



**Specifiche di progetto per il corso di
Basi di dati
Anno Accademico 2019 – 2020
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 produzione	8
2.2.1	Prodotto	8
2.2.2	Assemblaggio	9
2.2.3	Processo produttivo	10
2.2.4	Stoccaggio	11
2.3	Area vendita	11
2.3.1	Account	12
2.3.2	Acquisti	12
2.3.3	Consegna	12
2.3.4	Recensioni	13
2.3.5	Garanzia	13
2.3.6	Resi	13
2.4	Area assistenza	14
2.4.1	Guasti	14
2.4.2	Assistenza virtuale	14
2.4.3	Assistenza fisica	14
2.5	Area smontaggio	16
2.5.1	Refurbishment	16
2.5.2	Smaltimento	17
2.6	Data analytics	18
2.6.1	Diagnosi intelligente dei guasti: CBR	18
2.6.2	Efficienza del processo	19
2.6.3	Custom analytics	19
2.7	Dimensionamento del lavoro dei gruppi in base alla loro numerosità	19

Capitolo 1

Descrizione delle fasi di progettazione

Si desidera progettare un database relazionale su DBMS Oracle MySQL che permetta di gestire i dati relativi al sistema informativo di una azienda tramite 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 2.7 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 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 A3, A2 o formati più grandi) e il font utilizzato non dovrà essere più piccolo di 12 pt in ogni parte del diagramma. Non effettuare mai rescaling quando si stampa il diagramma, altrimenti il font e i tracciati divengono più piccoli e illeggibili. Si deve quindi usare il software di progettazione **impostando all'inizio il formato di pagina e, nel momento in cui si effettua la stampa, il diagramma deve essere stampato su carta 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 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.

1.1 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 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](https://draw.io), OmniGraffle [solo macOS], ConceptDraw [solo macOS], [Microsoft Visio](https://www.microsoft.com/visio), [Adobe Illustrator](https://www.adobe.com/it/illustrator.html). I tool sottolineati sono stati utilizzati in modo preferenziale dagli studenti del nostro corso di laurea in edizioni precedenti del corso.

Il diagramma E-R prodotto come risultato da questa fase 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.

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 devono essere trasformate mediante le opportune traduzioni (accorpamenti, introduzione di associazioni...), e così pure gli attributi composti/multivalore, come appreso durante il corso.

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

1.4 Individuazione di operazioni sui dati

Devono essere individuate almeno 8 operazioni significative da effettuare sui dati che dovranno 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 alle operazioni che contribuiscono in maniera apprezzabile a determinare le prestazioni del database durante il normale carico applicativo¹.

1.5 Analisi delle prestazioni delle operazioni

Nella documentazione che sarà oggetto di discussione in sede d'esame, per ogni operazione significativa scelta, 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.

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

Deve poi essere compilata la tavola degli accessi relativa a ogni operazione, ottenendo così una stima degli accessi in memoria (operazione elementare) necessari per ogni lettura o scrittura a entità e relazioni richieste per la sua esecuzione. Nella documentazione, per ogni operazione realizzata, dovranno essere riportate la tavola dei volumi e degli accessi relative.

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 elementari eseguite in presenza della ridondanza. Deve infine essere presa una decisione sul mantenere o no la ridondanza introdotta, in base al risparmio di operazioni

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

elementari 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 inevitabilmente mantenuta aggiornata per essere utilizzata. Più in dettaglio, una ridondanza richiederà una operazione di aggiornamento (o refresh), ogniqualvolta viene eseguita una modifica (update, insert o delete) sulla tabella sulla quale la ridondanza è basata.

Al fine di decidere se mantenere o meno una ridondanza, deve quindi essere 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 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, sempre in termini di operazioni elementari. In altre parole, il carico computazionale che si introduce per mantenere aggiornata la ridondanza deve essere motivato dal beneficio che comporta la sua presenza.

In maggior dettaglio, scelta un'operazione target T , con frequenza giornaliera f^T , di cui si intende migliorare la performance, si deve compilare la tavola degli accessi per calcolare il numero di operazioni elementari o^T necessari alla sua esecuzione. Il numero di operazioni elementari giornaliere sarà $n^T = f^T \cdot o^T$. A seguito dell'introduzione della ridondanza il numero di operazioni elementari diventerà o_{RID}^T e il numero di operazioni giornaliere sarà $n_{RID}^T = f^T \cdot o_{RID}^T$. Quindi il numero di operazioni elementari risparmiate sarà $\Delta_{read} = n^T - n_{RID}^T$.

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 richiederà un numero di operazioni elementari o^A . Il numero giornaliero sarà quindi $n^A = g^A \cdot o^A$. La ridondanza è effettivamente 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. Formalmente, si mantiene la ridondanza se $n_{RID}^T + n^A < n^T$, ovvero $n^A < n^T - n_{RID}^T$.

Per tutte le operazioni che coinvolgono ridondanze, la documentazione di progetto deve contenere l'analisi costo-beneficio descritta precedentemente.

