flessibilità, il tragitto del pool inizialmente inserito può essere soggetto a *variazioni* per andare incontro alle necessità dei potenziali utenti fruitori. Ciò significa che il proponente è disposto a considerare l'idea di effettuare delle variazioni sul tragitto del pool. L'utente proponente potrebbe per esempio accettare di interrompere il tragitto uscendo da un'autostrada, percorrendo un tratto di strada statale per permettere la salita o la discesa di un fruitore, per poi rientrare in autostrada. Le variazioni sono specificate dai fruitori come sarà descritto nel paragrafo 2.4.3, e devono essere approvate dal proponente affinché il fruitore possa effettivamente prendere parte a un pool. Per aiutare gli utenti fruitori a capire il grado di flessibilità offerto sul tragitto di un pool, l'utente proponente lo quantifica con un valore in una scala {basso, medio, alto}. Questa informazione è inserita fra le altre del pool. Una volta inserito, il pool ha un periodo di validità, cioè i potenziali fruitori possono prendere parte al pool fino a un determinato numero di ore dalla partenza, specificato dal proponente. Questo numero di ore non può essere inferiore a 48. Durante il periodo di validità, il pool è nello stato **aperto**, dopodiché passa nello stato **chiuso**. Un'ora prima della partenza il pool passa nello stato **partito**. Da questo momento, ogni attività sul pool è interdetta.

2.4.3 Utente fruitore

Per prendere parte a un pool, un utente fruitore si collega al sito, inserisce il tragitto e ricerca i pool su quel tragitto che si trovano in stato **aperto**. L'utente fruitore può quindi prenotare uno fra i pool trovati. Una *prenotazione di pool* è caratterizzata da un codice, un utente proponente, un utente fruitore, e naturalmente dall'identificativo del pool. Nel caso in cui l'utente fruitore non trovi pool in stato **aperto** sul tragitto d'interesse, una funzionalità di ricerca avanzata dà l'opportunità di ricercare i pool chiusi con tragitto simile a quello d'interesse per il fruitore, per i quali l'utente proponente abbia espresso flessibilità. Un tragitto simile a un tragitto d'interesse è un tragitto che può essere modificato⁷ per includere il luogo di partenza e/o di destinazione che l'utente fruitore desidera. La modifica è specificata dall'utente fruitore mediante l'inserimento di una variazione a corredo della prenotazione di pool.

Una *variazione* ha un codice, un tragitto target (quello del pool che intende modificare) e le informazioni proprie della variazione stessa, ovverosia la caratterizzazione della modifica del tragitto. Lo studente è chiamato a progettare un modo per rappresentare una variazione di un tragitto e misurarne l'entità, facendo così in modo che la variazione possa essere opportunamente memorizzata nel database.

⁷Si tenga presente che la modifica è comunque di entità lieve o moderata. Non è pensabile che a fronte di flessibilità, seppure alta, il proponente accetti di percorrere 20 chilometri in più rispetto al tragitto originario come conseguenza di una variazione!

Il proponente accetta o meno le variazioni proposte dai potenziali fruitori nelle prenotazioni di pool, valutandole anche sulla base della loro entità. Se accetta, il fruitore da potenziale diviene effettivo e prenderà realmente parte al pool. Se la variazione non è accettata, il potenziale fruitore è invece escluso dal pool e la prenotazione perde validità. Per avere più possibilità di trovare un pool del quale poter poi effettivamente usufruire, un utente fruitore può proporre la stessa variazione in prenotazioni di pool diverse⁸. Se più proponenti accettano la variazione, l'utente fruitore sceglie il pool sul quale viaggiare. Tutti i dati precedentemente descritti sono memorizzati nel database in modo tale che si possa conoscere la lista dei passeggeri e le eventuali variazioni di tragitto da effettuare, al momento della partenza.

Tutte le informazioni memorizzate permettono anche di determinare la *spesa* che il fruitore deve sostenere. In particolare, la spesa di un pool è data dalle quote parte derivanti dal costo operativo del veicolo, dall'attuale costo del carburante, dal consumo di carburante nelle varie tratte percorse, e dai costi di usura. In caso di variazioni, il proponente può aumentare il prezzo di una percentuale indicata all'atto della creazione del pool caratterizzato da flessibilità.

