PROGETTO SUPERMERCATO

Dario Stefano Shaheen 181066
Andrea Patricelli 179260

Analisi dei requisiti

individuazione dei concetti rilevanti

Problema
Si vuole realizzare una base di dati per l'organizzazione di un **supermercato**. Sebbene quanto richiesto sia stato semplificato e non copra interamente tutti gli aspetti di gestione, il sistema realizzato rappresenterà una base reale e significativa.
Il supermercato è strutturato in una serie di reparti. Ogni reparto ha un nome, un responsabile e una serie di impiegati, ognuno con una mansione specifica. Di ogni impiegato vogliamo memorizzare tutti i principali dati anagrafici (nome, indirizzo, telefono, codice fiscale, data di nascita,…) oltre alla data di assunzione. Ogni impiegato lavora in un determinato reparto ed ha un particolare livello (che ne determina lo stipendio insieme all'anzianità di servizio).
Il supermercato mette in vendita una serie di prodotti, dei quali vogliamo memorizzare nome, genere ("preparazione alimentare", "prodotto di profumeria",…), il prezzo al pubblico, il tempo massimo entro il quale il prodotto deve essere venduto (ad es. se il prodotto è deperibile, il tempo potrebbe essere di solo qualche giorno), il reparto in cui è posto in vendita, la quantità di prodotto presente in magazzino e la soglia minima (determinata dall'esperienza, o da un'apposita interrogazione) al di sotto della quale il prodotto va riordinato. A ciascun prodotto viene inoltre assegnato un codice interno. Alcuni prodotti non sono acquistati direttamente dai fornitori, ma "assemblati" o "preparati" direttamente nel supermercato (ad esempio, delle confezioni regalo di profumi in offerta speciale, o un piatto precotto) a partire da una serie di materie prime (che a loro volta sono prodotti disponibili nel magazzino). Per questi prodotti "composti" vogliamo conoscere anche tutti i prodotti "ingredienti" e la quantità di essi necessaria alla loro preparazione.
La base di dati contiene una lista dei fornitori a cui il supermercato fa riferimento. Per ogni fornitore vogliamo conoscere la ragione sociale, la partita IVA, la modalità di pagamento richiesta (es. bonifico a 60 giorni), oltre all'indirizzo e al recapito telefonico e di fax. Ogni fornitore può fornire, a un particolare prezzo, uno o più dei prodotti venduti dal supermercato, identificandoli con un codice interno (diverso da quello usato dal magazzino del supermercato!) da indicare negli ordini.
Per quel che riguarda i clienti del supermercato, alcuni di essi possono essere "fidelizzati" perché hanno richiesto una speciale tessera, che permette loro di accumulare punti con gli acquisti e ottenere sconti o regali. I clienti con tessera sono registrati nella base di dati con i loro dati anagrafici e il numero di tessera, e a ciascuno è associato il numero di punti correntemente accumulati.
Per semplificare, supponiamo che i "premi" delle raccolte punti siano a loro volta prodotti in vendita nel supermercato. Per ciascun premio, oltre al prodotto associato, vogliamo conoscere il numero di punti necessari ad ottenerlo. I punti si ottengono acquistando particolari prodotti. E' necessario mantenere una lista dei

prodotti che partecipano alla raccolta punti, unitamente al numero di punti ottenibili con il loro acquisto. Ogni volta che il cliente ritira un premio, vengono scalati i corrispondenti punti dalla sua tessera.
I registratori di cassa del nostro supermercato sono direttamente interfacciati con la base i dati (come succede molto spesso oggigiorno). Ogni volta che un prodotto viene venduto, la base di dati registra i dati della vendita (numero di scontrino, prodotto venduto, prezzo al quale è stato venduto) unitamente al codice del cliente al quale è stato venduto, se questo è titolare di una tessera. Contemporaneamente, vengono aggiornate la disponibilità del prodotto in magazzino e il numero di punti del cliente (se il prodotto in questione dava diritto a dei punti-raccolta).

GLOSSARIO DEI TERMINI

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Reparto	È un settore del supermercato. Contiene prodotti che hanno caratteristiche comuni.	Settore	Impiegato, prodotto
Impiegato	Dipendente del supermercato assegnato ad un certo reparto.	Dipendente, lavoratore	Reparto
Prodotto	tutto ciò che può essere offerto dal supermercato per acquisizione, uso o consumo.	Merce, oggetto da vendere	Reparto, scontrino (vendita), cliente
Cliente	Colui che consuma (compra) i prodotti del supermercato.	Consumatore	Prodotto, scontrino (vendita)
Fornitore	L'azienda oppure la persona fisica che fornisce i prodotti da vendere al supermercato		prodotto
Scontrino (vendita)	Documento emesso all'atto della vendita che registra i prodotti venduti.	Fattura, ricevuta	Prodotto, cliente
Premio	Prodotto ottenuto da un cliente fidelizzato tramite una raccolta punti		Prodotto, cliente, scontrino (vendita)

Frasi relative a: reparto

Il supermercato è strutturato in una serie di reparti. Ogni reparto ha un nome, un responsabile e una serie di impiegati, ognuno con una mansione specifica.

Frasi relative a: impiegato

Di ogni impiegato vogliamo memorizzare tutti i principali dati anagrafici (nome, indirizzo, telefono, codice fiscale, data di nascita,…) oltre alla data di assunzione. Ogni impiegato lavora in un determinato reparto ed ha un particolare livello (che ne determina lo stipendio insieme all'anzianità di servizio).

Frasi relative a: prodotto

Il supermercato mette in vendita una serie di prodotti, dei quali vogliamo memorizzare nome, genere ("preparazione alimentare", "prodotto di profumeria",…), il prezzo al pubblico, il tempo massimo entro il quale il prodotto deve essere venduto (ad es. se il prodotto è deperibile, il tempo potrebbe essere di solo qualche giorno), il reparto in cui è posto in vendita, la quantità di prodotto presente in magazzino e la soglia minima (determinata dall'esperienza, o da un'apposita interrogazione) al di sotto della quale il prodotto va riordinato.

A ciascun prodotto viene inoltre assegnato un codice interno. Alcuni prodotti non sono acquistati direttamente dai fornitori, ma "assemblati" o "preparati" direttamente nel supermercato (ad esempio, delle confezioni regalo di profumi in offerta speciale, o un piatto precotto) a partire da una serie di materie prime (che a loro volta sono prodotti disponibili nel magazzino). Per questi prodotti "composti" vogliamo conoscere anche tutti i prodotti "ingredienti" e la quantità di essi necessaria alla loro preparazione.

Ogni fornitore può fornire, a un particolare prezzo, uno o più dei prodotti venduti dal supermercato, identificandoli con un codice
interno (diverso da quello usato dal magazzino del supermercato!) da indicare negli ordini.

