1. **GIỚI THIỆU CHUNG**
2. **Tổng quan đề tài**
   1. **Giới thiệu về đề tài**

* Quản lý tuyển sinh là một phần quan trọng trong quá trình hoạt động của một trường đại học. Nó bao gồm việc thu thập và xử lý thông tin từ các ứng viên, tìm kiếm và thu hút các ứng viên tiềm năng, và xác định các tiêu chí để lựa chọn những ứng viên phù hợp với các chương trình học của trường.
* Trong quá trình tuyển sinh, việc quản lý thông tin và dữ liệu là rất quan trọng. Các bộ phận tuyển sinh phải đảm bảo rằng thông tin của ứng viên được thu thập và xử lý một cách chính xác và an toàn. Ngoài ra, họ cũng cần phải thường xuyên cập nhật và bảo trì các cơ sở dữ liệu để đảm bảo rằng thông tin được lưu trữ một cách hiệu quả và dễ dàng truy cập.
* Để thu hút và giữ chân các ứng viên tiềm năng, các bộ phận tuyển sinh cần phải có các chiến lược quảng cáo và marketing hiệu quả. Họ cần phải nghiên cứu và hiểu rõ đối tượng khách hàng của trường, từ đó thiết kế các chiến dịch quảng cáo và marketing phù hợp với nhu cầu của các ứng viên tiềm năng.
* Cuối cùng, để đảm bảo rằng chỉ những ứng viên phù hợp với các chương trình học của trường được nhận vào, các bộ phận tuyển sinh cần phải có các tiêu chí lựa chọn rõ ràng và công bằng. Họ cần phải đảm bảo rằng việc lựa chọn ứng viên dựa trên năng lực, tài năng và tiềm năng, không phân biệt chủng tộc, giới tính, tôn giáo hay bất kỳ yếu tố nào khác.
* Tóm lại, quản lý tuyển sinh là một hoạt động quan trọng trong hoạt động của một trường đại học. Để thành công trong việc này, các bộ phận tuyển sinh cần phải đảm bảo rằng thông tin của ứng viên được quản lý và bảo mật tốt, sử dụng các chiến lược quảng cáo và marketing hiệu quả, và áp dụng các tiêu chí lựa chọn công bằng và rõ ràng.
  1. **Mục đích và phạm vi của đề tài**
* Mục đích của đồ án thiết kế cơ sở dữ liệu quản lý tuyển sinh là tạo ra một hệ thống quản lý dữ liệu tuyển sinh toàn diện và hiệu quả. Cơ sở dữ liệu sẽ bao gồm một số thành phần chính, bao gồm tài khoản nhân viên, thông tin học sinh, chi tiết học sinh, thông tin thanh toán, yêu cầu của học sinh và thông tin năm học. Phạm vi của dự án là phát triển một cơ sở dữ liệu đáng tin cậy và thân thiện với người dùng, có thể quản lý và sắp xếp dữ liệu tuyển sinh một cách hiệu quả. Tài khoản nhân viên sẽ cho phép nhân viên được ủy quyền truy cập và sửa đổi cơ sở dữ liệu. Phần thông tin sinh viên sẽ bao gồm thông tin cá nhân cơ bản, chẳng hạn như tên và chi tiết liên lạc, trong khi phần chi tiết sinh viên sẽ bao gồm thông tin chi tiết hơn, chẳng hạn như thông tin liên lạc của người thân .Phần thanh toán sẽ theo dõi học phí và các chi phí khác, trong khi phần yêu cầu sẽ theo dõi các tài liệu và biểu mẫu cần thiết. Cuối cùng, phần năm học sẽ cho phép dễ dàng theo dõi và quản lý học sinh trong nhiều năm học. Nhìn chung, mục tiêu của dự án này là cung cấp một hệ thống tập trung và hợp lý để quản lý dữ liệu tuyển sinh của học sinh.
  1. **Đối tượng sử dụng phần mềm**
* Đối tượng sử dụng phần mềm là các nhân viên quản lí tuyển sinh
  1. **Các chức năng chính của phần mềm:**
* Quản lý thông tin học sinh như :điểm số, thông tin liên lạc,các thông tin liên quan.
* Quản lý chi phí riêng và chung của học sinh khi tuyển sinh.

