

Laboratorio di Sistemi Informativi

Base di dati per la gestione di una Palestra

BAJRON ISMAILAJ

Matricola: 2686563

Email: caiupi@yahoo.it

Indice generale

| | |
|---|-----------|
| Autore: BAJRON ISMAILAJ..... | 1 |
| 1 Analisi dei Requisiti..... | 1 |
| 1.1 Raccolta dei requisiti..... | 1 |
| 1.2 Eliminazione delle ambiguità..... | 2 |
| 1.3 Glossario dei termini | 2 |
| 1.4 Strutturazione dei requisiti..... | 2 |
| 1.5 Operazione su i dati | 3 |
| 2 Progettazione Concettuale..... | 4 |
| 2.1 Schema scheletro astratta..... | 4 |
| 2.2 Sviluppo delle componenti..... | 4 |
| 2.3 Schema finale..... | 6 |
| 2.4 Documentazione dello schema concettuale..... | 7 |
| 3 Progettazione Logica..... | 9 |
| 3.1 Tavola dei volumi..... | 9 |
| 3.2 Tavole delle Operazioni..... | 10 |
| 3.3 Eliminazione delle ridondanze..... | 10 |
| 3.4 Eliminazione delle generalizzazioni..... | 10 |
| 3.5 Partizionamento/accorpamento di concetti..... | 11 |
| 3.6 Il nuovo schema E-R ristrutturato..... | 11 |
| 3.7 Schema relazionale..... | 12 |
| 4 Progettazione Fisica..... | 13 |
| 4.1 Codice sorgente per il database Palestra..... | 13 |
| 4.2 Codifica Operazioni..... | 14 |
| 5 Lista dei Script e dei File..... | 18 |
| 5.1 Cartella sorgenti | 18 |
| 5.2 Cartella documenti..... | 18 |

I ANALISI DEI REQUISITI

1.1 Raccolta dei requisiti

Dopo vari colloqui con il committente per la realizzazione di una base di dati per la gestione di una palestra ecco quali sono la raccolta dei requisiti:

Nella palestra sono presenti diversi impianti, aree o zone. Area piscina, zona fitness, area pesi,

zona cyclette, sauna, sala di danza ecc. Ogni area è identificata con un nome accompagnata da una descrizione. In questi impianti si tengono diverse attività oppure no. Area piscina comprende la piscina per bambini sia quella olimpica.

Nella palestra sono organizzati delle attività quali yoga, spinningg, acquagym, hydrobike, body etc. Per ogni corso svolto è noto il nome del istruttore, il numero massimo e minimo di iscritti, in quali giorni della settimana si svolge ed a che ora. Inoltre si deve specificare l'impianto in qui si svolge un corso. Il corso consiste in una serie di lezioni di durata di una ora. Un corso può svolgersi anche con diverse lezioni al giorno e il sistema deve tenere traccia del ora d'inizio e della fine. La palestra viene frequentata da una clientela che si deve iscrivere. L'iscrizione avviene compilando un modulo con i seguenti dati: nome, cognome, indirizzo, codice fiscale, data di nascita, data inizio. Gli iscritti sono di due tipi, titolari di un abbonamento e quelli senza abbonamento. Per gli iscritti con abbonamento si devono specificare anche i dati relativi al abbonamento. Gli abbonati possono partecipare a uno o più corsi. Gli iscritti senza un abbonamento non possono partecipare ai corsi.

Ai interni della palestra è noto il codice fiscale, nome e cognome, il ruolo svolto nella palestra, il numero di telefono e l'elenco delle qualifiche. Inoltre il database deve memorizzare i dati relativi al contratto di assunzione del personale come data inizio e fine contratto e anche il tipo:

Indeterminato, a progetto etc..

Le persone che sono iscritte ai corsi devono presentare un certificato medico. Il database contiene l'informazione del medico, interno o esterno che ha redatto il certificato e la data del rilascio del certificato.

1.2 Eliminazione delle ambiguità

La palestra è composta da diverse sale, le quali posso essere composte a loro volta da diverse aree. Gli iscritti della palestra sono di due tipi quelli titolari di un abbonamento e quelli senza. Solo gli iscritti con un abbonamento possono partecipare a un corso.

1.3 Glossario dei termini

Si è identificato i seguenti termini principali con i quali si è costruito un glossario dei termini:

| Termine | Descrizione | Sinonimi | Collegamenti |
|-----------|---|---|--|
| Impianto | Aree della palestra | Area, Zona | Corso |
| Iscritto | Persona iscritta nella palestra. | Cliente, Abbonato, Partecipante ai corsi, Cliente senza Abbonamento | Abbonato, Corso. Personale |
| Corso | Attività svolta nella palestra. Percorso composto da varie lezioni. | Attività, Insieme di Lezioni | Abbonato, Impianto, Personale, Lezione |
| Personale | Persone che lavorano nella palestra. | Addetto, Medico, Istruttore, Interno | Corso, Iscritto |
| Abbonato | Persona iscritto nella palestra in possesso di un abbonamento. | Partecipante a un corso, iscritto ai corsi | Corso, Iscritto |

1.4 Strutturazione dei requisiti

A questo punto possiamo ristrutturare le nostre specifiche in base ai termini principali e modificando il testo eliminando le ambiguità e aggiungendo altri dati per la specifica.

| Frase relative al Impianto |
|--|
| Nella palestra sono presenti diversi impianti.: Piscina, sauna, sala danza, sala attrezzi composta da le aree: fitness, pesi, cyclette. Ogni area è identificata con un nome accompagnata da una descrizione. Area piscina comprende la piscina per bambini sia quella olimpica. |

| Frase relative al Iscritto |
|---|
| La palestra viene frequentata da una clientela che si deve iscrivere obbligatoriamente. L'iscrizione avviene compilando un modulo con i seguenti dati: nome, cognome, indirizzo, codice fiscale, data di nascita, data inizio. Gli iscritti sono di due tipi, titolari di un abbonamento e quelli senza abbonamento. Gli iscritti senza abbonamento non possono partecipare ai corsi. |

| Frase relative al personale |
|---|
| Al personale della palestra è noto il codice fiscale, che lo identifica, un nome, il numero di telefono e l'elenco delle qualifiche. Inoltre il database deve memorizzare i dati relativi al contratto di assunzione del personale con relativa data d'inizio e data fine rapporto lavorativo |

| Frase relative al Corso |
|---|
| Nella la palestra sono organizzati dei corsi quali <i>spinning</i> , <i>acquagym</i> , <i>hydrobike</i> , <i>body</i> , <i>corsi di danza</i> . Per ogni corso svolto è noto il nome del istruttore, il numero massimo e minimo di partecipanti, in quali giorni della settimana si svolge ed a che ora. Si deve specificare l'impianto in cui si svolge un corso. Il corso consiste in una serie di lezioni di durata di una ora. Un corso può svolgersi anche con diverse lezioni al giorno e il sistema deve tenere traccia del ora d'inizio e della fine. |

| Frase relative al Abbonato |
|---|
| Gli iscritti sono di due tipi, titolari di un abbonamento e quelli senza abbonamento. Per gli iscritti con abbonamento si devono specificare anche data inizio abbonamento e data fine abbonamento. Gli abbonati possono partecipare a uno o più corsi. |

| Frase relative al Visita Medica |
|---|
| Le persone che sono iscritte ai corsi devono presentare un certificato medico. Il database contiene l'informazione del medico, interno o esterno, che ha redatto il certificato e la data del rilascio del certificato. |

