DBMS MINI-PROJECT
ABSTRACT

TITLE: - **Alumni Management System For University.**

Group members:-

1)UTKARSH KADAM (202001103114)

2)SUMIT JAIN (202001103115)

3)SHIVRAJ DESHMUKH (202001103096)

4)MANTHAN WAHANE (202001103097)

Objectives :-

A) To build a system that will be able to manage alumni data of university and provide easy access to the alumni data.

B) To allow old and new students of a university to communicate with each other.

C) To reduce manual work.

Created for -

1) Schools
2) Colleges
3) University

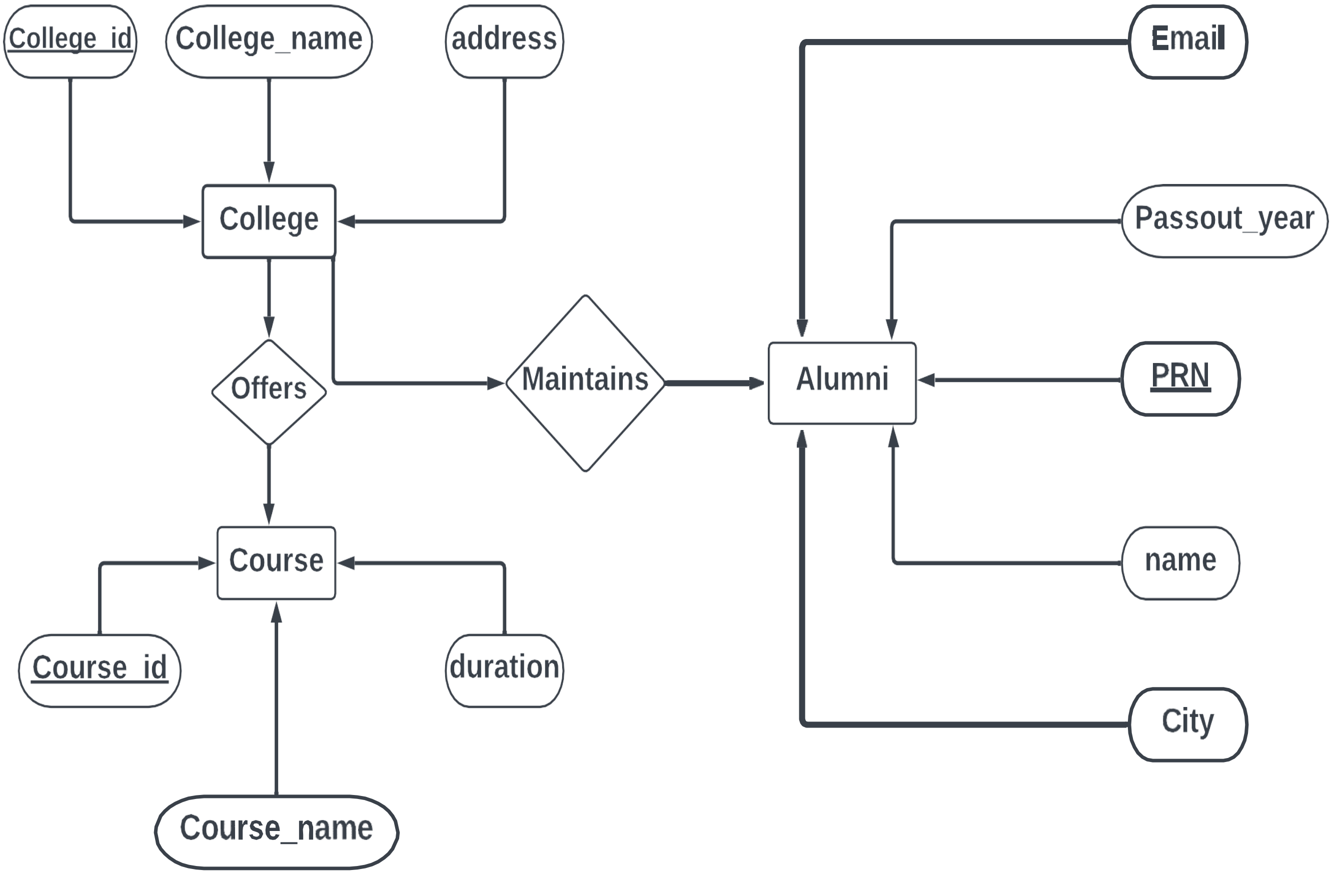
Functionality –

A) Once the student has graduated the records are inserted into database, thus making access to these records easy and less time consuming.