Frasi relative a: fornitore

La base di dati contiene una lista dei fornitori a cui il supermercato fa riferimento. Per ogni fornitore vogliamo conoscere la ragione sociale, la partita IVA, la modalità di pagamento richiesta (es. bonifico a 60 giorni), oltre all'indirizzo e al recapito telefonico e di fax. Ogni fornitore può fornire, a un particolare prezzo, uno o più dei prodotti venduti dal supermercato, identificandoli con un codice interno (diverso da quello usato dal magazzino del supermercato!) da indicare negli ordini.

Frasi relative a: cliente

Per quel che riguarda i clienti del supermercato, alcuni di essi possono essere "fidelizzati" perché hanno richiesto una speciale tessera, che permette loro di accumulare punti con gli acquisti e ottenere sconti o regali. I clienti con tessera sono registrati nella base di dati con i loro dati anagrafici e il numero di tessera, e

a ciascuno è associato il numero di punti correntemente accumulati.
Ogni volta che un
prodotto viene venduto, la base di dati registra i dati della vendita (numero di scontrino, prodotto venduto, prezzo al quale è stato venduto) unitamente al codice del cliente al quale è stato venduto, se questo è titolare di una tessera.

Frasi relative a: scontrino (vendita)

Ogni volta che un
prodotto viene venduto, la base di dati registra i dati della vendita (numero di scontrino, prodotto venduto, prezzo al quale è stato venduto) unitamente al codice del cliente al quale è stato venduto, se questo è titolare di una tessera.

Frasi relative a: premio

Per semplificare, supponiamo che i "premi" delle raccolte punti siano a loro volta prodotti in vendita nel supermercato. Per ciascun premio, oltre al prodotto associato, vogliamo conoscere il numero di punti necessari ad ottenerlo. I punti si ottengono acquistando particolari prodotti. E' necessario mantenere una lista dei prodotti che partecipano alla raccolta punti, unitamente al numero di punti ottenibili con il loro acquisto. Ogni volta che il cliente ritira un premio, vengono scalati i corrispondenti punti dalla sua tessera.

SCHEMA CONCETTUALE

Si veda il file "schema_concettuale.dia".

TABELLA DELLE ENTITA'

ENTITA'	ATTRIBUTI	IDENTIFICATORE
IMPIEGATO	Mansione, data assunzione, livello, codice fiscale, stipendio, telefono, indirizzo, reparto, responsabile(sì/no)	**CODICE FISCALE**
REPARTO	Responsabile, Nome	**NOME**
PRODOTTO	prezzo, codice interno, scadenza, qtà in magazzino, nome, soglia minima, genere, reparto	**CODICE INTERNO**
FORNITORE	Indirizzo, telefono, mod. pagamento, p.iva, ragione sociale	**P. IVA**
CLIENTE	codice fiscale, nome, indirizzo	**CODICE FISCALE**
CLIENTE FIDELIZZATO	codice fiscale, nome, indirizzo, num. tessera, punti accumulati	
PRODOTTO CHE PARTECIPA ALLA RACCOLTA PUNTI	Punti che fornisce, prezzo, codice interno, scadenza, qtà in magazzino, nome, soglia minima, genere, reparto	
PREMIO	prezzo, codice interno, scadenza, qtà in magazzino, nome, soglia minima, genere, reparto, punti	
PRODOTTO COMPOSTO	prezzo, codice interno, scadenza, qtà in magazzino, nome, soglia minima, genere, reparto	

TABELLA DELLE ASSOCIAZIONI

ASSOCIAZIONI	ENTITA' COINVOLTE	ATTRIBUTI
AFFERENZA	Impiegato, reparto	
APPARTENENZA	Prodotto, reparto	
SUDDIVISIONE	Prodotto, prodotto composto	Quantità
FORNITURA	Fornitore, prodotto	Codice interno
DISPONIBILITA'	Fornitore, prodotto	Prezzo richiesto
VENDITA	prodotto, cliente	Numero scontrino, data vendita
RITIRO	Cliente fidelizzato, premio	

REGOLE DI VINCOLO

1. Un impiegato non può lavorare in più reparti.
2. Il responsabile di un reparto deve essere un impiegato di esso.
3. Prezzo (prodotto) > 0, punti per vincerlo (premio) > 0, punti forniti (prodotto che partecipa alla raccolta punti) > 0.
4. In prodotto Quantità magazzino >= Soglia minima.
5. In impiegato stipendio > 0, livello >= 0
6. I prodotti forniti dal fornitore sono anche prodotti disponibili dal fornitore.
7. Un prodotto può essere presente una sola volta in uno scontrino.
8. Un cliente fidelizzato può ritirare lo stesso premio una e una sola volta.
9. Il prezzo richiesto dal fornitore per la fornitura di un prodotto da lui disponibile dev'essere minore o uguale al prezzo al pubblico del prodotto in questione.
10. Un prodotto non può essere composto da un prodotto del quale è un componente e viceversa.
11. Per un cliente fidelizzato il numero di tessera è unico (è una chiave candidata), il numero di punti accumulati dev'essere maggiore o uguale a zero.
12. Si considera molto venduto un prodotto le cui occorrenze vendute sono maggiori o uguali a 100.
13. Si considera molto venduto ad un cliente un prodotto le cui occorrenze vendute al cliente in questione sono maggiori o uguali a 20.

OPERAZIONI

1. Modifica del responsabile e degli impiegati di un reparto;
2. Determinazione delle vendite per un reparto in un particolare periodo;
3. Determinazione dei prodotti più venduti in un determinato reparto.
4. Modifica del prezzo di un prodotto;
5. Modifica dei dati riguardanti le scorte di prodotto disponibili;
6. Per i prodotti “composti”, specifica degli “ingredienti” e delle quantità necessarie alla preparazione;
7. Inclusione o esclusione di un prodotto dalla raccolta punti e indicazione del numero di punti raccolta forniti dal prodotto;
8. Determinazione dei prodotti sotto scorta;
9. Lista dei fornitori dai quali un determinato prodotto può essere acquistato, ordinati in base al prezzo richiesto;
10. Modifica del numero di punti necessari ad ottenere un determinato premio;
11. Verifica dei premi attualmente disponibili.
12. Inserimento/Modifica dei prodotti forniti da un fornitore.
13. Modifica del reparto di assegnazione di un impiegato e del suo livello.
14. Modifica del numero di punti assegnati al cliente;
15. Determinazione dei premi a cui un cliente ha diritto;
16. Ritiro di un premio da parte di un cliente;
17. Determinazione dei prodotti più acquistati da un cliente;
18. Determinazione della spesa totale effettuata da un cliente in un determinato periodo.

