**Sito di E-Ccomerce svillupato da:**

**Davide Lavalle**

**Silvio Russo**

**Ingegneria informatica e delle telecomunicazioni.**

**Anno accademico 2018-2019.**

**Materia: Sistemi informativi.**

1. **Descrizione sommaria dell’intero sistema.**

Il sito web sviluppato è, come da consegna, un e-commerce che si occupa della compravendita di componentistica Hardware di un personal computer, come Schede madri, processori o memorie.

Il software è diviso in “Area utente” ed “Area amministrazione” a cui è possibile accedere per mezzo della pagina di Login.

Dopo avere effettuato il Login, se l’utente è registrato, il sistema indirizzerà questo nella Home Page.

Se l’utente ha i permessi da amministratore gli verrà mostrato nel Footer della pagina, un pulsante ”Amministrazione”, che permette l’accesso alla pagina di amministrazione del sito.

Il sito è diviso nelle seguenti sezioni: Home Page, Chi siamo, Prodotti, Carrello e Area personale, tutte accessibili dal Header del sito.

L’Home page mostra al cliente dieci prodotti alla volta, ordinati per ordine di aggiunta.

Da qui è possibile aggiungere al carrello i prodotti scegliendone la quantità, e leggere una descrizione di ogni prodotto cliccando il pulsante “Maggiori informazioni”.

La sezione prodotti è accessibile dal Header del sito, un passaggio del puntatore sopra il relativo pulsante farà infatti aprire un menù a tendina che mostrerà un elenco dal quale è possibile selezionare la categoria dei prodotti da volere consultare.

Cliccando su una categoria verrà mostrata una vetrina simile alla Home page che presenta esclusivamente i prodotti della categoria selezionata.

Da qui, è possibile, come nella Home page, aggiungere prodotti al carrello o leggere maggiori informazioni riguardo al prodotto.

Il carrello mostrerà i prodotti che vi sono stati aggiunti, permettendo di rimuoverli, modificarne la quantità o acquistarli.

Premendo sul pulsante “Acquista” il sistema controlla che l’utente abbia aggiunto tutti i dati necessari per effettuare l’acquisto(Indirizzo di spedizione, indirizzo di fatturazione, metodo di pagamento), costringendolo ad inserirli nel caso non siano presenti.

Se l’utente ha inserito i dati necessari per l’acquisto questi gli vengono mostrati in una schermata dalla quale possono essere modificati e nel caso si volesse aggiungerli.

A questo punto l’utente può cliccare il pulsante “Paga”, completando l’acquisto, al quale verrà applicato uno sconto casuale pari se la cifra totale supera i 250€.

Sempre dall’ Header è possibile accedere alle pagine di gestione dell’account.

Passando il puntatore sopra il pulsante che presenta scritto il nome utente, si aprirà un menù a tendina con le opzioni: Il mio Account, I miei Ordini e Logout.

Dalla sezione “Il mio Account” è possibile modificare i dati utente, gli indirizzi di fatturazione e di spedizione e il metodo di pagamento.

Dalla sezione “I miei ordini”, è possibile visualizzare uno storico degli ordini effettuati.

Cliccando sul pulsante “Maggiori dettagli” è possibile visualizzare tutte le informazioni di pagamento (metodo di pagamento, corriere e indirizzi di spedizione e fatturazione), le informazioni sul prodotto acquistato ed inoltre è possibile scaricare, in formato PDF, la fattura relativa all’ordine cliccando sul pulsante “Scarica fattura”.

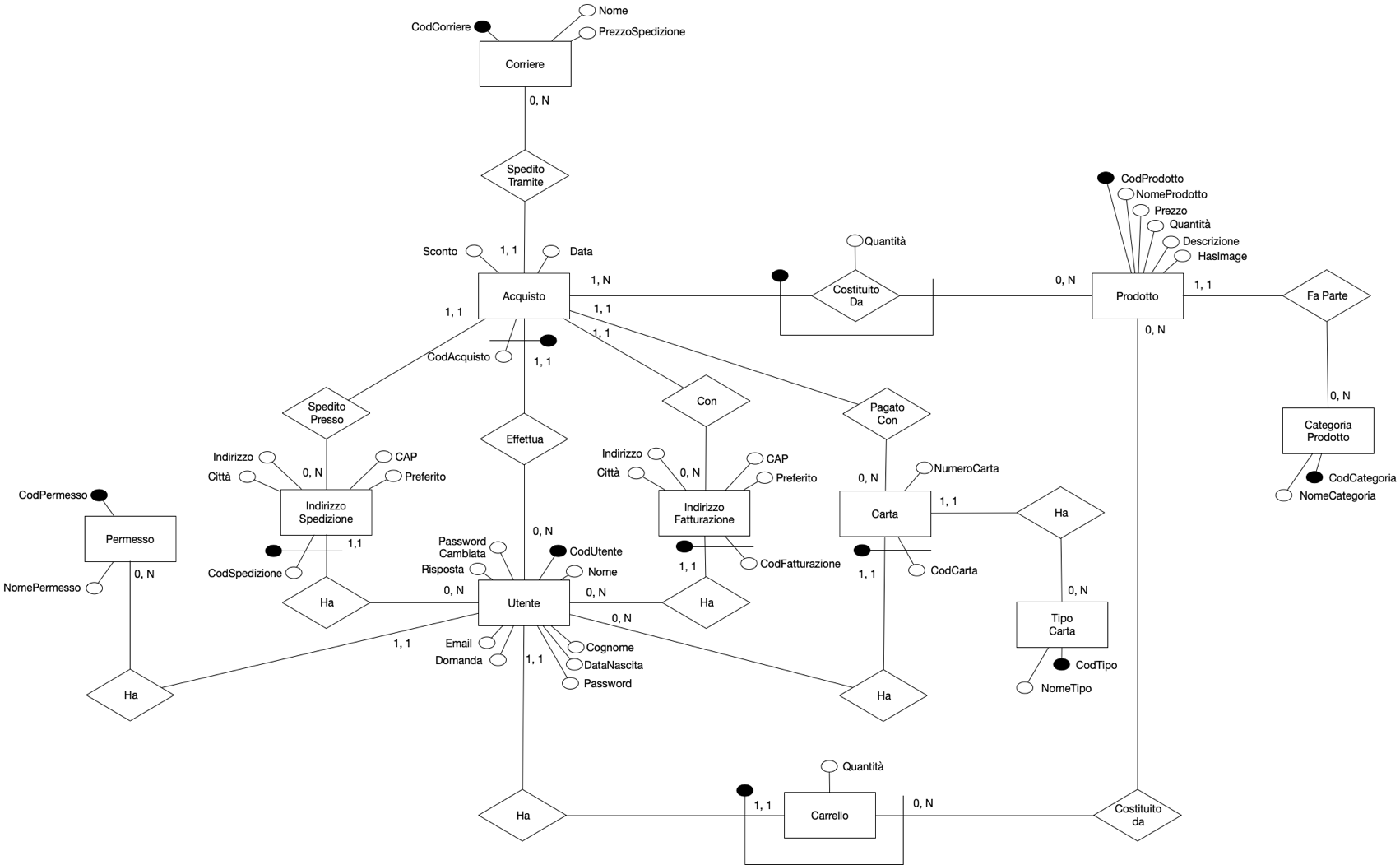
Per quanto riguarda la sezione di “Amministrazione” questa è accessibile, nel caso l’utente che ha effettuato il login abbia i permessi da amministratore, dal Footer del sito mediante il pulsante “Amministrazione”.

