



**Specifiche di progetto per il corso di
Basi di dati
A.A. 2020 – 2021
Corso di Laurea in Ingegneria Informatica**

Prof. Gigliola Vaglini, Ing. Francesco Pistolesi

Indice

1	Descrizione delle fasi di progettazione	3
1.1	Analisi delle specifiche	4
1.2	Progettazione concettuale	4
1.3	Ristrutturazione del diagramma E-R	5
1.4	Individuazione di operazioni sui dati	5
1.5	Analisi delle prestazioni delle operazioni	5
1.6	Progettazione logica	6
1.7	Analisi delle dipendenze funzionali e normalizzazione	7
1.8	Implementazione su DBMS Oracle MySQL	7
2	Specifiche	8
2.1	Visione d'insieme	8
2.2	Area generale	9
2.2.1	Accounting	9
2.2.2	Topologia dell'edificio	9
2.3	Area dispositivi	9
2.3.1	Dispositivi e smart plug	10
2.3.2	Interazione utente-dispositivo	10
2.4	Area energia	11
2.4.1	Sorgenti energetiche	11
2.4.2	Contabilizzazione e uso dell'energia	11
2.5	Area comfort	13
2.5.1	Trattamento aria	13
2.5.2	Smart lighting	13
2.6	Area sicurezza	14
2.6.1	Accessi e intrusioni	14
2.6.2	Serramenti	15
2.7	Area analytics	15
2.7.1	Abitudini degli utenti tramite association rule learning	15
2.7.2	Ottimizzazione dei consumi energetici	17
2.7.3	Custom analytics	18
2.8	Dimensionamento del lavoro dei gruppi in base alla loro numerosità	18

Capitolo 1

Descrizione delle fasi di progettazione

Si desidera progettare un database relazionale su DBMS Oracle MySQL che permetta di memorizzare i dati relativi a un sistema che gestisce una smart home (casa intelligente). Oltre a ospitare i dati, il database sarà dotato di alcune operazioni e funzionalità di data analytics che dovranno essere realizzate.

Il presente capitolo ha la funzione di spiegare come si svolgerà l'esame e quali sono i requisiti e lo scopo di tutte le fasi del lavoro, le tecniche che dovranno essere utilizzate, e gli artefatti che dovranno essere prodotti. Le specifiche del database da progettare e delle funzionalità da implementare sono fornite nel Capitolo 2.

Il progetto deve essere svolto in gruppi composti da due studenti. Solo in casi eccezionali, a fronte di disagi motivati, si può valutare l'eventualità che un gruppo sia composto da tre studenti oppure da uno; nella sezione 2.8 verranno specificate le parti che sono obbligatorie a seconda della dimensione del gruppo.

Il lavoro da svolgere è articolato nelle seguenti fasi:

1. Analisi delle specifiche;
2. Progettazione concettuale tramite un 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 tramite introduzione di ridondanze;
7. Traduzione dello schema concettuale nel modello logico relazionale;
8. Analisi delle dipendenze funzionali ed eventuale normalizzazione dello schema;
9. Scrittura di uno script MySQL che crei il database e lo popoli;
10. Implementazione delle funzionalità di analisi dei dati.

Come risultato finale dell'attività, oltre al **database** e allo **script** contenente il codice per la sua creazione e popolazione e per la realizzazione delle funzionalità richieste, in sede d'esame dovrà essere consegnata una **documentazione** nella

quale vengono spiegate e giustificate chiaramente tutte le scelte effettuate fase per fase. Più in dettaglio, deve essere consegnata una copia digitale (in PDF) della documentazione, con pagine numerate e provvista di indice. Devono inoltre essere consegnate due copie (in PDF) del diagramma E-R, una non ristrutturata e una ristrutturata. Il diagramma E-R deve essere ordinato e chiaramente leggibile, deve essere realizzato su un foglio digitale di dimensione appropriata (tipicamente A3, A2 o formati più grandi) e il font utilizzato non dovrà essere più piccolo di 12 pt in ogni parte del diagramma. Si deve quindi usare il software di progettazione **impostando all'inizio il formato di pagina e, nel momento in cui si effettua la stampa cartacea (o in PDF), scegliere un formato carta (o di pagina digitale) dello stesso formato scelto nel software**. Per esempio, se nel software si è impostato il formato di pagina A3, quando si stampa si deve stampare in A3. Non usare opzioni "Adatta alla pagina", "Fit to page", "Fit to width", "Fit to whatever".

L'esame si svolgerà sotto forma di colloquio in cui entrambi gli studenti che compongono 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.

1.1 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.

1.2 Progettazione concettuale

Una volta scelta una strategia di progettazione concettuale fra quelle viste a lezione, il diagramma entità-relazione verrà prodotto incrementalmente. La notazione da utilizzare per il diagramma entità-relazione è quella vista a lezione, presente sia nelle slide che nel libro di testo. Non sono accettate notazioni alternative non trattate nel corso (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 [Draw.io](#), [Microsoft Visio](#), [OmniGraffle](#) [solo macOS], [ConceptDraw](#) [solo macOS], [Adobe Illustrator](#). I tool sottolineati sono stati utilizzati in modo preferenziale dagli studenti in edizioni precedenti del corso.

