

Università degli Studi di Bologna

Facoltà di Scienze

Corso di studi in Informatica per il Management

**Relazione**

**di**

**BOSTARTER**

**(progetto di Basi Dati)**

Nome e Cognome: Zhiyuan Xie

Matricola: 0001088879

Docente del corso: Prof. Marco Di Felice

Tutor del corso: Dr. Leonardo Ciabattini

Anno accademico 2024-2025

**RACCOLTA/ANALISI DEI REQUISITI**

**STEP1.** Produrre un documento di specifica seguendo le buone prassi

Ogni **utente** dispone di indirizzo email (univoco), nickname, password, nome, cognome, anno di nascita, luogo di nascita.

La **competenza** dispone di un nome(univoco, di tipo stringa). La lista delle competenze è disponibile a tutti gli utenti, i quali possono associare a ciascuna competenza un livello(numero intero tra 0 e 5).

Alcuni utenti *-ma non tutti-* possono appartenere a due sotto-categorie:

* Un **amministratore** dispone di un codice di sicurezza ed è l’unico autorizzato a popolare la lista delle competenze.
* Un **creatore** dispone di #nr\_progetti (ridondanza concettuale) ed affidabilità e solo lui può inserire uno o più progetti.

Ogni **progetto** ha un nome univoco, una descrizione, una data di inserimento, una o più foto, un budget da raggiungere per l’avvio, una data limite entro cui raggiungere tale budget e uno stato (campo enum: aperto/chiuso). Ogni progetto è associato da un solo utente creatore.

Inoltre, ogni progetto prevede una lista di **reward**: una reward dispone di un codice univoco, una descrizione, una foto.

I progetti appartengono *esclusivamente* a due categorie:

* **I progetti hardware** dispongono di una lista delle componenti necessarie: ogni **componente** ha un nome univoco, una descrizione, un prezzo, una quantità (>0).
* **I progetti software** dispongono di una lista dei profili necessari per lo sviluppo. Ogni **profilo** dispone di un nome (es. “Esperto AI”) e possiede una lista di competenze, ciascuna con un livello (numero intero tra 0 e 5).

Ogni utente della piattaforma(comprende creatore e amministratore) può finanziare un progetto: ogni **finanziamento** dispone di un importo ed una data. Un utente potrebbe inserire più finanziamenti per lo stesso progetto, ma in date diverse.

Nel momento in cui la somma totale degli importi dei finanziamenti supera il budget del progetto, oppure il progetto resta in stato aperto oltre la data limite, lo stato di tale progetto diventa pari a chiuso: un progetto chiuso non accetta ulteriori finanziamenti.

Ad ogni finanziamento è associata una *sola* reward, tra quelle previste per il progetto finanziato.

Un utente può inserire commenti relativi ad un progetto. Ogni **commento** dispone di un id(univoco), una data ed un campo testo.

Il creatore può inserire massimo una **risposta** per ogni singolo commento(è sottointeso che sia l'utente creatore di quel progetto). Una risposta dispone di un id(univoco), una data ed un campo testo.

Ogni utente della piattaforma può inviare una **candidatura** a un numero qualsiasi di profili, a patto che, per ciascun profilo, l’utente disponga, per ciascuna competenza, di un livello superiore o uguale al valore richiesto.  
Un progetto software può ricevere un numero qualsiasi di candidature per un certo profilo.  
Il creatore può accettare o rifiutare la candidatura.

**STEP2.** Costruire un glossario dei termini

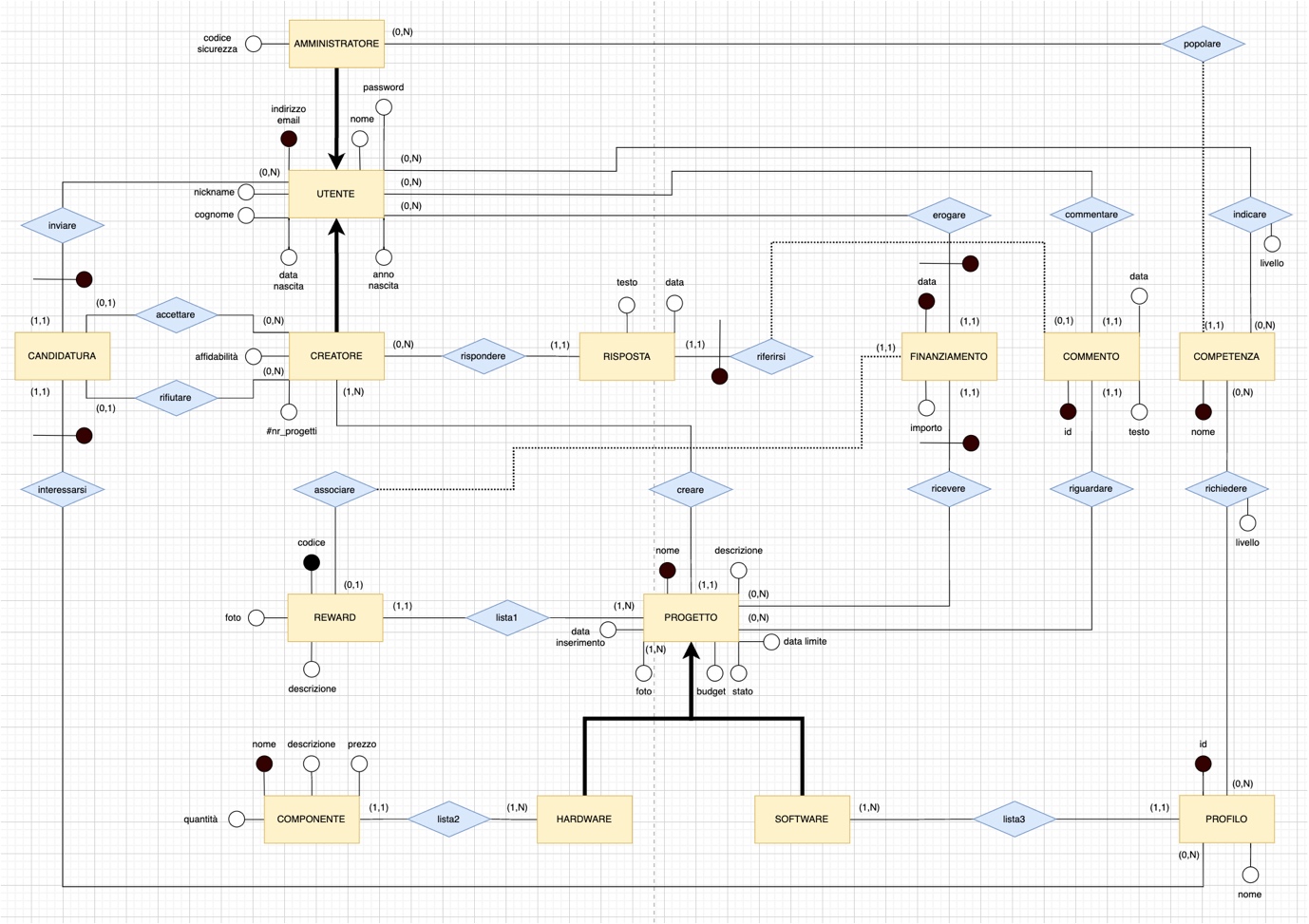
|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Utente | Persona che utilizza la piattaforma |  | Competenza, Finanziamento, Commento, Candidatura, Amministratore, Creatore |
| Competenza | Abilità o skill di un utente |  | Utente, Profilo, Amministratore |
| Amministratore | Utente autorizzato a popolare la lista delle competenze. |  | Utente, Competenza |
| Creatore | Utente autorizzato a inserire progetti e rispondere ai commenti. |  | Candidatura, Risposta, Utente,  Progetto |
| Progetto | Iniziativa di un creatore |  | Creatore, Reward, Prog. Hardware, Prog. Software,  Finanziamento,  Commento |
| Reward | Premio di un progetto |  | Progetto |
| Prog. Hardware | Tipologia di progetto |  | Progetto, Componente |
| Componente | Elementi fisici necessari per un prog. hardware |  | Prog. Hardware |
| Prog. Software | Tipologia di progetto |  | Progetto, Profilo |
| Profilo | Figura richiesta da un prog. software |  | Prog. Software, Competenza, Candidatura |
| Finanziamento | Contributo economico effettuato da un utente su un progetto |  | Utente, Progetto, Reward |
| Commento | Feedback di un utente su un progetto |  | Risposta, Utente, Progetto |
| Risposta | Replica del creatore a un commento |  | Creatore, Commento |
| Candidatura | Proposta lavorativa di un utente per un profilo |  | Utente, Creatore, Profilo |