TAVOLA DEI VOLUMI

CONCETTO	TIPO	VOLUME
IMPIEGATO	E	50
REPARTO	E	10
PRODOTTO	E	500
PRODOTTO CHE PARTECIPA ALLA RACCOLTA PUNTI	E	300
PREMIO	E	50
PRODOTTO COMPOSTO	E	100
FORNITORE	E	20
CLIENTE	E	10000
CLIENTE FIDELIZZATO	E	1000
AFFERENZA	R	50
APPARTENENZA	R	500
VENDITA	R	10000 AL GIORNO
RITIRO	R	5000
SUDDIVISIONE	R	500
FORNITURA	R	100 A SETTIMANA
DISPONIBILITA'	R	1000

TAVOLA DELLE OPERAZIONI

OPERAZIONE	TIPO	FREQUENZA
Op. 3	B	1 VOLTA OGNI 3 MESI
Op. 4	I	1 VOLTA AL MESE
Op. 5	I	1 VOLTA AL MESE
Op. 6	B	1 VOLTA OGNI 3 MESI
Op. 7	B	1000 VOLTE AL GIORNO
Op. 8	I	1 VOLTA AL MESE
Op. 9	B	1 VOLTA AL MESE
Op. 10	I	1 VOLTA AL GIORNO
Op. 11	I	1 VOLTA ALLA SETTIMANA
Op. 12	B	1 VOLTA AL MESE
Op. 13	I	1 VOLTA ALLA SETTIMANA
Op. 14	B	1 VOLTA ALLA SETTIMANA
Op. 15	B	OGNI VOLTA CHE UN IMPIEGATO AVANZA DI UN ANNO DI ANZIANITA'
Op. 16	B	OGNI VOLTA CHE UN CLIENTE FIDELIZZATO COMPRA UN PRODOTTO CHE PARTECIPA ALLA RACCOLTA PUNTI
Op. 17	I	OGNI VOLTA CHE IL CLIENTE LO RICHIEDE
Op. 18	B	1 VOLTA AL GIORNO
Op. 19	I	1 VOLTA AL MESE
Op. 20	I	1 VOLTA AL MESE

ANALISI DELLE OPERAZIONI

OP.3

CONCETTO	COSTRUTTO	ACCESSO	TIPO
REPARTO	E	1	S
APPARTENENZA	R	1	S

OP. 4

CONCETTO	COSTRUTTO	ACCESSO	TIPO
VENDITA	R	N	L
APPARTENENZA	R	1	L

OP. 5

CONCETTO	COSTRUTTO	ACCESSO	TIPO
VENDITA	R	N	L
APPARTENENZA	R	N	L

OP. 6

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRODOTTO	E	1	S

OP. 7

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRODOTTO	E	1	S

OP. 8

CONCETTO	COSTRUTTO	ACCESSO	TIPO
SUDDIVISIONE	R	1	L

OP. 9

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRODOTTO CHE FORNISCE PUNTI	E	1	S
PRODOTTO CHE FORNISCE PUNTI	E	1	L

OP.10

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PRODOTTO	E	N	L

OP. 11

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DISPONIBILITA'	R	N	L

OP. 12

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PREMIO	E	1	S

OP. 13

CONCETTO	COSTRUTTO	ACCESSO	TIPO
PREMIO	E	N	L

OP. 14

CONCETTO	COSTRUTTO	ACCESSO	TIPO
DISPONIBILITA'	R	N	S

OP. 15

CONCETTO	COSTRUTTO	ACCESSO	TIPO
AFFERENZA	R	1	S
IMPIEGATO	E	1	S

OP. 16

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE FIDELIZZATO	E	1	S

OP. 17

CONCETTO	COSTRUTTO	ACCESSO	TIPO
CLIENTE FIDELIZZATO	E	1	L
PREMIO	E	N	L

OP. 18

CONCETTO	COSTRUTTO	ACCESSO	TIPO
RITIRO	R	1	S
CLIENTE FIDELIZZATO	E	1	S
PRODOTTO	E	1	S

OP. 19

CONCETTO	COSTRUTTO	ACCESSO	TIPO
VENDITA	R	N	L

OP. 20

CONCETTO	COSTRUTTO	ACCESSO	TIPO
VENDITA	R	N	L
PRODOTTO	E	N	L

RISTRUTTURAZIONE DELLO SCHEMA E-R

ELIMINAZIONE DELLE GENERALIZZAZIONI

L'entità PRODOTTO CHE PARTECIPA ALLA RACCOLTA PUNTI si può facilmente accorpare nel padre PRODOTTO, aggiungendo a questo un attributo “numero di punti che fornisce”, che avrà valore zero nel caso che il prodotto non partecipi alla raccolta punti (modificando quindi il vincolo relativo a quest'attributo).
Anche l'entità PRODOTTO COMPOSTO può essere velocemente sottintesa inglobandola nel padre PRODOTTO e facendo sì che si abbia una relationship ricorsiva con ruoli (COMPOSTO e COMPONENTE) con diverse cardinalità da quelle precedenti ((0,N) e (0,N)).
L'entità PREMIO si può inglobare nel padre PRODOTTO aggiungendo a questo un attributo “numero di punti per vincerlo” con cardinalità (0,1).
Per quanto riguarda la parentela tra CLIENTE e CLIENTE FIDELIZZATO, comprendendo meglio il testo si può dedurre che nella base di dati non devono essere memorizzati tutti i clienti, ma solo i clienti fidelizzati, quindi si uniscono le due entità in un'entità sola chiamata CLIENTE FIDELIZZATO; gli attributi “numero di tessera” e “numero di punti accumulati” avranno cardinalità standard (1,1).

PARTIZIONAMENTO/ACCORPAMENTO DI ENTITA' E RELATIONSHIP

Innanzitutto è opportuno inserire una nuova entità, SCONTRINO, che detiene le informazioni “numero di scontrino”, “data scontrino” e “spesa totale”, in modo da facilitare la memorizzazione delle vendite, togliendo quindi gli attributi “numero di scontrino” e “data vendita” dalla relationship VENDITA.
SCONTRINO verrà collegato con l'entità PRODOTTO tramite una relationship PRESENZA e a CLIENTE FIDELIZZATO con una relationship SPESA. Possiamo dunque contrarre le tre relationship binarie VENDITA, PRESENZA e SPESA in un'unica relationship ternaria VENDITA, con cardinalità (0,N) su PRODOTTO, (1,N) su SCONTRINO e (0,N) su CLIENTE FIDELIZZATO.
Assumiamo che un prodotto può essere presente una sola volta su uno scontrino, quindi per indicare se in una stessa spesa sono state comprate una o più occorrenze dello stesso prodotto è necessario inserire l'attributo “quantità” alla relationship VENDITA.