Il diagramma E-R prodotto come risultato da questa fase deve contenere: entità e associazioni; attributi di entità e associazioni; identificatori primari delle entità; cardinalità delle associazioni e degli attributi (se opzionali o multivalore); eventuali generalizzazioni, attributi composti e multivalore.

1.3 Ristrutturazione del diagramma E-R

La ristrutturazione del diagramma entità-relazione prevede che siano eliminate le generalizzazioni e gli attributi composti/multivalore. Le generalizzazioni, gli attributi composti e gli attributi multivalore devono essere eliminati tramite una delle opzioni viste durante il corso. In questa fase possono anche essere effettuati partizionamenti/accorpamenti di entità e associazioni, dove si ritengano utili.

Questa fase produce come risultato un diagramma E-R ristrutturato, direttamente traducibile in uno schema di database nel modello logico relazionale.

1.4 Individuazione di operazioni sui dati

Devono essere individuate almeno 8 operazioni significative da effettuare sui dati. Queste operazioni dovranno essere implementate in linguaggio MySQL. Le operazioni individuate possono essere interrogazioni, inserimenti, modifica o cancellazioni. Con il termine "significative" si fa riferimento a operazioni che contribuiscono in maniera apprezzabile a determinare le prestazioni del database durante il normale carico applicativo¹. Queste operazioni possono essere operazioni semplici ma molto frequenti, oppure operazioni più complesse di analisi dei dati che, anche se eseguite più di rado, possono appesantire considerevolmente il carico applicativo.

1.5 Analisi delle prestazioni delle operazioni

Nella documentazione che sarà oggetto di discussione in sede d'esame, per ogni operazione significativa individuata al punto precedente, devono essere riportati nell'ordine: una descrizione chiara dell'operazione; l'input (ciò che è noto a priori ed è dato in ingresso all'operazione); l'output (ciò che l'operazione produce); la porzione del diagramma E-R interessata dall'operazione; la porzione di tavola dei volumi (le righe) con i volumi delle entità e le relazioni coinvolte; la tavola degli accessi.

Per analizzare le prestazioni delle operazioni la tavola dei volumi dovrà essere compilata stimando il numero di occorrenze per ciascuna entità o relazione. Dopodiché, deve essere data una stima sulla frequenza giornaliera dell'esecuzione di ciascuna operazione. Affinché l'analisi sia sufficientemente indicativa, le stime devono essere fatte con giudizio. La documentazione dovrà contenere la tavola dei volumi e le assunzioni sulla frequenza delle operazioni.

Per ogni operazione scelta nel Paragrafo 1.4, la tavola degli accessi conterrà tante righe quanti sono i concetti (entità e relazioni) attraversati dall'operazione. Ogni riga della tavola degli accessi deve contenere, nell'ordine: un identificatore incrementale, il nome del concetto coinvolto, il tipo di concetto (entità o relazione), il tipo di accesso (lettura o scrittura), il numero di accessi necessari, la descrizione della motivazione per la quale si effettuano gli accessi, cioè quale attributo (o attributi) si sta leggendo/scrivendo.

Dall'analisi delle prestazioni di ciascuna operazione significativa può emergere che alcune potrebbero trarre beneficio dall'introduzione di ridondanze. Ogniqualvolta si introduce una ridondanza per una operazione, per tale operazione deve essere compilata anche la tavola degli accessi che mostra il numero di operazioni

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

elementari² da eseguire in presenza della ridondanza. Deve infine essere presa una decisione sul mantenere o no la ridondanza introdotta, in base al risparmio di accessi che essa comporta. Si faccia attenzione che, scelta un'operazione, una ridondanza ne comporta naturalmente un alleggerimento del carico, ma, d'altra parte, la ridondanza deve essere mantenuta aggiornata per essere utilizzata. Più in dettaglio, una ridondanza richiederà un'operazione di aggiornamento (o refresh), ogniqualvolta viene eseguita una modifica (update, insert o delete) della tabella sulla quale la ridondanza è basata.

Per decidere se mantenere o meno una ridondanza, va studiata anche l'operazione di aggiornamento, la sua frequenza, la sua modalità (immediate, deferred, on demand) e la sua complessità in termini di operazioni elementari. Ciò permette di calcolare il rapporto costo-beneficio, dove il beneficio è il risparmio di operazioni elementari che la ridondanza comporta nell'esecuzione di un'operazione significativa (quelle al Paragrafo 1.4), mentre il costo è dovuto alle operazioni elementari necessarie per mantenere coerente la ridondanza. In altre parole, il carico computazionale che si introduce per mantenere aggiornata la ridondanza deve essere giustificato dal beneficio che la ridondanza produce. Per le operazioni significative che coinvolgono ridondanze, la documentazione di progetto deve contenere l'analisi costo-beneficio descritta sotto in dettaglio.

