

1. Database MySQL

1.1. Analisi

In base alla consegna, il progetto richiede la creazione di 3 entità: operatori, clienti ed acquisti.

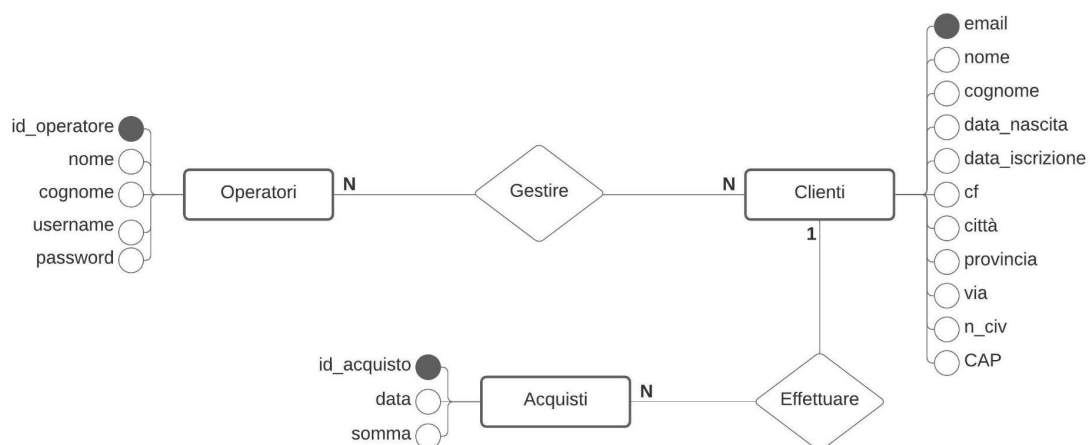
L'entità operatori contiene i dati relativi agli operatori che andranno ad utilizzare il portale e vengono inseriti da back-end, ha 5 attributi: id_operatore, nome, cognome, username e password.

In seguito, l'entità clienti contiene i dati anagrafici relativi ai clienti, ha 11 attributi: email, nome, cognome, data_nascita, data_iscrizione, cf, città, provincia, via, n_civ e CAP.

Infine, l'entità acquisti contiene i dati relativi agli acquisti effettuati dai clienti, ogni acquisto viene registrato come somma totale dell'acquisto e non viene suddiviso sui prodotti registrati, ha 3 attributi: id_acquisto, data e somma.

1.2. Modello E/R

Il primo passo della creazione di un database è sempre il modello concettuale, andiamo quindi a visualizzare il modello E/R.



Analizzando il modello E/R notiamo le relazioni tra le entità: tra "Operatori" e "Clienti" è presente una relazione N a N, sarà quindi necessaria la creazione di un'ulteriore tabella, ossia "Gestione";

proseguendo, tra “Clienti” e “Acquisti” è presente una relazione 1 a N, l’entità “Acquisti” conterrà quindi una chiave esterna riferentesi a “Clienti”.

1.3. **Modello logico**

Procediamo con la stesura del modello logico:

Legenda: **PK**, **FK**.

OPERATORI (**id_operatore**, nome, cognome, username, password);

CLIENTI (**email**, nome, cognome, data_nascita, data_iscrizione, cf, città, provincia, via, n_civ, CAP);

GESTIRE (**id_operatore**, **email**);

ACQUISTI (**id_acquisto**, data, somma, **email**).

1.4. **Creazione delle tabelle**

Proseguiamo quindi con la creazione delle tabelle in linguaggio MySQL:

CREATE TABLE operatori (

id_operatore int (5) PRIMARY KEY AUTO_INCREMENT,
nome varchar (128) NOT NULL,
cognome varchar (128) NOT NULL,
username varchar(128) NOT NULL,
password varchar(128) NOT NULL
);

CREATE TABLE clienti (

email varchar (128) PRIMARY KEY,
nome varchar (128) NOT NULL,
cognome varchar (128) NOT NULL,
cf varchar (16) NOT NULL,
data_nascita date NOT NULL,
città varchar (128) NOT NULL,
provincia varchar (128) NOT NULL,

```
via varchar (128) NOT NULL,  
n_civ int (5) NOT NULL,  
CAP int (5) NOT NULL,  
data_iscrizione date NOT NULL  
);
```

```
CREATE TABLE gestione (
```

```
id_operatore int (5),  
email varchar (128),  
PRIMARY KEY (id_operatore, email),  
FOREIGN KEY (id_operatore) REFERENCES operatori (id_operatore) ON  
DELETE CASCADE,  
FOREIGN KEY (email) REFERENCES clienti (email) ON DELETE CASCADE  
);
```

```
CREATE TABLE acquisti (
```

```
id_acquisto int (8) PRIMARY KEY AUTO_INCREMENT,  
data datetime NOT NULL,  
somma float (8, 2) NOT NULL,  
email varchar (128),  
FOREIGN KEY (email) REFERENCES clienti (email) ON DELETE CASCADE  
);
```

2. Portale Web

È necessario creare un portale web per interfacciarsi con il database che permetta la gestione dei dati, ossia l'aggiunta, la modifica e l'eliminazione di clienti dal database; deve inoltre essere implementato un sistema di autenticazione per gli operatori.