SCELTA DEGLI IDENTIFICATORI PRIMARI

Gli identificatori primari sono già tutti stabiliti per le entità; le relationship binarie avranno una chiave composta dalle chiavi delle due entità alle quali sono collegati, mentre la relationship ternaria VENDITA avrà un identificatore primario composto solo dalla chiave di SCONTRINO e dalla chiave di PRODOTTO, visto che un prodotto può ovviamente essere comprato da un cliente non fidelizzato, e quindi la chiave di CLIENTE FIDELIZZATO verrà considerata un attributo semplice della relationship VENDITA con cardinalità (0,1) (assumerà valore nullo se il prodotto in questione è stato comprato da un cliente non fidelizzato).

ANALISI DELLE RIDONDANZE

- attributo “livello” a IMPIEGATO: è una ridondanza, perchè è deducibile dalla data di assunzione di un impiegato; per comodità si è deciso di mantenere tale attributo;
- attributo “stipendio”a IMPIEGATO: è una ridondanza, perchè è deducibile dal livello di un impiegato; anche in questo caso si è deciso di mantenere l'attributo dato che, come nel caso precedente, la dipendenza tra i due attributi potrebbe variare e in caso di mancanza di ridondanza il calcolo potrebbe essere impreciso;
- attributo “reparto” a IMPIEGATO: è una ridondanza, perchè si può vedere a che reparto appartiene un impiegato tramite la relationship APPARTENENZA; questa è sconveniente, perchè oltre allo svantaggio in termini di spazio, mantenendola ci sarebbe perdita di tempo nell'aggiornamento, che è maggiore dl guadagno di tempo che ci sarebbe per controllare a quale reparto appartiene un impiegato;
- attributo “reparto” a PRODOTTO: è una ridondanza del tutto analoga a quella vista in precedenza e in modo altrettanto analogo si valuta che è sconveniente;
- attributo “responsabile” a IMPIEGATO: è una ridondanza, perchè si può vedere se un impiegato è responsabile o no del reparto a cui appartiene grazie all'attributo “responsabile” di REPARTO; è sconveniente, e la sua sconvenienza è determinata dalla perdita di tempo che si avrebbe con l'aggiornamento in caso di cambiamento del responsabile di un reparto;
- attributo “spesa totale” a SCONTRINO: è una ridondanza, perchè si potrebbe ricavare accedendo a VENDITA e a PRODOTTO; sarebbe eccessivamente oneroso ricavarla in quest'ultimo modo, quindi si preferisce mantenere l'attributo.

SCHEMA E-R RISTRUTTURATO

Si veda il file “schema_ristrutturato.dia”.

TRADUZIONE AL MODELLO RELAZIONALE

IMPIEGATO(**codice fiscale,** data assunzione, nome, data di nascita, livello, stipendio, mansione, telefono, indirizzo, reparto);

REPARTO(**nome**, responsabile);

PRODOTTO(**codice interno,** nome, genere, prezzo, scadenza, quantità magazzino, soglia minima, numero punti che fornisce, numero punti per vincerlo, reparto);

FORNITORE(**partita iva,** ragione sociale, modalità di pagamento richiesta, fax, indirizzo, telefono);

CLIENTE FIDELIZZATO(**codice fiscale**, nome, indirizzo di residenza, città di residenza, numero tessera, numero punti tessera);

SCONTRINO(**numero**, spesa, data);

AFFERENZA(**impiegato, reparto**);

SUDDIVISIONE(**composto, componente**, quantità);

FORNITURA(prodotto, fornitore, **codice fornitura**);

DISPONIBILITA'(**fornitore, prodotto**, prezzo richiesto);

VENDITA(**prodotto, scontrino**, cliente, quantità);

RITIRO(**cliente, prodotto**);

NOTA 1:
Gli attributi in grassetto sono le chiavi primarie.

NOTA 2:
Nonostante l'entità IMPIEGATO abbia cardinalità (1,1) con la relationship AFFERENZA, non è possibile omettere la relationship; se la omettessimo avremmo che un attributo di IMPIEGATO ha un vincolo d'integrità referenziale con REPARTO e che un attributo di REPARTO ha un vincolo d'integrità referenziale con IMPIEGATO, e ciò non renderebbe possibile l'inserimento di un impiegato o di un reparto a meno di non inserire un reparto con il valore di “responsabile” settato a NULL, oppure a meno di non inserire un impiegato con il valore di un eventuale attributo “reparto” settato a NULL .
E' possibile invece contrarre la relationship APPARTENENZA.

VINCOLI D'INTEGRITA' REFERENZIALE

- “Reparto” in PRODOTTO è una chiave esterna con vincolo d'integrità referenziale verso REPARTO.nome.
- “Responsabile” in REPARTO è una chiave esterna con vincolo d'integrità referenziale verso l'attributo IMPIEGATO.codice fiscale.
- “Impiegato” in AFFERENZA è una chiave esterna con vincolo d'integrità referenziale verso IMPIEGATO.codice fiscale.
- “Reparto” in AFFERENZA è una chiave esterna con vincolo d'integrità referenziale

verso REPARTO.nome.

- “Composto” e “componente” in SUDDIVISIONE sono chiavi esterne con vincolo d'integrità referenziale verso PRODOTTO.codice interno.
- “Prodotto” in FORNITURA è una chiave esterna con vincolo d'integrità referenziale verso PRODOTTO.codice interno.
- “Prodotto” in DISPONIBILITA' è una chiave esterna con vincolo d'integrità referenziale verso PRODOTTO.codice interno.
- “Prodotto” in VENDITA è una chiave esterna con vincolo d'integrità referenziale verso PRODOTTO.codice interno.
- “Prodotto” in RITIRO è una chiave esterna con vincolo d'integrità referenziale verso PRODOTTO.codice interno.
- “Fornitore” in FORNITURA è una chiave esterna con vincolo d'integrità referenziale verso FORNITORE.partita iva.
- “Fornitore” in DISPONIBILITA' è una chiave esterna con vincolo d'integrità referenziale verso FORNITORE.partita iva.
- “Scontrino” in VENDITA è una chiave esterna con vincolo d'integrità referenziale verso SCONTRINO.numero.
- “Cliente” in VENDITA è una chiave esterna con vincolo d'integrità referenziale verso CLIENTE FIDELIZZATO.codice fiscale.
- “Cliente” in RITIRO è una chiave esterna con vincolo d'integrità referenziale verso CLIENTE FIDELIZZATO.codice fiscale.