1.5 Operazione su i dati

In seguito elenchiamo la lista delle operazioni che effettueremo sul database.

- (Op1)- Inserisci corso (5 volta al anno)
- (Op2)- Elimina corso (5 volta al anno)
- (Op3)- Stampa partecipanti di un corso (2 volta a settimana)
- (Op4)- Inserisci lezione (30 al settimana)
- (Op5)- Inserisci iscritto (1 al giorno)
- (Op6)- Elimina iscritto (1 al mese)
- (Op7)- Elimina lezione (1 volta al mese)
- (Op8)- Inserisci dipendente (2 al anno)
- (Op9)- Aggiungi/rinnovi abbonamento (2 in settimana)
- (Op10)- Aggiungi partecipante (2 alla settimana)
- (Op11)- Stampa abbonamento; Controllo abbonamento scaduti (1 volta al giorno)
- (Op12)- Stampa le lezioni di oggi (1 volte al giorno)
- (Op13)- Stampa le lezioni di un di un istruttore (3 volta settimana)

2 PROGETTAZIONE CONCETTUALE

Per la progettazione concettuale si è usato una strategia mista, utilizzando inizialmente la strategia top-down per la creazione di uno schema scheletro contenente a livello astratto i concetti principali del database. In seguito usiamo la strategia bottom-up per sviluppare per ogni concetto principale un schema E-R che lo rappresenta. Alla fine abbiamo unito questi schemi per la creazione dello schema finale. Da specificare che per sviluppare certi concetti abbiamo usato anche una strategia inside-out che è sempre un sottoinsieme della strategia bottom-up.

2.1 Schema scheletro astratta

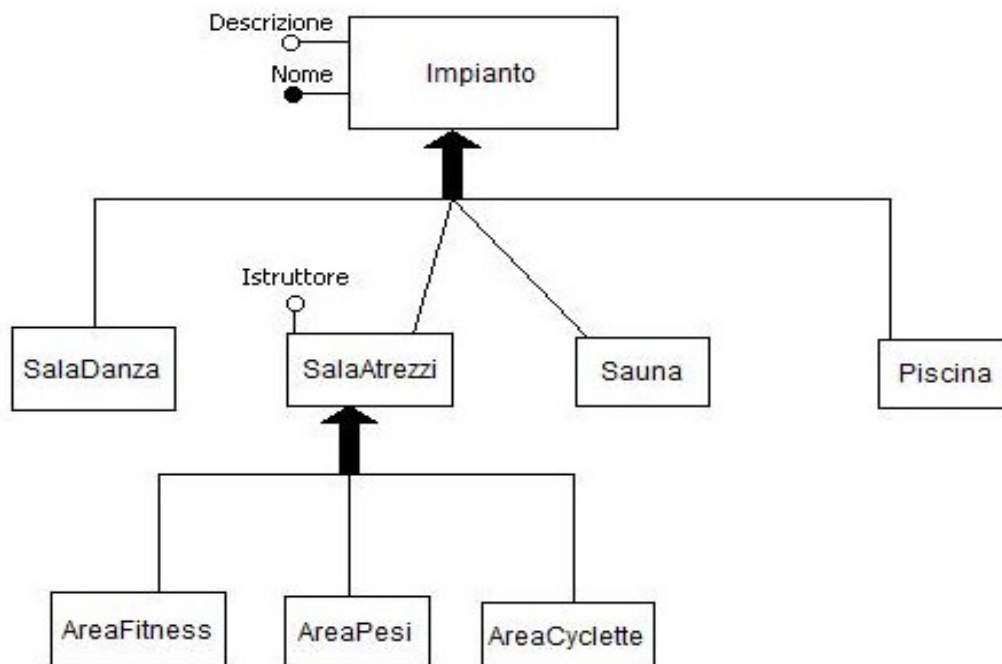
Utilizzando una strategia top-down siamo riusciti a costruire uno schema scheletro astratta che contiene i concetti principali quali: Impianto, Corso e Persona. Da sottolineare che l'entità persona comprende sia gli iscritti alla palestra sia il personale che lavorano in essa. Infatti la relazione che lo collega con l'entità Corso è Partecipa/Insegna per mettere in relazione il fatto che un iscritto partecipa ai corsi e anche che l'istruttore, che fa parte del personale della palestra, insegna ai corsi.