Il portale implementato si basa su php, specificamente sulla libreria "mysqli", viene inoltre utilizzato jQuery per il filtro e bootstrap per la grafica.

2.1. Autenticazione

Il sistema di autenticazione implementato funziona attraverso le "\$_SESSION" di php; i dati vengono inseriti attraverso un form, il tasto di login indirizza l'utente alla pagina "process.php" nella quale vengono effettuati i controlli sui dati: viene controllato che i campi siano effettivamente riempiti, in caso contrario si viene reindirizzati alla pagina di login con un messaggio di errore; in seguito viene effettuata la connessione al database per controllare le credenziali inserite con gli operatori registrati in tabella, se si riscontra un risultato l'utente viene indirizzato alla pagina "index.php", altrimenti viene reindirizzato alla pagina di login con un messaggio di errore.

Se si prova ad accedere direttamente alla pagina "index.php", viene effettuato un controllo sulla presenza della variabile "\$_SESSION['username']", in caso della sua assenza si viene indirizzati alla pagina di login in modo che non sia possibile aggirare l'autenticazione.

2.2. **Dashboard**

Nella pagina “index.php” è presente la dashboard dell’operatore, la prima cosa da specificare è la navbar, la quale presenta due tasti: “Anagrafe clienti”, un semplice tasto presente in ogni pagina che funziona come tasto home; e “Aggiungi cliente”, questo tasto è presente in ogni pagina a parte la pagina “addCliente.php” la quale è proprio la pagina a cui indirizza il tasto, in questo modo un operatore può aggiungere un cliente in qualsiasi momento stia navigando il portale.

A destra della navbar vengono indicati il nome ed il cognome dell’operatore con un menù a tendina che permette il logout il quale viene effettuato attraverso la pagina “logout.php” che elimina la sessione attraverso il comando “session_destroy()” e reindirizza al login.

In seguito troviamo i dati “Utenti registrati” e “Guadagno giornaliero” che vengono estratti dal database attraverso una query e mostrati a schermo. Il filtro e la tabella degli utenti dipendono uno dall’altro, infatti la tabella viene stampata dalla pagina “filtro.php” nel div “result” dalla pagina “index.php” attraverso jquery.

Nel file “filtro.php” l’input del form viene utilizzato per effettuare una query con l’operatore “LIKE” su ogni parametro della tabella “clienti”, se il form è vuoto viene mostrata tutta la tabella, mentre se iniziamo a inserire lettere o numeri nel form, la query viene aggiornata e la tabella viene ristampata con i risultati della query. In questo modo viene creato un filtro che può essere utilizzato su ogni colonna della tabella.

Infine, schiacciando sul tasto di collegamento nella colonna “Profilo” della tabella, si viene indirizzati sulla pagina “profiloCliente.php”, nella quale, grazie alla variabile “email” passata attraverso il link, viene effettuata una query per stampare una tabella con i dati del singolo cliente.

2.3. **Aggiungi cliente**

Nella pagina “addCliente.php” si può aggiungere un cliente al database, questo viene effettuato grazie ad un form con il metodo “\$_GET”, il quale una volta inseriti i dati, essi vengono raccolti dalla pagina “insertCliente.php”, che attraverso una variabile Boolean controlla se i dati stiano arrivando dalla pagine “addCliente.php” o “modificaCliente.php” e esegue una query per inserire o modificare un utente nel database. Inoltre, la data di iscrizione viene automaticamente impostata alla data corrente attraverso il metodo php “date”. Una volta effettuata l'azione l'utente può tornare alla dashboard attraverso un tasto home sottostante.

2.4. **Profilo cliente**

Nella pagina “profiloCliente.php” viene ricevuta la mail attraverso il link con la quale si effettua una query per estrarre i suoi dati anagrafici; viene quindi stampata una tabella che li contiene.

In fondo alla pagina sono presenti due tasti, uno per la modifica ed uno per la cancellazione dei dati dal database.

Se si schiaccia sul tasto di modifica si viene indirizzati nella pagina modifyCliente.php alla quale vengono passati tutti i dati del cliente per riempire automaticamente il form attraverso le “\$_SESSION”, l'email viene impostata su “readonly” in quanto la chiave primaria non può essere modificata. Inoltre la variabile Boolean sopracitata viene impostata a falsa in modo che il controllo effettuato su “insertCliente.php” esegui la query di modifica e non di inserimento.

Se invece si schiaccia sul tasto di cancellazione, viene eseguita una query sulla pagina “removeCliente.php” con l'email del cliente per cancellare il record attribuita ad essa, dopodiché si può tornare alla dashboard.