

# Progetto di Basi di Dati

Cosergate - Gestionale per le spese di casa

Anno Accademico 2012/2013

Riccardo Serafini, Luca Regazzi, Dario Serra, Daniele Sciuto

## **1. Raccolta delle specifiche della realtà di interesse espresse in linguaggio naturale, documentata.**

In tutte le case universitarie, durante la settimana i componenti dell'abitazione vanno a fare la spesa.

Non tutti comprano le stesse cose e non tutti consumano le stesse cose.

Nessuno vuole spendere più del necessario, soprattutto gli studenti universitari.

La questione della divisione delle spese, nelle case universitarie, è spesso causa di litigi.

Per risolvere tale inconveniente ci si organizza in modi diversi: c'è chi decide di fare tutta la spesa in comune e cucinare insieme oppure c'è chi decide di organizzarsi autonomamente in modo tale che ognuno compri i propri alimenti e se li cucini.

In entrambi i casi, comunque, compaiono delle voci di spesa che riguardano tutti i componenti dell'abitazione: bollette, prodotti per la pulizia, e in generale tutti i beni di uso comune.

La condivisione di tali beni comporta un periodico rituale di raccolta degli scontrini (spesso dimenticati nei portafogli), trascrizione dei costi e calcolo delle percentuali per definire chi deve soldi a chi.

Secondo noi è molto più comodo affidare la gestione della contabilità casalinga ad un programma. In questo modo si risolvono i problemi descritti e si può beneficiare di ulteriori vantaggi.

Ogni volta che un membro dell'abitazione va a fare la spesa, inserisce nel sistema i prodotti che ha acquistato.

Per ogni prodotto, specifica una breve descrizione, il prezzo, e quali coinquilini lo utilizzeranno.

Il sistema calcola automaticamente quanti soldi ognuno deve restituire agli altri. Essendo questa procedura così immediata, si può semplicemente evitare di scambiare il denaro ogni volta che si fa la spesa. Visto che di solito si abita con le stesse persone almeno per un anno, si può lasciare il conto "aperto" e restituirsi i soldi anche solo una volta all'anno, o semplicemente quando i debiti superano una soglia ragionevole.

Organizzandosi in modo efficace, ruotando le persone che vanno a fare la spesa, si possono tenere basse le differenze tra debiti e crediti.

Se si riesce a tenere questa soglia sotto un certo limite, si può anche evitare del tutto di scambiarsi il denaro.

Oltre a semplificare la procedura descritta in precedenza, l'utilizzo di questo sistema, può introdurre anche altri vantaggi.

Il programma registra lo storico degli acquisti, questo può essere molto utile per calcolare e presentare delle statistiche sulle spese casalinghe:

- Quanto si spende di media ogni settimana, ogni mese, ogni anno per la casa
- Quali prodotti si comprano più spesso
- Chi va a fare la spesa più spesso
- Monitorare la variazione di prezzo dei prodotti

L'afflusso di una grande quantità di dati relativi ai prodotti venduti nei supermercati, può inoltre rappresentare una risorsa per lo sviluppo di ulteriori funzionalità ed applicazioni.

Associando ad ogni prodotto il suo codice a barre, si può creare con il tempo un database unico di articoli, associando ad ognuno i diversi prezzi proposti da ogni supermercato.

Si potrebbe anche creare una tabella dei supermercati georeferenziata, in modo da mostrare su una mappa quelli più vicini a casa propria, e quelli che offrono prezzi più competitivi.

La raccolta di questi dati potrebbe essere molto facilitata dall'utilizzo di applicazioni per smartphone. Con la fotocamera si registrano i codici a barre dei prodotti nel momento in cui vengono inseriti nel carrello. In questo modo non c'è bisogno di riportare al computer le spese effettuate una volta arrivati a casa.

Il GPS del cellulare, inoltre, può tenere traccia del supermercato in cui ci si trova, e se il prodotto che si sta acquistando è disponibile nelle vicinanze ad un prezzo minore, l'utente può essere avvisato con una notifica.

Il sistema potrebbe aiutare anche le casalinghe a tenere traccia dei prodotti disponibili nella dispensa. Associando la data di scadenza ai generi alimentari, il sistema potrebbe notificare quando un prodotto sta per scadere.

Tutte queste funzionalità sono sicuramente ambiziose, ma se il prodotto riesce a raggiungere un discreto numero di utenti, sono certamente realizzabili.

Naturalmente nell'ambito del progetto di Basi di Dati non avremo la possibilità di sviluppare tutte queste possibilità, ma la nostra fase di Analisi dei Requisiti vuole essere più ampia e completa possibile per facilitare future espansioni.

## **2. Analisi dei Requisiti**

Si vuole realizzare una base di dati per gestire le spese all'interno di una abitazione.