IMPLEMENTAZIONE

DBMS utilizzato: FIREBIRD

OPERAZIONE 1

Creazione del database e delle relazioni

```
CREATE DATABASE 'D:\temp\Supermercato.fdb'
USER 'SYSDBA' PASSWORD 'masterkey';

CREATE TABLE Impiegato ( codice_fiscale CHAR(16), nome VARCHAR(30),
data_nascita TIMESTAMP, mansione VARCHAR(20), livello INTEGER, stipendio
DECIMAL, data_assunzione TIMESTAMP, indirizzo VARCHAR(20), telefono
VARCHAR(11),  PRIMARY KEY (codice_fiscale));

CREATE TABLE Reparto ( nome VARCHAR(20), responsabile CHAR(16), PRIMARY KEY
(nome), FOREIGN KEY (responsabile) REFERENCES Impiegato(codice_fiscale));

CREATE TABLE Prodotto ( codice_interno INTEGER, nome VARCHAR (30), genere
VARCHAR(20), prezzo DECIMAL, scadenza TIMESTAMP, quantita_magazzino INTEGER,
soglia_minima INTEGER, punti_forniti INTEGER, punti_per_vincerlo INTEGER, reparto
VARCHAR (20), PRIMARY KEY (codice_interno), FOREIGN KEY (reparto)
REFERENCES Reparto(nome));

CREATE TABLE Fornitore ( partita_iva CHAR(11), ragione_sociale VARCHAR(20),
modalita_pagamento VARCHAR(20), fax VARCHAR(11), indirizzo VARCHAR(20), telefono
VARCHAR(11), PRIMARY KEY (partita_iva));

CREATE TABLE Cliente_fidelizzato ( codice_fiscale CHAR(16), nome VARCHAR (30),
indirizzo VARCHAR(20), citta VARCHAR(20), numero_tessera INTEGER, punti_tessera
INTEGER, PRIMARY KEY (codice_fiscale));

CREATE TABLE Scontrino ( numero INTEGER, spesa DECIMAL, data TIMESTAMP,
PRIMARY KEY (numero));

CREATE TABLE Afferenza (impiegato CHAR(16), reparto VARCHAR (20), PRIMARY
KEY(impiegato, reparto), FOREIGN KEY (impiegato) REFERENCES Impiegato
(codice_fiscale), FOREIGN KEY (reparto) REFERENCES Reparto(nome));

CREATE TABLE Suddivisione (composto INTEGER, componente INTEGER, quantita
INTEGER, PRIMARY KEY (composto, componente), FOREIGN KEY (composto)
REFERENCES Prodotto(codice_interno), FOREIGN KEY (componente) REFERENCES
Prodotto (codice_interno));

CREATE TABLE Fornitura ( prodotto INTEGER, fornitore CHAR(11), codice_fornitura
INTEGER, PRIMARY KEY (codice_fornitura), FOREIGN KEY (prodotto) REFERENCES
Prodotto(codice_interno), FOREIGN KEY (fornitore) REFERENCES Fornitore
(partita_iva));

CREATE TABLE Disponibilita ( prodotto INTEGER, fornitore CHAR(11), prezzo_richiesto
```

INTEGER, PRIMARY KEY (prodotto, fornitore), FOREIGN KEY (prodotto) REFERENCES Prodotto(codice_interno), FOREIGN KEY (fornitore) REFERENCES Fornitore (partita_iva));

CREATE TABLE Vendita (prodotto INTEGER, scontrino INTEGER, quantita INTEGER, cliente CHAR(16), PRIMARY KEY (prodotto, scontrino), FOREIGN KEY (prodotto) REFERENCES Prodotto(codice_interno), FOREIGN KEY (scontrino) REFERENCES Scontrino(numero), FOREIGN KEY (cliente) REFERENCES Cliente_fidelizzato (codice_fiscale));

CREATE TABLE Ritiro (cliente CHAR(16), prodotto INTEGER, PRIMARY KEY (cliente, prodotto), FOREIGN KEY (cliente) REFERENCES Cliente_fidelizzato (codice_fiscale), FOREIGN KEY (prodotto) REFERENCES Prodotto (codice_interno));

CREATE EXCEPTION cambio_reparto 'impossibile assegnare il responsabile di un reparto ad un altro reparto';

OPERAZIONE 2

Inserimento di un impiegato:

```
set term !! ;

create procedure inserisci_impiegato
(codice_fiscale CHAR(16), nome VARCHAR(30), data_nascita TIMESTAMP, mansione VARCHAR(20), livello INTEGER, stipendio DECIMAL, data_assunzione TIMESTAMP, indirizzo VARCHAR(20), telefono INTEGER)
as
begin
  insert into Impiegato
   (codice_fiscale, nome, data_nascita, mansione, livello, stipendio, data_assunzione, indirizzo, telefono)
  values (:codice_fiscale, :nome, :data_nascita, :mansione, :livello, :stipendio, :data_assunzione, :indirizzo, :telefono);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_impiegato ('rssgsp74b14h245y', 'Giuseppe Rossi', 'FEB-14-1974','fruttivendolo',2, 600.0, 'FEB-14-1994', 'via_musone 16', 0854312119);

execute procedure inserisci_impiegato ('aaabbbcccgabghyu', 'john smith', 'FEB-15-1954', 'direttore', 10, 1400.0, 'MAR-15-1986', 'via pescara 14', 086234292);

execute procedure inserisci_impiegato ('aaabbbcccgabghyz', 'patty smith', 'FEB-16-1956', 'segretaria', 8, 1200.0, 'FEB-18-1988', 'via milano 18', 0844711788);

execute procedure inserisci_impiegato ('aaabbbcccgabghyk', 'stefano biondi', 'FEB-16-1934', 'magazziniere', 8, 1200.0, 'FEB-18-1988', 'via milano 19', 0844711799);
```

Inserimento di un reparto:

```
set term !! ;

create procedure inserisci_reparto
(nome VARCHAR(20), responsabile CHAR(16))
as
begin
  insert into Reparto
   (nome, responsabile)
  values (:nome, :responsabile);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_reparto ('ortofrutta', 'rssgsp74b14h245y');

execute procedure inserisci_reparto ('ortofrutta', 'rssgsp74b14h245z'); /* istruzione errata per verificare il vincolo d'integrita */

execute procedure inserisci_reparto ('direzione', 'aaabbbcccgabghyu');

execute procedure inserisci_reparto ('segreteria', 'aaabbbcccgabghyz');

execute procedure inserisci_reparto ('prodotticonfezionati', 'aaabbbcccgabghyk');
```

