|  |
| --- |
| Progetto Sito Web Basi di Dati |
| Negozio di videogiochi “VestaGames” |
|  |
| Il sito è visitabile all’url: “www.vestagames.it”. Hostato sui servizi Aruba |
|  |

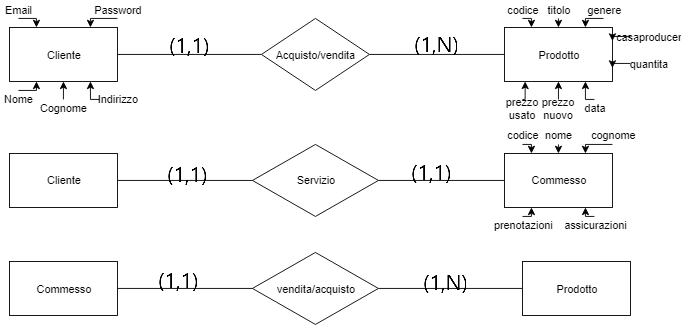
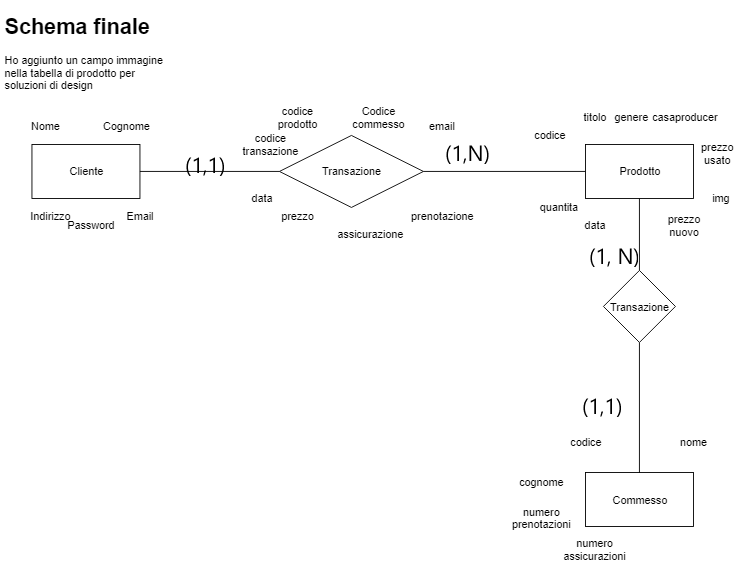
Progetto Sito Web Basi di Dati

Negozio di videogiochi “VestaGames”

# Requisiti

Si vuole realizzare un sito di un negozio di videogiochi che gestisce l’inventario, le prenotazioni, i clienti, le vendite e i commessi.

Ogni cliente può essere registrato o non registrato. Se registrato, l’utente è identificato da nome, cognome, indirizzo ed email. Un cliente può comprare o prenotare un videogioco a seconda della disponibilità, può prenotare solo registrato. I videogiochi, che possono essere nuovi o usati, sono descritti dal titolo, genere, prezzo, casa producer, data di uscita, quantità in negozio, codice prodotto e prezzo per chi lo porta usato in negozio. Ogni transazione (prenotazione, acquisto o vendita) deve essere registrata, si deve sapere quale commesso ha servito il cliente e se il prodotto è stato assicurato. I commessi sono tenuti a cercare di proporre un'assicurazione sul prodotto e vengono valutati in base a quante ne riescono a fare, stessa cosa per le prenotazioni dei titoli in uscita. Ogni commesso è identificato da nome, cognome, email codice commesso e quantità di assicurazioni fatte e prenotazioni fatte.

Schema Entità-Relazione:

Cliente(Email, Nome, Cognome, indirizzo, password)  
Commesso(codice\_commesso, nome, cognome, nprenotazioni, nassicurazioni)  
Ordini(codice\_transazione, codice\_prodotto, codice\_commesso, prenotazione, assicurazione,prezzo, email, Data)  
Prodotto(codice\_prodotto, Titolo, genere, Casa\_producer, quantita, data\_uscita, prezzo\_nuovo, prezzo usato, img)

NOTA: In Ordini codice\_prodotto e codice\_commesso sono foreign key e si riferiscono rispettivamente a codice\_prodotto in Prodotto e codice\_commesso in Commesso.