**STEP3.** Definire le operazioni sui dati

* Inserire/modificare/visualizzare/eliminare un *utente della piattaforma*(utente generale, creatore o amministratore);
* Inserire ed associare una nuova competenza assegnare una competenza ad un utente della piattaforma specificando il livello (numero intero da 0 a 5);
* Modificare/visualizzare/eliminare una competenza;
* Inserire ed associare un nuovo progetto(hardware o software) ad un creatore;
* Modificare/visualizzare/eliminare un progetto;
* Inserire ed associare una nuova reward ad un progetto;
* Modificare/visualizzare/eliminare una reward;
* Inserire ed associare un nuovo componente ad un progetto hardware;
* Modificare/visualizzare/eliminare una componente;
* Inserire ed associare un nuovo profilo ad un progetto software;
* Modificare/visualizzare/eliminare un profilo;
* Inserire un finanziamento ed associarlo ad un utente della piattaforma e ad un progetto;
* Modificare/visualizzare/eliminare un finanziamento;
* Inserire un commento ed associarlo ad un utente della piattaforma e ad un progetto;
* Inserire una risposta ed associarla ad un creatore e ad un commento;
* Modificare/visualizzare/eliminare un commento/risposta;
* Inserire una candidatura ed associarla ad un utente della piattaforma e ad un profilo;
* Modificare/accettare/rifiutare/visualizzare/eliminare una candidatura.

**PROGETTAZIONE CONCETTUALE**

**DIAGRAMMA E-R**

****

Link: [Diagramma-ER-Bostarter](https://drive.google.com/file/d/1IbB-dAA9J_h8mX7nhNb2TmoWmLeT0Bp_/view?usp=sharing)

**BUSINESS RULES**

Regole di Vincolo  
• COMPETENZA.nome è una stringa.  
• INDICARE.livello è un numero intero tra 0 e 5.

• RICHIEDERE.livello è un numero intero tra 0 e 5.  
• PROGETTO.stato è un campo di tipo enum (aperto/chiuso).  
• COMPONENTE.quantità deve essere maggiore di 0.  
• Un utente può inserire più finanziamenti per lo stesso progetto, purché in date diverse.  
• Un utente inserire una candidatura su un profilo SOLO se, per ogni competenza richiesta da un profilo, dispone di un livello superiore o uguale al valore richiesto.

• Il creatore può decidere di accettare o rifiutare una candidatura(mutua esclusione).

Regole di Derivazione  
• Quando la somma totale degli importi dei finanziamenti supera il budget del progetto oppure il progetto resta in stato aperto oltre la data limite, lo stato del progetto viene automaticamente impostato su "chiuso" e il progetto non accetta ulteriori finanziamenti.

**DIZIONARIO DELLE ENTITA’**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatore** |
| UTENTE | Persona che utilizza la piattaforma | indirizzo email, nickname, password, nome, cognome, anno nascita, luogo nascita | indirizzo email |
| COMPETENZA | Abilità o skill di un utente | nome | nome |
| AMMINISTRATORE | Utente autorizzato a popolare la lista delle competenze. | indirizzo email, nickname, password, nome, cognome, anno nascita, luogo nascita, codice sicurezza | indirizzo email |
| CREATORE | Utente autorizzato a inserire progetti e rispondere ai commenti. | indirizzo email, nickname, password, nome, cognome, anno nascita, luogo nascita, affidabilità, #nr\_progetti | indirizzo email |
| PROGETTO | Iniziativa di un creatore | nome, descrizione, data inserimento, foto, data limite, budget, stato | nome |
| REWARD | Premio di un progetto | codice, descrizione, foto | codice |
| PROG. HARDWARE | Tipologia di progetto | nome, descrizione, data inserimento, foto, data limite, budget, stato | nome |
| COMPONENTE | Elementi fisici necessari per un prog. hardware | nome, descrizione, prezzo, quantità | nome |
| PROG. SOFTWARE | Tipologia di progetto | nome, descrizione, data inserimento, foto, data limite, budget, stato | nome |
| PROFILO | Figura richiesta da un prog. software | id, nome | id |
| FINANZIAMENTO | Contributo economico effettuato da un utente su un progetto | indirizzoEmailUtente, nomeProgetto, data, importo | indirizzoEmailUtente, nomeProgetto, data |
| COMMENTO | Feedback di un utente su un progetto | id, data, testo | id |
| RISPOSTA | Replica del creatore a un commento | id, data, testo | id |
| CANDIDATURA | Proposta lavorativa di un utente per un profilo | indirizzoEmailUtente,  idProfilo | indirizzoEmailUtente,  idProfilo |

