Progetto “Basi di dati” 2019/2020



Realizzato da:

De Riggi Mario 0512106024

Rapa Giovanni 0512105967

Cirillo Antonio 0512106096

Vendi e compra appunti online

# 1. Raccolta delle specifiche e descrizione della realtà d’interesse

## Base di progetto: *NoteMarket* – vendita/acquisti appunti e videolezioni

DESCRIZIONE DELLA REALTÀ D’INTERESSE

L’obiettivo del progetto è quello di realizzare una base di dati per la gestione del sito *NoteMarket ,* che offre un servizio di acquisto e vendita di materiale didattico universitario, nello specifico appunti e videolezioni.

Gli appunti che possono essere acquistati sul sito possono essere sia in formato digitale che cartaceo, e prima di essere messi in vendita vengono visionati da moderatori, che sottopongono il materiale ad un’accurata selezione e controllo della qualità e coerenza dei contenuti.

Oltre ad appunti di ogni genere ed argomento, sul sito è possibile trovare anche interessanti videolezioni, tenuti da tutor specializzati in materia.

Il solo ed unico metodo di pagamento che accetta il sito è tramite carta di credito, che viene aggiunta in fase di registrazione all’account.

**SPECIFICHE**

Scopo del database è quello di memorizzare la grande quantità di appunti e videolezioni che vengono messi in vendita, oltre ai dati relativi allo staff e agli utenti iscritti al sito.

Per la creazione di un account, agli utenti è chiesto di fornire:

* nome;
* cognome;
* email;
* indirizzo;
* username;
* password;
* numero di telefono (anche più di uno);
* data di nascita;

Esistono due tipologie di account, a seconda se l’utente vuole iscriversi al sito come venditore o come acquirente:

* Account Venditore: L’utente crea un account venditore se è interessato solamente alla vendita di appunti.
* Account Acquirente: l’utente crea un account acquirente se è interessato solamente all’acquisto di appunti/videolezioni.

Tutti gli appunti presenti sul sito vengono forniti dai venditori. Degli appunti si memorizzano:

* Prezzo;
* Categoria;
* Data caricamento;
* Codice (che li identifica);

A seconda delle richieste dell’acquirente, il materiale può essere inviato sia sotto forma di file, quindi in digitale, o in formato cartaceo.

Per quanto riguarda il materiale digitale, si tiene traccia della dimensione del file e del tipo di documento ( png, pdf, jpg, documento word, ecc…).

Il materiale cartaceo invece viene spedito all’indirizzo segnalato dall’account dell’acquirente e ha quindi un costo di spedizione. Degli appunti cartacei si memorizza anche il numero di pagine del materiale.

Ovviamente non tutto il materiale che viene inviato dai venditori viene poi effettivamente messo in vendita e quindi memorizzato nel database: gli appunti, infatti, vengono sottoposti ad accurati controlli dai moderatori che si occupano di verificarne il contenuto e di approvarlo o rifiutarlo.

Dei moderatori, in quanto membri dello staff del sito, vengono memorizzati:

* Nome;
* Cognome;
* Codice fiscale;
* Indirizzo;
* Numero di telefono (eventualmente più di uno);
* Data di nascita;

Oltre ai moderatori, fanno parte dello staff anche i tutor, che si occupano di tenere le videolezioni. Dei tutor si tiene anche traccia della materia in cui sono specializzati.

Gli appunti quindi non sono l’unica forma di materiale che è possibile reperire sul sito. Infatti le videolezioni forniscono una valida alternativa, soprattutto per chi predilige questo metodo di apprendimento. Delle videolezioni vengono memorizzati i dati relativi a:

* Nome della videolezione;
* Argomento;
* Durata;
* Codice identificativo(ID);

Appunti e videolezioni sono acquistabili sul sito solo tramite carta di credito, di cui si memorizzano:

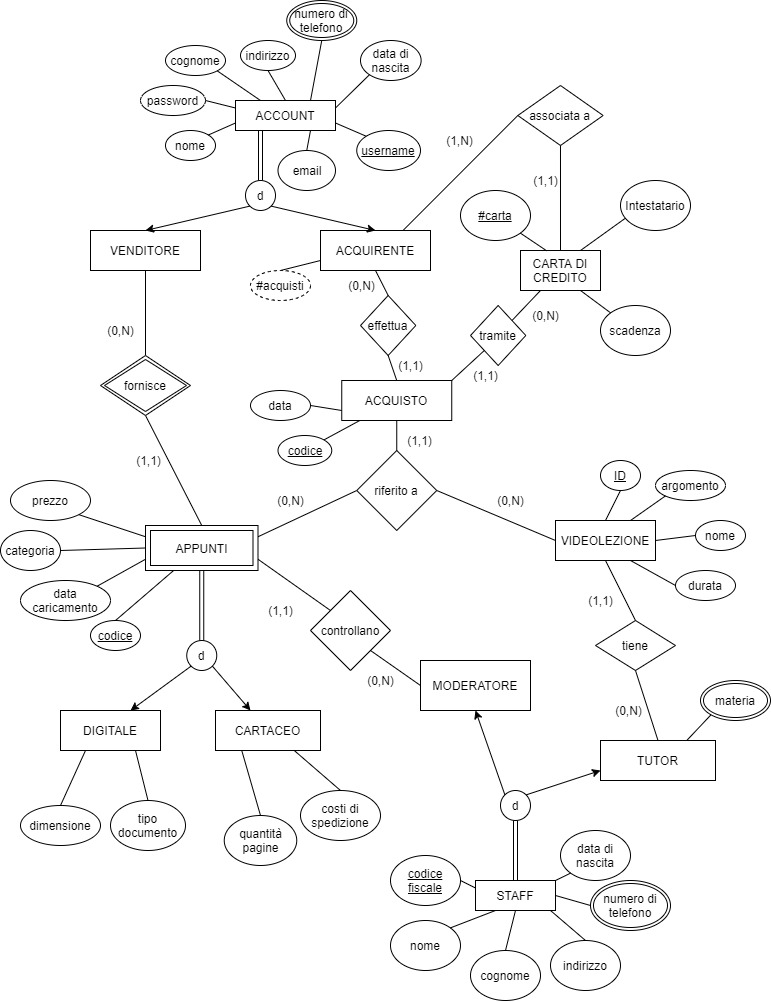
* Intestatario;
* Numero di carta;
* Data di scadenza;

