



DOCUMENTAZIONE DI PROGETTO

Anno Accademico 2019-2020

Descrizione dettagliata del sistema informativo progettato per la società:

eDevice

Arduini Luca

Casu Federico

De Marco Angelo

Indice

GLOSSARIO DEI TERMINI.....	4
PRESENTAZIONE DI ENTITÀ ED ASSOCIAZIONI	6
Area Produzione	6
Presentazione delle entità	6
Presentazione delle associazioni	9
Descrizione dettagliata di entità e associazioni.....	11
Area Vendita.....	20
Presentazione delle entità	20
Presentazione delle associazioni	21
Descrizione dettagliata di entità e associazioni.....	22
Area Assistenza.....	28
Presentazione delle entità	28
Presentazione delle associazioni	30
Descrizione dettagliata di entità e associazioni.....	32
Area Smontaggio	39
Presentazione delle entità	39
Presentazione delle associazioni	40
Descrizione dettagliata di entità e associazioni.....	42
RISTRUTTURAZIONE DIAGRAMMA	46
Traduzione generalizzazioni	46
Partizionamento di Entità	51
Elenco e analisi delle ridondanze	52
TAVOLA DEI VOLUMI.....	54
ANALISI DELLE OPERAZIONI	61
Elenco delle operazioni d’interesse	61
DATA ANALYTICS.....	79
Diagnosi intelligente dei guasti: CBR.....	79
Analisi dell’efficienza di una linea di produzione	81
Analisi delle vendite e pianificazione della produzione (custom analytics).....	83
TRADUZIONE IN MODELLO LOGICO	86
IMPLEMENTAZIONE SQL	98
Trigger generici.....	98
Business rule.....	103

Data Analytics	106
Retrieve & Reuse (Analytics 1).....	106
Analisi Performance Linea (Analytics 2)	107
Analisi delle vendite e pianificazione della produzione (custom analytics)	110
Procedure di Appoggio	118

GLOSSARIO DEI TERMINI

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Prodotto Elettronico	Un prodotto che l'azienda sceglie di produrre, e di inserire nel suo catalogo virtuale	Prodotto Smart Device	Parte Variante Linea Lotto
Assemblaggio	Fase che indica la composizione del prodotto, in termini di parti	Costituito Composizione	Parte Prodotto Materiale Operazione
Lotto	È l'entità fisica che si compone di più unità dello stesso prodotto		Magazzino UnitàPerse Linea Produzione Smaltimento
Unità	Con questo termine si identificano le singole unità presenti nei vari magazzini dell'azienda (sono unità singole, non lotti)		Lotto Reso Smaltimento Ricondizionamento Produzione Vendita
Magazzino	Un magazzino è una "semplice" raccolta di unità. Non c'entra potenzialmente niente con la sede della fabbrica che produce i lotti che gli vanno dentro.	Stoccaggio Ubicazione	Sede Lotto Unità
Account	Concetto che rappresenta solo l'identità virtuale dell'utente, e non quella fisica	Identità	Utente Ordine Reso
Ordine	Un ordine è un insieme di prodotti eventualmente acquistati da un account	Acquisto	Carrello Account
Spedizione	La spedizione di cui si parla non è una spedizione fisica, ma una serie di codifiche che può riguardare al massimo solo un ordine	Consegna Percorso	Account

Hub	Un hub è un centro non gestito dall'azienda, ma da terze parti	Centro Smistamento	Spedizione
Recensione	Una valutazione del cliente su un certo prodotto	Giudizio	Utente Prodotto
Garanzia	Possibilità di coprire alcune classi di guasti a diversi prezzi	Formula Copertura	Prodotto
Reso	Codifica delle richieste di reso dell'azienda	Richiesta Diritto di Recesso	Utente Motivazione
Rimedio	Il rimedio è una procedura eseguibile generalmente autonomamente dal tecnico o dall'utente stesso	Procedura Azione	Utente Tecnico Conoscenza Guasto
Sintomo	Un sintomo è un problema direttamente diagnosticato dall'utente, e può non essere specificato professionalmente. La richiesta iniziale in assistenza virtuale senza codice d'errore solitamente è rappresentata da uno o più sintomi	Problema	Assistenza Virtuale Domanda
Controllo Generale	Un controllo generale è un insieme di test eseguiti su un'unità ricondizionata.	Test-Tree Sequenza di Test	Unità Ricondizionate Test
Ricondizionamento	Una procedura di ricondizionamento coinvolge solo una parte dell'unità. Dunque, un'unità è detta ricondizionata quando sono stati eseguiti tutto un insieme di test	Sostituzione	Unità Ricondizionate Parte

PRESENTAZIONE DI ENTITÀ ED ASSOCIAZIONI

In questo capitolo verranno presentate le entità e le associazioni che compongono lo schema ER.

Si porta all'attenzione il fatto che tutte le scelte, le assunzioni, i collegamenti e persino eventuali nomi di cui sotto sono puramente relativi al diagramma non ristrutturato. Eventuali scelte, nuove assunzioni o modifiche dei nomi verranno ampiamente descritte in un paragrafo successivo.

Area Produzione

All'interno di quest'area risiedono tutte le entità e le associazioni che riguardano i prodotti, le loro modalità d'assemblaggio e di produzione in lotti, nonché lo stoccaggio delle unità nel magazzino.

Come si può notare nello schema, alcune entità (*Stazione, Operazione, Lotto, ...*) sono generalizzazioni tutte a due figli. Poiché si parla di produzione e smaltimento, analizzeremo in questa sezione soltanto le entità relative alla produzione, mentre lasceremo più avanti delucidazioni riguardanti le entità relative allo smaltimento.

Si noti che le entità e le associazioni sono scritte nello stesso modo nel quale appariranno su SQL, dunque non ci stupisce la scelta dell'utilizzo dell'UpperCamelCase e la completa assenza di accenti.

Presentazione delle entità

NOME	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Prodotto	I prodotti nel catalogo virtuale dell'azienda	CodProdotto Nome NumeroFacce DataCommercio	CodProdotto
CategoriaProdotto	Una lista di categorie di prodotti che l'azienda si preoccupa di produrre	Categoria Nome	Categoria
Variante	Insiemi di varianti possibili per i prodotti dell'azienda	CodVariante Descrizione	CodVariante
Varianza	Collegamenti possibili del prodotto e delle possibili varianti	Prezzo	CodProdotto CodVariante (entrambi identificatori esterni)
InfoVarianza	Informazioni, relative alla produzione e/o smaltimento, di ogni coppia prodotto-variante prodotta dall'azienda.	Ingombro Peso NumMINunitaLotto NumMINunitaRicond. NumMINunitaSmalt. PercentualeMAXricond.	CodProdotto CodVariante (entrambi identificatori esterni)
Parte	L'insieme delle parti, che, assemblate insieme, costituiscono i prodotti dell'azienda	CodParte Nome Prezzo Peso	CodParte

Materiale	I materiali attualmente conosciuti e utilizzati dall'azienda	Nome Valore CoeffSvalutazione	Nome
Operazione (ENTITA' PADRE)	Una riassuntiva descrizione delle operazioni dell'azienda	ID Descrizione Faccia Livello	ID CodProdotto CodVariante (quest'ultimi id. esterni)
Operazione Montaggio (ENTITA' FIGLIA)	Una riassuntiva descrizione delle operazioni da fare in atto di assemblaggio		
Sequenza	Una vera e propria legenda di tutte le sequenze di operazioni generate e mantenute dall'azienda	Codice	Codice Prodotto (l'ultimo è un identificatore esterno)
DatiSequenza	Esplosione delle sequenze, in termini di operazioni e loro ordine	NumOperazione	ID Prodotto Variante Sequenza (tutti identificatori esterni)
ClasseCampione	Identifica un Set di Operazioni campione con una lettera e una generale descrizione del tipo di operazioni contenute in esso	CodSet Descrizione	CodSet
Utensile	Strumenti di supporto allo svolgimento delle operazioni	Nome Tipologia	Nome Tipologia
Giunzione	Gli elementi che, insieme alle saldature, collegano due parti	CodiceGiunzione Tipo	CodiceGiunzione
Linea (ENTITA' PADRE)	Sequenze di stazioni organizzate alla produzione o allo smaltimento di un prodotto	CodLinea Tempo	CodLinea
LineaProduzione	Una vera e propria catena di montaggio di stazioni e operatori, nella quale si assemblano, in lotti, i prodotti dell'azienda		
Lotto (ENTITA' PADRE)	Insieme di un'unità dello stesso prodotto, indipendentemente che siano da stoccare in magazzino o da smontare	CodLotto	CodLotto

Lotto Produzione	Un lotto è un insieme di unità dello stesso prodotto, e costituisce il principio cardine del processo produttivo	(Aggiuntivi) DataProduzione DurataPreventivata DurataEffettiva UnitaPreviste UnitaEffettive Prima Produzione	
Stazione (ENTITA' PADRE)	Postazione nella quale uno o più operatori svolgono un insieme di operazione	CodStazione	CodStazione
Stazione Produzione	Postazione esclusivamente dedicata ad operazioni di montaggio	ClasseCampione TempoPrevisto Orientazione DaRecuperare	
UnitaPerse Produzione	Archivio delle unità che sono state perse da una stazione durante la produzione di un lotto	Quante	Lotto UltimaOperazione (identificatori esterni)
Dipendente (ENTITA' PADRE)	Insiemi di dipendenti di interesse alla base di dati: operatori e tecnici d'assistenza	CodFiscale Stipendio DataNascita Nome Cognome Citta	CodFiscale
Operatore (ENTITA' FIGLIA)	I dipendenti dell'azienda addetti a una linea di produzione		
Magazzino	Luogo in cui l'azienda stocka i suoi prodotti, lotto per lotto	CodMagazzino Sede	CodMagazzino
AreaMagazzino	Le aree logiche dei magazzini, organizzate e solitamente predisposte a una certa categoria di prodotto	Area Capienza CapienzaDisponibile	CodMagazzino (identificatore esterno)
Unita (ENTITA' PADRE)	Gli oggetti prodotti, venduti o rientrati in magazzino	UID	UID
Unita Disponibili (ENTITA' FIGLIA)	Le unità realmente in magazzino soggette a vendita		

Presentazione delle associazioni

NOME	DESCRIZIONE	COMPONENTI	ATTRIBUTI
VariantiAssumibili	Associazione che descrive il legame tra i prodotti e le varianti assumibili	Prodotto Varianza	
DescrizioneVariante	Associazione che lega ogni Varianza con la propria descrizione	Varianza Variante	
Dati	Associazione che lega le informazioni produttive di ogni coppia prodotto-variante.	Varianza InfoVarianza	
Composizione	L'esplosione del prodotto nelle sue parti.	Prodotto Parte	Pezzi
Struttura	Relazioni tra una parte e i materiali di cui è composta	Parte Materiale	Quantità
Riferita	Indica a quale prodotto si riferisce una certa operazione	Operazione Varianza	
Azione	Associazione che collega due parti mediante un'operazione	Parte (Base) Parte (Applicata) Operazione	Ripetizioni
Supporto	Espressione degli elementi di giunzione utilizzati in un'operazione	OperazioneMontaggio Giunzione	
Usa	Relazione che esprime gli utensili utilizzati allo svolgimento di un'operazione	Operazione Utensile	
SequenzaRelativa	Rappresenta la sequenza che si sceglie di "esplodere"	Sequenza DatiSequenza	
OperazioneRelativa	Associa la descrizione della sequenza alle operazioni che la compongono	DatiSequenza Operazione	
CampioneOperazione	Indica quali sono le operazioni scelte al costituirsi di un set di operazioni campione	ClasseCampione Operazione	

Scelta	Scelta della sequenza che una certa linea sceglie di adottare nel produrre un certo prodotto di una certa variante	Sequenza Linea	
Appartenenza	Appartenenza di una stazione a una linea di produzione	Stazione Linea	
Esecuzione	Associazione tra una stazione e le operazioni che la stessa esegue	Stazione Operazione	
LavoraIN	Operatori assegnati alla stazione attiva	Operatore Stazione	
Valutazione	Memorizza i tempi impiegati dagli operatori a svolgere i vari set di operazioni campione	Operatore ClasseCampione	TempoImpiegato
PerditaProduzione	Associazione che lega l'unità persa alla stazione che la perde	Stazione UnitaPerseProduzione	
PersaIN	Associa l'unità persa al lotto di cui quell'unità "dovrebbe" far parte	Lotto UnitaPerseProduzione	
UltimaOperazione	Esprime l'ultima operazione che è stata effettuata su un'unità prima che questa fosse scartata	UnitaPerseProduzione Operazione	
ProdottoDA	Relazione tra il lotto e la linea che lo produce	Linea Lotto	
Stoccaggio	Stoccaggio dei lotti all'interno delle aree apposite situate in un magazzino	Lotto AreaMagazzino	
Suddivisione	Associazione che esprime le possibili aree di un certo magazzino	AreaMagazzino Magazzino	
Predisposizione	Esprime la possibile predisposizione di un'area a una certa categoria di prodotto	AreaMagazzino CategoriaProdotto	
Provenienza Lotto	Rappresentazione dell'appartenenza di un'unità ad un certo lotto.	LottoProduzione UnitaDisponibili	

Tipo	Associazione che permette di ricavare la tipologia di prodotto e la variante per ogni unità	Varianza Unità	
------	---	-------------------	--

Descrizione dettagliata di entità e associazioni

Entità Prodotto

I prodotti venduti dall'azienda sono *elettronici* e hanno tutti un codice seriale che li identifica in maniera univoca (questo codice, come si vedrà successivamente, non è in questa entità). Ogni prodotto ha un **nome**, che nella maggior parte dei casi è costituito dal nome generale che un'alta percentuale di clientela vi assegna (ad esempio, molti chiameranno uno specifico computer semplicemente "*computer*"). Ogni prodotto ha poi modello, e numero di facce su cui può essere appoggiato [Specifiche, Par 2.2.1]. Si conviene infine che l'attributo **DataCommercio** indichi, nel formato "YYYY-MM" il mese e l'anno in cui un certo prodotto è entrato in commercio.

Entità CategoriaProdotto

Con Categoria Prodotto si intende la macro-classe cui è riferito un oggetto (telefonia, elettrodomestico, etc.). L'attributo **categoria** è un codice univoco, mentre *Nome* rappresenta una descrizione intuitiva e immediata della stessa.

Entità Variante

I prodotti dell'azienda sono catalogati e diversificati mediante *Varianti*, ovvero piccole modifiche alla versione base (che, per semplicità, si suppone una vera e propria variante codificata) che rendono l'oggetto diverso funzionalmente o esteticamente.

Ne sono esempi le varianti di colore per un telefono, o le varianti di grandezza del cestello per una lavatrice. Si noti bene che nell'entità "Variante" non è chiaro quale prodotto abbia quale variante, ma ci si limita ad una **Descrizione della Variante in sé** (mediante l'attributo *Descrizione*).

Entità Varianza e InfoVarianza

L'entità Varianza descrive i legami che esistono tra un prodotto e le possibili varianti nel quale il prodotto può essere commercializzato ad un certo prezzo (attributo **Prezzo**). Quest'entità ha bisogno, per sussistere, del codice di un prodotto e del codice di una variante (si rende chiaro l'utilizzo di due identificatori esterni). Quest'entità è pensata per l'**accesso comune**, che serva alla volontà più o meno intensa del cliente di voler *sfogliare (virtualmente)* un catalogo con dei prezzi.

D'ora in poi riferiremo sempre la coppia (prodotto ; variante) col **Varianza**.

L'entità InfoVarianza descrive invece una serie di informazioni riguardanti il lato operativo dell'azienda. Per ogni coppia di prodotto-variante vengono memorizzate le informazioni riguardanti il peso e l'ingombro (attributi *peso* e *ingombro*), il numero minimo di unità necessaria a comporre un lotto di produzione/ricondizionamento/smaltimento (attributi *NumMINunitaLotto*, *NumMINunitaRicondizionamento*, *NumMINunitaSmaltimento*) ed infine la percentuale massima di parti da sostituire oltre il quale non è più conveniente ricondizionare un'unità di quella coppia prodotto-variante (attributo *PercentualeMAXricondizionamento*).

Entità Parte

È un elemento *semplice*, che da solo non riesce a costituire un intero prodotto ("*un prodotto è costituito da più parti*") [Specifiche, Pag.8, Paragrafo 2.2.1]).

P.S: Non è detto che l'entità parte debba costituirsi di sole parti soggette a montaggio. Possono essere presenti, talvolta, parti davvero elementari che costituiscono dei *pezzi di ricambio*. Questi verranno usati nell'area Assistenza e nell'area Smontaggio!

Entità Materiale

L'azienda conosce, per esperienza diretta o meno, un certo numero di materiali, che vengono memorizzati mediante un **Nome** e un **Valore** (non esistono comunque due materiali con lo stesso nome).

I materiali costituiscono le parti che l'azienda utilizza per comporre prodotti, e possono essere recuperati, in parte, mediante un'operazione di smaltimento [*"Ciascuna parte ha un coefficiente di svalutazione, che permette di calcolarne il valore residuo, al momento dello smaltimento del prodotto"* Specifiche, Pag. 9, Par. 2.2.1]. Si discute delle operazioni di smaltimento nella parte relativa ad esso.

Entità Operazione (ENTITA' PADRE)

È un'azione, spesso elementare, che coinvolge **esattamente due parti** (si assume che il coinvolgimento di una terza richieda più operazioni). Le operazioni sono identificate mediante un ID Numerico.

Un'operazione è unica per varianza.

Notiamo che l'attributo **livello** descrive i vincoli di precedenza tecnologica necessari a descrivere in quale ordine possono essere montate le varie parti di un prodotto. Un esempio è che nessuna operazione con livello A può essere effettuata se esiste un'operazione di livello $< A$ da fare!

Le operazioni di **montaggio** assemblano una parte seguendo il vincolo di cui sopra.

Entità Sequenza, DatiSequenza

L'entità sequenza racchiude tutte le informazioni relative ad una sequenza di montaggio (o smontaggio). Una sequenza è identificata univocamente da un codice e da una varianza a cui si riferisce. Le informazioni tecniche riguardanti una sequenza sono contenute nell'entità DatiSequenza, che ha tra le altre cose il compito di descrivere il *numero* di una certa operazione in una certa sequenza. Il legame tra le due entità appena presentate è rappresentato dall'associazione SequenzaRelativa.

Entità Utensile

È un oggetto che aiuta l'operatore a compiere una certa operazione. Ogni utensile ha un **nome** (cacciavite, trapano) e un eventuale **tipologia** che costituisce un breve sunto delle caratteristiche dell'utensile (cacciavite a croce, taglio, trapano con una certa punta piuttosto che un'altra).

Entità Giunzione

Necessari (o quasi) a collegare due parti durante un'operazione di montaggio. Ogni elemento di giunzione ha un codice, il **CodGiunzione** e un **tipo** espresso col nome "commerciale" con il quale indichiamo tali elementi (vite, dado, fascetta, guarnizione...) seguito da un sunto delle specifiche dell'oggetto.

Entità Linea (ENTITA' PADRE)

Una Linea è l'insieme di operatori e stazioni che si preoccupano di assemblare (o smontare) un certo prodotto. Una linea è identificata da un codice, **CodLinea**. Ogni linea è poi caratterizzata da un **tempo** T, che indica il tempo, in minuti, che ogni stazione ha a disposizione per completare le operazioni a lei assegnate [*"Questo tempo è lo stesso in tutte le stazioni, e non può essere superato in nessuna stazione"* Specifiche Pag.11 Par 2.2.3].

Il tempo di una linea è un indicatore di estrema importanza nella valutazione dell'efficienza della linea stessa! Una linea può preoccuparsi di *una varianza alla volta*.

Potrebbe accadere che una Linea decida di cambiare sequenza di produzione, ma potrebbe anche decidere di cambiare varianza da produrre. A livello di database, il cambio di varianza non è permesso, e si rende necessaria la ricodifica della linea (anche se fisicamente resta unica).

È presente l'entità **LineaProduzione**, caratterizzata dal fatto che si preoccupa esclusivamente di montare un prodotto, e mai di smontarlo!

Entità Lotto

Un lotto è un insieme di unità dello stesso prodotto, e presenta caratteristiche diverse a seconda che esso sia di Produzione o di Smaltimento. Ad ogni modo, tutti i lotti sono caratterizzati da un codice, **CodLotto**.

Entità LottoProduzione

La linea di produzione produce unità in lotti, ovvero insiemi (imballati secondo regole aziendali) di unità dello stesso prodotto.

L'azienda decide a priori il numero di unità previste dal lotto (solitamente seguendo il numero consigliato in *InfoVarianza*), ma, a causa di rallentamenti nella produzione, le unità potrebbero essere inferiori (**mai superiori**). Stesso discorso vale per la durata della produzione del lotto, che potrebbe essere sia inferiore che superiore a quella prevista. La durata prevista e quella effettiva sono espresse in ore.

Ogni volta che una linea si svuota, e termina la produzione di un lotto, possono accadere principalmente due cose:

- Dopo una pausa di un certo tempo, la linea riprende con la produzione di un nuovo lotto
- La linea chiude definitivamente

Tuttavia, può accadere che alcune stazioni abbiano ancora unità da recuperare, che a livello informativo si traduce con $UnitaEffettive <> UnitaPreviste$.

Quando questo accade, le regole aziendali prevedono che quelle unità vengano rimesse nella Linea.

Il ciclo si ripete, fin quando tutte le stazioni associate alla linea che ha prodotto il lotto non hanno nulla da recuperare. Quando questo accade, il numero di $UnitaEffettive$ diventa uguale al numero di $UnitaPreviste$.

Prima Produzione è un attributo che si riferisce alla prima differenza tra $Unita Effettive$ e $Unità Previste$ (ovvero la differenza al primo ciclo di produzione). È un indicatore essenziale all'analisi della linea!

L'attributo $DurataEffettiva$ ha cardinalità (0,1) in quanto si conoscerà il suo valore solo alla fine della sua produzione, e fino a quel momento avrà valore NULL.

Entità Stazione e UnitaPerseProduzione

Una stazione è l'unità di lavoro elementare dell'azienda. Essa compie un certo numero di operazioni prefissate alla realizzazione di un certo lotto di prodotto.

L'attributo **orientazione** indica la faccia d'appoggio del "prodotto" che arriva alla stazione, e sussiste il seguente vincolo: *'Le operazioni di una stazione devono avere attributo **faccia** uguale all'attributo **orientazione**.*

Se così non fosse, il prodotto verrebbe ruotato nelle stazioni, ma questo non succede mai, in nessun caso [*"Le operazioni assegnate a una stazione non devono richiedere la rotazione del prodotto"* Specifiche Pag. 10, Par. 2.2.3]

Ogni stazione potrebbe impiegare più del tempo T previsto dalla Linea per effettuare le operazioni previste. In quel caso si dice che *l'unità è persa*. Quando un'unità è persa, l'entità **UnitaPerseProduzione** si arricchisce, memorizzando la stazione che ha perso l'unità, *l'ultima operazione*, in termini di codici, effettuata dalla stazione e il lotto d'interesse.

La scelta di identificare le perdite con CodLotto e UltimaOperazione deriva dal fatto che durante la produzione di un lotto posso naturalmente perdere più unità in "tempi" diversi. Qualora si perdessero più unità nella produzione dello stesso lotto, e tutte si fermassero alla stessa operazione, concorre in aiuto l'attributo **Quante**, che indica quante unità di quel lotto si sono fermate a quella operazione!

Infine, l'attributo **TempoPrevisto** in Stazione è il tempo previsto a svolgere le operazioni assegnate a quella stazione (per dettagli su come questo tempo è assegnato, visionare il paragrafo relativo alla Analytics 2).

Entità Dipendente (ENTITA' PADRE)

Sono i dipendenti d'interesse dell'azienda, e sono identificati mediante un **CodFiscale**, da cui dipende tutta l'anagrafica che l'azienda sceglie di memorizzare. Racchiude in sé una serie di informazioni anagrafiche relative alle persone che lavorano nell'azienda. Il significato degli attributi è il medesimo che si darebbe nella vita reale.

I dipendenti possono essere **operatori** (che si preoccupano della produzione o dello smontaggio di un prodotto), o **tecnici**, che si preoccupano delle richieste di assistenza all'azienda.

Entità Magazzino

Descrive grossolanamente i magazzini a cui l'azienda fa riferimento. Se ne riporta sempre una **sede** (oltre all'identificativo CodMagazzino).

Anche se non ne esiste un vincolo, e si è liberi di fare diversamente, spesso e volentieri i lotti prodotti da una linea di una certa sede, saranno conservati in magazzini della stessa sede, a meno che egli non sia pieno.

Ogni magazzino è suddiviso in Aree (si veda sotto).

Entità AreaMagazzino

L'entità AreaMagazzino rappresenta la suddivisione logica, in aree, dei magazzini. Queste aree sono organizzate e solitamente predisposte a una certa categoria di prodotto (si guardi l'associazione *Predisposizione*).

Leggere le Aree è importante: queste hanno infatti gli attributi capienza e capienza disponibile che, se sfruttati con criterio, possono restituire informazioni sull'interno magazzino. Un'area ha bisogno poi di un identificativo esterno, appunto magazzino.

Entità Unità

Una delle entità più importanti dell'azienda: rappresenta le unità concrete che sono state, o sono ancora, conservate in magazzino.

Di un'unità è utile conoscere parecchie informazioni, poiché saranno oggetto di vendite, reso, ricondizionamento, e tanto altro.

A questo proposito si è scelto di produrre per esse una generalizzazione (si veda in seguito).

Ogni unità ha uno **UID** (*Unity Identification Number*) che corrisponde a un codice seriale univoco per ognuna.

Si noti che, anche se può sembrare banale sottolinearlo, unità della stessa varianza hanno **sempre** le stesse caratteristiche.

Entità UnitàDisponibili

Figlia della generalizzazione *Unità*, non ha particolari caratteristiche, se non rappresentare le unità che sono **realmente in magazzino**, pronte alla vendita.

Associazione VariantiAssumibili

Cardinalità:

- (1, N) da parte di Prodotto, che può legarsi a più istanze di varianza
- (1, 1) da parte di Varianza, poiché un'istanza di varianza si lega ad uno e un solo prodotto

Associazione DescrizioneVariante

Cardinalità:

- (0, N) da parte di Variante, perché una variante potrebbe non essere definita per un certo prodotto oppure essere definita per più prodotti.
- (1, 1) da parte di Varianza, poiché un'istanza di questi può riferirsi solo ad una variante (in termini di descrizione)

Associazione Dati

Cardinalità:

- (1, 1) da parte di Varianza visto che una coppia prodotto-variante ha un set univoco di informazioni relative alla produzione.
- (1, 1) da parte di InfoVarianza perché ogni set d'informazione è associato ad una sola coppia prodotto-variante.

Associazione Composizione

Cardinalità:

- (1, N) da parte di Prodotto, poiché è composto sempre da almeno una parte
- (1, N) da parte di Parte, poiché essa può essere condivisa fra più prodotti (ad esempio, due telefoni potrebbero comporsi dello stesso e identico tipo di fotocamera)

A quest'associazione è scelto di aggiungersi l'attributo **Pezzi**, che indica quante unità di quella parte partecipano alla composizione di un certo prodotto.

Associazione Struttura

Cardinalità:

- (1, N) dal lato di Parte, poiché una parte può essere composta da più di un materiale
- (0, N) dal lato di Materiale, poiché un materiale potrebbe soltanto essere conosciuto, o recuperato, ma senza essere contenuto in nessuna parte.

Mediante l'attributo **quantità** è possibile conoscere il quantitativo in grammi dei materiali che costituiscono la parte d'interesse.

Associazione Riferita

Determina la varianza che l'operazione si preoccupa di contribuire a realizzare / smaltire.

Cardinalità:

- (1, 1) da parte di operazione, poiché si riferisce ad un'unica varianza (anche in casi di somiglianza di descrizione e/o parti coinvolte, il codice è comunque sempre diverso da varianza a varianza).
- (1, N) da parte di varianza, che ha ovviamente almeno un'operazione che ha essa stessa come oggetto.

Associazione Azione

Cardinalità:

- (0, N) in quanto una parte *applicata* può essere applicata a una *base* o meno, e in generale potrebbe essere coinvolta in più operazioni
- (1, N) da parte di *base* poiché essa può comparire in più situazioni di montaggio
- (1, 1) da parte di *Operazione* poiché si assume che un'operazione identifica sempre una e una sola coppia di parti d'interesse. Quest'ultima assunzione non significa tuttavia che la stessa coppia di parti non possa essere frutto di operazioni diverse

Si noti poi che ogni istanza di questa associazione rispetta i vincoli di precedenza tecnologica, e pertanto si assume che le parti coinvolte in un'istanza dell'associazione rispettano tali vincoli. Nella fattispecie notiamo che possiamo risalire a quale livello di montaggio risalgono le due parti coinvolte visitando l'istanza dell'entità Operazione legata all'istanza corrente di Azione (si veda attributo **livello** in Operazione).

L'attributo *Ripetizioni* indica quante volte l'operazione viene ripetuta su **singole parti identicamente uguali**, ma usate più volte (ad esempio pulire tutti e 4 i pannelli di una lavatrice, e altro).

Associazione Supporto

Cardinalità:

- (0, 1) da parte di operazione in quanto alcune parti potrebbero unirsi tra loro ad incastro e quindi senza l'utilizzo di giunzioni oppure con una *saldatura* (vedi Montaggio)
- (0, N) da parte di Giunzione poiché lo stesso elemento può non essere usato o essere usato da più operazioni

Poiché un'operazione interessa sempre due parti (vedi Operazione), **si assume che solo un elemento di giunzione possa essere coinvolto in un'operazione**.

Si assume che, qualora l'operazione preveda *ripetizioni* (attributo dell'associazione Azione) si utilizzi sempre lo stesso elemento di giunzione (se previsto).

Associazione Usa

Cardinalità:

- (0, N) da parte di Utensile poiché potrebbero esserci utensili non utilizzati in alcuna operazione
- (1, N) da parte di operazione poiché utilizza sempre almeno un utensile, ma potrebbe usarne più d'uno

Associazione SequenzaRelativa

- (1, 1) dal lato "DatiSequenza", un set di dati può riferirsi ad una sola sequenza.
- (1, N) dal lato "Sequenza", una sequenza ha almeno un set di dati che la descrivono ma può verificarsi che abbia più di set di dati. Questo si può verificare quando una sequenza può funzionare sia per il montaggio che per lo smontaggio di un prodotto e dunque possedere i dati relativi al montaggio e allo smontaggio.

Associazione RelativaA

Cardinalità:

- (1, 1) da parte di Sequenza, che si riferisce ad una e una sola Varianza
- (0, N) da parte di Varianza, che può essere montata / smontata utilizzando più sequenze

Associazione OperazioneRelativa

Cardinalità:

- (0, N) da parte di "Operazione". Si noti che la partecipazione opzionale deriva dal fatto che un'operazione può non essere legata a nessuna sequenza. Questa situazione si può creare quando l'azienda mette in listino un nuovo prodotto e dunque ci può essere un arco di tempo in cui le operazioni relative al nuovo prodotto non sono legate a nessuna sequenza fin quanto non verrà avviata la prima produzione.
- (1, 1) da parte di "DatiSequenza".

Associazione OperazioneCampione, Associazione Valutazione

Sono costrutti pensati per studiare l'efficienza della linea di produzione, e il loro funzionamento è spiegato nel paragrafo dedicato all'Analytics 2.

Cardinalità *OperazioneCampione*:

- (1, N) da parte di ClasseCampione, poiché essa è costituita da almeno un'operazione
- (1, 1) da parte di Operazione, poiché tutte le operazioni sono indicate come appartenenti ad una classe (questo non implica che sia proprio l'operazione test che viene fatta svolgere all'operatore per valutarne i tempi, ma implica che sia *simile*)

Cardinalità *Valutazione*

- (0, N) da parte di Operatore, poiché si vuole dare la possibilità di inserire Operatori senza essere obbligati all'inserimento imminente dei suoi test
- (0, N) da parte di ClasseCampione, poiché ogni classe è provata da nessuno o qualche dipendente (in realtà da tutti prima o poi).

Associazione Scelta

Cardinalità:

- (1, 1) da parte di Linea perché si decide a priori uno e un solo tipo di prodotto da smontare o da produrre
- (0, N) da parte di Sequenza perché essa può essere scelta da una o più linee, o anche da nessuno.

Associazione Appartenenza

Cardinalità:

- (1, 1) da parte di "Stazione".
- (1, N) da parte di "Linea".

Associazione Esecuzione

Cardinalità:

- (1, N) da parte di Stazione, poiché assegnata a un insieme di operazioni più o meno vasto
- (0, N) da parte di Operazione, che può essere assegnata a nessuna o a più di una stazione (si pensi a stazioni di linee diverse che producono la stessa cosa)

Associazione LavoraIN

Cardinalità:

- (0, 1) da Operatore, perché può essere o meno assegnato ad una sola stazione
- (1, N) da parte di Stazione poiché più di un lavoratore può essere assegnato a quella stazione

Si noti che l'associazione è legata esclusivamente ad **operatore**, e non a dipendenti (i tecnici non lavorano nelle stazioni).

Associazione PerditaProduzione

Cardinalità:

- (0, N) da parte di Stazione poiché una stazione può perdere o meno più unità
- (1, 1) da parte di UnitàPerseProduzione poiché un unità è persa in una e una sola stazione.

Associazione PersaIN

Cardinalità:

- (0, N) dal lato LottoProduzione. Si noti che la partecipazione opzionale deriva dal fatto che un certo lotto può non aver "perso" alcuna unità durante la fase di smaltimento.
- (1, 1) dal lato "UnitàPerseProduzione".

Associazione UltimaOperazione

Cardinalità:

- (1, 1) da parte di "UnitàPerseProduzione", che sono ferme ad una certa operazione.
- (0, N) da parte di Operazione, poiché la stessa può essere causa d'interruzione di montaggio di nessuna o di più unità.

Associazione ProdottoDA

Cardinalità:

- (1, 1) da parte di Lotto, poiché è prodotto in una e una sola linea di produzione
- (0, N) da parte di Linea, poiché essa può produrre nessuno o, naturalmente, più di un lotto

Associazione Stoccaggio

Cardinalità:

- (1, 1) da parte di Lotto poiché un lotto è stoccato in uno e un solo magazzino (**non possono mai essere spostati una volta stoccati**).
- (0, N) perché all'interno di un AreaMagazzino potrebbero non esserci lotti (magari appena aperto), o potrebbero essere presenti più d'uno.

L'associazione è legata direttamente a Lotto poiché ogni tipo di questi è comunque stoccato in magazzino. La sostanziale differenza è che:

- A. Un LottoProduzione *arriva* in magazzino
- B. Un LottoSmaltimento *parte dal* magazzino

Associazione Suddivisione

Cardinalità:

- (1, 1) da parte di AreaMagazzino, appartenente ad uno e un solo magazzino
- (1, N) da parte di Magazzino, suddiviso in una o più aree

Associazione Predisposizione

Cardinalità:

- (0, 1) da parte di AreaMagazzino, poiché *alcune* possono essere predisposte a una specifica Categoria di Prodotto
- (0, N) da parte di CategoriaProdotto, poiché alcune categorie possono essere predisposizione di Aree.

Associazione ProvenienzaLotto

Cardinalità:

- (0, N) da parte di LottoProduzione poiché egli contiene più unità.
- (1, 1) da parte di UnitàDisponibili poiché appartiene ad uno ed un solo lotto.

Associazione Tipo

Associazione che determina, in termini di codice, prodotto e la variante a cui l'unità si riferisce.