La spesa è identificata dallo scontrino, in cui è riportata la data e l'ora dell'acquisto, il nome del negozio e a volte anche il nome del cliente, se utilizza una tessera socio.

Ad ogni prodotto inserito nel database viene associato un nome, una quantità ed un prezzo, inoltre viene anche associato ai membri dell'abitazione che lo utilizzeranno e quindi concorreranno all'acquisto.

Ogni abitante è identificato da un'email, un nome utente, una password, nome e cognome e dalla casa in cui abita.

La casa è identificata dalla città e dall'indirizzo, e da un nome identificativo. Il sistema deve tenere conto della restituzione del denaro tra i componenti dell'abitazione, registrando la data della restituzione, gli abitanti coinvolti e l'importo restituito.

Ad ogni prodotto possono essere aggiunti dei commenti da parte degli utenti, identificati da un contenuto e da una data di inserimento.

Una volta inserite le spese, il sistema dovrà calcolare quanti soldi ogni inquilino deve restituire agli altri.

### **Specifica delle operazioni**

Il sistema deve permettere, e tenere traccia, delle seguenti operazioni: (il numero di operazioni si riferisce ad un singolo ambiente)

- Creazione/eliminazione di nuovi utenti (solo la prima volta)
- Creazione/eliminazione di nuovi ambienti (solo la prima volta)
- Inserimento/rimozione di un utente da uno o più ambienti (raramente)
- Inserimento/modifica/rimozione di una nuova spesa (2 volte a settimana)
- Inserimento/modifica/rimozione dei prodotti relativi ad una spesa (20 volte a settimana)
- Restituzione di denaro tra i componenti della casa (1 volta al mese)
- Commento ad un prodotto (1 volta al mese)

### **3. Progettazione concettuale della base di dati con la produzione di uno schema Entity-Relationship (ER), che modelli la realtà di interesse**

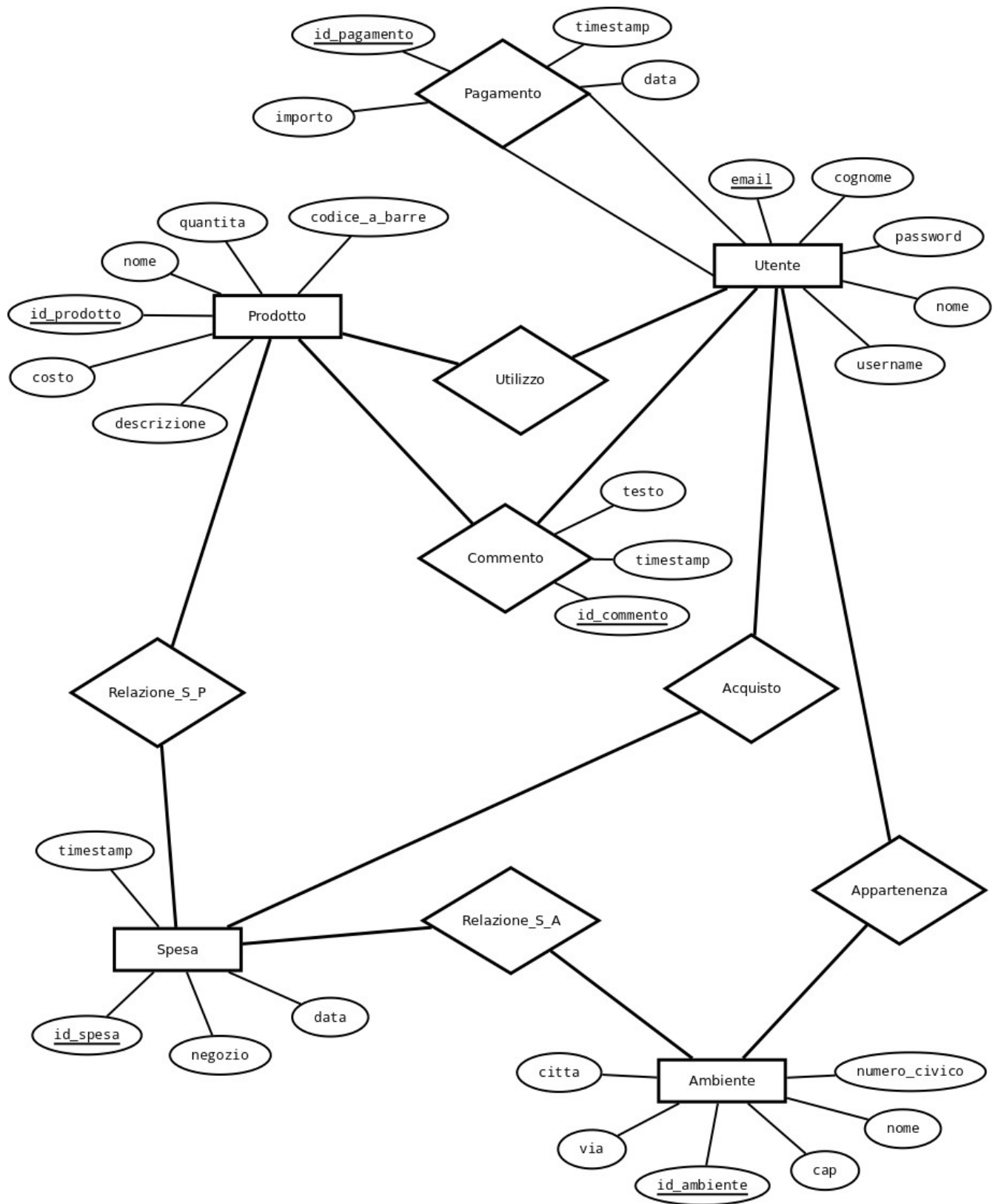
Analizzando la realtà di interesse abbiamo isolato diverse entità:

- Prodotto
- Utente
- Spesa
- Ambiente (casa)

Abbiamo inoltre riscontrato la necessità delle seguenti relazioni con attributi:

- Pagamento
- Commento

Da queste considerazioni abbiamo ricavato il seguente schema E/R:



## Dizionario dei dati

Entità:

Nome entità	Descrizione	Attributi	Identificatore
Utente	Utilizzatore del sistema, inquilino	nome (stringa) cognome (stringa) password (stringa) username (stringa)	Email(stringa)
Prodotto	Il singolo prodotto acquistato in una spesa (riga di uno scontrino)	nome (stringa) quantita (intero) descrizione (stringa) costo (decimale) codice a barre (intero)	Id_spesa(intero)
Spesa	Una spesa effettuata da un utente per la casa, identificata da uno scontrino	negozio (stringa) data (data) timestamp (data)	id_spesa (intero)

<b>Nome entità</b>	<b>Descrizione</b>	<b>Attributi</b>	<b>Identificatore</b>
Ambiente	L'abitazione per la quale sono effettuate le spese (può anche non essere una casa)	città (stringa) via (stringa) numero civico (stringa) cap (intero) nome (stringa)	id_ambiente (intero)

Relazioni:

<b>Nome relazione</b>	<b>Descrizione</b>	<b>Entità coinvolte</b>	<b>Attributi</b>
Pagamento	La restituzione di soldi tra due utenti	(ricorsiva) Utente (1,N)	id_pagamento (intero) timestamp (data) importo (decimale) data (data)
Commento	Un commento effettuato da un utente ad un prodotto	Prodotto (1,N) Utente (1,N)	id_commento (intero) timestamp (data) testo (stringa)
Relazione_S_A	Relazione per associare ogni spesa ad un ambiente	Spesa (1,1) Ambiente (1,N)	
Relazione_S_P	Relazione per associare ogni prodotto alla propria spesa	Prodotto (1,N) Spesa (1,1)	
Appartenenza	Relazione per associare ogni utente ai propri ambienti	Utente (1,N) Ambiente (1,N)	
Acquisto	Relazione per associare ogni spesa all'utente che l'ha effettuata	Utente (1,N) Spesa (1,N)	
Utilizzo	Relazione per associare ogni prodotto all'utente che lo utilizza e partecipa all'acquisto	Prodotto (1,N) Utente (1,N)	

## 4. Progettazione logica

### Ristrutturazione dello schema concettuale

Eliminazione delle ridondanze:

non abbiamo riscontrato ridondanze nello schema concettuale, e non riteniamo opportuno introdurne.

Eliminazione delle gerarchie:

non abbiamo riscontrato gerarchie nello schema concettuale.