1. **Phân tích yêu cầu**
   1. **Phân tích hiện trạng:**

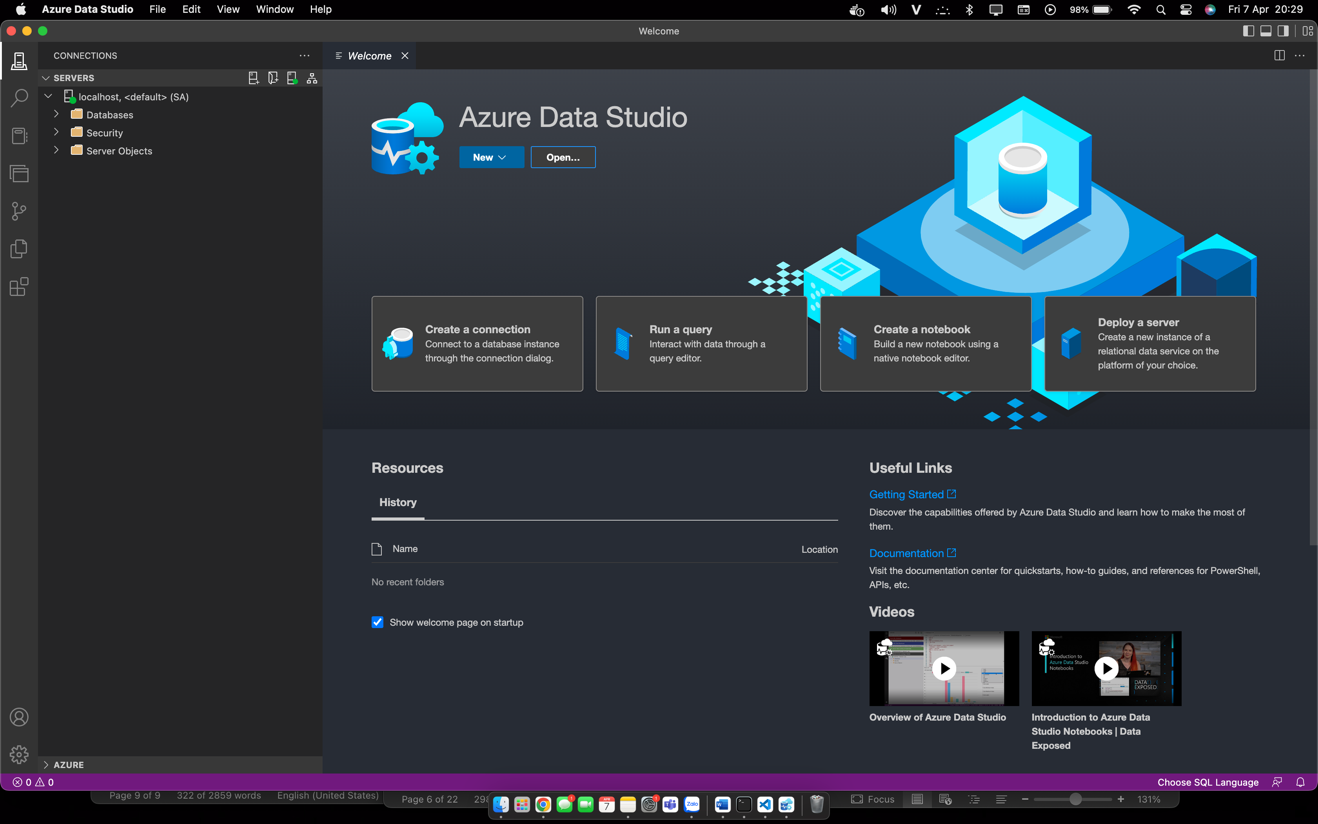
* Quản lý tuyển sinh là một phần quan trọng trong quá trình hoạt động của một trường đại học. Nó bao gồm việc thu thập và xử lý thông tin từ các ứng viên, tìm kiếm và thu hút các ứng viên tiềm năng, và xác định các tiêu chí để lựa chọn những ứng viên phù hợp với các chương trình học của trường.
* Trong quá trình tuyển sinh, việc quản lý thông tin và dữ liệu là rất quan trọng. Các bộ phận tuyển sinh phải đảm bảo rằng thông tin của ứng viên được thu thập và xử lý một cách chính xác và an toàn. Ngoài ra, họ cũng cần phải thường xuyên cập nhật và bảo trì các cơ sở dữ liệu để đảm bảo rằng thông tin được lưu trữ một cách hiệu quả và dễ dàng truy cập.
* Để thu hút và giữ chân các ứng viên tiềm năng, các bộ phận tuyển sinh cần phải có các chiến lược quảng cáo và marketing hiệu quả. Họ cần phải nghiên cứu và hiểu rõ đối tượng khách hàng của trường, từ đó thiết kế các chiến dịch quảng cáo và marketing phù hợp với nhu cầu của các ứng viên tiềm năng.
* Cuối cùng, để đảm bảo rằng chỉ những ứng viên phù hợp với các chương trình học của trường được nhận vào, các bộ phận tuyển sinh cần phải có các tiêu chí lựa chọn rõ ràng và công bằng. Họ cần phải đảm bảo rằng việc lựa chọn ứng viên dựa trên năng lực, tài năng và tiềm năng, không phân biệt chủng tộc, giới tính, tôn giáo hay bất kỳ yếu tố nào khác.
* Tóm lại, quản lý tuyển sinh là một hoạt động quan trọng trong hoạt động của một trường đại học. Để thành công trong việc này, các bộ phận tuyển sinh cần phải đảm bảo rằng thông tin của ứng viên được quản lý và bảo mật tốt, sử dụng các chiến lược quảng cáo và marketing hiệu quả, và áp dụng các tiêu chí lựa chọn công bằng và rõ ràng.
  1. **Thư viện và công cụ được sử dụng trong quá trình xây dựng**
* Trong quá trình xây dụng cơ sở dữ liệu về quản lý tuyển sinh, chương trình quản lý được sử dụng bởi ngôn ngữ SQL.
* SQL hay Structured Query Language, là ngôn ngữ truy vấn được sử dụng để tương tác với các hệ thống quản lý cơ sở dữ liệu quan hệ. SQL được phát triển vào những năm 1970 bởi các nhà khoa học máy tính Donald Chamberlin và Raymond Boyce tại IBM. Từ đó đến nay, SQL đã trở thành một trong những ngôn ngữ quản lý cơ sở dữ liệu phổ biến nhất trên thế giới. SQL cho phép người dùng truy vấn, cập nhật, xóa, và thêm dữ liệu vào cơ sở dữ liệu quan hệ. Nó cũng cung cấp các tính năng cho phép tạo bảng, chỉnh sửa cấu trúc cơ sở dữ liệu, và thực hiện các tác vụ phức tạp như kết hợp dữ liệu từ nhiều bảng. Với SQL, người dùng có thể truy vấn và xử lý dữ liệu một cách nhanh chóng và hiệu quả, đồng thời giảm thiểu khối lượng công việc thủ công trong việc quản lý cơ sở dữ liệu.
* Về phần IDE để dùng khởi tạo cơ sở dữ liệu và quản lý cơ sở dữ liệu:
* ***Azure Data Studio:*** là một công cụ quản lý và phát triển cơ sở dữ liệu đa nền tảng miễn phí từ Microsoft. Nó được thiết kế để làm việc với các cơ sở dữ liệu SQL Server, cơ sở dữ liệu Azure SQL và các hệ thống cơ sở dữ liệu khác, bao gồm cả PostgreSQL và MySQL.