Cardinalità:

- (1, 1) da parte di Unità in quanto questa può essere uno e un solo prodotto di una certa variante del catalogo
- (0, N) da parte di Varianza poiché possono esserci, naturalmente, più unità dello stesso prodotto e variante, ma potrebbero anche non essercene nessuna

Area Vendita

Presentazione delle entità

NOME	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Utente	L'utente che decide di registrarsi al sito per, generalmente, acquistare prodotti	CodFiscale Nome Cognome Citta Provincia Indirizzo NumeroTelefono	CodFiscale
Documento	Il database memorizza l'insieme dei documenti degli utenti che si iscrivono al sito web	Tipologia Numero Scadenza Ente	Numero
Account	L'account è l'"identità virtuale" che un utente utilizza per effettuare operazioni all'interno del sito	NomeUtente Password DomandaSicurezza Risposta IndirizzoConsegna Credito E-Mail	NomeUtente OR E-Mail
OrdineVendita	La richiesta di acquisto di alcuni prodotti effettuata da un account	CodOrdine Stato DataOrdine TotDaPagare	CodOrdine
Garanzia	Formule di estensione della garanzia (è inclusa anche quella base). L'estensione può riguardare tempi di scadenza o classi di guasti coperte.	CodGaranzia Descrizione Durata	CodGaranzia
Spedizione	Virtualizzazione del corriere che recupera le unità da spedire e le porta all'indirizzo di consegna dell'account.	Codice DataPrevista HubAttuale Stato	Codice
Motivazione	Le motivazioni e i difetti che un utente può utilizzare per effettuare una richiesta di reso	CodMotivazione Nome Descrizione	CodMotivazione
Reso	Richiesta di Reso sulla singola unità acquistata	Codice DataRichiesta Approvato DataApprovazione	Codice

UnitaVendute	Figlia dell'entità padre <i>Unità</i> , descrive le unità che sono state vendute	(Aggiuntivi) ClienteVendita	
UnitaRese	Figlia dell'entità padre <i>Unità</i> , descrive le unità che sono tornate in magazzino a seguito di richieste di reso accettate	(Aggiuntivi)	

Presentazione delle associazioni

NOME	DESCRIZIONE	COMPONENTI	ATTRIBUTI
Riconoscimento	Appartenenza di un documento alla persona che l'ha inserito.	Utente Documento	
Identita	Correlazione tra un utente e il suo, generalmente uno, account.	Utente Account	
Giudizio	Valutazione di un utente riguardo prodotti da lui acquistati	Account Prodotto	Valutazione Commento
Acquisto	Ordine associato a un account	OrdineVendita Account	
Carrello	Carrello dell'ordine richiesto	OrdineVendita Varianza Garanzia	Quantita Stato
Copertura	Relazione che lega le formule disponibili ai prodotti dell'azienda	Prodotto Garanzia	Costo
ClasseCoperta	Determina la classe di guasti per la quale la formula di garanzia è indicata	Garanzia ClasseGuasto	
Evasione	Associazione tra una spedizione e l'ordine preso in carico da questa	OrdineVendita Spedizione	

Consegna	Associazione che descrive l'oggetto di una spedizione e il destinatario	Account Spedizione	
RichiestaReso	Richiesta di reso da parte di un account con una motivazione presumibilmente valida	Motivazione Account Reso	Commento
Coinvolgimento	Unità oggetto della richiesta di reso	UnitaVendute Reso	
Restituzione	Unità che vengono restituite a seguito di un reso approvato	UnitaRese Reso	
ContenutaIN	Associazione che descrive l'ubicazione, nei magazzini dell'azienda, delle unità oggetto di reso	UnitaRese Magazzino	
RiferimentoOrdine	Associazione che determina l'ordine alla quale l'unità venduta è associata	UnitaVenduta OrdineVendita	
ProvenienzaLotto	Associazione che descrive la provenienza, in termini di LottoProduzione, di un'UnitaVenduta	UnitaVenduta LottoProduzione	

Descrizione dettagliata di entità e associazioni

Entità Utente

Un utente è una persona che decide di voler iscriversi al sito web per eventualmente effettuare acquisti. Ancor prima di creare un account, un utente è tenuto a inserire parte della sua anagrafica (*Nome, Cognome, CodFiscale...*).

Si assume poi che un utente **possa inserire un solo numero di telefono** (senza escludere la possibilità di modificarlo). L'*Indirizzo* che l'utente fornisce è tendenzialmente l'indirizzo di domicilio e potrebbe coincidere con l'indirizzo di spedizione.

Entità Documento

Un utente è **sempre** tenuto ad inserire un documento di riconoscimento in corso di validità (in caso contrario l'iscrizione non è considerata completata e l'utente viene eliminato).

Un documento di riconoscimento è uno fra i tanti validi (carta d'identità, passaporto, patente), e in ogni caso è caratterizzato da un **codice**, che corrisponde al reale codice (univoco) del documento.

Entità Account

Un account è l'identità virtuale dell'utente, attraverso la quale naviga e svolge tutte le eventuali operazioni effettuabili sul sito. Ogni account ha un **credito**, il saldo senza il quale l'ordine non viene mai evaso.

Un account è identificato in due modi: **NomeUtente** ed **E-Mail**. Ciò vuol dire che non esisteranno casi in cui qualcuno voglia utilizzare la stessa mail per creare due account diversi (anche per persone diverse).

L'attributo indirizzoConsegna ha cardinalità (0,1) in quanto se questo attributo ha valore NULL indica che l'indirizzo di consegna è lo stesso inserito in "utente".

Qualora i dati dell'account vengano dimenticati, si invia una mail con la domanda di sicurezza fornita in fase di registrazione, alla quale l'utente deve, naturalmente, rispondere correttamente.

Entità OrdineVendita

Costituisce gli ordini effettuati nel corso del tempo da tutti i clienti dell'azienda. Oltre al codice identificativo, risulta essere presente anche la data dell'ordine (attributo *DataOrdine*).

L'attributo **stato** può assumere ben 4 diversi valori, ognuno a significato diverso:

- **Pendente:** un ordine è in questo stato se le unità richieste non sono state trovate in magazzino. (Vedi *Carrello*)
- **In Processazione:** un ordine è in questo stato se le unità sono tutte disponibili, e rappresenta la preparazione concreta (recupero delle unità nei magazzini, imballaggio, ...) dell'ordine.
- **Spedito:** Un ordine è in questo stato se è stata commissionata una spedizione che si prenda carica dello stesso
- **Evaso:** un ordine è in questo stato è concluso, e il cliente ha ricevuto le unità dei prodotti che ha richiesto.

L'attributo stato ha valore di default 'In Processazione' che poi verrà appositamente modificato dai trigger presenti nel database.

L'attributo **TotDaPagare** è un attributo calcolato, che verrà esaminato in seguito.

Entità Garanzia

Le formule di garanzia messe a disposizione dall'azienda coprono gran parte dei guasti, e sono caratterizzate da un codice e da una *durata*. Una garanzia, come si vedrà in seguito, è poi adatta a coprire una *particolare classe di guasti*.

PS: Anche la garanzia base è codificata!

Entità Spedizione

Una spedizione è caratterizzata da un codice, una data di consegna prevista, un hub (in cui la spedizione è arrivata), e alcuni **stati**. Così come ordine, anche gli stati della spedizione registrano situazioni diverse:

- **Spedita:** Una volta che l'ordine a cui fa riferimento la spedizione passa nello stato di *Spedito*, la spedizione raggiunge questo stato.
- **In Transito:** Una spedizione è in questo stato quando raggiunge il primo Hub.
- **In Consegna:** la spedizione è in questo stato quando il corriere raggiunge l'ultimo Hub

- **Consegnata:** Lo stato della spedizione è questo quando è conclusa, e la merce ha raggiunto l'utente

Un Hub è un centro di smistamento, situato nella città della quale l'attributo Hub assume il valore.

Si assume che l'ultimo Hub sia una città situata nella stessa provincia della città del cliente (o province vicine in caso di scarsità di Hub). Si assuma anche che **non esistono due Hub nella stessa provincia**.

L'attributo HubAttuale ha cardinalità (0,1) in quanto nel mentre che il pacco non ha raggiunto un primo hub questo assume valore NULL.

Si ricordi anche che, nei termini gestiti dallo schema, la spedizione non risulta essere qualcosa di "concreto", ma solo una virtualizzazione di una consegna presa in carico da un certo corriere. In effetti, potrebbe capitare di trovare più codici di spedizioni destinate anche a due vicini di casa, con data prevista uguale. In molti di questi casi bisogna ricordare che una spedizione *reale*, nel vero significato del termine, può farsi carico di molti codici spedizione presenti nel DB. Per questo motivo, non sarà strano trovare numero di spedizioni simile al numero degli ordini!

Entità Motivazione

Rappresenta le motivazioni che un utente può utilizzare qualora voglia procedere a una richiesta di reso.

È utile ricordare che una motivazione non è sempre valida, o almeno non lo è in tutti i casi.

Si ricordi anche che tra le motivazioni, una ricopre un ruolo speciale: è *DirittoRecesso*, una motivazione che, se reale, deve essere incondizionatamente accettata.

Entità Reso

Un reso è costituito da un codice, ognuno dei quali rappresenta un disagio per il quale si è richiesto di sostituire una certa unità di prodotto. Un reso può essere o meno approvato e, qualora lo sia, si annota la data di approvazione. Qualora il reso venga accettato, si possono intraprendere diverse politiche:

- Un ordine riguardante lo stesso prodotto reso viene generato automaticamente per quell'utente
- L'utente può rinunciare al prodotto, e guadagnare uno sconto dell'80% sul prossimo prodotto che costi non più del prodotto reso

Entità (Figlia) UnitaVendute

Entità facente parte della generalizzazione di *Unità*.

Si assume che, quando un'unità è venduta, questa non è più presente nel magazzino, e lo spazio prima occupato si rende ora libero.

Entità (Figlia) UnitaRese

Questo tipo di unità costituisce tutte le restituzioni che l'azienda decide di accettare. Le unità di questo tipo saranno soggette a possibile ricondizionamento o, se proprio non c'è niente da fare, saranno smaltite.

Sono presenti nell'entità tutte le unità rese che stanno "transitando nel magazzino", aspettando di *essere abbastanza numerose per avviare una procedura di ricondizionamento*.

Associazione Riconoscimento

Cardinalità:

- (1, 1) da parte di "Documento" poiché esso appartiene sempre a uno e un solo utente
- (1, N) poiché un "Utente" è **tenuto** a inserire un documento, e potrebbe esserne richiesto più di uno (in caso di primo documento di dubbia autenticità, o con foto poco chiare)

Associazione Identità

Cardinalità:

- (1, 1) da parte di "Account" poiché è associato solamente ad un utente (e chiaramente non esistono account che non lo siano)
- (0, 1) da parte di un "Utente". La partecipazione opzionale deriva dal fatto che un utente **non è tenuto a creare un account immediatamente**.

Associazione Giudizio

Cardinalità:

- (0, N) da parte di "Account", che può effettuare nessuno o più giudizi sullo stesso prodotto
- (0, N) da parte di "Prodotto", che possono essere recensiti più di una volta

Se il giudizio esiste, deve avere necessariamente una **valutazione**, ovvero un intero su una scala di gradimento che va da 1 a 5, e può avere opzionalmente un **commento**, ovvero un campo testuale in cui l'utente può parlare di eventuali pregi e/o difetti del prodotto.

Si assume che l'utente non è tenuto ad effettuare giudizi, né deve farli in un tempo limitato dall'acquisto. È tuttavia buona norma pensare che il prodotto debba essere stato acquistato dall'account che effettua il giudizio.

L'attributo commento ha cardinalità (0,1) in quanto l'utente potrebbe decidere di dare un voto al prodotto ma senza aggiungere un commento, che quindi assumerebbe valore NULL.

Associazione Acquisto

Cardinalità:

- (0, N) da parte di "Account", che può effettuare, ovviamente, più di un ordine
- (1, 1) da parte di "OrdineVendita", poiché egli è associato solo ad un account, e non esiste se non è stato richiesto

Associazione Carrello

Cardinalità:

- (1, N) da parte di "Ordine" poiché, per come è costruito, esso riferisce sicuramente un prodotto a cui applica una garanzia (anche quella base), ma è ovvio che è un Ordine possa riferirsi a più prodotti
- (1, N) da parte di "Garanzia" poiché una stessa formula si riferisce a più prodotti e a più ordini. *Esempio:* L'ordine O56 è riferito a uno smartphone e a una cassa Bluetooth, sui quali viene applicata comunque la garanzia G21
- (0, N) da parte di "Varianza". Un prodotto, di una certa variante, potrebbe essere acquistato nessuna o più volte. Si noti che questo è possibile perché **prodotto è una virtualizzazione del catalogo aziendale, e non costituisce in alcun modo un magazzino consto di unità a codici diversi**.

L'attributo **quante** gestisce i casi in cui un ordine si riferisca allo stesso prodotto con una certa garanzia più di una volta (di default l'attributo è comunque posto a 1).

L'attributo *Stato* determina invece la presenza (controllata lato server) delle unità effettivamente richieste nel magazzino.

L'attributo **Categoria** determina la fascia d'usato scelta (Nuovo = Categoria A).

Associazione Copertura

Cardinalità:

- (1, N) da parte di "Prodotto", che può vedersi applicati più formule di garanzia.
- (1, N) da parte di "Garanzia", poiché la stessa formula si applica, in genere, a più prodotti. La partecipazione obbligatoria deriva dal fatto che anche la formula base è una garanzia!

L'attributo **costo** determina il prezzo della formula, che non è universale in quanto può cambiare da prodotto a prodotto.

Associazione ClasseCoperta

Cardinalità:

- (1, 1) da parte di "Garanzia".
- (0, N) da parte di "ClasseGuasto". La partecipazione opzionale deriva dal fatto che attualmente l'azienda non abbia predisposto una garanzia che copre una certa ClasseGuasto.

Associazione Evasione

Cardinalità:

- (0, 1) da parte di "OrdineVendita" poiché può avere al massimo una sola spedizione di riferimento
- (1, 1) da parte di "Spedizione", poiché devi riferirsi ad un unico ordine. Si osservi che, per quanto questa scelta possa risultare assai lontana dalla realtà, quasi surreale, è perfettamente in linea con la scelta dell'azienda di considerare le spedizioni come astrazioni virtuali della consegna dell'ordine

Naturalmente la partecipazione da parte di Ordine è limitata ai soli ordini il cui valore sull'attributo stato sia *Spedito*.

Associazione Consegna

Cardinalità:

- (0, N) da parte di "Account", che può essere soggetto a più Spedizioni
- (1, 1) da parte di "Spedizione", che può riferirsi ad un solo Account (anche qui, come sopra, sembra non essere rappresentata una situazione reale).

Associazione RichiestaReso

Cardinalità:

- (0, N) da parte di "Account", che può formulare nessuna o più di una richiesta
- (1, N) da parte di "Motivazione", poiché una motivazione può apparire più volte
- (1, 1) da parte di "Reso", poiché essa fa fede solo e soltanto a un account e a una motivazione

Con l'attributo **commento** si gestiscono i casi in cui l'utente voglia inserire particolari aggiuntivi (motivazioni aggiuntive, ma magari di meno importanza, disagi particolari). Per questo motivo l'attributo commento ha cardinalità (0,1).

In particolare, se l'utente inserisce come motivazione *Altro*, e non inserisce un commento, il reso viene incondizionatamente rifiutato.

Associazione Coinvolgimento

Cardinalità:

- (1, 1) da parte di "Reso", che può coinvolgere una e una sola unità vendute (da *Specifiche*)
- (0, N) da parte di "UnitaVendute", che possono essere oggetto di reso anche più di una volta (magari la prima richiesta di reso era stata rifiutata, e ne è stata effettuata una seconda).

Attenzione: La relazione si lega a Unità Vendute e non ad Unità Rese poiché quest'ultima non è un archivio!

Associazione Restituzione

Cardinalità:

- (1, 1) da parte di "UnitaRese", poiché un'unità è resa solo grazie a una ed una sola richiesta di reso
- (0, 1) da parte di "Reso", poiché una richiesta riguarda una e una sola unità che è rientrata in magazzino. Tuttavia, per poter essere restituita, la richiesta di reso deve essere approvata e dunque può accadere che un'unità possa essere rifiutata (partecipazione opzionale).

Associazione ContenutaIN

Cardinalità:

- (1, 1) da parte di "UnitaRese", che può essere stoccata in una e una sola area magazzino (**e non può mai spostarsi, almeno a livello di sistema informativo**)
- (0, N) da parte di "AreaMagazzino", che può naturalmente ospitare nessuna o più unità rese.

Si noti che Non Esistono aree nel magazzino adibite esclusivamente alle unità rese.

Associazione RiferimentoOrdine

Cardinalità:

- (1, 1) da parte di "UnitaVendute", che si acquistano in un unico ordine
- (1, N) da parte di "OrdineVendita", che può riferirsi a più unità acquistate

Associazione ProvenienzaLotto

Cardinalità:

- (1, 1) da parte di "UnitaVenduta", poiché essa si riferisce ad un unico lotto di provenienza
- (0, N) da parte di "LottoProduzione", che potrebbe esserle totalmente invenduto

Si faccia caso al fatto che quest'associazione permette *sempre* di analizzare la provenienza delle unità, anche qualora queste siano state rese. È molto utile, specie quando si voglia analizzare tutto un lotto basandoci su un difetto che un cliente riscontra.

Area Assistenza

Qui di seguito è presentata una descrizione dell'area dell'azienda relativa ai guasti e gestione delle richieste di assistenza.

Si noti che le entità e le associazioni sono scritte nello stesso modo nel quale appariranno su SQL, dunque non ci stupisce la scelta dell'utilizzo dell' UpperCamelCase e la completa assenza di accenti.

Presentazione delle entità

Si assume che, in caso i campi "identificatore" e "attributi" siano vuoti, l'entità in questione eredita gli attributi dall'entità genitore. Gli attributi aventi l'asterisco (*) sono da considerarsi in aggiunta a quelli ereditati dall'entità genitore.

NOME	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Guasto	Malfunzionamento di tipo hardware o software.	Codice Nome	Codice
ClasseGuasto	Classificazione dei guasti	Nome Descrizione	Nome
Tecnico (ENTITA' FIGLIA)	I tecnici dell'azienda	Occupato* FascioOraria*	
Assistenza (ENTITA' PADRE)	Procedura con il quale il cliente segnala il guasto e, in base alla tipologia e gravità, risolve il malfunzionamento o chiede l'intervento del servizio di riparazione dell'azienda.	CodAssistenza DataRichiesta	CodAssistenza
AssistenzaVirtuale NOCodErrore (ENTITA' FIGLIA)	Procedura online con il quale l'utente, fornendo un iniziale descrizione del guasto e successivamente rispondendo a delle domande, riceve un 'rimedio' per il guasto segnalato.	Riuscita*	
AssistenzaVirtuale, CodErrore (ENTITA' FIGLIA)	Procedure online con il quale l'utente, attraverso l'inserimento del codice di errore visualizzato dal dispositivo guasto, riceve un 'rimedio'.	CodErrore*	

AssistenzaFisica (ENTITA' FIGLIA)	Procedura online con il quale si richiede l'intervento di un tecnico (in loco o in azienda) per guasti di una certa entità non risolvibili dall'utente.	TecniciRichiesti	
Rimedio	Azione con la quale si provvede alla riparazione dell'unità danneggiata	CodRimedio Descrizione	CodRimedio
Sintomo	Problematiche riscontrate da chi fa richiesta di assistenza fisica	CodSintomo Descrizione	CodSintomo
Conoscenza	Base di conoscenza di fondamentale aiuto ai tecnici, lungo il percorso che porta alla risoluzione di un intervento fisico	A.I.M. DataRisoluzione	A.I.M.
InterventoFisico	Intervento di riparazione, compiuto da un 'tecnico', direttamente a casa del cliente che ha richiesto assistenza. Tale intervento può trasformarsi in un intervento in sede dipendentemente dalla gravità del guasto.	Ticket Data Domicilio Stato OreLavoro QuantiAssegnati	Ticket
Preventivo	Documentazione rilasciata prima dell'intervento fisico che riassume le spese da affrontare per aggiustare l'unità malfunzionante.	Codice DataRilascio	Codice
Fattura	Documento fiscale rilasciato al termine dell'intervento fisico.	Codice DataRilascio TotaleNetto CostoOra ModalitaPagamento InGaranzia	Codice
OrdineParti	Procedura interna all'azienda con il quale si gestiscono le parti da sostituire negli interventi fisici a domicilio.	Codice DataRichiesta DataPrevistaConsegna DataConsegna	Codice
Domanda	Quesito, avente risposta binaria, che descrive i "sintomi" del guasto.	CodDomanda Testo	CodDomanda

CentroAssistenza	Luogo in cui vengono compiuti gli interventi di manutenzione/ riparazione relativi alle Assistenze Fisiche.	CodCentro Indirizzo Citta Provincia	CodCentro
------------------	---	--	-----------

Presentazione delle associazioni

NOME	DESCRIZIONE	COMPONENTI	ATTRIBUTI
AutoDiagnosi	Associazione che rappresenta la procedura online con il quale l'utente richiede 'rimedio' per il guasto (non identificato da codice di errore) della sua unità.	AssistenzaVirtuale NOCodErrore Rimedio Domanda	Risposta
Procedura Immediata	Associazione che rappresenta la procedura online con il quale l'utente richiede 'rimedio' per il guasto, per mezzo di un codice di errore, della sua unità.	AssistenzaVirtuale CodErrore Rimedio	
Classe	Associa un guasto alla classificazione per esso pensata	ClasseGuasto Guasto	
IdentificazioneGuasto	Guasto relativo al codice di errore indicato dall'utente che richiede assistenza virtuale	AssistenzaVirtuale CodErrore Guasto	
SintomoIniziale	Rappresenta il sintomo iniziale per il quale l'utente sceglie di avviare una procedura di assistenza virtuale senza codice d'errore	AssistenzaVirtuale NOErrore Sintomo	
SintomiAccusati	Elenco dei sintomi accusati da un utente che fa richiesta di assistenza fisica	Sintomo AssistenzaFisica	
SintomiMemorizzati	Sintomi memorizzati a un intervento salvato nella base di conoscenza	Conoscenza Sintomo	

RimediUtilizzati	Rimedi utilizzati per risolvere uno o più guasti indicati nella base di conoscenza	Conoscenza Rimedio	
GuastiDiagnosticati	Guasto diagnosticato nell'intervento memorizzato nella conoscenza	Conoscenza Guasto	
Relativo	Rapporto tra i guasti possibili e i prodotti su cui si verificano.	Guasto Prodotto	
RiferitoA	Associazione che lega l'Assistenza ad una certa unità venduta.	Assistenza UnitàVendute	
Diagnosi	Associazione che descrive l'operazione preliminare svolta prima che l'operatore intervenga sul guasto; tale azione produce un preventivo.	InterventoFisico Preventivo	
RichiestaParti	Associazione che lega ogni OrdineParte al relativo intervento fisico che ne necessita le parti nuove.	OrdineParte InterventoFisico	
ListaGuasti	Associazione che lega i guasti riscontrati da un operatore e il preventivo da lui stilato.	Guasto Preventivo	Prezzo
Stilato	Associazione che determina, per ogni preventivo, l'operatore che lo ha stilato.	Preventivo Tecnico	
Incarico	Relazione che associa ad ogni richiesta di intervento fisico un operatore dell'azienda.	InterventoFisico Tecnico	
Evasa	Associazione che lega ogni fattura alla relativa prestazione compiuta nell'intervento fisico.	Fattura InterventoFisico	
Attivazione	Associazione che lega una prenotazione on-line al relativo intervento fisico.	AssistenzaFisica InterventoFisico	

Ricambio	Associazione che descrive le parti ordinate dal tecnico per poter adempiere all'assistenza fisica a domicilio.	OrdineParti Parte	
Afferenza	Associazione che lega ad ogni Tecnico il CentroAssistenza a cui afferisce.	Tecnico CentroAssistenza	

Note e assunzioni

1. Si assume che nel caso in cui l'assistenza fisica venga fatta a domicilio, il tecnico assegnato emana un solo ordine con il quale richiede tutte le parti necessarie per l'intervento di manutenzione. Se l'assistenza avviene in azienda si assume che il tecnico non evada alcun ordine perché attinge direttamente al magazzino delle parti.
2. L'opzionalità dell'assistenza fisica deriva dal fatto che l'intervento è attivato se e solo se l'utente ha accettato il preventivo.
3. Si assume che l'ordine relativo all'intervento fisico è fatto solo una volta che il cliente ha accettato il preventivo. Da ciò segue che le parti ordinate saranno anche le parti effettivamente sostituite all'atto della riparazione.
4. Si assume che un intervento fisico comprenda anche la fase di diagnosi del guasto e stilatura del preventivo. Tale fase si suppone che non preveda costi per l'utente ma che sia offerta dall'azienda come parte del programma del suo sistema di Assistenza.
5. Si noti che l'OrdineParti può essere richiesto da un solo tecnico visto che è associato ad intervento fisico (tramite una relazione uno a uno) e ad un intervento fisico corrisponde un solo operatore.

Descrizione dettagliata di entità e associazioni

Entità Guasto

Un guasto può essere riscontrato dal cliente con un tangibile malfunzionamento dell'oggetto oppure può essere un errore software non riscontrabile nel normale utilizzo. Inoltre, è possibile che un guasto può essere direttamente segnalato autonomamente dal dispositivo tramite un codice di errore. L'attributo "CodErrore" rappresenta il codice che appare nel display del prodotto quando il sistema riscontra un malfunzionamento.

Entità Assistenza

L'assistenza è una procedura telematica, con il quale l'azienda si mette a disposizione nei confronti dei clienti, atta a risolvere i guasti che i dispositivi possono avere. Ogni richiesta di assistenza da parte di un cliente è salvata dall'azienda con un codice univoco e una data di richiesta (attributi *CodAssistenza* e *DataRichiesta*).

Entità AssistenzaVirtualeCodErrore

I dispositivi smart, più in generale i dispositivi dotati di un display ed una centralina, sono in grado di avvisare l'utilizzatore di possibili malfunzionamenti / guasti (Esempio: condizionatore che entra in modalità "protezione" e mostra il codice errore 45000). In questi casi la procedura di assistenza si limita a suggerire al cliente un Rimedio relativo al Guasto collegato con il codice d'errore emesso dal dispositivo (Esempio: al codice d'errore 45000 è relativo il guasto 'filtro sporco/intasato';

l'assistenza restituisce il Rimedio "sostituire filtro"). Si assume che se un guasto visualizzi un codice di errore allora la procedura di manutenzione sia nota e tale da poter esser svolta dall'utente.

Entità AssistenzaVirtualeNOCodErrore

Non sempre i dispositivi sono in grado di suggerirci il guasto. In tali occasioni l'azienda offre una procedura più specifica e personalizzata per ogni situazione. L'AssistenzaVirtualeNOCodErrore si occupa di compiere un'indagine sulla base dei "sintomi" presentati dall'unità rotta. Tale procedura può portare ad esito positivo, se quindi si trattava di un guasto noto e facilmente risolvibile dall'utente, oppure negativo (attributo *Riuscita*). In quest'ultimo caso viene attivato un AssistenzaFisica.

L'attributo *riuscita* ha cardinalità (0,1) perché inizialmente il suo valore è sconosciuto e quindi assume valore NULL.

Entità AssistenzaFisica

I Guasti di una certa entità o di cui la procedura di AssistenzaVirtuale non è riuscita a produrre un Rimedio vengono presi in carico dall'AssistenzaFisica. Prevedendo la visione dell'unità rotta da parte di un Tecnico dell'azienda, l'AssistenzaFisica produce un preventivo dei costi e una stima dei tecnici richiesti per compiere la riparazione (attributo *TecniciRichiesti*).

Entità Rimedio

Un Rimedio è una semplice serie di operazione, eseguibili direttamente dal cliente, che permettono di risolvere un Guasto. L'attributo *Descrizione* presenta le operazioni da compiere (Esempio: il Rimedio per uno smartphone frozen prevede la pressione di una sequenza di tasti per un certo intervallo di tempo; ogni smartphone ha la propria sequenza, ognuna descritta dall'attributo *Descrizione*).

Entità InterventoFisico

Ogni AssistenzaFisica, il cui Preventivo è stato accettato, si trasforma in un InterventoFisico. In questa fase dell'assistenza l'addetto dell'azienda svolge la riparazione a casa del cliente oppure preleva l'unità per svolgere le operazioni di riparazione in un centro assistenza a cui afferisce. Si noti che l'attributo *Domicilio* è settato di default a '1'. Si assume dunque che ogni InterventoFisico sia, di default, programmato per essere svolto a domicilio e solo successivamente, nel caso si renda necessario svolgere la riparazione in un centro assistenza, posto a '0'.

L'attributo *OreLavoro* riporta la durata dell'intervento. Tale attributo è utile per il calcolo del costo della mano d'opera. L'attributo *Stato* riporta a che punto si trova lo svolgimento delle operazioni (valori: *AttesaAccettazione*, *InCorso*, *Finito*).

Gli attributi *OreLavoro* e *Data* hanno cardinalità (0,1) in quanto prima che l'intervento venga effettuato non si conoscono il numero di ore che il tecnico impiega per riparare il guasto ed inoltre prima che questo avvenga non si conosce la data della riparazione.

Entità Preventivo

L'addetto dell'azienda che prende in carico l'AssistenzaFisica si occupa di rilasciare un Preventivo, ossia un resoconto delle operazioni e delle spese da sostenere per la riparazione dell'unità. L'entità Preventivo riporta queste informazioni.

Entità Fattura

La Fattura è un documento fiscale che riporta i costi delle parti sostituite, mano d'opera, IVA, informazioni sul metodo di pagamento e informazioni relative al fatto che un intervento può essere volto in garanzia o meno (attributo *InGaranzia*). Inoltre, essendo un documento, la fattura è valida come documento di validità della garanzia offerta dalla riparazione.

Entità OrdineParti

Ogni InterventoFisico può prevedere la sostituzione di una o più parti. Quest'ultime sono reperite dall'addetto che si fa carico dell'intervento tramite un ordine. L'attributo DataConsegna ha cardinalità (0,1) in quanto indica la data effettiva della consegna che avrà valore NULL fin quando non verrà effettuata la consegna della spedizione.

Entità Domanda

Una Domanda è una frase che appare al cliente durante la procedura di AssistenzaVirtuale NOCodiceErrore e descrive uno stato, riconducibile ad un Guasto, in cui l'unità rotta sembra trovarsi.

Esempio: Un'aspirapolvere che si accende ma fatica ad aspirare lo sporco. L'utente avvia una procedura online di AssistenzaVirtualeNOCodErrore e come prima cosa gli viene richiesto di rispondere ad una Domanda del tipo "La spina è inserita e/o la corrente di casa è staccata?". Se la risposta è affermativa vuol dire che il cliente ha verificato e il guasto non è riconducibile al motivo suggerito dalla Domanda, dunque il sistema sottopone il cliente un'altra Domanda. Se la risposta è negativa vuol dire che il cliente si è accorto che l'unità non era in possesso dei requisiti richiesti dalla Domanda e che dunque il Guasto è riconducibile a tale motivazione. Se le motivazioni non sono gravi ed esiste Rimedio allora la procedura restituisce quest'ultimo. Il sistema continua ad inviare le Domande se il Rimedio non si è rilevato efficace. Inoltre, è possibile che ogni Domanda abbia avuto risposta negativa: in questi casi viene automaticamente attivata un'AssistenzaFisica.

Associazione RichiestaParti

Cardinalità:

- (0, 1) dal lato "Intervento Fisico" visto che un intervento fisico può prevedere al più un ordine oppure, nel caso di preventivo non accettato, prevederne zero.
- (1, 1) dal lato "Ordine" visto che ogni ordine è relativo ad uno e un solo intervento fisico.

Entità ClasseGuasto

L'entità ClasseGuasto indica la categoria generale a cui i clienti dell'azienda associano una certa tipologia di guasti. Si noti che questa entità descrive i guasti non dal punto di vista dell'azienda ma dal punto di vista del consumatore. L'attributo *Descrizione* presenta una breve descrizione testuale delle caratteristiche comuni della classe.

Entità Tecnico

Un Tecnico dell'azienda è un dipendente specializzato che si occupa della riparazione e manutenzione delle unità vendute dall'azienda. L'attributo *FasciaOraria* indica, appunto, una fascia oraria giornaliera in cui il Tecnico è reperibile per compiere un InterventoFisico. L'attributo *Occupato* assume i valori "0" o "1".

Entità Sintomo

Un Sintomo rappresenta il modo in cui un malfunzionamento o guasto si manifesta all'utilizzatore del dispositivo.

Entità Conoscenza, Associazione SintomiMemorizzati, Associazione RimediUtilizzati, Associazione GuastiDiagnosticati

Il funzionamento di questa entità e di queste associazioni è ampiamente analizzato nel paragrafo dedicato all'analytics 1.

Cardinalità *SintomiMemorizzati*:

- (1, N) da parte di "Conoscenza", che memorizza sempre almeno un sintomo per il guasto;
- (0, N) da parte di "Sintomo".

Cardinalità *RimediUtilizzati*:

- (1, N) da parte di "Conoscenza", in quanto ogni risoluzione di guasto si riferisce ad uno o più rimedi;
- (0, N) da parte di "Rimedio", poiché egli può riferirsi ad uno o più interventi coperti da A.I.M.

Cardinalità *GuastiDiagnosticati*:

- (1, 1) da parte di "Conoscenza", poiché ogni A.I.M. si riferisce ad un solo guasto
- (0, N) da parte di "Guasto", poiché un guasto potrebbe non essere mai memorizzato nella Conoscenza, o potrebbe esserlo più di una volta (si guardi l'analytics 1).

Entità CentroAssistenza

L'entità CentroAssistenza descrive il luogo nel quale l'azienda provvede alla manutenzione (ordinaria o straordinaria) delle unità vendute. Inoltre, vengono svolte le riparazioni che non possono essere effettuate a domicilio. Ad ogni centro assistenza afferiscono un certo numero di tecnici.

Associazione RiferitoA

Cardinalità:

- (0, N) dal lato "UnitàVendute", poiché ad una certa unità possono essere capitati più guasti e dunque più richieste di Assistenza. La partecipazione opzionale è data dal fatto che un'unità può non essere mai stata guasta e dunque non è mai stata richiesta un'assistenza a lei relativa.
- (1, 1) dal lato "Unità" poiché un'Assistenza è univoca ed è relativa ad una sola unità.

Associazione RichiestaAssistenza

Cardinalità:

- (1, N) dal lato "Utente", perché un utente può richiedere una o più assistenze.
- (1, 1) dal lato "Assistenza", poiché ogni Assistenza è relativa alla richiesta di un solo Utente.

Associazione AutoDiagnosi

Cardinalità:

- (1, N) dal lato "AssistenzaVirtualeNOCodErrore".
- (1, N) dal lato "Rimedio".
- (1, N) dal lato "Domanda".

Associazione ProceduraImmediata

Cardinalità:

- (1, N) dal lato “Rimedio” visto che un rimedio può essere consigliato dal sistema a più utenti che fanno richiesta di assistenza online. Ciò deriva dal fatto che la medesima procedura di riparazione veloce può essere efficace per più di un dispositivo e dunque per diversi codici d’errore (Esempio: il reboot è considerato un ‘Rimedio’ per tutti i dispositivi come smartphone, tablet, smartwatches nel caso di blocco del dispositivo).
- (1, 1) dal lato “AssistenzaVirtualeCodErrore” perché la richiesta di assistenza tramite l’inserimento di un codice errore prevede un solo rimedio possibile da applicare.

Associazione Risoluzione

Cardinalità:

- (1, N) dal lato “Rimedio” perché un rimedio può risolvere più di un guasto.
- (1, N) dal lato “Guasto” perché un guasto può essere risolto da più di un rimedio.

