

Fontanot Luca

Ingegneria Informatica **IN0501095**

Progetto Basi di Dati

Database per l'app del supermercato

Sommario del progetto

Introduzione e descrizione del progetto.....	4
Requisiti dell'applicazione	4
Azioni che bisogna svolgere sul database	4
Modello concettuale.....	4
Diagramma E-R.....	4
Dizionario dei dati	5
Dizionario delle relazioni	5
Tavola dei volumi	5
Analisi	6
Analisi ridondanze	6
Accorpamento Partizionamento di E-R.....	6
Analisi dei vincoli	6
Scelta degli identificatori	7
Struttura del database	7
Diagramma E-R ristrutturato	7
Schema logico	8
Creazione della struttura del database	9
Tabella utenti	9
Tabella categorie	10
Tabella prodotti	11
Tabella coupons	12
Tabella coupons utilizzati	13
Tabella degli ordini.....	14
Tabella dei prodotti degli ordini	15
Tabella della profilazione	16
Creazione delle viste	17
Statistiche dei coupon più utilizzati in un periodo.....	17
Fatturato generato dall'app nei vari mesi	17
Creazione delle procedure.....	18
Trovare gli utenti che rispettano una determinata profilazione	18
Elenco dei coupon utilizzati dall'utente	18
Aggiornamento della profilazione	18
Apertura di una ricevuta	18
Creazione di Triggers	19
Verifica della correttezza email.....	19
Verifica della scadenza del coupon	19
Verifica quantità dello sconto	19

Calcolo del Prezzo totale all'inserimento.....	19
Normalizzazione	20
Prima forma normale	20
Seconda forma normale	20
Terza forma normale	20

Introduzione e descrizione del progetto

Si vuole sviluppare un database per una catena di supermercati. L'obiettivo dell'azienda è di creare un'applicazione mobile, e hanno bisogno di un database per salvare e gestire i dati degli utenti.

Requisiti dell'applicazione

- Un utente per utilizzare l'applicazione deve creare un account fornendo nome cognome e la propria e-mail.
 - Con la e-mail il supermercato vuole inviare ai propri utenti delle newsletter in base ai prodotti che vengono acquistati.
- L'utente viene incentivato ad utilizzare l'applicazione fornendo dei coupons che permettono di avere degli sconti sui prodotti del catalogo.
 - I coupon possono essere utilizzati al massimo una volta per utente.
 - L'utente deve mostrare in cassa il proprio identificativo e riceverà lo sconto dei prodotti, e verrà salvata la ricevuta nell'applicazione.

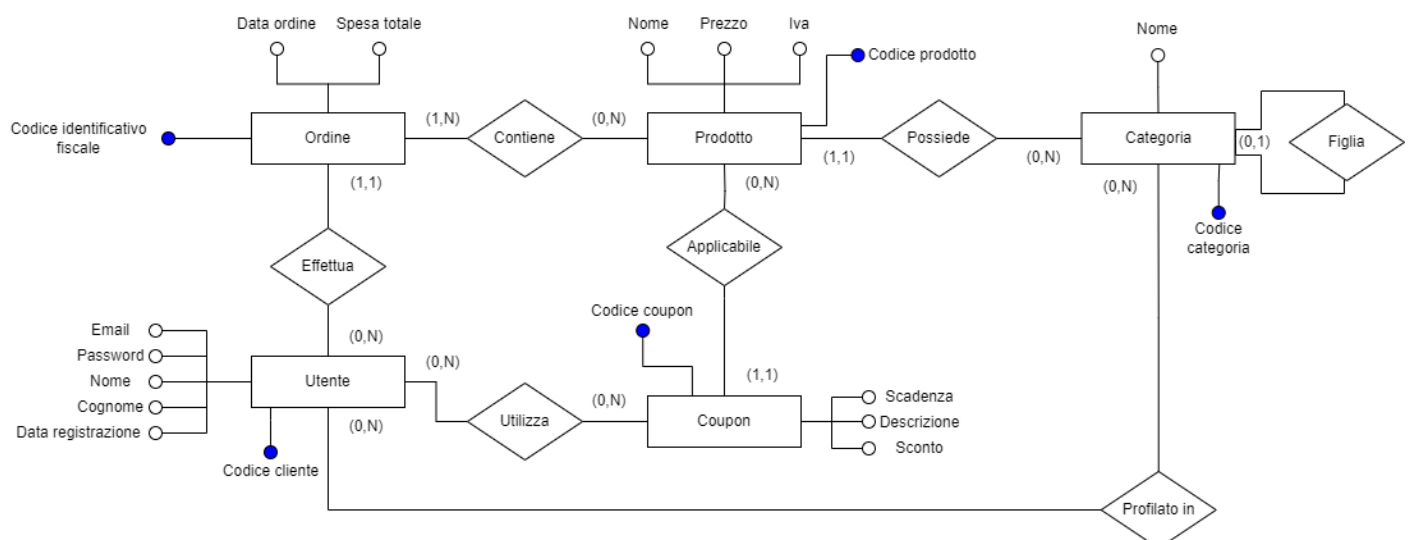
Azioni che bisogna svolgere sul database

Definiamo le operazioni di lettura e scrittura principali che vengono svolte sul database.