Azure Data Studio có nhiều chức năng quan trọng trong việc quản lý cơ sở dữ liệu, bao gồm:

* Quản lý đối tượng cơ sở dữ liệu: Azure Data Studio cung cấp các công cụ để quản lý các đối tượng cơ sở dữ liệu, bao gồm các bảng, chế độ xem, thủ tục lưu trữ, hàm và quan hệ.
* Cập nhật cơ sở dữ liệu: Azure Data Studio cho phép người dùng cập nhật dữ liệu và cấu trúc cơ sở dữ liệu bằng cách sử dụng các truy vấn SQL và các công cụ khác.
* Kiểm tra hiệu suất: Azure Data Studio cho phép người dùng xem các câu truy vấn và phân tích hiệu suất của chúng, giúp tìm ra các vấn đề về hiệu suất và tối ưu hóa truy vấn để cải thiện hiệu suất cơ sở dữ liệu.
* Xây dựng truy vấn: Azure Data Studio cung cấp các công cụ để xây dựng truy vấn SQL và tối ưu hóa chúng để cải thiện hiệu suất và tăng tính khả dụng của cơ sở dữ liệu.
* Tích hợp: Azure Data Studio tích hợp tốt với các công cụ khác của Microsoft, bao gồm Azure Cloud Shell và Visual Studio Code, giúp người dùng quản lý cơ sở dữ liệu một cách dễ dàng và hiệu quả hơn.