**DIZIONARIO DELLE RELAZIONI**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazione** | **Descrizione** | **Componenti** | **Attributi** |
| popolare | Un amministratore popola la lista delle competenze | AMMINISTRATORE,  COMPETENZA |  |
| inviare | Un utente della piattaforma invia una candidatura | UTENTE,  CANDIDATURA |  |
| erogare | Un utente della piattaforma può erogare un finanziamento | UTENTE,  FINANZIAMENTO |  |
| commentare | Un utente della piattaforma può scrivere un commento | UTENTE, COMMENTO |  |
| indicare | Un utente della piattaforma può indicare il livello delle competenze che possiede | UTENTE, COMPETENZA | livello |
| accettare | Un creatore può accettare una candidatura | CREATORE,  CANDIDATURA |  |
| rifiutare | Un creatore può rifiutare una candidatura | CREATORE,  CANDIDATURA |  |
| interessarsi | Una candidatura interessa un profilo | CANDIDATURA,  PROFILO |  |
| rispondere | Un creatore può rispondere ai commenti | CREATORE,  RISPOSTA |  |
| riferirsi | Una risposta si riferisce ad un commento specifico | RISPOSTA,  COMMENTO |  |
| associare | Un finanziamento viene associato ad una reward | FINANZIAMENTO,  REWARD |  |
| creare | Un creatore può creare progetti | CREATORE,  PROGETTO |  |
| ricevere | Un progetto riceve dei finanziamenti | PROGETTO,  FINANZIAMENTO |  |
| riguardare | Un commento riguarda un progetto specifico | COMMENTO,  PROGETTO |  |
| richiedere | Un profilo richiede un livello minimo a delle competenze | PROFILO,  COMPETENZA | livello |
| lista1 | La lista delle reward di un progetto | PROGETTO,  REWARD |  |
| lista2 | La lista delle componenti di un progetto hardware | PROG. HARWARE, COMPONENTE |  |
| lista3 | La lista dei profili necessari di un progetto software | PROG. SOFTWARE,  PROFILI |  |

**PROGETTAZIONE LOGICA**

**RISTRUTTURAZIONE DELLO SCHEMA CONCETTUALE**

* Eliminazione delle generalizzazioni:

L’entità utente è una generalizzazione parziale🡪

SOL3: Sostituzione delle generalizzazione con relazioni tra entità genitore ed entità figlie(vincolo: un’occorrenza del padre non può partecipare in contemporanea a tutte le sue figlie).

L’entità progetto è una generalizzazione totale🡪

SOL1: Accorpamento delle entità figlie nell’entità genitore(si sottintende il fatto che se un progetto ha componenti, allora è una progetto hardware. Se ha profili, allora è un progetto software. Non è possibile avere componenti e profili allo stesso tempo).

* Eliminazione degli attributi multivalore:

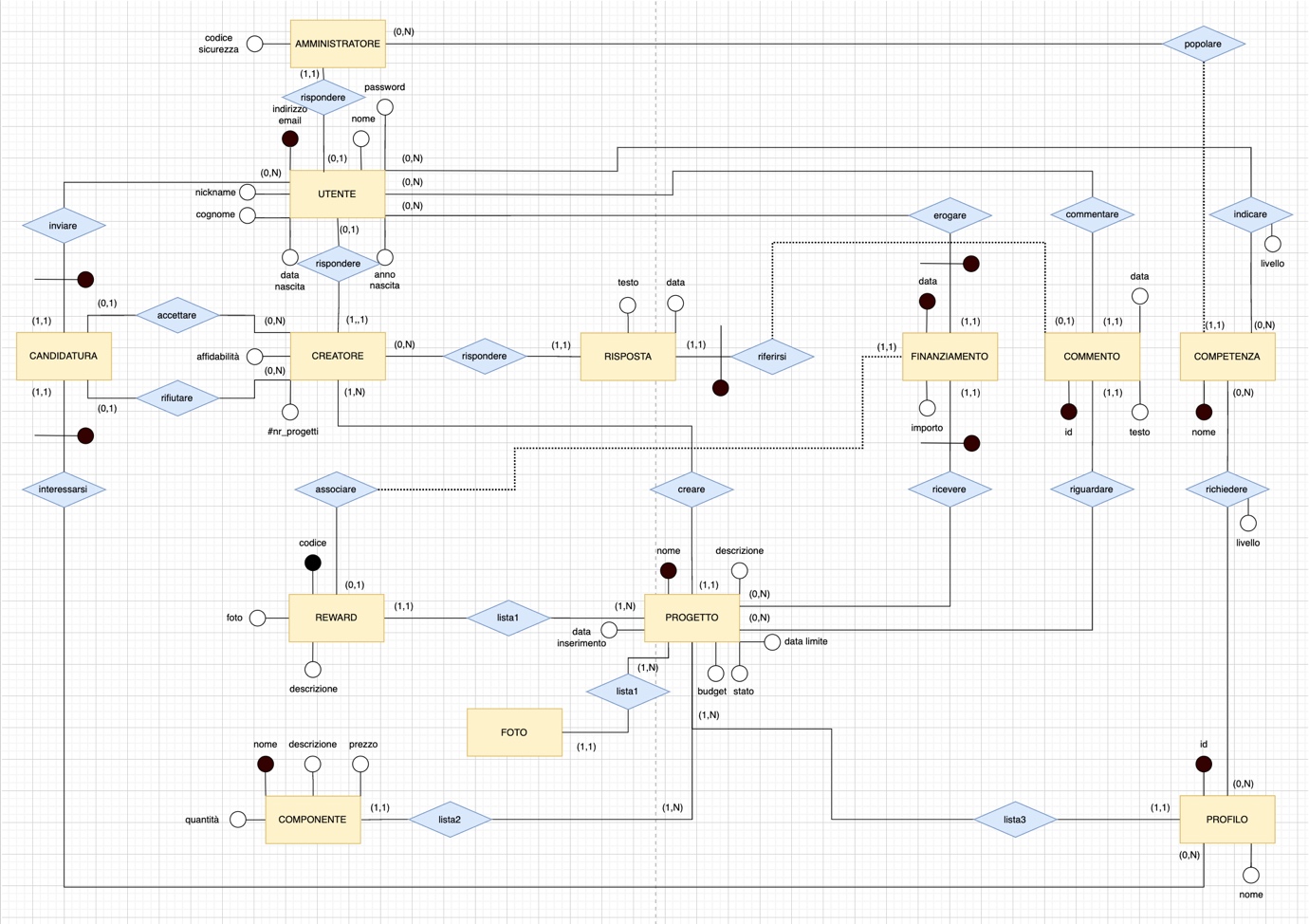
PROGETTO.foto 🡪 foto diventa un’entità collegata a PROGETTO da una cardinalità(1,n).

