

Indice

1	Introduzione	2
1.1	Descrizione del problema	2
2	Progettazione Concettuale	3
2.1	Class Diagram	3
2.2	Ristrutturazione del Class Diagram	4
2.2.1	Analisi delle chiavi	4
2.2.2	Analisi degli attributi derivati	4
2.2.3	Analisi delle ridondanze	4
2.2.4	Analisi degli attributi strutturati	4
2.2.5	Analisi degli attributi a valore multiplo	4
2.2.6	Analisi delle gerarchie di specializzazione	4
2.3	Class Diagram ristrutturato	5
2.4	Dizionario delle Classi	6

1 Introduzione

Il seguente elaborato ha lo scopo di documentare la progettazione e lo sviluppo di una base di dati relazionale del DBMS PostgreSQL per il corso di Basi di Dati I. Il Database nasce per la gestione di un Hackathon, ovvero una maratona di hacking.

1.1 Descrizione del problema

Il documento descrive la progettazione e lo sviluppo di una base di dati relazionale finalizzata alla realizzazione di un sistema per la gestione di Hackathon, consentendo l'organizzazione degli eventi e la storicizzazione dei progetti realizzati dai vari team.

Il sistema consente la gestione completa di un Hackathon, permettendo la registrazione degli utenti, la formazione dei team e la pubblicazione dei progetti. Ogni team collabora per risolvere il problema proposto, caricando periodicamente aggiornamenti relativi ai propri progressi, che possono essere visionati e commentati dai giudici. I giudici vengono selezionati dall'organizzatore tra gli utenti registrati alla piattaforma e, al termine dell'hackathon, assegnano una valutazione (da 0 a 10) a ciascun team. La piattaforma raccoglie le valutazioni e genera automaticamente la classifica finale. In questo modo, il sistema permette non solo di organizzare e monitorare le attività dell'hackathon, ma anche di storicizzare i progetti e i risultati dei vari team.

2 Progettazione Concettuale

In questo capitolo verrà affrontata la progettazione del database al suo livello più alto. Dall'analisi dei requisiti da soddisfare, si giungerà alla definizione di uno schema concettuale indipendente dalla struttura dei dati e dall'implementazione fisica. Tale schema è rappresentato con un Class Diagram UML, che metterà in evidenza le entità del problema, i relativi attributi, le relazioni tra esse e gli eventuali vincoli.