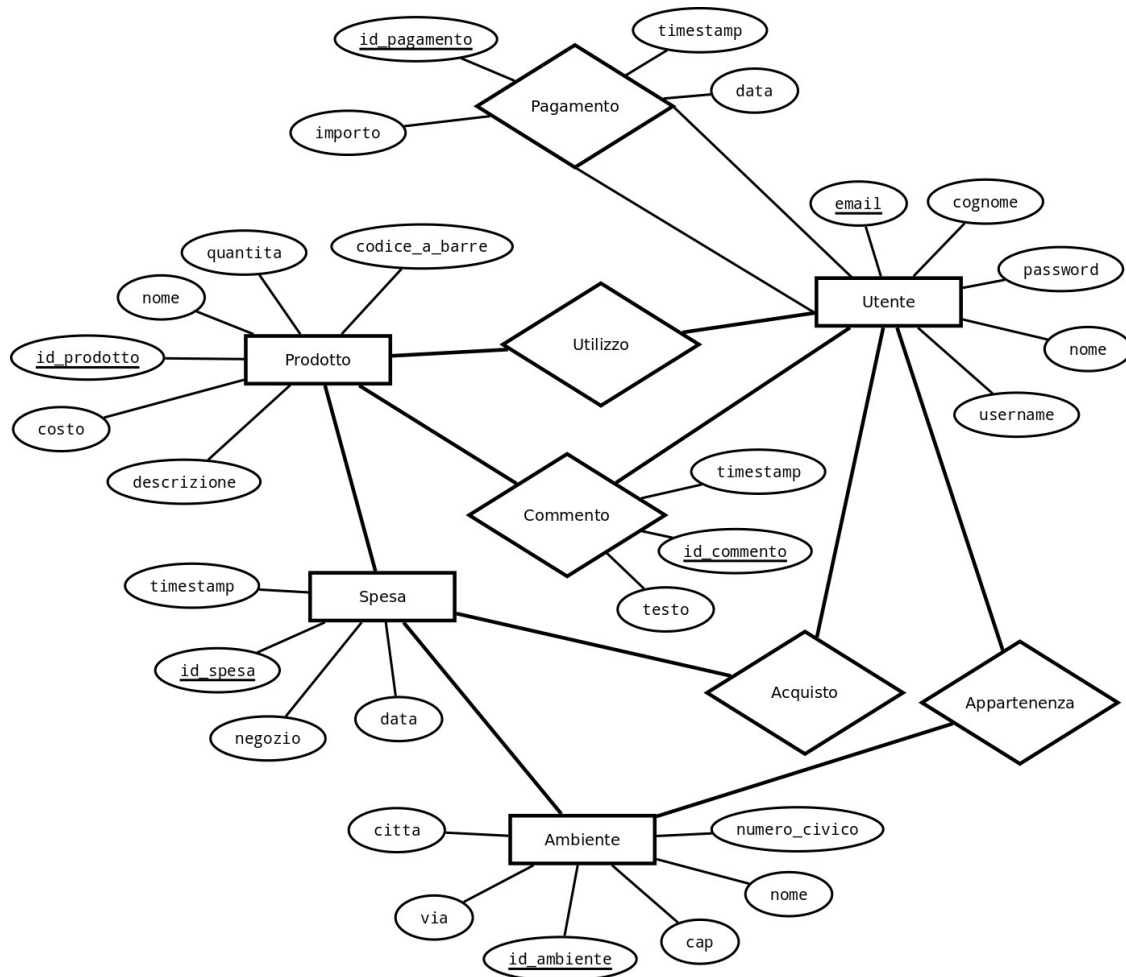
Accorpamenti:

Si nota che le Relazioni S\_A e S\_P (spesa ambiente e spesa prodotto) possono essere accorpate a Spesa, in quanto sono relazioni (1,1)(1,N).

Basta semplicemente aggiungere a Spesa le chiavi esterne di Prodotto e Ambiente.

In questo modo tra Spesa e Ambiente, e tra Spesa e Prodotto ci saranno relazioni (1,N).

Ecco come appare lo schema E/R finale:



## Tavola dei volumi e delle operazioni

Concetto	Tipo	Volume
Prodotto	E	580 x N x anno
Spesa	E	60 X N X anno
Ambiente	E	N
Pagamento	R	20 x N x anno
Commento	R	100 x N x anno
Appartenenza	R	4 x N
Acquisto	R	60 x N x anno
Utilizzo	R	1000 x N x anno

## Elenco degli identificatori principali:

Nome entità	Attributo chiave principale
Ambiente	id_ambiente (intero)
Utente	email (stringa)
Prodotto	id_prodotto (intero)
Spesa	id_spesa (intero)

## Traduzione dello schema concettuale nello schema relazionale

Entità-Relazione	Traduzione	Vincolo di integrità referenziale
Ambiente	ambiente( <u>id_ambiente</u> , citta, via, cap, numero_civico, nome)	
Utente	utente( <u>email</u> , nome, cognome, password, username)	
Prodotto	prodotto( <u>id_prodotto</u> , nome, quantita, costo, descrizione, codice_a_barre, spesa)	spesa → spesa.id_spesa
Spesa	spesa( <u>id_spesa</u> , negozio, data, timestamp, ambiente, cliente)	ambiente → ambiente.id_ambiente cliente →



		utente.email
Appartenenza	appartenenza( <u>id_ambiente</u> , <u>id_utente</u> )	id_ambiente → ambiente.id_ambiente id_utente → utente.email
Pagamento	pagamento( <u>id_pagamento</u> , importo, data, timestamp, id_pagante, id_creditore)	id_pagante → utente.email id_creditore → utente.email
Utilizzo	utilizzo( <u>prodotto</u> , <u>utente</u> )	prodotto → prodotto.id_prodotto utente → utente.email
Commento	commento( <u>id_commento</u> , testo, timestamp, id_prodotto, id_utente)	id_prodotto → prodotto.id_prodotto id_utente → utente.email