Anche i tragitti eseguiti in modalità di car pooling sono tracciati come descritto nel precedente paragrafo. In particolare, i dati di posizione del veicolo sono salvati a intervalli di tempo regolari e sono condivisi per analisi avanzate dei dati in grado di determinare le criticità in essere della viabilità, e poterle anche prevedere con una determinata percentuale di attendibilità.

2.5 Area ride sharing on demand

Il ride sharing è un servizio di condivisione su richiesta (come quello realizzato all'estero da Uber) dove un soggetto privato usa la sua vettura per offrire una sorta di "servizio taxi", dando così la possibilità di condividere parti del tragitto che sta percorrendo ad altre persone iscritte al servizio, a fronte di una richiesta che queste inviano in tempo reale. Il ride sharing è reso possibile grazie all'impiego di moderne tecnologie e dispositivi quali localizzatori GPS, smartphone e social network.

2.5.1 Ruolo del proponente

Nell'ambito del ride sharing, l'utente proponente, appena prima di iniziare a percorrere un tragitto, indica che per quel tragitto è disposto a erogare un servizio di ride sharing.

Uno *sharing* è quindi caratterizzato da un utente proponente, da un tragitto e da un orario di partenza. Sulla base dei tempi di percorrenza medi memorizzati nel database e del livello di traffico presente attualmente sul tragitto, è possibile stimare l'arrivo della vettura nei vari punti del tragitto. I tempi medi di percorrenza sono memorizzati in un'area del database e sono costantemente raffinati con i dati collezionati dalle autovetture iscritte al servizio. Questo dà ai fruitori la possibilità di utilizzare un'apposita app sul proprio smartphone in grado di visualizzare una mappa grafica animata sulla quale è possibile vede-

⁸Ciascuna di queste prenotazioni, essendo relative a tragitti percorsi nella stessa fascia temporale dello stesso giorno, è naturalmente riferita a veicoli diversi.

re tutte le autovetture attualmente nelle vicinanze che sono disposte a fornire sharing per un tragitto d'interesse.

2.5.2 Fruitori di sharing e gestione delle chiamate

Una volta selezionato il veicolo del proponente, l'utente fruitore inserisce nel sistema una chiamata. La selezione avviene in base alla reputazione del proponente, ma anche sulla base della rapidità con cui il proponente è in grado di raggiungere la posizione in cui si trova il fruitore.

Una chiamata è caratterizzata da un codice, dal codice dell'utente fruitore scelto, dalla sua posizione attuale e dalla sua destinazione desiderata, nonché da un timestamp di chiamata. La posizione attuale e la destinazione desiderata sono espresse come identificativo del chilometro di una strada. La chiamata è inizialmente nello stato **pending**. L'utente proponente visualizza sulla plancia del veicolo la chiamata e può decidere, come risposta, se accettarla o ignorarla. Se la chiamata è ignorata, questa va in stato **rejected** e l'utente fruitore si vede costretto a selezionare un altro proponente. La chiamata resta comunque memorizzata nel database, assieme al timestamp di risposta.

Quando una chiamata è accettata, questa va in stato **accepted** e il timestamp di risposta è memorizzato assieme al cambio di stato. Da questo momento, il proponente si impegna a raggiungere il fruitore. Quando il proponente raggiunge il fruitore, il fruitore sale sulla vettura e inizia la corsa. La corsa è ovviamente relativa alla chiamata **accepted** ed è soggetta a tracking come descritto nel paragrafo 2.3.4. Quando la corsa termina, il fruitore lascia il veicolo e il sistema salva il timestamp di fine corsa. Queste informazioni sono memorizzate nel database.

2.5.3 Sharing multiplo

Le autovetture che stanno effettuando sharing, possono ospitare più corse simultaneamente. Ciò significa che un proponente può accettare nuove chiamate anche durante una corsa e quindi ospitare più fruitori contemporaneamente, fino al numero massimo di posti messi a disposizione dall'autoveicolo.