Questa sezione permette di: Aggiungere corrieri, Gestire gli account degli utenti, Aggiungere prodotti, Gestire i prodotti e visualizzare gli ordini degli utenti.

L’amministratore ha quindi pieno controllo su tutti i dati, da quelli personali degli utenti, a quelli dei prodotti.

1. **Diagrammi ERD**

La base di dati progettata è quella rappresentata dal seguente schema ER:

****

Da questo, date le cardinalità 1,1 sono state eliminate le seguenti relazioni:

1. La relazione “Ha” tra le entità “Permesso” e “Utente” è stata inglobata nell’entità “Utente” mediante l’attributo “RefPermesso”.
2. La relazione “Ha” tra le entità “Carrello” e “Utente” è stata inglobata nell’entità “Carrello” mediante l’attributo “RefUtente”.
3. La relazione “Fa parte” tra le entità “Prodotto” e “Categoria Prodotto” è stata inglobata nell’entità “Prodotto” mediante l’attributo “RefCategoria”.
4. La relazione “Ha” tra le entità “Carta” e “Utente” è stata inglobata nell’entità “Carta” mediante l’attributo “RefUtente”.
5. La relazione “Ha” tra le entità “Indirizzo Spedizione” e “Utente” è stata inglobata nell’entità “Indirizzo Spedizione” mediante l’attributo “RefUtente”.
6. La relazione “Ha” tra le entità “Indirizzo Fatturazione” e “Utente” è stata inglobata nell’entità “Indirizzo Fatturazione” mediante l’attributo “RefUtente”.
7. La relazione “Spedito Presso” tra le entità “Acquisto” e “Indirizzo Spedizione” è stata inglobata nell’entità “Acquisto” mediante l’attributo “RefSpedizione”.
8. La relazione “Con” tra le entità “Acquisto” e “Indirizzo Fatturazione” è stata inglobata nell’entità “Acuisto” mediante l’attributo “RefFatturazione”.
9. La relazione “Effettua” tra le entità “Acquisto” e “Utente” è stata inglobata nell’entità “Acquisto” mediante l’attributo “RefUtente”.
10. La relazione “Pagagto con” tra le entità “Acquisto” e “Carta” è stata inglobata nell’entità “Acquisto” mediante l’attributo “RefCarta”.
11. La relazione “Spedito Tramite” tra le entità “Acquisto” e “Corriere” è stata inglobata nell’entità “Acquisto” mediante l’attributo “RefCorriere”.
12. La relazione “Ha” tra le entità “Carta” e “Tipo Carta” è stata inglobata nell’entità “Carta” mediante l’attributo “RefTipo”.

Data la presenza di pochi attributi nell’entità “Carrello” si è preferito inglobare la relazione “Costituito da” tra le entità “Carrello” e “Prodotto” dentro l’entità “Carrello” mediante l’aggiunta dell’attributo “RefProdotto” e “Quantità”.

La relazione “Costituito da” tra le entità “Acquisto” e “Prodotto” è stata rinominata nella relazione “Info Acquisto” aggiungendo, inoltre, l’attributo “RefUtente” dentro tale relazione a causa del vincolo di integrità referenziale tra le entità “Acquisto” e “Utente”.