Operazione	Tipo	Frequenza
Creazione di un ordine	Interattiva	Ogni 10 minuti
Elenco dei coupon validi utilizzati dall'utente	Interattiva	Ogni giorno
Aggiornamento della profilazione	Batch	Ogni giorno
Trovare gli utenti che rispettano una determinata profilazione	Interattiva	Ogni 7 giorni
Statistiche dei coupon più utilizzati in un periodo	Interattiva	Ogni 10 giorni
Fatturato generato dall'app nel mese in corso	Interattiva	Ogni 10 giorni
Calcolo dell'importo speso in un ordine	Batch	Ogni nuovo scontrino
Visualizzazione di uno scontrino	Interattiva	Ogni ora
Login o registrazione di un utente	Interattiva	Ogni ora

Modello concettuale

Diagramma E-R



Dizionario dei dati

Nel dizionario dei dati definiamo le tabelle dei dati che la nostra applicazione ha bisogno.

Entità	Descrizione	Attributi	Identificatore
Utenti	Elenco degli utenti registrati	Codice cliente, E-mail, Password, Nome, Cognome, Data registrazione, Coupon usati	Codice utente
Coupons	Elenco dei coupon disponibili	Codice coupon, Codice prodotto associato, Descrizione coupon, Data di scadenza, Sconto	Codice coupon
Categorie prodotti	Elenco delle categorie dei prodotti	Codice categoria, Nome di categoria, Categoria padre	Codice categoria
Ordini	Elenco degli acquisti dell'utente	Codice identificativo fiscale, Codice utente, Spesa totale, Data ordine, Elenco prodotti ordinati	Codice identificativo fiscale, Codice utente
Prodotti	Elenco dei prodotti della catena	Codice prodotto, Nome, Prezzo unitario, Iva, Categoria prodotto	Codice prodotto

Dizionario delle relazioni

Il dizionario delle relazioni serve a definire le relazioni che ci saranno nel database e il loro senso pratico.

Relazione	Tipologia	Scopo
Utente → Effettua → Ordine	1 a molti	Il cliente può effettuare nessuno o N ordini
Ordine → Effettuato → Utente	1 a 1	Un ordine deve essere effettuato da uno e un solo cliente
Utente → Utilizza → Coupon	0 a molti	Ogni utente può aver utilizzato uno o più coupon
Coupon → Utilizzato → Utente	0 a molti	Ogni coupon può essere usato da uno o più utenti
Utente → Profilato → Categoria	0 a molti	Ogni utente viene collegato alle categorie alle quali presenta maggiore interesse
Categoria → Profilata → Utente	0 a molti	Ogni categoria si trova nella profilazione di uno o più utenti
Ordine → Contiene → Prodotti	1 a molti	Ogni ordine deve includere almeno un prodotto acquistato
Prodotti → Contenuti → Ordini	0 a molti	Ogni prodotto può trovarsi in uno o più ordini
Prodotto → Posseduto → Categoria	1 a 1	Ogni prodotto deve trovarsi in una categoria
Categoria → Possiede → Prodotti	0 a molti	Ogni categoria può avere uno o più prodotti
Coupon → Applicabile → Prodotto	1 a 1	Ogni coupon deve riferirsi ad un prodotto
Prodotto → Applicabili → Coupon	0 a molti	Ogni prodotto può avere uno o più coupon
Categoria → Figlia → Categoria	0 a 1	Ogni categoria può avere una categoria padre più generica

Tavola dei volumi

La tavola dei volumi contiene una stima della quantità dei dati presenti nel database. Dal volume possiamo poi fare delle decisioni se ci conviene o meno calcolare dei dati per delle determinate operazioni.

Concetto	Tipo	Volume
Utenti	E	10 000
Coupons	E	100
Categorie	E	100
Prodotti	E	Categorie * 20 = 2 000
Ordini	E	Utenti * 3 = 30 000
Utilizzo coupons	R	Ordini * 2 = 600 000
Profilazione	R	Utenti * 10 = 100 000
Prodotti negli ordini	R	Ordini * 10 = 600 000

Analisi

Analisi ridondanze

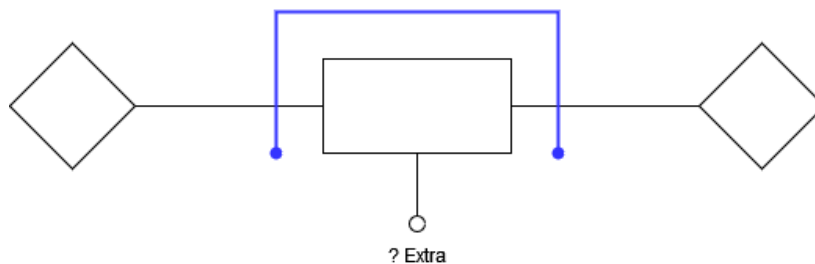
Analizzando il dizionario dei dati possiamo notare che il campo “Spesa totale” dell’ordine può essere calcolato partendo dai prezzi dei singoli prodotti. Ci sono però dei motivi per il quale è preferibile utilizzare un campo calcolato anziché calcolarlo ogni volta:

- Nel caso di una visualizzazione complessiva degli ordini e nel calcolo del fatturato abbiamo un risparmio non indifferente di calcolo. Per ogni prodotto dobbiamo tenere in considerazione: Prezzo, Iva e Sconto coupon. Siccome la tavola dei volumi ci dice che avremo oltre 600000 ordini nell’app, il calcolo risulta poco scalabile e un valore pre-calcolato di tipo numerico non impatta tanto la dimensione del nostro database.
- Il prezzo del prodotto può variare nel tempo, rendendo l’eventuale calcolo incorretto sugli ordini passati.