Glossario

|  |  |
| --- | --- |
| TERMINE | DESCRIZIONE |
| Account Acquirente | È la tipologia di account che permette l’acquisto di appunti/videolezioni. |
| Account Venditore | È la tipologia di account che permette la vendita di appunti. |
| Acquisto | Avviene quando un acquirente decide di acquistare appunti/videolezioni. L’acquisto ha un codice che lo identifica e vengono memorizzati i dati relativi alla data in cui esso avviene. |
| Carta di credito | Collegata ad un account acquirente permette l’acquisto del materiale. Si assume appartenga ad un solo account Acquirente. |
| Appunti | Materiale utile per l’apprendimento di determinate nozioni. Possono essere di diverse tipologie (digitale o cartaceo) e sono forniti dai venditori. |
| Videolezioni | Video in cui tutor specializzati tengono lezioni relative ad un determinato argomento. |
| Moderatori | Verificano la qualità del contenuto del materiale inviato prima di renderlo disponibile per la vendita. |
| Tutor | Professionista specializzato in una o più materie che si occupa di tenere videolezioni. |

# 2. Progettazione concettuale della base di dati

Il seguente modello rappresenta concettualmente tutto ciò che verrà gestito all’interno della base di dati da implementare. Il modello utilizzato è quello **EER** (*Enhanced Entity-Relationship*).



# Dizionario dei dati – *NoteMarket*

## ENTITÀ

* **Account.** Un account è l’area personale dell’utente attraverso cui può accedere al sito ed effettuare vendite/acquisti a seconda della tipologia di account. Esso possiede i seguenti attributi:
* **Username**: è il nickname scelto dall’utente. Lo identifica univocamente sul sito (due utenti non possono avere lo stesso username);
* **Password**: parola chiave che l’utente fornisce al momento della registrazione che verrà poi usata per tutte le successive procedure di login;
* **e-mail**: indirizzo di posta elettronica inserito in fase di registrazione. Richiesto assieme alla password in fase di login;
* **nome**: Nome dell’utente;
* **cognome**: Cognome dell’utente;
* **data di nascita**: Data di nascita dell’utente;
* **numero di telefono**: anch’esso richiesto in fase di registrazione. Serve in caso sia necessario contattare il cliente/venditore;

Un account può essere di due tipologie:

* **Account venditore**: Account riservato agli utenti che intendono mettere in vendita degli appunti.
* **Account acquirente**: Account riservato agli utenti che si iscrivono al sito con l’intento di acquistare appunti o videolezioni.

Se un utente intende sia vendere che acquistare, deve creare due account diversi, uno da venditore e uno da acquirente.

* Carta di credito. Entità che rappresenta lo strumento attraverso il quale uno specifico account può effettuare acquisti. Possiede i seguenti attributi:
* Numero carta (#carta): numero di carta che identifica univocamente la carta stessa;
* Scadenza: data di scadenza della carta, oltre la quale la suddetta non sarà utilizzabile per l’acquisto sul sito;
* Intestatario: attributo rappresentante la persona a cui la carta è stata intestata al momento della creazione.
* Acquisto. È l’entità che rappresenta un acquisto, riferito ad appunti o videolezioni. Ha come attributi la data in cui l’acquisto è stato effettuato e un codice, che lo identifica.
* Appunti. Gli appunti rappresentano la principale fonte di guadagno del sito. Vengono caricati dai venditori e prima di essere messi in vendita vengono sottoposti a controlli per verificarne la qualità. Degli appunti si memorizzano:
  + Prezzo: il prezzo di vendita;
  + Categoria: a quale categoria di appunti appartiene (riassunti, tesi, saggi, ecc…)
  + Data Caricamento: la data in cui gli appunti sono stati caricati sul sito;
  + Codice: codice univoco che li identifica;

A seconda delle richieste dell’acquirente, il materiale acquistato può essere inviato in formato digitale o cartaceo.