Nella versione finale del database, è richiesta la presenza di almeno due ridondanze e, per ciascuna ridondanza, devono essere presenti tra le operazioni realizzate 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.

Eventuali vincoli di integrità generici devono essere implementati mediante trigger MySQL, almeno 2 vincoli si richiede che siano presenti.

1.7 Analisi delle dipendenze funzionali e normalizzazione

Per ciascuna relazione (tabella) individuata, devono essere ricercate tutte le dipendenze funzionali non banali. E' richiesto che il database progettato sia in forma normale di Boyce-Codd.

Qualora la base di dati non lo sia (per relazioni che non coinvolgono ridondanze appositamente introdotte) occorre effettuare una opportuna decomposizione delle relazioni interessate.

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 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 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 trattare efficientemente i dati a supporto delle funzionalità del sistema informativo di *eDevice*, una grande impresa che si occupa di produzione, vendita, riparazione, ricondizionamento e smaltimento di apparecchiature elettroniche multimarca, in maniera ecosostenibile, altamente tecnologica e informatizzata.

Grazie a questi moderni sistemi di gestione, la pianificazione, l'effettuazione degli interventi e la misurazione delle performance sono assistite da una grande mole di dati utile a ottimizzare le varie attività, ad esempio grazie all'impiego di approcci basati su intelligenza computazionale e data mining.

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

2.2 Area produzione

In quest'area del database sono memorizzate le informazioni relative ai prodotti, al processo di produzione e allo stoccaggio.

2.2.1 Prodotto

Un prodotto elettronico (smartphone, notebook, lavatrice, ecc.) è caratterizzato da una marca, un modello, un codice seriale, un eventuale nome, un numero di facce su cui può essere appoggiato¹, un anno e un mese di produzione. Alcuni modelli di prodotto possono prevedere varianti, come per esempio le varianti di colore e dimensione degli smart device. Nel caso degli elettrodomestici, come le lavatrici, un esempio di variante può essere la diversa capacità del cestello. Esistono infatti lavatrici che possono lavare diversi quantitativi massimi di biancheria, espressi in chilogrammi.

Un prodotto è costituito da più parti, come mostra la vista esplosa in Figura 1. Ogni *parte* è caratterizzata da un codice, un nome, un prezzo, un peso, e da un insieme di materiali di cui è composta. Per ogni materiale (rame, alluminio, ferro...)

¹Una lavatrice è un parallelepipedo rettangolo con sei facce di appoggio; un tablet ha due facce d'appoggio (superiore e inferiore).

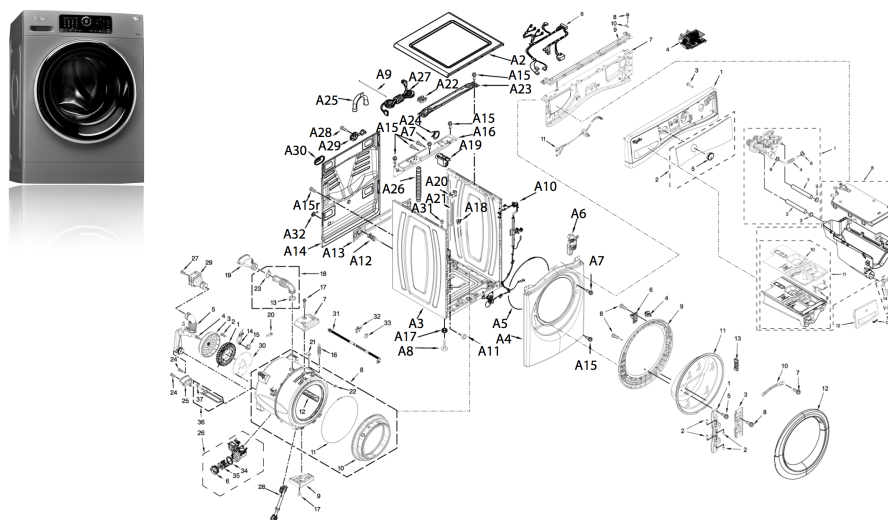


Figura 1: Diagramma esploso delle parti di una moderna lavabiancheria.

di cui è composta una parte, è necessario conoscere il relativo quantitativo, espresso in grammi. Ogni materiale ha un valore (in Euro) al chilogrammo. Ciascuna parte ha un coefficiente di svalutazione che permette di calcolarne il valore residuo, al momento dello smaltimento del prodotto.

2.2.2 Assemblaggio

Durante l'assemblaggio, ciascuna parte del prodotto è fissata a un'altra parte (o a più parti) mediante elementi di giunzione, o saldature. Ogni elemento di giunzione ha un tipo (per esempio, vite, rivetto, bullone, fascetta, ma anche altri), e alcune caratteristiche che lo contraddistinguono dagli altri. Le caratteristiche sono diverse da un elemento di giunzione all'altro. Per esempio, le caratteristiche di una vite possono essere la lunghezza, il diametro e il passo di filettatura; una guarnizione è invece caratterizzata dal materiale, dal diametro e dallo spessore. Ogni caratteristica ha un'unità di misura.

