Text

Description automatically generated

GRUPPO

NOME MATRICOLA

Jinpeng Zhang 886854

Rebecca Frisoni 885771

Martina Ragusa 885113

A.A. 2022/2023

WebApp per la creazione e gestione di prove ed esami

Graphical user interface, application, icon

Description automatically generated

Table of Contents

[Introduzione 2](#_Toc137204366)

[Funzionalità principali 2](#_Toc137204367)

PROGETTAZIONE CONCETTUALE E LOGICA 2-4

QUERY PRINCIPALI 5-6

PRINCIPALI SCELTE PROGETTUALI 6

# Introduzione

Il gruppo ha scelto di sviluppare il progetto per la gestione degli esami universitari. In questo documento verrà illustrata la struttura del progetto e come è stato sviluppato. Inizialmente verranno descritte le funzionalità principali fornite dall’applicazione. Successivamente verrà spiegato come è stata effettuata la modellazione concettuale e logica della base di dati e verrà illustrata una selezione di query importanti del progetto. Per finire, verranno specificate le scelte progettuali che sono state utilizzate.

# Funzionalità principali

L’applicativo sviluppato permette di gestire le funzioni principali riguardanti gli esami universitari. In particolare, i docenti hanno la possibilità di creare nuovi esami, composti da una o più prove. Gli studenti possono sostenere tali prove ad uno degli appelli messi a disposizione. Il superamento di una prova viene registrato con il voto e la data che specifica fino a quanto tale votazione è valida.

Vi è la possibilità di verificare lo stato di tutti gli studenti, visualizzando le prove che sono state sostenute e quali sono state superate, oltre che lo stato degli appelli, avendo la possibilità di appurare la lista degli studenti che hanno superato le prove. È inoltre possibile acquisire l’elenco degli studenti che possono registrare l’esame, a seguito del superamento di tutte le prove che lo compongono.

Progettazione concettuale e logica

* 1. Progettazione concettuale

Abbiamo utilizzato il modello concettuale per modellare i dati della nostra base di dati. Come primo step abbiamo individuato le entità principali coinvolte nel dominio di interesse: Studenti, Esami, Prove e Docenti. La classe Studenti presenta tre attributi: il nome (di tipo stringa), il cognome (stringa) e il numero di matricola (stringa), che costituisce la chiave primaria. La classe Esami contiene le informazioni relative agli esami specificando tre attributi: il nome dell’esame (stringa), il codice identificativo (stringa), che costituisce la chiave primaria, e il numero delle prove (di tipo intero). La classe Docenti contiene le informazioni relative ai docenti, specificate dagli attributi: nome (stringa), cognome (stringa), indirizzo e-mail (stringa), password (stringa) e l’ID (chiave primaria, di tipo intero). Successivamente, vi è la classe Prove, che contiene, appunto, informazioni riguardanti la prova stessa, quali la tipologia (stringa), il tipo di voto (stringa), e l’identificativo (chiave primaria, di tipo intero).

La classe Studenti è messa in relazione con la classe Esami attraverso una relazione ‘molti a molti’ Registrazione, la quale contiene informazioni sul voto (di tipo int), la data di superamento (Date) e se l’esame è stato superato o meno (stato superamento, di tipo bool). Studenti possiede un’altra relazione ‘molti a molti’ “Appelli” che contiene informazioni riguardanti un appello dell’esame. Essa ha come attributi: il voto (int), la data superamento (Date), la data di scadenza del voto (Date) e lo stato superamento (bool). La classe Esami è, a sua volta, messa in relazione con la classe Docenti attraverso la relazione ‘molti a molti’ denominata “CreanoEsami”. Essa è inoltre messa in relazione con la classe Prove con una ‘uno a molti’, denominata “Associate”. La classe Prove è, a sua volta, messa in relazione con la classe Docenti con una relazione ‘molti a uno’ denominata “CreanoProve”.

A picture containing text, diagram, screenshot, number

Description automatically generated

* 1. Progettazione logica

Successivamente, partendo dalla progettazione concettuale, abbiamo proseguito con la progettazione logica.

Abbiamo trasformato le relazioni ‘molti a uno’ e ‘uno a molti’ aggiungendo le chiavi esterne necessarie. In Prove è stata aggiunta la chiave esterna IdE di tipo stringa, che punta ad Esami. Per la relazione CreanoEsami, invece, è stata creata un’ulteriore tabella Creazione, contenente le due chiavi esterne IdD per Docenti (di tipo int) e IdE per Esami (di tipo stringa). Successivamente sono state trasformate le relazioni molti a molti. All’interno di Registrazione sono state aggiunte le due chiavi esterne IdS e IdE, entrambe di tipo stringa, rispettivamente per la tabella Studenti e per la tabella Esami. Similmente, all’interno di Appelli sono state aggiunte IdS e IdP, chiavi esterne per Studenti, di tipo stringa, e Prove, di tipo int.

A picture containing text, diagram, parallel, plan

Description automatically generated

Query principali

Di seguito verranno commentate alcune delle query principali, in particolare quella per visualizzare la lista degli studenti che hanno superato un esame in particolare e quella per visualizzare la lista degli esami che sono stati superati da uno studente.

Per quanto riguarda la prima query, necessaria per avere una visione di tutti gli studenti che hanno superato un determinato esame, la tabella degli Studenti viene messa in relazione con la tabella Appelli utilizzando l’attributo idS di entrambe le tabelle. Vengono poi effettuate due ulteriori operazioni di join con le tabelle Prova ed Esame, utilizzando, rispettivamente, idP e idE. Il risultato delle operazioni di giunzione viene filtrato selezionando solamente gli esami il cui idE della tabella Studente coincida con l’omonimo attributo della tabella Esame. Per selezionare solamente gli esami che sono stati effettivamente superati, si verifica che lo “stato\_superamento” sia “True”.

È stato utilizzata la clausola GROUP BY(Esame.idE, Studente.idS) per effettuare l’operazione di raggruppamento e dunque partizionare le righe del risultato utilizzando i due attributi idE e idS. Infine, viene utilizzata la clausola HAVING per mostrare solamente gli studenti che hanno superato le prove che ammontano a 100 con la percentuale di superamento.

A picture containing text, screenshot, font

Description automatically generated

La query seguente, invece, viene utilizzata per visualizzare la lista di tutti gli esami che sono stati superati da uno studente, ovvero deve essere verificata la condizione per cui tutte le prove presentino come stato di superamento “true”. Per realizzare questa query, dunque, vengono effettuate due operazioni di join: la prima mette in relazione la tabella Esame con la tabella Prova, confrontando l’attributo idE presente in entrambe le tabelle; la seconda viene utilizzata per legare la tabella Appelli. Il risultato delle operazioni di giunzione viene filtrato verificando che idS della tabella Appelli coincida con idS della tabella Esame e che idP della tabella Appelli abbia lo stesso valore di idP di Prova. Si selezionano poi solamente gli esami che sono stati superati utilizzando “Appelli.stato\_superamento == True”.

Infine, vengono utilizzate le clausole GROUP BY e HAVING. La prima viene adoperata per eseguire l’operazione di raggruppamento sull’attributo idE della tabella Esame, mentre la seconda, come nella query descritta in precedenza, permette di selezionare solamente le prove che ammontano a 100 con la percentuale di superamento.

A picture containing text, font, screenshot

Description automatically generated

Principali scelte progettuali