La base di dati risultante è quindi la seguente:

**Entità:**

**Acquisto** (CodAcquisto, RefSpedizione, RefFatturazione, RefCorriere, RefCarta, RefUtente, Data, Sconto)

**Carrello** (RefUtente, RefProdotto, Quantità)

**Carta** (CodCarta, RefUtente, RefTipo, NumeroCarta)

**Categoria Prodotti** (CodCategoria, NomeCategoria)

**Corriere** (CodCorriere, Nome, PrezzoSpedizione)

**Indirizzo** **Fatturazione** (CodFatturazione, RefUtente, Indirizzo, Citta, CAP, Preferito)

**Indirizzo** **Spedizione** (CodSpedizione, RefUtente, Indirizzo, Citta, CAP, Preferito)

**Permesso** (CodPermesso, NomePermesso)

**Prodotto** (CodProdotto, NomeProdotto, RefCategoria, Prezzo, Quantità, Descrizione, HasImage)

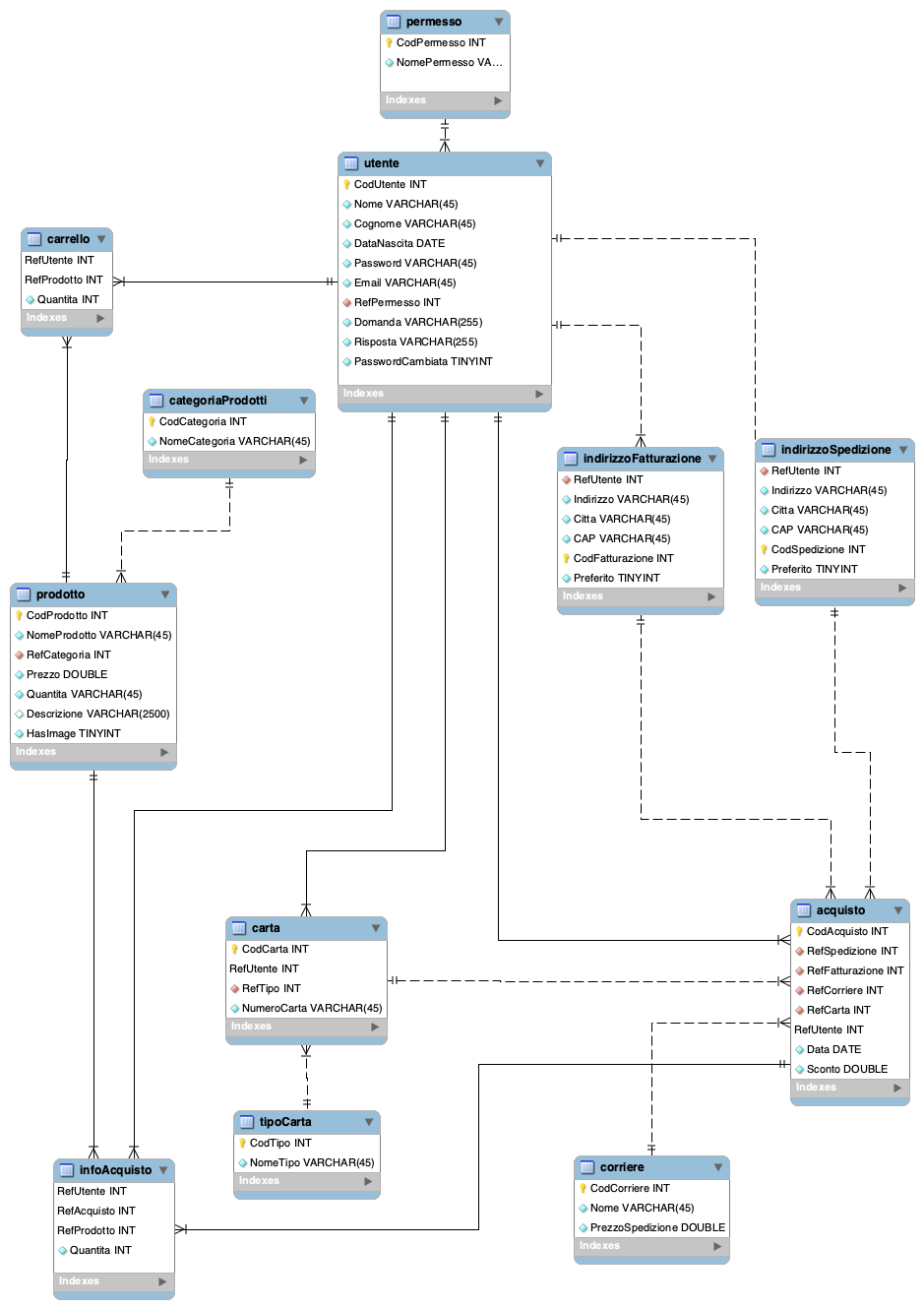
**Tipo** **Carta** (CodTipo, NomeTipo)

**Utente** (CodUtente, Nome, Cognome, DataNascita, Password, Email, RefPermesso, Domanda, Risposta, PasswordCambiata)

**Relazioni:**

**Info** **Acquisto** (RefUtente, RefAcquisto, RefProdotto, Quantità)

Lo schema E-R risultante è quello presente nella pagina successiva:

****

1. **Descrizione di ogni tabella e della scelta del tipo di dato per ogni campo.**

**Acquisto**

Tabella utilizzata per la memorizzazione dei dati di un acquisto la cui chiave primaria è la coppia di attributi ‘CodAcquisto’ e ‘RefUtente’, in questo modo ad ogni utente possono corrispondere da 1 ad N acquisti.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodAcquisto | Int | (PK)Codice che identifica un acquisto |
| RefSpedizione | Int | (NN)Chiave esterna all’attributo CodSpedizone della tabella indirizzoSpedizione |
| RefFatturazione | Int | (NN)Chiave esterna all’attributo CodFatturazione della tabella indirizzoFatturazione |
| RefCorriere | Int | (NN)Chiave esterna all’attributo ‘CodCorriere’ della tabella corriere |
| RefCarta | Int | (NN)Chiave esterna all’attributo ‘CodCarta’ della tabella carta |
| RefUtente | Int | (PK)Chiave esterna all’attributo CodUtente della tabella utente |
| Data | Data | (NN)Data dell’acquisto nel formato AAAA-MM-GG |
| Sconto | Double | (NN) Flag che indica se è stato applicato o meno uno sconto all’acquisto |

**Carrello**

Tabella utilizzata per tenere memoria degli elementi che vengono aggiunti al Carrello la cui chiave primaria è la coppia ‘RefUtente’ e ‘RefProdotto’.