* Partizionamento/accorpamento di concetti:

Nessuna operazione necessaria, poiché è necessario avere una stima sul volume dei dati per

un’indicazione se/come partizionare/accorpare entità.

Dopo aver apportato le seguenti modifiche, il diagramma ER risulta così:



Link: [Diagramma-ER-2.0](https://drive.google.com/file/d/1oqoFXEKOS1dXuxWIRi6LQue4Lr0LzIMT/view?usp=sharing)

**ANALISI DELLA RIDONDANZA #nr\_progetti (attributo di creatore)**

Tabelle dei volumi:

1. Aggiungere(write) un nuovo progetto ad un utente creatore esistente (1 volte/mese, interattiva)
2. Visualizzare(read) tutti i progetti e tutti i finanziamenti (1 volta/mese, batch)
3. Contare(read) il numero di progetti associati ad uno specifico utente (3 volte/mese, batch)

Coefficienti per l’analisi:

wI = 1

wB = 0,5

a = 2

Tabella dei volumi:

10 progetti

3 finanziamenti per progetto

5 utenti

2 progetti per Utente

|  |  |  |
| --- | --- | --- |
|  | CON | SENZA |
| 1) | 1\*1\*(2\*3)=6 | 1\*1\*(2\*2)=4 |
| 2) | 1\*0,5\*(5\*2 + 10\*3)= 20 | 20 |
| 3) | 3\*0,5\*(1)=1,5 | 3\*0,5\*(2)=3 |
|  | ***TOTALE = 27,5*** | TOTALE = 27 |

**LISTA DELLE TABELLE CON VINCOLI DI CHIAVI**

UTENTE(**indirizzoEmail**, nickname, password, nome, cognome, anno nascita, luogo nascita)

INDICARE(**indirizzoEmailUtente, nomeCompetenza,** livello)

COMPETENZA(**nome**, indirizzoEmailAmministratore)

AMMINISTRATORE(**indirizzoEmailUtente**, codice sicurezza)

CREATORE(**indirizzoEmailUtente**, affidabilità, #nr\_progetti)

PROGETTO(**nome**, descrizione, data inserimento, data limite, budget, stato, indirizzoEmailCreatore)

FOTO(**nomeProgetto**, immagine)

COMPONENTE(**nome**, descrizione, prezzo, quantità, nomeProgetto)

REWARD(**codice**, descrizione, foto, nomeProgetto)

PROFILO(**id**, nome, nomeProgetto)

RICHIEDERE(**idProfilo, nomeCompetenza**, livello)

FINANZIAMENTO(**indirizzoEmailUtente, nomeProgetto, data**, importo, codiceReward)

COMMENTO(**id**, data, testo, indirizzoEmailUtente, nomeProgetto)

RISPOSTA(**idCommento**, data, testo, indirizzoEmailCreatore)

CANDIDATURA(**indirizzoEmailUtente, idProfilo**)

ACCETTARE(**indirizzoEmailUtente, idProfilo, indirizzoEmailCreatore)**

RIFIUTARE(**indirizzoEmailUtente, idProfilo, indirizzoEmailCreatore)**

**LISTA DEI VINCOLI INTER-RELAZIONALI**

|  |  |  |
| --- | --- | --- |
| INDICARE.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| INDICARE.nomeCompetenza | -----------> | COMPETENZA.nome |
| COMPETENZA.indirizzoEmailAmministratore | -----------> | AMMINISTRATORE.indirizzoEmailUtente |
| AMMINISTRATORE.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| CREATORE.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| PROGETTO.indirizzoEmailCreatore | -----------> | CREATORE.indirizzoEmailUtente |
| FOTO.nomeProgetto | -----------> | PROGETTO.nome |
| COMPONENTE.nomeProgetto | -----------> | PROGETTO.nome |
| REWARD.nomeProgetto | -----------> | PROGETTO.nome |
| PROFILO.nomeProgetto | -----------> | PROGETTO.nome |
| RICHIEDERE.idProfilo | -----------> | PROFILO.id |
| RICHIEDERE.nomeCompetenza | -----------> | COMPETENZA.nome |
| FINANZIAMENTO.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| FINANZIAMENTO.nomeProgetto | -----------> | PROGETTO.nome |
| FINANZIAMENTO.codiceReward | -----------> | REWARD.codice |
| COMMENTO.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| COMMENTO.nomeProgetto | -----------> | PROGETTO.nome |
| RISPOSTA.indirizzoEmailCreatore | -----------> | CREATORE.indirizzoEmailUtente |
| RISPOSTA.idCommento | -----------> | COMMENTO.id |
| CANDIDATURA.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| CANDIDATURA.idProfilo | -----------> | PROFILO.id |
| ACCETTARE.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| ACCETTARE.idProfilo | -----------> | PROFILO.id |
| ACCETTARE.indirizzoEmailCreatore | -----------> | CREATORE.indirizzoEmailUtente |
| RIFIUTARE.indirizzoEmailUtente | -----------> | UTENTE.indirizzoEmail |
| RIFIUTARE.idProfilo | -----------> | PROFILO.id |
| RIFIUTARE.indirizzoEmailCreatore | -----------> | CREATORE.indirizzoEmailUtente |
|  |  |  |

**NORMALIZZAZIONE**

L'obiettivo della normalizzazione è garantire che lo schema relazionale sia privo di anomalie di aggiornamento, inserimento e cancellazione. La normalizzazione si articola in due aspetti fondamentali:

1. L'eliminazione delle ridondanze fisiche (duplicazione di dati nelle tabelle)
2. L'identificazione e gestione controllata delle ridondanze concettuali (attributi derivabili)

**STEP1.**

Il primo passo consiste nella verifica formale che tutte le relazioni soddisfino almeno la Terza Forma Normale.