* Digitale: il materiale acquistato viene spedito via mail all’acquirente. Si tiene traccia della dimensione del file e del tipo di documento (pdf, jpg, presentazione PowerPoint, ecc…);
* Cartaceo: il materiale viene spedito tramite corriere all’indirizzo fornito dall’acquirente. Presenta quindi dei costi di spedizione; inoltre si tiene traccia della quantità di pagine che compongono gli appunti.
* Videolezioni. Assieme agli appunti, costituiscono il materiale acquistabile sul sito. Le videolezioni, al contrario degli appunti, non sono fornite dai venditori, ma dal sito stesso. Delle videolezioni si tiene traccia della durata e dell’argomento che tratta.
* Staff. Affinché il sito possa funzionare nel migliore dei modi, c’è bisogno di uno staff valido che si occupi delle diverse mansioni. Le figure che compongono lo staff sono:
* Moderatori: si occupano del controllo degli appunti. Prima di essere messi in vendita, infatti, i moderatori ne verificano la qualità e controllano se il materiale è vendibile.
* Tutor: presentano le videolezioni. Dei Tutor si tiene traccia della/e materia/e in cui è specializzato.

Di tutti i membri dello staff vengono memorizzate le medesime informazioni, ovvero:

* Nome;
* Cognome;
* Codice fiscale;
* Data di nascita;
* Indirizzo;
* Numero di telefono (eventualmente più di uno);

RELAZIONI

1. **Fornisce.** È la relazione che lega le entità *Venditore* e *Appunti*. Non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Ogni venditore può fornire infiniti appunti, ma anche nessuno **(0,N)**.
* Gli appunti sono stati forniti da uno e un solo venditore **(1,1)**.

1. **Effettua.** È la relazione che lega le entità *Acquirente* e *Acquisto*. Non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Ogni acquirente può effettuare un numero indefinito di acquisti, ma anche nessuno **(0,N).**
* Un singolo acquisto si riferisce ad uno ed un solo acquirente **(1,1);**

1. **Associata a.** È la relazione che lega le entità *Acquirente* e *Carta di credito.* Non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Ad un account Acquirente possono essere associate infinite carte di credito, ma è necessario associarne almeno una **(1,N).**
* Una determinata carta di credito può essere associata ad uno ed un solo account Acquirente **(1,1)**.

1. **Tramite**. È la relazione che lega le entità *Acquisto* e *Carta di credito.* Non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Con una carta di credito si possono effettuare infiniti acquisti, o anche nessuno **(0,N).**
* Un determinato acquisto può essere effettuato con una sola carta di credito ( non si può effettuare un acquisto utilizzando più carte di credito contemporaneamente) **(1,1).**

1. **Riferito a.** È la relazione che lega l’entità *Acquisto* con le entità *Appunti* e *Videolezione.* Infatti un acquisto non deve necessariamente riferirsi ad appunti ma anche a videolezioni. Si stabiliscono le seguenti cardinalità:

* Un acquisto si riferisce necessariamente ad una videolezione o a un blocco di appunti **(1,1).**
* Gli appunti che possono essere acquistati, così come le videolezioni, possono essere infiniti **(0,N).**

1. **Controllano.** È la relazione che lega le entità *Moderatore* e *Appunti.* È necessaria affinche i moderatori siano messi in relazione con gli appunti che controllano periodicamente. La relazione non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Un moderatore può controllare un numero indefinito di appunti, ma anche nessuno **(0,N).**
* Un determinato blocco di appunti viene controllato da un solo moderatore, e non può non essere controllato **(1,1).**

1. **Tiene.** È la relazione che lega le entità *Tutor* e *Videolezione.* Non possiede attributi. Si stabiliscono le seguenti cardinalità:

* Un tutor può tenere un numero indefinito di videolezioni, ma anche nessuna **(0,N).**
* Una videolezione è tenuta da uno e un solo tutor **(1,1).**

# Vincoli di integrità e regole di vincolo

L’elenco che segue rappresenta i vincoli non esprimibili attraverso il modello EER.

* I pagamenti ai venditori vengono effettuati dal sito;
* Ogni volta che degli appunti vengono acquistati, non vengono eliminati dal database, ma viene venduta una copia;
* Si assume che alla creazione del database siano già disponibili appunti anche in assenza di account registrati;
* È possibile effettuare un solo acquisto alla volta;
* Quando la spedizione degli appunti cartacei è gratuita, l’attributo *costo di spedizione* è contrassegnato come NULL;
* Il sito non accetta carte di credito scadute;

# 3. Elenco delle procedure per la gestione dei dati

**Op1**: Inserimento di un account. (3600/anno)

**Op2**: Inserimento di un nuovo numero di telefono ad un account. (6/anno)

**Op3**: Inserimento di una nuova carta di credito collegata ad un account. (3600/anno)

**Op4**: Inserimento di un nuovo acquisto. (2500/anno)

**Op5**: Inserimento di un nuovo appunto. (4000/anno)

**Op6**: Inserimento di una video-lezione. (1300/anno)

**Op7**: Inserimento di un membro dello staff. (20/anno)

**Op8**: Inserimento di un numero di telefono per un membro dello staff. (6/anno)

**Op9**: Inserimento di una specializzazione per un tutor. (2/anno)

**Op10**: Stampa dei dati relativi ai membri dello staff. (12/anno)

**Op11**: Stampa dei dati relativi ai Moderatori. (12/anno)

**Op12**: Stampa dei dati relativi ai Tutor. (12/anno)

**Op13**: Stampa dati relativi ad un account. (12/anno)

**Op14**: Stampa dati relativi a tutti gli account. (12/anno)

**Op15**: Stampa di tutti gli acquisti avvenuti in un determinato intervallo di date. (12/anno)

**Op16**: Stampa di tutti gli acquirenti. (12/anno)

**Op17**: Stampa di tutti i venditori. (12/anno)

**Op18**: Stampa dei dati degli appunti. (12/anno)

**Op19**: Stampa di tutti gli appunti digitali. (12/anno)

**Op20**: Stampa di tutti gli appunti cartacei. (12/anno)

**Op21**: Stampa tutte le video-lezioni, tranne dei tutor che insegnano una determinata materia. (12/anno)

**Op22**: Stampa tutti gli appunti con prezzo inferiore ad una determinata cifra. (20/anno)

**Op23:** Stampa del codice degli acquisti di appunti che non sono in formato digitale, la cui spedizione è gratuita. (12/anno)