2.2 Sviluppo delle componenti

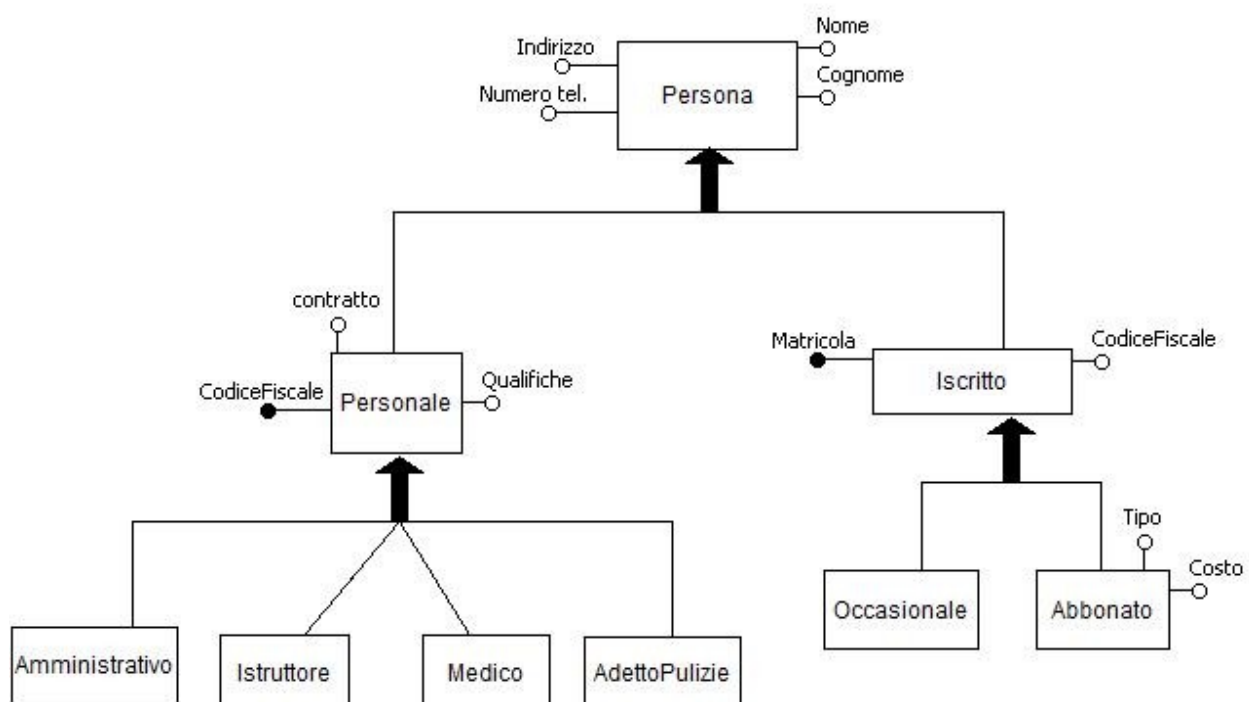
Impianto

Abbiamo rappresentato l'impianto tramite una gerarchia di generalizzazioni tra entità. In questa gerarchia abbiamo entità che hanno solo una occorrenza come sala danza o sauna, ma anche entità con più occorrenze come piscina. Infatti con l'entità piscina rappresentiamo occorrenze come piscina per bambino o quella per adulti; olimpica. Come vedremo più in avanti abbiamo deciso di dividere virtualmente l'entità piscina olimpica considerando ogni corsia di una piscina come una occorrenza a parte dell'entità piscina.



Persona

Anche per definire il concetto di persona si fa uso di una gerarchia di generalizzazioni. Ogni persona del database ha come attributo un nome, cognome, indirizzo e numero di telefono. Il personale della palestra è identificato dal codice fiscale invece gli iscritti alla palestra si identificano univocamente tramite la matricola; Un intero univoco. Questo perché ci possono essere persone che non sono in possesso del codice fiscale, per esempio, i stranieri/turisti. L'attributo Contratto dell'entità Personale sarà ulteriormente sviluppato poiché si tratta di un attributo composto.



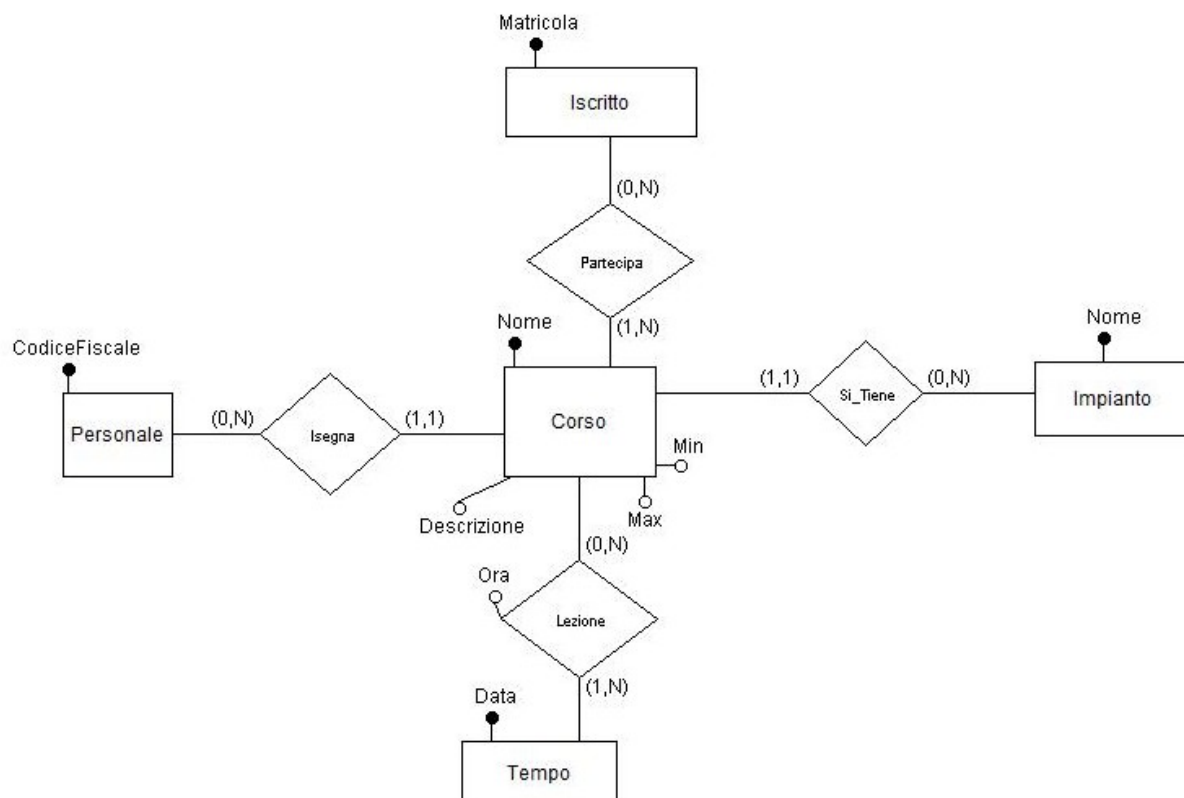
Abbiamo due tipologie di iscritti alla palestra, occasionale e abbonati. L'entità abbonato sono gli

iscritti con un abbonamento e hanno come attributi: Tipo e Costo che fanno riferimento al tipo e costo del abbonamento sottoscritto con la palestra.

Corso

Nella schema E-R che segue rappresentiamo sia il concetto Corso ma anche il collegamento di esso con il resto dei concetti principali poiché rappresenta il nucleo del nostro schema.

Il corso è identificato dal nome univoco e ha come attributi Min e Max che sono rispettivamente il numero minimo e numero massimo di allievi che partecipano al corso. Visto che ogni corso è composto da varie lezioni che si distinguono tramite la data in cui si svolgono, si è cercato di rappresentare le lezioni come le occorrenze di una associazione per collegare Corso con Tempo. Abbiamo usato come attributo Data per identificare il Tempo e Ora per l'associazione Lezione. Ogni corso ha un istruttore e ogni corso si svolge in un impianto.