Accorpamento | Partizionamento di E-R

Nel diagramma E/R ci sono diversi casi in cui abbiamo delle relazioni di tipo $(x,N) - (x,N)$.

Per rappresentare queste relazioni nel database, considerando che bisogna anche effettuare la normalizzazione, conviene creare un’ulteriore entità con una chiave rappresentata dagli identificatori delle due entità all’estremità della relazione.



Le relazioni da partizionare sono:

- Ordine – Contiene – Prodotto
 - Come extra inseriamo la quantità di prodotti acquistati e l’eventuale coupon applicato.
- Utente – Utilizza – Coupon
 - Si aggiunge l’extra di “Data utilizzo” che servirà per calcolare i coupon più usati in un periodo.
- Utente – Profilato – Categoria
 - Si aggiunge un extra di “Data creazione” per far decadere le profilazioni vecchie che potrebbero non essere precise e corrette.

Analisi dei vincoli

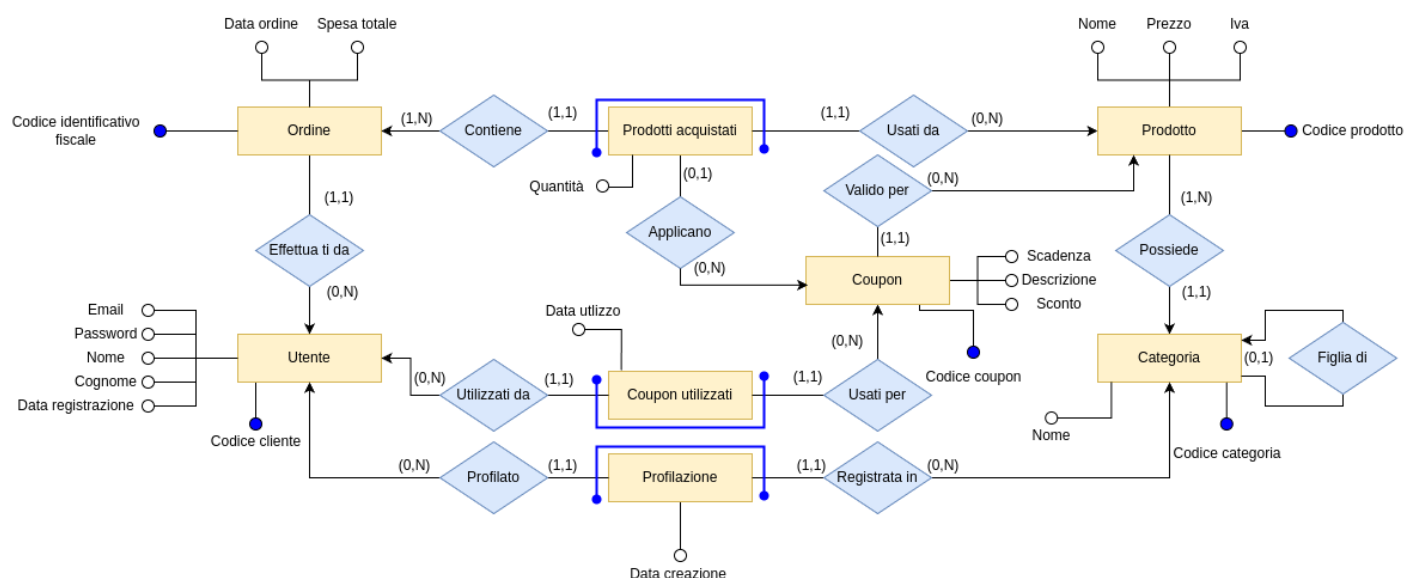
- L’indirizzo e-mail deve rispettare il formato standard:
 - Regex di verifica: `[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+\.[a-zA-Z]{2,6}`
 - Il significato del Regex è il seguente:
 - Ci devono essere dei caratteri alfanumerici (o i simboli - _ .).
 - Dopo i caratteri alfanumerici ci deve essere una @ seguita da altri caratteri alfanumerici.
 - Al termine della stringa ci deve essere un punto seguito da una serie di caratteri alfabetici al minimo di 2 e al massimo di 6 caratteri.
- La data di utilizzo del coupon deve essere antecedente alla data di scadenza del coupon stesso.
- Il valore dell’iva di un prodotto deve essere un numero compreso tra 0 e 100 (anche se in realtà le quote iva sono fisse a scaglioni, ma per praticità e scalabilità meglio lasciare il campo di inserimento libero).
- Lo sconto di un coupon non può superare il prezzo del prodotto a cui fa riferimento.