**Op24**: Stampa del numero degli acquisti di un account acquirente che ha effettuato almeno un acquisto. (12/anno)

**Op25**: Stampa dell’appunto con il prezzo maggiore. (12/anno)

**Op26**: Stampa della lista dei Tutor il cui cognome inizia con una lettera o una stringa data. (12/anno)

# 4. Progettazione logica

# Tavola dei volumi

|  |  |  |
| --- | --- | --- |
| **CONCETTO** | **TIPO** | **VOLUME** |
| ACCOUNT | E | 7000 |
| VENDITORE | E | 3000 |
| ACQUIRENTE | E | 4000 |
| CARTA DI CREDITO | E | 7000 |
| ACQUISTO | E | 2000 |
| APPUNTI | E | 4500 |
| DIGITALE | E | 4000 |
| CARTACEO | E | 500 |
| VIDEOLEZIONE | E | 2250 |
| STAFF | E | 200 |
| MODERATORE | E | 100 |
| TUTOR | E | 100 |
| Associata a | R | 7000 |
| Fornisce | R | 21000 |
| Controllano | R | 10000 |
| Tiene | R | 400 |
| Riferito a | R | 3000 |
| Tramite | R | 14000 |
| Effettua | R | 14000 |

# Tavola delle operazioni

|  |  |  |
| --- | --- | --- |
| **OPERAZIONE** | **TIPO** | **FREQUENZA** |
| OP.1 | I | 3600/anno |
| OP.2 | I | 6/anno |
| OP. 3 | I | 3600/anno |
| OP. 4 | I | 2500/anno |
| OP. 5 | I | 4000/anno |
| OP. 6 | I | 1300/anno |
| OP. 7 | I | 20/anno |
| OP. 8 | I | 6/anno |
| OP. 9 | I | 2/anno |
| OP. 10 | B | 12/anno |
| OP. 11 | B | 12/anno |
| OP. 12 | B | 12/anno |
| OP. 13 | B | 12/anno |
| OP. 14 | B | 12/anno |
| OP. 15 | B | 12/anno |
| OP. 16 | B | 12/anno |
| OP. 17 | B | 12/anno |
| OP. 18 | B | 12/anno |
| OP. 19 | B | 12/anno |
| OP. 20 | B | 12/anno |
| OP. 21 | I | 12/anno |
| OP. 22 | I | 20/anno |
| OP. 23 | B | 12/anno |
| OP. 24 | B | 12/anno |
| OP. 25 | B | 12/anno |
| OP. 26 | I | 12/anno |

# Tavole degli accessi

Il dato ridondante di cui analizzare le influenze nelle prestazioni del database è ***#acquisti***, attributo dell’entità Acquisto.

Per facilitarci le cose, ipotizziamo che un accesso in scrittura equivalga a due accessi in lettura(**1S=2L**).

Le operazioni che coinvolgono il dato ridondante sono:

**OP.4**

**CON DATO RIDONDANTE**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO DI ACCESSO |
| ACQUISTO | E | 1 | S |
| Effettua | R | 1 | S |
| ACQUIRENTE | E | 1 | L |
| ACQUIRENTE | E | 1 | S |

**#accessi = (2 x 3S x 2.500) + 1L x 2.500 = 17.500 accessi**

**SENZA DATO RIDONDANTE**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO DI ACCESSO |
| ACQUISTO | E | 1 | S |
| Effettua | R | 1 | S |

**#accessi = 2S x 2 x 2.500 = 10.000 accessi**

**OP.16**

**CON DATO RIDONDANTE**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO DI ACCESSO |
| ACQUIRENTE | E | 1 x 4000 | L |

**#accessi = (1x4000) L x 12 =48.000 accessi**

**SENZA DATO RIDONDANTE**

|  |  |  |  |
| --- | --- | --- | --- |
| CONCETTO | COSTRUTTO | ACCESSI | TIPO DI ACCESSO |
| ACQUIRENTE | E | 2 | L |
| effettua | R | 2 | L |

**#accessi = 4L x 50 = 200 accessi**

**TOTALE ACCESSI CON ATTRIBUTO RIDONDANTE = 17.500 + 48.000 =65.500 accessi**

**TOTALE ACCESSI SENZA ATTRIBUTO RIDONDANTE = 10.000 + 200 = 10.200 accessi**

**MEMORIA (#acquisti) = 2 byte x 4000 = 8000 byte**

In definitiva, confrontando il numero di accessi con e senza dato ridondante, notiamo come convenga eliminarlo.