2.3 Schema finale

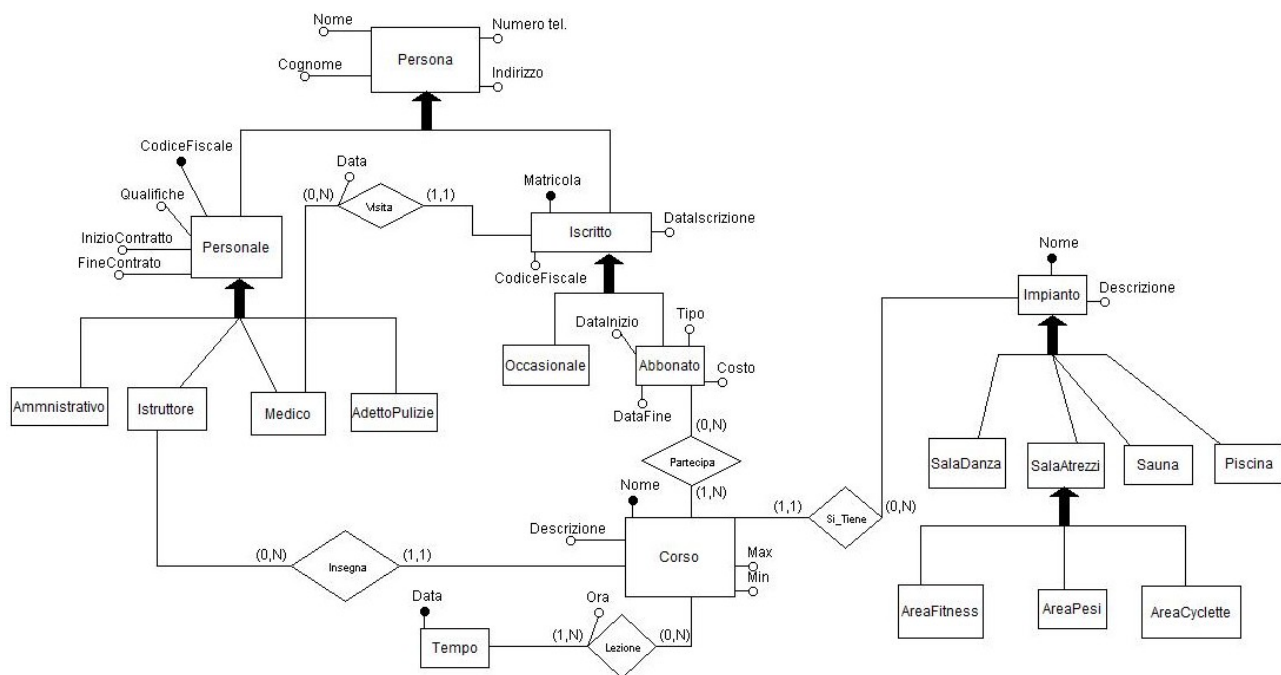
In questo schema finale abbiamo riunito tutti gli schemi precedenti.

Visto che ogni iscritto alla palestra ha bisogno di un certificato medico, cioè di fare una visita medica, abbiamo messo in associazione le entità Medico e Iscritto tramite la relazione Visita. L'Entità Iscritto partecipa in questa associazione con una cardinalità che va da 0 a 1 che significa che eventualmente l'iscritto può avere un certificato con un altro medico che non fa parte del personale della palestra.

Il personale della palestra ha anche due attributi in più che sono InizioContratto e FineContratto che indicano i dati relativi al contratto di lavoro del personale.

Gli abbonati alla palestra hanno in più gli attributi relativi a la data di inizio e fine di un

abbonamento.



Schema Concettuale Finale

Solo gli istruttori insegnano nei corsi di conseguenza abbiamo messo in relazione tramite Insegna. Ai corsi possono partecipare solo gli abbonati della palestra perciò abbiamo la relazione partecipa che gli collega.

2.4 Documentazione dello schema concettuale

Dizionario dei dati: In seguito la tabella che contiene le entità dello schema finale:

| Nome Entità | Descrizione | Attributi | Identificatore |
|----------------------|--|------------------------------|----------------|
| Impianto | Impianto della palestra | Nome, Descrizione | Nome |
| Sala Danza | Sala della palestra | | |
| Sala Attrezzi | Sala della palestra che contiene attrezzi per l'attività fisica. | | |
| Sauna | Area della palestra. | | |
| Piscina | Piscina della palestra | | |
| Area Fitness | Area della palestra adatta per fitness | | |
| Area Pesi | Area della palestra che contiene attrezzi per l'attività fisica. | | |
| Area Cyclette | Area della palestra con cyclette | | |
| Corso | Attività che si svolgono nella palestra. | Nome, Descrizione, Min , Max | Nome |
| Tempo | Entità che rappresenta il tempo, la data. | Data | Data |

| <i>Nome Entità</i> | <i>Descrizione</i> | <i>Attributi</i> | <i>Identificatore</i> |
|------------------------|---|---|-----------------------|
| Persona | Le persone coinvolte nella palestra. | Nome, Cognome, Tel, Indirizzo. | |
| Iscritto | Clienti della palestra. | Matricola, Codice Fiscale, DataIscrizione. | Matricola |
| Personale | Persone che lavorano nella palestra. | Codice Fiscale, Qualifiche, Inizio Contratto, Fine Contratto. | Codice Fiscale |
| Amministrativo | Personale del settore amministrativo. | | |
| Istruttore | Istruttore di un corso della palestra | | |
| Medico | Medico della palestra | | |
| Addetto Pulizie | Persone addetta a svolgere lavoro di pulizia. | | |
| Abbonato | Clienti con un abbonamento della palestra. | Data Inizio, Data Fine, Tipo, Costo. | |
| Occasionale | Iscritti senza abbonamento. | | |

Le relazioni dello schema finale e le cardinalità con le entità ad esse collegate.

| <i>Relazione</i> | <i>Descrizione</i> | <i>Entità Coinvolte</i> | <i>Attributi</i> |
|------------------|---|----------------------------------|------------------|
| Insegna | Associa un istruttore al corso | Istruttore (0, N) Corso (1,1) | |
| Visita | Associa un medico con un iscritto nella palestra. | Medico (0,N) Iscritto (1,1) | Data |
| Partecipa | Associa gli abbonati della palestra a dei corsi. | Abbonato (0,N) Corso (1,N) | |
| Si Tiene | Associa i corsi al impianto in cui si svolgono. | Corso (1,1) Impianto (0,N) | |
| Lezione | Associa il corso ad un unità del tempo. | Corso (0,N) Tempo (1,N) | Ora |