Associazione IdentificazioneGuasto

Cardinalità:

- (1, 1) da parte di “AssistenzaVirtualeConErrore”, che si riferisce solo ad un guasto.
- (0, N) da parte di “Guasto”, che può essere riferito da nessuno o più codici d’errore.

Associazione Relativo

Cardinalità:

- (1, N) dal lato “Prodotto” perché ogni prodotto presenta uno o più guasti catalogati.
- (1, N) dal lato “Guasto” perché lo stesso guasto può essere comune a diversi prodotti (esempio: la rottura del display LCD è comune a tutti gli smartphone).

Associazione RichiestaAssistenza

Cardinalità:

- (1, N) dal lato “Account” perché un utente, identificato a mezzo del suo account, può richiedere una o più assistenze.
- (1, 1) dal lato “Assistenza”.

Associazione Diagnosi

Cardinalità:

- (1, 1) dal lato “Intervento Fisico” perché un operatore produce un, e uno solo, preventivo per ogni chiamata a domicilio.
- (1, 1) dal lato “Preventivo” perché un preventivo è relativo ad uno e uno solo intervento fisico.

Associazione Ricambio

Cardinalità:

- (1, N) dal lato “Ordine” visto che un guasto può aver danneggiato più parti e dunque il relativo ordine deve prevederne altrettante.
- (0, N) dal lato “Parte” visto che lo stesso guasto può essere riscontrato da più unità e dunque la stessa parte può essere coinvolta da più ordini. La partecipazione opzionale deriva dal fatto che è possibile che una certa parte non sia stata mai oggetto di guasti (esempio: è piuttosto raro che il cestello di una lavatrice subisca danni a seguito di un guasto) e dunque non sia mai stata ordinata.

Associazione ProblemiDiagnosticati

Cardinalità:

- (1, N) dal lato "Guasto". Questo perché un guasto può essersi presentato su varie unità e dunque la perizia dei vari operatori ha fatto includere quel guasto su diversi preventivi.
- (1, N) dal lato "Preventivo" visto che un la richiesta di assistenza fisica deve prevedere almeno la presenza di un guasto sull'unità del cliente ma è possibile che vengano riscontrati ulteriori guasti della stessa gravità o collaterali. Tutto ciò implica che nel preventivo deve essere presente almeno un guasto ma è possibile che ci siano più voci inerenti a guasti diversi.

Associazione Stilato

Cardinalità:

- (1, 1) dal lato "Preventivo". Ogni preventivo può essere stilato da un unico operatore a seguito della perizia dell'unità guasta.
- (0, N) dal lato "Tecnico". Un operatore compie numerosi interventi al giorno e dunque, nell'arco della sua carriera, può stilare numerosi preventivi. Non è da escludersi il fatto che un Tecnico, se lavora in coppia con un collega più qualificato, può non aver mai stilato un Preventivo (partecipazione opzionale).

Associazione Incarico

Cardinalità:

- (1, N) dal lato "Intervento Fisico". Un Intervento Fisico può essere assegnato ad almeno un Tecnico oppure, nel caso di interventi a prodotti grandi e ingombranti, 2 o più Tecnici.
- (0, N) dal lato "Tecnico". Un operatore, nell'arco della sua carriera lavorativa in azienda, ha compiuto numerosi interventi fisici e può compierne dei nuovi sino a che avrà un contratto di lavoro con l'azienda oppure sino a quando cambierà mansione.

Associazione Evasa

Cardinalità:

- (1, 1) dal lato "Fattura". Si noti che ogni fattura può al più corrispondere ad un intervento fisico.
- (0, 1) dal lato "Intervento Fisico". La partecipazione opzionale deriva dal fatto che se il preventivo non viene accettato dal cliente allora non viene evasa nessuna fattura (di fatto non viene compiuta nessuna prestazione di riparazione).

Associazione Attivazione

Cardinalità:

- (0, 1) dal lato "Assistenza Fisica". La partecipazione opzionale deriva dal fatto che possono esserci richieste online pendenti, cioè richieste su cui l'azienda e il cliente non si sono ancora accordati sulla data ma comunque sono valide e prese in considerazione dal reparto che si occupa del sistema di Assistenza online.
- (1, 1) dal lato "Intervento Fisico" visto che un intervento può essere attivato e successivamente associato ad una ed una sola richiesta online.

Associazione IdentificazioneGuasto

Cardinalità:

- (1, 1) da parte di "Assistenza VirtualeCodErrore".
- (0, N) da parte di "Guasto". La partecipazione opzionale deriva dal fatto che una procedura di Assistenza VirtualeCodErrore può non portare al riconoscimento di un guasto. Inoltre, una procedura può riconoscere più guasti.

Associazione SintomoIniziale

Cardinalità:

- (1, 1) da parte di "AssistenzaVirtualeNoErrore".
- (1, N) da parte di "Sintomo".

Per una più dettagliata descrizione si rimanda al capitolo riguardante l'analytics 1.

Associazione ListaGuasti

Cardinalità:

- (1, N) da parte di "Preventivo".
- (1, N) da parte di "Guasto".

L'associazione ha l'attributo *Prezzo* che descrive il costo di riparazione associato a ciascun guasto presente in un Preventivo.

Associazione Classe

Cardinalità:

- (0, 1) da parte di "Guasto". Si noti che la partecipazione opzionale deriva dal fatto che un certo Guasto può non essere comune e dunque non avere una classificazione in termini di ClasseGuasto.
- (1, N) da parte di "ClasseGuasto", poiché ad ogni classe possono riferirsi uno o più guasti.

Associazione Afferenza

Cardinalità:

- (1, 1) da parte di "Tecnico".
- (1, N) da parte di "CentroAssistenza".

Area Smontaggio

All'interno di quest'area risiedono tutte le entità e le associazioni che riguardano le procedure di smontaggio, recupero eco-sostenibile dei materiali e ricondizionamento delle unità oggetto di reso o classificate "End-Of-Life".

Si noti che le entità e le associazioni sono scritte nello stesso modo nel quale appariranno su SQL, dunque non ci stupisce la scelta dell'utilizzo dell'UpperCamelCase e la completa assenza di accenti.

Presentazione delle entità

Si assume che, in caso i campi "identificatore" e "attributi" siano vuoti, l'entità in questione eredita gli attributi dall'entità genitore.

NOME	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Test	Procedura eseguita sull'unità oggetto di reso e atta a verificare lo stato del prodotto.	Codice Nome Livello TestPadre Importanza	Codice
Unità End-Of-Life (ENTITA' FIGLIA)	Prodotto, guasto o a fine vita, ritirato dall'azienda e inserito nelle procedure di smaltimento.	GradoUsura	
Unità Ricondizionate (ENTITA' FIGLIA)	Prodotto che ha superato i controlli previsti dalla procedura di refurbishment o, in caso negativo, cui è stato sottoposto a riparazione "rigenerativa".	Grado Ricondizionato	
LineaSmaltimento (ENTITA' FIGLIA)	Insieme di stazioni il cui operato produce come risultato finale lo smontaggio, fino alla profondità scelta, di tutte le unità di un LottodiSmontaggio.		
StazioneSmaltimento (ENTITA' FIGLIA)	Una stazione è l'unità elementare di ogni linea di smaltimento. In una stazione vengono compiute un certo numero di operazioni il cui risultato è il disassemblamento parziale di una unità End-Of-Life.	Livello ParteTarget	

LottoSmaltimento (ENTITA' FIGLIA)	Insiemi di unità End-Of-Life dello stesso prodotto, stoccate in magazzino e destinate alla linea di smontaggio.	GradoUsura	
UnitaPerse Smaltimento	Archivio delle unità che sono state perse da una stazione durante la produzione di un lotto	Quante	Lotto UltimaOperazione (identificatori esterni)
Operazione Smontaggio (ENTITA' FIGLIA)	Una riassuntiva descrizione delle operazioni da fare in atto di assemblaggio		

Presentazione delle associazioni

NOME	DESCRIZIONE	COMPONENTI	ATTRIBUTI
ControlloGenerale	Associazione che descrive la procedura di controllo dello stato del prodotto in base ai test del Test-Tree.	UnitaRicondizionate Test	Superato
ControlloTest	Associazione che lega ogni test con la parte di cui compie il controllo delle funzionalità.	Test Parte	
Ricondizionamento	Associazione che riporta quali parti sono state sostituite/ricondizionate, a seguito del fallimento di una serie di test, di un'unità resa.	Parte UnitaRicondizionate	Quantità
RecuperoMateriale	Associazione che tiene traccia dei materiali, e della loro quantità, recuperati da ogni unità End-Of-Life.	UnitaEnd-Of-Life Materiale StazioneSmaltimento	Quantità
RecuperoParte	Associazione che tiene traccia di tutte le parti recuperate da un'unità End-Of-Life.	UnitaEnd-Of-Life Parte StazioneSmaltimento	Quantità
Albero	Associazione che lega ogni test alla tipologia di prodotto sul quale è eseguito	Prodotto Test	
CostituisceLotto	Rappresentazione dell'appartenenza di un'unità ad un certo lotto.	UnitaEndOfLife LottoSmaltimento	

UltimaOperazione	Esprime l'ultima operazione che è stata effettuata su un'unità prima che questa fosse scartata	UnitàPerseSmaltimento OperazioneSmontaggio	
ScartoSmaltimento	Esprime il legame tra un'unità persa in fase di smaltimento e la stazione nella quale viene persa.	StazioneSmaltimento UnitàPerseSmaltimento	
PersaIN	Esprime il legame tra un'unità persa e il lotto di cui faceva parte.	LottoSmaltimento UnitàPerseSmaltimento	

Note e assunzioni

1. Si noti che la procedura di smontaggio può anche non completarsi; ciò dipende dalle politiche aziendali e dalle condizioni di usura del lotto. In questi casi la linea di smaltimento si conclude in una stazione intermedia avente livello = profondità (attributo associazione *Smaltimento*).
2. Si suppone che ogni stazione della linea di smaltimento, al completamento delle operazioni previste, "produca" una o più parti considerate target. Esempio: si consideri una lavatrice; una delle parti target della lavatrice è il cestello. Se per arrivare a smontare il cestello si impiegano 4 stazioni in un tempo T pari a $4 \cdot \text{Tempo}$ (*Tempo* = attributo di *Linea*), ossia la somma del tempo fisso che ogni stazione della linea ha a disposizione per compiere le operazioni.

Descrizione dettagliata di entità e associazioni

Entità Test

Un record dell'entità test rappresenta un nodo del Test-Tree. Notiamo che l'intero albero dei test può essere ricostruito grazie agli attributi "Livello" e "TestPadre". Assumendo che la radice dell'albero abbia TestPadre = NULL e Livello = 0, notiamo che tutti i figli della radice hanno Livello = 1, i figli dei figli della radice hanno Livello = 2 e così via per ogni nodo. Inoltre, è bene notare che i figli di un nodo, e quindi i sottotest da eseguire nel caso il test fallisce, hanno livello uguale al livello del padre + 1 e l'attributo TestPadre assume il valore del codice del test fallito. Attraverso questa rappresentazione è possibile svolgere tutti i test della procedura di controllo seguendo due regole:

1. Se il test fallisce allora si compiono tutti i test aventi Livello = LivelloTestFallito + 1. Gli stessi devono essere relativi al test precedentemente fallito e dunque devono avere TestPadre = CodTestFallito.
2. Se un test fallisce e nel database non esiste nessun test avente i requisiti di prima allora quella parte del prodotto è da sostituire/ricondizionare.

L'attributo "Importanza" assume valori da 0 a 100 e descrive il peso, in termini di costi di riparazione, della/e parte/i oggetto del test. La procedura di ricondizionamento è avviata se e solo se la somma dell'attributo "Importanza" di ogni istanza di test, il cui esito si è rilevato negativo, è minore dell'attributo "Soglia" dell'entità Prodotto. La somma dell'attributo Importanza di tutti i nodi di un Test-Tree relativo ad un prodotto è pari a 100.

L'attributo Test Padre ha cardinalità (0,1) in quanto, come già descritto, un test potrebbe non avere un test padre.

Entità Unità End-Of-Life

Un prodotto, classificato come Unità End-Of-Life, può essere, ad esempio, un frigorifero. L'azienda, all'atto della consegna di un nuovo frigorifero, dispone il ritiro del vecchio per occuparsi dello smaltimento. La procedura di smontaggio e smaltimento dell'unità prevede il recupero delle parti non usurate che hanno superato i controlli, il recupero dei materiali non pericolosi per l'ambiente (esempio: rivestimenti in plastica, pannelli di alluminio, ecc. ecc.) ed infine lo smaltimento sicuro delle sostanze inquinanti (esempio: gas refrigerante).

Entità UnitàPerseSmaltimento

Le procedure della fase di smaltimento possono andare incontro a degli intoppi. Tali intoppi si traducono in unità che vengono scartate dalla linea (anche se successivamente recuperate). Si classificano queste unità con la nomenclatura "UnitàPerseSmaltimento".

Entità OperazioneSmontaggio

Un "OperazioneSmontaggio" è un'entità duale all'OperazioneMontaggio vista nell'area Produzione.

Entità StazioneSmaltimento

Una stazione recupera, dipendentemente dal tipo di unità, una o più parti riutilizzabili e un certo quantitativo di materiale/i. Esempio: consideriamo un lotto di smartphone, con grado di usura basso; ogni unità del lotto è smontata in toto (profondità di smontaggio massima) visto che è probabile recuperare una buona quantità di parti e materiale. Ogni stazione della linea smaltimento produce un quantitativo di materiale e parti pari al livello a cui si trova. Nel nostro caso la stazione 1 (livello = 1) della linea di smontaggio produce 1 Schermo TouchScreen e 20 gr di plastica, la stazione 2 (livello = 2) produce una batteria 2500 mA, 5 gr di Silicio e 4 gr di alluminio, così per tutte le stazioni. L'attributo livello descrive, con un intero non negativo, la profondità di

disassemblamento che l'unità corrente raggiunge con le operazioni della stazione in cui si trova. Più il livello è maggiore più il prodotto è stato disassemblato. L'attributo TempoStazione indica quando tempo ha/hanno impiegato l'operatore/ gli operatori a compiere tutte le operazioni della stazione e dunque a "produrre" le parti riutilizzabili e il materiale recuperato.

Entità UnitàRicondizionate

Un'unità ricondizionata è un'unità resa sottoposta al processo di controllo generale e, nel caso ci fossero stati dei test non superati, sottoposta ad una o più operazioni di ricondizionamento (Si veda associazione "Operazione Ricondizionamento"). L'attributo grado descrive il livello di usura delle parti riguardanti l'estetica del prodotto. L'attributo assume valori di tipo carattere (Esempio: A, B, C, ...). La qualità del prodotto, dal punto di vista estetico, è migliore tanto quanto l'attributo "Grado" ha valore vicino, in ordine alfabetico, ad A. Il prezzo dell'unità dipende dall'attributo "Grado". Si assume che il prezzo massimo di un prodotto refurbished è quello assunto dalle unità aventi Grado = A. I prezzi delle unità con grado inferiore sono scontati del $K \cdot 5\%$, dove K è il numero di gradi di inferiorità rispetto al grado massimo (Esempio: telefono ricondizionato di grado B ha un prezzo scontato del 5% rispetto allo stesso telefono di grado A; videocamera di grado C ha un prezzo scontato del 10% rispetto alla stessa videocamera di grado A).

L'attributo Grado ha cardinalità (0,1) in quanto si verrà a conoscenza del grado delle unità solo dopo che queste saranno ricondizionate, e quindi l'attributo avrà inizialmente valore NULL.

Entità LottoSmaltimento

Un lotto di smontaggio è un insieme di unità End-Of-Life dello stesso prodotto, stoccate in magazzino e destinate alla linea di smontaggio. L'attributo "GradoUsura" descrive il livello di usura, comune a tutte le unità del lotto, riscontrabile sulle unità che compongono il lotto.

L'attributo DataInizioSmaltimento ha cardinalità (0,1) in quanto dato che lo smaltimento di un lotto ha inizio solo quando viene raggiunto un numero congruo di unità, quest'attributo assume NULL fino a quel momento.

Entità LineaSmaltimento

Una linea di smontaggio rappresenta un insieme di stazioni il cui operato produce come risultato finale lo smontaggio, fino alla profondità scelta, di tutte le unità di un LottodiSmontaggio.

Associazione ControlloGenerale

Cardinalità:

- (1, N) dal lato "Unità Resa", ogni unità resa inserita nel programma di refurbishment deve essere sottoposta a più di un test.
- (1, N) dal lato "Test", i test della procedura di controllo generale sono sottoposti a molte unità rese.

Associazione ControlloTest

Cardinalità:

- (1, N) dal lato "Parte", ogni parte può essere soggetta a diversi test ma comunque, per rispettare un certo livello di qualità dei prodotti ricondizionati, è soggetta ad almeno un test.
- (1, N) dal lato "Test", i test della procedura di controllo generale prevedono il coinvolgimento di almeno una parte che compone il prodotto oppure è possibile che ne coinvolgono diverse contemporaneamente.

Associazione Ricondizionamento

Cardinalità:

- (0, N) dal lato “Unità Resa”, in generale un’unità resa è oggetto di diverse sostituzioni di parti non funzionanti/ usurate che hanno fallito i test ma può capitare, se l’unità non ha fallito nessun test, che l’unità non sia manipolata prima di essere considerata vendibile nuovamente (partecipazione opzionale).
- (0, N) dal lato “Parte”, una parte può essere stata sostituita da diverse unità. Inoltre, nel caso di parti sottoposte ad usura lieve (Esempio: spina della corrente di un microonde), può non essere risultato necessario sostituire quella parte per nessuna unità soggetta a procedura di ricondizionamento.

Associazione Recupero Materiale

Cardinalità:

- (1, N) dal lato “Unità End-Of-Life”, il processo di recupero materiale/i può essere stato applicato ad una o più unità classificate ‘End-Of-Life.
- (0, N) dal lato “Materiale”, un materiale può essere recuperato da una o più unità ‘End-Of-Life. Esempio: il rame è comunemente usato come conduttore elettrico; ciò implica che il rame può essere estratto sia dal sistema di alimentazione di un frigorifero, sia dalle piste stampate di una motherboard di un portatile. La partecipazione opzionale deriva dal fatto che i materiali “tossici” non possono essere recuperati.
- (0, N) dal lato “Stazione Smaltimento”, una stazione recupera uno o più tipologie di materiali, ognuno dei quali in una certa quantità espressa dall’omonimo attributo dell’associazione. Può verificarsi che in una certa stazione non venga recuperato alcun materiale (partecipazione opzionale).

Associazione Recupero Parte

Cardinalità:

- (1, N) dal lato “Unità End-Of-Life”, il processo di recupero parte/i può essere stato applicato ad una o più unità classificate ‘End-Of-Life.
- (0, N) dal lato “Parte”. La partecipazione opzionale deriva dal fatto che alcune parti soggette a forte usura non sono mai recuperate.
- (0, N) dal lato “Stazione Smaltimento”, una stazione recupera uno o più parti. È possibile che una stazione può recuperare nessuna parte: questo dipende dalla profondità di smontaggio e dal livello di usura dell’unità (partecipazione opzionale).

Associazione Albero

Cardinalità:

- (1, N) dal lato “Prodotto”, un prodotto può prevedere più di un Test nella procedura di ricondizionamento.
- (1, 1) dal lato “Test”, un Test può essere compiuto su un solo Prodotto. Seppur ci possono essere dei Test molto simili tra loro (Esempio: controllo resa cromatica dello schermo dei telefoni) si assume che ogni test sia univoco visto che ogni parte può presentare delle specifiche di funzionamento diverse da prodotto a prodotto.

Associazione PersaIN

Cardinalità:

- (0, N) dal lato "LottoSmaltimento". Si noti che la partecipazione opzionale deriva dal fatto che un certo lotto può non aver "perduto" alcuna unità durante la fase di smaltimento.
- (1, 1) dal lato "UnitàPerseSmaltimento".

Associazione ScartoSmaltimento

Cardinalità:

- (0, N) dal lato "StazioneSmaltimento". Si noti che la partecipazione opzionale deriva dal fatto che una certa stazione può non aver "perduto" alcuna unità durante la fase di smaltimento.
- (1, 1) dal lato "UnitàPerseSmaltimento".

Associazione UltimaOperazione

Cardinalità:

- (0, N) dal lato "Operazione". Si noti che la partecipazione opzionale deriva dal fatto che una certa operazione di una linea smaltimento può non aver "perduto" alcuna unità durante il suo smaltimento.
- (1, 1) dal lato "UnitàPerseSmaltimento".

Associazione CostituisceLotto

Cardinalità:

- (0, 1) dal lato "UnitàEndOfLife".
- (0, N) dal lato "LottoSmaltimento".

RISTRUTTURAZIONE DIAGRAMMA

In questo paragrafo saranno presentate le scelte eseguite in fase di ristrutturazione, con riferimento alle generalizzazioni e ad eventuali accorpamenti eseguiti.

Si facciano presente tre cose:

- L'analisi delle ridondanze, un passo propedeutico alla traduzione in schema logico, sarà eseguita nei paragrafi successivi.
- Tutto i paragrafi successivi a questo saranno da considerarsi riferiti solo ed esclusivamente al diagramma ristrutturato (a meno di eccezioni eventualmente specificate).
- Non sono stati utilizzati attributi multivalore.

Traduzione generalizzazioni

Il diagramma, nella sua interezza, presenta otto generalizzazioni che necessitano di essere tradotte. Nonostante alcune (o tutte) le generalizzazioni sono state brevemente presentate sopra, le studieremo nuovamente tutte quante, al fine di giustificare le scelte fatte.

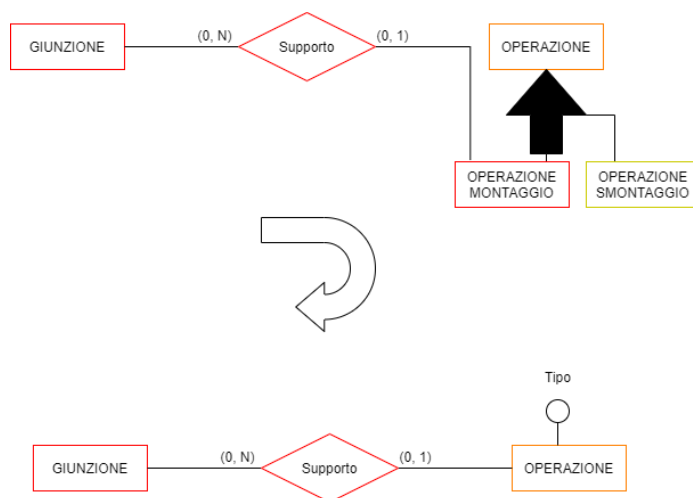
A. OPERAZIONE

Operazione è l'entità padre di una generalizzazione *totale ed esclusiva* che si divide in *Operazione Di Montaggio* e *Operazione Di Smontaggio*. Si assume che un'operazione non può essere contemporaneamente di entrambi i tipi (generalizzazione esclusiva), anche se alcune potrebbero sembrare simili.

Ristrutturazione: Poiché non c'è sufficiente diversificazione dei figli (a meno dell'associazione *Supporto*, che è esclusiva delle operazioni di montaggio) si sceglie **di accorpare le entità figlie nel padre**.

Cambiamenti Significativi:

- 1) Nasce un attributo **Tipo** dentro operazione: questo assume sempre un valore tra 'M' (montaggio) e 'S' (smontaggio). Non vi è possibilità che questo attributo sia NULL in quanto la generalizzazione è totale.
- 2) L'associazione **Supporto** si lega all'entità che resta (quella padre), e non vede cambiamenti di cardinalità (la partecipazione era già opzionale). Chiaramente si assume implicitamente che in una generica istanza di *Supporto* troveremo solo operazioni a codice 'M'.



B. LINEA

Nel diagramma non ristrutturato è evidente come ci sia una linea di smaltimento ed una di produzione. È ovvio che il significato intrinseco delle linee è piuttosto diverso, ma hanno le stesse e identiche caratteristiche: sono associate a un tempo, producono o smaltiscono lotti, perdono unità e così via. La generalizzazione è *totale ed esclusiva*.

Perché non è sovrapposta? Può capitare che una linea sia a volte di smaltimento e a volte di produzione, a seconda di diversi contesti aziendali. Eppure, la generalizzazione è da considerarsi esclusiva in quanto, una linea non è mai contemporaneamente entrambe le cose!

Ristrutturazione: Si sceglie di *accorpare le figlie nel padre*.

Cambiamenti Significativi:

- 1) Si aggiunge un attributo **Tipo** che può assumere due valori: 'P' (Produzione) o 'S' (Smaltimento). Questi non può mai assumere NULL a causa della totalità della generalizzazione, ma può essere modificato mediante operazioni DML.

Le associazioni che legano le linee alle stazioni e ai lotti vengono valutate nelle ristrutturazioni successive, poiché generalizzazioni anch'esse.

C. STAZIONE

Una stazione può occuparsi, come le linee, di produrre o smaltire un prodotto, eseguendo operazioni di montaggio o di smontaggio. La generalizzazione è *totale ed esclusiva*, in quanto una stazione non può essere di due tipi diversi contemporaneamente.

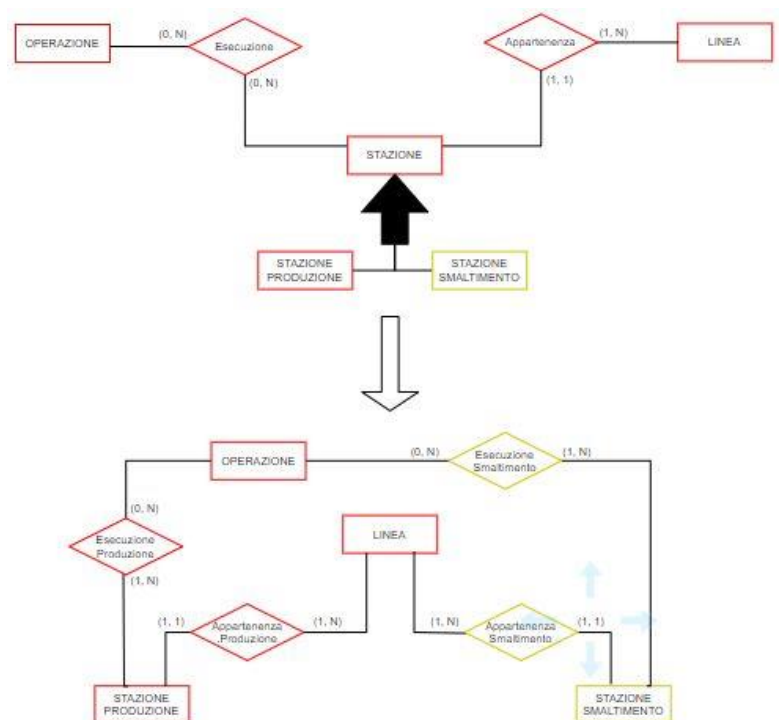
Ristrutturazione: Motivati del fatto che le stazioni siano soggette a regole diverse (in particolare le stazioni di produzione sono osservate con maggiore attenzione a causa della loro maggiore numerosità), si sceglie di **accorpare il padre nelle figlie**.

Cambiamenti significativi:

- 1) L'attributo (per altro identificativo) *CodStazione* cade in entrambe le entità figlie. Chiaramente non c'è perdita di record poiché la generalizzazione è totale.

- 2) L'associazione *esecuzione*, inizialmente connessa solo al padre, si divide in due associazioni (il cui significato è abbastanza ovvio) con l'unica differenza che la cardinalità dell'associazione da produzione è (1, N), da smaltimento è (0, N). Questa scelta è giustificata dal fatto che, quando una linea cambia da produzione a smaltimento, tutte le stazioni ad essa associate diventano del nuovo tipo; a differenza della produzione, però, un prodotto potrebbe non essere smontato fino alla fine e, dunque, possono esistere stazioni di smaltimento che non compiono alcuna operazione!

- 3) Le associazioni di *appartenenza* restano entrambe, ma si collegano ora entrambe alla nuova entità linea ristrutturata in una singola entità.

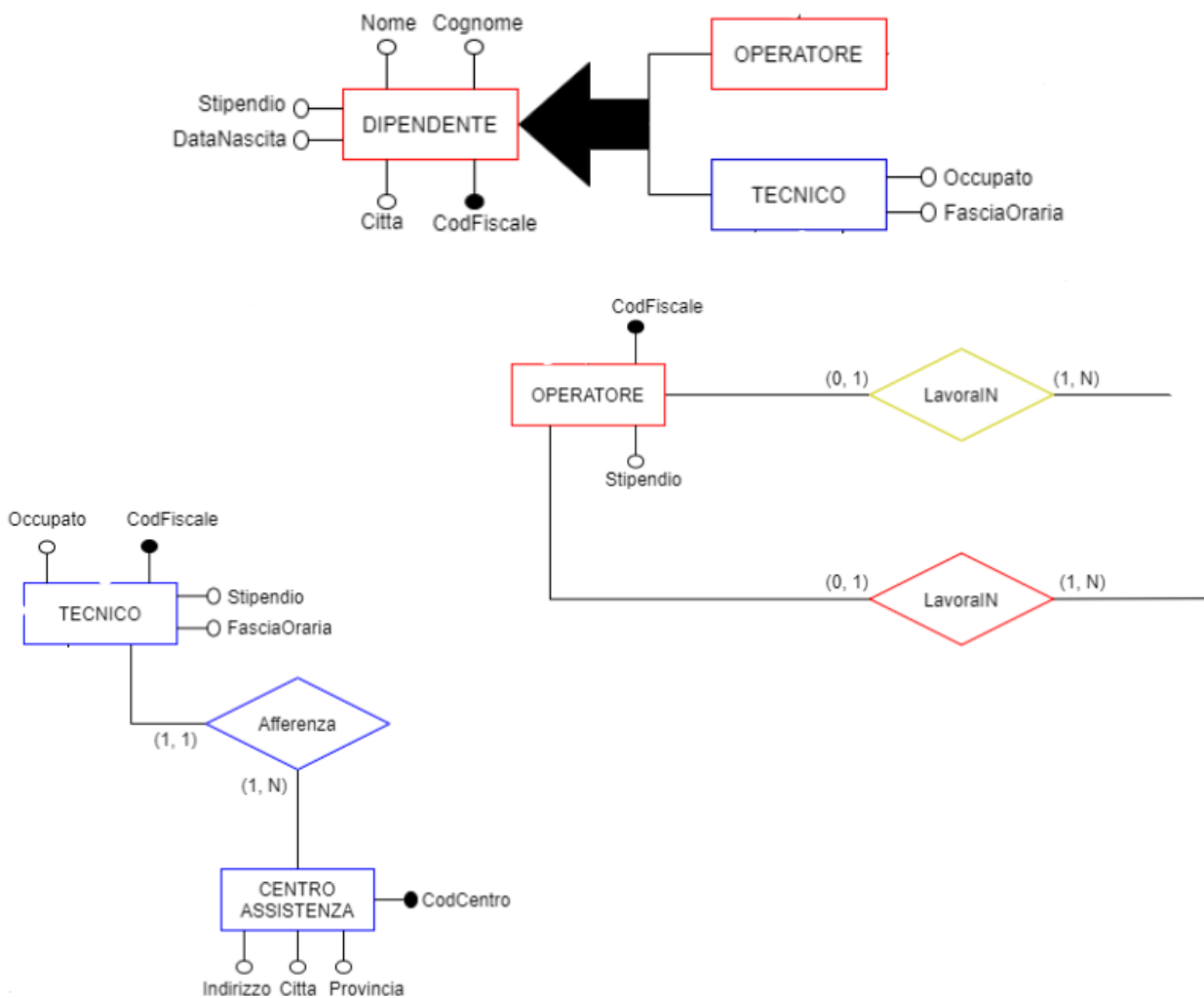


- 4) L'associazione *perde* si divide in due, di ugual cardinalità. Nascono le associazioni **PerditaProduzione** e **ScartoSmaltimento** (entrambe rappresentano il concetto di perdita temporanea dell'unità).

D. DIPENDENTE

I dipendenti di cui l'azienda vuole tenere traccia sono *Operatori* o *Tecnici* (si giustifica così la generalizzazione totale ed esclusiva).

Ristrutturazione: Poiché gli operatori e i tecnici hanno diversificazione si sceglie di accorpare il padre nelle figlie.



Cambiamenti Significativi:

- Gli attributi **CodFiscale** e **Stipendio**, prima presenti nell'entità padre, collassano nelle entità figlie.
- L'associazione *LavoraIn* viene duplicata per l'entità operatore. Una si collega a **StazioneProduzione** mentre l'altra si collega a **StazioneSmaltimento**. Si noti che la cardinalità dell'associazione è rimasta invariata visto che un operatore può lavora in una sola sola stazione (odi produzione o di smaltimento) e mai in due stazioni diverse contemporaneamente.

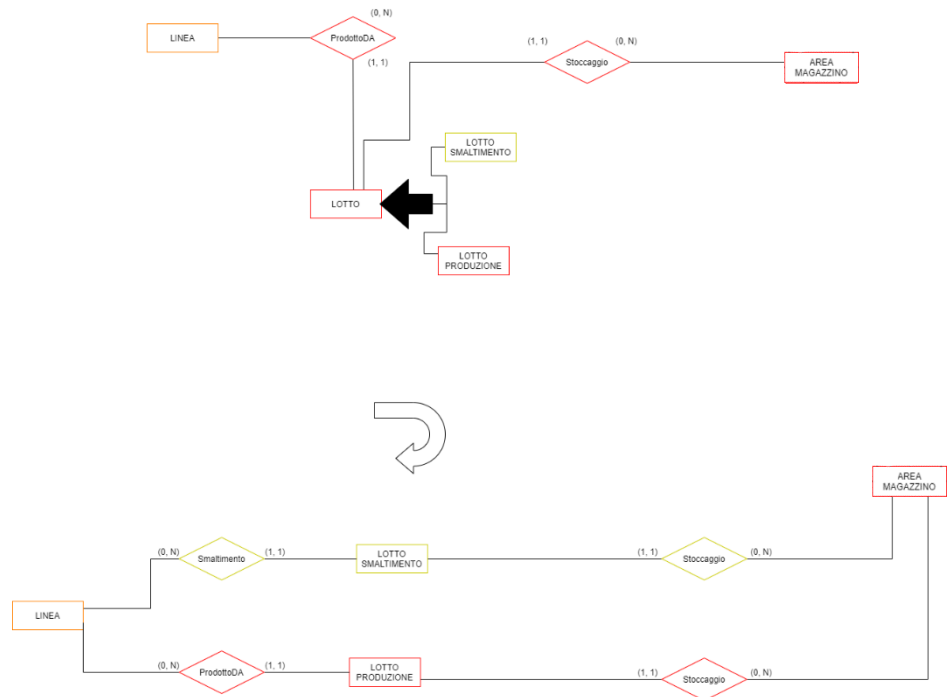
E. LOTTO

Un lotto può essere di produzione o di smaltimento (generalizzazione totale ed esclusiva).

Ristrutturazione: considerata la mole di attributi diversi che identificano i vari tipi di lotti, si sceglie di **accorpare il padre nelle figlie**.

Cambiamenti Significativi:

- Le associazioni che collegano i lotti alle linee restano, con la differenza che ora tutte e due si legheranno all'unica entità rimasta, *Linea*.
- L'associazione che legava il lotto
- divide in due, nascono le associazioni **Stoccaggio**
- Le associazioni che legavano i lotti alle unità *Vendute, End - Of - Life, e Disponibili* si conservano invariate.