2.1 Class Diagram

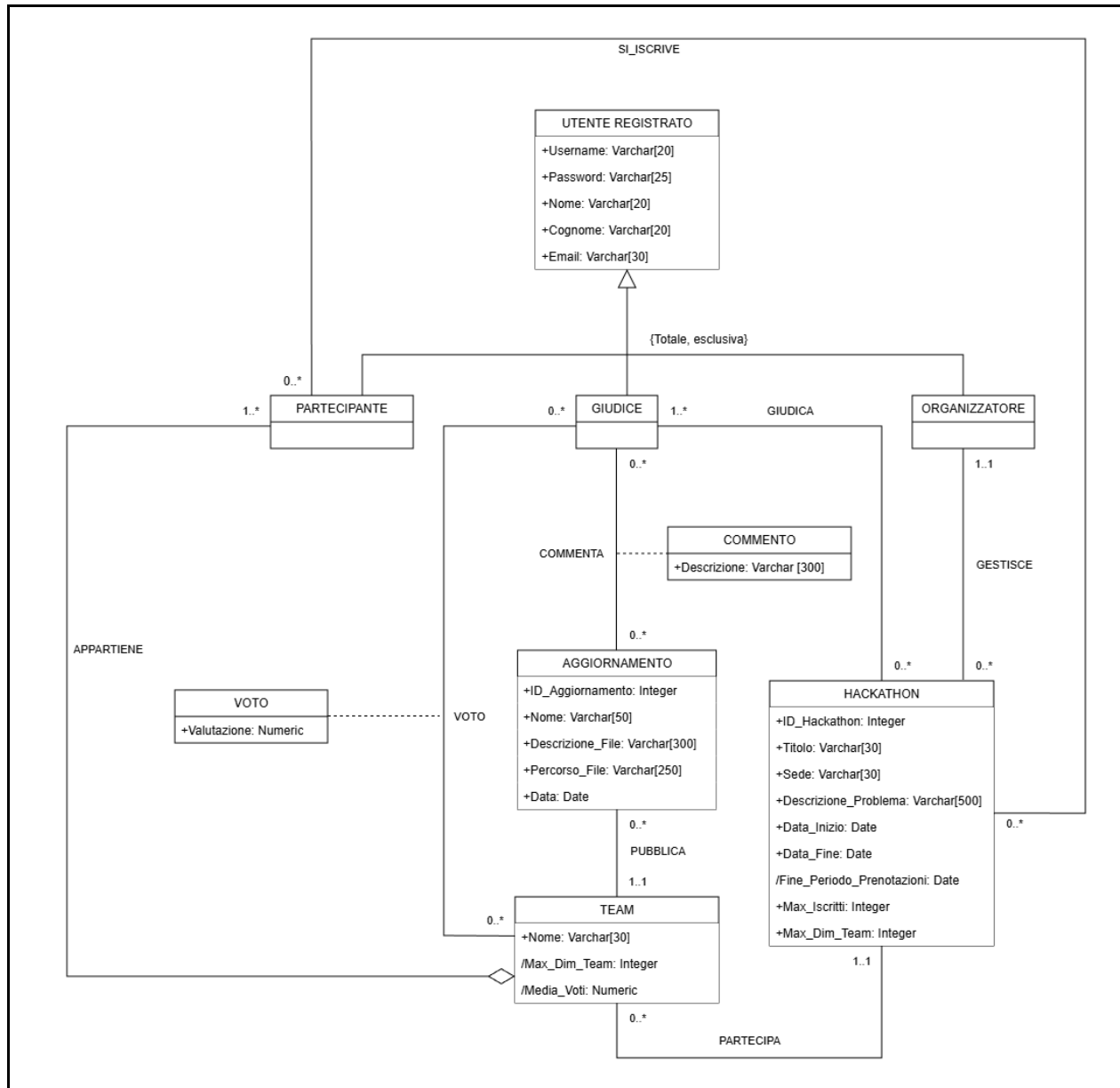


Figura 1: Class Diagram UML

2.2 Ristrutturazione del Class Diagram

Al fine di derivare lo schema logico, è necessario rielaborare il Class Diagram iniziale mediante un processo di ristrutturazione; questa fase ha lo scopo di analizzare il diagramma e rimuovere, se presenti, le criticità strutturali per garantire una corretta implementazione sul DBMS PostgreSQL. La fase di ristrutturazione seguirà i seguenti punti:

- Analisi delle chiavi
- Analisi degli attributi derivati
- Analisi delle ridondanze
- Analisi degli attributi strutturati
- Analisi degli attributi a valore multiplo
- Analisi delle gerarchie di specializzazione