Scelta un'operazione significativa target T , avente frequenza giornaliera f^T , nella tavola degli accessi si trova il numero di operazioni elementari o^T necessari all'esecuzione dell'operazione T . Il numero di operazioni elementari giornaliere è $n^T = f^T \cdot o^T$. A seguito dell'introduzione della ridondanza, il numero di operazioni elementari diventerà o_{RID}^T (dove $o_{RID}^T < o^T$) e il numero di operazioni giornaliere sarà $n_{RID}^T = f^T \cdot o_{RID}^T$. Quindi il numero di operazioni elementari risparmiate per eseguire T sarà: $\Delta = n^T - n_{RID}^T$.

L'operazione A che aggiorna la ridondanza avrà una frequenza giornaliera g^A e richiederà un numero di operazioni elementari o^A per l'aggiornamento e nella giornata avremo $n^A = g^A \cdot o^A$. La ridondanza è conveniente se il numero di operazioni elementari effettuate in presenza di ridondanza ($n_{RID}^T + n^A$) è inferiore al numero effettuato da T in assenza di ridondanza, cioè se $n^A < \Delta$.

Nella versione finale del database, è richiesta la presenza di almeno due ridondanze. Fra le otto operazioni significative del Paragrafo 1.4 ci devono essere un'operazione di lettura e una di scrittura che impattino con essa.

1.6 Progettazione logica

Il diagramma entità-relazione ristrutturato deve essere tradotto nel modello logico relazionale, producendo così lo schema del database come sarebbe prodotto dall'algoritmo di traduzione automatica, compresi tutti i vincoli di integrità referenziale necessari al suo corretto impiego. La traduzione del diagramma E-R ristrutturato nello schema del database non può contenere scelte progettuali, ma è solo il risultato dell'algoritmo di traduzione.

Eventuali vincoli di integrità generici devono essere implementati mediante check o trigger MySQL. È richiesta la presenza di almeno 2 vincoli di integrità generici.

²In questo ambito, un'operazione elementare è un accesso alla memoria. L'accesso in lettura comporta una operazione elementare; l'accesso in scrittura due operazioni elementari

1.7 Analisi delle dipendenze funzionali e normalizzazione

Per ciascuna relazione (tabella) individuata, devono essere ricercate tutte le dipendenze funzionali non banali. È richiesto che il database progettato sia in forma normale di Boyce-Codd. Qualora la base di dati non lo sia, occorre effettuare una opportuna decomposizione delle relazioni non in forma normale di Boyce-Codd.

Il database potrà contenere parti non normalizzate solo a causa della presenza di ridondanze appositamente introdotte per aumentare le prestazioni. Tali ridondanze devono essere chiaramente indicate nel diagramma E-R, per esempio usando un colore dedicato.

1.8 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 definite e poterne mostrare output sensati nella documentazione e durante la prova orale. Durante l'esame può infatti essere 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 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 di data analytics e di back-end 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 i dati a supporto delle funzionalità del sistema informativo di *mySmarthome*, un sistema che si occupa di gestione e ottimizzazione delle funzionalità di moderne case intelligenti, le cosiddette *smart home*.

In una smart home (vedi Fig. 3) sono disponibili molteplici servizi di gestione che migliorano l'efficienza energetica e il comfort degli abitanti. Per esempio, la temperatura delle varie stanze della casa si adatta ai cambiamenti climatici e alle esigenze delle persone che in determinato momento sono presenti. La gestione dell'irrigazione è ottimizzata tenendo in considerazione le condizioni atmosferiche. Le luci sono automaticamente regolate in base ai desideri di illuminazione delle persone ed ai vari momenti della giornata.

Grazie a questi moderni sistemi di gestione, i dispositivi smart e le fonti energetiche possono essere gestiti al meglio. In particolare, dal punto di vista energetico, i consumi possono essere ottimizzati promuovendo la collaborazione di vicinati di smart home, minimizzando così il ricorso a fonti energetiche non rinnovabili.

Le informazioni memorizzate nelle varie aree del database sono descritte dettagliatamente nei prossimi paragrafi.



Figura 1: Rappresentazione schematizzata delle varie funzionalità di una smart home

2.2 Area generale

Nell'area generale del database sono memorizzate le informazioni sugli utenti che abitano la casa e sulla struttura della casa, rappresentata dalle informazioni relative alle stanze, le loro caratteristiche e la loro ubicazione all'interno dell'edificio.

2.2.1 Accounting

Un abitante della smart home si può registrare al servizio accedendo alla app di *mySmartHome*. Per registrarsi, ogni utente deve indicare codice fiscale, nome, cognome, data di nascita e numero di telefono. L'utente viene memorizzato nel database assieme alla data in cui si è iscritto.

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

2.2.2 Topologia dell'edificio

La casa è suddivisa in stanze. Ciascuna stanza ha un nome, una larghezza, una lunghezza, un'altezza, ed è situata a un determinato piano dell'edificio. Più stanze della casa possono avere lo stesso nome. Per esempio, se una casa ha più camere, per tutte queste stanze il nome sarà camera. In ogni stanza sono presenti dispositivi ed elementi di condizionamento dell'aria con i quali gli utenti possono interagire tramite impostazioni personalizzate. Nelle seguenti sezioni sono spiegati i dettagli su questi dispositivi e i particolari dell'interazione con l'utente.