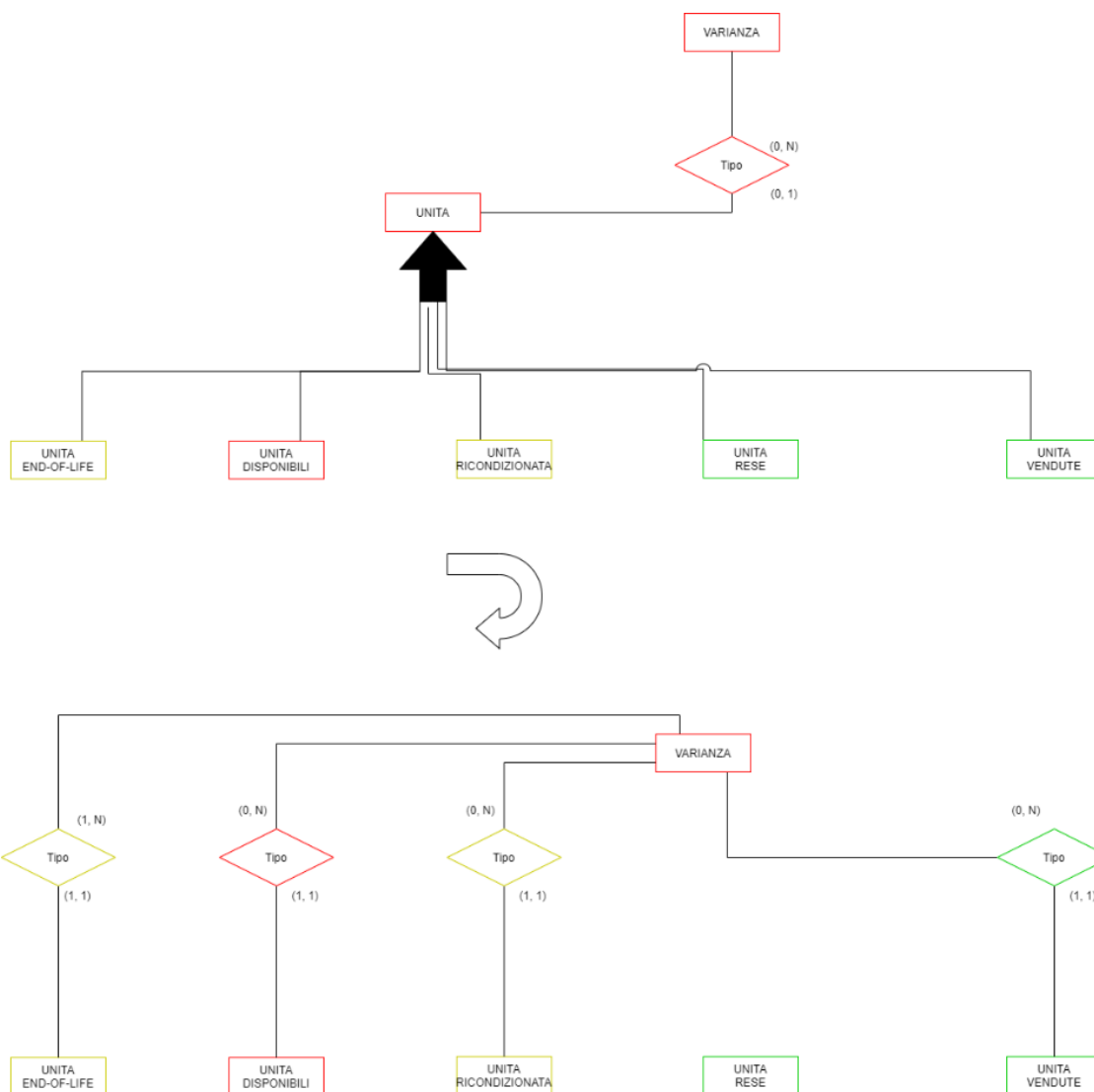


F. UNITA

Generalizzazione *Totale e Sovrapposta*, in quanto la stessa unità si può ripetere, a seconda del suo stato attuale, anche contemporaneamente (è il caso di unità vendute, che costituisce un archivio di tutte le vendite, e che dunque ripete unità che ad esempio sono momentaneamente rese).

Ristrutturazione:

A causa dei diversi concetti che rappresentano, e delle svariate associazioni diverse di cui fanno parte, si sceglie di **accorpare il padre nelle figlie**.



Cambiamenti Significativi:

- 1) L'associazione *Tipo*, legata all'entità padre con partecipazione opzionale, si ripete 4 volte tra i vari figli, a eccezione dell'entità **unità Rese**, di cui non si mantiene questa ridondanza (il prodotto e la variante di un'unità resa è reperibile dall'archivio delle vendite). Poiché questo era ciò che stava alla base della partecipazione opzionale, **la partecipazione di tutte e 4 le associazioni diventa obbligatoria da parte delle unità.**
- 2) Tutte le altre associazioni che coinvolgevano solo le figlie restano immutate

G. ASSISTENZA VIRTUALE E ASSISTENZA

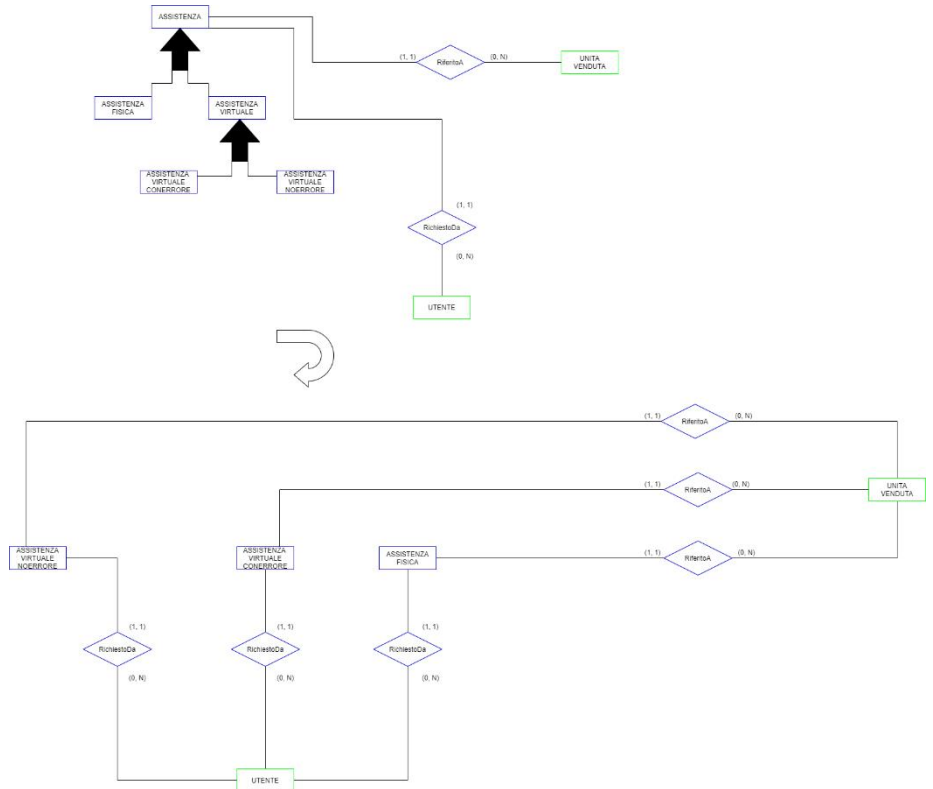
Per quanto riguarda l'assistenza virtuale, questa era inizialmente suddivisa in assistenza con e senza codice d'errore, mediante una generalizzazione *totale ed esclusiva*.

L'assistenza in generale era suddivisa invece in virtuale e fisica, mediante una generalizzazione *totale e sovrapposta*, in quanto lo stesso codice di assistenza virtuale poteva presentarsi in assistenza fisica (questo accade quando alla stessa richiesta di assistenza è suggerito l'intervento di un tecnico).

Ristrutturazione: Si sceglie, dal basso verso l'alto, di tradurre entrambe le generalizzazioni accorpendo il padre nelle figlie. Ciò che resta sono dunque tre entità:
AssistenzaVirtualeNoErrore,
AssistenzaVirtualeConErrore,
AssistenzaFisica

Cambiamenti significativi:

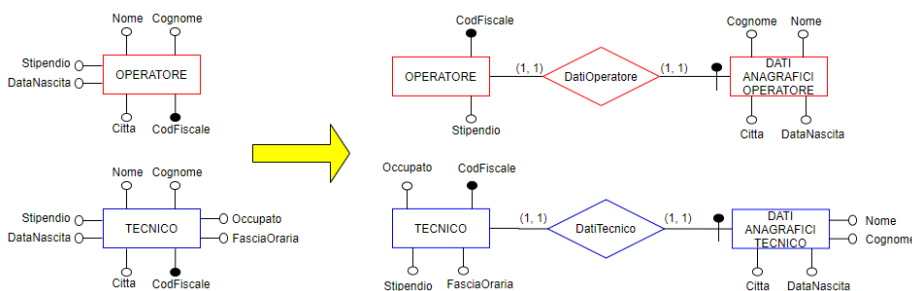
- 1) che collegava l'entità *Assistenza* ad *UnitaVenduta*, e *RichiestoDA*, che collegava l'entità *Assistenza* all'utente che ne faceva richiesta, devono entrambe triplicarsi, mantenendo la stessa cardinalità.



Partizionamento di Entità

Si sceglie di partizionare le due entità *Operatore* e *Tecnico* introducendo due nuove entità che conservino la loro anagrafica (in particolare *DatiAnagraficiOperatore* e *DatiAnagraficiTecnico*).

Perché questo? Potrebbe capitare, e in realtà succede abbastanza spesso, che non si è interessati all'anagrafica dell'operatore (o del tecnico), ma solo a informazioni di tipo "lavorativo" (se un tecnico è occupato, in quale stazione lavora un operatore, quanto guadagnano...).



Viceversa, per operazioni statistiche che riguardino l'anagrafica di operatori e tecnici (dove vivono, quanti anni hanno, ecc.) potrebbe rendersi particolarmente utile cercare in una tabella che

contiene tutte e sole queste informazioni.

Da qui l'idea di operare questa **decomposizione Verticale**, legando *DatiAnagrafici* al lavoratore mediante chiave esterna (codice fiscale).

Elenco e analisi delle ridondanze

Con lo scopo di ridurre gli accessi per calcolare i dati di interesse per le operazioni implementate, si è deciso di inserire e mantenere delle ridondanze all'interno della nostra base di dati.

Nello specifico, consultando le tavole degli accessi è risultato conveniente mantenere quattro ridondanze.

Le motivazioni delle scelte sono illustrate successivamente nella documentazione, precisamente nel [paragrafo in cui vengono descritte le operazioni](#) che le sfruttano.

Attributo CapienzaDisponibile dell'entità AreaMagazzino

Quest'attributo ha il compito di tenere traccia dello spazio disponibile all'interno delle varie aree di stoccaggio. Questo dato costituisce una ridondanza in quanto potrebbe essere calcolato a partire dalla capienza di un'area magazzino, sottraendo a questa lo spazio occupato dalle unità disponibili stoccate in essa. Questo calcolo comporta una enorme mole di accessi nel database, è stata quindi valutata la possibilità di mantenere aggiornato questo dato attraverso un trigger che andasse ad aggiornare l'area ancora disponibile quando delle unità vengono vendute oppure immagazzinate.

Tavola degli accessi relativa all'aggiornamento della ridondanza

Concetto	Costrutto	Accessi	Tipo
LottoProduzione	E	1	L
ProdottoDA	R	1	L
Scelta	R	1	L
RelativaA	R	1	L
InfoVarianza	E	1	L
AreaMagazzino	E	1	S

$$g^A = 3$$

$$o^A = 1 + 1 + 1 + 1 + 1 + 2 = 7$$

$$n^A = g^A * o^A = 21$$

Caso in Cui l'aggiornamento derivi da una vendita

Concetto	Costrutto	Accessi	Tipo
Unità Disponibili	E	1	L
Tipo	R	1	L
InfoVarianza	E	1	L
AreaMagazzino	E	1	S

$$g^A = 500$$

$$o^A = 1 + 1 + 1 + 2 = 5$$

$$n^A = g^A * o^A = 2500$$

Attributo TotaleDaPagare dell'entità OrdineVendita

Questo attributo rappresenta l'importo totale che il cliente ha pagato per un ordine.

Considerando la frequenza con cui l'azienda accede a questi dati è risultato conveniente mantenere questa ridondanza implementando un trigger che ne aggiorni continuamente il valore.

Tavola degli accessi relativa all'aggiornamento della ridondanza

Concetto	Costrutto	Accessi	Tipo
InfoVarianza	E	3	L
Ordine Vendita	E	1	S

$$g^A = 2500$$

$$o^A = 3 + 2 = 5$$

$$n^A = g^A * o^A = 12500$$

Attributo TotaleNetto dell'entità Fattura

Questo attributo rappresenta l'importo totale del costo della riparazione per una specifica assistenza fisica. È stato implementato un trigger che ne calcola il valore non appena viene inserita una nuova istanza di fattura nel database.

Tavola degli accessi relativa alla Ridondanza

Concetto	Costrutto	Accessi	Tipo
Fattura	E	1	S
Evasa	R	1	L
InterventoFisico	E	1	L
RichiestaParti	R	1	L
Ricambio	R	2	L
Parte	E	2	L

$$g^A = 50$$

$$o^A = 2 + 1 + 1 + 1 + 2 + 2 = 9$$

$$n^A = g^A * o^A = 450$$

Attributo UnitàEffettive dell'entità LottoProduzione

Questa ridondanza merita una particolare attenzione in quanto per essa è stata presa una decisione particolare, per la quale rimandiamo alla discussione delle tavole degli accessi delle [operazioni 10 e 11](#).

TAVOLA DEI VOLUMI

Concetto	Tipo	Volume	Motivazione
Prodotto	E	20	L'azienda produce 20 prodotti (ipotesi)
Variante	E	60	(ipotesi)
Varianza	E	$20 * 4 = 80$	In media ogni prodotto ha 4 varianti
Parte	E	360	Ogni prodotto è composto in media da 20 parti, ma alcuni prodotti hanno delle parti in comune (ipotesi)
Materiale	E	17	I materiali diversi che compongono le parti dei prodotti sono 17 (ipotesi)
Operazione	E	$80 * 30 = 2400$	La costruzione di una varianza richiede in media 30 operazioni
Utensile	E	100	Gli utensili a disposizione degli operatori sono circa 100 (ipotesi)
Giunzione	E	30	(ipotesi)
Linea	E	$20 * 10 = 200$	L'azienda possiede in media 10 linee per ogni prodotto. Ogni giorno alcune di queste linee (in media 2) vengono dedicate allo smaltimento di prodotti.
Lotto Produzione	E	$(6.000.000 + 20.000.000) / 10 = 2.600.000$	Considerando che in media ogni lotto di prodotti contiene 10 unità possiamo ricavare il numero di lotti fabbricati sommando tutte le unità disponibili e vendute e dividerle per 10
Stazione Produzione	E	$198 * 8 = 1584$	Ogni linea di produzione ha in media 8 stazioni.
Unità Disponibili	E	$300.000 * 20 = 6.000.000$	Per ogni prodotto si hanno in media 300.000 unità disponibili
Magazzino	E	15	L'azienda possiede 15 magazzini (ipotesi)
Classe Campione	E	100	Prendendo in considerazione le operazioni che vengono effettuate per produrre i propri prodotti l'azienda ha suddiviso le operazioni in 100 classi campione (ipotesi)
Info Varianza	E	80	Quante sono le istanze di "Varianza"
Categoria Prodotto	E	6	I prodotti sono divisi in 6 categorie (ipotesi)
Dati Sequenza	E	160	Quante sono le istanze di "Sequenza Relativa"
Sequenza	E	$80 * 2 = 160$	In media ogni "Varianza" ha due sequenze
Area Magazzino	E	$15 * 10 = 150$	Ogni magazzino è suddiviso in media in 10 aree

Operatore	E	2000	L'azienda ha in tutto 2000 operatori (ipotesi)
Dati Anagrafici Operatore	E	2000	Quante sono le istanze di "Operatore"
Unita Perse Produzione	E	$2.600.000 / 4 = 650.000$	In media viene persa un'unità ogni 4 lotti di produzione
Tipo (Disponibili)	R	6.000.000	Quante sono le istanze di "Unità Disponibili"
Composizione	R	$20 * 20 = 400$	Ogni prodotto è composto in media da 20 parti
Struttura	R	$360 * 2,5 = 900$	Ogni parte è composta in media da 2,5 materiali
LavoraIN (Produzione)	R	$1584 * 1,2 = 1900$	In ogni stazione di produzione lavorano in media 1,2 operatori
Supporto	R	$(2400 * 70) / 100 = 1680$	Il 70% delle operazioni richiede una giunzione
Usa	R	$2400 * 1,7 = 4080$	Ogni operazione usa in media 1,7 utensili
Scelta	R	$20 * 3 = 60$	Ogni prodotto ha in media 3 sequenze di operazioni possibili per essere assemblato
Esecuzione Produzione	R	$1584 * 3.75 = 5940$	Ogni prodotto richiede in media 30 operazioni che verranno eseguite in 8 stazioni di produzione. Quindi ogni stazione esegue in media 3.75 operazioni
ProdottoDa	R	2.600.000	Quante sono le istanze di "Lotto Produzione"
Riferita	R	600	Quante sono le istanze di "Operazione"
Varianti Assumibili	R	80	Quante sono le istanze di "Varianza"
Relativa A	R	160	Quante sono le istanze di "Sequenza"
Descrizione Variante	R	80	Quante sono le istanze di "Varianza"
Azione	R	2400	Quante sono le istanze di "Operazione"
Valutazione	R	$2000 * 5$	Ogni operatore è sottoposto ad una media di 5 operazioni sulle quali riceve una valutazione.
Campione Operazione	R	2400	Quante sono le istanze di "Operazione"
Operazione Relativa	R	600	Quante sono le istanze di "Dati Sequenza"
Sequenza Relativa	R	160	Quante sono le istanze di "Sequenza"
Appartenenza Produzione	R	1584	Quante sono le istanze di "Stazione Produzione"
Dati Operatore	R	2000	Quante sono le istanze di "Dati Anagrafici Operatore"

Perdita Produzione	R	650.000	Quante sono le istanze di "Unita Perse Produzione"
Ultima Operazione	R	650.000	Quante sono le istanze di "Unita Perse Produzione"
Persa In (Produzione)	R	650.000	Quante sono le istanze di "Unita Perse Produzione"
Provenienza Lotto	R	20.000.000	Quante sono le istanze di "Unita Vendute"
Predisposizione	R	$150 * 0,9 = 135$	In media 0,9 aree magazzino hanno una predisposizione
Suddivisione	R	150	Quante sono le istanze di "Area Magazzino"
Stoccaggio (Produzione)	R	2.600.000	Quante sono le istanze di "Lotto Produzione"
Tipo Categoria	R	20	Quante sono le istanze di "Prodotto"
Utente	E	16.000.000	(ipotesi)
Documento	E	18.000.000	Quante sono le istanze di "Riconoscimento"
Account	E	15.000.000	(Ipotesi)
Ordine Vendita	E	19.000.050	(ipotesi)
Spedizione	E	19.000.000	Quante sono le istanze di "Evasione"
Garanzia	E	7	L'azienda offre 7 tipo di garanzia (ipotesi)
Reso	E	$7.000 * 20 = 140.000$	Ogni prodotto è stato reso in media 7.000 volte (ipotesi)
Unità Vendute	E	$20 * 1.000.000 = 20.000.000$	Per ogni prodotto sono state vendute in media 1.000.000 di unità
Unità Rese	E	$20 * 13 = 260$	Per ogni prodotto ci sono 13 unità rese in attesa di essere inviate all'area smontaggio
Motivazione	E	8	Le motivazioni di reso di che l'azienda propone di default sono 8 (ipotesi)
Tipo (Vendute)	R	20.000.000	Quante sono le istanze di "Unità Vendute"
Riconoscimento	R	$15.000.000 * 1,2 = 18.000.000$	Ogni utente ha registrato in media 1,2 documenti
Identità	R	15.000.000	Quante sono le istanze di "Account"
Giudizio	R	$20.000.000 / 2,5 = 8.000.000$	In media ogni 10 prodotti venduti 2,5 di questi ricevono un giudizio
Acquisto	R	19.000.050	Quante sono le istanze di "Ordine Vendita"
Carrello	R	$19.000.050 * 3 = 57.000.150$	Per ogni "Ordine Vendita" vi sono in media 3 prodotti acquistati
Evasione	R	19.000.000	Sono state evase 19.000.000 spedizioni (ipotesi)

Consegna	R	19.000.000	Quante sono le istanze di “Spedizione”
Copertura	R	$20 * 5 = 100$	Ogni prodotto può essere coperto in media con una tra 5 possibili garanzie diverse
Richiesta Reso	R	140.000	Quante sono le istanze di “Reso”
Coinvolgimento	R	140.000	Quante sono le istanze di “Reso”
Restituzione	R	140.000	Quante sono le istanze di “Reso”
Classe Coperta	R	7	Quante sono le istanze di “Garanzia”
Riferimento Ordine	R	20.000.000	Quante sono le istanze di “Unita Vendute”
Contenuta IN	R	260	Quante sono le istanze di “Unita Rese”
Provenienza Lotto	R	20.000.000	Quante sono le istanze di “Unita Vendute”
Centro Assistenza	E	250	L'azienda ha 250 centri assistenza
Guasto	E	180	I possibili guasti dei prodotti sono 180 (ipotesi)
Assistenza Fisica	E	$16.000.000 / 35 = 457.142$	In media un utente ogni 35 ha richiesto assistenza fisica.
Assistenza Virtuale Con Errore	E	$16.000.000 / 15 = 1.066.666$	In media un utente ogni 15 ha richiesto Assistenza Virtuale CON Errore
Assistenza Virtuale No Errore	E	$16.000.000 / 30 = 533.333$	In media un utente ogni 30 ha richiesto Assistenza Virtuale NO Errore
Rimedio	E	$180 / 1.5 = 120$	Ogni rimedio risolve in media 1,5 guasti
Preventivo	E	457.000	Quante sono le istanze di “Diagnosi”
Intervento Fisico	E	457.000	Quante sono le istanze di “Attivazione”
Fattura	E	400.000	Quante sono le istanze di “Evasa”
Ordine Parti	E	300.000	Quante sono le istanze di “Richiesta Parti”
Domanda	E	$120 * 7 = 840$	Ogni rimedio prevede in media 7 domande
Sintomo	E	100	L'azienda per i suoi prodotti ha stilato una lista di possibili sintomi inerenti a i guasti che potrebbero verificarsi. (ipotesi)
Conoscenza	E	$533.333 + 1.5 * 457.142 = 1.219.046$	Si assume che in Conoscenza vengano memorizzati tutti i sintomi iniziali delle procedure di “Assistenza Virtuale NO CodErrore” ed i sintomi derivanti dalle “Assistenze Fisiche” dove quest'ultime presentano in media 1.5 sintomi accusati
Classe Guasto	E	20	(ipotesi)

Tecnico	E	300	L'azienda ha 300 tecnici (ipotesi)
Dati Anagrafici Tecnico	E	300	Quante sono le istanze di "Tecnico"
Ricambio	R	$300.000 * 2 = 600.000$	Ogni "Ordine Parti" richiede in media 2 parti
Procedura Immediata	R	1.066.666	Quante sono le istanze di "Assistenza Virtuale CON Errore"
Lista Guasti	R	$457.000 * 1.6 = 731.200$	Ogni preventivo risolve in media 1,7 guasti
Relativo	R	$20 * 12 = 240$	Ogni prodotto ha in media 10 possibili guasti
Riferitoa (Assistenza Fisica)	R	457.142	Quante sono le istanze di "Assistenza Fisica"
RichiestoDa (Assistenza Fisica)	R	457.142	Quante sono le istanze di "Assistenza Fisica"
RiferitoA (Assistenza Virtuale CON Errore)	R	1.066.666	Quante sono le istanze di "Assistenza Virtuale CON Errore"
RichiestoDa (Assistenza Virtuale CON Errore)	R	1.066.666	Quante sono le istanze di "Assistenza Virtuale CON Errore"
RiferitoA (Assistenza Virtuale NO Errore)	R	533.333	Quante sono le istanze di "Assistenza Virtuale NO Errore"
RichiestoDa (Assistenza Virtuale NO Errore)	R	533.333	Quante sono le istanze di "Assistenza Virtuale NO Errore"
Evasa	R	400.000	Alcuni interventi fisici non comportano una fattura in quanto il cliente può decidere di non far riparare il prodotto
Diagnosi	R	457.000	Quante sono le istanze di "Intervento Fisico"
Stilato	R	437.000	Quante sono le istanze di "Preventivo"
Incarico	R	$437.000 * 1,1 = 480.700$	Ogni intervento fisico necessita in media di 1,1 dipendenti
Attivazione	R	457.000	Non tutte le richieste di assistenza fisica sono ancora state accontentate
Richiesta Parti	R	300.000	Alcuni interventi fisici non richiedono pezzi di ricambio
Sintomi Memorizzati	R	1.300.000	(ipotesi)
Sintomi Accusati	R	$457.142 * 1,5 = 685.713$	In media ogni assistenza fisica va a rimediare a 1.5 sintomi riscontrati dai clienti.
Sintomo Iniziale	R	533.333	Quante sono le istanze di "Assistenza Virtuale NO CodErrore"
Rimedi Utilizzati	R	1.219.046	Quante sono le istanze di "Conoscenza"
Identificazione Guasto	R	1.066.666	Quante sono le istanze di "Assistenza Virtuale CON Errore"

Guasto Diagnosticato	R	1.219.046	Quante sono le istanze di “Conoscenza”
Autodiagnosi	R	$120 * 4 * 533.333 = 256.000.000$	Considerando che per ogni rimedio sono previste un massimo di 7 domande ma non necessariamente vengono visualizzate tutte si può stimare che ne vengono poste 4 per ogni effettiva procedura di rimedio e da cui segue il calcolo
Dati Tecnico	R	300	Quante sono le istanze di “Tecnico”
Classe	R	180	Quante sono le istanze di “Guasto”
Afferenza	R	300	Quante sono le istanze di “Tecnico”
Test	E	$25 * 20 = 500$	L'albero di test di ogni prodotto è formato in media da 25 test
Unità Ricondizionate	E	$20 * 20.000 = 400.000$	Per ogni prodotto sono state ricondizionate in media 20.000 di unità
Unità End-Of-Life	E	$20 * 10.000 = 200.000$	Per ogni prodotto sono state smaltite in media 10.000 di unità
Stazione Smaltimento	E	$2 * 8 = 16$	Le linee di smontaggio sono in media 2 al giorno ed ognuna di esse ha in media 8 stazioni
Lotto (Smaltimento)	E	16000	(ipotesi)
Unità Perse Smaltimento	E	3200	In media viene persa un'unità ogni 5 lotti smaltimento
Tipo (End-Of-Life)	R	200.000	Quante sono le istanze di “Unità End-Of-Life”
Tipo Ricondizionata	R	300.000	Quante sono le entità di “Unità Ricondizionata”
Albero	R	500	Ogni test fa parte dell'albero di un singolo prodotto
Controllo Generale	R	$400.000 * 12 = 4.800.000$	Per ogni unità ricondizionata sono previsti al massimo 25 test, ma in media ne vengo fatti 12 perché può capitare che il problema venga trovato subito.
Controllo Test	R	$360 * 1,4 = 504$	Ogni parte può essere sottoposta in media a 1,4 test
Ricondizionamento	R	$400.000 * 2 = 800.000$	Perché ricondizionare le unità ha richiesto in media 2 parti nuove
LavoraIN (Smontaggio)	R	$16 * 1,2 = 19$	In ogni stazione di produzione lavorano in media 1,2 operatori
Costituisce Lotto	R	199.800	Poiché le unità appena smaltite devono ancora essere assegnate ad uno specifico lotto
Recupero Materiale-	R	$200.000 * 3 = 600.00$	In media da ogni unità End-Of-Life si recuperano 3 differenti materiali

Recupero Parte	R	$200.000 * 8 = 1.600.000$	Considerando che ogni prodotto è costituito in media da 20 parti, ma di queste non tutte possono essere recuperate, da ogni unità smaltita si recupereranno in media 8 parti
Smaltimento	R	16.000	Quante sono le entità di "Lotto Smaltimento"
Appartenenza Smaltimento	R	16	Quante sono le istanze di "Stazione Smontaggio"
Esecuzione Smaltimento	R	$20 * 20 = 400$	Smaltire un prodotto richiede in media 20 operazioni
Ultima Operazione	R	3200	Quante sono le istanze di "Unità Perse Smaltimento"
Scarto Smaltimento	R	3200	Quante sono le istanze di "Unità Perse Smaltimento"
Persa In (Smaltimento)	R	3200	Quante sono le istanze di "Unità Perse Smaltimento"
Stoccaggio (Smaltimento)	R	16.000	Quante sono le entità di "Lotto Smaltimento"

ANALISI DELLE OPERAZIONI

Elenco delle operazioni d'interesse

In questa sezione vengono elencate le operazioni più corpose messe a nota dell'azienda, seguendo la regola dell'80 - 20 (l'80% delle operazioni usa sempre lo stesso 20% dei dati).

Si faccia presente che l'insieme di operazioni possibili è decisamente più grande della lista di cui sotto. Si presenta l'implementazione SQL di alcune operazioni, si rimanda alla lettura del file *ImplementazioneSQL* per ulteriore visione.

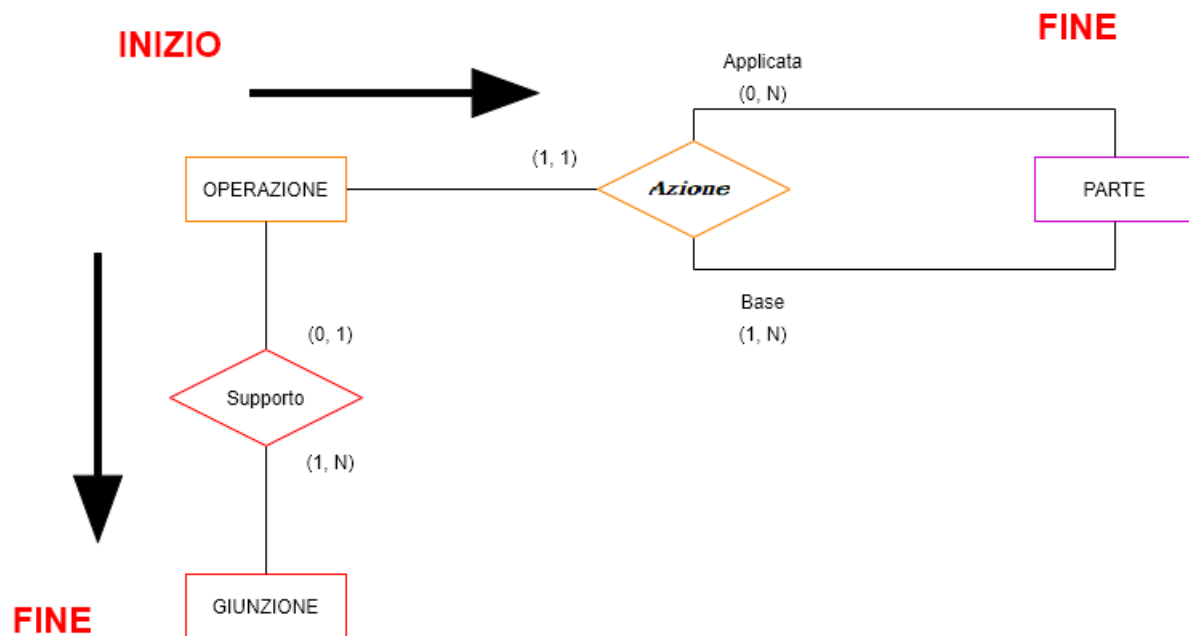
Operazione	Descrizione
Op. 1	Ricavare la descrizione di una operazione, la/le parte/i coinvolta/e, ed elemento di giunzione utilizzato (con relativa descrizione della stessa).
Op. 2	Scontare del 10% i prodotti usciti in commercio più di un anno fa.
Op. 3	Considerato un Unità Resa, si ricavi la richiesta di reso di riferimento, indicandone motivazione e codice fiscale, nome e cognome del cliente che ne ha fatto richiesta.
Op. 4	Considerato un cliente, considerare quanto hanno speso nell'ultimo ordine effettuato con successo.
Op. 5	Inserimento di una nuova assistenza virtuale senza codice d'errore.
Op. 6	Considerato un Tecnico, ricavare gli Interventi Fisici programmati per la giornata.
Op. 7	Attivazione di una procedura di controllo dello stato di un'unità.
Op. 8	Considerato un prodotto, visualizzare la quantità media di materiali (espressa in kg) oggetto di procedura di recupero.
Op. 9	Considerato un magazzino, visualizzarne la capienza disponibile.
Op. 10	Inserisci un'unità persa per una stazione di produzione (conoscendo a priori l'ultima operazione e il lotto).
Op. 11	Restituire quante unità si stanno perdendo al momento per un certo lotto fornito da utente
Op. 12	Considerate le fatture del giorno, ricavare la somma dei totali netti

Operazione	Tipo	Frequenza
Op. 1	I	300 al giorno
Op. 2	B	1 all'anno
Op. 3	I	50 volte al giorno
Op. 4	I	10000 al giorno
Op. 5	I	50 volte al giorno
Op. 6	I	500 volte al giorno
Op. 7	I	50 al giorno
Op. 8	I	1 al mese
Op. 9	I	10 volte al giorno
Op. 10	I	100 volte al giorno
Op. 11	I	10 volte al giorno
Op. 12	B	2 volte al giorno

Operazione 1: Ricavare la descrizione di una operazione, la/le parte/i coinvolta/e, ed elemento di giunzione utilizzato (con relativa descrizione della stessa).

Dati in INPUT: Codice di un'operazione

Porzione di diagramma E- R interessata:



Spiegazione Operazione:

Considerata un'operazione, questa può riferirsi al massimo a due sole parti, e può utilizzare un solo elemento di giunzione. Per compiere quest'operazione abbiamo dunque bisogno di letture in montaggio e in supporto. Non abbiamo invece bisogno di scritture.

C'è bisogno di una lettura in Operazione e in Giunzione poiché non c'è altro modo di ricavare la descrizione dell'operazione e dell'elemento utilizzato. Le letture in parte vengono giustificate dal fatto che si vogliono invece ricavare informazioni sulle parti interessate.