Regole aziendali per lo schema concettuale finale.

| Regole di vincolo |
|---|
| <ul style="list-style-type: none"> · (RV1) Ogni corso deve avere un numero di allievi compresi fra 1 e 20. · (RV2) La data di inizio abbonamento deve essere minore della data fine abbonamento. · (RV3) La data di inizio contratto deve essere minore della data di fine contratto. · (RV4) Solo gli abbonati possono partecipare ai corsi. · (RV5) Il prezzo di un abbonamento deve essere maggiore di 0. · (RV6) La durata di una lezione per un corso deve essere di una ora. · (RV7) Ogni iscritto della palestra deve avere un certificato medico. · (RV8) Ogni corso si svolge al proprio impianto. · (RV9) Il minimo numero degli iscritti a un corso deve essere minore al numero massimo iscritti a un corso. · (RV10) Le lezioni si possono svolgere se al corso partecipano un numero di allievi compresi fra minimo e massimo iscritti. |
| Regole di derivazione |
| Non sono presenti regole di derivazione. |

3 PROGETTAZIONE LOGICA

Visto che in questa fase della progettazione dobbiamo ristrutturare lo schema E-R ereditato da la fase precedente abbiamo bisogno di creare un'idea su le prestazioni della nostra base di dati. Per questo servono le tabelle dei volumi dei dati e quella delle frequenze delle operazioni.

3.1 Tavola dei volumi

Per ogni entità e relazione dello schema concettuale contiamo il numero delle occorrenze.

| Concetto | Tipo | Volume |
|-----------------|------|--------|
| Impianto | E | 16 |
| Sala Danza | E | 2 |
| Sala Attrezzi | E | 6 |
| Sauna | E | 1 |
| Piscina | E | 7 |
| Area Fitness | E | 2 |
| Area Pesi | E | 2 |
| Area Cyclette | E | 2 |
| Corso | E | 15 |
| Tempo | E | 300 |
| Persona | E | 315 |
| Iscritto | E | 300 |
| Personale | E | 15 |
| Amministrativo | E | 2 |
| Istruttore | E | 10 |
| Medico | E | 1 |
| Addetto Pulizie | E | 2 |
| Abbonato | E | 200 |
| Occasionale | E | 100 |
| Insegna | R | 15 |
| Visita | R | 150 |
| Partecipa | R | 400 |
| Si Tiene | R | 15 |
| Lezione | R | 600 |

3.2 Tavole delle Operazioni

Per ogni operazione indichiamo il tipo (interattiva o batch) e la frequenza in cui si eseguono. Questa tabella è collegata alla lista delle operazioni fatta nella fase di analisi dei requisiti. Giustamente la frequenza va rivista.

| Operazione | Tipo | Frequenza |
|---|------|------------------------|
| Op. 1 Inserisci corso | I | 5 volte al anno |
| Op. 2 Elimina corso | I | 5 volte al anno |
| Op. 3 Stampa partecipanti di un corso | B | 2 volte alla settimana |
| Op. 4 Inserisci lezione | I | 30 volte settimana |
| Op. 5 Inserisci iscritto | I | 1 al giorno |
| Op. 6 Elimina iscritto | I | 1 volta al mese |
| Op. 7 Elimina lezione | I | 1 volta al mese |
| Op. 8 Inserisci dipendente | I | 2 volte al anno |
| Op. 9 Aggiungi/rinnovi abbonamento | I | 2 volte alla settimana |
| Op. 10 Aggiungi partecipante | I | 2 volte settimana |
| Op. 11 Stampa abbonamento controllo scaduti | B | 1 volta al giorno |
| Op. 12 Stampa le lezioni di oggi | B | 1 volta al giorno |
| Op. 13 Stampa le lezioni di un di un istruttore | B | 3 volta settimana |

3.3 Eliminazione delle ridondanze

Non state rilevate ridondanze sugli attributi delle entità e sulle relazioni identificate durante la progettazione .

3.4 Eliminazione delle generalizzazioni

Nello schema concettuale sono presenti due gerarchie di generalizzazioni che devono essere eliminate.

- Per quando riguarda la gerarchia di generalizzazioni con padre l'entità Impianto abbiamo deciso per l'accorpamento delle figlie nella generalizzazione del padre. Questo anche perché le occorrenze delle figlie sono in un numero molto piccolo. Visto che l'entità Impianto ha una chiave; Nome che lo identifica univocamente e visto che le entità figlie non contengono attributi, non abbiamo bisogno di aggiungere un altro attributo per identificare le figlie.

- Per la gerarchia di generalizzazioni in riferimento al entità padre Persona la scelta ovvia è l'accorpamento del padre della generalizzazione delle figlie. Di conseguenza l'entità Persona viene eliminata e i suoi attributi vengono aggiunti alle entità Personale e Iscritto.

- Le figlie dell'entità Personale: Amministrativo, Istruttore, Medico e Addetto Pulizie, vengono accorpate nell'entità Personale. L'attributo Qualifiche dell'entità Personale viene usato per descrivere le occorrenze eliminate.