**Prima Forma Normale (1NF)**

Una relazione è in 1NF se e solo se tutti gli attributi contengono solo valori atomici e non sono complessi(più tipi).

Tutte le relazioni del nostro schema soddisfano la 1NF poiché:

* Ogni attributo contiene valori atomici
* Non sono presenti attributi multivalore o gruppi ripetuti

Si nota che l’attributo potenzialmente multivalore è *luogo di nascita* di UTENTE, che per semplicità abbiamo lasciato così, ma si potrebbe sostituire con *paese, provincia, città e numero civico.*

**Seconda Forma Normale (2NF)**

Una relazione è in 2NF se e solo se è in 1NF e ogni attributo non-primo dipende funzionalmente in modo completo dalla chiave primaria(ovvero che non ci sono dipendenze parziali).

Le tabelle con chiavi composte richiedono particolare attenzione:

1. **INDICARE(indirizzoEmailUtente, nomeCompetenza, livello)**:
   * Dipendenza funzionale: {indirizzoEmailUtente, nomeCompetenza} → {livello}
   * Non esistono dipendenze parziali del tipo {indirizzoEmailUtente} → {livello} o {nomeCompetenza} → {livello}
2. **RICHIEDERE(idProfilo, nomeCompetenza, livello)**:
   * Dipendenza funzionale: {idProfilo, nomeCompetenza} → {livello}
   * Non esistono dipendenze parziali
3. **FINANZIAMENTO(indirizzoEmailUtente, nomeProgetto, data, importo, codiceReward)**:
   * Dipendenza funzionale: {indirizzoEmailUtente, nomeProgetto, data} → {importo, codiceReward}
   * Non esistono dipendenze parziali
4. **CANDIDATURA(indirizzoEmailUtente, idProfilo)**, **ACCETTARE** e **RIFIUTARE**:
   * Non contengono attributi non-primi
   * Risultano automaticamente in 2NF

Queste tabelle e le restanti con chiave semplice sono dunque in 2NF.

**Terza Forma Normale (3NF)**

Una relazione è in 3NF se e solo se è in 2NF e per ogni dipendenza funzionale non banale X → A, almeno una delle seguenti condizioni è verificata:

* X è una superchiave della relazione
* A è un attributo primo
* A appartiene a X

Analizziamo tabella per tabella, risulta che in tutte le dipendenze funzionali non banali X → A, X è sempre una superchiave, soddisfacendo così la prima condizione della 3NF.

**Forma Normale di Boyce-Codd (FNBC)**

Una relazione è in BCNF se e solo se, per ogni dipendenza funzionale non banale X → Y, X è una superchiave della relazione.

Nel nostro schema, in ogni tabella:

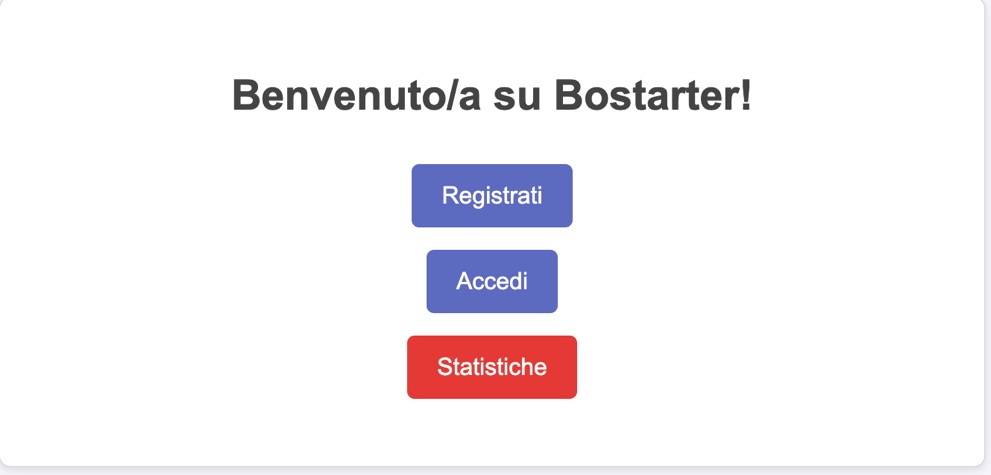
* Le uniche dipendenze funzionali sono della forma "chiave → attributi"
* Non esistono determinanti che non siano superchiavi
* Tutte le tabelle risultano in FNBC

**STEP2.**

Dopo aver verificato l'assenza di ridondanze fisiche mediante la normalizzazione, procediamo all'identificazione delle ridondanze concettuali. Sono attributi derivabili da altri attributi o relazioni presenti nel database. Nel nostro schema sono presenti i seguenti casi:

1. **Attributo #nr\_progetti in CREATORE**:
   * Definizione: conteggio del numero di progetti inseriti dall'utente creatore
   * Derivabilità: potrebbe essere calcolato dinamicamente con una query di conteggio sulla tabella PROGETTO
   * Giustificazione: l'analisi dei volumi ha dimostrato la convenienza di mantenere questo valore pre-calcolato
   * Controllo: viene aggiornato tramite trigger ogni volta che un creatore inserisce un nuovo progetto
2. **Attributo affidabilità in CREATORE**:
   * Definizione: percentuale di progetti che hanno ottenuto almeno un finanziamento
   * Derivabilità: calcolabile mediante query complesse che coinvolgono le tabelle PROGETTO e FINANZIAMENTO
   * Giustificazione: necessario per operazioni frequenti (come visualizzare la classifica degli utenti)
   * Controllo: aggiornato tramite trigger all'inserimento di nuovi progetti (denominatore) e nuovi finanziamenti (numeratore)
3. **Attributo stato in PROGETTO**:
   * Definizione: indicatore binario (aperto/chiuso) dello stato del progetto
   * Derivabilità: determinabile confrontando la somma dei finanziamenti con il budget o la data corrente con la data limite
   * Giustificazione: semplifica le query di selezione dei progetti disponibili per finanziamento
   * Controllo: gestito tramite trigger quando la somma dei finanziamenti supera il budget e tramite evento MySQL per la verifica della data limite