Al momento dell’acquisto vengono prelevati i dati dal carrello per essere memorizzati nella tabella infoAcquisto.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| RefUtente | Int | (PK)Chiave esterna all’attributo CodUtente della tabella utente |
| RefProdotto | Int | (PK)Chiave esterna all’attributo CodProdotto della tabella prodotto |
| Quantita | Int | Numero di occorrenze aggiunte al carrello di uno stesso prodotto |

**Carta**

Tabella utilizzata per memorizzare i dati relativi ad u metodo di pagamento inserito dall’utente, la cui chiave primaria è la coppia ‘CodCarta’ e ‘RefUtente’.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodCarta | Int | (PK) Codice che identifica una carta |
| RefUtente | Int | (PK) Chiave esterna all’attributo CodUtente della tabella utente |
| RefTipo | Int | (NN) Chiave esterna all’attributo ‘CodTipo’ della tabella tipoCarta |
| NumeroCarta | Varchar | (NN)Numero identificativo della carta |

**Categoria Prodotti**

Tabella che contiene tutte le categorie dei prodotti venduti dall’E-commerce, la cui chiave primaria è ‘CodCategoria’.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodCategoria | Int | (PK)Codice che identifica una categoria |
| NomeCategoria | Varchar | (NN)Nome della categoria  (es. RAM, processori) |

**Corriere**

Tabella utilizzata per tenere memoria dei dati relativi ai corrieri, la cui chiave primaria è ‘CodCorriere’.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodCorriere | Int | (PK)Codice che identifica una categoria |
| Nome | Varchar | (NN)Nome del corriere (es. TNT, UPS) |
| PrezzoSpedizione | Double | (NN)Costo della spedizione di un determinato corriere |

**Indirizzo** **Fatturazione**

Tabella utilizzata per tenere memoria dei dati degli indirizzi di fatturazione degli utenti, la cui chiave primaria è la coppia ‘CodFatturazione’ e ‘RefUtente’.

Contiene tutti i dati dell’indirizzo di fatturazione, inoltre è presente un attributo ‘Preferito’ di tipo boolean per indicare se l’indirizzo di fatturazione è il preferito dell’utente.

Nel caso un utente inserisca per la prima volta un indirizzo di fatturazione, questo verrà impostato come preferito fino a quando l’utente non ne imposterà un altro.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodFatturazione | Int | (PK)Codice che identifica un indirizzo di fatturazione |
| RefUtente | Int | (PK)Chiave esterna all’attributo CodUtente della tabella utente |
| Indirizzo | Varchar | (NN)Indirizzo di fatturazione. (es. Via Balsamo) |
| Citta | Varchar | (NN)Città dell’indirizzo di fatturazione |
| CAP | Varchar | (NN)CAP dell’indirizzo di fatturazione |
| Preferito | TinyInt - Boolean | (NN)Flag che indica se un determinato indirizzo sia quello preferito dell’utente |

**Info** **Acquisto**

Tabella utilizzata per tenere memoria degli acquisti effettuati dall’utente la cui chiave primaria è la tripla ‘RefUtente’, ‘RefAcquisto’ e ‘RefProdotto’.

L’esigenza di creare questa tabella nasce per permettere che ad uno stesso acquisto possano essere associati più prodotti, inoltre rende più semplice l’implementazione di uno storico degli acquisti dell’utente.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| RefUtente | Int | (PK)Chiave esterna all’attributo CodUtente della tabella utente |
| RefAcquisto | Int | (PK)Chiave esterna all’attributo CodAcquisto della tabella acquisto |
| RefProdotto | Int | (PK)Chiave esterna all’attributo CodProdotto della tabella prodotto |
| Quantita | Int | (NN)Numero di occorrenze di uno stesso prodotto |

**Indirizzo** **Spedizione**

Tabella utilizzata per tenere memoria dei dati degli indirizzi di spedizione degli utenti, la cui chiave primaria è la coppia ‘CodSpedizione’ e ‘RefUtente’.

Contiene tutti i dati dell’indirizzo di spedizione, inoltre è presente un attributo ‘Preferito’ di tipo boolean per indicare se l’indirizzo di spedizione è il preferito dell’utente.

Nel caso un utente inserisca per la prima volta un indirizzo di spedizione, questo verrà impostato come preferito fino a quando l’utente non ne imposterà un altro.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodSpedizione | Int | (PK)Codice che identifica un indirizzo di fatturazione |
| RefUtente | Int | (PK)Chiave esterna all’attributo CodUtente della tabella utente |
| Indirizzo | Varchar | (NN)Indirizzo di fatturazione (es. Via Balsamo) |
| Citta | Varchar | (NN)Città dell’indirizzo di fatturazione |
| CAP | Varchar | (NN)CAP dell’indirizzo di fatturazione |
| Preferito | TinyInt - Boolean | (NN)Flag che indica se un determinato indirizzo sia quello preferito dell’utente |

**Permesso**

Tabella utilizzata per identificare i privilegi di accesso degli utenti.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodPermesso | Int | (PK)Codice che identifica un permesso |
| NomePermesso | Varchar | (NN)Nome del tipo di permesso  (es. Amministratore) |

**Prodotto**

Tabella utilizzata per memorizzare i prodotti acquistabili nel sito, la cui chiave primaria è ‘CodProdotto’.