# Eliminazione delle generalizzazioni

Quando si parla di eliminazione delle generalizzazioni, è possibile scegliere tra diverse alternative:

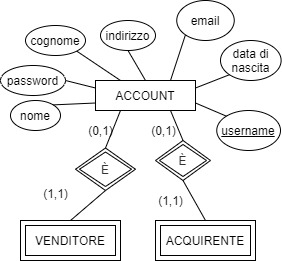
1. Accorpamento delle figlie della generalizzazione nel padre;
2. Accorpamento del padre della generalizzazione nelle figlie;
3. Sostituzione della generalizzazione con associazioni;

In questo caso, si sceglie di applicare l’alternativa 3: tutte le entità vengono mantenute; le entità figlie vengono messe in associazione con l’entità padre e sono identificate esternamente tramite l’associazione. La cardinalità (0,1) indica che per tale associazione l’entità può avere zero o una entità figlio. Mentre (1,1) che l’entità figlio può avere uno e un solo padre.

Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo

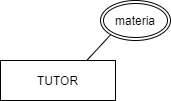
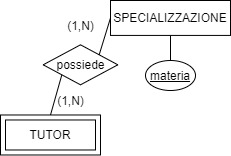
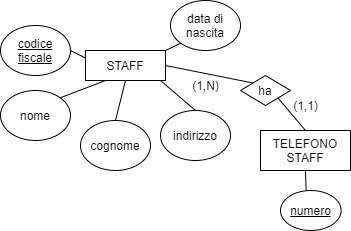
Descrizione generata automaticamente



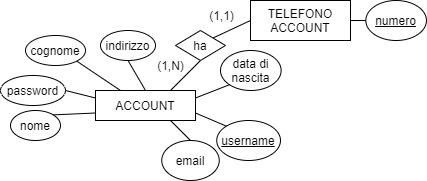
# Eliminazione attributi multivalore

Mentre gli attributi del modello relazionale assumono solo valori atomici, il modello EER permette di specificare attributi composti (che assumono una sequenza di valori non omogenei) e attributi multivalore (che assumono una sequenza di valori omogenei). Occorre dunque rimuovere questi attributi mediante una fase preliminare di ristrutturazione del modello EER.

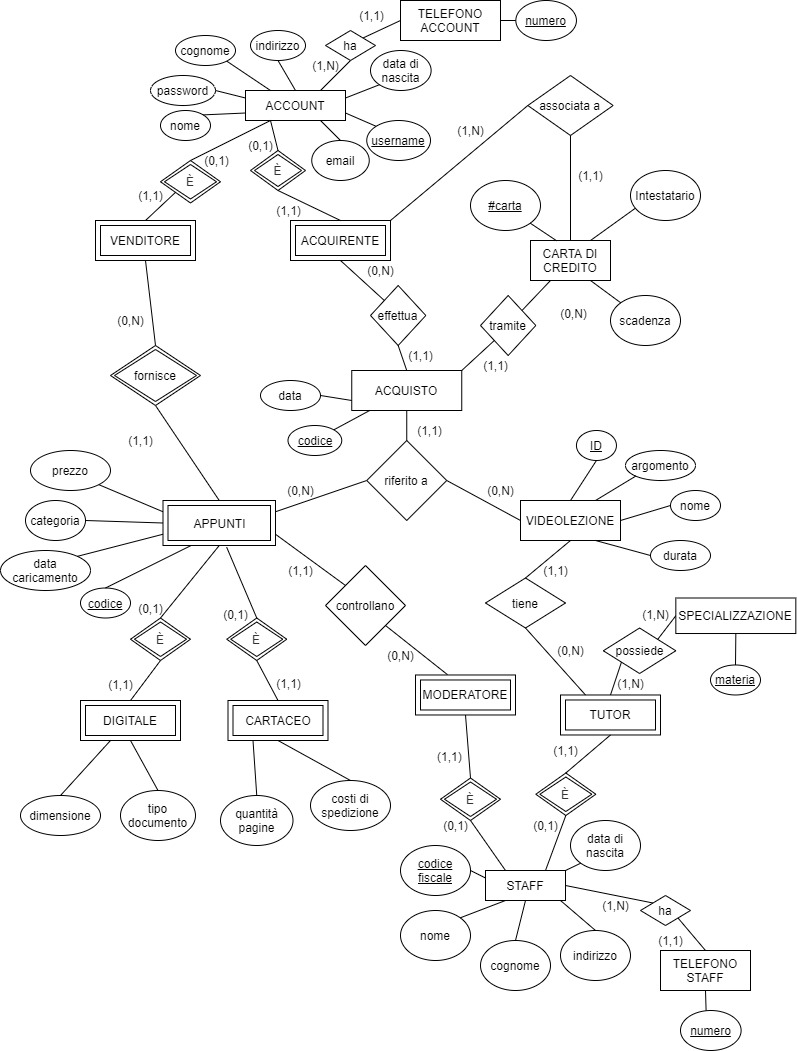
La rimozione degli attributi multivalore è semplice:

Immagine che contiene testo

Descrizione generata automaticamente



# Ristrutturazione dello schema concettuale



# Mapping

Account (Username, e-mail, nome, cognome, password, indirizzo, data di nascita);

Venditore (Account.username↑);

Acquirente (Account.username↑);

Telefono Account (Numero, Account.username↑);

Carta di credito (#carta, scadenza, intestatario, acquirente.Account.username↑);

Acquisto (codice, data, acquirente.account.username↑, CartadiCredito.#carta↑, Appunti.#codice↑, venditore.account.username↑, videolezione.id↑);

Appunti (codice, venditore.account.username↑, data caricamento, categoria, prezzo, moderatore.staff.codicefiscale↑);

Digitale (venditore.account.username↑, appunti.codice↑, dimensione, tipo documento);

Cartaceo (venditore.account.username↑, appunti.codice↑, quantità pagine, costi di spedizione);

Videolezioni (id, argomento, nome, durata, tutor.Staff.codicefiscale↑);

Tutor (Staff.codicefiscale↑);

Specializzazione (Materia);

Possiede (Specializzazione.materia↑, tutor.staff.codicefiscale↑);

Moderatore (staff.codicefiscale↑);

Staff (codicefiscale, nome, cognome, indirizzo, data di nascita);

Telefono (numero, staff.codicefiscale↑);

# Normalizzazione dello schema concettuale

Prima di procedere con la normalizzazione, è bene richiamare le seguenti definizioni.