Tavola degli Accessi:

Concetto	Costrutto	Accessi	Tipo
Operazione	E	1	L
Montaggio	R	1	L
Supporto	R	1	L
Giunzione	E	1	L
Parte	E	2	L

Implementazione SQL

```
SELECT O.ID,  
       A.Base as 'Parte base',  
       PB.Nome as 'Nome Parte BASE',  
       IFNULL(A.Applicata, ' - ') as 'Parte applicata',  
       IFNULL(PA.Nome, ' - ') as 'Nome Parte APPLICATA',  
       IFNULL(O.Giunzione, 'Saldatura') as 'Elemento di giunzione',  
       O.Descrizione  
FROM azione A INNER JOIN operazione O ON A.Operazione = O.ID  
              INNER JOIN parte PB ON PB.CodParte = A.Base  
              INNER JOIN parte PA ON PA.CodParte = A.Applicata  
WHERE O.ID = 1;
```

Operazione 2: Scontare del 10% tutti i prodotti usciti in commercio più di un anno fa

Dati in INPUT: Nessuno

Dati in OUTPUT: Nessuno

Porzione di diagramma E - R interessato:

INIZIO

PRODOTTO

FINE

Tavola degli Accessi:

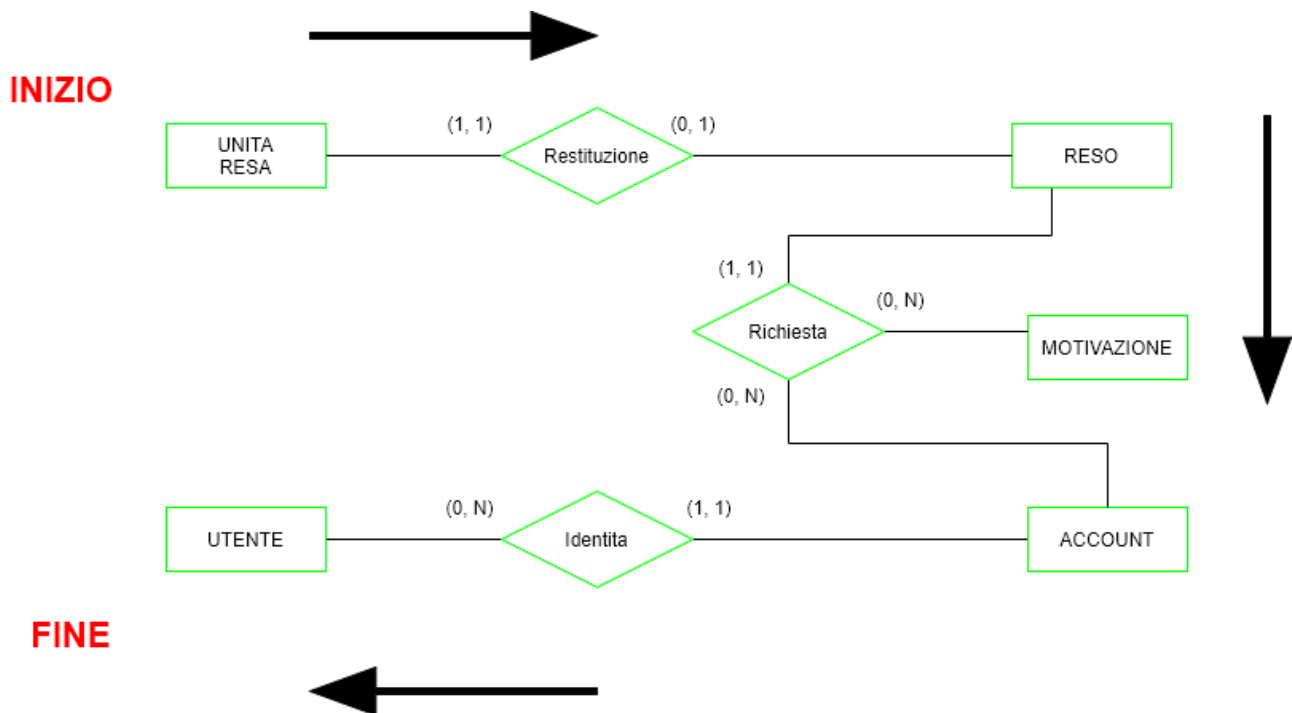
Concetto	Costrutto	Accessi	Tipo
Prodotto	E	15	L
Prodotto	E	15	S

Operazione 3: Considerato un Unità Resa, si ricavi la richiesta di reso di riferimento, indicandone motivazione e codice fiscale, nome e cognome del cliente che ne ha fatto richiesta

Dati in INPUT: UID di un'unità

Dati in OUTPUT: Codice Fiscale, nome e cognome di un cliente

Porzione del diagramma E - R interessata:



Spiegazione dell'operazione:

Avendo il codice di unità in mano, si legge in restituzione il Codice del Reso che l'ha "fatta restituire". Dopodiché si legge in richiesta quale codice di motivazione vi era associato, e si legge in motivazione la descrizione di tale codice.

Infine, sempre da richiesta, si legge l'account e, da questi, si ricavano i dati dell'utente legato all'account.

Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo
Restituzione	R	1	L
Richiesta	R	1	L
Motivazione	E	1	L
Identita	R	1	L
Utente	E	1	L

Implementazione SQL

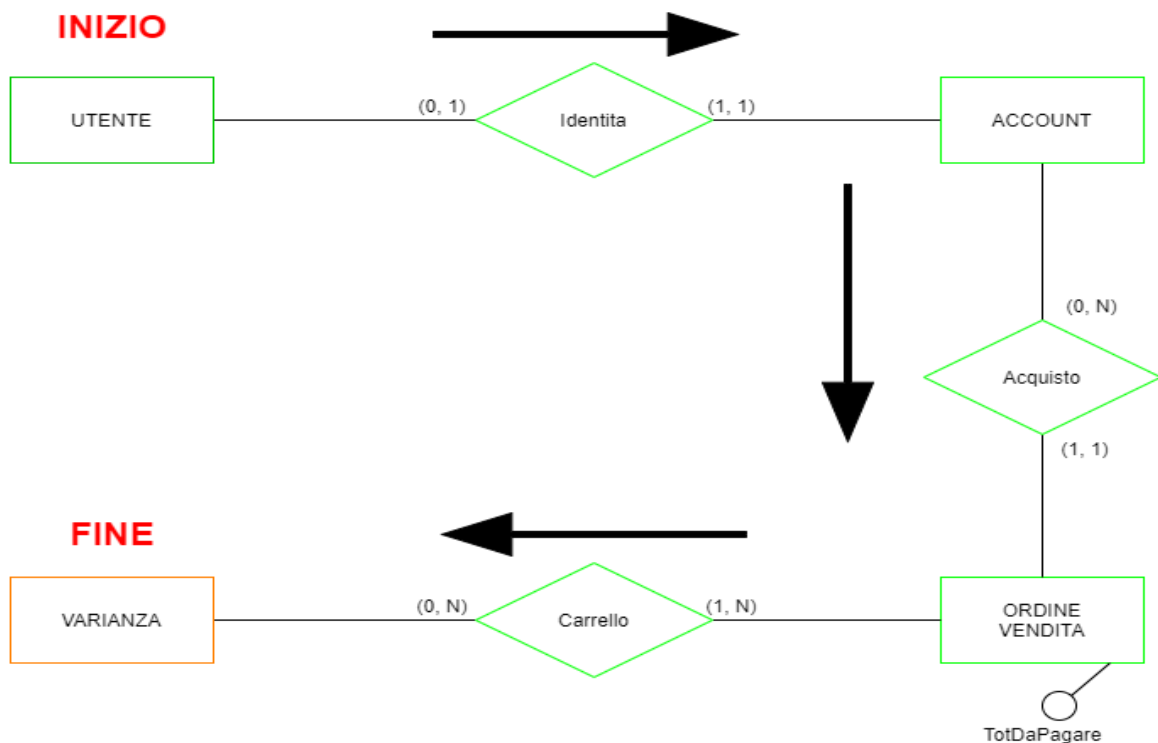
```
SELECT UR.UID,  
       UR.Reso,  
       R.Commento,  
       U.CodFiscale as 'Codice fiscale',  
       U.Nome,  
       U.Cognome  
FROM unitarese UR  
     INNER JOIN  
reso R  
     ON UR.Reso = R.Codice  
     INNER JOIN  
 richiestareso RR  
     ON RR.Reso = R.Codice  
     INNER JOIN  
account A  
     ON A.NomeUtente = RR.Account  
     INNER JOIN  
utente U  
     ON A.Utente = U.CodFiscale  
WHERE UR.UID = @input;
```

Operazione 4: Considerato un cliente, considerare quanto hanno speso nell'ultimo ordine effettuato con successo

Dati in INPUT: Codice fiscale di un utente

Dati in OUTPUT: Un numero indicante la spesa effettuata nell'ultimo ordine

Porzione di Diagramma Coinvolta:



Spiegazione Operazione:

Un utente compra, tendenzialmente, diversi ordini, e l'azienda conta circa 2500 ordini al giorno. Quest'operazione è effettuata sulla base periodica di voler assegnare dei buoni sconto a utenti che spendono abbastanza in un singolo ordine.

Si sceglie di valutare la ridondanza **TotDaPagare (attributo della entità OrdineVendita)**, ottenibile valutando i singoli prezzi dei record di *Varianza* scelti da quell'ordine. Assumendo che una persona acquisti mediamente 3 prodotti ad ordine:

Tavola degli Accessi (SENZA RIDONDANZA)

Concetto	Costrutto	Accessi	Tipo
Identità	R	1	L
Acquisto	R	1	L
OrdineVendita	E	1	L
Carrello	R	3	L
Varianza	E	3	L

Tavola degli Accessi (CON RIDONDANZA)

Concetto	Costrutto	Accessi	Tipo
Identità	R	1	L
Acquisto	R	1	L
OrdineVendita	E	1	L

Si procede all'analisi della ridondanza:

- $f^T = 2500$
- $o^T = 9$
- $n^T = 2500 * 9 = 22500$
- $o_{RID}^T = 3$
- $n_{RID}^T = 7500$
- $\Delta_{read} = n^T - n_{RID}^T = 15000$
- $g^A = 2500$
- $o^A = 3 + 2 = 5$
- $n^A = g^A * o^A = 12500$

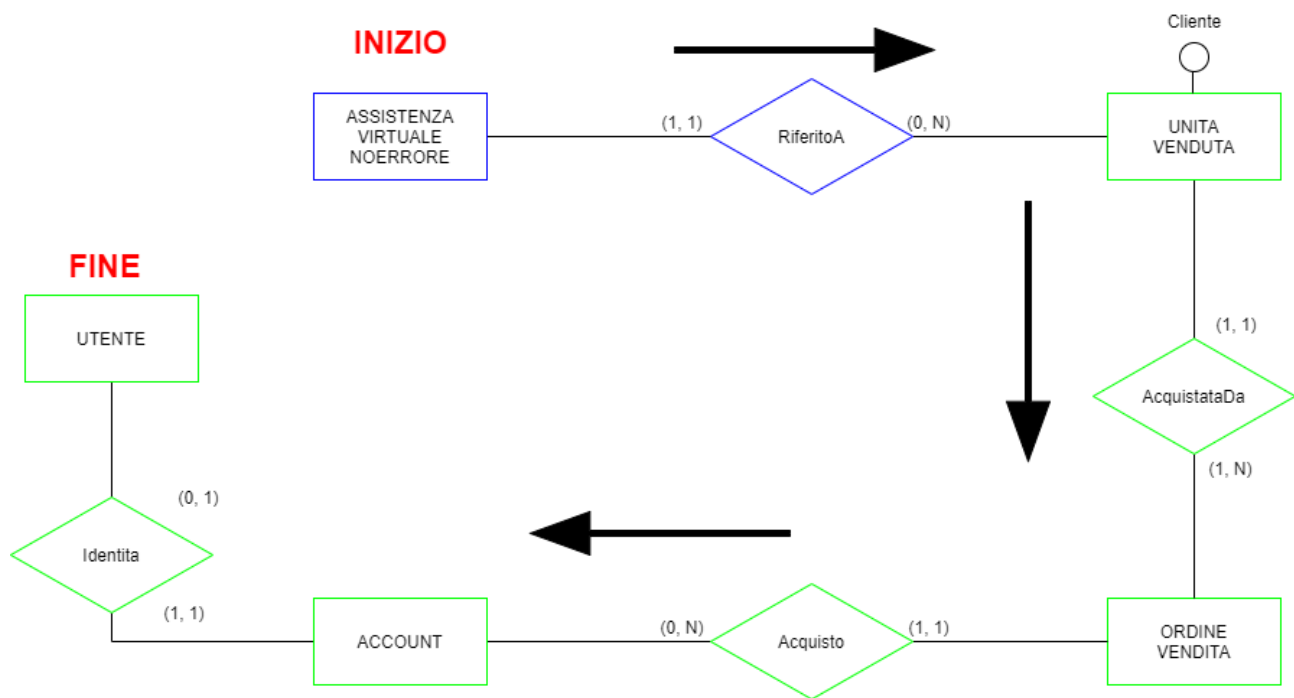
Rimandando alla [tavola](#) relativa all'aggiornamento della ridondanza segue che è conveniente

Operazione 5: Attivazione di un'Assistenza Virtuale (in assenza di un codice d'errore) da parte di un Utente.

Dati in INPUT: Codice Fiscale dell'Utente e UID dell'unità guasta.

Dati in OUTPUT: nuova istanza della tabella AssistenzaVirtualeNOErrore.

Porzione di Diagramma E-R coinvolta:



Spiegazione Operazione

L'attivazione di un'Assistenza Virtuale prevede un'operazione di inserimento nella tabella AssistenzaVirtualeNOErrore. L'insert produrrà esito positivo (nel database comparirà una nuova istanza della tabella AssistenzaVirtualeNOErrore) se esiste un'UnitaVenduta avente *UID* uguale a quello fornito e se l'utente che ha comprato quell'unità è lo stesso che richiede l'assistenza.

Si valuta la ridondanza **CienteVendita (attributo nell'entità UnitaVendute)** ricavabile a ritroso mediante l'attributo *Ordine* già presente in UnitaVenduta.

Tavola degli Accessi (SENZA RIDONDANZA):

Concetto	Costrutto	Accessi	Tipo
AssistenzaVirtualeNOErrore	E	1	S
RiferitoA	R	1	L
UnitaVenduta	E	1	L
AcquistataDA	R	1	L
Acquisto	R	1	L
Identita	R	1	L

Quotidianamente si vuole

In media 350 accessi elementari al giorno (l'operazione è eseguita 50 volte al giorno)

Tavola degli Accessi (CON RIDONDANZA)

Concetto	Costrutto	Accessi	Tipo
AssistenzaVirtualeNOErrore	E	1	S
RiferitoA	R	1	L
UnitaVenduta	E	1	L
Identita	R	1	L
Identita	R	1	L

Si procede all'analisi della ridondanza:

- $f^T = 50$
- $o^T = 7$
- $n^T = 50 * 7 = 350$
- $o_{RID}^T = 5$
- $n_{RID}^T = 250$
- $\Delta_{read} = n^T - n_{RID}^T = 100$
- $g^A = 7500$ (Unità Vendute al giorno)
- $o^A = 1 + 1 + 2 = 4$
- $n^A \gg \Delta_{read}$

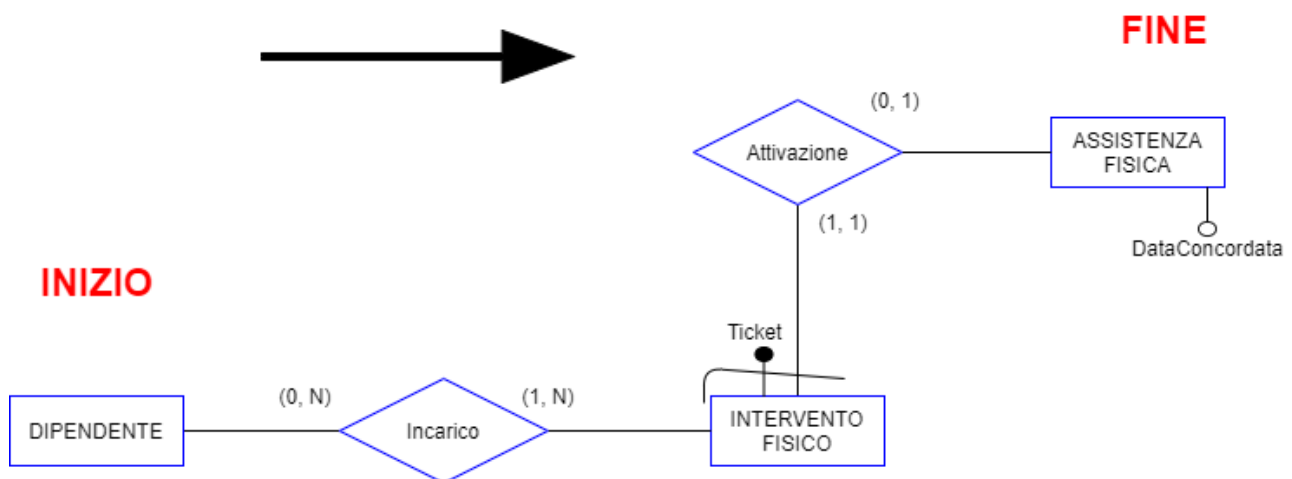
Si sceglie di **Eliminare la Ridondanza !**

Operazione 6: Considerato un Tecnico, ricavare gli Interventi Fisici programmati per la giornata.

Dati in INPUT: Codice fiscale del tecnico.

Dati in OUTPUT: Tabella le cui righe rappresentano, per un dato Tecnico, gli interventi fisici programmati nella giornata odierna.

Porzione di Diagramma E-R coinvolta



Spiegazione Operazione

Ogni giorno, ogni tecnico dell'azienda, ha una lista di interventi fisici da compiere nell'arco della giornata. Dunque, dato un tecnico, si vuole ricavare la lista degli interventi aventi *DataConcordata* = giorno corrente.

Tavola degli Accessi

Concetto	Costrutto	Accessi	Tipo
Incarico	R	3	L
InterventoFisico	E	3	L
Attivazione	R	3	L
AssistenzaFisica	E	3	L

(Si è assunto che, in media, un tecnico compia tre interventi al giorno).

Implementazione SQL:

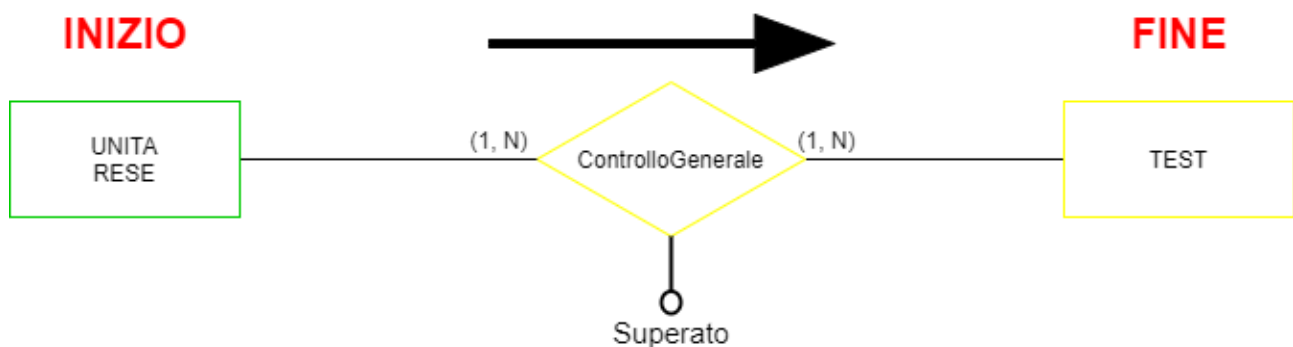
```
SELECT T.CodiceFiscale AS Tecnico,
       INF.Data as Giorno,
       I.InterventoFisico as 'Intervento Fisico',
       U.CodFiscale as 'Utente',
       CONCAT(U.Indirizzo, ', ', U.Citta, ', ', U.Provincia) as 'Indirizzo'
FROM Tecnico T
      INNER JOIN
    incarico I
      ON T.CodiceFiscale = I.Tecnico
      INNER JOIN
    interventofisico INF
      ON INF.Ticket = I.InterventoFisico
      INNER JOIN
    assistenzafisica A
      ON A.CodiceAssistenza = INF.AssistenzaFisica
      INNER JOIN
    unitavendute UV
      ON UV.UID = A.UnitaVendute
      INNER JOIN
    account AC
      ON AC.NomeUtente = UV.ClienteVendita
      INNER JOIN
    utente U
      ON U.CodFiscale = AC.Utente
WHERE INF.Data = '2020-05-06'
      AND INF.Stato <> 'Finito';
```

Operazione 7: Attivazione di una procedura di controllo dello stato di un'unità.

Dati in INPUT: codice UID di un'unità oggetto di reso, codice del Test, esito del test.

Dati in OUTPUT: nuove istanze della tabella ControlloGenerale;

Porzione di Diagramma E-R coinvolta



Spiegazione Operazione

Una procedura di ricondizionamento prevede, prima di procedere alla sostituzione delle parti usurate/rotte, un controllo generale delle condizioni delle parti che costituiscono il prodotto. Ogni unità è sottoposta ad una serie di Test. Il compimento del Test e l'esito dello stesso sono salvati nella tabella ControlloGenerale. Dunque, ogni procedura di controllo necessita, per essere rintracciabile nella sua interezza, di N inserimenti nella tabella ControlloGenerale (N = Numero test previsti per la tipologia di prodotto relativa all'unità).

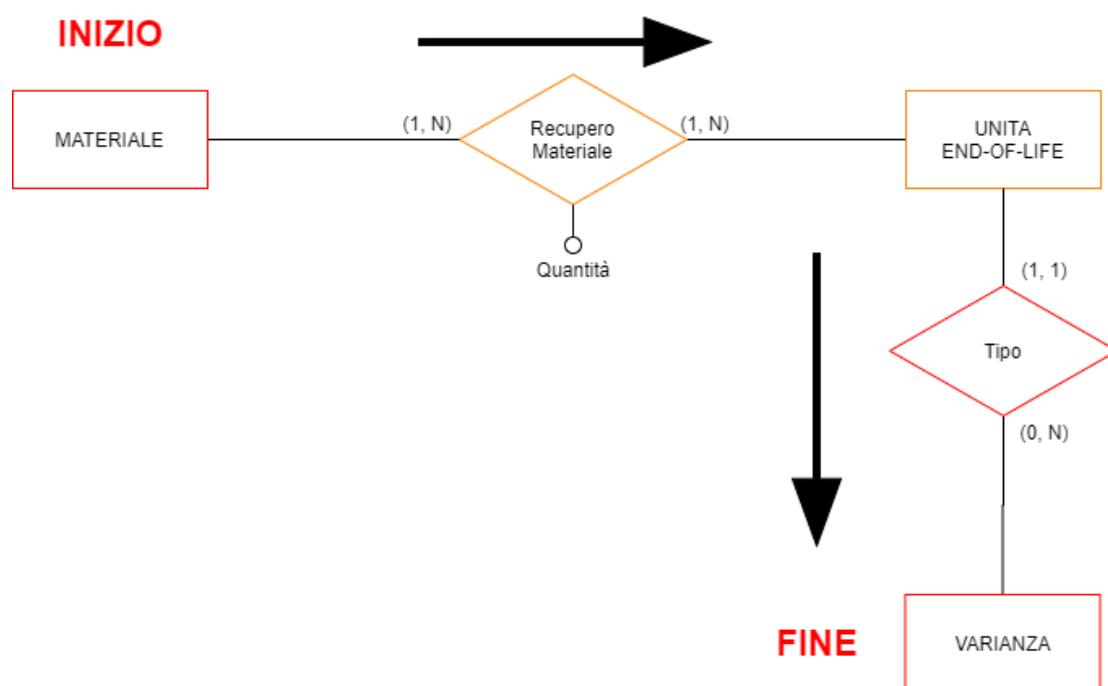
Tavola degli Accessi

Concetto	Costrutto	Accessi	Tipo
UnitaRese	E	1	L
ControlloGenerale	R	1	S
Test	E	1	L

Operazione 8: Considerato un prodotto, visualizzare la quantità media di materiali (espressa in kg) oggetto di procedura di recupero.

Dati in INPUT: Codice Prodotto

Dati in OUTPUT: Tabella le cui righe rappresentano i prodotti a listino e le cui colonne rappresentano i materiali oggetto di recupero durante la procedura di smaltimento.



Porzione di Diagramma E-R coinvolta

Spiegazione Operazione

Considerata un'UnitaEnd-Of-Life, è possibile ricavare quali materiali e in quali quantità sono stati recuperati. Notando che ogni unità fa parte di una certa tipologia di prodotto, possiamo conoscere quanti kg di materiale/i sono stati recuperati in tutte le procedure di smaltimento e farne una media.

Tavola degli Accessi

Concetto	Costrutto	Accessi	Tipo
UnitaEnd-Of-Life	E	1	L
RecuperoMateriale	R	3	L
Tipo	R	1	L
Materiale	E	3	L

(Si è assunto che, in media, da ogni unità vengano recuperati 3 materiali).

Considerato che per ogni prodotto in media esistono 10.000 Unità End-Of-Life, si hanno circa 80.000 accessi totali in lettura alla base di dati.

Implementazione SQL

```
WITH MaterialiRecuperati AS (
SELECT DISTINCT RM.Materiale
FROM recuperomateriale RM
    INNER JOIN
    unitaendoflife U
    ON RM.UnitaEndOfLife = U.UID
    INNER JOIN
    prodotto P
    ON P.CodProdotto = U.Prodotto
WHERE P.CodProdotto = 'Aiph2017'
)
SELECT GROUP_CONCAT(CONCAT('TRUNCATE(SUM(IF(RM.Materiale = '', A.Materiale, '', RM.Quantita, 0)), 4) AS'', A.Materiale, ' (grammi)', '''))
FROM MaterialiRecuperati A
INTO @pivot_query;

SET @pivot_query = CONCAT('SELECT U.Prodotto, ', @pivot_query, 'FROM recuperomateriale RM INNER JOIN UnitaEndOfLife U ON RM.UnitaEndOfLife = U.UID
WHERE U.Prodotto = '', 'Aiph2017', ''');

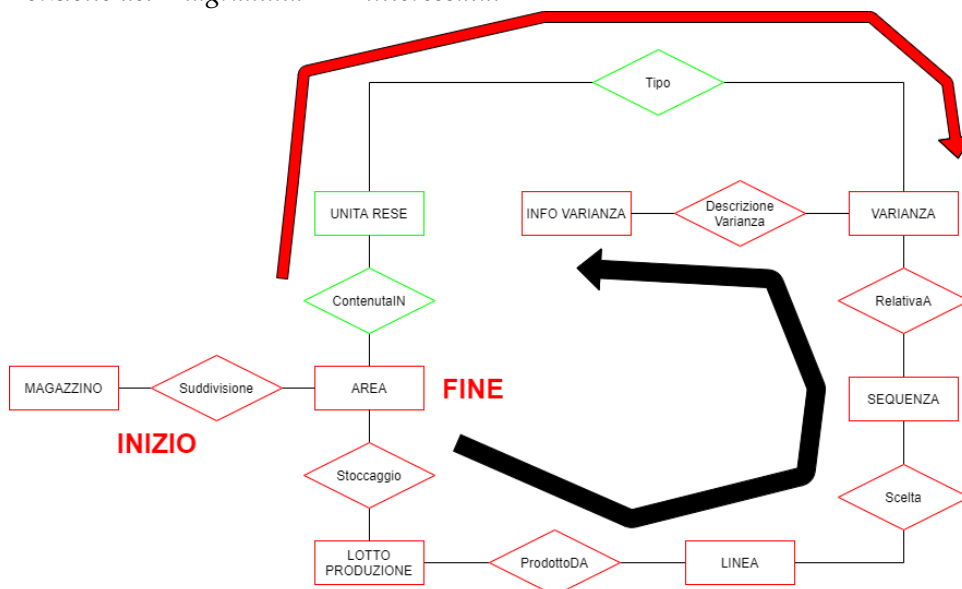
PREPARE sql_statement FROM @pivot_query;
EXECUTE sql_statement;
```

Operazione 9: Verificare lo spazio disponibile di una certa area di un certo magazzino

Dati in INPUT: Un codice di magazzino e una sua area

Dati in OUTPUT: Lo spazio disponibile rimasto in quell'area

Porzione del Diagramma E-R interessata:



Spiegazione Operazione:

Nel valutare gli accessi di quest'operazione, e nell'evitare di fare conti troppo esagerati, in proporzione diremo che un'area magazzino ospita 5 lotti di produzione di ognuno 10 unità, e 50 unità rese. (Si ricordi che, per come sono stati pensati, i lotti smaltimento non fanno "ingombro"). Valutiamo quindi la ridondanza CapienzaDisponibile (attributo nell'entità Area).

Senza Ridondanza

Concetto	Costrutto	Accessi	Tipo
Magazzino	E	1	L
Suddivisione	R	1	L
Area	E	1	L
Stoccaggio	R	5	L
ProdottoDA	R	5	L
Scelta	R	5	L
RelativaA	R	5	L
DescrizioneVarianza	R	5	L
Info Varianza	E	55	L
ContenutaIN	R	50	L
Tipo	R	50	L

Con Ridondanza

Concetto	Costrutto	Accessi	Tipo
Magazzino	E	1	L
Suddivisione	R	1	L
Area	E	1	L

Si procede all'analisi della ridondanza:

- $f^T = 20$
- $o^T = 183$
- $n^T = 20 * 183 = 3660$
- $o_{RID}^T = 3$
- $n_{RID}^T = 30$
- $\Delta_{read} = n^T - n_{RID}^T = 3630$

Aggiornamento Ridondanza in caso di Stoccaggio Lotto

- $g^A = 3$
- $o^A = 1 + 1 + 1 + 1 + 1 + 2 = 7$
- $n^A = g^A * o^A = 21$

Aggiornamento Ridondanza in caso di Vendita

- $g^A = 500$
- $o^A = 1 + 1 + 1 + 2 = 5$
- $n^A = g^A * o^A = 2500$

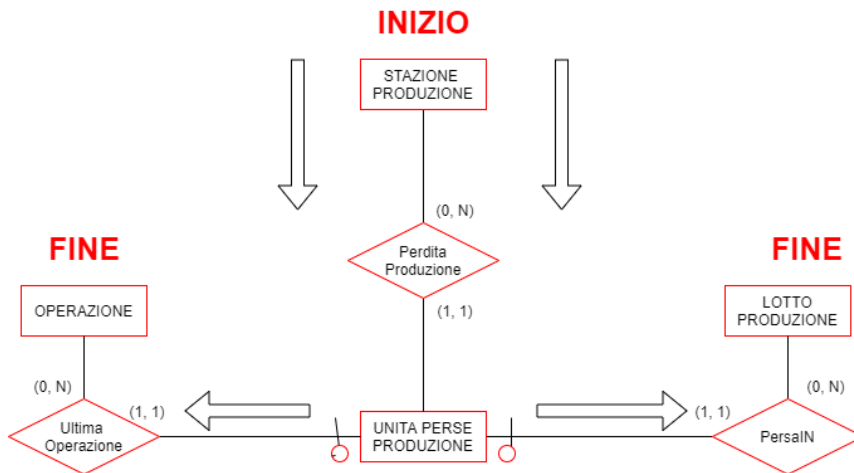
Poiché in entrambe le possibilità la ridondanza risulta conveniente, si sceglie di mantenerla.

Operazione 10: Inserisci un'unità persa per una stazione di produzione (conoscendo a priori l'ultima operazione e il lotto)

Dati in INPUT: Un ID operazione e un CodLotto identificativo di un lotto produzione

Dati in OUTPUT: Nessuno

Porzione di Diagramma E-R coinvolta:



Spiegazione Operazione:

Nel compiere questa operazione, si notano subito due problemi:

- Va aggiornata la ridondanza DaRecuperare presente in Stazione Produzione (incrementata almeno)
- Va aggiornata, alla fine del tempo T della linea, la ridondanza che indica il numero di Unità Effettive (attributo di LottoProduzione)

Tavole degli Accessi

Senza Ridondanze

Concetto	Costrutto	Accessi	Tipo
Operazione	E	1	L
LottoProduzione	E	1	L
UltimaOperazione	R	1	S
PersaIN	R	1	S
UnitaPerseProduzione	E	1	S
StazioneProduzione	E	1	L
PerditaProduzione	E	1	S

Solo con ridondanza "Da Recuperare"

Concetto	Costrutto	Accessi	Tipo
Operazione	E	1	L
LottoProduzione	E	1	L
UltimaOperazione	R	1	S
PersaIN	R	1	S
UnitaPerseProduzione	E	1	S
StazioneProduzione	E	1	L
PerditaProduzione	E	1	S
StazioneProduzione	E	1	S

Solo con ridondanza “UnitàEffettive”

Concetto	Costrutto	Accessi	Tipo
Operazione	E	1	L
LottoProduzione	E	1	L
UltimaOperazione	R	1	S
PersaIN	R	1	S
UnitaPerseProduzione	E	1	S
StazioneProduzione	E	1	L
PerditaProduzione	E	1	S
LottoProduzione	E	1	S

Con Entrambe le ridondanze

Concetto	Costrutto	Accessi	Tipo
Operazione	E	1	L
LottoProduzione	E	1	L
UltimaOperazione	R	1	S
PersaIN	R	1	S
UnitaPerseProduzione	E	1	S
StazioneProduzione	E	1	L
PerditaProduzione	E	1	S
StazioneProduzione	E	1	S
LottoProduzione	E	1	S

- $f^T = 10$
- $o^T = 11$
- $n^T = 10 * 11 = 110$

In presenza di “DaRecuperare”

- $o_{RID}^T = 11$
- $n_{RID}^T = 110$
- $\Delta_{read} = n^T - n_{RID}^T = 0$
- $g^A = 10$
- $o^A = 1 + 2 = 3$
- $n^A = g^A * o^A = 30$

In presenza di “UnitàEffettive”

- $o_{RID}^T = 11$
- $n_{RID}^T = 110$
- $\Delta_{read} = n^T - n_{RID}^T = 0$
- $g^A = 10$
- $o^A = 1 + 2 = 3$
- $n^A = g^A * o^A = 30$

In Presenza di Entrambe

- $o_{RID}^T = 11$
- $n_{RID}^T = 110$
- $\Delta_{read} = n^T - n_{RID}^T = 0$

- $g^A = 10$
- $o^A = 1 + 1 + 2 + 2 = 8$
- $n^A = g^A * o^A = 80$

Nel valutare le ridondanze rimandiamo prima alla lettura dell'operazione successiva.

Operazione 11: Restituire quante unità si stanno perdendo al momento per un certo Lotto fornito da utente

Dati in INPUT: Codice identificativo di un lotto produzione

Dati in OUTPUT: Un numero che indichi quante unità quel lotto perde in quel momento

Porzione di diagramma E-R coinvolta:



Spiegazione Operazione:

- Anche qui concorre la ridondanza Unita Effettive (attributo nell'entità LottoProduzione) .Si noti che l'operazione è particolarmente semplice mantenendola (1 sola lettura).

In particolare, nel calcolo che segue assumiamo per semplicità che una linea sia composta di 8 stazioni e tre di esse perdono un'unità.

Tavole degli Accessi:

Senza Ridondanze

Concetto	Costrutto	Accessi	Tipo
LottoProduzione	E	1	L
PersaIN	R	3	L

Con ridondanza "UnitaEffettive"

Concetto	Costrutto	Accessi	Tipo
LottoProduzione	E	1	L

- $f^T = 4$
- $o^T = 4$
- $n^T = 4 * 4 = 16$

In presenza di "DaRecuperare"

- $o_{RID}^T = 3$
- $n_{RID}^T = 12$
- $\Delta_{read} = n^T - n_{RID}^T = 4$
- $g^A = 10$
- $o^A = 1 + 2 = 3$
- $n^A = g^A * o^A = 30$

In presenza di "UnitàEffettive"

- $o_{RID}^T = 1$
- $n_{RID}^T = 4$
- $\Delta_{read} = n^T - n_{RID}^T = 12$
- $g^A = 10$
- $o^A = 1 + 2 = 3$
- $n^A = g^A * o^A = 30$

In Presenza di Entrambe

- $o_{RID}^T = 1$
- $n_{RID}^T = 4$
- $\Delta_{read} = n^T - n_{RID}^T = 12$
- $g^A = 10$
- $o^A = 1 + 1 + 2 + 2 = 8$
- $n^A = g^A * o^A = 80$

Lo scenario migliore è dunque quello in cui Δ_{read} è massimo e n^A è minimo: scenario che si presenta mantenendo unicamente la ridondanza *UnitaEffettive*.