L’attributo ‘HasImage’ specifica se un determinato prodotto presenta un’immagine o meno, la sua aggiunta è stata ritenuta necessaria per permettere una maggiore completezza, efficienza e sicurezza, nelle fasi di aggiunta, modifica e ricerca di un prodotto.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodProdotto | Int | (PK)Codice che identifica un prodotto |
| NomeProdotto | Varchar | (NN)Nome del prodotto |
| RefCategoria | Int | (NN)Chiave esterna all’attributo CodCategoria della tabella categoriaProdotti |
| Prezzo | Double | (NN)Prezzo del prodotto |
| Quantita | Varchar | (NN)Numero di occorrenze presenti in magazzino di uno stesso prodotto |
| Descrizione | Varchar | Descrizione del prodotto  (MAX 2500 caratteri) |
| HasImage | TinyInt - Boolean | (NN)Flag che identifica la presenza o meno di un’immagine associata al prodotto |

**Tipo** **Carta**

Tabella utilizzata per tenere memoria dei dati relativi alle carte accettate dall’E-commerce, la cui chiave primaria è ‘CodTipo’.

È stata ritenuta necessaria questa tabella per assicurarsi che l’utente non aggiunga metodi di pagamento non accettati dall’E-commerce.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodTipo | Int | (PK)Codice che identifica una carta |
| NomeTipo | Varchar | (NN)Nome della carta (es. Visa) |

**Utente**

Tabella utilizzata per memorizzare i dati degli account degli utenti, la cui chiave primaria è ‘CodUtente’.

L’attributo ‘PasswordCambiata’ serve per indicare se un amministratore ha cambiato la password di un utente, in questo caso l’utente, al prossimo accesso, viene indirizzato automaticamente alla pagina per la modifica della password.

|  |  |  |
| --- | --- | --- |
| **Nome Attributo** | **Tipo Attributo** | **Motivo** |
| CodUtente | Int | (PK)Codice che identifica una carta |
| Nome | Varchar | (NN)Nome dell’utente |
| Cognome | Varchar | (NN)Cognome dell’utente |
| DataNascita | Date | (NN)Data di nascita dell’utente nel formato:  AAAA-MM-GG |
| Password | Varchar | (NN)Password dell’account |
| Email | Varchar | (NN)Email dell’utente |
| RefPermesso | Int | (NN)Chiave esterna all’attributo ‘CodPermesso’ della tabella permesso |
| Domanda | Varchar | (NN)Domanda impostata dall’utente per la modifica della password dell’utente |
| Risposta | Varchar | (NN)Risposta impostata dall’utente per la modifica della password dell’utente |
| PasswordCambiata | TinyInt - Boolean | (NN)Flag che indica se un amministratore ha modificato la password dell’utente |

1. **Lista query utilizzate**

**Indice**

1. **Classi**
   1. Corriere.php
   2. indirizzoSpedizione.php
   3. indirizzoFatturazione.php
   4. Carta.php
   5. Utente.php
   6. Prodotto.php
   7. Ordine.hp
2. **Admin**
   1. ModifyAccount.php
   2. ModifyProduct.php
   3. VisualizeProduct

1. **User**
   1. NewCarta.php
   2. SelectCorriere.php
2. **General**
   1. CategoriaSelezionata.php
   2. FooterIndex.php
   3. Header.php
   4. Index.php
   5. ResetPassword.php
   6. Search.php
   7. Command.php

**Classi:**

**Corriere.php**

1. Query utilizzata all’interno del **costruttore** della classe corriere per impostare i valori delle variabili:

”SELECT Nome, PrezzoSpedizione FROM corriere WHERE codCorriere = “.$this->cod;

1. Query utilizzata nel metodo **addCorriereIntoDb** della classe Corriere per verificare che non ci siano altri corrieri con lo stesso nome:

"SELECT \* FROM corriere WHERE NomeCorriere = “.$nomeCorriere;

1. Query utilizzata nel metodo **addCorriereIntoDb** della classe Corriere per impostare il valore del codice del nuovo corriere da aggiungere:

"SELECT COUNT(\*) FROM corriere”;

1. Query utilizzata nel metodo **addCorriereIntoDb** della classe Corriere per inserire il corriere nel database:

"INSERT INTO Corriere (CodCorriere, Nome, PrezzoSpedizione) VALUES (".$codCorriere.", '".$nomeCorriere."', “.$prezzoSpedizione.”)";

**indirizzoSpedizione.php**

1. Query utilizzata nel **costruttore** della classe indirizzoSpedizione per impostare i valori delle variabili:

”SELECT Indirizzo, Citta, CAP, Preferito FROM indirizzoSpedizione WHERE CodSpedizione = ".$this->codIndirizzo." AND refUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoSpedizioneIntoDB** della classe indirizzoSpedizione per verificare che quell’indirizzo sia già stato inserito:

"SELECT \* FROM indirizzoSpedizione WHERE Indirizzo = '".$indirizzo."' AND Citta = '".$citta."' AND Cap = '".$cap."' AND RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoSpedizioneIntoDB** della classe indirizzoSpedizione per impostare il valore del codice dell’indirizzo da inserire:

"SELECT COUNT(\*) FROM indirizzoSpedizione WHERE RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoSpedizioneIntoDB** della classe indirizzoSpedizione per inserire l’indirizzo nel database:

"INSERT INTO indirizzoSpedizione (refUtente, Indirizzo, Citta, cap, codSpedizione, Preferito) VALUES (“.$codUtente.", '".$indirizzo."', '".$citta."', '".$cap."', ".$codice.", “.$preferito.")";

1. Query utilizzata nel metodo **deleteFromDB** della classe indirizzoSpedizione per eliminare l’indirizzo dal database:

"DELETE FROM indirizzoSpedizione WHERE CodSpedizione = ".$codSpedizione." AND RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **setIndirizzo** della classe indirizzoSpedizione per impostare il valore nel database:

"UPDATE indirizzoSpedizione SET Indirizzo = '".$newIndirizzo."' WHERE CodSpedizione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setCitta** della classe indirizzoSpedizione per impostare il valore nel database:

"UPDATE indirizzoSpedizione SET citta = '".$newCitta."' WHERE CodSpedizione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setCap** della classe indirizzoSpedizione per impostare il valore nel database:

"UPDATE indirizzoSpedizione SET cap = '".$newCap."' WHERE CodSpedizione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setDefault** della classe indirizzoSpedizione per impostare il valore nel database:

"UPDATE indirizzoSpedizione SET preferito = '".$preferito."' WHERE CodSpedizione = ".$this->codIndirizzo." AND refUtente = ".$this->codUtente;

**indirizzoFatturazione.php**

1. Query utilizzata nel **costruttore** della classe indirizzoFatturazione per impostare i valori delle variabili:

”SELECT Indirizzo, Citta, CAP, Preferito FROM indirizzoFatturazione WHERE CodFatturazione = ".$this->codIndirizzo." AND refUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoFatturazioneIntoDB** della classe indirizzoFatturazione per verificare che quell’indirizzo non sia già stato inserito:

"SELECT \* FROM indirizzoFatturazione WHERE Indirizzo = '".$indirizzo."' AND Citta = '".$citta."' AND Cap = '".$cap."' AND RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoFatturazioneIntoDB** della classe indirizzoFatturazione per impostare il valore del codice dell’indirizzo da inserire:

"SELECT COUNT(\*) FROM indirizzoFatturazione WHERE RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addIndirizzoFatturazioneIntoDB** della classe indirizzoFatturazione per inserire l’indirizzo nel database:

"INSERT INTO indirizzoFatturazione (refUtente, Indirizzo, Citta, cap, codFatturazione, Preferito) VALUES (".$codUtente.", '".$indirizzo."', '".$citta."', '".$cap."', ".$codice.", “.$preferito.")";

1. Query utilizzata nel metodo **deleteFromDB** della classe indirizzoFatturazione per cancellare l’indirizzo dal database:

"DELETE FROM indirizzoFatturazione WHERE CodFatturazione = ".$codFatturazione." AND RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **setIndirizzo** della classe indirizzoFatturazione per impostare il valore nel database:

"UPDATE indirizzoFatturazione SET Indirizzo = '".$newIndirizzo."' WHERE CodFatturazione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setCitta** della classe indirizzoFatturazione per impostare il valore nel database:

"UPDATE indirizzoFatturazione SET citta = '".$newCitta."' WHERE CodFatturazione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setCap** della classe indirizzoFatturazione per impostare il valore nel database:

"UPDATE indirizzoFatturazione SET cap = '".$newCap."' WHERE CodFatturazione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **setDefault** della classe indirizzoFatturazione per impostare il valore nel database:

"UPDATE indirizzoFatturazione SET preferito = '".$preferito."' WHERE CodFatturazione = ".$this->codIndirizzo." AND refUtente = “.$this->codUtente;

**Carta.php**

1. Query utilizzata nel **costruttore** della classe Carta per impostare i valori delle variabili:

”SELECT CodTipo, NomeTipo, NumeroCarta FROM carta, tipoCarta WHERE refTipo = codTipo AND CodCarta = ".$this->codCarta." AND refUtente = “.$this->codUtente;

1. Query utilizzata nel metodo **addCartaIntoDb** della classe Carta per verificare se quell’utente ha inserito già una carta con lo stesso numero:

"SELECT \* FROM carta WHERE Numero = '".$numero."', AND codTipo = ".$codTipo." AND RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addCartaIntoDb** della classe Carta per impostare il valore del codice della carta da inserire nel database:

SELECT COUNT(\*) FROM carta WHERE RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **addCartaIntoDb** della classe Carta per inserire la carta nel database:

"INSERT INTO carta (CodCarta, RefUtente, RefTipo, NumeroCarta) VALUES (".$codCarta.", ".$codUtente.", ".$codTipo.", ‘“.$numero."')";

1. Query utilizzata nel metodo **deleteCarta** della classe Utente per venire a conoscenza di quante carte ha inserito l’utente:

"SELECT COUNT(\*) FROM carta WHERE RefUtente = “.$codUtente;

1. Query utilizzata nel metodo **deleteCarta** della classe Utente per cancellare nel database la carte dell’utente:

"DELETE FROM carta WHERE RefUtente = ".$codUtente." AND CodCarta = “.$codCarta."";

1. Query utilizzata all’interno di un ciclo for nel metodo **deleteCarta** della classe Utente per diminuire di 1 i codici delle altre carte relative a quell’utente:

"UPDATE carta SET CodCarta = ".($i-1)." WHERE CodCarta = “.$i."";

**Utente.php**

1. Query utilizzata nel **costruttore** della classe Utente per impostare i valori delle variabili:

”SELECT Password, Nome, Cognome, dataNascita, email, refpermesso, domanda, risposta, PasswordCambiata FROM utente WHERE codUtente = “.$this->id;

1. Query utilizzata nel **costruttore** della classe utente per impostare i valori dell’array “Carrello”:

”SELECT codProdotto, carrello.quantita FROM carrello, prodotto WHERE RefProdotto = codProdotto AND refutente = “.$this->id;

1. Query utilizzata nel **costruttore** della classe Utente per impostare il numero totale di indirizzi di fatturazione:

"SELECT COUNT(\*) FROM indirizzoFatturazione WHERE refutente = “.$this->id;

1. Query utilizzata nel **costruttore** della classe Utente per impostare il numero totale di indirizzi di spedizione:

"SELECT COUNT(\*) FROM indirizzoSpedizione WHERE refutente = “.$this->id;