I prodotti sono solitamente montati in una linea di assemblaggio, eseguendo una *sequenza* di operazioni. Ciascuna operazione aggiunge una parte al prodotto, agendo su una delle sue facce. Un'operazione è codificata tramite un identificatore intero, ha un nome, ed è eseguita usando uno o più utensili, per esempio, cacciavite, pinze, ecc. Ogni parte può essere montata solo se una o più altre parti sono già state montate. Questo vale anche per le operazioni di smontaggio dei prodotti elettronici effettuata durante gli interventi di assistenza, ricondizionamento o smaltimento ecosostenibile: ciascuna parte può essere smontata (quindi estratta) dal prodotto solo se sono già state estratte altre parti. Questi vincoli di montaggio/smontaggio sono detti *vincoli di precedenza tecnologica*.

Non esiste un'unica sequenza per montare (o smontare) un prodotto. Nelle aziende produttrici, i prodotti sono assemblati seguendo sequenze altamente efficienti, che ottimizzano uno o più indicatori di performance, come per esempio la

minimizzazione del numero di volte che il prodotto deve essere ruotato durante il montaggio, la minimizzazione dei cambi di attrezzatura necessaria per operazioni sequenziali, e così via.

2.2.3 Processo produttivo

I prodotti elettronici sono prodotti in lotti. Ogni lotto è identificato da un codice, da una data di produzione, da una sede di produzione e dal prodotto fabbricato. Un lotto produce più unità di uno stesso prodotto. Un lotto può richiedere più giorni per essere prodotto integralmente — cioè per produrre tutte le unità da esso previste — a causa dei rallentamenti e/o imprevisti che si verificano durante la produzione. Il database deve quindi memorizzare sia la durata preventivata che la durata effettiva della produzione di ogni singolo lotto.

Prima di iniziare la produzione di un lotto, si deve decidere la sequenza di operazioni per assemblare le relative unità di prodotto. La sequenza è una fra le tante possibili, ed è scelta sulla base degli indicatori di performance che l'azienda desidera ottimizzare durante la produzione dello specifico lotto.



Figura 2: Linea di produzione allestita per l'assemblaggio di un lotto di lavatrici. Sono visibili le stazioni e gli operatori.

Deve essere implementata una funzionalità che permette la generazione di sequenze valide, cioè di sequenze che rispettano le precedenze tecnologiche. Una volta decisa la sequenza da adottare per la produzione di un lotto, questa deve essere memorizzata nel database. La linea di produzione deve poi essere organizzata in stazioni. In ogni stazione, un operatore esegue un sottoinsieme delle operazioni della sequenza di produzione. Per ogni stazione, il database memorizza l'orientazione del prodotto² e l'operatore che deve eseguire le operazioni assegnate alla stazione. Le operazioni assegnate a una stazione non devono richiedere la rotazione del prodotto.

La linea è infine associata a un tempo T , espresso in minuti. T è il tempo massimo che l'operatore ha a disposizione per eseguire tutte le operazioni assegnate alla sua stazione, su ciascuna unità a cui lavora; questo tempo è lo stesso in tutte

²L'orientazione può per esempio essere espressa dalla faccia del prodotto che è appoggiata sul piano di lavoro.

le stazioni e non può essere superato in alcuna stazione. Ogni T minuti un nuovo prodotto esce quindi dalla linea di produzione.

Il tempo impiegato dall'operatore per eseguire le operazioni assegnate alla sua stazione può però variare, a causa di imprevisti e cali di performance. Se l'operatore di una data stazione non riesce a eseguire tutte le operazioni nel tempo T , deve scartare l'unità a cui sta lavorando³ e inserirla fra le unità perse. La produzione è quindi tanto più efficiente quante meno unità perse produce. Il database deve tenere traccia del numero di unità perse di ogni lotto, memorizzando l'insieme di operazioni non eseguite su di essa nell'ultima stazione prima dello scarto. Sulla base di questi dati, una funzionalità lato server deve consentire l'analisi dell'efficienza della linea di produzione. Lo studente può proporre degli indicatori di performance e costruire una o più funzioni per misurare la performance della produzione. Si noti che, al termine della produzione del lotto, le unità perse devono essere riprese e immesse nella linea di produzione per essere completate. Tutto questo comporta un calo delle performance dovuto ai ritardi nella produzione.

Naturalmente, lo studio della distribuzione delle unità perse fra le stazioni della linea di produzione può aiutare a capire se il tempo T fissato è troppo basso, oppure se ci sono criticità dovute a qualche operatore che magari non ha le abilità necessarie a eseguire in tempo le operazioni assegnate. Per questo motivo, è utile salvare nel database il valore medio e la varianza dei tempi di esecuzione di un set di operazioni campione da parte degli operatori, per esempio svitare una vite, agganciare una fascetta, disaccoppiare un pannello ecc. Questo aiuta ad assegnare ogni lavoratore alla stazione dove sono eseguite le operazioni per le quali l'operatore è più veloce.