A questo proposito, si sceglie in ogni caso di eliminare la ridondanza DaRecuperare, mentre si sceglie (anche se il rapporto costi - benefici **sembra** essere contrario), di mantenere la seconda ridondanza. Questo perché si porta dietro un carico informativo usato e, in generale, utilizzabile per altri scopi. Ad esempio:

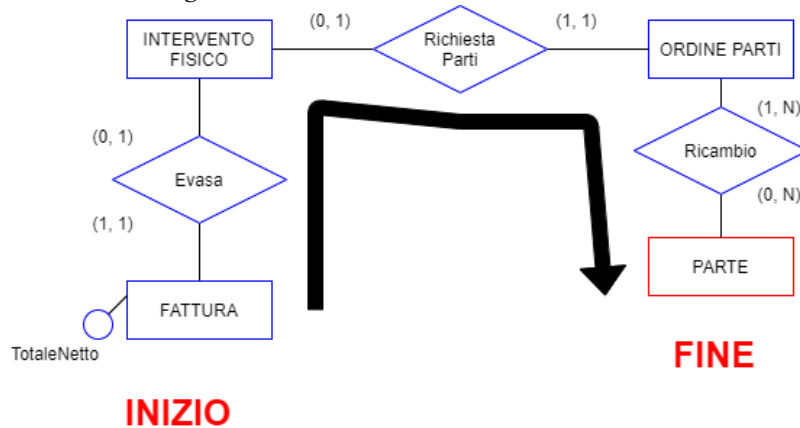
- Se a fine giornata si dovessero portare i lotti *finiti* (senza unità da recuperare) in magazzino, risulterebbe particolarmente vantaggioso discriminare velocemente quali lotti lo sono e quali no. La ridondanza *UnitaEffettive* rappresenta tra le altre cose anche questa informazione (infatti se $UnitàPreviste = UnitàEffettive$ il lotto è finito);
- Analizzare una linea (come fa l'azienda) è un processo che deve coinvolgere i dati rilevati solo dai lotti finiti e *soprattutto* deve memorizzare il primo scostamento dalle unità previste. Dunque, per i motivi di cui immediatamente sopra, la ridondanza trova utilità anche in questa analisi;
- Ed altri possibili utilizzi che l'azienda potrebbe voler implementare in una fase successiva.

Operazione 12: Considerate le fatture del giorno, ricavare la somma dei totali netti

Dati in INPUT: Nessuno

Dati in OUTPUT: Somma dei totali netti di ogni fattura la cui data sia il giorno corrente

Porzione del diagramma E-R interessata:



Spiegazione Operazione:

Ricordiamo ai lettori che *TotaleNetto* (attributo dell'entità Fattura) è un attributo derivabile moltiplicando il numero 10 (che è il pagamento orario dei tecnici) per le ore di lavoro dell'intervento che si sta fatturando, a cui si aggiunge la spesa delle parti di ricambio.

Nel calcolare gli accessi di questa operazione, assumeremo che mediamente un'intervento fisico richieda di ordinare 2 pezzi di ricambio, e che tuttavia il 33% degli interventi non richieda alcun ricambio.

Assumiamo poi che in un giorno vengano erogate 50 fatture.

L'operazione è la seguente: "Considerate le fatture del giorno, ricavare la somma dei totali netti"

Senza Ridondanza

Concetto	Costrutto	Accessi	Tipo
Fattura	E	200	L
Evasa	R	200	L
InterventoFisico	E	200	L
RichiestaParti	R	134	L
Ricambio	R	268	L
Parte	E	268	L

Con Ridondanza *TotaleNetto*

Concetto	Costrutto	Accessi	Tipo
Fattura	E	200	L

Si procede all'analisi della ridondanza:

- $f^T = 2$
- $o^T = 320$
- $n^T = 1 * 3 = 2540$
- $o_{RID}^T = 200$
- $n_{RID}^T = 400$
- $\Delta_{read} = n^T - n_{RID}^T = 2140$
- $g^A = 200$
- $o^A = 2 + 1 + 1 + 1 + 2 + 2 = 9$
- $n^A = g^A * o^A = 1800$

Si sceglie di **mantenere la ridondanza**.

DATA ANALYTICS

In questa fase verranno esaminate le tre data analytics più importanti dell'azienda solo dal punto di vista concettuale. Il codice delle stesse sarà introdotto ed esaminato più avanti nella documentazione.

Diagnosi intelligente dei guasti: CBR

L'obiettivo di questa analytics è, fondamentalmente, creare e sfruttare una *base di conoscenza*, che riesca a darci tutte le informazioni affinché si riesca a catalogare un guasto come "simile a un altro".

Come già suggerito dalle specifiche, la modellizzazione del ragionamento umano avviene in quattro fasi: *retrieve, reuse, revise, retain*. Anzitutto spiegheremo le prime due, che sono le più importanti a livello concettuale:

Fase 1: RETRIEVE

In questa fase si prendono in Input i sintomi di un guasto, e si cercano casi di guasto *simili* che siano stati risolti in precedenza.

Si assume che due guasti siano *simili se e solo se hanno in comune almeno il 66% dei sintomi*. La scelta di una percentuale così particolare deriva dal fatto che un guasto si presenta con in media "soli" tre sintomi e, di conseguenza, sono da considerarsi simili guasti che abbiano 2/3 (due terzi) dei sintomi in comune.

A livello di schema, per rappresentare questo concetto ci viene in aiuto l'entità *Conoscenza*, che costituisce un archivio storico dei guasti risolti, e la data in cui essi sono stati risolti. Esaminiamo per bene gli attributi di questi:

- **A. I. M. :** Acronimo di *Artificial Intelligence Memory*, indica il codice del "ricordo". Un codice di questo tipo racchiude, come vedremo in seguito, gran parte delle informazioni necessarie alla risoluzione del guasto
- **DataRisoluzione:** la data, nel formato "YYYY-MM-GG" in cui il guasto è stato risolto. Se per qualche motivo il pezzo in assistenza non è mai tornato al cliente (problemi di pagamento, altro), la base di conoscenza deve comunque arricchirsi dell'esperienza!
- **Cardinalità (1, 1) dell'associazione collegata a Guasto:** quest'associazione verrà accorpata, in fase di traduzione, all'entità *Conoscenza*, che sarà arricchita dell'attributo "Guasto" che indica solo e soltanto il **codice del guasto risolto**

La conoscenza è legata poi (cardinalità 1, N) sia a sintomo, e si indicano espressamente i sintomi indicati dall'utente in fase di richiesta assistenza fisica, sia a rimedio, dove si esprimono i rimedi utilizzati per risolvere il guasto.

Analizziamo ora il processo di richiesta:

- Anzitutto, la CBR entra in funzione solo in caso di **assistenza fisica**; non è infatti uno strumento a disposizione dell'utente, ma del tecnico.
- Quando (per motivi vari) si va incontro a una richiesta di assistenza fisica, l'utente è tenuto ad indicare un gruppo di sintomi (almeno 1)

Ora l'obiettivo di questa fase è trovare i guasti che, nella base di conoscenza, presentavano inizialmente gli stessi sintomi (almeno il 66%). Da qui si ricavano codici di guasti risolti in date diverse, potenzialmente con rimedi diversi:

- Se il numero di guasti precedentemente risolti è ≤ 10 , allora verranno considerati tutti
- Se il numero di guasti precedentemente risolti è > 10 , si cercherà di considerare i dieci più recenti nell'anno corrente, con la possibilità di andare ancora più indietro qualora non si arrivi al numero massimo di guasti trovati (10).

Attenzione: Il numero massimo (10) è da **ricercare, ma non è necessario**.

Si ricava inoltre un valore, *Compatibilità*, che esprime in percentuale il numero dei sintomi in comune con quel guasto. È chiaro che più questo valore è alto, più il guasto ha una priorità maggiore nell'essere considerato.

Alla fine, si troveranno i 10 casi di guasti con maggiore compatibilità (recente). Non è detto, inoltre che in questi 10 casi non ci siano ripetizioni dello stesso guasto (magari a sintomi diversi).

Chiameremo questo tipo di guasti *GuastoProbabile* e formeremo record del tipo: A. I. M. / Data / *GuastoProbabile* / *Rimedio* (identificati dalla coppia A. I. M. / *Rimedio*).

Termina qui la fase di **Retrieve**.

Fase 2: REUSE

In questa fase si rappresenta l'essenza della funzionalità, il "consiglio" vero e proprio. Il tecnico è in grado di visualizzare, al termine del Retrieve, una lista di rimedi e il codice del guasto per il quale sono stati utilizzati (chiaramente i rimedi sono ricavati dalla contestualizzazione di cui sopra).

Dopodiché, a fianco, si visualizza uno **score**, che determina l'*efficacia del rimedio* sul guasto. Per calcolare l'efficacia di un rimedio, si fa fede a due semplici linee indicative:

1. Si scorre l'elenco dei 10 (o meno) casi di guasto probabile presenti in memoria: per ogni caso, si ricerca il rimedio nell'elenco dei rimedi utilizzati. Per ogni caso in cui il rimedio è presente, si assegna un punteggio iniziale in base alla compatibilità calcolata:
 - *Compatibilità 80% - 100%* : $5 * (10 / \text{NumeroGuasti})$ punti di score
 - *Compatibilità 66% - 80%* : $3 * (10 / \text{NumeroGuasti})$ punti di score

Per ogni caso, si sommano i punteggi di score. Questa somma iniziale determina un valore che chiameremo *ScoreIniziale*

2. Si conta quante volte il rimedio è stato utilizzato nei casi di guasto probabile, e si aggiunge allo score un numero pari a questa conta Il tecnico visualizza infine i rimedi suggeriti con relativo score, e termina così la fase di **reuse**.

Fase 3 e 4: REVISE e RETAIN

In questa fase il tecnico prova a risolvere i sintomi del guasto operando i rimedi (partendo ovviamente da quelli con score più alti).

Se i rimedi non dovessero essere sufficienti, o se si ritiene di utilizzarne altri diversi, finora mai utilizzati, si controlla che quel rimedio esista già nel DB. Se non esistono, vengono prima inseriti i rimedi, e poi si cerca di risolvere il guasto.

Se il guasto viene risolto e questi aveva almeno il 90% di compatibilità, si aprono due strade:

- Se si ritiene che i rimedi utilizzati siano simili o pressoché identici a quelli utilizzati nei guasti precedenti ad alta compatibilità, si aggiorna la base di conoscenza aggiornando la data del

guasto più compatibile alla data della riparazione, aggiungendo i nuovi sintomi (se sono presenti)

- Se si ritiene che i rimedi siano particolarmente diversi, allora verranno aggiunti anche i nuovi rimedi, ma nella base di conoscenza sarà introdotto un nuovo record (il guasto non verrà ricodificato, ma si scrive comunque una nuova riga con i dati).

Se il guasto viene risolto ma questi aveva meno del 90% di compatibilità, il tecnico è incaricato di inserire il nuovo guasto, codificandolo e inserendo una descrizione, dopodiché di introdurre la riparazione nella base di conoscenza (arricchendo il tutto con i sintomi trovati e i rimedi utilizzati).

Analisi dell'efficienza di una linea di produzione

Una linea di produzione ha principalmente due indicatori di performance:

1. Quante unità perse in media per lotto
2. Velocità di produzione di un lotto

Spiegazione punto 1: Come può leggersi sopra, un'unità è da considerarsi "momentaneamente persa" se, trascorso il tempo T indicato dalla linea, non sono riuscite ad effettuarsi tutte le operazioni previste dalla stazione che la sta effettuando.

Poiché una stazione può perdere ripetutamente delle unità, il conteggio medio non è mai da considerarsi esatto, ma con alcune regole adottate dall'azienda può considerarsi sufficientemente affidabile.

Assumiamo dunque che il numero di unità perse a cui facciamo riferimento sia la differenza iniziale tra gli attributi *Unità Previste* e *Unità Effettive* di Lotto di Produzione. Come è stato precedentemente indicato, questo numero arriva ad essere uguale solo quando tutte le unità perse sono state recuperate.

Si sceglie come indicatore di performance il numero **iniziale**, nessun numero intermedio!

Questo significa che se, ad esempio, le stazioni della linea A dovessero perdere 10 unità prima di iniziare a recuperarle, il numero dovrà essere esattamente 10, non uno di più, non uno di meno.

Questo numero non rappresenta tuttavia alcuna media. La media di cui al punto 1 si riferisce alla media di tutte le unità perse inizialmente ogni 5 lotti. Non è previsto ricalcolo / aggiornamento della media prima che siano stati prodotti altri 5 lotti successivi.

Si evince dunque il fatto che la linea sia tanto più produttiva quanto più piccola è, nel tempo, questa media.

Spiegazione punto 2: Quanto è veloce una linea? Anche questo indicatore non sembra troppo reale e preciso, ma, come sopra, ci affidiamo a delle semplici regole per cercare di ricavarne un dato d'interesse.

Anzitutto il dato più importante è il tempo T della linea. Più questo tempo è ridotto, più le stazioni lavorano meno. Tuttavia, come illustrato nelle stesse specifiche, un tempo T troppo basso potrebbe aumentare il numero delle unità perse che, come sappiamo, rappresenta un controsenso dell'indicatore 1.

L'obiettivo da raggiungere è trovare il **minimo tempo T affinché il numero delle unità perse sia il più ridotto possibile.**

A questo punto, sorge però un'altra problematica: un operatore potrebbe essere assegnato a una stazione riferita ad operazioni sulle quali egli è molto lento. Ciò significa che, per permettere il numero minimo di unità perse, si può procedere in due modi:

- "Sperare" che sia l'unico lento e che le unità perse della linea dipendano solo da lui (per come è stata costruita la media, questo significherebbe infatti che si perde solo 1 unità)
- Aumentare il tempo T della linea, permettendogli di svolgere le stesse operazioni, ma con più tempo a disposizione.

Ci accorgiamo che entrambe le "soluzioni" sono errate e prive di metodo.

L'azienda decide dunque di introdurre una nuova soluzione: *le classi Campione.*

Una classe Campione è un set di operazioni che l'operatore svolge quando viene assunto. Tutti gli operatori devono provare tutte le classi Campione. In questo modo, l'azienda memorizza i tempi, in minuti, di tutti i dipendenti circa l'esecuzione di ogni singolo set di operazioni previsto dall'azienda.

Ora che i concetti sono leggermente più chiari, esaminiamo con più attenzione i passi che portano a capire quanto una linea di produzione è efficiente:

A. Scelta delle operazioni da assegnare ad una stazione

Anzitutto, vengono assegnate ad una stazione operazioni di vario tipo, con l'importante assunzione che le operazioni di riferimento siano **della stessa classe (anche dello stesso livello, per rappresentare il vincolo di precedenza tecnologica)**. Si noti che appartenere alla stessa classe non significa far parte del Set di Operazioni Campione relative a una certa classe Campione, ma rappresenta comunque un'operazione simile.

B. Assegnazione degli Operatori

Si assegnano gli operatori alle stazioni, uno per volta. Per ogni stazione, si ricava la classe di riferimento, e vi si assegna l'operatore che, per quella classe, ha il tempo di esecuzione minore (ciò è possibile grazie ai test effettuati). Lo scorrimento degli operatori è sequenziale: se esistono 3 stazioni di classe A verranno assegnati i tre operatori più veloci a gestire operazioni di classe A.

Può accadere che un operatore venga assegnato ad operazioni di classe B perché è il più veloce tra i disponibili ad eseguire operazioni di quella classe, anche se in realtà era più veloce ad eseguire operazioni di classe A.

Questo non assicura dunque la sequenza di operatori potenzialmente più veloce, ma assicura invece che eventuali operazioni lente vengano eseguite soltanto alla fine. Se si scegliesse di assegnare, ad esempio, prima tutte le stazioni di classe A, poi tutte quelle di classe B, e così via, si rischierebbe di lasciare "scoperta" (in termini di velocità) una stazione potenzialmente iniziale. Questo costituirebbe un danno in quanto le unità perse, quando vengono recuperate, devono ripercorrere tutte le stazioni a partire dalla stessa che le ha perse (e dunque si rende chiaro che è preferibile *perderle alla fine*).

Qualora una stazione preveda più operatori, si assegna il più veloce e si va avanti. Terminati i primi assegnamenti si fanno tutti gli altri, seguendo le stesse regole del primo. Quest'ultima assunzione evita di assegnare solo coppie o triple di operatori particolarmente veloci, e cerca di creare un "equilibrio" nello svolgimento delle operazioni.

C. Calcolo iniziale del tempo della linea

Per ogni stazione, si calcola il tempo previsto al completamento di tutte le operazioni previste in base a due regole:

- Se l'operatore che vi lavora è soltanto uno, il tempo corrisponde ai minuti, arrotondati per eccesso, impiegato dall'operatore ad eseguire il set di operazioni campione relative alla classe di operazioni della stazione
- Se gli operatori che vi lavorano sono più di uno, si prende il tempo dell'operatore più lento (arrotondato per eccesso) e si moltiplica per un fattore 1,5 (il risultato è comunque arrotondato per eccesso).

Il tempo iniziale della linea è uguale al **massimo** dei tempi associati ad ogni stazione.

D. Analisi della performance della Linea

Operazione che avviene ogni cinque lotti. Oltre alla *media* delle unità perse (di cui al punto 1) è presente ora il fattore tempo linea. Si tenga inoltre presente il numero di unità effettive previste dal lotto d'interesse (attributo *UnitaPreviste* in Lotto di Produzione)

- Se il tempo della Linea non è mai stato modificato, e *media* è inferiore a 10% delle unità previste, la linea è considerata Ottima
- Se la *media* è compresa tra il 10% e il 50%, il tempo della linea è aumentato di tre minuti, e la valutazione è InValutazione
- Se la *media* è maggiore del 50%, la linea è da considerarsi InChiusura e il tempo della linea aumenta di 5 minuti. Se la valutazione è la stessa anche nella prossima valutazione, la linea passerà nello stato Chiusa (eventualmente la stessa postazione fisica verrà assegnata ad altri operatori o a solo parte degli stessi, ma comunque sarà tutto creato da 0 con nuova codifica).

Analisi delle vendite e pianificazione della produzione (custom analytics)

L'Azienda è interessata a svolgere delle indagini di mercato relative alla vendita dei propri prodotti. Su base trimestrale, grazie ai dati ricavati a seguito dell'indagine, pianifica il piano produttivo dei successivi tre mesi. Il piano produttivo è studiato non solo in ottica di un maggiore guadagno ma è sviluppato per rendere più efficiente ed ecosostenibile l'attività produttiva dell'azienda.

Il processo produttivo dell'Azienda si basa sulla produzione, suddivisa in lotti, di una certa quantità di prodotti presenti in listino. Il piano produttivo, oltre ai dati di vendita, tiene conto della situazione delle unità stoccate nei vari magazzini. Questo può significare che per un certo trimestre, se i valori degli *indici di convenienza produttiva* sono inferiori ad una certa soglia, un prodotto può non essere inserito nel piano produttivo e dunque non essere prodotto alcun lotto. In caso contrario il numero di lotti da produrre (e le relative unità per lotto) sono decisi dipendentemente dal valore degli *indici*.

Qui di seguito è fornita una descrizione per ogni *indice di convenienza produttiva* (nota: ogni indice si riferisce alle informazioni, aggregate e non, di un prodotto):

1. GNPU, Guadagno Netto Per Unità. Il guadagno netto è dato dalla differenza tra il prezzo di vendita del prodotto e il CP.

2. CP, **C**osto di **P**roduzione. Il costo di produzione è dato dalla somma dei costi delle parti e la somma della manodopera impegnata nell'assemblaggio.
3. CMP, **C**osto **M**edio **P**arte. Costo medio delle parti di cui è composto il prodotto.
4. PPR, **P**ercentuale **P**arti **R**ecupero. Questo indice indica quante parti, in percentuale rispetto al numero totale di parti che compone il prodotto, sono state recuperate mediamente nelle procedure di smaltimento svolte nei tre mesi precedenti.
5. CMM, **C**osto **M**edio **M**ateriale. Costo medio dei materiali di cui è composto il prodotto.
6. CMS, **C**oefficiente **M**edio **S**valutazione. Media dei coefficienti di svalutazione dei materiali che compongono il prodotto.
7. PMR, **P**ercentuale **M**ateriale **R**ecupero. Quantità percentuale (espressa in termini di peso) di materiali recuperati, mediante le procedure di smaltimento, rispetto al peso totale del prodotto.
8. UP, **U**nità **P**erse. Numero di unità perse dalle linee di produzione nel trimestre precedente.
9. UV, **U**nità **V**endute. Numero di unità vendute nel trimestre precedente.
10. US, **U**nità **S**toccate. Numero di unità stoccate nei magazzini dell'Azienda (il dato si riferisce all'ultimo giorno del trimestre).
11. UR, **U**nità **R**icondizionate. Numero di unità ricondizionate pronte ad essere commercializzate (il dato si riferisce all'ultimo giorno del trimestre).
12. USM, **U**nità **S**Maltite. Numero di unità oggetto di smaltimento.

A sostegno delle scelte produttive si è deciso di introdurre una tabella contenente uno snapshot della situazione relativa alla produzione e vendita di prodotti. Lo schema di questa tabella è il seguente:

DatiVendita (Prodotto, Variante, Anno, Trimestre, GNPU, CP, CMP, PPR, CMM, PMR, UP, UV, US, UR, USM).

Gli attributi della tabella rappresentano gli indici discussi appena sopra. Grazie a questa tabella è possibile calcolare lo *score di produzione* di ogni prodotto. Il significato dello *score* è dato dall'espressione:

$$S(p) = GNPU(p) (UV(p) - UP(p)) - [UR(p) (GNPU(p) * 0.25) + US(p) (GNPU(p) * 0.75)] + USM(p) (CMP(p) * PPR(p) + CMM(p)*PMR(p)*CMS(p))$$

Si noti che lo score rappresenta un numero dato dalla differenza della somma del guadagno ottenuto tramite la vendita e il recupero di materiali/parti (quest'ultimo da intendere come risparmio medio ottenuto grazie al riutilizzo, per la produzione di nuovi lotti, delle parti e materiali oggetto di recupero) e il guadagno sospeso, ossia il guadagno che si otterrà in futuro dalla vendita delle unità stoccate in magazzino e delle unità ricondizionate.

L'analisi di mercato influenza le scelte produttive in funzione dei risultati degli anni precedenti. Per questo motivo si è deciso di introdurre un'ulteriore tabella in cui sono riportati i dati relativi agli score ottenuti dai prodotti nei trimestri precedenti. Lo schema della tabella è il seguente:

Score(Prodotto, Variante, Anno, Trimestre, Punteggio).

L'attivazione della produzione di un lotto viene consigliata in funzione dello score del trimestre precedente, denominato $S(p)$, e dello score medio dei precedenti 4 trimestri, denominato $SAP(p)$:

- A. Se $S(p) < SAP(p)*0.5$ si suggerisce di non attivare la produzione di un lotto di prodotti p.
- B. Se $SAP(p)*0.5 \leq S(p) \leq SAP(p)$ si suggerisce di attivare la produzione di un lotto di prodotti p avente un numero di unità uguale al numero di unità prodotte con il lotto più recente.

- C. Se $SAP(p) \leq S(p) \leq SAP(p) \cdot 1.5$ si suggerisce di attivare la produzione di un lotto di unità p avente un numero di unità, aumentato del 25%, uguale al numero di unità prodotto con il lotto più recente.
- D. Se $S(p) > SAP \cdot 1.5$ si suggerisce di attivare la produzione di un lotto di unità p avente un numero di unità pari al doppio delle unità prodotto con il lotto più recente.

Per una più veloce e facile fruizione di queste informazioni si è deciso di implementare una funzione, chiamata *scoreProdotto()*, che preso in input il codice identificativo di un prodotto p e la sua variante v restituisce una stringa in funzione dei 4 casi presentati di sopra:

Caso A. OUTPUT = "Produzione svantaggiosa: indice di guadagno inferiore alla metà della media abituale".

Caso B. OUTPUT = "Produzione normale: indice di guadagno accettabile".

Caso C. OUTPUT = "Produzione vantaggiosa: indice di guadagno superiore alla media".

Caso D. OUTPUT = "Produzione ideale: indice di guadagno vicino al doppio della media abituale".

TRADUZIONE IN MODELLO LOGICO

In questa fase si indicano tutte le scelte fatte circa il passaggio da *Modello Concettuale* a **Modello Logico**. Di ogni traduzione, valuteremo inoltre la relazione fornita in “output”, valutandone dipendenze funzionali ed eventuali forme normali.

NB: Qualora le entità non abbiano differenze dalla loro presentazione iniziale (inizio della documentazione) non verrà spiegato nulla.

- 1) **Prodotto** (CodProdotto, Nome, NumeroFacce, DataCommercio)

FD = { $\text{CodProdotto} \rightarrow \text{Nome}, \text{NumeroFacce}, \text{DataCommercio}$ }

Forma Normale: La relazione è in BCNF

- 2) **Variante** (CodVariante, Descrizione)

FD = { $\text{CodVariante} \rightarrow \text{Descrizione}$ }

Forma Normale: La relazione è in BCNF

- 3) **Varianza** (Prodotto, Variante, Prezzo)

Gli identificatori esterni diventano attributi della tabella (si accorpano le associazioni *DescrizioneVariante* e *VariantiAssumibili*). Sono presenti *vincoli d'integrità referenziale* con *CodProdotto* in *Prodotto* e *CodVariante* in *Variante*.

FD = { $\text{Prodotto}, \text{Variante} \rightarrow \text{Prezzo}$ }

Forma Normale: La relazione è in BCNF

- 4) **Parte** (CodParte, Nome, Prezzo, Peso)

FD = { $\text{CodParte} \rightarrow \text{Nome}, \text{Prezzo}, \text{Peso}$ }

Forma Normale: La relazione è in BCNF

- 5) **Materiale** (Nome, Valore, CoeffSvalutazione)

FD = { $\text{Nome} \rightarrow \text{Valore}, \text{CoeffSvalutazione}$ }

Forma Normale: La relazione è in BCNF

- 6) **Composizione** (Prodotto, Parte, Pezzi)

Inizialmente relazione molti a molti, si acquisiscono come attributi gli identificatori di prodotto e parte (deve sussistere un *vincolo di integrità referenziale*)

FD = { $\text{Prodotto}, \text{Parte} \rightarrow \text{Pezzi}$ }

Forma Normale: La relazione è in BCNF

- 7) **Struttura** (Parte, Materiale, Quantita)

Inizialmente relazione molti a molti, si acquisiscono come attributi gli identificatori di materiale e parte (deve sussistere un *vincolo di integrità referenziale*)

$FD = \{ \text{Parte, Materiale} \rightarrow \text{Quantita} \}$

Forma Normale: La relazione è in BCNF

8) **Operazione** (ID, Livello, Faccia, Tipo, ClasseCampione, Giunzione, Prodotto, Variante)

Nell'entità operazione collassano gli identificatori di Classe Campione, Giunzione e Varianza, per via delle tre associazioni a cardinalità massima 1 che lega operazione alle tre entità. Per Classe, Prodotto e Varianza sussiste un *vincolo di integrità referenziale*, mentre per Giunzione si introduce la possibilità di valore NULL (nel caso di saldature)

$FD = \{ \text{ID, Sequenza} \rightarrow \text{Livello, Faccia, Tipo, ClasseCampione, Giunzione, Prodotto, Variante} \}$

Forma Normale: La relazione è in BCNF

9) **ClasseCampione** (CodSet, Descrizione)

$FD = \{ \text{CodSet} \rightarrow \text{Descrizione} \}$

Forma Normale: La relazione è in BCNF

10) **Utensile** (Nome, Tipologia)

$FD = \{ \}$ (solo banali)

Forma Normale: La relazione è in BCNF

11) **Giunzione** (CodGiunzione, Tipo)

$FD = \{ \text{CodGiunzione} \rightarrow \text{Tipo} \}$

Forma Normale: La relazione è in BCNF

12) **Usa** (Operazione, Utensile, Tipologia)

Inizialmente relazione molti a molti, esiste ora un *vincolo di integrità referenziale* su Operazione e su Utensile-Tipologia.

$FD = \{ \}$ (solo banali)

Forma Normale: La relazione è in BCNF

13) **Azione** (Base, Applicata, Operazione, Ripetizioni)

Esiste il vincolo di integrità referenziale (Base e Applicata con Parte, Operazione con i rispettivi identificatori di operazione)

$FD = \{ \text{Base, Applicata, Operazione} \rightarrow \text{Ripetizioni} \}$

Forma Normale: La relazione è in BCNF

14) **Linea** (CodLinea, Tempo, Tipo, Sequenza, Prodotto, Varianza)

Si ereditano gli attributi Sequenza, Prodotto e Varianza mediante l'associazione *Scelta* (che viene tradotta per via della sua cardinalità 1, 1).

FD = { $\text{CodLinea} \rightarrow \text{Tempo}, \text{Tipo}, \text{Sequenza}, \text{Prodotto}, \text{Varianza}$ }

Forma Normale: La relazione è in BCNF

15) **Magazzino** (CodMagazzino, Sede)

FD = { $\text{CodMagazzino} \rightarrow \text{Sede}$ }

Forma Normale: La relazione è in BCNF

16) **LottoProduzione** (CodLotto, DataProduzione, DurataPreventivata, DurataEffettiva, UnitaPreviste, UnitaEffettive, PrimaProduzione, Magazzino, Area, Linea)

Magazzino, Area e Linea sono derivati dalla traduzione delle due associazioni a cardinalità (1,1), e sussistono i due vincoli d'integrità referenziale con le chiavi di AreaMagazzino e Linea.

FD = { $\text{CodLotto} \rightarrow \text{DataProduzione}, \text{DurataPreventivata}, \text{DurataEffettiva}, \text{UnitaPreviste}, \text{UnitaEffettive}, \text{PrimaProduzione}, \text{Magazzino}, \text{Area}, \text{Linea}$ }

Forma Normale: La relazione è in BCNF

17) **LottoSmaltimento** (CodLotto, DataInizioSmaltimento, Magazzino, Area, Linea)

Magazzino, Area e Linea sono derivati dalla traduzione delle due associazioni a cardinalità (1,1), e sussistono i due vincoli d'integrità referenziale con le chiavi di AreaMagazzino e Linea.

FD = { $\text{CodLotto} \rightarrow \text{DataInizioSmaltimento}, \text{Magazzino}, \text{Area}, \text{Linea}$ }

Forma Normale: La relazione è in BCNF

18) **StazioneProduzione** (CodStazione, Orientazione, Classe, TempoPrevisto, Linea)

Classe e Linea rispettano un vincolo d'integrità referenziale rispettivamente con CodClasse (classe campione) e CodLinea (linea).

FD = { $\text{CodStazione} \rightarrow \text{Orientazione}, \text{Classe}, \text{TempoPrevisto}, \text{Linea}$ }

Forma Normale: La relazione è in BCNF

19) **StazioneSmaltimento** (CodStazione, Livello, ParteTarget, Linea)

Linea risponde alle stesse regole di cui sopra.

FD = { $\text{CodStazione} \rightarrow \text{Livello}, \text{ParteTarget}, \text{Linea}$ }

Forma Normale: La relazione è in BCNF

20) **EsecuzioneProduzione** (StazioneProduzione, Operazione)

Traduzione dell'omonima relazione molti a molti, vale l'integrità referenziale con gli identificatori della stazione produzione e di operazione.

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

21) **EsecuzioneSmaltimento** (StazioneSmaltimento, Operazione)

Traduzione dell'omonima relazione molti a molti, vale l'integrità referenziale con gli identificatori della stazione smaltimento e di operazione.

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

22) **Valutazione** (ClasseCampione, Operatore, TempoImpiegato)

Traduzione dell'omonima molti a molti, vale l'integrità referenziale con l'identificatore di Operatore (CodFiscale) e con l'identificatore di ClasseCampione (CodSet).

FD = { ClasseCampione, Operatore \rightarrow TempoImpiegato }

Forma Normale: La relazione è in BCNF

23) **UnitaDisponibili** (UID, Prodotto, Variante, LottoProduzione)

Prodotto, *Variante* e *LottoProduzione* fuoriescono dalla traduzione delle associazioni *Tipo* e *Provenienza Lotto* che essendo entrambe relazioni con cardinalità (1,1) dal lato di *UnitaDisponibili* vi collassano.

FD = { UID \rightarrow Prodotto, Variante, LottoProduzione }

Forma Normale: La relazione è in BCNF

24) **Utente** (CodFiscale, Nome, Cognome, Città, Provincia, Indirizzo, NumeroTelefono)

FD = { CodFiscale \rightarrow Nome, Cognome, Città, Provincia, Indirizzo, NumeroTelefono }

Forma Normale: La relazione è in BCNF

25) **Documento** (Numero, Tipologia, Scadenza, Ente, Utente)

Utente è acquisito dalla traduzione di Riconoscimento, e sussiste un vincolo di integrità referenziale con CodFiscale in *Utente*.

FD = { Numero \rightarrow Tipologia, Scadenza, Ente, Utente }

Forma Normale: La relazione è in BCNF

26) **Account** (NomeUtente, Email, Password, DomandaSicurezza, Risposta, Credito, Utente, IndirizzoConsegna)

Utente è acquisito dalla traduzione di identità, e sussiste un vincolo di integrità referenziale con CodFiscale in *Utente*.

FD = { NomeUtente \rightarrow Email, Password, DomandaSicurezza, Risposta, IndirizzoConsegna, Credito, Utente | Email \rightarrow NomeUtente }

Forma Normale: La relazione è in 3NF

27) **OrdineVendita** (CodOrdine, Stato, DataOrdine, TotDaPagare, Account)

Account è acquisito dalla traduzione dell'associazione *Acquisto*. Vale l'integrità referenziale con NomeUtente.

$FD = \{ \text{CodOrdine} \rightarrow \text{Stato}, \text{DataOrdine}, \text{TotDaPagare}, \text{Account} \}$

Forma Normale: La relazione è in BCNF

- 28) **Spedizione** (Codice, DataPrevista, HubAttuale, Stato, AccountConsegna, OrdineVendita)
AccountConsegna e *OrdineVendita* derivano rispettivamente dalle associazioni *consegna* e *evaso*, e sussiste vincolo d'integrità referenziale rispettivamente con NomeUtente e CodOrdine.

$FD = \{ \text{Codice} \rightarrow \text{DataPrevista}, \text{HubAttuale}, \text{Stato}, \text{AccountConsegna}, \text{OrdineVendita} \}$

Forma Normale: La relazione è in BCNF

- 29) **Garanzia** (CodGaranzia, Descrizione, Durata, ClasseGuasti)
ClasseGuasti deriva dalla associazione *ClasseCoperta* e quindi sussiste un vincolo d'integrità referenziale con la chiave di *ClasseGuasto*.

$FD = \{ \}$ (solo banali)

Forma Normale: La relazione è in BCNF

- 30) **Motivazione** (CodMotivazione, Nome, Descrizione)

$FD = \{ \}$ (solo banali)

Forma Normale: La relazione è in BCNF

- 31) **Reso** (Codice, DataRichiesta, Approvato, DataApprovazione, UnitaVendute)
UnitaVendute deriva dalla traduzione dell'associazione *coinvolgimento* e sussiste un vincolo di integrità con UID.

$FD = \{ \text{Codice} \rightarrow \text{DataRichiesta}, \text{Approvato}, \text{DataApprovazione}, \text{UnitaVendute} \}$

Forma Normale: La relazione è in BCNF

- 32) **RichiestaReso** (Reso, Motivazione, Account, Commento)
Traduzione dell'omonima molti a molti, sussistono i vincoli di integrità referenziale con gli identificatori di Reso, Motivazione e Account.

$FD = \{ \text{Reso}, \text{Motivazione}, \text{Account} \rightarrow \text{Commento} \}$

Forma Normale: La relazione è in BCNF

- 33) **UnitaVendute** (UID, DataVendita, OrdineVendita, Prodotto, Variante, LottoProduzione)
Prodotto e *Variante* derivano dalla traduzione dell'associazione *Tipo*. *LottoProduzione* deriva dalla traduzione dell'associazione *ProvenienzaLotto*.