Với các chức năng quan trọng này, Azure Data Studio là một công cụ quan trọng và hữu ích cho bất kỳ ai làm việc trong lĩnh vực quản lý và phát triển cơ sở dữ liệu.

-Hình ảnh trực quan của tool:



1. **Thiết kế hệ thống:**
   1. **Các thành phần chính của cơ sở dữ liệu quản lý tuyển sinh:**

* Một phòng ban quản lý tuyển sinh bao gồm những bảng chính sau:
  + Student
  + Student Details
  + Student Payments
  + Student Requirements
  + Student School Year
  + Employees Accounts
* Những thành phần phụ được phân bổ trong các bảng tương ứng như sau:
  + Student: Thông tin chính của học sinh
    - Student\_ID: Mã ID chính của học sinh
    - First\_Middle\_Name: Tên đệm của học sinh
    - Last\_Name: Tên lót của học sinh
    - Age: Tuổi của học sinh
    - Gender: Giới tính của học sinh
    - Address: Địa chỉ thương trú của học sinh
    - Date\_Of\_Birth: Ngày tháng năm sinh của học sinh
    - Student\_Phone\_Numbers: Số điện thoại của học sinh
  + Student\_Details: Thông tin chi tiết của học sinh
    - Details\_ID: Mã ID của thông tin chi tiết của học sinh
    - Student\_ID: Mã ID chính của học sinh
    - Father\_name: Họ và tên bố của học sinh
    - Father\_work: Công việc hiện tại của bố học sinh
    - Mother\_name: Họ và tên mẹ của học sinh
    - Mother\_work: Công việc hiện tại của mẹ học sinh
    - Parents\_address: Địa chỉ của gia đình học sinh
    - Parents\_phonenumbers: Số điện thoại của gia đình học sinh
    - Regular\_contact\_address: Địa chỉ thường xuyên liên lạc
    - Regular\_contact\_phone: Số điện thoại thường xuyên liên lạc
  + Student\_Payments: Thông tin chi tiết giao dịch của học sinh
    - Payment\_ID: Mã ID giao dịch của học sinh
    - Student\_ID: Mã ID chính của học sinh
    - Name\_Payables: Tên của giao dịch
    - Amount\_Paid: Số lượng tiền giao dịch
    - Balance\_Return: Số tiền thừa trả lại cho học sinh
    - Payment\_Date\_and\_time: Ngày và giờ giao dịch
  + Student\_Requirements: Tiêu chí đầu vào của học sinh
    - Requirements\_ID: Mã ID của tiêu chí đầu vào
    - Student\_ID: Mã ID chính của học sinh
    - Entrance\_Exam\_Score: Điểm thi đầu vào của học sinh
    - Priority\_Point: Điểm ưu tiên của học sinh
  + Student\_School\_Year:
    - Student\_ID: Mã ID chính của học sinh
    - Student\_Year\_Entry: Năm nhập học của học sinh
    - Semester: Kì nhập học của học sinh
  + Employees\_Accounts:
    - Employees\_ID: Mã ID của nhân viên quản lý
    - Email: Email của nhân viên quản lý
    - First\_Middle\_Name: Họ và tên đệm của nhân viên
    - Last\_Name: Tên lót của nhân viên
  1. Lược đồ ER:

**Diagram

Description automatically generated**

* Ở đây chúng ta có thể thấy Student là table trung tâm, dữ liệu chính của cơ sở dữ liệu quản lý tuyển sinh
* Từ bảng Student chúng ta có thể thấy rõ các mối quan hệ của các bảng còn lại, cụ thể như sau:
  + Student quan hệ 1 – n với Student\_Payments
  + Student quan hệ 1 – 1 với Student\_Details
  + Student quan hệ 1 – n với Student\_Requirements
  + Student quan hệ n – 1 với Student\_School\_Year
  + Student quan hệ n – 1 với Employees\_Account
  + Employees\_Account quan hệ 1 – n với Student\_Payment