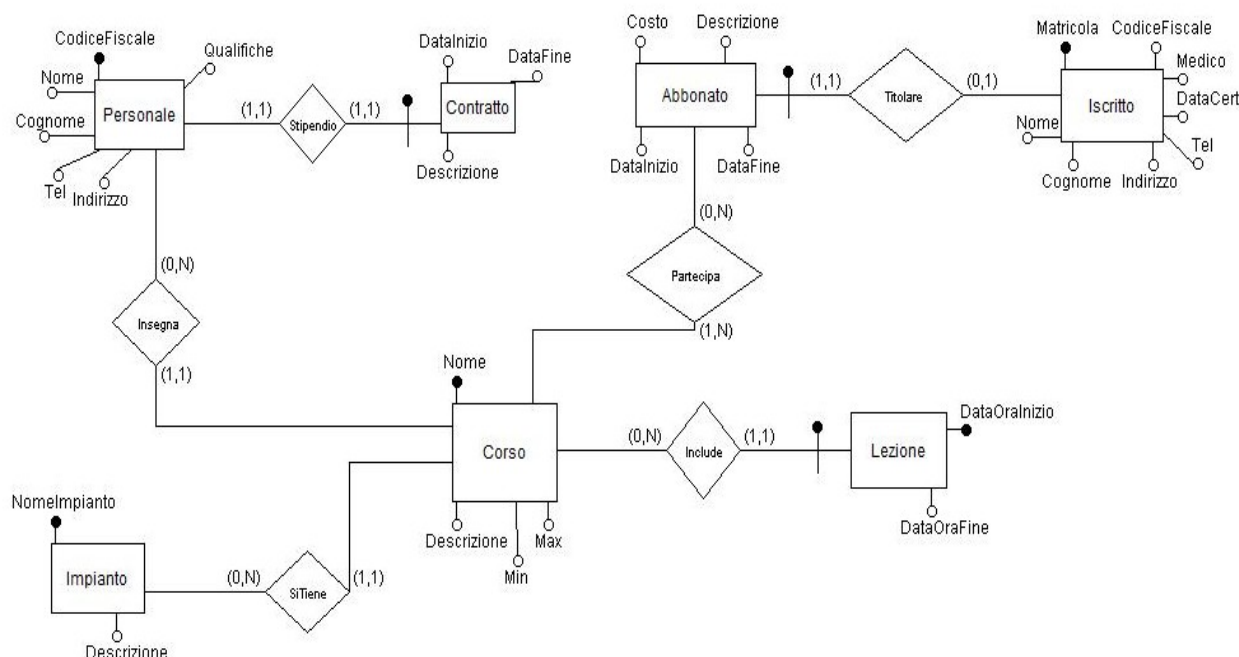
- L'entità Occasionale viene accorpata nell'entità Iscritto. Le entità Iscritto e Abbonato vengono

collegate tramite la relazione Titolare.

3.5 Partizionamento/accorpamento di concetti

- L'Entità Personale viene partizionata creando una nuova entità Contratto con attributi: Descrizione, DataInizio e DataFine. L'entità Contratto e Personale vengono collegate tramite associazione uno a uno con la relazione Stipendio. Questo perché non siamo interessati ad avere dati relativi al contratto di un istruttore ogni volta che dobbiamo consultare i dati generici di un istruttore.
- Per una migliore rappresentazione delle lezioni di un corso eliminiamo l'entità tempo e convertiamo la relazione Lezione in una entità la quale viene collegata con Corso tramite associazione Include. Ogni lezione ha un due attributi che sono DataOraInizio e DataOraFine. Inoltre ogni lezione viene identificata tramite il nome del corso che fa parte (identificatore esterno) e la DataOraInizio.
- La relazione Visita viene eliminata. Per questo al entità Iscritto vengono aggiunte due nuovi attributi; Medico e DataCertificato che sono rispettivamente il codice fiscale del medico che ha eseguito la visita e la data in qui si è svolta. Questo anche in considerazione che molte visite vengono eseguite non dal medico della palestra.

3.6 Il nuovo schema E-R ristrutturato



Identificatori Principali

| Entità | Identificatore |
|-----------|---------------------------|
| Personale | CodiceFiscale |
| Contratto | Personale (CodiceFiscale) |

| Entità | Identificatore |
|----------|---------------------------|
| Corso | Nome |
| Impianto | NomeImpianto |
| Lezione | Corso(Nome),DataOraInizio |
| Iscritto | Matricola |
| Abbonato | Iscritto(Matricola) |

Analizzando il nuovo schema concettuale ristrutturato si nota che tutte le associazioni presenti sono in forma normale di Boyce e Codd

3.7 Schema relazionale

Nella traduzione dello schema concettuale nello schema relazionale abbiamo il seguente risultato:

Personale (CodiceFiscale, Nome, Cognome, Tel, Indirizzo, Qualifiche)

Contratto (PERSONALE, Descrizione, DataInizio, DataFine)

Impianto(Nome, Descrizione)

Corso(Nome, IMPIANTO, PERSONALE, Descrizione, Min, Max)

Iscritto(Matricola, CodiceFiscale, Nome, Cognome, Medico, DataCert, Indirizzo, Tel, DataIscrizione)

Abbonato(ISCRITTO, Descrizione, Costo, DataInizio, DataFine)

Partecipa(CORSO, ABBONATO)

Lezione(CORSO, DataOraInizio, DataOraFine)

Riferimenti:

| Entità/Relazione | Riferimento |
|--|---|
| Personale (<u>CodiceFiscale</u> , Nome, Cognome, Tel, Indirizzo, Qualifiche) | - |
| Contratto (<u>PERSONALE</u> , Descrizione, DataInizio, DataFine) | PERSONALE → Personale.CodiceFiscale |
| Impianto(<u>Nome</u> , Descrizione) | - |
| Corso(<u>Nome</u> , IMPIANTO, PERSONALE, Descrizione, Min, Max) | IMPIANTO → Impianto.NomeImpianto PERSONALE → Personale.CodiceFiscale |
| Iscritto(<u>Matricola</u> , CodiceFiscale, Nome, Cognome, Medico, DataCert, Indirizzo, Tel, DataIscrizione) | - |
| Abbonato(<u>ISCRITTO</u> , Descrizione, Costo, DataInizio, DataFine) | ISCRITTO → Iscritto.Matricola |
| Partecipa(<u>CORSO</u> , <u>ABBONATO</u>) | CORSO → Corso.Nome ABBONATO → Abbonato.Iscritto |
| Lezione(<u>CORSO</u> , <u>DataOraInizio</u> , DataOraFine) | CORSO → Corso.Nome |

4 PROGETTAZIONE FISICA

Per lo sviluppo del database si è usato il motore INNODB poiché più flessibile di MyISAM nella fase dello sviluppo. Per una lettura migliore del codice SQL consiglio il file sorgente

TuttoPalestra.sql poiché include anche i commenti necessari per capire meglio le istruzioni del programma.

I caratteri in blu rappresentano un codice sql.