$FD = \{ \text{UID} \rightarrow \text{DataVendita}, \text{OrdineVendita}, \text{Prodotto}, \text{Variante}, \text{LottoProduzione} \}$

Forma Normale: La relazione è in BCNF

34) **UnitaRese** (UID, Reso, Magazzino, Area)

Reso, Magazzino e Area derivano dalla traduzione delle associazioni *ContenutaIn* e *Restituzione* e sussiste il vincolo d'integrità referenziale con gli identificatori delle entità Reso e AreaMagazzino.

$FD = \{ UID \rightarrow Reso, Magazzino, Area \}$

Forma Normale: La relazione è in BCNF

35) **Giudizio** (Account, Prodotto, Valutazione, Commento)

Traduzione dell'omonima molti a molti, sussiste il vincolo d'integrità referenziale con gli identificatori di Account e Prodotto.

$FD = \{ Account, Prodotto \rightarrow Valutazione, Commento \}$

Forma Normale: La relazione è in BCNF

36) **Carrello** (Prodotto, Variante, OrdineVendita, Garanzia, Quantita, Stato, Categoria)

Traduzione dell'omonima molti a molti, sussiste il vincolo di integrità referenziale con gli identificatori di Varianza, Ordine e Garanzia.

$FD = \{ Prodotto, Variante, OrdineVendita, Garanzia \rightarrow Quantita, Stato, Categoria \}$

Forma Normale: La relazione è in BCNF

37) **Copertura** (Prodotto, Garanzia, Costo)

Traduzione dell'omonima molti a molti, sussiste il vincolo di integrità referenziale con gli identificatori delle entità Prodotto e Garanzia

$FD = \{ Prodotto, Garanzia \rightarrow Costo \}$

Forma Normale: La relazione è in BCNF

38) **Guasto** (Codice, Nome)

$FD = \{ Codice \rightarrow Nome \}$

Forma Normale: La relazione è in BCNF

39) **AssistenzaVirtualeNOErrore** (CodiceAssistenza, DataRichiesta, Riuscita, Sintomo, UnitaVendute)

Sintomo e *UnitaVenduta* derivano dalla traduzione delle associazioni *SintomoIniziale* e *RiferitoA*, sussistono vincoli di integrità referenziale con le entità Sintomo e Unitavendute.

$FD = \{ CodAssistenza \rightarrow DataRichiesta, Riuscita, Utente, UnitaVendute \}$

Forma Normale: La relazione è in BCNF

40) **AssistenzaVirtualeCONErrore** (CodiceAssistenza, DataRichiesta, CodErrore, Guasto, Rimedio, UnitaVendute)

Guasto, Rimedio e UnitaVendute derivano dalla traduzione delle associazioni *IdentificazioneGuasto*, *ProceduraImmediata* e *RiferitoA*, sussistono vincoli di integrità referenziale con le entità Rimedio, Guasto e Unitavendute.

$FD = \{ CodAssistenza \rightarrow DataRichiesta, CodErrore, Guasto, Rimedio, UnitaVendute \}$

Forma Normale: La relazione è in BCNF

- 41) **AssistenzaFisica** (CodiceAssistenza, DataRichiesta, TecniciRichiesti, UnitàVendute, Sintomo)

Sintomo e *UnitàVendute* derivano dalla traduzione delle associazioni *SintomiAccusati* e *RiferitoA*, sussistono vincoli di integrità referenziale con le entità Sintomo e UnitàVendute.

$FD = \{ \text{CodAssistenza} \rightarrow \text{DataRichiesta}, \text{DataConcordata}, \text{Utente}, \text{UnitàVendute} \}$

Forma Normale: La relazione è in BCNF

- 42) **Rimedio** (CodiceRimedio, Descrizione)

$FD = \{ \text{CodRimedio} \rightarrow \text{Descrizione} \}$

Forma Normale: La relazione è in BCNF

- 43) **Sintomo** (CodSintomo, Descrizione)

$FD = \{ \text{CodSintomo} \rightarrow \text{Descrizione} \}$

Forma Normale: La relazione è in BCNF

- 44) **Conoscenza** (AIM, DataRisoluzione, Guasto)

Guasto deriva dalla traduzione dell'associazione *GuastoDiagnosticato* (sussiste vincolo d'integrità referenziale con CodGuasto).

$FD = \{ \text{AIM} \rightarrow \text{DataRisoluzione}, \text{Guasto} \}$

Forma Normale: La relazione è in BCNF

- 45) **InterventoFisico** (Ticket, Domicilio, Stato, OreLavoro, AssistenzaFisica, Preventivo)

Gli attributi AssistenzaFisica e Preventivo derivano dalle due associazioni con cardinalità (1,1) dal lato di intervento fisico con le omonime entità. Sussistono quindi due vincoli di integrità referenziale.

$FD = \{ \text{Ticket} \rightarrow \text{Domicilio}, \text{Stato}, \text{OreLavoro}, \text{AssistenzaFisica}, \text{Preventivo} \}$

Forma Normale: La relazione è in BCNF

- 46) **Preventivo** (Codice, DataRilascio, Tecnico)

$FD = \{ \text{Codice} \rightarrow \text{DataRilascio}, \text{Tecnico} \}$

Forma Normale: La relazione è in BCNF

- 47) **Fattura** (Codice, DataRilascio, TotaleNetto, ModalitàPagamento, InGaranzia, InterventoFisico)

InterventoFisico deriva dall'associazione *Evasa* (esiste vincolo d'integrità referenziale con Ticket).

FD = { Codice \rightarrow DataRilascio, TotaleNetto, Firma, Pagamento, InGaranzia, InterventoFisico }

Forma Normale: La relazione è in BCNF

- 48) **OrdineParti** (Codice, DataRichiesta, DataPrevistaConsegna, DataConsegna, InterventoFisico)

InterventoFisico deriva dalla traduzione dell'associazione *RichiestaParti*, sussiste quindi un vincolo di integrità referenziale.

FD = { Codice \rightarrow DataRichiesta, DataPrevistaConsegna, DataConsegna, InterventoFisico }

Forma Normale: La relazione è in BCNF

- 49) **Domanda** (CodiceDomanda, Testo)

FD = { CodiceDomande \rightarrow Testo }

Forma Normale: La relazione è in BCNF

- 50) **Relativo** (CodGuasto, CodProdotto)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Guasto e Prodotto

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

- 51) **AutoDiagnosi** (AssistenzaVirtualeNOErrore, Rimedio, Domanda, Risposta)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di AssistenzaVirtualeNoErrore, Rimedio e Domanda.

FD = { AssistenzaVirtualeNOErrore, Rimedio, Domando \rightarrow Risposta }

Forma Normale: La relazione è in BCNF

- 52) **SintomiAccusati** (Sintomo, AssistenzaFisica)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Sintomo e AssistenzaFisica

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

- 53) **SintomiMemorizzati** (Sintomo, Conoscenza)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Sintomo e Conoscenza

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

- 54) **RimediUtilizzati** (Conoscenza, Rimedio)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Conoscenza e Rimedio

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

55) **ListaGuasti** (Preventivo, Guasto, Prezzo)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Preventivo e Guasto

FD = { Preventivo, Guasto → Prezzo }

Forma Normale: La relazione è in BCNF

56) **Incarico** (Tecnico, InterventoFisico)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Tecnico e InterventoFisico

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

57) **Ricambio** (OrdineParti, Parte)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di OrdineParti e Parte

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

58) **Test** (Codice, Nome, Livello, TestPadre, Importanza, Prodotto)

Prodotto deriva dalla traduzione dell'associazione *Albero* e sussiste il vincolo d'integrità referenziale con l'identificatore di Prodotto

FD = { Codice → Nome, Livello, TestPadre, Importanza, Prodotto }

Forma Normale: La relazione è in BCNF

59) **UnitaEnd-Of-Life** (UID, GradoUsura, Prodotto, Variante, LottoSmaltimento)

Prodotto e *Variante* derivano dalla traduzione dell'associazione *Tipo*. *LottoSmaltimento* deriva dalla traduzione dell'associazione *ProvenienzaLotto*. Sussistono quindi vincoli di integrità referenziale con le chiavi delle entità *Varianza* ed *LottoSmaltimento*.

FD = { UID → GradoUsura, Prodotto, Variante, LottoSmaltimento }

Forma Normale: La relazione è in BCNF

60) **UnitaRicondizionate** (UID, Grado, Ricondizionamento, Prodotto, Variante)

Prodotto e *Variante* derivano dalla traduzione dell'associazione *Tipo*. Esiste il vincolo di integrità referenziale con la chiave di *Varianza*.

FD = { UID → Grado, Ricondizionamento, Prodotto, Variante }

Forma Normale: La relazione è in BCNF

61) **ControlloGenerale** (UnitaRicondizionaete, Test, Superato)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di UnitaRicondizionate e Test.

FD = { UnitaRese, Test \rightarrow Superato }

Forma Normale: La relazione è in BCNF

62) **ControlloTest** (Test, Parte)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Test e Parte

FD = { } (solo banali)

Forma Normale: La relazione è in BCNF

63) **Ricondizionamento** (Parte, UnitaRicondizionate, Quantita)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di Parte e UnitaRicondizionate.

FD = { Parte, UnitaRicondizionate \rightarrow Quantita }

Forma Normale: La relazione è in BCNF

64) **RecuperoMateriale** (UnitaEnd-Of-Life, Materiale, StazioneSmaltimento, Quantita)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di UnitaEOL, Materiale e StazioneSmaltimento

FD = { UnitaEnd-Of-Life, Materiale, StazioneSmaltimento \rightarrow Quantita }

Forma Normale: La relazione è in BCNF

65) **RecuperoParte** (UnitaEnd-Of-Life, Parte, StazioneSmaltimento, Quantita)

Traduzione dell'omonima relazione molti a molti, sussistono vincoli d'integrità referenziale con gli identificatori di UnitaEnd-Of-Life, Parte e StazioneSmaltimento

FD = { UnitaEnd-Of-Life, Parte, StazioneSmaltimento \rightarrow Quantita }

Forma Normale: La relazione è in BCNF

66) **AreaMagazzino** (Area, CodMagazzino, Capienza, CapienzaDisponibile)

L'attributo CodMagazzino deriva dalla chiave esterna con Magazzino e quindi seguirà un *vincolo di integrità referenziale* con Magazzino.

FD = { Area, CodMagazzino \rightarrow Capienza, CapienzaDisponibile }

Forma Normale: La relazione è in BCNF

67) **CategoriaProdotto** (Categoria, Nome)

FD = { Categoria \rightarrow Nome }

Forma Normale: La relazione è in BCNF

68) **CentroAssistenza** (CodCentro, Indirizzo, Citta, Provincia)

$FD = \{ _CodCentro \rightarrow \text{Indirizzo, Citta, Provincia} \}$

Forma Normale: La relazione è in BCNF

69) **ClasseGuasto** (Nome, Descrizione)

$FD = \{ _Nome \rightarrow \text{Descrizione} \}$

Forma Normale: La relazione è in BCNF

70) **DatiAnagraficiOperatore** (CodFiscale, Nome, Cognome, Citta, DataNascita)

L'attributo CodFiscale deriva dalla chiave esterna con Operatore e quindi seguirà un *vincolo di integrità referenziale* con Operatore.

$FD = \{ _CodFiscale \rightarrow \text{Nome, Cognome, Citta, DataNascita} \}$

Forma Normale: La relazione è in BCNF

71) **DatiAnagraficiTecnico** (CodFiscale, Nome, Cognome, Citta, DataNascita)

L'attributo CodFiscale deriva dalla chiave esterna con Tecnico e quindi seguirà un *vincolo di integrità referenziale* con Tecnico.

$FD = \{ _CodFiscale \rightarrow \text{Nome, Cognome, Citta, DataNascita} \}$

Forma Normale: La relazione è in BCNF

72) **DatiSequenza** (CodiceSequenza, CodProdotto, CodVariante, ID, NumOperazione)

Gli attributi CodProdotto e CodVariante derivano dalla chiave esterna con Sequenza e quindi seguirà un *vincolo di integrità referenziale* con Sequenza.

$FD = \{ _CodiceSequenza, _CodProdotto, _CodVariante, ID \rightarrow \text{NumOperazione} \}$

Forma Normale: La relazione è in BCNF

73) **InfoVarianza** (CodProdotto, CodVariante, Ingombro, Peso, NumMinUnitaRicondizionamento, NumMinUnitaSmaltimento, PercentualeMAXRicondizionamento, NumMinUnitaLotto)

Gli attributi CodProdotto e CodVariante derivano dalla chiave esterna con Varianza e quindi seguirà un *vincolo di integrità referenziale* con Varianza.

$FD = \{ _CodProdotto, _CodVariante \rightarrow \text{Ingombro, Peso, NumMinUnitaRicondizionamento, NumMinUnitaSmaltimento, PercentualeMAXRicondizionamento, NumMinUnitaLotto} \}$

Forma Normale: La relazione è in BCNF

74) **Operatore** (CodFiscale, Stipendio)

$FD = \{ _CodFiscale \rightarrow \text{Stipendio} \}$

Forma Normale: La relazione è in BCNF

75) **Sequenza** (CodProdotto, CodVariante, Codice)

Gli attributi CodProdotto e CodVariante derivano dalla chiave esterna con Varianza e quindi seguirà un *vincolo di integrità referenziale* con Varianza.

FD = { CodProdotto, CodVariante → Codice }

Forma Normale: La relazione è in BCNF

76) **Tecnico** (CodFiscale, Occupato, Stipendio, FasciaOraria)

FD = { CodFiscale → Occupato, Stipendio, FasciaOraria }

Forma Normale: La relazione è in BCNF

77) **UnitaPerseProduzione** (Lotto, UltimaOperazione, StazioneProduzione, Quante)

Gli attributi ID e CodLotto derivano dalle chiavi esterna con Operazione e LottoProduzione, quindi seguiranno due vincoli di integrità referenziale.

StazioneProduzione deriva dall'associazione PerditaProduzione che essendo una (1,1) dal lato di UnitaPerseProduzione vi crollerà e quindi sussisterà un vincolo di integrità referenziale con la chiave di StazioneProduzione.

FD = { Lotto, UltimaOperazione, CodProdotto, CodVariante, Quante }

Forma Normale: La relazione è in BCNF

78) **UnitaPerseSmaltimento** (Lotto, UltimaOperazione, StazioneSmaltimento, Quante)

Gli attributi ID e CodLotto derivano dalle chiavi esterna con Operazione e LottoSmaltimento, quindi seguiranno due vincoli di integrità referenziale.

StazioneSmaltimento deriva dall'associazione PerditaSmaltimento che essendo una (1,1) dal lato di UnitaPerseSmaltimento vi crollerà e quindi sussisterà un vincolo di integrità referenziale con la chiave di StazioneSmaltimento.

FD = { Lotto, UltimaOperazione, CodProdotto, CodVariante, Quante }

Forma Normale: La relazione è in BCNF

IMPLEMENTAZIONE SQL

In questo paragrafo verranno descritti i codici relativi all'implementazione del DB.

Saranno esaminati con particolare attenzione i trigger che permettono di rispettare la coerenza della base di dati, le procedure necessarie allo svolgimento di particolari operazioni, ecc.

Trigger generici

Di seguito una lista dei trigger generici che si vogliono implementare per dare un'idea del concetto di azienda che si è voluto costruire.

- ❖ Trigger su *Controllo Generale* che controlla a che soglia (inteso come somma delle importanze dei test falliti) siamo arrivati al momento e, nel caso, modifica lo stato dell'unità da ricondizionata a unità da smaltire (end of life).

- Anzitutto, dichiariamo le variabili che ci serviranno. Poiché il trigger è su *controllo generale*, si deve ottenere il codice del prodotto (la soglia massima è in quella relazione) che possiamo ottenere dall'archivio delle vendite (scegliamo di prendere anche la variante, che ci servirà in seguito)
- Dopodiché, prendiamo la soglia massima oltre la quale deve essere smaltita l'unità imponendo, nella tabella *Prodotto*, l'uguaglianza della chiave con il valore da noi appena ricavato.
- Calcoliamo la *somma attuale* sommando le importanze dei test finora effettuati dall'unità in considerazione (*NEW.UnitaResa*), ma si farà caso solo ai test falliti (*superato = FALSE*).
- Infine, se la *somma attuale* è superiore o uguale alla soglia massima, inseriamo i dati dell'unità (incluso prodotto e variante) nella tabella *UnitaEOL*, ed eliminiamo la stessa unità dalla tabella *UnitaRicondizionata*.

```
8 DELIMITER $$
9 • DROP TRIGGER IF EXISTS Controllo_Soglia_Smaltimento;
10 CREATE TRIGGER Controllo_Soglia_Smaltimento
11 AFTER INSERT ON ControlloGenerale FOR EACH ROW
12 BEGIN
13
14 DECLARE Prodotto VARCHAR(20) DEFAULT '';
15 DECLARE Variante VARCHAR(100) DEFAULT '';
16 DECLARE Soglia INTEGER DEFAULT 0;
17 DECLARE SommaAttuale INTEGER DEFAULT 0;
18
19 SELECT UV.Prodotto, UV.Variante INTO Prodotto, Variante
20 FROM UnitaVenduta UV
21 WHERE UV.UID = NEW.UnitaRicondizionata;
22 SET Soglia =
23 (
24 SELECT IV.SogliaSmaltimento
25 FROM InfoVarianza IV
26 WHERE IV.Prodotto = Prodotto AND IV.Variante = Variante
27 );
28
29 SET SommaAttuale =
30 (
31 SELECT SUM(Importanza)
32 FROM ControlloGenerale INNER JOIN Test ON Test = CodTest
33 WHERE UnitaRicondizionata = NEW.UnitaRicondizionata AND Superato = FALSE
34 );
35
36 IF(SommaAttuale >= Soglia) THEN
37 SELECT 'Somma importanza dei test troppo alta! Unita da smaltire!';
38
39 INSERT INTO UnitaEndOfLife
40 VALUES (NEW.UnitaResa, SommaAttuale, NULL, Prodotto, Variante);
41
42 DELETE FROM UnitaRicondizionata
43 WHERE UID = NEW.UnitaRicondizionata;
44
45 END IF;
46
47 END $$
48 DELIMITER ;
```

Si faccia attenzione al fatto che, poiché il trigger è di tipo AFTER, il test che "fa superare la soglia" è comunque eseguito, e non impedito.

- ❖ Si vuole implementare un trigger per la gestione delle unità disponibili. In particolare, si vuole verificare che esista un ordine, che richiede un'unità della stessa tipologia di prodotto e variante dell'unità disponibile appena inserita, che sia ancora non evadibile (dunque il relativo record nella tabella Carrello abbia il valore 0 nel campo 'Stato'). Nel caso in cui esistano più ordini che rispettano i requisiti allora si prende quello più vecchio in ordine temporale. Una volta individuato l'ordine si cambia lo stato del record della tabella Carrello che si riferisce a quest'ultimo mettendo il valore 1 nel campo 'Stato'. Inoltre, si controlla che l'unità disponibile appena inserita sia l'ultima necessaria a far cambiare lo stato complessivo dell'ordine da 'Pendente' ad 'In Processazione'. Ciò significa che nessun record della tabella Carrello, relativo all'ordine che abbiamo preso in considerazione all'inizio, abbia il valore 0 nel campo 'Stato'. Se succede questo allora l'ordine può essere spedito e passa alla fase 'In Processazione'.

```

DELIMITER $$
CREATE TRIGGER ordine_pendente_piuVecchio_soddisfattibile
AFTER INSERT ON unitadisponibili
FOR EACH ROW
BEGIN
    DECLARE codiceOrdine INT;
    DECLARE accountUtente VARCHAR(30) DEFAULT NULL;
    DECLARE garanzia VARCHAR(45) DEFAULT NULL;
    DECLARE unitaNONdisponibili INT;

    DECLARE cursore CURSOR FOR
    SELECT OV.CodOrdine, OV.Account as Utente, C.Garanzia
    FROM ordinevendita OV INNER JOIN carrello C on OV.CodOrdine = C.OrdineVendita
    WHERE C.Prodotto = NEW.Prodotto AND C.Variante = NEW.Variante AND
    C.Stato = 0 and OV.DataOrdine =
    (select MIN(OV1.DataOrdine)
    FROM ordinevendita OV1 INNER JOIN carrello C1 on OV1.CodOrdine = C1.OrdineVendita
    WHERE C1.Prodotto = NEW.Prodotto AND C1.Variante = NEW.Variante AND
    C1.Stato = 0) /* Carrello.Stato = 0 -> Unità non disponibile*/
    LIMIT 1;

    OPEN cursore;
    FETCH cursore INTO codiceOrdine, accountUtente, garanzia;
    UPDATE carrello C SET C.Stato = 1
    WHERE C.Prodotto = NEW.Prodotto AND
    C.Variante = NEW.Variante AND
    C.OrdineVendita = codiceOrdine AND
    C.Garanzia = garanzia;
    CLOSE cursore;

    SELECT COUNT(*) INTO unitaNONdisponibili
    FROM carrello
    WHERE OrdineVendita = codiceOrdine AND
    Stato = 0;

    IF unitaNONdisponibili = 0 THEN
        UPDATE ordinevendita SET Stato = 'In Processazione'
        WHERE CodOrdine = codiceOrdine;
    END IF;

END $$
DELIMITER ;

```

- ❖ Trigger che, quando una spedizione viene modificata (in termini di nuovo Hub), controlla che la spedizione sia arrivata nella stessa provincia dell'utente a cui sta consegnando. Se la risposta è positiva, il trigger deve modificare lo stato della spedizione in "In Consegna"

- Trigger che richiede il recupero della provincia dell'utente a cui si sta consegnando. Per recuperarla si recupera da utente l'account ad egli associato, e dall'account le spedizioni ad egli collegate
- Tra tutte le spedizioni, si sceglie quella che si sta modificando, e si ricava la provincia dell'utente
- Infine, se NEW.HubAttuale (che è sempre il nome di una provincia) diventa uguale alla provincia appena ricavata, lo stato della spedizione diventa "In Consegna", altrimenti il trigger non fa nulla.

```

6 CREATE TRIGGER Verifica_Spedizione BEFORE UPDATE ON Spedizione
7 FOR EACH ROW
8 BEGIN
9
10 DECLARE Provincia VARCHAR(50) DEFAULT '';
11
12 SET Provincia =
13 (
14 SELECT U.Provincia
15 FROM Utente U INNER JOIN Account A ON A.Utente = U.CodFiscale
16 INNER JOIN Spedizione S ON S.AccountConsegna = A.NomeUtente
17 WHERE S.CodConsegna = NEW.CodConsegna
18 );
19
20 IF(NEW.HubAttuale = Provincia) THEN
21 SET NEW.Stato = 'InConsegna'
22 END IF;
23 END $$
24 DELIMITER ;

```

- ❖ Quando un ordine è passato in processazione, inserire il totale da pagare (è pronto per essere spedito, le unità ci sono tutte)

- È importante ricordare che il totale da pagare è calcolato nel seguente modo:
- Se il prodotto è di categoria 0, allora si sta comprando un'unità nuova e, dunque, il prezzo è dato semplicemente dal prodotto "prezzo" * "quantità".
- Se il prodotto è di altra categoria, allora deve applicarsi uno sconto pari a "Categoria * 5%".

```

1 DROP TRIGGER IF EXISTS AggiornaDaPagare;
2 DELIMITER $$
3 CREATE TRIGGER AggiornaDaPagare BEFORE UPDATE ON Ordine
4 FOR EACH ROW
5 BEGIN
6 DECLARE daPagare INTEGER DEFAULT 0;
7
8 SET daPagare = (
9 SELECT SUM(IF(C.Categoria <> 0, (IV.Prezzo - IV.Prezzo * (5 * C.Categoria / 100)) * C.Quantita, IV.Prezzo * C.Quantita)) AS TotDaPagare
10 FROM Carrello C INNER JOIN Varianza IV ON IV.CodProdotto = C.Prodotto AND IV.CodVariante = C.Variante
11 WHERE C.OrdineVendita = NEW.CodOrdine
12 );
13
14 IF(NEW.Stato = 'InProcessazione') THEN
15 SET NEW.TotDaPagare = daPagare;
16 END IF;
17
18 END $$
19 DELIMITER ;

```

- ❖ Trigger che, al momento dell'accettazione di un reso, si occupa di aggiungere l'unità in questione alle unità rese calcolando anche il magazzino in cui questa dovrà essere immagazzinata

- Non appena il trigger rileva l'accettazione del reso, si ricava il codice del prodotto dell'unità in questione
- Viene quindi controllato se altre unità rese di quel prodotto sono già in un magazzino, in tal caso il trigger assegnerà a questa unità la stessa area magazzino delle precedenti con lo scopo di facilitare la successiva fase di recupero
- Se il trigger non trova alcun magazzino vuol dire che non vi sono altre unità rese dello stesso prodotto in alcun magazzino.

Si procederà calcolando quale sia il magazzino migliore per accogliere lei e le successive unità. Per far questo il trigger cercherà un'area magazzino con la giusta predisposizione e con area disponibile massima.

- Una volta trovata la assegnerà all'unità e questa diventerà poi l'area in cui verranno inviate le unità dello stesso prodotto che verranno successivamente rese.

```
DELIMITER $$
DROP TRIGGER IF EXISTS accettazione_reso$$

CREATE TRIGGER accettazione_reso
AFTER UPDATE ON Reso
FOR EACH ROW
BEGIN
    declare MagazzinoTarget int default 0;
    declare AreaTarget int default 0;
    declare ProdottoTarget varchar(10) default '';

    IF new.Approvato = 1 THEN
        select UV.Prodotto into ProdottoTarget
        from UnitaVendute UV
        where UV.UID = new.UnitaVendute;

        select distinct IFNULL(UR.Magazzino, 0), IFNULL(UR.Area, 0) into MagazzinoTarget, AreaTarget
        from UnitaRese UR natural join UnitaVendute UV
        where UV.Prodotto = ProdottoTarget;

        IF MagazzinoTarget = 0 THEN
            WITH
            MagazziniTarget as (
                select Magazzino, Area, CapienzaDisponibile
                from AreaMagazzino
                where Predisposizione = (
                    select Categoria
                    from Prodotto
                    where CodProdotto = ProdottoTarget)
            )
            select Magazzino, Area into MagazzinoTarget, AreaTarget
            from MagazziniTarget
            where CapienzaDisponibile = (
                select MAX(M2.CapienzaDisponibile)
                from MagazziniTarget M2
            )

            limit 1;
        END IF;

        INSERT INTO UnitaRese
        SELECT UV.UID, new.Codice, MagazzinoTarget, AreaTarget
        FROM UnitaVendute UV
        WHERE UV.UID = new.UnitaVendute;
    END IF;
END $$
delimiter ;
```

- ❖ Trigger che, al momento dell'inserimento di una fattura nel database si occupa di calcolarne il totaleNetto

Quando il database rileva l'inserimento di una nuova fattura viene automaticamente calcolato il suo totaleNetto:

La somma viene ricavata moltiplicando il numero delle ore di lavoro per 10€ (tariffa del tecnico) e sommando a questa il costo totale delle parti sostituite.

```
create trigger calcola_TotaleNetto
before insert on fattura for each row
begin
    declare ore float default 0;
    declare costoPartiRicambio float default 0;

    select ifnull(OreLavoro,0) into ore
    from interventofisico
    where ticket= new.InterventoFisico;

    WITH
    partiTarget as (
        select R.Parte
        from ricambio R
        where R.ordineParti = (
            select O.Codice
            from ordineparti O
            where O.InterventoFisico = new.InterventoFisico
        )
    )
    select ifnull(sum(prezzo),0) into costoPartiRicambio
    from partiTarget PT inner join parte P on PT.parte=p.codParte;

    set new.totaleNetto = ore*10 + costoPartiRicambio;
END $$
delimiter ;
```

- Dopo che un'assistenza virtuale senza codice d'errore è stata modificata, viene controllato se questa è fallita, e se è quindi necessario un intervento fisico. Se sì, aggiungere un record in AssistenzaFisica relativo allo stesso utente e alla stessa unità

Questo trigger ha il semplice compito di aggiungere una nuova AssistenzaFisica qualora l'assistenza virtuale non abbia successo.

```
4 • DROP TRIGGER IF EXISTS Richiesta_Assistenza_Fisica;
5 DELIMITER $$
6 • CREATE TRIGGER Richiesta_Assistenza_Fisica AFTER UPDATE ON AssistenzaVirtualeNoCodErrore
7 FOR EACH ROW
8 BEGIN
9
10 IF(NEW.Riuscita = 0 AND OLD.Riuscita IS NULL) THEN
11     INSERT INTO AssistenzaFisica
12     VALUES (NEW.CodAssistenza, CURRENT_DATE, 1, NEW.UnitaVenduta);
13 END IF;
14 END $$
15
16 DELIMITER ;
```


Business rule

Solitamente le business rule si presentano come trigger di tipo BEFORE INSERT in quanto devono evitare che si manifesti una realtà aziendale incoerente con quella pensata. Di seguito riportiamo le business rule che si è deciso di implementare:

- ❖ Quando si inserisce un test in controllo generale, si controlli che il test padre sia stato effettuato. Se il test padre non è stato effettuato, restituire errore; se è stato effettuato con successo, impedire il nuovo test. Permetti l'inserimento se e solo se il test padre ha fallito.

- Anzitutto ricaviamo il test padre utilizzando il codice del test che si vuole inserire (chiave della relazione *Test*)
- Dopodiché si controlla che il test padre, su quell'unità, sia già stato effettuato
- Se e solo se il test è stato effettuato, si ci assicura che il test sia fallito
- Se il test è invece stato superato (esito = 1), il trigger deve impedire l'inserimento e restituire errore poiché un sotto-test si compie se e solo se il test padre relativo è fallito!

```
5 DELIMITER $$
6 • DROP TRIGGER IF EXISTS Valuta_Test_Padre;
7 CREATE TRIGGER Valuta_Test_Padre BEFORE INSERT ON ControlloGenerale
8 FOR EACH ROW
9 BEGIN
10
11 DECLARE controllo INTEGER DEFAULT 0;
12 DECLARE esito INTEGER DEFAULT 0;
13
14 SET @TestPadre =
15 (
16 SELECT TestPadre
17 FROM Test
18 WHERE CodTest = NEW.Test
19 );
20
21
22 SELECT COUNT(*), Esito INTO controllo, esito
23 FROM ControlloGenerale
24 WHERE UnitaRicondizionata = NEW.UnitaRicondizionata AND Test = @TestPadre;
25
26 IF(controllo = 0) THEN
27 SIGNAL SQLSTATE '45000'
28 SET MESSAGE_TEXT = 'Test Padre mai effettuato! Test non valido.';
29
30 IF(esito = 1) THEN
31 SIGNAL SQLSTATE '45000'
32 SET MESSAGE_TEXT = 'Test padre già riuscito, test privo di senso!';
33 END IF;
34
35 END $$
36 DELIMITER ;
```

- ❖ Se viene inserito un test il cui livello non sia pari al livello del test padre + 1, allora l'inserimento deve essere bloccato (ciò deve essere sempre rispettato a causa della struttura ad albero dei test).

- Trigger molto semplice, che recupera il livello del padre direttamente dalla tabella *Test*, andando a cercare il test che abbia come codice il valore del test padre del test che si sta inserendo.
- Se il livello del figlio non differisce da quello del padre di 1, (e solo 1), l'inserimento è immediatamente interrotto

```
DROP TRIGGER IF EXISTS test_okay;

DELIMITER $$

CREATE TRIGGER test_okay
BEFORE INSERT ON test
FOR EACH ROW
BEGIN

DECLARE livelloPADRE INT;
SELECT Livello INTO livelloPADRE
FROM Test
WHERE Codice = NEW.TestPadre;

IF (NEW.Livello - livelloPADRE) <> 1 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Test non conforme alle politiche di precedenza dei livelli padre-figlio';
END IF;

END $$

DELIMITER ;
```

- ❖ Si implementa un trigger che verifica, ad ogni inserzione del carrello, se l'unità è presente o meno nel magazzino. Se non c'è unità, quel record di carrello vede il suo stato settarsi a 0, e indica appunto che l'unità è momentaneamente non disponibile. L'ordine di riferimento si setta in "Pendente" qualora un solo record del carrello risulti con Stato = 0

- Anzitutto si contano le unità del prodotto e della variante richiesta all'interno della relazione *UnitaDisponibili*
- Se il conto restituisce 0 o un valore minore del numero di unità previste dal carrello allora il carrello deve modificare il suo stato in 0 (altrimenti resta quello di default, 1).
- Se è stata effettuata questa modifica, l'ordine di riferimento deve modificarsi in "pendente".

```
DROP TRIGGER IF EXISTS check_disponibilita_unita_carrello;

DELIMITER $$

CREATE TRIGGER check_disponibilita_unita_carrello
BEFORE INSERT ON Carrello
FOR EACH ROW
BEGIN

DECLARE flag INT DEFAULT 0;
SELECT COUNT(*) INTO flag
FROM unitadisponibili UD
WHERE UD.Prodotto = NEW.Prodotto AND
      UD.Variante = NEW.Variante;

IF flag < NEW.Quantita THEN
    SET NEW.Stato = 0;
    UPDATE ordinevendita
    SET Stato = 'Pendente'
    WHERE CodOrdine = NEW.OrdineVendita;
END IF;

END $$

DELIMITER ;
```


- ❖ Event che quotidianamente si occupa di controllare se le istanze di *Unità Rese* hanno raggiunto un numero consono affinché venga iniziata la fase di ricondizionamento.

Poiché il processo di ricondizionamento per unità di uno stesso prodotto viene avviato solo quando raggiungono un numero congruo prestabilito dall'azienda, questa ha predisposto un event che monitora quotidianamente questo aspetto.

- Vengono dapprima contate quante sono le unità di uno stesso prodotto tra le Unità Rese e se queste sono in numero maggiore o uguale a quello prefissato
- I gruppi di unità che soddisfano questo prerequisito verranno quindi inserite fra le unità prossime alla procedura di ricondizionamento e cancellate fra quelle rese poiché ormai il loro destino è quello di essere ricondizionate

```

1  delimiter $$
2  • DROP EVENT IF EXISTS monitora_resi$$
3
4  • CREATE EVENT monitora_resi
5  ON SCHEDULE EVERY 1 DAY starts '2020-08-15 01:00:00' do
6  BEGIN
7      create temporary table ProdottiTarget(
8          CodProdotto varchar(15) NOT NULL,
9          primary key(CodProdotto)
10     );
11
12     insert into ProdottiTarget
13     WITH
14     contoUnita as (
15         select UV.Prodotto as CodProdotto, count(*) as NumUnita
16         from UnitàRese UR natural join UnitàVendute UV
17         group by UV.Prodotto
18     )
19     select CodProdotto
20     from contoUnita CU natural join InfoVarianza IV
21     where CU.NumUnita >= IV.NumMinimoUnitàRicond;
22
23     insert into UnitàRicondizionate
24     select UR.UID, NULL, UV.Prodotto, UV.Variante, 0
25     from UnitàRese UR natural join UnitàVendute UV
26     where UV.CodProdotto in (
27         select CodProdotto
28         from ProdottiTarget
29     );
30
31     delete from UnitàRese
32     where UID in (
33         select UID
34         from UnitàRicondizionate
35     );
36 END $$
37 delimiter ;

```

Data Analytics

In questo paragrafo verrà presentata l'implementazione SQL delle tre data analytics presentate [precedentemente](#).

Retrieve & Reuse (Analytics 1)

Considerato che la fase di Revise e Retain non sono, per come descritte, automatizzabili (perché a discrezione del tecnico), di seguito si introduce una procedura che permette al tecnico di interagire con la base di conoscenza (Retrieve), ed eventualmente di considerare rimedi suggeriti da essa (Reuse).

Nella prima parte della procedura si guarda alla fase di *Retrieve*: viene calcolato il numero di sintomi accusati (in sostanza il numero di problemi che riscontra l'utente). Dopodiché si guarda alle esperienze passate nella base di conoscenza, e per uguaglianza di sintomi (numero di sintomi $\geq 66/100 \times$ numeroSintomiAccusati) si raccolgono le 10 (al massimo) esperienze passate con maggiore compatibilità (espressa come *percentuale di sintomi uguali*).