Se un utente fruitore non riesce a trovare un proponente che sta percorrendo un tragitto che include quello d'interesse, non potendo proporre variazioni di tragitto — come invece avviene nel caso del car pooling —, può avvalersi di uno *sharing multiplo*. In questo caso, una funzionalità di analisi dei dati proporrà al fruitore la possibilità di usufruire di più vetture in modo sequenziale per raggiungere la destinazione desiderata. La funzionalità è in grado di trovare un insieme di veicoli in sharing ciascuno dei quali sta percorrendo un tragitto che può combinarsi con il tragitto percorso da un altro veicolo, e quello di quest'ultimo veicolo con il tragitto di un altro ancora. Ciò permette al fruitore di raggiungere la destinazione desiderata percorrendo parti del suo tragitto su veicoli diversi.

Deve essere progettata e implementata una versione della funzionalità sopra descritta. Le scelte progettuali effettuate devono essere opportunamente spiegate nella documentazione. Se si usano formule, spiegarle opportunamente, specialmente nel caso in cui includano parametri, i valori dei quali devono essere motivati nella documentazione con spiegazioni verosimili, anche se esse si basano su aspetti semplici e intuitivi.

Gli sharing multipli, le loro chiamate e le loro corse sono memorizzate nel database come spiegato per il ride sharing a singolo veicolo.

Lo sharing multiplo è obbligatorio solo per i gruppi eccezionalmente composti da tre studenti.

2.6 Area social

L'area social contiene le valutazioni degli utenti proponenti e fruitori. Sulla base di queste valutazioni (e non solo) si basa la quantificazione della reputazione di un utente.

2.6.1 Valutazione del conducente e del passeggero

Alla riconsegna di un veicolo in car sharing, oppure al termine di un tragitto di pooling o di una corsa in sharing, il proponente è valutato dal fruitore con un punteggio da una a cinque stelle relativamente a un insieme di aspetti, e da una breve recensione testuale. Gli aspetti includono un giudizio inerente alla persona, al suo comportamento, alla sua serietà, e al piacere di viaggio (nel caso ci si riferisca a un pooling o a un ride sharing). Una valutazione ha un codice, il codice dell'utente proponente, il codice dell'utente fruitore (quello che esprime il giudizio), il tragitto percorso, il numero di stelle assegnate per ciascun aspetto, e la recensione testuale.

Analogamente, il fruitore esprime un giudizio sul proponente. Anche questa valutazione è memorizzata nel database. Le valutazioni danno la possibilità di effettuare un ranking grazie al quale i proponenti e i fruitori possono rispettivamente accettare (o meno) l'offerta o la richiesta di servizi di condivisione.

2.7 Area analytics

L'area analytics contiene funzionalità lato server (data tier) che permettono di analizzare i dati alla ricerca di informazioni utili a migliorare costantemente il servizio offerto dall'azienda.

2.7.1 Affidabilità di un utente

Questa funzionalità deve fornire un giudizio di un utente dipendentemente dal ruolo svolto. Il giudizio è una valutazione riepilogativa che può essere articolata sulla base del rispetto degli orari, del rispetto dei limiti di velocità, ma anche sulla base degli aspetti descritti nel paragrafo 2.6.1.

Il modo di aggregare i dati e le scelte fatte devono essere descritti nella documentazione.

2.7.2 Tempi di percorrenza e rilevazione delle criticità

Questa funzionalità analytics deve analizzare i tempi attuali di percorrenza dei vari tratti di strada sui quali ci sono tragitti di sharing o pooling e raffinare i tempi medi di percorrenza presenti nel database. Lo studente progetti un

modo per poter dotare la funzionalità della capacità di rilevare prima possibile le criticità della viabilità, come per esempio l'inizio del formarsi di una coda.

La funzionalità può essere molto utile perché, condividendo questi dati sulle reti sociali, si può fare in modo che altri veicoli modifichino il loro tragitto in tempo reale per cercare di non peggiorare ulteriormente la situazione viaria sui tragitti già compromessi.

Le scelte progettuali effettuate e la funzionalità stessa devono essere opportunamente spiegate nella documentazione di progetto.