Inserimento di un fornitore:

```
set term !! ;

create procedure inserisci_fornitore
(partita_iva CHAR(11), ragione_sociale VARCHAR(20), modalita_pagamento VARCHAR(20), fax VARCHAR(11), indirizzo VARCHAR(20), telefono VARCHAR(11))
as
begin
  insert into Fornitore
   (partita_iva, ragione_sociale, modalita_pagamento, fax, indirizzo, telefono)
  values (:partita_iva, :ragione_sociale, :modalita_pagamento, :fax, :indirizzo, :telefono);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_fornitore ('32200928311', 'fornitore_slc', 'assegno', '085715444', 'via laquila 38', '08577888');

execute procedure inserisci_fornitore ('34400928311', 'fornitore_spa', 'bonifico', '03415444', 'via coppito 58', '08625558');
```

execute procedure inserisci_fornitore ('05400928311', 'fornitore_snc', 'contanti', '0341588444', 'via coppito 78', '08627558');

Inserimento di un prodotto:

```
set term !! ;

create procedure inserisci_prodotto
(codice_interno INTEGER, nome VARCHAR (30), genere VARCHAR(20), prezzo DECIMAL, scadenza TIMESTAMP, quantita_magazzino INTEGER, soglia_minima INTEGER, punti_forniti INTEGER, punti_per_vincerlo INTEGER, reparto  VARCHAR (20))
as
begin
  insert into Prodotto
   (codice_interno, nome, genere, prezzo, scadenza, quantita_magazzino, soglia_minima, punti_forniti, punti_per_vincerlo, reparto)
  values (:codice_interno, :nome, :genere, :prezzo, :scadenza, :quantita_magazzino, :soglia_minima, :punti_forniti, :punti_per_vincerlo, :reparto);
end!!

set term ; !!
```

Esempi di esecuzione:

execute procedure inserisci_prodotto (4758, 'biscotti abcd', 'dolciumi', 3.70, 'SEP-16-2012', 200, 20, 0, NULL, 'prodotticonfezionati');

execute procedure inserisci_prodotto (6473, 'arance bbbb', 'frutta', 2.10, 'MAR-15-2009', 100, 50, 0, NULL, 'ortofrutta');

execute procedure inserisci_prodotto (3749, 'cioccolatini abcd', 'dolciumi', 30.0, NULL, 10, 5, 0, 150, 'prodotticonfezionati');

execute procedure inserisci_prodotto (9505, 'cornetti abcd', 'dolciumi', 3.0, NULL, 30, 10, 5, NULL, 'prodotticonfezionati');

execute procedure inserisci_prodotto (9507, 'farina', 'farinacei', 0.99, 'SEP-16-2012' , 30, 5, 0, NULL, 'prodotticonfezionati');

execute procedure inserisci_prodotto (9508, 'uova', 'alimenti generici', 1.73, 'SEP-16-2009' , 90, 50, 1, NULL, 'prodotticonfezionati');

Inserimento di un cliente fidelizzato:

```
set term !! ;

create procedure inserisci_cliente
(codice_fiscale CHAR(16), nome VARCHAR (30), indirizzo VARCHAR(20), citta VARCHAR(20), numero_tessera INTEGER, punti_tessera INTEGER)
as
begin
  insert into Cliente_fidelizzato
```

```
   (codice_fiscale, nome, indirizzo, citta, numero_tessera, punti_tessera)
  values (:codice_fiscale, :nome, :indirizzo, :citta, :numero_tessera, :punti_tessera);
end!!
set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_cliente ('ajskdlddlsorugim', 'paolo bianchi', 'via_romana 49',
'roma', 4839, 0);

execute procedure inserisci_cliente ('asdfghjklpoiuytr', 'pietro marrone', 'via_quercia 7',
'roma', 8402, 0);

execute procedure inserisci_cliente ('alsjdi89f08p730n', 'giovanni verdi', 'via_giuli 23',
'roma', 5648, 0);

execute procedure inserisci_cliente ('aaaaaaaassssdddd', 'anna neri', 'via_torino 61',
'roma', 9304, 0);
```

Inserimento di uno scontrino:

```
set term !! ;

create procedure inserisci_scontrino
(numero INTEGER, spesa DECIMAL, data TIMESTAMP)
as
begin
  insert into Scontrino
   (numero, spesa, data)
  values (:numero, :spesa, :data);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_scontrino (3557, 123.46, 'FEB-11-2009 17:19');

execute procedure inserisci_scontrino (3558, 1.73, 'FEB-11-2009 17:25');
```

Inserimento di un'afferenza:

```
set term !! ;

create procedure inserisci_afferenza
(impiegato CHAR(16), reparto VARCHAR (20))
as
begin
  insert into Afferenza
   (impiegato, reparto)
  values (:impiegato, :reparto);
end!!
```

```
set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_afferenza ('rssgsp74b14h245y', 'ortofrutta');

execute procedure inserisci_afferenza ('aaabbbcccgabghyu', 'direzione');

execute procedure inserisci_afferenza ('aaabbbcccgabghyz', 'segreteria');

execute procedure inserisci_afferenza ('aaabbbcccgabghyk', 'prodotticonfezionati');
```

Inserimento di una fornitura:

```
set term !! ;

create procedure inserisci_fornitura
(prodotto INTEGER, fornitore CHAR(11), codice_fornitura INTEGER)
as
begin
  insert into Fornitura
   (prodotto, fornitore, codice_fornitura)
  values (:prodotto, :fornitore, :codice_fornitura);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_fornitura (9507, '32200928311',  787 );

execute procedure inserisci_fornitura (9508, '05400928311', 790 );

execute procedure inserisci_fornitura (6473, '34400928311', 791 );
```

Inserimento di una suddivisione:

```
set term !! ;

create procedure inserisci_suddivisione
(composto INTEGER, componente INTEGER, quantita INTEGER)
as
begin
  insert into Suddivisione
   (composto, componente, quantita)
  values (:composto, :componente, :quantita);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_suddivisione (4758, 9507, 40);

execute procedure inserisci_suddivisione (4758, 9508, 80);
```

Inserimento di una disponibilità:

```
set term !! ;

create procedure inserisci_disponibilita
(prodotto INTEGER, fornitore CHAR(11), prezzo_richiesto DECIMAL)
as
begin
  insert into Disponibilita
   (prodotto, fornitore, prezzo_richiesto)
  values (:prodotto, :fornitore, :prezzo_richiesto);
end!!

set term ; !!
```

Esempi di esecuzione:

```
execute procedure inserisci_disponibilita (9507, '32200928311', 0.5 );

execute procedure inserisci_disponibilita (9508, '32200928311', 0.2 );

execute procedure inserisci_disponibilita (6473, '34400928311', 1.0 );
```

