**Medical Student Database and Implementation**

Final Project | ISM 6216| Advanced Database Management |

Text

Description automatically generated

**Shivanvitha Narala** **U41460696**  shivanvitha@usf.edu

**Summary of Contents**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Title** | **Page No.** |
| 1. | Purpose | **3** |
| 2. | About the Dataset | **4** |
| 3. | Entities to be Tracked | **4** |
| 4. | Entities with Attributes Nested | **4** |
| 5. | Entity Relationship Documentation (ERD) | 6 |
| 6. | Table Views | **7** |
| 7. | Data Summary | **12** |
| 8. | Data Integrity | **12** |
| 9. | Queries | **15** |
| 10. | Stored Procedure | **22** |
| 11. | Performance Tuning | **23** |
| 13. | Data Visualization | **32-35** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Area** | **Description** | **Group Member** | **Weight** |
| **Database Design** | This step entails creating a logical database design that will be used in subsequent implementations. | Shivanvitha, rahul | 25% |
| **Query Writing** | This section provides you with another opportunity to write SQL queries, investigate transactions, and even conduct some database programming for stored procedures. | Shivanvitha, mysoora,taruni | 25% |
| **Performance Tuning** | You can use this section to build on and expand on your previous experiments with indexing, optimizer modes, partitioning, parallel execution, and any other approaches you choose to investigate further. | Taruni,Rahul,mysoora | 25% |
| **Data**  **Visualization** | You are free to investigate any other things that interest you.  DBA scripts, database security, interface design, data visualization, data mining, and NoSQL databases are among the suggestions. | Shivanvitha,mysoora | 25% |

# 

# Purpose

This article outlines the Advanced database management project's final project, which is a database of medical student information. The database integrity, data management, data production, and data loading have all been described using the entities and attributes involved in the database. Writing queries to obtain the data, as well as performance tweaking and other scripts, are all part of this project.

# About the Dataset

We're working on a medical student database with a variety of objects such as student

information, program information, task information, and student enrollments. We're putting

together a database for a student management system by connecting numerous attributes.

This project will be designed and implemented using a relational database. As part of the

project design, we created the Entity Relationship Diagram below, which displays entity

properties and relationships between entities. During the project's implementation, entities

will be treated as tables, attributes as columns, and tuples as rows.

**Entities Identified to be Tracked**

* Student(Person)
* StudentEnrollment
* Rotation
* Evaluator(Person)
* EvaluatorAssignment
* EvaluatorSpeciality
* Domain(Competency
* Competency(Competency)
* Task
* EPA(Competency)
* EPARotationMapping
* Track
* Program

**Entities with Attributes Nested**

* **Student(Person)**

1. StudentID
2. FirstName
3. LastName
4. Gender
5. CurrentYear
6. Phone
7. Email
8. MedicalTrack
9. TrackID
10. ProgramID

* **StudentEnrollment**

1. EnrolledID
2. RotationName
3. StartDate
4. EndDate
5. RotationID
6. StudentID

* **Rotation**

1. RotationID
2. Name
3. Description

* **Evaluator (Person)**

1. EvalId
2. FirstName
3. LastName
4. Gender
5. Role

* **EvaluatorAssignment**

1. AssignmnetID
2. RotationName
3. StartDate
4. EndDate
5. RoatationID
6. EvalID

* **EvaluatorSpeciality**

1. EvalId
2. Speciality

* **Domain (Competency)**

1. DomainId
2. Name
3. Description EPAId

* **Competency (Competency)**

1. CompetencyID
2. Name
3. Description DomainID

* **Task**

1. TaskId
2. DateRequested
3. DateCompleted
4. Results
5. Comment
6. Type
7. StudentID
8. EvalId
9. CompetecnyID

* **EPA(Competency)**

1. EPAId
2. Name
3. Speciality

* **EPARotationMapping**

1. MAGId
2. RotationId
3. EPAId

* **Track**

1. TrackId
2. Name
3. ProgramId
4. Description

* **Program**

1. ProgramId
2. Name
3. Description

**Entity Relationship Diagram representing Database Design**

Diagram

Description automatically generated

## **Table Views**

**Student (Person:)**

This table contains details of each student such as StudentID(primary key),Gender, CurrentYear, First Name, Last Name, Phone, Email, MedicalTrack,TrackID and ProgramID.

Graphical user interface, application, table

Description automatically generated

Competency(Competency): This table contains details such as CompetencyID(primary key), Name, Description and DomainID(Foreign Key).

Table

Description automatically generated

Domain(Competency): This table contains details such as DomainId(Primary Key), Name, Description and EPAId(Foreign Key).

Table

Description automatically generated

EPA(Competency): This table contains details such as EPAId(Primary Key), EPAName, Speciality and TrackID(Foreign Key).

Graphical user interface, table

Description automatically generated

EPARotationMapping: This table contains all details such as MAGId(Primary Key) and Foreign keys -RotationID, EPAId.