Scelta degli identificatori

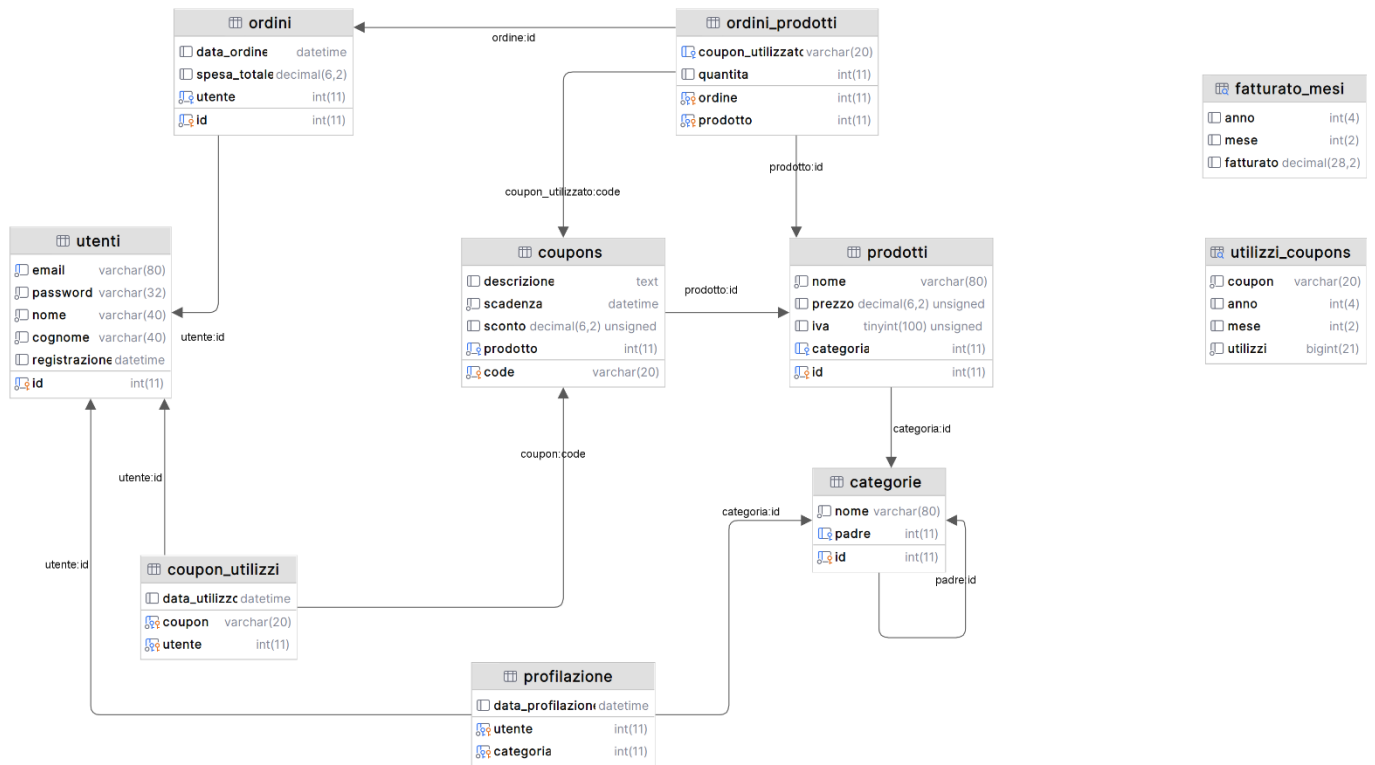
Entità	Identificatore	Tipo e note
Utente	Codice cliente	Tipo intero con l'auto incremento.
Ordine	Codice identificativo fiscale	Sarebbe il numero della ricevuta, normalmente si tratta di un codice che viene fornito quando lo scontrino viene registrato dal sistema fiscale. Se non viene fornito un identificativo fiscale vale un numero intero auto incrementato.
Prodotto	Codice prodotto	Anche noto come EAN, su tutti i prodotti venduti viene applicato un codice a barre numerico. Se il codice a barre non è disponibile si applica un numero intero auto incrementato.
Categoria	Codice categoria	Codice identificativo interno come numero intero auto incrementato.
Coupon	Codice coupon	Stringa alfanumerica identificativa del coupon, viene scelta alla creazione.

Struttura del database

Diagramma E-R ristrutturato



Schema logico



Creazione della struttura del database

Il codice SQL di creazione del database è riferito al file ./code/directives/Creazione_db.sql

Tabella utenti

La tabella utilizza **id** come chiave primaria e identificatore dell'utente. La e-mail inoltre è impostata come valore univoco, in quanto deve essere utilizzato per effettuare il login e si evita che più utenti siano associati alla stessa e-mail.

L'e-mail deve rispettare uno standard di formattazione: per farlo viene utilizzato un **TRIGGER**.

La data di **registrazione** è un campo che imposta automaticamente la data e ora della creazione dell'account. Lo scopo di questo campo è per puri fini statistici.

	id	email	password	nome	cognome	registrazione
1	1	f1deaafc0-7579-4a70-bf2b-5725aaf8382aFloriano1...	6af846a4b265ec0fb9a71af19cf30c9a	Costanzo	Cruciani	2024-05-24 12:17:11
2	2	197d4fa5-9dcf-444b-ae6-a1a8d6aab59fZoe.Datti...	3275f538740c98394904d9f32228e8e6	Glauco	Palmisani	2024-05-24 12:17:11
3	3	c8913e4c-31ed-4fca-acd9-c4e57a4b3b16Enea.Tava...	2b16188cd15cffb033c6e29a87420bf9	Floriano	De Palma	2024-05-24 12:17:11
4	4	e5672e0c-d1e8-4a4e-b851-b8c0831eef37Teodora44...	fd0a7777c9e3555bfd1faf25046c60af	Crescente	Vicinanza	2024-05-24 12:17:11
5	5	be285630-4971-45b2-a15b-6f1516cbd7c2Nino63@gm...	a318f3a61c4eb63afe11af2c294bf047	Pollione	Campanini	2024-05-24 12:17:11
6	6	bd19752d-12d9-4a62-9e0f-46e5485bf73cErmilo32@...	ccd4a43a41bddb88fce3044ae9613f59	Gioventino	Ferraris	2024-05-24 12:17:11
7	7	ba5003af-dc05-42a8-b50d-fde1e68ac010Ulpiano_T...	f286c44741638ae7ea12f5f25d870fd5	Alida	Maggiore	2024-05-24 12:17:11
8	8	02bfe932-e168-4cc3-9d16-9f095c338b22Andromeda...	38eb20d347431ef7ece6a83bb68a4789	Michela	Fantozzi	2024-05-24 12:17:11