4.1 Codice sorgente per il database Palestra

```
drop database if exists palestra;
```

```
create database palestra;
use palestra;
create table personale(
codiceFiscale char(16) primary key,
nome char(16),
cognome char(16),
tel char(16),
indirizzo varchar(32),
qualifiche varchar(32)
) ENGINE=INNODB;
```

```
create table contratto(
codicePersonale char(16) primary key,
descrizione varchar(256),
dataInizio date,
dataFine date,
foreign key (codicePersonale) references personale(codiceFiscale) on delete cascade on update cascade
) ENGINE=INNODB;
```

```
create table impianto(
nome char(16) primary key,
descrizione varchar(32)
) ENGINE=INNODB;
```

```
create table corso(
nome char(16) primary key,
nomeImpianto char(16),
istruttore char(16),
descrizione varchar(256),
minIsritto int(2),
maxIsritto int(2),
foreign key (nomeImpianto) references impianto(nome) on delete no action on update cascade,
```

```
foreign key (istruttore) references personale(codiceFiscale) on delete no action on update cascade
) ENGINE=INNODB;
```

```
create table lezione(
inizio datetime,
nomeCorso char(16),
fine datetime,
primary key(inizio, nomeCorso),
foreign key (nomeCorso) references corso(nome) on delete cascade on update cascade
) ENGINE=INNODB;
```

```
create table iscritto(
matricola int(6) primary key AUTO_INCREMENT,
codiceFiscale char(16) unique,
nome char(16),
cognome char(16),
medico char(16),
dataCert date,
indirizzo varchar(32),
tel char(16)
) ENGINE=INNODB;
```

```
create table abbonato(
matricolaAbbonato int(6) primary key,
costo numeric(6,2),
tipo char(32),
dataInizio date,
dataFine date,
foreign key (matricolaAbbonato) references iscritto(matricola) on delete cascade on update cascade
) ENGINE=INNODB;
```

```
create table partecipa(
abbonato int(6),
nomeCorso char(16),
primary key(abbonato, nomeCorso),
foreign key (abbonato) references abbonato(matricolaAbbonato) on delete cascade on update cascade,
foreign key (nomeCorso) references corso(nome) on delete cascade on update cascade
) ENGINE=INNODB;
```

4.2 Codifica Operazioni

-(Op1)- Inserisci corso:

```
insert into corso values
('Classica2','Danza','PPSDFDSR10M082K','Danza: Classica',45,20);
```

Per controllare che il numero massimo e minimo degli iscritti rispetti certi parametri si è usato un trigger: corso_bi. Per ogni nuovo inserimento controlla se le regole di vincolo aziendali sono soddisfatte altrimenti gli si assegna dei valore di default. Nel esempio il numero minimo dei iscritti supera i vincoli predisposti, il sistema gli assegna dei valori di default.

```

DELIMITER $$
DROP TRIGGER IF EXISTS `corso_bi` $$
CREATE TRIGGER corso_bi
BEFORE insert ON corso
FOR EACH ROW
BEGIN
IF NEW.maxIsritto > 20 OR NEW.maxIsritto < 2 THEN
SET NEW.maxIsritto = 20;
END IF;
IF NEW.minIsritto < 1 OR NEW.minIsritto > NEW.maxIsritto THEN
SET NEW.minIsritto = 1;
END IF;
ENDE

```

(Op2)-Elimina corso

```
Delete from corso where nome="Classica2";
```

Il nome del corso è una chiave primaria inoltre quando elimino i dati in corso si elimina automaticamente anche nella tabella partecipa e lezione.

Quando eliminino in partecipa i dati in corso rimangono.

(Op3)-Stampa partecipanti di un corso.

Per questa operazione si è scelto di creare una procedura: stampaPartecipanti().

```

DELIMITER $$
DROP PROCEDURE IF EXISTS `stampaPartecipanti` $$
create PROCEDURE stampaPartecipanti (IN corso char(16) )
COMMENT "Questa procedura restituisce gli abbonati a un corso "
BEGIN
select corso.nome as NomeCorso, personale.nome as NomeIstruttore, personale.cognome as
CognomeIstruttore, personale.codiceFiscale,
corso.MinIsritto as MinimIsrittiCorso, corso.maxIsritto as MassimIsrittiCorso
from personale, corso
where corso.nome = corso and corso.istruttore = personale.codiceFiscale;
select iscritto.matricola, iscritto.Nome, iscritto.cognome, iscritto.codicefiscale, iscritto.tel
from partecipa, iscritto
where partecipa.nomeCorso=corso
and partecipa.abbonato=iscritto.matricola;
END;

```

Stampa due tabelle:

Nella prima i dati del istruttore e il numero minimo e massimo dei partecipanti al corso.

Nella seconda tabella stampa i dati dei partecipanti al corso.

Per chiamare la procedura :

```
call stampapartecipanti('classica');
```

(Op4)-inserisci lezione.

Per questa operazione abbiamo implementato una procedura: nuovaLezione:

```

DELIMITER $$
DROP PROCEDURE IF EXISTS `nuovaLezione` $$
create PROCEDURE palestra.nuovaLezione(IN inizio DATETIME, nomeC char(16) )
COMMENT "Questa procedura inserisce una nuova lezione "
BEGIN

```

```

declare x, tmp, tmp2 int(2);
declare fine datetime;
set fine=date_add(inizio,interval 1 hour);
set tmp=(select corso.miniscritto from corso where nome=nomeC);
set tmp2=(select corso.maxiscritto from corso where nome=nomeC);
set x=(select count(*) from partecipa where partecipa.nomeCorso=nomeC);
if x<tmp then select "Impossibile aggiungere questa lezione. Il numero dei partecipanti è minore
del limite dei partecipanti del corso. Servono più partecipanti";
elseif x>tmp2 then select "Impossibile aggiungere questa lezione. Il numero dei partecipanti al corso
è maggiore del consetito. Devi eliminare dei partecipanti";
else insert into lezione values(inizio,nomeC,fine);
end if;
END;

```

La procedura controlla se il corso di qui la lezione fa parte ha un numero di partecipanti compresi fra il minimo e massimo iscritti al corso. Se questo controllo ha una buona fine aggiunge la lezione al database. Per chiamare la procedura:

```
call nuovaLezione('2012-01-15 08:00:00','Classica');
```

(Op5)-Inserisci iscritto.

Per questa operazione si è implementata una procedura iscriviti;

```

DELIMITER $$
DROP PROCEDURE IF EXISTS `iscrivi` $$
CREATE PROCEDURE iscriviti(IN codicefiscalel char(16),nomel char(16), cognomel char(16),medicol
char(16),
dataCertl date, indirizzol varchar(32), tell char(16))
COMMENT "Questa procedura restituisce il numero di matricola libero e memorizza il nuovo iscritto "
BEGIN
DECLARE tmp, indice, y int(6);
set indice = 1;
set y = (select count(*) from iscritto);
ciclo: loop
set tmp=(select matricola from iscritto where matricola=indice);
if tmp is null then leave ciclo; end if;
if indice=y then set indice= y+1; leave ciclo; end if;
set indice=indice+1;
end loop;
select "L'indice libero e: "+indice ;
insert                                     into                                     iscritto
values(indice,codicefiscalel,nomel,cognomel,medicol,dataCertl,indirizzol,tell,CURDATE());
end;

```

La procedura trova il primo indice più piccolo che serve a memorizzare l'iscritto. In questo modo manteniamo in maniera efficiente l'indice Matricola.