2.2.4 Stoccaggio

I prodotti di un lotto sono stoccati in uno dei magazzini dell'azienda. Ogni magazzino ha un codice, una capienza in metri quadri e una predisposizione che lo rende adatto allo stoccaggio di determinate classi di prodotti. Per esempio, magazzini predisposti per lo stoccaggio di elettrodomestici hanno delle pavimentazioni caratterizzate da gettate di cemento più profonde e da solai più spessi, per garantire la stabilità.

Ciascun magazzino può contenere unità di lotti diversi. Il database deve memorizzare sempre l'attuale contenuto di ogni magazzino e l'ubicazione delle varie unità presenti, lotto per lotto. Il concetto di ubicazione può essere definito a piacere, per esempio partizionando logicamente i magazzini in aree, ciascuna delle quali è in una specifica posizione del magazzino e contiene unità di un unico lotto, o anche di più lotti, ma comunque tutte dello stesso prodotto.

Le informazioni sul contenuto dei magazzini non devono essere sovrascritte nel tempo. Deve essere possibile analizzare l'evoluzione temporale dello stato dei magazzini.

2.3 Area vendita

L'area vendita si occupa di memorizzare le informazioni relative al marketplace dal quale i clienti possono acquistare i prodotti.

³Questa unità è un'unità *incompleta* che non può proseguire nelle seguenti stazioni della linea di montaggio. Ogni operatore può eseguire solo le operazioni assegnate alla sua stazione. Il lavoro che non è stato possibile eseguire nelle stazioni precedenti non può essere recuperato.

2.3.1 Account

Un utente si iscrive accedendo all'apposita area del sito web, fornendo le sue informazioni di anagrafica. In dettaglio, ogni utente deve indicare 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 sistema crea quindi un nuovo 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.

2.3.2 Acquisti

Una volta in possesso di un account, il cliente può collegarsi allo store online ed effettuare ordini di acquisto.

Dopo aver selezionato il prodotto (o i prodotti) da acquistare — ed eventualmente anche il numero di unità per ogni prodotto — il cliente può creare un nuovo ordine. Un ordine è caratterizzato da un codice ordine, un codice cliente⁴, i codici dei prodotti acquistati, un istante temporale di effettuazione dell'ordine, e uno stato. Lo stato di un ordine passa attraverso i seguenti valori: *in processazione*, *in preparazione*, *spedito*, *evaso*. Un ordine deve necessariamente seguire la precedente sequenza di stati.

Se un prodotto non è presente a magazzino, l'utente può ugualmente effettuare un ordine di acquisto. L'ordine è in questo caso memorizzato nel database in stato *pendente*. Gli ordini pendenti non sottraggono credito al cliente. Quando la produzione di un lotto rende disponibili i prodotti relativi a ordini pendenti, tali ordini passano automaticamente in stato *in processazione*, e da lì negli altri stati, secondo l'ordine sopra descritto. Settimanalmente, alcune funzionalità di back-end confezionano dei report che analizzano le vendite e gli ordini pendenti. Tali report segnalano alla direzione quantità indicative di prodotti da produrre.

2.3.3 Consegna

Le unità di prodotto da spedire per ogni di acquisto sono selezionate da una procedura di back-end che dà priorità di evasione alle unità prodotte da più tempo. Le unità selezionate sono spedite all'indirizzo del cliente, se questo non specifica un indirizzo di consegna differente. Ogni spedizione è identificata da un codice, ha una data di consegna prevista, ed è tracciata segnalando il percorso seguito attraverso più hub, ovverosia centri smistamento. Lo stato di una spedizione può essere *spedita*, *in transito*, *in consegna* e *consegnata*. Quando la spedizione lascia l'ultimo hub passa nello stato *in consegna*. Dopo la consegna, la spedizione passa nello stato *consegnata*.

⁴Il codice cliente si può per semplicità pensare uguale al nome utente.

2.3.4 Recensioni

I prodotti acquistati in un dato ordine di acquisto possono essere recensiti dal cliente utilizzando un punteggio di qualità formato da vari giudizi che valutano un insieme di criteri quali affidabilità, esperienza d'uso, performance, e così via.

Ogni giudizio può essere pensato su una scala discreta, per esempio scarso, sufficiente, buono, ottimo. Un'alternativa è un giudizio dato usando un numero di stelle da 1 a 5.

Le recensioni includono opionalmente un campo testuale.

2.3.5 Garanzia

L'utente ha per legge 24 mesi di garanzia su ogni prodotto acquistato. La garanzia del produttore copre guasti che non sono generati da incuria o uso improprio delle apparecchiature.

L'azienda mette anche a disposizione formule di estensione della garanzia. Tali formule sono caratterizzate da un codice, un costo, e dal prodotto a cui possono essere applicate. Una data formula di estensione di garanzia può essere applicata a più prodotti.

Vi sono formule che semplicemente aumentano il periodo di validità della garanzia a 36 o 48 mesi. Altre, talora più convenienti, estendono il periodo di validità della garanzia, ma limitano la copertura a classi di guasti⁵. Per esempio, nel caso degli elettrodomestici, potrebbero essere disponibili formule che coprono guasti ai motori e all'elettronica, ma che non coprono la rottura di materiali plastici, o degli chassis.