Una superchiave di uno schema di relazione R = {A1, A2, …, An} è un insieme di attributi S ⊆ R con la proprietà che non esistono due tuple t1e t2 in qualche stato di relazione legale di r ∈ R tali che t1[S] = t2[S].

Una chiave k è una superchiave con la proprietà aggiuntiva che la rimozione da k di qualche attributo fa perdere a k la proprietà di superchiave. Tra tutte le chiavi candidate, una di esse viene scelta arbitrariamente ed è detta *chiave primaria (PK)*.

Un attributo A di uno schema di relazione R che ricorre in qualche chiave di R è detto attributo primo di R (*non-primo* altrimenti).

Occorre, a questo punto, normalizzare ogni relazione analizzando le *dipendenze funzionali* delle stesse. Una dipendenza funzionale (FD) è un vincolo tra due insiemi di attributi del database.

Supponiamo che lo schema di database relazionale abbia n attributi A1, …, An e che l’intero database sia descritto da uno schema di relazione universale R = {A1, …, An}. Una *dipendenza funzionale*, denotata con X→Y, con X, Y⊆R, specifica un vincolo sulle possibili tuple che possono formare un’istanza di relazione r di R. Il vincolo stabilisce che se X → Y, allora ∀t1, t2 in R tali che t1[X] = t2[X], deve valere t1[Y] = t2[Y].

Ciò significa che i valori della componente Y di una tupla di R *dipendono da* (o *sono determinati da*) i valori della componente X. Alternativamente, i valori della componente X di una tupla *determinano univocamente* (o *funzionalmente*) i valori della componente Y.

Sia F l’insieme delle dipendenze funzionali di uno schema di relazione R. Alcune di queste dipendenze possono essere inferite (o *indotte*) da F: l’insieme di tali dipendenze è detto *chiusura di F*, ed è denotato con F+.  
Lo schema, tuttavia, non possiede dipendenze funzionali, ma verrà analizzato un esempio generico per esporre il concetto di normalizzazione.

*1NF*

La prima forma normale (1NF) è stata definita per non consentire attributi *multivalued*, *composti* e loro combinazioni: gli unici valori consentiti da 1NF sono valori *atomici* (indivisibili). L’idea è formare una nuova relazione per ogni valore *non* atomico.  
In particolar modo, nei database moderni tale forma normale è superata, in quanto l’eliminazione di attributi multivalore e/o composti avviene già in fase di *ristrutturazione* dello schema (già esposta precedentemente), per cui ogni relazione è già in 1NF.

*2NF*

La seconda forma normale (2NF) è basata sul concetto di dipendenza funzionale *piena*. Una dipendenza funzionale X → Y è piena se la rimozione di qualche attributo A da X implica che la dipendenza non vale più (∀ A ∈ X, X\ {A} → Y); una dipendenza funzionale X → Y è parziale se qualche attributo A ∈ X può essere rimosso e la dipendenza vale ancora (∃ A ∈ X, X\ {A} → Y).  
Uno schema di relazione R è in 2NF se R è in 1NF e, inoltre, ogni attributo *non* primo A in R è pienamente funzionalmente dipendente della chiave primaria di R.

In sostanza, abbiamo la violazione della 2NF se la chiave primaria è multipla e gli attributi non chiave non dipendono dall’*intera* chiave primaria. Per normalizzare, creiamo una relazione per ogni chiave parziale con i relativi attributi che influenza (è importante mantenere la relazione principale con la chiave multipla e gli attributi che dipendono da essa).

R (A, B, C, D, E, F, G, H), con F= {B → A, C → DG, D → AH, E → BF, CE → H}

*Chiave candidata*: CE  
{CE}+ = AHDGBFEC = R  
{C}+ = CDGAH  
{E}+ = EBFA  
Quindi CE è minimale.  
*Decomponiamo in 2NF*:  
Le dipendenze che violano la 2NF sono C→DG e E→BF.  
Otteniamo: R1= {C, A, D, G, H}, R2= {E, B, F, A}, R3= {C, E}.

*3NF*

La terza forma normale (3NF) è basata sul concetto di dipendenza *transitiva*. Una dipendenza funzionale X → Y in uno schema di relazione R è una dipendenza transitiva se esiste un insieme di attributi Z che *non* è sottoinsieme di alcuna chiave di R e valgono X → Z e Z → Y.

Uno schema di relazione R è in 3NF se R è in 2NF e, inoltre, nessun attributo *non primo* di R è transitivamente dipendente dalla chiave primaria. Occorre decomporre la relazione R in un insieme di relazioni che includono gli attributi non-chiave che funzionalmente determinano altri attributi non-chiave.