Query utilizzate sulla tabella utenti:

Login utente

```
SELECT id FROM utenti WHERE email = ? AND password = MD5(?);
```

La query viene usata per autenticare l'utente, cercando la coppia **email-password** che corrisponde ad un record. La **password** è salvata come **MD5** per evitare la possibilità di decodificare la password in caso di esfiltrazione dei dati.

Inserimento utente

```
INSERT INTO utenti (email, password, nome, cognome) VALUES (?, MD5(?), ?, ?);
```

L'inserimento di un utente avviene senza specificare l'**id**, in quanto essendo un numero auto incrementato permette di ottenere automaticamente il primo numero disponibile.

Se la e-mail dell'utente risultasse già utilizzata, la query fallirebbe con errore di "Valore duplicato per il campo email", mentre se la mail risultasse scritta in un formato non valido, l'errore verrebbe lanciato dal **TRIGGER**.

*NOTE: Nel codice del progetto, durante la creazione dei dati, il calcolo **MD5** non viene effettuato con la funzione MD5() di SQL, in quanto viene effettuato un mass insert.*

Tabella categorie

Il campo **padre** fa riferimento all'identificatore **id** di un altro record categoria. Il padre è NULL qualora la categoria non sia figlia di nessuno. Il **padre** diventa automaticamente NULL in caso di cancellazione del record a cui fa riferimento.

id	nome	padre
11	Automotive	<null>
12	Electronics	<null>
13	Health	<null>
14	Industrial	<null>
15	Outdoors	6
16	Computers	<null>
17	Electronics	8
18	Beauty	<null>
19	Beauty	10
20	Toys	<null>

Note: Nel codice del progetto la seguente funzione elabora i dati ottenendo una struttura ad albero:

```
while(true){
  let atLeastOneChild = false;
  for (let i = 0; i < categorie.length; i++){
    if (categorie[i].padre !== null){
      atLeastOneChild = true;
      for (let j = 0; j < categorie.length; j++){
        if (categorie[j].id === categorie[i].padre){
          categorie[j].children.push(categorie[i])
          break;
        }
      }
      categorie.splice(i, 1)
      i--;
    }
  }
  if (!atLeastOneChild) break;
}
```

```
{
  "id":1,
  "title":"Padre",
  "children":[
    {
      "id":2,
      "title":"Figlio"
    }
  ]
}
```

Query utilizzate per la tabella categorie:

Lista categorie

```
SELECT * FROM categorie;
```

La creazione della vista ad albero deve poi essere fatta da un software grafico in quanto non è possibile una rappresentazione dalla forma tabellare.

Inserimento categoria

```
INSERT INTO categorie (nome, padre) VALUES (?, ?);
```

Come per gli utenti, non inseriamo l'identificatore perché verrà scelto in automatico.

Tabella prodotti

Il campo **id** dovrebbe rappresentare il codice a barre (EAN), se non avessimo il codice verrebbe utilizzato il prossimo valore disponibile. Questo però è pericoloso in quanto, se in futuro avessimo un prodotto che utilizza lo stesso codice EAN, si creerebbe un conflitto.

L'**iva** con il tipo **TINYINT(100) UNSIGNED** permette di rispettare il vincolo per il quale il valore deve essere compreso tra 0 e 100.

Nel campo **categoria** è possibile trovare valore NULL perché, se si dovesse cancellare una categoria, non vogliamo perdere tutti i prodotti. Questo comporta, però, una revisione manuale per associare una categoria al più presto ai prodotti vacanti.

Un'altra opzione sarebbe stata impostare una categoria di default e assegnare il valore in caso di cancellazione, ma di fatto il risultato è lo stesso.

	id	nome	prezzo	iva	categoria	
1	1	Recycled Fresh Keyboard	2.32	14		41
2	2	Recycled Metal Soap	1.23	3		5
3	3	Refined Metal Salad	2.49	17		37
4	4	Intelligent Rubber Shoes	2.62	6		36
5	5	Recycled Metal Bike	2.36	3		39
6	6	Electronic Granite Pants	1.06	17		13
7	7	Oriental Steel Chips	2.17	12		50
8	8	Bespoke Metal Bacon	2.20	10		9

Lista di query per la tabella dei prodotti:

Lista dei prodotti

```
SELECT * FROM prodotti LIMIT ? OFFSET ?;
```

Dalla tavola dei volumi è previsto che ci siano oltre 2000 prodotti nella tabella; pertanto, è buona pratica impostare un limite di prodotti e un offset in modo da visualizzare i prodotti per pagine.