Ogni stanza della casa è caratterizzata da uno o più punti di accesso, generalmente porte. Una porta permette l'accesso a una stanza da un'altra stanza, dall'esterno, o anche da una parte esterna collegata alla casa. Per esempio, una stanza può affacciarsi su un terrazzo (o balcone), al quale si accede da una porta-finestra. Una stanza può anche avere una o più finestre. Finestre e portefinestre sono orientate verso un punto cardinale¹.

Porte e portefinestre rappresentano quindi punti di accesso a una stanza. In generale, una finestra non è propriamente da considerarsi un punto di accesso, bensì un punto di intrusione monitorato dal sistema di controllo degli accessi (si veda il Paragrafo 2.6).

Le informazioni sulle stanze della casa, sulla loro struttura, ubicazione e sui loro punti di accesso devono essere memorizzate nel database, ottenendo quindi la topologia dell'edificio.

2.3 Area dispositivi

Quest'area del database contiene le informazioni relative all'utilizzo dei dispositivi elettrici ed elettronici della smart home.

¹Considerare gli otto punti cardinali N, NE, NW, S, SE, SW, E, W.

2.3.1 Dispositivi e smart plug

I dispositivi (come elettrodomestici, televisori, ecc.) di una casa possono essere resi intelligenti tramite il collegamento a *smart plug* compatibili con i moderni sistemi di domotica come Alexa e Google Home. Una tipica smart plug con presa americana è mostrata in Fig. 2.

Una smart plug è una sorta di adattatore collegato alla presa elettrica, munito di collegamento wi-fi. Ogni dispositivo è quindi collegato alla sua smart plug che ne permette l'alimentazione e lo arricchisce di funzionalità, quali il controllo vocale (in presenza o remoto), oppure la gestione e il reporting dei dati d'uso, tramite app. Una smart plug è caratterizzata da un codice e uno stato che può essere *attivo* o *inattivo*. Se la smart plug è inattiva non è collegata ad alcun dispositivo; viceversa è attiva.



Figura 2: Una smart plug per il collegamento di dispositivi al sistema di gestione della smart home.

2.3.2 Interazione utente-dispositivo

La smart plug comunica al gestore centrale tutte le interazioni che l'utente ha col dispositivo, permettendo di creare un registro di utilizzo per ogni dispositivo. I dispositivi sono caratterizzati da un consumo energetico (in kW) che può essere fisso o variabile. I dispositivi a consumo energetico fisso sono di tipo ON/OFF. L'utente li accende in un determinato momento e li spegne successivamente. I dispositivi con consumo variabile possono invece essere accesi e regolati dall'utente in base alle sue preferenze. Il consumo energetico del dispositivo varia dipendentemente dalla regolazione. Tipicamente, la regolazione è digitale, cioè il dispositivo ha un numero finito di livelli di potenza. Per esempio, un termoventilatore può essere impostato a una certa temperatura — che corrisponde a un determinato consumo energetico — per un determinato periodo di tempo. Poi può essere nuovamente regolato, oppure l'utente può spegnere il dispositivo. Ci sono infine dispositivi caratterizzati da un ciclo di funzionamento non interrompibile, come per esempio la lavatrice. In questo caso, il dispositivo è caratterizzato da programmi con varie durate e vari livelli di consumo energetico. Per questi dispositivi, il database registra il consumo medio (in kW) per ogni programma.

Oltre che regolare il livello di potenza, l'utente può anche interagire con certi dispositivi impostandone una **partenza differita**. Per esempio, l'utente in un dato giorno può programmare la partenza della lavatrice, con un certo programma di lavaggio, per un determinato orario del giorno successivo.

Ogni volta che l'utente interagisce col dispositivo, il database tiene traccia di questa interazione. Tutti i dispositivi della smart home dotati di smart plug sono monitorati e **i dati relativi alle interazioni con l'utente sono memorizzati nel database²**.

Alcuni dispositivi sono usati da una sola persona, altri da più persone. Le interazioni con i dispositivi usati da più persone devono tenere traccia di chi ha usato il dispositivo, con quali regolazioni (se ce ne sono) e per quanto tempo.

2.4 Area energia

Quest'area del database contiene le informazioni relative all'utilizzo dell'energia della smart home.

2.4.1 Sorgenti energetiche

Le moderne smart home sono dotate di una o più sorgenti di energia rinnovabile come pannelli fotovoltaici (o anche piccoli generatori eolici). Lo svantaggio principale dell'energia rinnovabile consiste nel fatto che essa deve essere utilizzata nel momento in cui viene generata. L'energia prodotta e **non utilizzata può essere immessa nella rete elettrica**, oppure stoccata in batterie³. Inviando l'eccedenza sulla rete, il gestore effettua uno storno in bolletta, pagando quindi all'utente la quantità di energia immessa in rete. La Fig. 3 mostra il sistema di gestione dell'energia di una smart home.