R1= {C, A, D, G, H}, R2= {E, B, F, A}, R3= {C, E},

con F= {B → A, C → DG, D → AH, E → BF, CE → H}

*Decomponiamo in 3NF*:  
Le dipendenze che violano la 3NF sono D→AH e B→A.  
Otteniamo: R11= {C, D, G}, R12= {D, A, H}, R21= {E, B, F}, R22= {B, A}, R3= {C, E}.

Non avendo effettuato normalizzazione il mapping sarà uguale al precedente.

# 5. Realizzazione della base di dati in MySQL

drop database if exists NoteMarket;

create database NoteMarket;

use NoteMarket;

CREATE TABLE Account (

Username VARCHAR(20) PRIMARY KEY NOT NULL,

email VARCHAR(30) NOT NULL,

nome VARCHAR(30) NOT NULL,

cognome VARCHAR(20) NOT NULL,

psw VARCHAR(20) NOT NULL,

indirizzo VARCHAR(30) NOT NULL,

dataDiNascita DATE NOT NULL

);

CREATE TABLE Venditore (

username VARCHAR(20) PRIMARY KEY NOT NULL,

FOREIGN KEY (username)

REFERENCES Account (Username) on delete cascade on update cascade

);

CREATE TABLE Acquirente (

username VARCHAR(20) PRIMARY KEY NOT NULL,

FOREIGN KEY (username)

REFERENCES Account (Username) on delete cascade on update cascade

);

CREATE TABLE TelefonoAccount (

Numero INT UNSIGNED PRIMARY KEY NOT NULL,

username VARCHAR(20) NOT NULL,

FOREIGN KEY (username)

REFERENCES Account (Username)

);

CREATE TABLE CartaDiCredito (

numeroCarta CHAR(16) PRIMARY KEY NOT NULL,

scadenza CHAR(5) NOT NULL,

intestatario VARCHAR(20) NOT NULL,

username VARCHAR(20) NOT NULL,

FOREIGN KEY (username)

REFERENCES Acquirente(Username)

);

CREATE TABLE Staff (

codiceFiscale CHAR(16) PRIMARY KEY NOT NULL,

nome VARCHAR(10) NOT NULL,

cognome VARCHAR(10) NOT NULL,

indirizzo VARCHAR(30) NOT NULL,

dataNascita DATE NOT NULL

);

CREATE TABLE Moderatore (

codiceFiscale CHAR(16) PRIMARY KEY NOT NULL,

FOREIGN KEY (codiceFiscale)

REFERENCES Staff (codiceFiscale) on delete cascade on update cascade

);

CREATE TABLE Tutor (

codiceFiscale CHAR(16) PRIMARY KEY NOT NULL,

FOREIGN KEY (codiceFiscale)

REFERENCES Staff (codiceFiscale) on delete cascade on update cascade

);

CREATE TABLE Videolezione (

Id INT UNSIGNED PRIMARY KEY NOT NULL,

argomento VARCHAR(20) NOT NULL,

nome VARCHAR(30) NOT NULL,

durata FLOAT UNSIGNED NOT NULL,

codFiscTutor CHAR(16) NOT NULL,

FOREIGN KEY (codFiscTutor)

REFERENCES Tutor (codiceFiscale) on delete cascade on update cascade

);

CREATE TABLE Appunti (

codice INT UNSIGNED NOT NULL,

usernameVenditore VARCHAR(20) NOT NULL,

PRIMARY KEY (codice , usernameVenditore),

dataCaricamento DATE NOT NULL,

categoria ENUM('tesi', 'saggi', 'schema concettuale') NOT NULL,

prezzo FLOAT UNSIGNED NOT NULL,

codiceFiscaleModeratore CHAR(16) NOT NULL,

FOREIGN KEY (usernameVenditore)

REFERENCES Venditore (username),

FOREIGN KEY (codiceFiscaleModeratore)

REFERENCES Moderatore(codiceFiscale)

);

CREATE TABLE Acquisto (

codice INT UNSIGNED auto\_increment PRIMARY KEY NOT NULL,

codiceAppunti INT UNSIGNED,

data DATE NOT NULL,

usernameAcquirente VARCHAR(20) NOT NULL,

numeroCarta CHAR(16) NOT NULL,

usernameVenditore VARCHAR(20),

idVideolezione INT UNSIGNED,

FOREIGN KEY (usernameAcquirente)

REFERENCES Acquirente (Username),

FOREIGN KEY (numeroCarta)

REFERENCES CartaDiCredito (numeroCarta),

FOREIGN KEY (codiceAppunti)

REFERENCES Appunti (codice),

FOREIGN KEY (usernameVenditore)

REFERENCES Venditore (username),

FOREIGN KEY (idVideolezione)

REFERENCES Videolezione (id)

);

CREATE TABLE Digitale (

usernameVenditore VARCHAR(20) NOT NULL,

codice INT UNSIGNED NOT NULL,

primary key(usernameVenditore, codice),

dimensione INT UNSIGNED NOT NULL,

tipoDocumento ENUM("pdf", "jpg", "doc") NOT NULL,

FOREIGN KEY (usernameVenditore)

REFERENCES Venditore (username),

FOREIGN KEY (codice)

REFERENCES Appunti (codice) on delete cascade on update cascade

);

CREATE TABLE Cartaceo (

usernameVenditore VARCHAR(20) NOT NULL,

codice INT UNSIGNED NOT NULL,

primary key(usernameVenditore, codice),

quantitàPagine INT UNSIGNED NOT NULL,

costiSpedizione FLOAT UNSIGNED,

FOREIGN KEY (usernameVenditore)