1. Query utilizzata nel **costruttore** della classe Utente per impostare il numero totale di carte di credito:

"SELECT COUNT(\*) FROM carta WHERE refutente = “.$this->id;

1. Query utilizzata nel **costruttore** della classe Utente per impostare il numero totale di ordini effettuati:

"SELECT COUNT(\*) FROM acquisto WHERE refutente = “.$this->id;

1. Query utilizzata nel metodo **getIndirizzoFatturazionePreferito** della classe utente per cercare il codice relativo all’indirizzo di fatturazione preferito dall’utente:

"SELECT CodFatturazione FROM indirizzoFatturazione WHERE RefUtente = ".$this->id." AND Preferito = 1”;

1. Query utilizzata nel metodo **getIndirizzoSpedizionePreferito** della classe utente per cercare il codice relativo all’indirizzo di spedizione preferito dall’utente:

"SELECT CodSpedizione FROM indirizzoSpedizione WHERE RefUtente = ".$this->id." AND Preferito = 1”;

1. Query utilizzata nel metodo **setNome** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET nome = '".$newNome."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setCognome** della classe utente per aggiornare i valori nel database:

”UPDATE utente SET cognome = '".$newCognome."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setNascita** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET dataNascita = '".$newNascita."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setEmail** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET email = '".$newEmail."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setPermesso** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET refPermesso = '".$newPermesso."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setDomanda** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET domanda = '".$newDomanda."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setRisposta** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET risposta = '".$newRisposta."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setPasswordCambiata** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET passwordCambiata = '".$passCambiata."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **setPassword** della classe utente per aggiornare i valori nel database:

"UPDATE utente SET password = '".$newPassword."' WHERE codUtente = “.$this->id;

1. Query utilizzata nel metodo **aggiungiProdottoAlCarrello** della classe Utente per cercare la quantità presente di quel prodotto nel carrello:

”SELECT Quantita FROM Carrello WHERE RefUtente = ".$this->id." AND RefProdotto = “.$prodotto->getCodProdotto();

1. Query utilizzata nel metodo **aggiungiProdottoAlCarrello** della classe utente per aggiornare nel database la quantità presente di quel prodotto nel carrello:

”UPDATE Carrello SET Quantita = ".$newQuantita." WHERE RefUtente = ".$this->id." AND RefProdotto = “.$prodotto->getCodProdotto();

1. Query utilizzata nel metodo **aggiungiProdottoAlCarrello** della classe Utente per inserire il prodotto nel carrello:

”INSERT INTO Carrello VALUES (".$this->id.", ".$prodotto->getCodProdotto().", “.$quantita.")";

1. Query utilizzata nel metodo **isProdottoNelCarrello** della classe utente per verificare se quel prodotto è nel carrello:

"SELECT \* FROM Carrello WHERE RefUtente = ".$this->id." AND RefProdotto = “.$prodotto->getCodProdotto();

1. Query utilizzata nel metodo **setQuantitaProdottoCarrello** della classe utente per aggiornare nel database il valore della quantità di un prodotto presente nel carrello:

"UPDATE Carrello SET Quantita = ".$quantita." WHERE RefUtente = ".$this->id." AND RefProdotto = “.$codProdotto;

1. Query utilizzata nel metodo **rimuoviDalCarrello** della classe utente per eliminare nel database un prodotto dal carrello:

"DELETE FROM carrello WHERE RefProdotto = ".$prodotto->getCodProdotto()." AND RefUtente = “.$this->id;

1. Query utilizzata nel metodo **creaOrdine** della classe utente per impostare il valore di CodAcquisto del prossimo ordin da inserire:

"SELECT COUNT(\*) FROM acquisto WHERE RefUtente = “.$this->getCod();

1. Query utilizzata nel metodo **creaOrdine** della classe utente per inserire l’acquisto nel database:

”INSERT INTO acquisto VALUES (".$codAcquisto.", ".$codIndirizzoSpedizione.", ".$codIndirizzoFatturazione.", ".$codCorriere.", ".$codCarta.", ".$this->getCod().", '".$data."', “.$sconto.")";

1. Query utilizzata nel metodo **creaOrdine** della classe Utente per inserire nel database i prodotti acquistati nell’ordine:

”INSERT INTO infoAcquisto VALUES (".$this->getCod().", ".$codAcquisto.", ".$prodotto->getCodProdotto().", ".$this->carrello[$i][1].")";

**Prodotto.php**

1. Quei utilizzata nel **costruttore** della classe Prodotto per impostare le variabili:

”SELECT NomeProdotto, CodCategoria, NomeCategoria, Prezzo, Quantita, Descrizione, HasImage FROM prodotto, categoriaProdotti WHERE refCategoria = codCategoria AND CodProdotto = “.$this->codProdotto;

1. Query utilizzata nel metodo **addProductIntoDB** della classe Prodotto per verificare che non ci siano altri prodotti con lo stesso nome:

"SELECT \* FROM prodotto WHERE NomeProdotto = ".$nomeProdotto;

1. Query utilizzata nel metodo **addProductIntoDB** della classe Prodotto per impostare il valore di “CodProdotto” per il nuovo prodotto da inserire:

”SELECT COUNT(\*) FROM prodotto”;

1. Query utilizzata nel metodo **addProductIntoDB** della classe Prodotto per inserire il prodotto nel database:

”INSERT INTO prodotto(CodProdotto, NomeProdotto,RefCategoria,Prezzo,Quantita,Descrizione) VALUES(".($lastCode).", ‘“.$nomeProdotto."','".$codCategoria."','".$prezzo."','".$quantita."','".$descrizione."')";

1. Query utilizzata nel medoto **setNomeProdotto** della classe prodotto per aggiornare il valore nel database:

"UPDATE prodotto SET NomeProdotto = '".$newNomeProdotto."' WHERE CodProdotto = “.$this->getCodProdotto();