Per poter immettere energia sulla rete elettrica, occorre installare un particolare tipo di contatore: il contatore bidirezionale. Questo contatore è dotato di un modulo che, in ogni momento, misura i flussi di elettricità in entrata e in uscita e li invia al gestore della smart home. Così, quando la smart home preleva energia dalla rete, il contatore conteggia un prelievo. D'altra parte, quando viene immessa energia in rete dal proprio impianto, il contatore misura una immissione.

2.4.2 Contabilizzazione e uso dell'energia

I contatori bidirezionali registrano informazioni sul passaggio di elettricità nelle **diverse fasce orarie⁴**. Questo perché l'energia ha generalmente un diverso prezzo di mercato nelle varie fasce orarie. Il database deve mantenere le informazioni relative ai consumi (o alle eccedenze energetiche) per ogni fascia oraria. Inoltre, il database registra **il livello di produzione energetica dei pannelli fotovoltaici durante tutta la giornata**.

²I dati sono inviati tramite la rete wi-fi direttamente al gestore della smart home, il quale li invierà al database.

³Molto spesso si preferisce inviare l'eccedenza di energia sulla rete, anziché immagazzinarla in batterie. Le batterie sono infatti molto costose e sofisticate, oltreché limitate dal punto di vista dell'energia accumulabile.

⁴Solitamente le fasce orarie sono F1, F2 ed F3 che corrispondono a mattino, pomeriggio e sera.

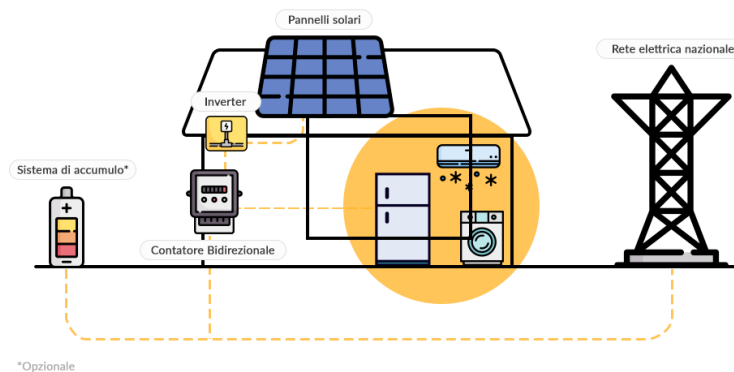


Figura 3: Sistema di gestione energetica di una smart home munita di pannelli fotovoltaici, sistema di accumulo (opzionale) e contatore bidirezionale. L'inverter trasforma la corrente elettrica continua (CC), in uscita dai pannelli solari, in corrente alternata (CA) a 220 Volt adatta all'immissione in rete e al consumo in home.

Il livello di produzione energetica di un pannello fotovoltaico varia dipendentemente dall'irraggiamento. Le informazioni relative alla produzione energetica possono essere memorizzate nel database al variare della quantità di luce solare da cui i pannelli sono irraggiati, oppure con una determinata cadenza. Nel primo caso, il database effettua memorizzazioni asincrone, ovverosia, una determinata funzionalità è invocata dal gestore della smart home per memorizzare nel database un cambio di produzione dei pannelli. D'altra parte, se si sceglie di memorizzare la produzione energetica dei pannelli con una certa cadenza, si effettua una memorizzazione sincrona. Per esempio, ogni 15 minuti, il database è aggiornato relativamente alla produzione energetica dei pannelli. È importante sottolineare che questi aggiornamenti non comportano l'eliminazione del vecchio dato di produzione energetica⁵. Il database contiene quindi tutte le informazioni relative alla produzione energetica dei pannelli, per tutti i giorni.

In generale, non è detto che il contratto di fornitura energetica della smart home preveda un prezzo dell'energia diverso per ogni fascia oraria. Ci sono piani tariffari (detti mono-orari) caratterizzati da uno stesso costo dell'energia per tutte le fasce orarie.

L'utente può dividere la giornata in fasce orarie (anche più di tre) e decidere in quali fasce utilizzare l'energia prodotta dai pannelli fotovoltaici (qualora ve ne sia), e in quali altre fasce reimmetterla nella rete elettrica. Il sistema di gestione della smart home genera suggerimenti di utilizzo dei dispositivi dipendentemente dalla disponibilità di energia rinnovabile. Per esempio, l'utente potrebbe ricevere una notifica nella sua app che lo invita ad avviare la lavatrice quando c'è un adeguato livello di energia prodotta dai pannelli fotovoltaici e si presume che questo livello di potenza si mantenga per buona parte della durata del ciclo di funzionamento. Queste considerazioni sono effettuate da una funzionalità di back-end⁶, da implementare,

⁵Quello relativo alla precedente memorizzazione, sia nel caso asincrono sia nel caso sincrono.