I codici identificativi sono stati impostati con la chiave AUTO\_INCREMENT in modo che durante una qualsiasi insert se non viene inserito alcun valore in quel campo, phpMyAdmin lo aggiunge da solo prendendo il valore dalla riga più grande tra quelle esistenti nella tabella e l’aumenta di 1.

I valori che riguardano le Date sono stati memorizzati come interi in modo da facilitare gli ordinamenti con le query. Il codice PHP del sito si occupa di manipolarle e trasformarle nel formato dd-mm-yyyy o viceversa a seconda della necessità

# Descrizione sito

Ho acquistato un pacchetto di web hosting dal fornitore Aruba.

I codici identificativi sono stati impostati con la chiave AUTO\_INCREMENT in modo che durante una qualsiasi insert se non viene inserito alcun valore in quel campo, phpMyAdmin lo aggiunge da solo prendendo il valore dalla riga più grande tra quelle esistenti nella tabella e l’aumenta di 1.

I valori che riguardano le Date nelle tabelle sono stati memorizzati come interi in modo da facilitare gli ordinamenti con le query. Il codice PHP del sito si occupa di manipolarli e trasformarli nel formato dd-mm-yyyy o viceversa a seconda della necessità.

Il sito è pensato per simulare un sistema che gestisce un negozio di videogiochi.

Nella homepage puoi: controllare il catalogo (eseguendo una ricerca attraverso il motore di ricerca che è stato implementato attraverso le funzioni mysql MATCH AGAINST), ordinarlo in base al prezzo o alla data di uscita più recente, effettuare il login e registrarsi. Dopo aver scelto un titolo il sito ti indirizza ad una pagina dove ti vengono fornite le info del gioco e hai la possibilità di venderlo, acquistarlo oppure prenotarlo a seconda della disponibilità (Se non è uscito non puoi venderlo né comprarlo, solo prenotarlo. Se è terminato puoi prenotarlo oppure venderlo). Ogni volta che viene eseguita una transazione viene preso a caso un commesso dalla tabella Commesso, viene preso il suo codice e inserito, insieme ai dati del prodotto e del cliente, in una nuova tupla nella tabella Ordini.   
Il sito è fornito di un pannello amministratore (accessibile con le giuste credenziali nel login), dove un ipotetico gestore del negozio può controllare, cancellare o aggiungere tutti i dati relativi all’inventario, i commessi, le transazioni e i clienti.

La parte grafica e di CSS è stata fatta utilizzando una libreria grafica chiamata Bootstrap.

## Elenco interrogazioni fatte per pagina:

NOTA: Alcune interrogazioni, quelle fatte con “?”, sono query fatte con lo statement di mysqli. Durante execute() dove c’è “?” viene messo il valore che gli viene assegnato durante il bind\_param().

Index.php:

* SELECT codice\_prodotto, Titolo, Genere, casa\_producer as 'Software House', prezzo\_nuovo, prezzo\_usato, data\_uscita FROM Prodotto WHERE MATCH(Titolo, Genere, casa\_producer) AGAINST(+? in boolean mode)
* SELECT codice\_prodotto, Titolo, Genere, casa\_producer, prezzo\_nuovo, prezzo\_usato, data\_uscita FROM Prodotto order by data\_uscita desc
* SELECT codice\_prodotto, Titolo, Genere, casa\_producer, prezzo\_nuovo, prezzo\_usato, data\_uscita FROM Prodotto order by prezzo\_nuovo desc
* SELECT codice\_prodotto, Titolo, Genere, casa\_producer,prezzo\_nuovo,prezzo\_usato, data\_uscita, img FROM Prodotto order by rand() limit 0,6

Login.php:

* SELECT Email, Nome, Password FROM Cliente WHERE Email=? AND Password=?

Registration.php:

* INSERT INTO Cliente VALUES (?, ?, ?, ?, ?)

Buy.php:

* SELECT \* FROM Prodotto where codice\_prodotto=?