**DESCRIZIONE DELL’APP WEB**



Ecco un’overview delle feature aggiuntive e delle scelte architetturali che ho introdotto oltre alle operazioni e statistiche richieste:

* **Gestione ruoli in sessione e filtraggio UI**: ho centralizzato il ruolo dell’utente ($\_SESSION['ruolo']) in fase di login, e nel front-end mostro/nascondo dinamicamente pulsanti e form in base a questo valore.
* **Gestione eccezioni**: ho cercato di gestire tutte le eccezioni possibili seguendo una validazione a più livello, ovvero sia a livello di database(stored procedures) sia a livello applicativo(script .php).
* **Modal dinamici per input contestuale**: per ogni operazione di inserimento ho utilizzato dei modali JavaScript/CSS anziché pagine separate, migliorando l’usabilità senza ricaricare l’intera pagina.
* **Logout “vero” e prevenzione back-button**: alla pressione di “Esci” la sessione viene distrutta (session\_destroy()), e grazie a header di cache-control/social headers impedisco il ritorno alla dashboard tramite il tasto “Indietro” del browser, simulando un ambiente di produzione più sicuro.
* **Modulo di logging esterno**: hai isolato tutta la logica di scrittura su MongoDB in un file log\_event.php (incluso con require\_once) che espone solo la funzione logEvent(). Così gli script rimangono inalterati: basta invocare logEvent(“message”)garantendo un tracciamento centralizzato e manutenibile.
* **Vincolo del creatore**: i creatori possono eseguire le operazione aggiuntive solo su progetti creati da lui stesso.
* **Operazione inserisciComponente**: ho aggiunto la possibilità per i creatori di aggiungere componenti necessari per un progetto.
* **Visualizzare il profilo candidato**: quando il creatore apre la lista di candidature, può vedere per ogni candidatura il profilo del candidato.

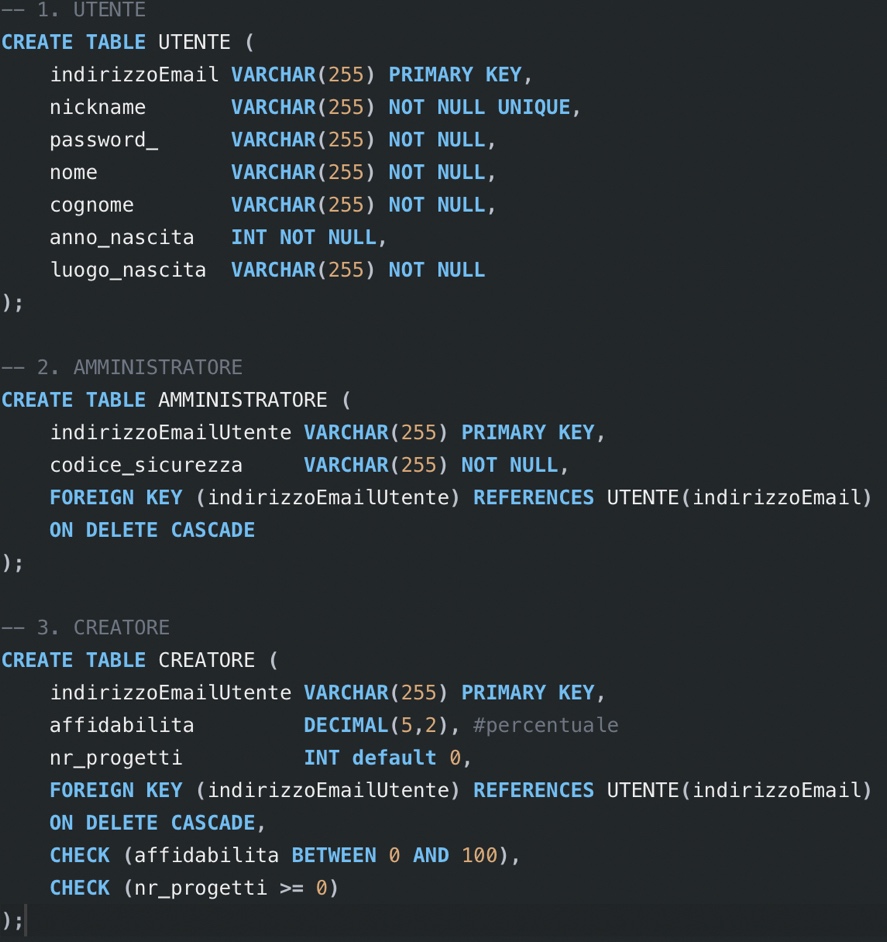
Infine ho creato un diagramma di flusso semplificato degli script .php per tenere traccia delle interazioni tra le pagine:

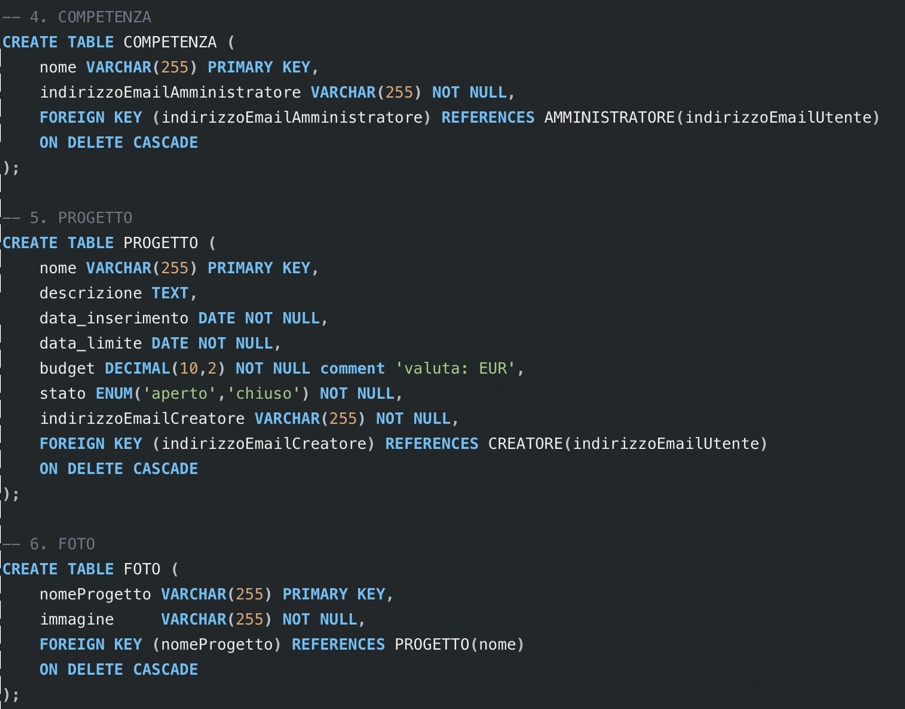
**A diagram of a website

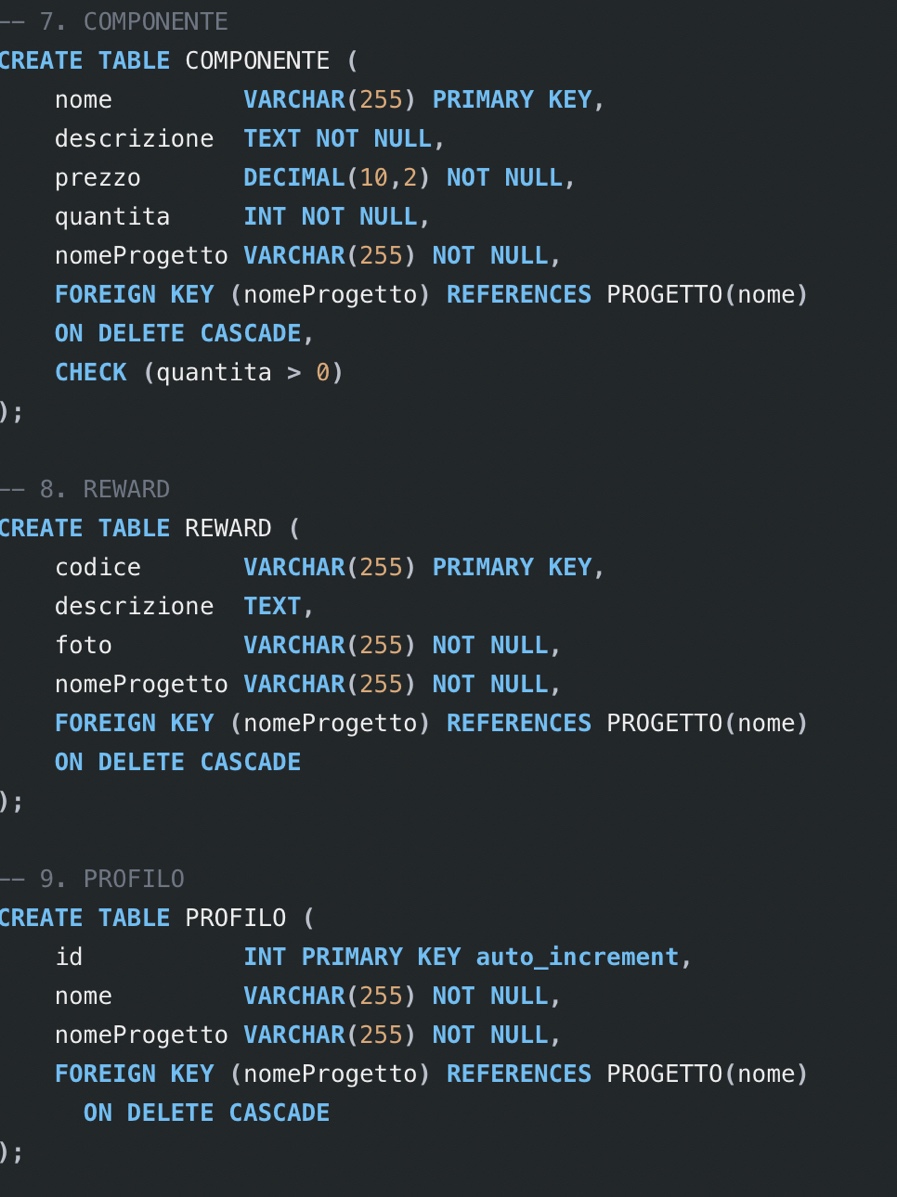
AI-generated content may be incorrect.**