2.2.1 Analisi delle chiavi

2.2.2 Analisi degli attributi derivati

2.2.3 Analisi delle ridondanze

2.2.4 Analisi degli attributi strutturati

2.2.5 Analisi degli attributi a valore multiplo

2.2.6 Analisi delle gerarchie di specializzazione

2.3 Class Diagram ristrutturato

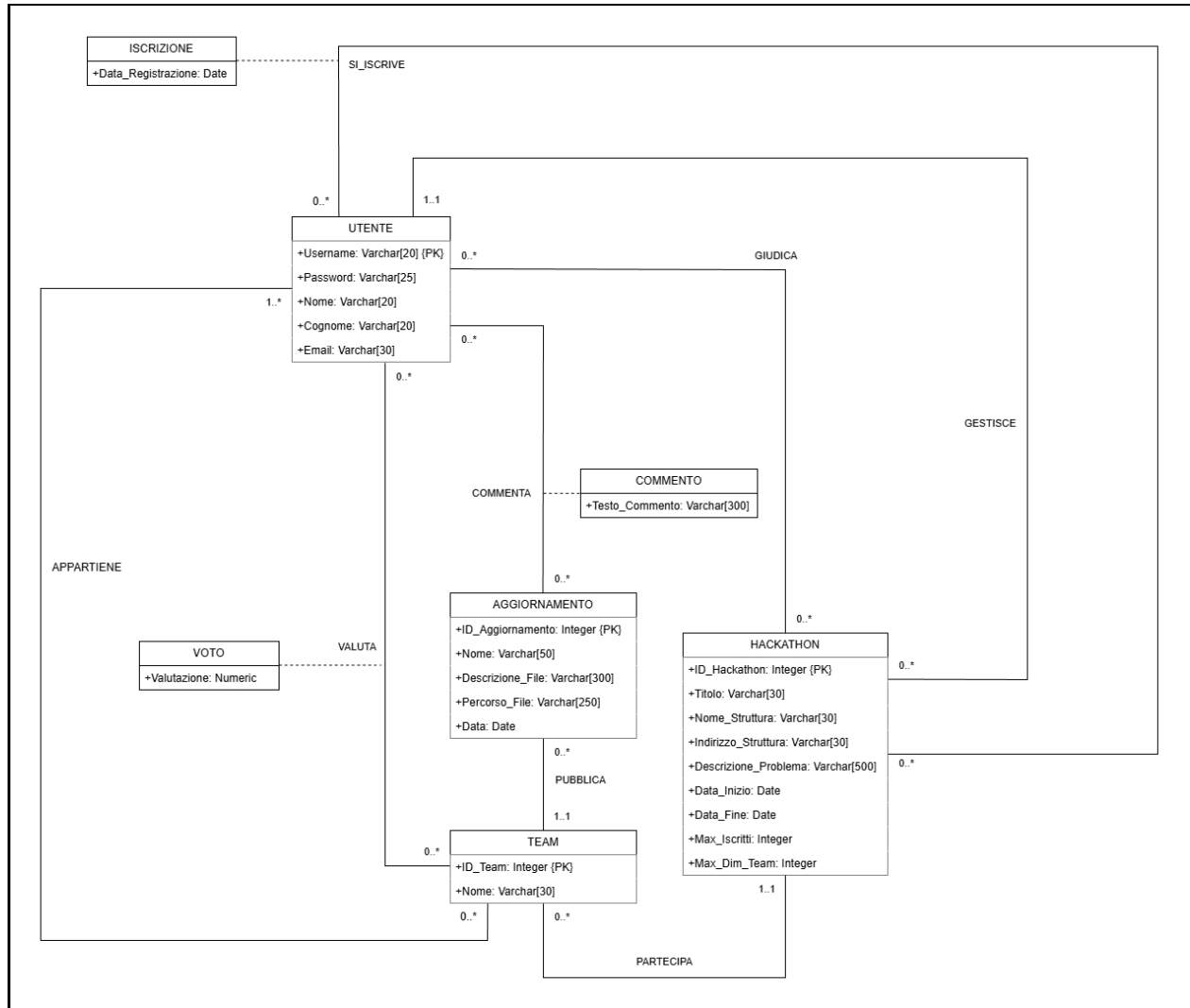


Figura 2: Class Diagram ristrutturato UML

2.4 Dizionario delle Classi

Classe	Descrizione	Attributi
UTENTE	Descrive ciascun utente iscritto alla piattaforma, non obbligatoriamente iscritto ad un Hackathon. Quest'ultimo può essere nominato giudice da un organizzatore, o organizzare un Hackathon diventando dunque organizzatore.	<ul style="list-style-type: none">• Username(varchar(20)): Username identificativo per un utente.• Password(varchar(25)): Password di un utente.• Nome(varchar(20)): Nome di un utente.• Cognome(varchar(20)): Cognome di un utente.• Email(varchar(30)): Email di un utente.
HACKATHON	Descrive una maratona di hacking, organizzata da un utente, nella quale team e utenti partecipanti si affrontano, dando una propria risoluzione del problema proposto.	<ul style="list-style-type: none">• ID_Hackathon(Integer): Codice identificativo di un Hackathon.• Titolo(varchar(30)): Nome dell'Hackathon.• Nome_struttura(varchar(30)): Nome della struttura che ospita un Hackathon.• Indirizzo_struttura(varchar(30)): Indirizzo della struttura.• Descrizione_problema(varchar(500)): Descrizione del problema proposto.• Data_inizio(Date): Data di inizio Hackathon.• Data_fine(Date): Data di fine Hackathon.• Max_iscritti(Integer): Numero massimo di utenti iscritti.• Max_dim_team(Integer): Dimensione massima di un team.
AGGIORNAMENTO	Descrive il contenuto di un aggiornamento in merito al problema proposto, viene caricato da un team.	<ul style="list-style-type: none">• ID_Aggiornamento(Integer): Codice identificativo di un aggiornamento.• Nome(varchar(50)): Nome dell'aggiornamento.• Descrizione_File(varchar(300)): Breve descrizione riguardante il contenuto dell'aggiornamento.• Percorso_File(varchar(250)): Stringa che indica il percorso del file.• Data(Date): Data in cui un aggiornamento viene caricato.
TEAM	Descrive un team di lavoro composto da utenti.	<ul style="list-style-type: none">• ID_Team(Integer): Codice identificativo di un team.• Nome(varchar(30)): Nome di un team.

Documentazione Hackathon per Basi Di Dati

COMMENTO	Descrive un commento, scritto da un giudice, per un aggiornamento caricato da un team.	<ul style="list-style-type: none">• Testo_commento(Varchar(300)): Contenuto del commento scritto dal giudice.
VOTO	Descrive un voto assegnato da un giudice ad un team a fine Hackathon.	<ul style="list-style-type: none">• Valutazione(Numeric(2,0)): Voto assegnato da un giudice.
ISCRIZIONE	Descrive un utente iscritto ad un Hackathon.	<ul style="list-style-type: none">• Data_registrazione(Date): Data di registrazione di un utente ad un Hackathon.