B) Search feature to find alumni in certain company,city,country .

C) Show if any alumni is hiring for vacant position in their company.

ER Diagram


College_id
College_name
address
College
Offers
Maintains
Course
Course_id
duration
Course_name
Email
Passout_year
Alumni
PRN
name
City

College

College_id	College_name	address

Primary Key :- College_id

Functional Dependencies
College_id -> College_name
College_id -> address

1NF

All values stored in table are single valued and atomic.
Hence table is in **1NF.**

2NF

The candiates keys are { College_id},
The set of key attributes are: { College_id }
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not all key attributes
checking FD: College_id --> College_name
Condition satisfied.
checking FD: College_id --> address
Condition satisfied.
Hence table is in **2NF.**

3NF

Find all cadnidate keys.
The candiates keys are { College_id}
The set of key attributes are: { College_id }
for each FD, check whether the LHS is superkey or the RHS are all key attributes
checking functional dependency College_id --> College_name
Condition satisfied.
checking functional dependency College_id --> address
Condition satisfied.
Hence table is in 3**NF.**

Course

Course_id	Course_name	duration	College_id

Primary Key :- course_id

Functional Dependencies
Course_id -> Course_name
Course_id -> duration
Course_id -> college_id

1NF
All values stored in table are single valued and atomic.
Hence table is in **1NF.**

2NF
The candidate key is {course_id},
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not all key attributes
checking FD: Course_id --> Course_name
Condition satisfied.
checking FD: Course_id --> duration
Condition satisfied.
checking FD: Course_id --> college_id
Condition satisfied.
Hence table is in 2**NF.**

3NF
The candidate key is {course_id},
The set of key attributes are: { course_id }
for each FD, check whether the LHS is superkey or the RHS are all key attributes
checking functional dependency Course_id --> Course_name
Condition satisfied.
checking functional dependency Course_id --> duration
Condition satisfied.
checking functional dependency Course_id --> college_id
Condition satisfied.
Hence table is in 3**NF.**

Alumni

PRN	name	College _id	Course _id	Passout _year	email	Current_ company	City

Primary Key :- PRN

Functional Dependencies
PRN -> name
PRN -> College_id
PRN -> Course_id
PRN -> Passout_year
PRN -> email
PRN -> Current_company
PRN -> City

1NF

All values stored in table are single valued and atomic.
Hence table is in **1NF.**

2NF

The candidates keys are { PRN},
The set of key attributes are: { PRN }
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not all key attributes
checking FD: PRN --> name
Condition satisfied.
checking FD: PRN --> College_id
Condition satisfied.
checking FD: PRN --> Course_id
Condition satisfied.
checking FD: PRN --> Passout_year
Condition satisfied.
checking FD: PRN --> email
Condition satisfied.
checking FD: PRN --> Current_company
Condition satisfied.
checking FD: PRN --> City
Condition satisfied.
Hence table is in **2NF.**

3NF

```
The candidates keys are { PRN},
The set of key attributes are: { PRN }
for each FD, check whether the LHS is superkey or the RHS are all key attributes
checking functional dependency PRN --> name
Condition satisfied.
checking functional dependency PRN --> College_id
Condition satisfied.
checking functional dependency PRN --> Course_id
Condition satisfied.
checking functional dependency PRN --> Passout_year
Condition satisfied.
checking functional dependency PRN --> email
Condition satisfied.
checking functional dependency PRN --> Current_company
Condition satisfied.
checking functional dependency PRN --> City
Condition satisfied.
Hence table is  in 3NF.
```