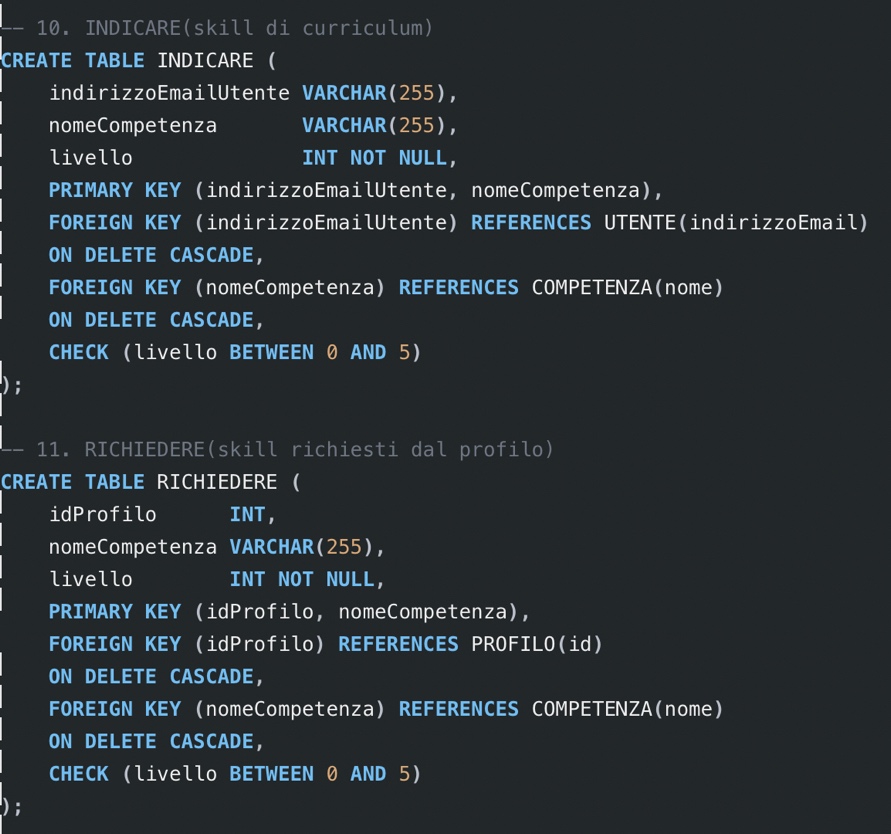
**CODICE SQL**

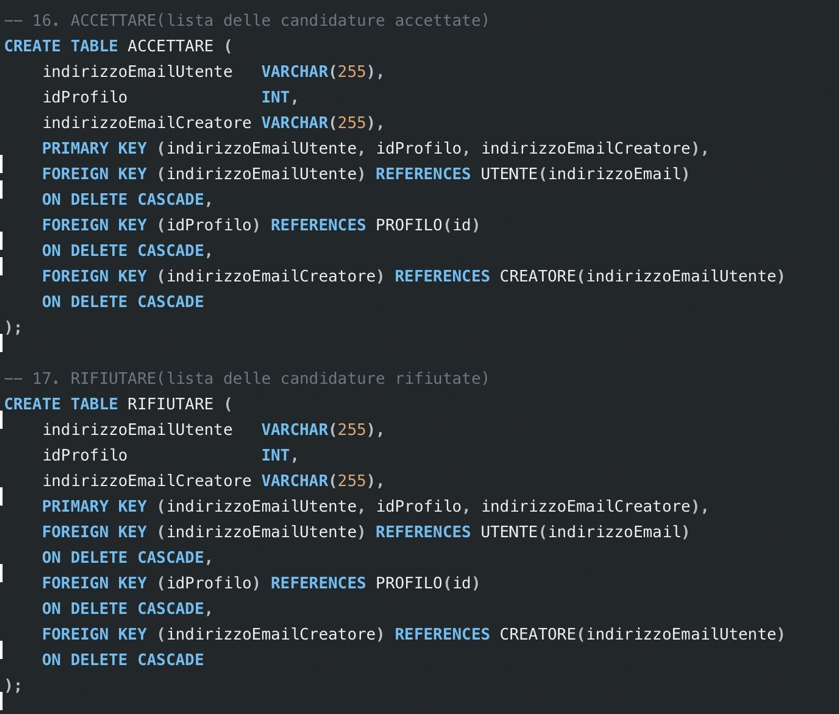
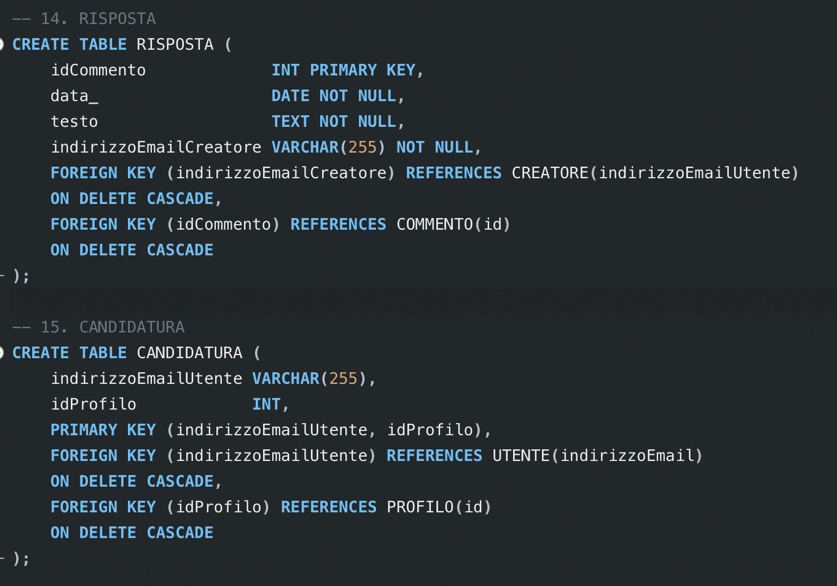
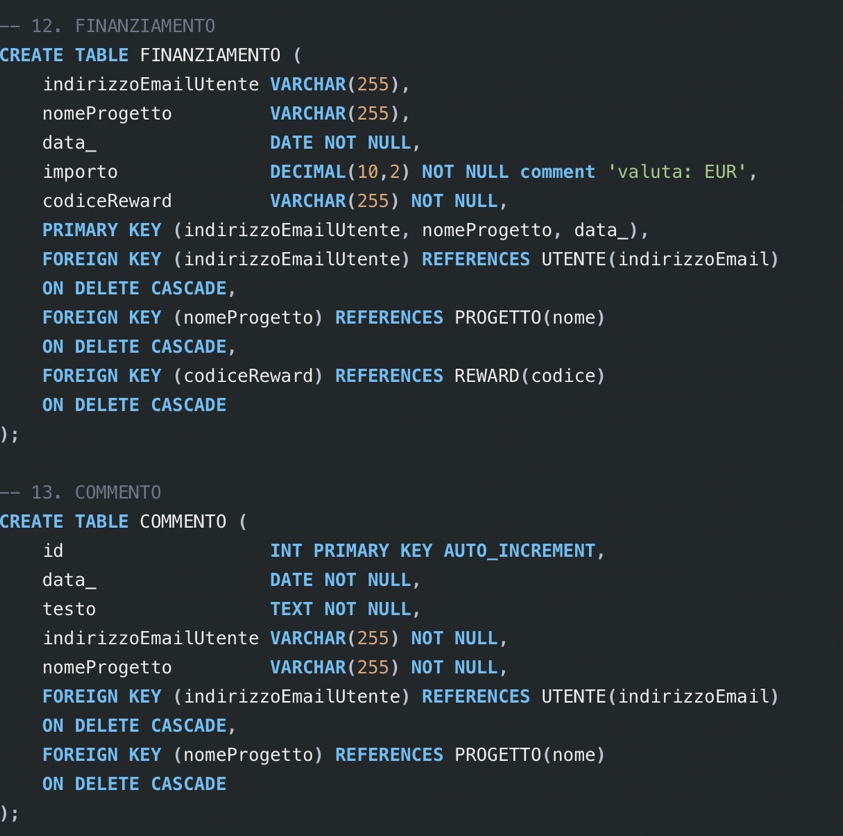
Queste sone le **tabelle:**











Queste sono le **stored procedures:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with text and numbers

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with text and images

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with white and green text

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A computer screen with white and blue text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with text and numbers

AI-generated content may be incorrect.

Infine, abbiamo **view**, **event** e **trigger**:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.**r**:

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.