⁶Una funzionalità di back-end è una funzionalità lato server, direttamente implementata nel data tier, quindi nel DBMS. Un modo di realizzarla è implementare una stored procedure.

che, sulla base dei dati registrati in situazioni di irraggiamento simili memorizzate in precedenza, effettua delle previsioni sulla durata di un livello di irraggiamento dei pannelli. Quando l'utente riceve un suggerimento sulla app, può scegliere di accogliere il suggerimento (acconsentendo quindi all'avvio della lavatrice), oppure no. I suggerimenti e le scelte degli utenti sono memorizzati nel database.

2.5 Area comfort

In quest'area, il database memorizza le informazioni relative all'impostazione degli elementi di condizionamento dell'aria e di illuminazione, effettuate dagli utenti della smart home.

2.5.1 Trattamento aria

Gli abitanti di una smart home possono interagire con gli elementi di condizionamento tramite una app. Gli elementi di condizionamento si occupano di riscaldamento e raffreddamento dell'aria.

Quando un utente interagisce con un elemento di condizionamento, effettua un'impostazione. In particolare, ogni abitante⁷ può impostare una temperatura e un livello di umidità desiderata per ogni stanza della casa. Le impostazioni sono caratterizzate da un istante di inizio e fine. L'istante di fine di una impostazione coincide con l'istante di inizio di una nuova impostazione, o con lo spegnimento dell'elemento di condizionamento. Un'impostazione può anche essere ricorrente, in quel caso, l'utente decide di rendere l'impostazione attiva per determinati giorni della settimana, o in particolari giorni di determinati mesi. Per esempio, tutti i sabati e le domeniche la temperatura del soggiorno deve essere di 20 gradi in inverno e 27 d'estate.

Il mantenimento di una determinata temperatura in una data stanza dipende dalla temperatura interna iniziale e dal livello di dispersione dell'edificio. Il livello di efficienza energetica dell'edificio stabilisce, per ogni stanza, l'energia necessaria per riscaldare (o raffreddare) la stanza di 1 grado a partire da una temperatura di partenza interna e da una temperatura esterna.

Gli elementi di condizionamento sono programmabili, quindi l'utente può impostare avvii e spegnimenti differiti.

Il database memorizza i dati di interazione con gli elementi di condizionamento relativi ad avvii, spegnimenti, e alle impostazioni di temperatura effettuate da ogni utente per ogni stanza. Il database è inoltre dotato di una funzione di back-end, da implementare, capace di stimare i consumi derivanti da una determinata impostazione relativa a un elemento di condizionamento per un dato giorno, in base anche alla considerazione (o alla stima) dell'energia prodotta.

2.5.2 Smart lighting

Anche gli elementi di illuminazione di una casa intelligente possono essere impostati dagli utenti. In particolare, ogni stanza è dotata di uno o più elementi di illuminazione come faretti, spot, lampade, neon, e così via, che possono essere individualmente impostati generando configurazioni di smart lighting.

⁷L'abitante deve essere un utente registrato, come spiegato nel Paragrafo 2.2.

Ogni elemento di illuminazione è caratterizzato da una temperatura di colore⁸ e da un'intensità. Alcuni elementi illuminazione hanno un'intensità regolabile e/o una temperatura di colore regolabile. Ciò fa sì che l'utente possa configurare una combinazione di luce dei vari elementi di illuminazione presenti in una stanza che rispecchia le sue preferenze. L'utente può scegliere alcune impostazioni predefinite, o usate in precedenza, settando così delle combinazioni di intensità e temperatura di colore dei vari elementi di illuminazione di una stanza. Un esempio di configurazione di smart lighting con vari elementi di illuminazione è mostrata in Figura 4.



Figura 4: Una configurazione di smart lighting in una moderna abitazione. Sono visibili i vari elementi di illuminazione, impostati in modo da emettere luce di un determinato colore e di una determinata intensità.

Tutte le impostazioni relative agli elementi di illuminazione, una volta attivate dall'utente, si mantengono attive fintantoché l'utente non le disattiva con una nuova impostazione, o con lo spegnimento.

Il database deve memorizzare le informazioni sulle impostazioni degli elementi di illuminazione in ogni parte della giornata, e deve rendere anche possibile capire quali elementi di illuminazione vengono mantenuti spenti, in quali giorni, e per quanto tempo.

2.6 Area sicurezza

Quest'area del database mantiene le informazioni relative agli accessi alla smart home. Inoltre, in quest'area sono memorizzate le informazioni sullo stato dei serramenti.

2.6.1 Accessi e intrusioni

I sistemi di controllo degli accessi della smart home permettono di rilevare continuamente se qualcuno accede alle stanze, e quanto vi permane. Nel caso in cui un

⁸La temperatura di colore è misurata in Kelvin ed esprime quanto la luce sia calda o fredda. Temperature di colore più alte corrispondono a luci più fredde, tendenti quindi al bianco, come quelle dei neon.

utente registrato entri in una stanza della smart home, si parla di *accesso*. D'altra parte, se i sistemi di controllo degli accessi rilevano l'accesso di una persona non registrata, si parla di *intrusione*.