Nel codice del progetto c'è anche l'opzione di ricerca dell'id, in tal caso la query diventa:

```
SELECT * FROM prodotti WHERE id LIKE ? LIMIT ? OFFSET ?;
```

L'utente inserisce un numero e questo viene utilizzato nella ricerca con %s%. Se la stringa di ricerca fosse vuota allora la scrittura `id LIKE '%%'` è una condizione "inutile" e non ha alcun effetto sul risultato.

Inserimento di un prodotto

```
INSERT INTO prodotti (id, nome, prezzo, iva, categoria) VALUES (?, ?, ?, ?, ?);
```

Tabella coupons

La chiave primaria è una stringa alfanumerica che deve essere scelta dal personale del supermercato e non deve essere uguale a nessun coupon creato in precedenza.

Nel caso in cui il prodotto associato al coupon dovesse venir cancellato, è stata impostata la regola **ON DELETE CASCADE**, la quale cancella anche il coupon.

Tra i vincoli viene richiesto di imporre che lo sconto non deve eccedere il costo del prodotto stesso. La regola si imposta attraverso un **TRIGGER**, il quale impedisce la creazione del coupon nel caso in cui il vincolo non fosse rispettato.

	code	descrizione	scadenza	sconto	prodotto
1	1360gxRDZZ	Unde texo conscendo nam sufficio vomer ocer.	2024-07-17 04:00:23	0.22	93
2	26vh19qhhV	Amo cuppedia est exercitationem clibanus clamo conscendo ...	2024-10-23 01:00:12	0.71	272
3	3LNICjaoTE	Tertius defessus delectus denique suppellex fugiat evenie...	2025-02-23 13:58:40	0.94	355
4	3r4NQ6s6Ka	Commodo cornu verumtamen corrigo villa viduo.	2024-06-25 07:40:14	0.86	340
5	4cn396sDPI	Placeat causa triduana.	2025-04-11 11:39:45	0.35	297
6	4hP1HpuwIM	Vilicus vinitor cibus thesis vindico crastinus comburo de...	2025-01-01 06:17:35	0.82	910
7	5kti6RadIQ	Totam adificio abbas.	2025-04-30 05:26:15	0.42	279
8	6FMXbSOPx8	Tabula corrumpo sto terra.	2024-11-03 20:52:11	0.35	576

Lista di query per la tabella dei coupons:

Lista dei coupons

```
SELECT * FROM coupons;
```

Nel codice del progetto, come per i prodotti, viene utilizzata una visuale a schede con la possibilità di ricerca del codice coupon:

```
SELECT * FROM coupons WHERE code LIKE ? LIMIT ? OFFSET ?;
```

Inserimento di un coupon

```
INSERT INTO coupons (code, descrizione, scadenza, sconto, prodotto) VALUES (?, ?, ?, ?, ?);
```




Tabella coupons utilizzati

Questa tabella mette in relazione i singoli coupons con i singoli utenti, più l'aggiunta del campo **data_utilizzo**. All'eliminazione del coupon o dell'utente anche tutte le relazioni associate sono eliminate in automatico.

La chiave primaria è la coppia **coupon utente**, questo garantisce il vincolo che ogni utente può utilizzare ogni coupon al massimo una volta.

Per impostare il vincolo sulla data di utilizzo dei coupons, la quale deve essere necessariamente antecedente alla data di scadenza del coupon, viene creato un [TRIGGER](#), il quale lancia un errore in caso di violazione.

Alla tabella dei coupons utilizzati è associata la vista [utilizzi_coupon](#) e la procedura [sp_couponUsati](#).

	 coupon ▼	÷  utente ▼	÷  data_utilizzo ▼	÷
1	1360gxRDZZ	931	2024-05-24 12:19:05	
2	1360gxRDZZ	2002	2024-05-24 12:18:17	
3	1360gxRDZZ	3360	2024-05-24 12:19:51	
4	1360gxRDZZ	4541	2024-05-24 12:18:30	
5	1360gxRDZZ	4927	2024-05-24 12:19:43	
6	1360gxRDZZ	5368	2024-05-24 12:18:35	
7	1360gxRDZZ	5678	2024-05-24 12:19:49	
8	1360gxRDZZ	5808	2024-05-24 12:18:05	

Lista di query per la tabella dei coupon utilizzati:





Inserimento di un utilizzo

```
INSERT INTO coupon_utilizzati (coupon,utente) VALUES (?,?);
```

Tabella degli ordini

In caso di eliminazione di un utente, verranno cancellati anche tutti gli ordini a lui associati.

I dati degli ordini vengono utilizzati nella vista [fatturato_mesi](#) e nella procedura [sp_ricevuta](#).

 id	 data_ordine	 spesa_totale	 utente
1	1 2024-01-23 09:55:39	19.82	2240
2	2 2024-03-31 11:42:01	5.56	2570
3	3 2023-09-19 07:37:18	16.19	969
4	4 2023-08-01 20:48:39	16.80	400
5	5 2023-12-17 22:00:19	13.21	5820
6	6 2023-10-27 05:49:26	11.27	519
7	7 2023-07-25 16:31:28	4.52	5293
8	8 2024-03-17 10:05:49	10.15	1239

Lista di query per la tabella degli ordini:

Lista degli ordini

```
SELECT * FROM ordini LIMIT ? OFFSET ?;
```

Dato che sono previsti oltre 30000 ordini, usiamo una visualizzazione in pagine

Creazione di un ordine

```
INSERT INTO ordini (id, data_ordine, spesa_totale, utente) VALUES (?, ?, ?, ?);
```

Tabella dei prodotti degli ordini

La tabella dei prodotti degli ordini mette in relazione un ordine con tutti i prodotti acquistati dall'utente. La chiave primaria è composta dalla coppia **ordine-prodotto**, in quanto per ogni ordine non si può avere un prodotto più volte. Un prodotto può essere preso in quantità maggiori di 1.