Nella seconda parte della procedura si passa alla fase di *Reuse*, dove si calcolano i rimedi da considerare (in base alle AIM di cui sopra) e 2 indici: il primo è uno score che deriva *dalla presenza* del rimedio in un guasto ad alta compatibilità; il secondo è uno score che deriva *unicamente da quante volte* il rimedio si ripete nei guasti ad alta compatibilità.

```
3 • CREATE PROCEDURE Retrieve( IN _assistenzaFisica VARCHAR(20) )
4 BEGIN
5
6 DECLARE quantiSintomi INTEGER DEFAULT 0;
7
8 SET quantiSintomi =
9 (
10 SELECT COUNT(*)
11 FROM SintomiAccusati
12 WHERE AssistenzaFisica = _assistenzaFisica
13 );
14
15
16 BEGIN
17 CREATE OR REPLACE VIEW Retrieve AS
18 (
19 SELECT C.AIM, C.Guasto AS GuastoProbabile, COUNT(*)*100/quantiSintomi AS Compatibilita
20 FROM Conoscenza C INNER JOIN SintomiMemorizzati SM ON SM.Conoscenza = C.AIM
21 WHERE SM.Sintomo IN (
22 SELECT Sintomo
23 FROM SintomiAccusati
24 WHERE AssistenzaFisica = _assistenzaFisica
25 )
26 GROUP BY C.AIM
27 HAVING COUNT(*) >= quantiSintomi * 66 / 100
28 ORDER BY C.Data
29 LIMIT 10
30 );
31
32 CREATE OR REPLACE VIEW RimediDaConsiderare AS
33 SELECT RU.Rimedio, RU.AIM
34 FROM RimediUtilizzati RU
35 WHERE RU.AIM IN (SELECT AIM FROM Retrieve);
36
37 CREATE OR REPLACE VIEW Fase_Reuse AS
38 SELECT R.Rimedio, SUM(IF( RC.Compatibilita BETWEEN 66 AND 79, 5*(10/COUNT(*)), 3*(10/COUNT(*)))) AS PartialScore1
39 FROM RimediDaConsiderare RC NATURAL JOIN Retrieve R
40 GROUP BY R.Rimedio;
41
42 CREATE OR REPLACE VIEW Fase_Reuse_2 AS
43 SELECT R.Rimedio, COUNT(*) AS PartialScore2
44 FROM RimediDaConsiderare RC NATURAL JOIN Retrieve R
45 GROUP BY R.Rimedio;
46
47 CREATE OR REPLACE VIEW Final_Phase AS
48 SELECT Rimedio, PartialScore1+PartialScore2 AS Score
49 FROM Fase_Reuse NATURAL JOIN Fase_Reuse_2;
50
51 SELECT *
52 FROM Final_Phase
53 ORDER BY Score DESC;
54
55 END;
56
57 END $$
58
59 DELIMITER ;
60
```

Analisi Performance Linea (Analytics 2)

```
-- Creazione e inizializzazione Tabella
DROP TABLE IF EXISTS ValutazioneLinee;
CREATE TABLE ValutazioneLinee(
  Linea VARCHAR(50) NOT NULL,
  LottoRiferimentoPrecedente INTEGER DEFAULT 0,
  NumeroLottiProdotti INTEGER DEFAULT 0,
  ValutazioneAttuale VARCHAR(20) DEFAULT NULL,
  PRIMARY KEY (Linea)
) ENGINE = InnoDB DEFAULT CHAR SET latin1;

-- Primo Riempimento: inseriamo solo linee che hanno prodotto lotti INTERI (ovvero recuperando tutte le unità perse per quel lotto
INSERT INTO ValutazioneLinee
SELECT L.CodLinea, 0, COUNT(*), NULL
FROM Linea L INNER JOIN LottoProduzione LP ON L.CodLinea = LP.Linea
WHERE LP.UnitaEffettive = LP.UnitaPreviste
GROUP BY L.CodLinea;
```

Di sopra il codice relativo alla creazione della tabella utilizzata dall'analisi. La *ValutazioneLinee* ha un record per ogni linea, e ha un certo numero di proprietà:

- **LottoRiferimentoPrecedente** deve riferirsi all'*ultimo lotto* analizzato (si ricordi che l'analisi non può farsi ogni volta che si produce un lotto)
- **NumeroLottiProdotti** è il numero di lotti prodotti dalla linea
- **Valutazione Attuale** indica l'ultima Valutazione che è stata dedicata alla linea (in termini delle stringhe descritte nel paragrafo dedicato). Questo valore all'atto della creazione (e anche del primo riempimento) è inizializzato a NULL, che indica dunque una valutazione mai effettuata.

Ogni linea è poi inserita a primo riempimento della tabella seguendo lo schema *CodLinea, 0, COUNT(*), NULL*.

Lo 0 come lotto precedente è chiaramente un errore, poiché verosimilmente nessun lotto ha quel codice. Come si vedrà di seguito lo 0 fa da *marca*, e indica quando ricercare il primo lotto della linea in atto di prima valutazione.

```
CREATE PROCEDURE PerformanceLinea()
BEGIN

  DECLARE lineaCorrente VARCHAR(20) DEFAULT '';
  DECLARE ultimaValutazione VARCHAR(30) DEFAULT '';
  DECLARE ultimoLotto INTEGER DEFAULT 0;
  DECLARE unitaPrevisteMediamente INTEGER DEFAULT 0;
  DECLARE perditaMedia INTEGER DEFAULT 0;
  DECLARE finito INTEGER DEFAULT 0;

  -- Stabilisco le linee di cui fare la valutazione
  DECLARE linee CURSOR FOR
  SELECT VL.Linea, VL.ValutazioneAttuale
  FROM ValutazioneLinee VL
  WHERE (VL.LottoRiferimentoPrecedente = 0 AND VL.NumeroLottiProdotti >= 5)
  OR (VL.LottoRiferimentoPrecedente <> 0 AND (
    SELECT COUNT(*)
    FROM LottoProduzione
    WHERE Linea = VL.Linea AND DataProduzione > (
      SELECT LP2.DataProduzione
      FROM LottoProduzione LP2
      WHERE LP2.CodLotto = VL.LottoRiferimentoPrecedente
    )) >= 5);
```

Nello spezzone di codice sopra è evidenziata la dichiarazione di variabili che serviranno: Linea Produzione e Ultima Valutazione sono le informazioni relative alla linea che si sta esaminando al momento. Questa è identificata da un *Cursore*, che recupera le linee di cui fare la valutazione, ricordando che La valutazione è effettuata solo se sono stati prodotti almeno 5 lotti dall'ultima valutazione!
In Particolare:

- Se **LottoRiferimentoPrecedente** = 0, allora la linea è candidata alla sua prima valutazione, e vi partecipa se ha prodotto almeno 5 lotti (informazione diretta da NumeroLotti)
- **Altrimenti**, si controlla che ci siano stati almeno 5 lotti dall'ultima valutazione basandoci sul fatto che due lotti non hanno mai data produzione identica!

```
INSERT INTO ValutazioneLinee
SELECT L.CodLinea, 0, COUNT(*), NULL
FROM Linea L INNER JOIN LottoProduzione LP ON L.CodLinea = LP.Linea
WHERE LP.UnitaEffettive = LP.UnitaPreviste AND L.CodLinea NOT IN ( SELECT VL2.Linea FROM ValutazioneLinee VL2)
GROUP BY L.CodLinea;
```

Nella procedura si rilancia una INSERT molto simile a quella fatta durante primo riempimento. Questo perché si vuole aggiungere alle linee a cui viene fatta la valutazione le nuove linee, **prima di aprire il Cursore!**

```
OPEN linee;

scanner : LOOP
  FETCH linee INTO lineaCorrente, ultimaValutazione;
  IF( finito = 1 ) THEN
    LEAVE scanner;
  END IF;

  SET ultimoLotto =
  (
    SELECT IF(LottoRiferimentoPrecedente = 0, (SELECT LP.CodLotto
      FROM LottoProduzione LP
      WHERE LP.Linea = lineaCorrente
      AND LP.DataProduzione = (
        SELECT MIN(LP2.DataProduzione)
        FROM LottoProduzione LP2
        WHERE LP2.Linea = lineaCorrente)), LottoRiferimentoPrecedente)

    FROM ValutazioneLinee
    WHERE Linea = lineaCorrente
  );
```

Si apre il cursore, e nella fase di Fetch si preleva una linea e la sua valutazione attuale (servirà in seguito). La prima variabile di cui si vuole parlare è **UltimoLotto**, che determina l'ultimo lotto della precedente valutazione:

- Questo coincide con **LottoRiferimentoPrecedente**, se questi è diverso da 0
- Altrimenti, UltimoLotto coincide con, si perdoni il controsenso, il primo lotto della linea, in termini di DataProduzione!

```

SET unitaPrevisteMediamente =
(
    SELECT AVG(UnitaPreviste)
    FROM Linea L INNER JOIN LottoProduzione LP ON Linea = CodLinea
    WHERE LP.Linea = lineaCorrente AND LP.DataProduzione > (
        SELECT DataProduzione
        FROM LottoProduzione
        WHERE CodLotto = ultimoLotto
    );
);

-- Nel valutare la linea corrente, abbiamo bisogno anzitutto della media unità perse nei lotti che vanno dal primo non considerato al corrente
SELECT AVG(LP.UnitaPreviste - LP.PrimaProduzione) INTO @mediaPerse
FROM Linea L INNER JOIN LottoProduzione LP ON LP.Linea = L.CodLinea
-- NATURAL JOIN ValutazioneLinee VL -- Deve essere una linea valutata!
WHERE LP.Linea = lineaCorrente AND LP.DataProduzione > (
    SELECT DataProduzione
    FROM LottoProduzione
    WHERE CodLotto = ultimoLotto
);

```

Si passa alla variabile **UnitaPrevisteMediamente**, che calcola una media delle unità previste basandosi solo nei lotti prodotti dall'ultimo lotto considerato in poi.

Dopodiché si memorizza in **MediaPerse** la media delle Prime Discrepanze tra le unità previste e quelle effettivamente realizzate in *PrimaProduzione*.

```

SET @lottoNuovo =(
    SELECT LP.CodLotto
    FROM LottoProduzione LP
    WHERE LP.Linea = lineaCorrente AND LP.DataProduzione >= ALL
    (
        SELECT DataProduzione
        FROM LottoProduzione
        WHERE Linea = lineaCorrente
    );

```

Nella variabile **LottoNuovo** memorizziamo l'ultimo lotto prodotto dalla linea al tempo della valutazione

```

SET perditaMedia = @mediaPerse/unitaPrevisteMediamente * 100;
IF(perditaMedia < 10 AND ultimaValutazione IS NULL) THEN
    UPDATE ValutazioneLinee
    SET ValutazioneAttuale = 'Ottima', LottoRiferimentoPrecedente = @lottoNuovo
    WHERE Linea = lineaCorrente;
END IF;
IF(perditaMedia < 10 AND ultimaValutazione IS NOT NULL) THEN
    UPDATE ValutazioneLinee
    SET ValutazioneAttuale = 'Ottimizzata', LottoRiferimentoPrecedente = @lottoNuovo
    WHERE Linea = lineaCorrente;
END IF;
IF(perditaMedia BETWEEN 10 AND 49) THEN
    UPDATE ValutazioneLinee
    SET ValutazioneAttuale = 'InValutazione', LottoRiferimentoPrecedente = @lottoNuovo
    WHERE Linea = lineaCorrente;

    UPDATE Linea
    SET Tempo = Tempo + 3
    WHERE CodLinea = lineaCorrente;
END IF;

```

Chiude le danze il calcolo di **perditaMedia**, che rappresenta la Percentuale di unità perse rispetto alle previste in media. Dopodiché si procede a delle casistiche che assegnano alla linea una valutazione coerente con i casi descritti ampiamente nel paragrafo dedicato a questa analisi (sopra si riportano i primi 3).

Analisi delle vendite e pianificazione della produzione (custom analytics)

Creazione delle tabelle a supporto dell'analisi di mercato:

```
CREATE TABLE DatiVendita (  
    Prodotto VARCHAR(10) NOT NULL,  
    Variante VARCHAR(15) NOT NULL,  
    Anno INT NOT NULL,  
    Trimestre INT NOT NULL,  
    GNPV INT DEFAULT 0,  
    CP INT DEFAULT 0,  
    CMP INT DEFAULT 0,  
    PPR INT DEFAULT 0,  
    CMM INT DEFAULT 0,  
    CMS INT DEFAULT 0,  
    PMR INT DEFAULT 0,  
    UP INT DEFAULT 0,  
    UV INT DEFAULT 0,  
    US INT DEFAULT 0,  
    UR INT DEFAULT 0,  
    USM INT DEFAULT 0,  
    PRIMARY KEY (Prodotto, Variante, Anno, Trimestre)  
);  
  
CREATE TABLE Score (  
    Prodotto VARCHAR(10) NOT NULL,  
    Variante VARCHAR(15) NOT NULL,  
    Anno INT NOT NULL,  
    Trimestre INT NOT NULL,  
    Punteggio INT,  
    PRIMARY KEY (Prodotto, Variante, Anno, Trimestre)  
);  
  
CREATE TABLE LOG_Unita (  
    DataAggiornamento DATE NOT NULL,  
    Prodotto VARCHAR(10) NOT NULL,  
    Variante VARCHAR(15) NOT NULL,  
    UP INT DEFAULT 0,  
    UV INT DEFAULT 0,  
    US INT DEFAULT 0,  
    UR INT DEFAULT 0,  
    USM INT DEFAULT 0,  
    PRIMARY KEY (DataAggiornamento, Prodotto, Variante)  
);
```

La tabella 'DatiVendita' possiamo considerarla come una materialized view le cui colonne contengono dati prelevati dal database e aggregati in maniera tale da rispondere alle esigenze e agli scopi dell'analisi di mercato che stiamo conducendo. Per conoscere il significato delle colonne si rimanda il lettore al paragrafo in cui è spiegata l'idea alla base dell'analisi di mercato.

Si vuole implementare un aggiornamento di tipo parziale. Per questo motivo si è deciso di implementare una tabella di log a supporto di 'DatiVendita', nella fattispecie abbiamo creato la tabella LOG_Unita. Quest'ultima conterrà, per ogni coppia prodotto-variante, le informazioni soggette a continue variazioni. Si noti che, in generale, il numero di unità vendute ma così come il numero di unità prodotte possono variare con una certa frequenza nell'arco di un trimestre e dunque si è preferito 'salvare di volta in volta' i cambiamenti. Questo favorisce l'aggiornamento della materialized view visto che non si è costretti a 'setacciare' tutto il database per recuperare queste informazioni ed inoltre senza una log table sarebbe stato impossibile implementare un aggiornamento parziale dei dati.

Implementazione dei trigger per la gestione delle tabelle di log relative alla materialized view 'DatiVendita':

```
DELIMITER $$
CREATE TRIGGER Aggiorna_LOG_UnitaPerse
AFTER INSERT ON unitaperseproduzione
FOR EACH ROW
BEGIN
    DECLARE unita INT;
    DECLARE prodo_tto VARCHAR(10);
    DECLARE varia_nte VARCHAR(15);
    DECLARE flag INT DEFAULT 0;

    SET unita = NEW.Quante;

    SELECT UD.Prodotto INTO prodo_tto
    FROM unitaperseproduzione UPP INNER JOIN lottoproduzione LP ON UPP.Lotto = LP.CodLotto
        INNER JOIN unitadisponibili UD ON UD.LottoProduzione = UPP.Lotto;

    SELECT UD.Variante INTO varia_nte
    FROM unitaperseproduzione UPP INNER JOIN lottoproduzione LP ON UPP.Lotto = LP.CodLotto
        INNER JOIN unitadisponibili UD ON UD.LottoProduzione = UPP.Lotto;

    SELECT 1 INTO flag
    FROM LOG_Unita
    WHERE Prodotto = prodo_tto AND
        Variante = varia_nte;

    IF flag = 1 THEN
        UPDATE LOG_unita
        SET UP = UP + unita, DataAggiornamento = CURRENT_DATE()
        WHERE log_unita.Prodotto = prodo_tto AND log_unita.Variante = varia_nte;
    ELSEIF flag = 0 THEN
        INSERT INTO LOG_Unita(DataAggiornamento, Prodotto, Variante, UP, UV, US, UR, USM)
        VALUES (CURRENT_DATE(), prodo_tto, varia_nte, unita, 0, 0, 0, 0);
    END IF;

END $$
DELIMITER ;
```

Ogni qual volta viene persa un'unità (in realtà possono essere più di una: ecco perché si inizializza la variabile unita = NEW.Quante) in una linea di produzione viene inserita questa informazione nella tabella di log. Notiamo che se in LOG_Unita è già presente un record relativo alla coppia prodotto-variante allora è sufficiente aumentare il valore della colonna 'UP' (UP = unità perse). Nel caso in cui non esista ancora un record (può succedere subito dopo che si è aggiornata la tabella 'DatiVendita': se il record aveva un valore in DataAggiornamento <= alla data a cui si vuole allineata la materialized view allora le informazioni vengono inserite in DatiVendita e il record viene cancellato da LOG_Unita; si rimanda il lettore alla fine del paragrafo dove verrà esaminato l'aggiornamento parziale della tabella) allora si effettua l'inserimento di un nuovo record nella log table e si inizializza a 'unita' il valore della colonna 'UP'.

Si noti infine che sia nel caso il record sia già presente sia nel caso il record sia stato appena creato, si setta il valore di DataAggiornamento alla data corrente. È necessario far ciò per tenere traccia delle nuove informazioni inserite.

```
DELIMITER $$
CREATE TRIGGER Aggiorna_LOG_UnitaVendute
AFTER INSERT ON unitavendute
FOR EACH ROW
BEGIN
    DECLARE flag INT DEFAULT 0;

    SELECT 1 INTO flag
    FROM LOG_Unita
    WHERE Prodotto = NEW.Prodotto AND
           Variante = NEW.Variante;

    IF flag = 1 THEN
        UPDATE LOG_Unita
        SET UV = UV + 1, DataAggiornamento = CURRENT_DATE()
        WHERE LOG_Unita.Prodotto = NEW.Prodotto AND
              LOG_Unita.Prodotto = NEW.Prodotto;
    ELSEIF flag = 0 THEN
        INSERT INTO LOG_Unita(DataAggiornamento, Prodotto, Variante, UP, UV, US, UR, USM)
        VALUES (CURRENT_DATE(), NEW.Prodotto, NEW.Variante, 0, 1, 0, 0, 0);
    END IF;

END $$
DELIMITER ;
```



```

DELIMITER $$
CREATE TRIGGER Aggiorna_LOG_UnitaStoccate
AFTER INSERT ON unitadisponibili
FOR EACH ROW
BEGIN
    DECLARE flag INT DEFAULT 0;
    DECLARE var INT DEFAULT 0;

    SELECT 1 INTO flag
    FROM LOG_Unita
    WHERE Prodotto = NEW.Prodotto AND
        Variante = NEW.Variante;

    IF flag = 1 THEN
        UPDATE LOG_Unita
        SET US = US + 1, DataAggiornamento = CURRENT_DATE()
        WHERE LOG_Unita.Prodotto = NEW.Prodotto AND
            LOG_Unita.Prodotto = NEW.Prodotto;
    ELSEIF flag = 0 THEN
        SELECT COUNT(*) INTO var
        FROM unitadisponibili
        WHERE Prodotto = NEW.Prodotto AND
            Variante = NEW.Variante;
        INSERT INTO LOG_Unita(DataAggiornamento, Prodotto, Variante, UP, UV, US, UR, USM)
        VALUES (CURRENT_DATE(), NEW.Prodotto, NEW.Variante, 0, 0, var, 0, 0);
    END IF;

END $$

```

```

DELIMITER $$
CREATE TRIGGER Aggiorna_LOG_UnitaRicondizionate
AFTER INSERT ON unitaricondizionate
FOR EACH ROW
BEGIN
    DECLARE flag INT DEFAULT 0;
    DECLARE var INT DEFAULT 0;

    SELECT 1 INTO flag
    FROM LOG_Unita
    WHERE Prodotto = NEW.Prodotto AND
           Variante = NEW.Variante;

    IF flag = 1 THEN
        UPDATE LOG_Unita
        SET UR = UR + 1, DataAggiornamento = CURRENT_DATE()
        WHERE LOG_Unita.Prodotto = NEW.Prodotto AND
              LOG_Unita.Prodotto = NEW.Prodotto;
    ELSEIF flag = 0 THEN
        INSERT INTO LOG_Unita(DataAggiornamento, Prodotto, Variante, UP, UV, US, UR, USM)
        VALUES (CURRENT_DATE(), NEW.Prodotto, NEW.Variante, 0, 0, 0, 1, 0);
    END IF;

END $$
DELIMITER ;

```

I tre trigger sopra presentati funzionano tutti con la stessa logica. Ogni qual volta avviene un inserimento nelle tabelle UnitaXX (dove XX prende il valore di 'vendute', 'stoccate', 'ricondizionate' riferendoci al primo, secondo e terzo trigger rispettivamente) si effettua un controllo nella tabella di log per verificare se già esiste un record avente nella coppia prodotto-variante gli stessi valori di NEW.Prodotto e NEW.Variante; se questo accade il trigger incrementa di uno il valore della colonna che si riferisce alla tabella su cui si è attivato il trigger. Facciamo un esempio: se viene inserito una nuova unità stoccata allora si attiverà il trigger 'Aggiorna_LOG_UnitaStoccate e aumenterà di uno il valore della colonna 'US'.

Nel caso in cui non esiste di già un record che abbia gli stessi valori di NEW.Prodotto, NEW.Variante nella coppia prodotto-variante allora si inserisce un nuovo record nella tabella di log e si inizializza ad 1 il valore della colonna inerente alla tabella su cui si è attivato il trigger.

```

DELIMITER $$
CREATE TRIGGER Aggiorna_LOG_UnitaSmaltite1
AFTER INSERT ON recuperomateriale
FOR EACH ROW
BEGIN
    DECLARE flag INT DEFAULT 0;
    DECLARE var INT DEFAULT 0;

    SELECT COUNT(*) INTO flag
    FROM recuperomateriale
    WHERE UnitaEndOfLife = NEW.UnitaEndOfLife;

    SELECT 1 INTO var
    FROM log_unita
    WHERE Prodotto = (SELECT Prodotto FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife) AND
        Variante = (SELECT Variante FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife);

    IF flag <= 1 THEN
        IF var = 1 THEN
            UPDATE log_unita
            SET USM = USM + 1, DataAggiornamento = CURRENT_DATE()
            WHERE Prodotto = (SELECT Prodotto FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife) AND
                Variante = (SELECT Variante FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife);

        ELSEIF flag = 0 THEN
            INSERT INTO LOG_Unita(DataAggiornamento, Prodotto, Variante, UP, UV, US, UR, USM)
            VALUES (CURRENT_DATE(),
                (SELECT Prodotto FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife),
                (SELECT Variante FROM unitaendoflife WHERE UID = NEW.UnitaEndOfLife),
                0, 0, 0, 0, 1);

        END IF;
    END IF;

END $$
DELIMITER ;

```

Gli ultimi due trigger possiamo considerarli gemelli. Il primo trigger opera sulla tabella *RecuperoMateriale* mentre il secondo trigger opera sulla tabella *RecuperoParte*. Notiamo che siamo interessati a salvare il numero di unità oggetto di procedura di smaltimento. Questo significa che siamo interessati a trovare le unità che sono state 'donatrici' di almeno una parte oppure di una qualche quantità di materiali. Non sarebbe stato sufficiente controllare i nuovi inserimenti nella tabella *UnitaEndOfLife* perché a priori non possiamo conoscere se quell'unità, durante la procedura di smaltimento, ci potrà portare del ricavo.

Spieghiamo il funzionamento del trigger 'Aggiorna_LOG_UnitaSmaltimento1':

Si contano i record presenti nella tabella *RecuperoMateriale* e *RecuperoParte* aventi valore nella colonna 'UnitaEndOfLife' uguale a *NEW.UnitaEndOfLife*. Se il record appena inserito è anche l'unico presente nella tabella *RecuperoMateriale* allora è plausibile che l'unità a cui ci si riferisce abbia appena iniziato la procedura di smaltimento e quindi deve essere inserita nella log table. L'inserimento avviene se e solo se non sia presente alcun record, relativo all'unità appena inserita, nella tabella *RecuperoParte*. Questo perché si è interessati a conoscere il numero di unità che hanno portato un ricavo dal loro smaltimento ma non ci interessa conoscere qual è l'entità del guadagno. Sarà sufficiente aggiornare al log table al verificarsi del primo inserimento dell'unità in una delle due tabelle e ignorare gli eventuali inserimenti successivi.

```

DELIMITER $$
CREATE PROCEDURE MV_partial_refresh(IN _var DATE)
BEGIN
    DECLARE prezzo INT;
    DECLARE cp INT;
    DECLARE cmp INT;
    DECLARE ppr INT;
    DECLARE cmm INT;
    DECLARE cms INT;
    DECLARE pmr INT;
    DECLARE prodotto VARCHAR(10);
    DECLARE variante VARCHAR(15);
    DECLARE up INT;
    DECLARE uv INT;
    DECLARE us INT;
    DECLARE ur INT;
    DECLARE usm INT;

    DECLARE flag INT DEFAULT 1;

    DECLARE cursore CURSOR FOR
    SELECT Prodotto, Variante, UP, UV, US, UR, USM
    FROM log_unita
    WHERE _var >= DataAggiornamento;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET flag = 0;

    ciclo: LOOP
        IF flag = 1 THEN
            FETCH cursore INTO prodotto, variante, up, uv, us, ur, usm;
            /*Salvo la coppia prodotto-variante, prelevata dalla log table, il cui ultimo aggiornamento è avvenuto prima della data specificata.*/
            SELECT V.Prezzo INTO prezzo FROM Varianza V WHERE V.Prodotto = prodotto AND V.Variante = variante;
            /*Salvo nella variabile cp il prezzo attuale di vendita di quella variante del prodotto.*/
            SELECT SUM(P.Prezzo*C.Pezzi) INTO cp FROM Parte P INNER JOIN Composizione C ON P.CodParte = C.Parte WHERE C.Prodotto = prodotto;
            /*Calcolo il costo di produzione di quella tipologia di prodotto.*/
            SELECT AVG(Prezzo) INTO cmp FROM Parte WHERE Prodotto = prodotto;
            /*Calcolo il costo medio delle parti che compongono il prodotto.*/
            WITH A AS (
                SELECT UEOL.UID,
                    SUM(Quantita) AS tot,
                    (SELECT COUNT(*) FROM Composizione WHERE Prodotto = prodotto) AS Parti
                FROM recuperoparte RP INNER JOIN unitaendoflife UEOL ON RP.UnitaEndOfLife = UEOL.UID
                WHERE UEOL.Prodotto = prodotto
                GROUP BY UID
            )
            SELECT AVG((tot*100)/Parti) INTO ppr /*Calcolo la media delle percentuali di parti recuperate sino in quel momento.*/
            FROM A;

            SELECT AVG(M.Valore) INTO cmm /*Calcolo la media dei valori dei materiali presenti nel prodotto.*/
            FROM Materiale M INNER JOIN Struttura S ON M.Nome = S.Materiale
            WHERE S.Parte IN (SELECT CodParte
                            FROM Parte P INNER JOIN Composizione C ON P.CodParte = C.Parte
                            WHERE C.Prodotto = prodotto);
        END IF;
    END LOOP;

```

```

SELECT AVG(M.CoeffSvalutazione) INTO cms          /*Calcolo la media dei coefficienti di svalutazione dei materiali presenti*/
FROM Materiale M INNER JOIN Struttura S ON M.Nome = S.Materiale          /*nel prodotto.*/
WHERE S.Parte IN (SELECT CodParte
                  FROM Parte P INNER JOIN Composizione C ON P.CodParte = C.Parte
                  WHERE C.Prodotto = prodotto);

WITH B AS (
SELECT UEOL.UID,
      SUM(Quantita) AS tot,
      (SELECT peso FROM infovarianza WHERE Prodotto = prodotto AND Variante = variante) AS PesoTot
FROM recuperomateriale RM INNER JOIN unitaendoflife UEOL ON RM.UnitaEndOfLife = UEOL.UID
WHERE UEOL.Prodotto = prodotto
GROUP BY UID
)
SELECT AVG((tot*100)/PesoTot) INTO ppr          /*Calcolo la media della percentuale di materiale recuperato sino al momento*/
FROM B;          /*per quel prodotto.*/

INSERT INTO DatiVendita VALUES (prodotto, variante, YEAR(_var), trimestre(_var), (prezzo-cp), cp, cmp, ppr, cmm, cms, pmr, up, uv, us, ur, usm);
ELSEIF flag = 0 THEN
  LEAVE ciclo;
END IF;
END LOOP;

DELETE FROM log_unita WHERE DataAggiornamento <= _var;

END $$
DELIMITER ;

```

La procedure MV_partial_refresh si occupa di aggiornare, parzialmente sino ad una certa data passata come input della procedura, la tabella DatiVendita. Per fare questo si considerano tutti i record presenti nella tabella di log (LOG_Unita) tali che l'ultimo la data dell'ultimo aggiornamento effettuato sui record sia minore o uguale della data desiderata avuta come input dalla procedura. Si noti che dai record estratti dalla tabella LOG_Unita si vuole ottenere quelle informazioni che sono frutto delle attivazioni dei trigger sopra elencati. Gli indici usati per l'analisi di mercato che sono soggetti a frequenti cambiamenti sono esclusivamente quelli contenuti nella tabella di log, il che giustifica il trattamento a parte compiuto su di essi. Inoltre, per alleggerire le operazioni in seguito all'inserito di un record inerente all'analisi di mercato, si è deciso di implementare esclusivamente i trigger relativi alle unità (vendute, disponibili, ricondizionate, EndOfLife) dal momento che gli indici riguardanti i costi sono onerosi da calcolare (query su più tabelle) e soggetti a meno cambiamenti. Il calcolo di questi si rimanda durante l'esecuzione della procedura MV_partial_refresh.

Per ogni record estratto dalla tabella LOG_Unita che rispetti i criteri sopra citati vengono calcolati gli indici CP, CMP, PPR, CMM, CMS, PMR e in seguito viene inserito un nuovo record nella tabella DatiVendita. Si noti che non siamo interessati ad aggiornare record già presenti nella tabella DatiVendita ma piuttosto dobbiamo considerare la tabella come uno storico delle analisi di mercato. Questo perché si vuole valutare la convenienza di produzione di un certo prodotto-variante relativamente ai precedenti trimestri.

Una volta compiuti tutti gli inserimenti nella tabella DatiVendita si cancellano tutti i record della tabella LOG_Unita che sono stati usati nella corrente esecuzione della procedura.

```
DROP EVENT IF EXISTS MV_deferred_refresh;
```

```

CREATE EVENT MV_deferred_refresh
ON SCHEDULE EVERY 3 MONTH
STARTS '2020-04-01 23:55'
DO
CALL MV_partial_refresh(CURRENT_DATE());

```

L'event qui di fianco richiama la procedura MV_partial_refresh passando come input la data corrente.

Questo perché si vuole aggiornare la materialized view trimestralmente, in accordo con le tempistiche fissate dall'analisi di mercato.

```
DELIMITER $$
CREATE PROCEDURE BuildScore(IN _Prodotto VARCHAR(10), IN _Variante VARCHAR(15))
BEGIN
    DECLARE GuadagnoVendita INT DEFAULT 0;
    DECLARE GuadagnoSospeso INT DEFAULT 0;
    DECLARE Trime_stre INT;
    DECLARE An_no INT;
    DECLARE Score INT DEFAULT 0;

    IF MONTH(CURRENT_DATE()) <= 3 THEN
        SET Trime_stre = 1;
    ELSEIF (MONTH(CURRENT_DATE()) > 3 AND MONTH(CURRENT_DATE()) <= 6) THEN
        SET Trime_stre = 2;
    ELSEIF (MONTH(CURRENT_DATE()) > 6 AND MONTH(CURRENT_DATE()) <= 9) THEN
        SET Trime_stre = 3;
    ELSEIF (MONTH(CURRENT_DATE()) > 9) THEN
        SET Trime_stre = 4;
    END IF;

    SET GuadagnoVendita = (SELECT GNPV*(UV - UP) + USM*(CMP*PPR + CMV*PMR*CMS)
        FROM DatiVendita DV
        WHERE DV.Prodotto = _Prodotto AND
              DV.Variante = _Variante AND
              DV.Anno = YEAR(CURRENT_DATE()) AND
              MONTH(CURRENT_DATE()) <= DV.Trimestre*3 AND
              MONTH(CURRENT_DATE()) > DV.Trimestre*3 - 3);

    SET GuadagnoSospeso = (SELECT UR*(GNPU*0.25) + US*(GNPU*0.75)
        FROM DatiVendita DV
        WHERE DV.Prodotto = _Prodotto AND
              DV.Variante = _Variante AND
              DV.Anno = YEAR(CURRENT_DATE()) AND
              MONTH(CURRENT_DATE()) <= DV.Trimestre*3 AND
              MONTH(CURRENT_DATE()) > DV.Trimestre*3 - 3);

    SET Score = GuadagnoVendita - GuadagnoSospeso;
    SET An_no = YEAR(current_date());

    INSERT INTO Score(Prodotto, Variante, Anno, Trimestre, Punteggio)
    VALUES (_Prodotto, _Variante, An_no, Trime_stre, Score);

END $$
DELIMITER ;
```

La procedure BuildScore permette di calcolare lo score di un prodotto-variante. Viene chiamata all'interno della procedura MV_partial_refresh visto che si è intenzionati a calcolare lo score per ogni coppia prodotto-variante del quale si è compiuto un inserimento nella tabella DatiVendita.

Procedure di Appoggio

Oltre alle procedure sopra descritte, si è scelto di metterne altre a disposizione dell'azienda. Riportiamo nella parte sottostante alcune di esse, ma rimandiamo alla visione del file *ImplementazioneSQL* per avere una visione più ampia del codice:

- **MostraOrario:** è una procedura destinata agli utenti che decidano di richiedere assistenza fisica. Mostra l'orario dei tecnici *liberi* della stessa provincia dell'utente che richiede assistenza; sarà poi discrezione dell'utente decidere di richiedere per quelle fasce orarie o rallentare il processo imponendo una fascia diversa.
- **AssegnaIncarichi:** procedura che assegna un numero di incarichi richiesti al tecnico che li richiede (comunque al massimo 3). Gli incarichi sono ricercati **nella stessa provincia del tecnico** in due modi:
 - Si dà precedenza agli interventi fisici *non completamente assegnati* (non tutti i tecnici voluti sono stati trovati disponibili nella fascia oraria richiesta).
 - Se i precedenti non fossero abbastanza, si cerca la richiesta di assistenza fisica *più datata*, e si assegna al tecnico (creando un nuovo record in *InterventoFisico*). Si noti che l'intervento così creato è "Assegnato" se e solo se era stato richiesto un unico tecnico.

- **AssegnaOperatore:** procedura che assegna ad una stazione un operatore *libero* (ovvero che non è impegnato né in una stazione di produzione né in una stazione di smaltimento). L'operatore è scelto in base al tempo impiegato nel compiere il set di operazioni per quella classe campione (è scelto il più veloce disponibile). Questa procedura è decisamente utile per compiere il passo B dell'analytics numero 2.
- **TempoLinea:** procedura che assegna il tempo T della linea secondo le specifiche descritte al punto C dell'analytics numeri 2 (tempo massimo previsto di ogni stazione arrotondato per eccesso).