Sia gli accessi che le intrusioni sono caratterizzati da una persona (registrata o non registrata), una stanza e da una coppia di istanti temporali che indicano l'ingresso e l'uscita, e da un punto di accesso. In caso di intrusione, il punto di accesso (punto di intrusione) può anche essere anche una finestra⁹. Il sistema di controllo degli accessi scatta una foto dell'intruso che verrà memorizzata nel database.

Le informazioni continuamente memorizzate in quest'area hanno il duplice scopo di analizzare le abitudini degli abitanti della smart home, e di rilevare possibili intrusioni.

2.6.2 Serramenti

Le finestre, le portefinestre e le porte esterne dell'edificio sono caratterizzate da serramenti. I serramenti possono essere tapparelle, persiane o possono essere semplicemente rappresentati da porte più solide, come le porte blindate o i portoncini. Ogni serramento è unicamente abbinato a una finestra, una portafinestra o a una porta esterna. Un serramento ha uno stato che può essere alternativamente chiuso o aperto. Lo stato è impostato dall'utente e permane da un istante iniziale a un istante finale. Quando un utente cambia lo stato di un serramento, il serramento passa nello stato alternativo fino a un nuovo cambio di stato.

Le informazioni sui serramenti sono costantemente memorizzate nel database. Queste informazioni sono particolarmente utili in caso di intrusione, per ricostruire la dinamica dell'effrazione.

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 il comfort e l'efficienza energetica della smart home.

Gli studenti sono chiamati a implementare le funzionalità di analisi dei dati descritte nei seguenti paragrafi con spirito d'iniziativa. In questa fase della progettazione, sarà valutata la creatività e la capacità di proporre soluzioni che, seppur semplici, rappresentino validi strumenti di data analytics utili nel contesto applicativo della smart home descritto in questo documento di specifica.

Nella documentazione, le funzionalità implementate devono essere spiegate in dettaglio, in un paragrafo dedicato, mettendone in luce i punti di forza ed eventuali aspetti critici. Non è sufficiente consegnare il codice commentato.

2.7.1 Abitudini degli utenti tramite association rule learning

L'Association Rule Learning è uno strumento di Data Mining¹⁰ che permette di individuare le cosiddette *regole di associazione*, cioè relazioni di implicazione interessanti

⁹Si pensi per esempio al caso in cui un malintenzionato acceda abusivamente a un'abitazione introducendosi dalla finestra

¹⁰Il Data Mining è una disciplina della computer science che ha lo scopo di estrarre informazione nascosta in grandi volumi di dati.

nascoste nei dati.

Consideriamo un insieme $I = \{i_1, \dots, i_n\}$ di n item (oggetti) a valore binario e un insieme di transazioni (database) $D = \{t_1, \dots, t_m\}$. Ogni transazione è una riga di D , possiede un ID, ed è caratterizzata dalla presenza un sottoinsieme degli oggetti contenuti in I , detto *itemset*. Una *regola di associazione* (detta anche *regola associativa*) è un'implicazione $X \Rightarrow Y$, dove $X, Y \subset I$ e $X \cap Y = \emptyset$. Gli itemset X e Y sono l'antecedente e il conseguente della regola. Un itemset contenente k item è detto *k-itemset*.

Per fare un esempio, pensiamo al contesto degli acquisti effettuati in un supermercato. Sia $I = \{\text{latte}, \text{pane}, \text{burro}, \text{birra}\}$ l'insieme di tutti gli item. Il database con le transazioni (acquisti) è rappresentato nella Tabella 2.1, dove 1 indica la presenza di un oggetto in una transazione, e 0 l'assenza. Un esempio di regola di associazione potrebbe essere: $\{\text{burro}, \text{pane}\} \Rightarrow \{\text{latte}\}$. Questa regola indica che se un cliente acquista pane e burro, allora comprerà anche il latte.

ID	latte	pane	burro	birra
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Tabella 2.1: Un semplice database (tabella) di transazioni per la ricerca di regole di associazione.

Ma quanto è interessante questa regola? Per saperlo, occorre calcolare due indici detti *supporto* e *confidenza*. Data una regola $X \Rightarrow Y$, il supporto è la percentuale di transazioni del database D in cui compaiono sia gli item di X che quelli di Y , e si indica con $\text{supp}(X \cup Y)$; la confidenza misura quante volte gli item di Y appaiono in transazioni che contengono quelli in X . Dato il supporto $\text{supp}(X \cup Y)$ di una regola $X \Rightarrow Y$, la confidenza si calcola come:

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}, \quad (2.1)$$

dove $\text{supp}(X)$ è il numero di transazioni che contengono X .

Le regole associative con supporto e confidenza superiori a soglie prescelte sono dette *regole forti*, e sono quelle d'interesse perché esprimono implicazioni molto utili per studiare la realtà dei dati.

Un algoritmo per generare regole associative forti è l'algoritmo *Apriori*¹¹. Apriori genera dapprima un elenco di itemset in cui ciascun itemset supera un supporto minimo scelto dall'utente (questi itemset sono detti *large itemset*). Per ogni large itemset, l'algoritmo genera tutte le possibili regole associative ottenute partizionando gli item che compongono che compongono il large itemset in due itemset disgiunti: uno di questi costituisce l'antecedente, l'altro il conseguente della regola. Per ogni regola, l'algoritmo valuta se la confidenza supera la confidenza minima scelta dall'utente. In caso negativo, la regola viene scartata.