Per chiamare la procedura: Inseriamo un nuovo dato nella tabella iscritto.

```
call iscriviti('AFJDFWDIXZ45DF54','Mario','Rossi','BSDFBSHDTYR3456H','2001-01-02','Via corsica
12','055556357');
```

(Op6)-elimina iscritto

```
delete from iscritto where matricola =7;
```

Eliminando un iscritto si eliminano anche eventuali sue occorrenze su le tabelle abbonato e partecipa. In questo caso si elimina l'iscritto con matricola 7.

(Op7)-Elimina lezione: Ricordiamo che la chiave primaria della tabella lezione è DataOraInizio e nome del corso.

```
delete from lezione where inizio='2012-01-11 12:00:00' and nomecorso='Classica';
```

(Op8)-Inserisci dipendente: Inseriamo un nuovo dipendente nel nostro database.

```
insert into personale values
```

```
('MAWQDHJARE254HSD','Valerio','Rossi','05576545','Via barbera 1','Istruttore');
```

-(Op9)-Aggiungi abbonato. Un iscritto della palestra precisamente quello con la matricola 13 vuole abbonarsi.

```
insert into abbonato values
```

```
(13,50.00,'Abbonamento mensile','2011-12-30','2012-01-30');
```

Inoltre per rispettare le regole di vincolo quali; Data inizio abbonamento minore di data fine abbonamento e che il costo del abbonamento deve essere maggiore di 0 abbiamo implementato un trigger: Abbonato_bi

```
DELIMITER $$
```

```
DROP TRIGGER IF EXISTS `abbonato_bi` $$
```

```
CREATE TRIGGER abbonato_bi
```

```
BEFORE insert ON abbonato
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF NEW.datainizio > NEW.datafine THEN
```

```
SET NEW.datafine = NEW.datainizio;
```

```
END IF;
```

```
IF NEW.costo < 1 THEN
```

```
SET NEW.costo = 1;
```

```
END IF;
```

```
END
```

(Op10)-Aggiungi partecipante: Vogliamo aggiungere un iscritto della palestra con un abbonamento a partecipare a un corso. Per questa operazione abbiamo implementato una procedura: aggPartecipante che prende come input il numero di matricola del abbonato e il nome del corso in qui si vuole registrare.

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS `aggPartecipante` $$
```

```
create PROCEDURE aggPartecipante (IN matricola int(6), nomeC char(16) )
```

```
COMMENT "Questa procedura aggiunge un partecipante "
```

```
BEGIN
```

```
declare tmp, x int(2);
```

```
set tmp = (select maxIscritto from corso where corso.nome=nomeC);
```

```
set x = (select count(*) from partecipa where partecipa.nomeCorso=nomeC);
```

```
if x<tmp then insert into partecipa values(matricola,nomeC);
```

```
elseif tmp = x then select 'Impossibile eseguire l azione: Raggiunto numero massimo partecipanti al corso';
```

```
end if;
```

```
END;
```

Se si è raggiunti il numero massimo dei partecipanti ad un corso, questa procedura non aggiunge il nuovo partecipante avvisando tramite un messaggio. Per chiamare la procedura:

`call aggpartecipante(2,'body');`

(Op11)-Stampa abbonamenti scaduti: Una volta al giorno si controlla se nel database abbiamo dei abbonamenti scaduti.

Per Implementarla abbiamo usato una vista: abbonamentiScaduti

`create view abbonamentiScaduti as select * from abbonato where dataFine < CURRENT_TIMESTAMP;`

Per visualizzare il contenuto della vista basta eseguire la comanda:

`select * from abbonamentiScaduti;`

(Op12)-Stampa le lezioni di oggi: Una volta al giorno si stampano le lezioni che si svolgono quel giorno.

`select nomeCorso, Inizio as DataInizio, Fine as DataFine from lezione where date(inizio)=curdate();`

(Op13)-Stampa le lezioni di un di un istruttore.

Per implementare questa operazione abbiamo usato una vista: Istruttore

Istruttore contiene i dati relativi ai istruttori e i rispettivi corsi che tengono.

`create view istruttore as select personale.nome, personale.cognome, personale.codicefiscale, corso.nome as nomeCorso`

`from corso, personale where personale.codicefiscale=corso.istruttore;`

A questo punto eseguiamo un select su istruttore in base al suo codice fiscale.

`select istruttore.nome, istruttore.cognome, lezione.nomecorso, lezione.inizio, lezione.fine from lezione, istruttore`

`where lezione.nomecorso = istruttore.nomecorso and istruttore.codicefiscale='PPSDFDSR10M082K';`

5 LISTA DEI SCRIPT E DEI FILE

Questo progetto è stato realizzato su un sistema Windows Vista. Per lo sviluppo si è utilizzato MySQL QUERY BROWSER e MS-DOS basato su un server MySQL 5.5. Considerando il fatto che per lo sviluppo del database si è usato le nuove funzionalità quali; viste, procedure e trigger si deve usare un server MySQL 5.0 o superiore per l'esecuzione. Il progetto contiene i seguenti file dove nella cartella principale abbiamo:

File: **TuttoPalestra.sql** : Contiene l'implementazione fisica del database insieme con i trigger, procedure e i dati per il popolamento del database in un unico file. Cioè tutti i file raggruppati in un unico file per una esecuzione veloce. Inoltre in questo file si trovano anche i commenti per una lettura più comprensibile.

File: **PalestraRelazione.pdf** : Contiene la relazione del progetto in formato pdf.

5.1 Cartella sorgenti

In questa cartella si trovano i vari script per l'implementazione del database

Cartella: Separati : Contiene i singoli file che hanno formato tuttoPalestra.sql, come:

PalestraDati.sql : Contiene i dati per il popolamento del database.

Palestra.sql: Contiene l'implementazione fisica del database senza trigger, procedure e dati.

Più gli altri file che contengono le procedure e i trigger, ognuno in un file a parte.

5.2 Cartella documenti

Per lo sviluppo dei diagrammi E-R si è usato JDR versione 1.3.

La cartella documenti contiene i seguenti file prodotti con JDR e con i rispettivi JPG :

1. Iniziale.er #Primo schema scheletro: Fase concettuale
2. Persona.er #Rappresenta il concetto Persona : Fase concettuale
3. Corso.er #Concetto Corso: Fase concettuale
4. Impianto.er #Concetto Impianto : Fase concettuale
5. Palestra.er #Prodotto finale : Fase Concettuale
6. Normalizzato.er #Lo schema normalizzato : Fase Logica

Inoltre in questa cartella troviamo anche il file palestra.odt che è la sorgente della relazione sviluppato in OpenOffice.