2.3.6 Resi

Quando un prodotto viene venduto usando una piattaforma online⁶, il cliente ha un determinato numero di giorni per effettuare il reso. Una richiesta di reso ha un codice e si riferisce a un'unica unità acquistata. I prodotti restituiti dai clienti sono stoccati nei magazzini dell'azienda. In ogni momento, il database dà la possibilità di conoscere il numero di unità rese di ogni prodotto, e la relativa motivazione di reso in un insieme di motivazioni che l'azienda ritiene valide (ad es. *difetto estetico*, *difetto funzionale*, *insoddisfazione*); ogni motivazione ed ogni difetto hanno un codice, un nome e una descrizione. Il database deve memorizzare le informazioni per discernere le richieste di reso caratterizzate da una motivazione (malfunzionamento, rottura,...), da quelle effettuate semplicemente perché il cliente si è avvalso del diritto di recesso. Nel primo caso, il reso deve essere richiesto e approvato prima di effettuare la restituzione; nel secondo caso il reso è incondizionatamente accettato.

Le unità rese sono analizzate dall'azienda e, dopo aver valutato una serie di parametri di qualità, avviate al refurbishment.

⁵Una classe di guasti è un insieme di guasti che possono coinvolgere il prodotto. A tali guasti si applica la formula di estensione della garanzia.

⁶Sono i prodotti per esempio "rimandati indietro" quando si acquista su Amazon, sia perché non ci soddisfano oppure perché affetti da difetti estetici o vizi di conformità.

2.4 Area assistenza

Questa area del database contiene i dati relativi ai guasti e agli interventi di assistenza effettuati sui prodotti venduti dall'azienda.

2.4.1 Guasti

Quando un prodotto si guasta, il cliente può utilizzare un servizio presente sul sito web dell'azienda per richiedere assistenza. Un guasto è identificato da un codice, un nome, una descrizione e un insieme di codici d'errore. Due prodotti diversi che manifestano lo stesso guasto possono mostrare sul display due codici d'errore diversi. L'assistenza si divide in *virtuale* e *fisica*.

2.4.2 Assistenza virtuale

L'assistenza virtuale è pensata per rendere il cliente capace di effettuare una diagnosi dei guasti del prodotto, permettendogli così di risolverli autonomamente, se possibile. Questo tipo di assistenza si basa sull'interpretazione dei codici d'errore forniti dalle apparecchiature in caso di guasto, ma può anche basarsi sulla descrizione dei malfunzionamenti e/o guasti riscontrati dal cliente.

In dettaglio, ogni prodotto ha un insieme di codici di errore, ciascuno associato a un guasto. Per ogni codice di errore esistono uno o più rimedi. Un rimedio è caratterizzato da un codice e da una descrizione, e rappresenta un'azione che il cliente può eseguire autonomamente — come ad esempio la pulitura di un filtro di una lavatrice, l'esecuzione di una particolare procedura di reset su un tablet, la simultanea pressione di un insieme di tasti, ecc.

È possibile che un'apparecchiatura si guasti senza mostrare codici di errore. In questo caso, l'assistenza virtuale è basata su una serie di informazioni che l'utente deve fornire per ottenere il rimedio. Dato il guasto, il sistema online pone una domanda, alla quale si può rispondere solo con un "sì" o un "no". Dipendentemente dalla risposta, l'applicativo mostra una nuova domanda. Questa procedura è ripetuta fintantoché non esiste un rimedio da applicare, memorizzato nel database. Se il rimedio non ha successo, l'utente lo esprime al sistema e quest'ultimo pone una nuova domanda. L'utente risponde nuovamente con "sì" o "no", finché il sistema non fornisce un nuovo rimedio. Qualora con questo tipo di auto-diagnosi non si riesca a riparare il dispositivo, c'è bisogno della riparazione/sostituzione di una o più parti del prodotto. Questa eventualità richiede necessariamente il ricorso all'assistenza fisica. Un esempio di assistenza virtuale per un guasto è mostrato in Figura 3.

2.4.3 Assistenza fisica

L'assistenza fisica prevede l'intervento di tecnici qualificati. Un intervento di assistenza fisica può essere effettuato presso uno dei centri assistenza dell'azienda oppure presso il domicilio del cliente, per esempio nel caso di un elettrodomestico. Gli interventi effettuati a domicilio possono comunque prevedere il successivo trasporto del prodotto presso un centro assistenza, dipendentemente dall'entità del guasto.