¹¹L'algoritmo Apriori non è il più efficiente, ma è comunque quello storicamente più famoso.

I passi eseguiti in loop dall'algorithmo per trovare l'insieme L dei large itemset sono i seguenti, dove $k = 2, \dots, |I|$:

1. ricavare l'insieme L_{k-1} dei large itemset composti da $(k-1)$ item (anche detti $(k-1)$ -itemset;
2. passo di Join: generare l'insieme C_k degli itemset candidati composti da k item;
3. passo di Pruning: qualunque $(k-1)$ -(itemset) non large non può essere un sottoinsieme large di un k -(itemset), perciò tali k -(itemset) non sono mantenuti in C_k .
4. trasferire in L gli itemset candidati rimasti in C_k

Per ogni large itemset $l \in L$ trovato, generare tutte le regole associative partizionando in due insiemi disgiunti il large itemset. Le regole saranno del tipo $l_1 \Rightarrow l_2$, dove $l_1, l_2 \subset l$ e $l_1 \cap l_2 = \emptyset$. Mantenere solo le regole con confidenza superiore alla soglia prestabilita (regole forti¹²).

La funzionalità di cui il database deve essere dotato deve trovare le regole associative forti per studiare le abitudini degli utenti relativamente all'utilizzo dei vari dispositivi o della permanenza nelle varie stanze. Nella documentazione, descrivere come si è implementata la funzionalità, cioè la modellizzazione degli item e la struttura di itemset e regole. La documentazione deve poi mostrare alcuni esempi di utilizzo con l'aiuto di screenshot e figure. Scegliere un solo contesto in cui trovare le regole (per esempio, i dispositivi, oppure gli elementi di condizionamento). Popolare la tabella che la funzionalità deve usare per generare le regole, tramite uno script SQL.

2.7.2 Ottimizzazione dei consumi energetici

A partire dallo stato attuale di produzione energetica derivante dai pannelli fotovoltaici e della richiesta energetica dell'edificio — derivante dalla somma dei consumi istantanei dei dispositivi attivi — questa funzionalità deve proporre piani di attivazione e regolazione dei dispositivi atti a migliorare l'efficienza energetica. Questi piani possono suggerire, per un dato giorno, quando attivare e come regolare i vari dispositivi nell'ottica di comfort ed efficienza energetica (minimizzazione dei consumi).

L'efficienza energetica si raggiunge, per esempio, anticipando o postponendo l'accensione di alcuni dispositivi dipendentemente dalla presenza di energia solare, che per l'utente è gratuita. Tutto questo ottimizza l'impiego dell'energia rinnovabile disponibile. Le moderne case intelligenti sfruttano anche le previsioni meteo per proporre piani di attivazione e regolazione dei dispositivi per il giorno successivo, spesso su scala oraria.

La funzionalità dovrebbe risolvere un problema di ottimizzazione molto complesso. In questo progetto, è sufficiente proporre un approccio euristico che sfrutta le conoscenze acquisite sui database per proporre piani di utilizzo dell'energia che sono intuitivamente convenienti per l'utente, in ottica di comfort ed efficienza energetica. Il comfort e l'efficienza energetica devono essere misurati con metriche scelte

¹²Sono regole forti perché l'algorithmo fornisce itemset $l \in L$ con supporto superiore alla soglia, e quindi ogni regola $l_1 \Rightarrow l_2$ generata partizionando l in due insiemi disgiunti l_1 ed l_2 ha il supporto di l (in simboli, $\text{supp}(l_1 \Rightarrow l_2) = \text{supp}(l)$)

(o introdotte) liberamente dagli studenti. Gli studenti possono proporre liberamente soluzioni, anche basate sul semplice buon senso. Le scelte fatte devono essere riportate nella documentazione.

2.7.3 Custom analytics

Gli studenti propongano una funzionalità analytics a piacere e spieghino nella documentazione l'importanza e l'impiego di tale funzionalità, nonché le scelte fatte per l'implementazione.

2.8 Dimensionamento del lavoro dei gruppi in base alla loro numerosità

Il progetto dovrebbe essere svolto da **2 studenti**. In casi eccezionali, i gruppi possono essere composti da 3 studenti, oppure da 1 solo studente. Questi casi devono rispettare quanto segue:

- i gruppi composti da 3 studenti svolgeranno il progetto nella sua interezza;
- i gruppi composti da 2 studenti possono non realizzare l'Area Sicurezza (Paragrafo 2.6) e la Custom analytics (Paragrafo 2.7.3).
- i gruppi composti da 1 studente possono non realizzare l'Area Sicurezza (Paragrafo 2.6) e l'Area Energia (Paragrafo 2.4). L'Analytics 1 (Paragrafo 2.7.1) è l'unica obbligatoria.