Checkout.php (alcune query vengono ripetute a seconda del caso):

* UPDATE `Sql1182390\_1`.`Prodotto` SET `quantita` = quantita-1 WHERE `Prodotto`.`codice\_prodotto` = ?
* SELECT codice\_commesso from Commesso order by rand() limit 0,1
* UPDATE `Sql1182390\_1`.`Prodotto` SET `quantita` = quantita-1 WHERE `Prodotto`.`codice\_prodotto` = ?
* SELECT codice\_commesso from Commesso order by rand() limit 0,1
* INSERT INTO `Sql1182390\_1`.`Ordini` (`codice\_transazione` ,`codice\_prodotto` ,`codice\_commesso` ,`prenotazione` ,`assicurazione` ,`prezzo` ,`email` ,`Data`) VALUES (null,?,?,0,1,?,null,?)
* UPDATE `Sql1182390\_1`.`Commesso` SET `nassicurazioni` = nassicurazioni+1 WHERE codice\_commesso = ?
* UPDATE `Sql1182390\_1`.`Prodotto` SET `quantita` = quantita+1 WHERE `Prodotto`.`codice\_prodotto` = ?
* UPDATE `Sql1182390\_1`.`Commesso` SET `nprenotazioni` = nprenotazioni+1 WHERE codice\_commesso = ?

/panel/staff.php:

* SELECT \* FROM Commesso WHERE MATCH(Nome, Cognome) AGAINST(+? IN BOOLEAN MODE) OR codice\_commesso=?
* SELECT \* FROM Commesso order by (nprenotazioni+nassicurazioni) desc
* SELECT \* FROM Commesso

/panel/inventario.php:

* SELECT \* FROM Prodotto WHERE MATCH(Titolo, Genere, casa\_producer) AGAINST(+? IN BOOLEAN MODE) OR codice\_prodotto=?
* SELECT \* FROM Prodotto WHERE data\_uscita > ?
* SELECT \* FROM Prodotto WHERE data\_uscita < ?
* SELECT \* FROM Prodotto order by Titolo, Genere

/panel/transactions.php:

* SELECT \* FROM Ordini where codice\_transazione='.$\_GET['cercare']
* SELECT Prodotto.codice\_prodotto, Prodotto.titolo, Prodotto.genere, Prodotto.Casa\_producer, Prodotto.quantita, Prodotto.data\_uscita, Prodotto.prezzo\_nuovo,Prodotto.prezzo\_usato FROM Prodotto, Ordini WHERE Ordini.codice\_transazione=".$\_GET['cercare']." AND Ordini.codice\_prodotto=Prodotto.codice\_prodotto
* SELECT Commesso.codice\_Commesso, Commesso.nome, Commesso.Cognome, Commesso.nprenotazioni, Commesso.nassicurazioni FROM Commesso, Ordini WHERE Ordini.codice\_transazione=".$\_GET['cercare']." AND Ordini.codice\_commesso=Commesso.codice\_commesso
* SELECT Cliente.Email, Cliente.Nome, Cliente.cognome, Cliente.indirizzo FROM Cliente, Ordini WHERE Ordini.codice\_transazione=".$\_GET['cercare']." AND Ordini.email=Cliente.Email
* SELECT \* FROM Ordini order by Data desc

/panel/customers.php:

* SELECT Email, Nome, Cognome, Indirizzo FROM Cliente WHERE MATCH(Email,Nome, Cognome) AGAINST(+? IN BOOLEAN MODE)
* SELECT Email, Nome, Cognome, Indirizzo FROM Cliente order by Cognome, Nome

/panel/confirmed.php:

* DELETE FROM Commesso WHERE codice\_commesso=?
* DELETE FROM Cliente WHERE Email=?
* DELETE FROM Prodotto WHERE codice\_prodotto=?
* SELECT codice\_Commesso FROM Commesso WHERE codice\_Commesso=?
* INSERT INTO Commesso (`codice\_Commesso`, `Nome`, `Cognome`, `nprenotazioni`, `nassicurazioni`) VALUES (?, ?, ?, ?, ?)
* SELECT email FROM Cliente WHERE Email=?
* INSERT INTO Cliente (`Email`, `Nome`, `Cognome`, `indirizzo`, `password`) VALUES (?, ?, ?, ?, ?)
* SELECT titolo FROM Prodotto WHERE Titolo=?
* INSERT INTO `Prodotto` (`codice\_prodotto`, `Titolo`, `Genere`, `Casa\_producer`, `quantita`, `data\_uscita`, `prezzo\_nuovo`, `prezzo\_usato`, img) VALUES (NULL,?, ?, ?, ?, ?, ?, ?,?)
* INSERT INTO Ordini VALUES (NULL,?, ?, ?, ?, ?, ?, ?)