REFERENCES Venditore (username),

FOREIGN KEY (codice)

REFERENCES Appunti (codice) on delete cascade on update cascade

);

CREATE TABLE Specializzazione (

Materia VARCHAR(20) PRIMARY KEY NOT NULL

);

CREATE TABLE Possiede (

Materia VARCHAR(20) NOT NULL,

codiceFiscaleTutor CHAR(16) NOT NULL,

primary key(Materia, codiceFiscaleTutor),

FOREIGN KEY (Materia)

REFERENCES Specializzazione (Materia),

FOREIGN KEY (codiceFiscaleTutor)

REFERENCES Tutor (codiceFiscale)

);

CREATE TABLE TelefonoStaff (

numero INT UNSIGNED PRIMARY KEY NOT NULL,

codiceFiscaleStaff CHAR(16) NOT NULL,

FOREIGN KEY (codiceFiscaleStaff)

REFERENCES Staff (codiceFiscale)

);

Popolamento

INSERT INTO account VALUES

('Givaa', 'rapagiovanni@gmail.com', 'Giovanni', 'Rapa', 'pippo', 'Via Aldo Moro, 54', '1999-12-19'),

('Derimax', 'deriggimario@yahoo.com', 'Mario', 'De Riggi', 'peppino', 'Via Fontanelle, 32', '2000-11-23'),

('Sayril', 'cirilloantonio@gmail.com', 'Antonio', 'Cirillo', 'bellazio', 'Via dei porticati, 77', '1999-12-20'),

('Elliot', 'adrianoamato@libero.it', 'Adriano', 'Amato', 'lolle', 'Viale degli astronauti, 120', '1999-12-31'),

('yantCaccia', 'antoniocacciapuoti@libero.it', 'Antonio', 'Cacciapuoti', 'broxia', 'Via dei territori, 54', '1999-10-10'),

('Rifle', 'andreafucile@yahoo.com', 'Andrea', 'Fucile', 'rippe', 'Via appia, 34', '2000-09-10'),

('paglianese', 'angeloafeltra@libero.it', 'Angelo', 'Afeltra', 'chemasto', 'Viale dei pineti, 73', '1999-07-22'),

('AuronVS', 'carmineamendola@gmail.com', 'Carmine', 'Amendola', 'iloveyou', 'Viale degli eroi, 25', '1994-05-26');

INSERT INTO venditore VALUES

('Givaa'),

('Derimax'),

('Elliot');

INSERT INTO acquirente VALUES

('Sayril'),

('yantCaccia'),

('paglianese');

INSERT INTO CartaDiCredito VALUES

(3505371321836, '09/13', 'Antonio Cacciapuoti', 'yantCaccia'),

(3736482213047, '07/22', 'Antonio Cirillo', 'Sayril'),

7482398492889, '07/24', 'Angelo Afeltra', 'paglianese');

(18, 'Derimax', '2015-11-22', 'tesi', 35, 'FRTCPD88T14A456C'),

(300, 'Derimax', '2011-11-02', 'tesi', 25, 'FRTCPD88T14A456C');

INSERT INTO acquisto (codiceAppunti, data, usernameAcquirente, numeroCarta, usernameVenditore, idVideolezione) VALUES

(24, '2019-12-22', 'Sayril', 3736482213047, 'Givaa', null),

(null, '2018-10-24', 'yantCaccia', 3505371321836, null, 35),

(22, '2019-12-31', 'paglianese', 7482398492889, 'Elliot', null),

(300, '2019-11-21', 'Sayril', 3736482213047, 'Derimax', null);

INSERT INTO Digitale VALUES

('Givaa', 30, 330, 'pdf'),

('Derimax', 18, 20, 'jpg'),

('Elliot', 22, 5, 'doc');

INSERT INTO Cartaceo VALUES

('Givaa', 24, 130, 23.50),

('Derimax', 300, 100, null);

INSERT INTO Specializzazione VALUES

('Arte'),

('Giurisprudenza'),

('Storia');

INSERT INTO Possiede VALUES  
 ('Storia', 'DHYFRM35R18C522V'),

('Giurisprudenza', 'PCCBCC73R06D781L'),

('Arte', 'DHYFRM35R18C522V');

INSERT INTO telefonostaff VALUES

(3454323232, 'FRTCPD88T14A456C'),

(3313245465, 'PCCBCC73R06D781L'),

3283445678, 'DHYFRM35R18C522V'),

(3327689574, 'PRSFNC72C04F657E');

6. Test dell’applicazione

Immagine che contiene screenshot

Descrizione generata automaticamenteImmagine che contiene screenshot

Descrizione generata automaticamente**Op1**: Inserimento di un account. (3600/anno)

Immagine che contiene screenshot

Descrizione generata automaticamente

# Immagine che contiene screenshot Descrizione generata automaticamente

Immagine che contiene screenshot

Descrizione generata automaticamenteImmagine che contiene screenshot

Descrizione generata automaticamente**Op21**: Stampa tutte le video-lezioni, tranne dei tutor che insegnano una determinata materia. (12/anno)

Immagine che contiene screenshot

Descrizione generata automaticamenteImmagine che contiene screenshot

Descrizione generata automaticamente**Op22**: Stampa tutti gli appunti con prezzo inferiore ad una determinata cifra. (20/anno)

**Op10**: Stampa dei dati relativi ai membri dello staff. (12/anno)

Immagine che contiene screenshot

Descrizione generata automaticamente