Alla tabella degli ordini è associato un [TRIGGER](#) che aggiorna la **spesa_totale** dell'ordine ogni volta che un nuovo prodotto viene aggiunto.

Se si utilizza un coupon si riporta il codice nel campo **coupon_utilizzato**, altrimenti con il valore NULL si indica che non è stato applicato nessuno sconto.

	ordine	prodotto	coupon_utilizzato	quantita
1	15342	576	ZUUDw2LfjQ	8
2	2511	576	ZUUDw2LfjQ	4
3	2326	576	ZUUDw2LfjQ	10
4	14364	844	ZGrWX7RDnh	5
5	14132	844	ZGrWX7RDnh	6
6	14030	844	ZGrWX7RDnh	9
7	13545	844	ZGrWX7RDnh	10
8	13272	844	ZGrWX7RDnh	1

Lista di query per la tabella dei prodotti degli ordini:

Inserimento di un prodotto nell'ordine

```
INSERT INTO ordini_prodotti (ordine, prodotto, coupon_utilizzato, quantita) VALUES  
(?, ?, ?, ?);
```

Tabella della profilazione




La tabella della profilazione contiene le relazioni tra gli utenti e le categorie di prodotti che hanno acquistato recentemente. Come per le tabelle precedenti, alla cancellazione di una categoria o di un utente, tutte le profilazioni associate vengono eliminate.

Siccome la profilazione non è una tabella utile per il normale funzionamento dell'app, l'inserimento dei dati viene effettuato esclusivamente dalla procedura [sp_updateProfilation](#), la quale deve essere chiamata in maniera programmata ogni giorno nel momento con meno traffico per evitare rallentamenti.

```
CALL sp_updateProfilation();
```

Quando si vuole inviare una newsletter per una categoria di prodotti si utilizza [sp_ottieniUtentiProfilazione](#) che fornisce all'amministratore la lista degli utenti interessati ad una categoria.

```
CALL sp_ottieniUtentiProfilazione(?);
```

	 utente ▼	÷  categoria ▼	÷  data_profilazione ▼
1	51	2	2024-05-24 12:19:56
2	51	10	2024-05-24 12:19:56
3	51	18	2024-05-24 12:19:56
4	51	22	2024-05-24 12:19:56
5	131	36	2024-05-24 12:19:56
6	241	12	2024-05-24 12:19:56
7	241	16	2024-05-24 12:19:56

Creazione delle viste

Il codice SQL di creazione delle viste è riferito al file ./code/directives/Creazione_viste.sql

Statistiche dei coupon più utilizzati in un periodo

Questa vista permette di analizzare per ogni periodo temporale la quantità di coupon utilizzati per ogni coupon.

	🔍 coupon ⌵	÷	📅 anno ⌵	÷	📅 mese ⌵	÷	🔍 utilizzi ⌵	÷
1	cVbPcpP1fW		2024		5		10	
2	Q4sjaZA10V		2024		5		9	
3	fcJjiw0u3P		2024		5		5	
4	TcZSqZAW2U		2024		5		9	
5	hjeUf5HriF		2024		5		10	
6	uwW1DTruqZ		2024		5		8	
7	4cn396sDPI		2024		5		15	
8	IU5g5Nn3Wf		2024		5		12	
9	x5MRHFnxWp		2024		5		10	
10	7ejyohC76Q		2024		5		7	
11	LAaDnrI9Xf		2024		5		9	
12	ZUUDw2LfjQ		2024		5		3	
13	bHgJHH2H0k		2024		5		9	

Fatturato generato dall'app nei vari mesi

Questa vista permette di analizzare la quantità di incassi per ogni mese, sommando i campi di spesa totale degli ordini e raggruppandoli in mese-anno rispetto alla data dell'ordine.

	📅 anno ⌵	÷	📅 mese ⌵	÷	🔍 fatturato ⌵	÷
1	2024		5		13252.90	
2	2024		4		18716.66	
3	2024		3		18542.17	
4	2024		2		16241.51	
5	2024		1		19193.63	
6	2023		12		19465.11	
7	2023		11		17522.37	
8	2023		10		18571.60	
9	2023		9		18177.58	
10	2023		8		18793.63	
11	2023		7		18618.39	
12	2023		6		18110.09	
13	2023		5		3734.97	

Creazione delle procedure

Il codice SQL di creazione delle viste è riferito al file `./code/directives/Creazione_procedure.sql`

Trovare gli utenti che rispettano una determinata profilazione

Fornendo alla procedura l'id della categoria, verrà fornito l'elenco degli utenti e le loro rispettive e-mail per inviare la newsletter.