1. Query utilizzata nel metodo **setCodCategoria** della classe prodotto per trovare il nome della categoria riferita a quel codice:

"SELECT NomeCategoria FROM CategoriaProdotti WHERE CodCategoria = ‘".$newCodCategoria;

1. Query utilizzata nel metodo **setCodCategoria** della classe prodotto per aggiornare i valori nel database:

"UPDATE prodotto SET RefCategoria = '".$newCodCategoria."' WHERE CodProdotto = “.$this->getCodProdotto();

1. Query utilizzata nel metodo **setNomeCategoria** della classe prodotto per trovare il codice della categoria riferita a quel nome:

"SELECT CodCategoria FROM CategoriaProdotti WHERE NomeCategoria = ‘".$newNomeCategoria;

1. Query utilizzata nel metodo **setPrezzo** della classe prodotto per aggiornare il valore nel database:

"UPDATE prodotto SET Prezzo = '".$newPrezzo."' WHERE CodProdotto = “.$this->getCodProdotto();

1. Query utilizzata nel metodo **setQuantita** della classe prodotto per aggiornare il valore nel database:

"UPDATE prodotto SET Quantita = '".$newQuantita."' WHERE CodProdotto = “.$this->getCodProdotto();

1. Query utilizzata nel metodo **setDescrizione** della classe prodotto per aggiornare il valore nel database:

"UPDATE prodotto SET descrizione = '".$newDescrizione."' WHERE CodProdotto = “.$this->getCodProdotto();

1. Query utilizzata nel metodo **setHasImage** della classe prodotto per aggiornare il valore nel database:

”UPDATE prodotto SET HasImage = '".$hasImage."' WHERE CodProdotto = ".$this->getCodProdotto()."";

**Ordine.php**

1. Query utilizzata nel **costruttore** della classe ordine per impostare le variabili:

”SELECT RefSpedizione, RefFatturazione, RefCorriere, RefCarta, Data, Sconto FROM acquisto WHERE codAcquisto = ".$codAcquisto." AND RefUtente = “.$codUtente;

**Admin:**

**modifyAccount.php**

1. Query che restituisce il codice utente in base alla mail inserita.

"SELECT CodUtente FROM utente where Email = '".$\_POST['EmailUtente']."'";

**modifyProduct.php**

1. Query utilizzata per cercare un prodotto in base al nome inserito.

"SELECT CodProdotto FROM prodotto WHERE NomeProdotto LIKE '%".$\_GET["NomePrecedenteProdotto"]."%'";

**visualizeProduct.php**

1. Query che restituisce il codice utente in base alla mail inserita.

"SELECT CodUtente FROM utente where Email = '".$\_POST['EmailUtente']."'";

**User:**

**newCarta.php**

1. Query utilizzata per mostrare l’elenco delle tipologie di carta disponibili nel form per l’inserimento di un nuovo metodo di pagamento.

"SELECT NomeTipo, CodTipo FROM tipoCarta ORDER BY CodTipo";

**selectCorriere.php**

1. Query utilizzata per mostrare l’elenco dei corrieri disponibili nel Database.

"SELECT COUNT(\*) FROM corriere";

**General:**

**categoriaSelezionata.php**

1. Query utilizzata per mostrare i prodotti appartenenti alla categoria selezionata dall’utente dal menù a tendina del header.

"SELECT codProdotto FROM prodotto, categoriaProdotti WHERE refCategoria = codCategoria AND nomeCategoria = '".$\_GET["nomeCategoria"]."'";

**footerIndex.php**

1. Query utilizzata per calcolare il numero di prodotti presenti nel Database.

"SELECT COUNT(\*) FROM prodotto";

**header.php**

1. Query che restituisce l’elenco delle categorie prodotto presenti nel Database.

"SELECT NomeCategoria FROM categoriaProdotti";

**index.php**

1. Query utilizzata per contare il numero di prodotti presenti per poi gestirne la visualizzazione.

"SELECT COUNT(\*) FROM prodotto";

**resetPassword.php**

1. Query utilizzata per cercare un utente in base alla mail inserita per effettuare il reset della password.

"SELECT\* FROM utente where Email = '".$\_POST['EmailUtente']."'";

**search.php**

1. Query utilizzata per effettuare la ricerca per nome di un prodotto nel campo “cerca” del sito, restituisce il codice del prodotto in base al nome e alla categoria specificata.

"SELECT CodProdotto FROM prodotto, categoriaProdotti WHERE RefCategoria = CodCategoria AND NomeProdotto LIKE '%".$\_GET["NomeProdotto"]."%'";

**Command.php**

1. Query utilizzata durante la **fase di login** per verificare se esiste un utente con la mail e password inserite dall’utente:

“SELECT codUtente FROM utente WHERE email = '".$\_POST[email]. "' AND Password = '" .$\_POST[password]. “'";

1. Query utilizzata durante la **fase di inserimento di un nuovo prodotto** per creare l’oggetto prodotto per impostare la variabile “HasImage”:

“SELECT CodProdotto FROM prodotto WHERE NomeProdotto = ‘".$\_POST['NomeProdotto']."'";

1. Query utilizzata durante la **fase di registrazione** per verificare se esiste già un account con l’e-mail inserite dall’utente:

"SELECT \* FROM utente WHERE Email = ‘".$\_POST['Email'];

1. Query utilizzata per trovare il codice dell’indirizzo di fatturazione di default dell’utente:

"SELECT CodFatturazione FROM indirizzoFatturazione WHERE Preferito = 1”;

1. Query utilizzata per trovare il codice dell’indirizzo di spedizione di default dell’utente:

SELECT CodSpedizione FROM indirizzoSpedizione WHERE Preferito = 1";