Inserimento di una vendita:

```
set term !! ;

create procedure inserisci_vendita
(prodotto INTEGER, scontrino INTEGER, quantita INTEGER, cliente CHAR(16))
as
DECLARE VARIABLE punti_spesa INTEGER;
begin

  insert into Vendita
   (prodotto, scontrino, quantita, cliente)
  values (:prodotto, :scontrino, :quantita, :cliente);

SELECT punti_forniti FROM Prodotto WHERE codice_interno = :prodotto into
punti_spesa;

UPDATE Prodotto SET quantita_magazzino = (quantita_magazzino - :quantita);

UPDATE Cliente_fidelizzato SET punti_tessera = (punti_tessera + :punti_spesa);

end!!

set term ; !!
```

Esempi di esecuzione:

execute procedure inserisci_vendita (3749, 3557, 4, 'ajskdlddlsorugim');

execute procedure inserisci_vendita (9508, 3558, 1, NULL);

Inserimento di un ritiro di un premio da parte di un cliente fidelizzato:

```
set term !! ;

create procedure inserisci_ritiro
(cliente CHAR(16), prodotto INTEGER)
as
begin
  insert into Ritiro
   (cliente, prodotto)
  values (:cliente, :prodotto);
end!!

set term ; !!
```

Esempio di esecuzione:

execute procedure inserisci_ritiro ('ajskdlddlsorugim', 3749);

OPERAZIONE 3

Modifica del responsabile di un reparto:

```
set term !! ;

create procedure modifica_resp_reparto
(reparto VARCHAR (20), nuovo_resp CHAR (16))
as
begin
  UPDATE Reparto SET responsabile = :nuovo_resp WHERE nome = :reparto;

end!!

set term ; !!
```

Esempio di esecuzione:

execute procedure modifica_resp_reparto ('ortofrutta', 'aaabbbcccgabghyu');

Modifica dei dati di un impiegato:

```
set term !! ;
```

```
create procedure modifica_imp_cf
(impiegato CHAR(16), codice_fiscale CHAR(16)  )
as
begin

  UPDATE Impiegato  SET codice_fiscale = : codice_fiscale  WHERE codice_fiscale =
:impiegato;

end!!
set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_resp_reparto ('ortofrutta', 'aaabbbcccgabghyu' );


set term !! ;

create procedure modifica_imp_nome
(impiegato CHAR(16), nome VARCHAR(30))
as
begin

  UPDATE Impiegato  SET nome = :nome  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!


set term !! ;

create procedure modifica_imp_datanascita
(impiegato CHAR(16), data_nascita TIMESTAMP)
as
begin

  UPDATE Impiegato  SET data_nascita = :data_nascita  WHERE codice_fiscale =
:impiegato;

end!!

set term ; !!


set term !! ;

create procedure modifica_imp_mansione
(impiegato CHAR(16), mansione VARCHAR(20))
as
begin
```

```
  UPDATE Impiegato  SET mansione = :mansione  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!


set term !! ;

create procedure modifica_imp_stipendio
(impiegato CHAR(16), stipendio DECIMAL)
as
begin

  UPDATE Impiegato  SET stipendio = :stipendio  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!


set term !! ;

create procedure modifica_imp_data_ass
(impiegato CHAR(16), data_assunzione TIMESTAMP)
as
begin

  UPDATE Impiegato  SET data_assunzione = :data_assunzione  WHERE codice_fiscale
= :impiegato;

end!!

set term ; !!


set term !! ;

create procedure modifica_imp_indirizzo
(impiegato CHAR(16), indirizzo VARCHAR(20))
as
begin

  UPDATE Impiegato  SET indirizzo = :indirizzo  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!

set term !! ;

create procedure modifica_imp_telefono
```

```
(impiegato CHAR(16), telefono VARCHAR(11))
as
begin

  UPDATE Impiegato  SET telefono = :telefono  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!
```

OPERAZIONE 4

Determinazione delle vendite per un reparto in un particolare periodo:

```
SELECT Vendita.* FROM ((Scontrino JOIN Vendita ON Scontrino.numero =
Vendita.scontrino AND Scontrino.data >= data_inizio AND Scontrino.data <= data_fine)
JOIN Prodotto ON Vendita.prodotto = Prodotto.codice_interno AND Prodotto.reparto =
rep);
```

NOTA
data_inizio, data_fine e rep sono da specificare ogni volta che si vuole eseguire la query inserendovi valori reali.

Esempio:

```
SELECT Vendita.* FROM ((Scontrino JOIN Vendita ON Scontrino.numero =
Vendita.scontrino AND Scontrino.data >= 'FEB-16-1906' AND Scontrino.data <= 'FEB-16-
2009') JOIN Prodotto ON Vendita.prodotto = Prodotto.codice_interno AND
Prodotto.reparto = 'prodotticonfezionati');
```

OPERAZIONE 5

Determinazione dei prodotti più venduti in un determinato reparto:

```
SELECT codice_interno, SUM (quantita) FROM (Vendita JOIN Prodotto ON
Vendita.Prodotto = Prodotto.codice_interno AND Prodotto.reparto = rep) GROUP BY
codice_interno HAVING SUM (quantita) > 100;
```

NOTA
rep è una variabile da specificare con un valore reale quando si vuole eseguire la query.

Esempio:

```
SELECT codice_interno, SUM (quantita) FROM (Vendita JOIN Prodotto ON
Vendita.Prodotto = Prodotto.codice_interno AND Prodotto.reparto = 'prodotticonfezionati')
GROUP BY codice_interno HAVING SUM (quantita) > 100;
```

OPERAZIONE 6

Modifica del prezzo di un prodotto:

```
set term !! ;

create procedure modifica_prezzo_prodotto
(prodotto INTEGER, prezzo DECIMAL)
as
begin

UPDATE Prodotto SET prezzo = :prezzo WHERE codice_interno = :prodotto;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_prezzo_prodotto (6473, 4.99);
```

OPERAZIONE 7

Modifica dei dati riguardanti le scorte di prodotto disponibili:

```
set term !! ;

create procedure modifica_quantita_prodotto
(prodotto INTEGER, nuova_quantita INTEGER)
as
begin

UPDATE Prodotto SET quantita_magazzino = :nuova_quantita WHERE codice_interno
= :prodotto;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_quantita_prodotto(6473, 12);
```

OPERAZIONE 8

Per i prodotti “composti”, specifica degli “ingredienti” e delle quantità necessarie alla preparazione:

```
SELECT componente, quantita FROM Suddivisione WHERE composto = comp;
```

NOTA
comp deve essere specificato dandogli un valore reale ogni volta che si vuole eseguire la query.