	☐ utente ▼	÷ ☐ email ▼	÷ ☐ data ▼
1	51	6f28a444-5a91-4ba8-8e99-98d98455a8eaGiorgia_Costantin70@yahoo.it	2024-05-24 12:19:56
2	1022	1e1dae24-3b84-4f15-9cf1-e1cf8249336cDiana_Consiglio@libero.it	2024-05-24 12:19:56
3	1287	0f48f5d8-794a-4c76-9da4-478d47345846Michele_Zedda89@libero.it	2024-05-24 12:19:56
4	2218	1a57a780-4592-4314-a1d6-4582a55e37b2Siria.Bongiovanni74@gmail.com	2024-05-24 12:19:56
5	3634	d34b06eb-7801-48de-918e-58016ab4de6eLuminosa.Piva@gmail.com	2024-05-24 12:19:56
6	4309	b3672002-ec33-4054-97bb-2aa0a24d0e22Matteo70@hotmail.com	2024-05-24 12:19:56
7	5263	b2eaaa1b-6252-4686-a7d1-4583a27db10dRegolo_Fiorentino99@libero.it	2024-05-24 12:19:56
8	5358	d807a266-5c27-40a9-aea7-81393f41d3f4Giasone75@yahoo.it	2024-05-24 12:19:56

Per chiamare la procedura si utilizza:

```
CALL sp_ottieniUtentiProfilazione(id);
```

Elenco dei coupon utilizzati dall'utente

La procedura richiede l'ID dell'utente, e in output fornisce l'elenco dei coupon utilizzati.

	☐ coupon ▼
1	13G0gxRDZZ
2	EfZctKLM6P

Per chiamare la procedura si utilizza:

```
CALL sp_couponUsati(id);
```

Aggiornamento della profilazione

La procedura di aggiornamento della profilazione deve essere chiamata dal codice una volta al giorno, nel momento di traffico meno intenso. Non fornisce nessun output e si occupa solo di aggiornare la tabella **profilazione**.

Apertura di una ricevuta

Questa procedura riceve in input l'ID di un ordine, e ritorna la lista di prodotti acquistati, aggiungendo, se usato, il coupon e lo sconto che è stato scalato dall'ordine. Se il coupon è NULL significa che il prezzo non è alterato da nessuno sconto.

	☐ prodotto ▼	÷ ☐ nome ▼	÷ ☐ prezzo ▼	÷ ☐ iva ▼	÷ ☐ quantita ▼	÷ ☐ sconto ▼	÷ ☐ coupon ▼
1	44	Generic Concrete Bacon	2.74	1	6	<null>	<null>
2	408	Fantastic Fresh Hat	1.72	14	7	<null>	<null>
3	576	Unbranded Steel Towels	1.72	13	4	0.11	ZUUDw2LfjQ
4	810	Gorgeous Cotton Pants	1.27	6	4	<null>	<null>
5	903	Practical Concrete Pants	1.71	17	5	<null>	<null>
6	913	Sleek Cotton Table	2.65	9	3	<null>	<null>

Per utilizzare la procedura si utilizza:

```
CALL sp_ricevuta(id);
```

Creazione di Triggers

Il codice SQL di creazione delle viste è riferito al file `./code/directives/Creazione_triggers.sql`

Verifica della correttezza email

Questo controllo, effettuato prima dell'inserimento, utilizza una regular expression per verificare se la stringa della mail rispetta il pattern un indirizzo standard. Se non trova il pattern, quindi se ci sono 0 conferme, viene inviato il segnale custom 10001 con il messaggio "Email non valida", e l'inserimento del record non viene effettuato.

Verifica della scadenza del coupon

Il trigger viene utilizzato prima dell'inserimento di un coupon utilizzato, siccome il campo **data_utilizzo** viene impostata automaticamente come il `CURRENT_TIMESTAMP` (a meno di casi speciali), se la data è maggiore della data di scadenza allora l'errore 10002 con il messaggio "Coupon scaduto" impedisce l'inserimento del record.

Verifica quantità dello sconto

Ultimo trigger di verifica, il quale controlla se il prezzo del coupon è maggiore del prezzo del prodotto. In caso il vincolo non sia rispettato viene inviato l'errore 10003 con il messaggio "Coupon con sconto troppo alto" impedendo l'inserimento.

Calcolo del Prezzo totale all'inserimento

Lo scopo di questo trigger è di essere eseguito dopo l'inserimento di un prodotto dell'ordine. Viene verificato costo e sconto del prodotto e lo si aggiunge al valore della spesa totale dell'ordine con il seguente calcolo:

$$S = S + q * (\text{prezzo} - \text{sconto}) \frac{1 + \text{vat}}{100}$$

Normalizzazione

Prima forma normale

La prima forma normale richiede che tutte le colonne devono essere atomiche:

- Dobbiamo avere massimo una colonna per valore
- Bisogna evitare unità ripetitive

Il database è in prima forma normale dopo la procedura “ACCORPAMENTO | PARTIZIONAMENTO”, la quale ha reso tutte le colonne atomiche.

Seconda forma normale

Una tabella è in seconda forma normale se è in prima forma normale e se ciascuna colonna dipende (in senso stretto) dalla primary key.

Dato che le viste non contano sono considerate nella normalizzazione, il database rispetta la regola ed è in seconda forma normale.

Terza forma normale

Una tabella è in terza forma normale se è in seconda forma normale e se ogni attributo dipende solo dalla chiave primaria:

- Tutte le colonne sono indipendenti tra loro
- Non ci sono colonne calcolate

Il nostro database non è in terza forma normale, in quanto il campo **spesa_totale** della tabella **ordini** è un campo calcolato.