Il cliente fa richiesta di un intervento di assistenza sul portale web dell'azienda: ad ogni richiesta è associato un ticket, cioè a un codice che identifica univoca-

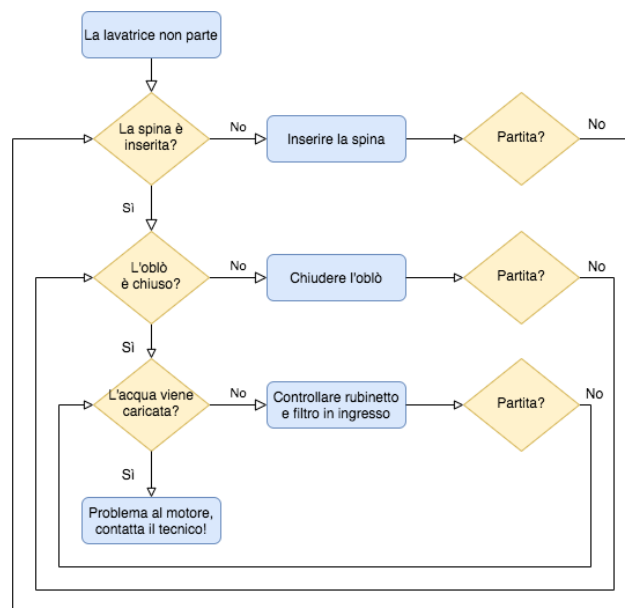


Figura 3: Esempio di procedura di assistenza virtuale per la mancata partenza di una lavatrice.

mente l'intervento. All'atto della richiesta, il cliente visualizza anche un insieme di possibili opzioni relative a giorno e fascia oraria in cui l'intervento può essere effettuato. Queste opzioni sono visualizzate dipendentemente dalle varie disponibilità dei tecnici. L'azienda crea gruppi di interventi giornalieri per ogni tecnico, e fa in modo che il tecnico faccia meno chilometri possibili per eseguire tutti gli interventi di una giornata. Una funzionalità lato server fa in modo che gli interventi siano settimanalmente associati ai vari tecnici, raggruppandoli per prossimità geografica.

Quando un tecnico si reca presso il domicilio del cliente per un intervento, effettua una diagnosi di guasto e fornisce un preventivo per la riparazione. Il preventivo è memorizzato nel database. Se e quando il preventivo è accettato⁷, si provvede alla riparazione. Se è necessario prelevare il prodotto per portarlo presso un centro assistenza dell'azienda, ciò viene fatto al momento dell'accettazione del preventivo.

Una riparazione prevede la sostituzione di una o più parti. Se la riparazione può essere effettuata presso il domicilio del cliente, le parti da sostituire vengono ordinate. L'ordine per le parti di ricambio è caratterizzato da un codice, una data, un tecnico che lo richiede, una data prevista di consegna, e una data effettiva di arrivo. Naturalmente, ogni parte di ricambio ha un prezzo. Quando le parti arrivano, l'azienda programma un secondo intervento presso il cliente, per eseguire l'effettiva riparazione. Se è stato necessario prelevare il prodotto per portarlo presso un centro assistenza dell'azienda, nel secondo intervento sarà solo prevista la consegna del prodotto riparato.

⁷Il preventivo può anche non essere immediatamente accettato dal cliente, il quale può riservarsi un po' di tempo per valutare la sua decisione.

Se l'intervento avviene entro il termine della garanzia⁸, la riparazione è a carico del produttore. Se effettuato fuori garanzia, l'intervento ha un costo dato dalla somma dei costi delle parti di ricambio più il costo orario della manodopera del tecnico, moltiplicato per le ore di lavoro. A fine intervento, il tecnico emette una ricevuta fiscale, la quale ha un codice univoco, riporta tutte le voci di costo, e la modalità di pagamento utilizzata.

Il pagamento può avvenire in contanti o con carte di debito/credito, direttamente al tecnico, il quale è provvisto di POS. L'intervento di riparazione ha una garanzia di un anno sui pezzi di ricambio e di sei mesi su eventuali altri guasti che coinvolgono le parti sostituite.

2.5 Area smontaggio

I prodotti oggetto di reso da parte dei clienti e i prodotti *end-of-life* — cioè i prodotti giunti a fine vita — sono soggetti a smontaggio.

Più in dettaglio, i prodotti oggetto di reso sono smontati, testati e nuovamente assemblati per essere reimmessi sul mercato come prodotti *ricondizionati* (detti anche *refurbished*), prodotti pari al nuovo venduti però a prezzi scontati. I prodotti *end-of-life* sono invece smontati per smaltire correttamente le parti di cui sono composti, in ottica di ecosostenibilità e riuso dei materiali.

2.5.1 Refurbishment

I prodotti restituiti dai clienti sono stoccati nei magazzini dell'azienda. Quando il numero di unità rese di un dato prodotto supera una determinata soglia⁹, l'azienda può configurare lo smontaggio di un lotto di unità, in ottica di ricondizionamento. Indipendentemente dalla motivazione di reso, ciascuna unità resa deve essere sottoposta a un *controllo generale*. Questo controllo è eseguito tramite una sequenza di test, che forma il *test tree* del prodotto. Un esempio di test tree è mostrato in Figura 4. La root rappresenta il controllo generale e il livello di nodi immediatamente sottostante contiene sei test. Un *test* è caratterizzato da un codice, un nome e una o più parti di cui il test verifica il funzionamento. Un test può avere successo o fallire. Se un test ha successo, si passa al successivo sullo stesso livello del test tree. Se un test fallisce, possono essere previsti sotto-test. I sotto-test hanno la stessa caratterizzazione data per i test, e possono a loro volta prevedere ulteriori sotto-test. Per esempio, se il secondo test in Figura 4 fallisce, sono previsti quattro sotto-test, il terzo dei quali prevede un ulteriore sotto-test. Quando un sotto-test fallisce¹⁰ e non prevede altri sotto-test, viene eseguita un'operazione di ricondizionamento.