Esempio:

SELECT componente, quantita FROM Suddivisione WHERE composto = 4758;

OPERAZIONE 9

Inclusione o esclusione di un prodotto dalla raccolta punti:

```
set term !! ;

create procedure inclusione_raccolta_punti
(prodotto INTEGER, punti INTEGER)
as
begin

UPDATE Prodotto SET punti_forniti = :punti WHERE codice_interno = :prodotto;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure inclusione_raccolta_punti ( 6473, 30);
```

Indicazione del numero di puntiraccolta forniti dal prodotto:

```
set term !! ;

create procedure punti_forniti
(prodotto INTEGER)
returns( punti INTEGER)
as
begin

SELECT punti_forniti FROM Prodotto WHERE codice_interno = :prodotto into :punti;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure punti_forniti (6473);
```

OPERAZIONE 10

Determinazione dei prodotti sotto scorta:

SELECT nome, codice_interno FROM Prodotto WHERE quantita_magazzino >= soglia_minima;

OPERAZIONE 11

Lista dei fornitori dai quali un determinato prodotto può essere acquistato, ordinati in base al prezzo richiesto:

SELECT fornitore FROM Disponibilita WHERE prodotto = prod ORDER BY prezzo_richiesto;

NOTA
prod è un valore che deve essere sostituito con un valore reale all'atto di eseguire la query.

Esempio:

SELECT fornitore FROM Disponibilita WHERE prodotto = 6473 ORDER BY prezzo_richiesto;

OPERAZIONE 12

Modifica del numero di punti necessari ad ottenere un determinato premio:

```
set term !! ;

create procedure modifica_punti_premio
(prodotto INTEGER, punti INTEGER)
as
begin

UPDATE Prodotto SET punti_per_vincerlo = :punti WHERE codice_interno = :prodotto;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_punti_premio ( 9508, 300);
```

OPERAZIONE 13

Verifica dei premi attualmente disponibili:

SELECT nome, codice_interno FROM Prodotto WHERE punti_per_vincerlo > 0 AND quantita_magazzino > 0;

OPERAZIONE 14

Inserimento/Modifica dei prodotti forniti da un fornitore:

```
set term !! ;

create procedure modifica_fornitura
(prod INTEGER, fornitore CHAR(11))
as
begin

UPDATE Fornitura SET fornitore = :fornitore WHERE prodotto = :prod;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_fornitura ( 6473,  '32200928311');
```

```
set term !! ;

create procedure modifica_disponibilita
(prod INTEGER, fornitore CHAR (11), prezzo_ric DECIMAL )
as
begin

UPDATE Disponibilita SET fornitore = :fornitore, prezzo_richiesto = :prezzo_ric WHERE
prodotto = :prod;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_disponibilita ( 6473,  '32200928311', 10);
```

OPERAZIONE 15

Modifica del reparto di assegnazione di un impiegato:

```
set term !! ;

create procedure modifica_imp_reparto
(impiegato CHAR(16), reparto VARCHAR (20)  )
as

DECLARE VARIABLE x CHAR(16);

begin

select responsabile from Reparto where nome = :reparto into :x;

IF( x <> :impiegato) THEN
BEGIN
   UPDATE Afferenza SET reparto = :reparto  WHERE impiegato = :impiegato;
END
 IF( x = :impiegato) THEN EXCEPTION cambio_reparto;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_imp_reparto ( 'aaabbbcccgabghyu', 'ortofrutta');
```

Modifica del livello di un impiegato:

```
set term !! ;

create procedure modifica_imp_livello
(impiegato CHAR(16), livello INTEGER)
as
begin

  UPDATE Impiegato  SET livello = :livello  WHERE codice_fiscale = :impiegato;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_imp_livello ( 'aaabbbcccgabghyu', 10);
```

OPERAZIONE 16

Modifica del numero di punti assegnati al cliente:

```
set term !! ;

create procedure modifica_punti_cliente
(cliente CHAR(16), punti INTEGER)
as
begin

  UPDATE Cliente_fidelizzato  SET punti_tessera= :punti  WHERE codice_fiscale =
:cliente;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure modifica_punti_cliente ( 'asdfghjklpoiuytr', 161);
```

OPERAZIONE 17

Determinazione dei premi a cui un cliente ha diritto:

```
SELECT Prodotto.nome, Prodotto.codice_interno FROM (Cliente_fidelizzato JOIN
Prodotto ON punti_tessera >= punti_per_vincerlo) WHERE codice_fiscale = cf;
```

NOTA
cf è una variabile da specificare con un valore reale quando si vuole eseguire la query.

Esempio:

```
SELECT Prodotto.nome, Prodotto.codice_interno FROM (Cliente_fidelizzato JOIN
Prodotto ON punti_tessera >= punti_per_vincerlo) WHERE codice_fiscale =
'asdfghjklpoiuytr';
```

OPERAZIONE 18

Ritiro di un premio da parte di un cliente:

```
set term !! ;

create procedure ritira_premio
(cliente CHAR(16), premio INTEGER)
as
DECLARE VARIABLE punti_totali INTEGER;
DECLARE VARIABLE punti_da_sottrarre INTEGER;
```

```
begin

SELECT punti_tessera FROM Cliente_fidelizzato WHERE codice_fiscale = :cliente
INTO :punti_totali;

SELECT punti_per_vincerlo FROM Prodotto WHERE codice_interno = :premio into
:punti_da_sottrarre;

  UPDATE Cliente_fidelizzato  SET punti_tessera= (:punti_totali - :punti_da_sottrarre)
WHERE codice_fiscale = :cliente;

 insert into Ritiro
   (cliente, prodotto)
  values (:cliente, :premio);

UPDATE Prodotto SET quantita_magazzino = (quantita_magazzino – 1) WHERE
codice_interno = :premio;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure ritira_premio ( 'asdfghjklpoiuytr', 3749);
```

OPERAZIONE 19

Determinazione dei prodotti più acquistati da un cliente:

```
SELECT prodotto, SUM (quantita) FROM Vendita WHERE Cliente = cf GROUP BY
prodotto HAVING SUM (quantita) > 20;
```

NOTA
cl è una variabile da specificare con un valore reale quando si vuole eseguire la query.

Esempio:

```
SELECT prodotto, SUM (quantita) FROM Vendita WHERE Cliente = 'ajskdlddlsorugim'
GROUP BY prodotto HAVING SUM (quantita) > 20;
```

OPERAZIONE 20

Determinazione della spesa totale effettuata da un cliente in un determinato periodo:

```
set term !! ;

create procedure spesa_totale
(cliente CHAR(16), data_inizio TIMESTAMP, data_fine TIMESTAMP )
returns(spesa_totale DECIMAL)
as
begin

SELECT SUM (spesa) FROM(Scontrino JOIN Vendita ON Scontrino.numero =
Vendita.scontrino AND cliente = :cliente AND data >= :data_inizio AND data <= :data_fine)
into :spesa_totale;

end!!

set term ; !!
```

Esempio di esecuzione:

```
execute procedure spesa_totale ( 'ajskdlddlsorugim', 'FEB-10-1901', 'FEB-14-2009');
```