Table

Description automatically generated

Evaluator(Person): This table contains details such as EvaluatorID(Primary Key), FirstName, LastName, Street, City, State, Zip, Gender and Role.

Graphical user interface, text, application

Description automatically generated

EvaluatorAssignment: This table contains details such as AssignmnetID(Primary Key), RotationName, StartDate, EndDate, RoatationID(Foreign key) and EvaluatorID(Foreign Key).

Table

Description automatically generated

Rotation: This table contains details such as RotationID(Primary Key), Name and Description.

Table

Description automatically generated

StudentEnrollment: This table contains details such as EnrolledID(Primary key), RotationName, StartDate, EndDateand Foreign keys such as RotationID and StudentID.

Table

Description automatically generated

Task: This table contains details such as TaskId(Primary Key), DateRequested, DateCompleted, Results, Comment, Type and Foreign keys such as StudentID, EvalId and CompetecnyID.

Table

Description automatically generated

Track: This table contains details such as TrackId(Primary Key), Name, ProgramId(Foreign Key) and Description.

Table

Description automatically generated

## **Data Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Columns** | **# of constraints** | **# of Records** |
| Student(Person) | 11 | 3 | 200 |
| StudentEnrollment | 5 | 3 | 100 |
| Rotation | 3 | 1 | 9 |
| Evaluator(Person) | 9 | 4 | 150 |
| EvaluatorAssignment | 5 | 3 | 115 |
| EvaluatorSpeciality | 6 | 2 | 56 |
| Domain(Competency) | 4 | 2 | 55 |
| Competency(Competency) | 4 | 2 | 199 |
| Task | 9 | 6 | 379 |
| EPA(Competency) | 4 | 2 | 281 |
| EPARotationMapping | 3 | 3 | 100 |
| Track | 4 | 2 | 3 |
| Program | 4 | 1 | 47 |

## **Data Integrity**

In real life scenarios it is important to maintain the data integrity and there are certain constraints which we use to maintain integrity like Primary key usage, Foreign key usage, unique, check and not null parameters.

* PRIMARY KEY – unique value in a table which is used to identify a certain attribute such that no two tables coincide.

* FOREIGN KEY – This is the unique primary key of another table so that there is some to tables.

* UNIQUE – This is used to get rid of duplicate values.

* CHECK –This is used to check for a specific condition in a table.

* NOT NULL – This is used to prevent null values in a table.

## **Queries**

1. Fetch top 1000 student details from student table.

Graphical user interface, application, table

Description automatically generated

1. Fetch top 1000 enrollment details from student enrollment table.

Table

Description automatically generated

1. Fetch top 1000 Task details from task table.

Table

Description automatically generated

1. Fetch top 1000 track details from track table.

Table

Description automatically generated

1. Fetch top 1000 rotation details from Rotation table.

Table

Description automatically generated

1. Fetch top 1000 Assignment details from EvaluatorAssignment table.

Table

Description automatically generated

1. Fetch top 1000 Evaluator details from Evaluator table.

Graphical user interface, table

Description automatically generated

1. Fetch top 1000 Mapping details from EPARotationMapping table.

Table

Description automatically generated with medium confidence

1. Fetch top 1000 EPA details from EPA table.

Table

Description automatically generated

1. Fetch top 1000 Domain details from Domain table.

Graphical user interface

Description automatically generated with low confidence

11) Fetch top 1000 Competency details from Competency table.

Graphical user interface, table

Description automatically generated

## **STORED PROCEDURE**

USE [FeedForward\_SSA]

GO

GO

CREATE PROCEDURE dbo.GetStudentsDetails

@City nvarchar(100)

AS

SET NOCOUNT ON;

SELECT \* from Person.Student where City like '%'+@City+'%'

GO

DECLARE @return\_value int

EXEC @return\_value = [dbo].[GetStudentsDetails]

@City = N'e'

GO

Table

Description automatically generated

## **PERFORMANCE TUNING**

1. Change Tracker Code Execution:

Graphical user interface, text, application, email

Description automatically generated

1. Version after Insertion:

Execution:

Graphical user interface, text, application

Description automatically generated

1. Version after deletion:

Execution:

Graphical user interface, text, application, email

Description automatically generated

1. Version after Updation:

Execution:

Table

Description automatically generated with medium confidence

1. Checking the update statement:

Execution:

Graphical user interface, text

Description automatically generated

## **PARALLELISM**

1. Setting Max Degree of Parallelism to 4 and checking if it reflected: Execution:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Comparing with and without MAXDOP:

Execution:

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

1. With and witho
2. ut using hint, comparing the execution time taken:

Execution:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

## **Data Visualization**

1)The below pie chart shows the students with pass/fail distribution.

Chart, pie chart

Description automatically generated

2)The below chart shows studenrts who passed/failed based on gender.

Graphical user interface, text, application, Word

Description automatically generated

3) The below chart shows number of students passed/failed with reference to trackId based on gender.

Chart, bar chart

Description automatically generated