Per chiarire, utilizziamo un esempio. Data una lavatrice, esempi di test possono essere *controllo del motore*, *controllo della resistenza*, *controllo del cestello*, eccetera. Se il test *controllo del motore* fallisce, il motore subisce altri sotto-test, per esempio, *controllo dell'avvolgimento*, *controllo dei cuscinetti*, *controllo dei giochi dell'albero*, e così via. Se, per esempio, il controllo dei cuscinetti è l'unico sotto-test che fallisce, e non prevede sotto-test, allora il motore viene smontato e i cuscinetti vengono sostituiti. A questo punto, il test del motore viene nuovamente eseguito, e se ha

⁸Oppure entro il termine di un'eventuale contratto di estensione della garanzia preventivamente attivato dal cliente a copertura dei danni sul prodotto.

⁹La soglia di unità varia da prodotto a prodotto.

¹⁰A qualsiasi livello del test tree.

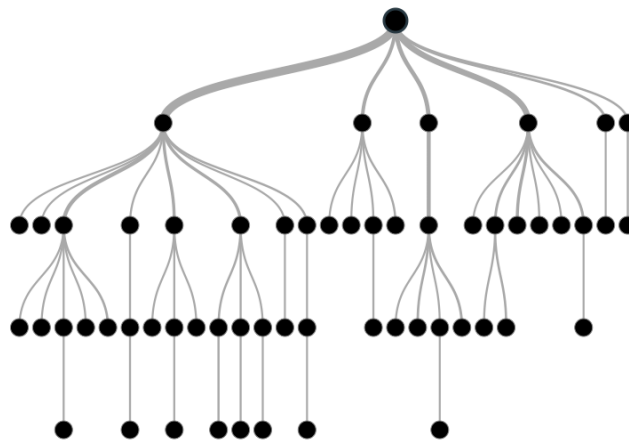


Figura 4: Esempio di test tree, dove la root è il "Controllo generale", mentre il primo livello contiene i test (*controllo del motore, controllo della resistenza, controllo del cestello...*), il secondo livello i sotto-test (*controllo dell'avvolgimento, controllo dei cuscinetti...*), e così via.

successo il motore può dirsi ricondizionato. Un'unità può dirsi ricondizionata quando sono state eseguite le operazioni di ricondizionamento di tutti i test (o sotto-test) falliti che non prevedono ulteriori sotto-test.

L'azienda può stabilire una sua politica secondo la quale, per esempio, se più del 70% dei sotto-test del test sul motore fallisce, è prevista la sostituzione dell'intero motore. Il database deve contenere l'albero dei test per tutti i prodotti, nonché la politica di sostituzione delle parti, qualora l'azienda le preveda. Per ogni unità ricondizionata, il database deve contenere tutte le informazioni relative al processo di ricondizionamento in termini di test (o sotto-test) falliti e parti sostituite.

Una volta ricondizionata, ogni unità è stoccata in un magazzino, ricodificata, e resa disponibile a un prezzo scontato nel negozio online. Per diversificare i prodotti ricondizionati dal punto di vista estetico, su taluni possono essere previsti interventi sulla scocca esterna. Per esempio, un lotto di smartphone ricondizionati potrebbe essere diviso in unità di categoria A e categoria B. Le prime si differenziano dalle seconde per il fatto di aver subito anche la sostituzione del case esterno. Le unità di categoria A saranno vendute a un prezzo scontato rispetto a un'unità nuova, ma comunque maggiore rispetto alle unità ricondizionate di categoria B dello stesso prodotto. Il database deve implementare questo vincolo di prezzo.

2.5.2 Smaltimento

I prodotti da smaltire subiscono il processo di smontaggio come i prodotti da ricondizionare, con la differenza che i prodotti da smaltire devono essere smontati per recuperare materiali e parti da riusare nella produzione di altri prodotti. Altri scopi includono l'estrazione di materiali che necessitano di un opportuno processo di smaltimento, per esempio i materiali pericolosi e gli inquinanti.

I prodotti destinati allo smaltimento sono smontati in lotti che contengono più unità dello stesso prodotto. Le linee di smontaggio per lo smaltimento dei prodotti end-of-life sono organizzate in stazioni di lavoro, così come descritto nel Paragrafo 2.2.3 per la produzione.

Mentre quando si assembla un prodotto è sempre necessario eseguire tutte le operazioni, durante lo smaltimento lo smontaggio può avvenire anche in modo incompleto, cioè senza disassemblare tutte le parti. Quando si smonta un prodotto end-of-life lo scopo principale è recuperare i materiali con il maggior valore di rivendita, oppure le parti in condizioni migliori per essere riutilizzate.

Per far sì che la profondità di smontaggio¹¹ sia ottimizzata, i lotti di smontaggio sono organizzati con unità in analoghe condizioni di usura. Una volta deciso quanto si deve smontare il prodotto, si individua la sequenza di operazioni di smontaggio da eseguire (si veda il Paragrafo 2.2.3 per i dettagli). Dovendo soddisfare le precedenti fra le operazioni, anche lo smontaggio può essere eseguito seguendo differenti sequenze di operazioni.

Stabilita la sequenza di smontaggio, per ogni unità smontata, c'è bisogno di memorizzare le parti e i materiali recuperati. Per le parti è sufficiente il codice, per i materiali occorre la quantità in chilogrammi. È inoltre necessario memorizzare il tempo che ogni operatore impiega ad arrivare a determinati target, per esempio, nel caso di una lavatrice, lo smontaggio del motore, l'estrazione del cestello e così via. Se questo tempo è troppo variabile, la linea non è ottimizzata. I prodotti che non riescono a soddisfare il tempo fissato T — il cui significato è analogo a quello descritto nel Paragrafo 2.2.3 — devono essere scartati. Più prodotti si scartano peggiore è l'efficienza della sequenza di smontaggio sullo specifico lotto. Il database deve memorizzare i prodotti scartati e l'ultima operazione eseguita prima dello scadere del tempo T .

2.6 Data 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.

Gli studenti possono 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 per la valutazione della performance aziendale.

Le funzionalità implementate devono essere spiegate in dettaglio nella documentazione, in un paragrafo dedicato, mettendo in luce i punti di forza. Non è sufficiente commentare il codice.

2.6.1 Diagnosi intelligente dei guasti: CBR

Questa funzionalità implementa un sistema intelligente *Case-Based Reasoning (CBR)* che supporta i tecnici nella diagnosi dei guasti. Il CBR è un paradigma dell'intelligenza artificiale che fornisce una modellizzazione del ragionamento umano nel risolvere problemi sfruttando una *base di conoscenza*, una memoria che contiene problemi precedentemente risolti, con le soluzioni che hanno portato alla loro risoluzione.

¹¹La profondità di smontaggio è il livello fino al quale si smonta un prodotto.

In particolare, si rappresenta un guasto tramite l'insieme dei suoi sintomi.

La funzionalità da progettare prende in ingresso i sintomi di un guasto e cerca *casì di guasto simili* precedentemente risolti all'interno della base di conoscenza. Questa è la fase di *retrieve*. Il concetto di somiglianza fra due casi di guasto è lasciato a discrezione dello studente. Più un caso precedentemente risolto è simile al nuovo caso, più è probabile che il rimedio che ha portato alla risoluzione del caso simile sia adatto a risolvere il nuovo caso. La funzionalità deve restituire in uscita i possibili rimedi per il nuovo caso, associati a uno score tanto maggiore quanto più è alta la somiglianza col caso di guasto precedente nel quale il rimedio è stato applicato con successo. Questa è la fase di *reuse*.

Il nuovo caso deve essere risolto, con l'aiuto dei rimedi suggeriti, eventualmente adattandoli, cioè modificandone o togliendone alcuni, o sostituendoli con altri. Questa è la fase di *revise*.

Se il revise ha comportato un insieme di rimedi che è ritenuto sufficientemente diverso da quelli già presenti per i casi già risolti, la funzionalità deve fare *retain*, ovvero sia aggiungere il nuovo caso alla base di conoscenza, cioè i suoi sintomi e i rimedi adottati.

Tutte le scelte fatte devono essere opportunamente descritte in un paragrafo della documentazione.

2.6.2 Efficienza del processo

Questa funzionalità deve analizzare la qualità dei processi produttivi, in termini di attinenza ai parametri di processo, come descritto nel Paragrafo 2.2.2, per esempio minimizzazione delle unità scartate, rispetto dei tempi, e così via.

Proporre uno o più indici per la valutazione della qualità del processo produttivo, e implementare una funzionalità analytics per effettuare automaticamente tale valutazione. Il modo di aggregare i dati e le scelte fatte devono essere spiegati in dettaglio nella documentazione.

2.6.3 Custom analytics

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

2.7 Dimensionamento del lavoro dei gruppi in base alla loro numerosità

Il progetto deve essere svolto da **2 studenti**. Le funzionalità relative allo smaltimento (Paragrafo 2.5.2) e l'analytics 3 (Paragrafo 2.6.3) sono opzionali.

In casi eccezionali, i gruppi possono essere composti da 3 studenti, oppure da 1 solo studente. Questi casi devono essere approvati dai docenti e devono rispettare quanto segue:

- i gruppi composti da 3 studenti svolgeranno il progetto nella sua interezza;
- i gruppi composti da 1 studente possono non realizzare l'area smontaggio e la parte relativa all'assistenza fisica (Paragrafo 2.4.3). L'analytics 1 (Paragrafo 2.6.1) è l'unica obbligatoria.