1. Triển khai và kiểm thử phần mềm:

* Việc triển khai và kiểm thử phần mềm quản lý tuyển sinh là một công việc rất quan trọng trong quá trình xây dựng một hệ thống quản lý tuyển sinh chuyên nghiệp. Cơ sở dữ liệu quản lý tuyển sinh là một cơ sở dữ liệu được thiết kế để quản lý các hoạt động tuyển sinh của trường đại học, bao gồm việc quản lý thông tin đăng ký, thông tin hồ sơ ứng viên, xếp hạng và chọn lọc các ứng viên phù hợp với yêu cầu của trường. Quá trình triển khai và kiểm thử phần mềm quản lý tuyển sinh bao gồm nhiều bước, bao gồm cài đặt, cấu hình, kiểm tra tính năng, đảm bảo tính ổn định, và đảm bảo tính bảo mật của phần mềm. Những bước này cần được thực hiện một cách cẩn thận và có kế hoạch để đảm bảo rằng cơ sở dữ liệu được triển khai và kiểm tra một cách hiệu quả và đạt được các yêu cầu của trường.

1. **Hướng dẫn sử dụng****, cách cài đặt phần mềm:**

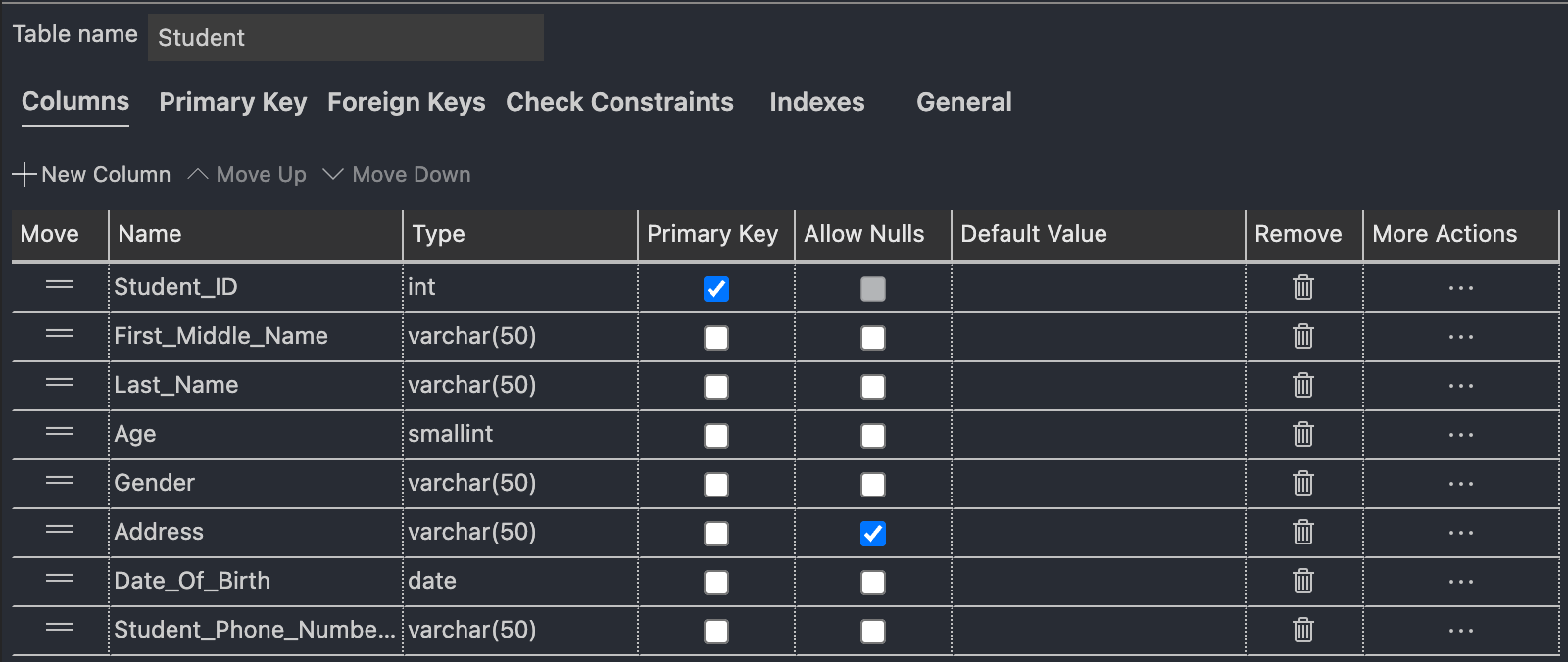
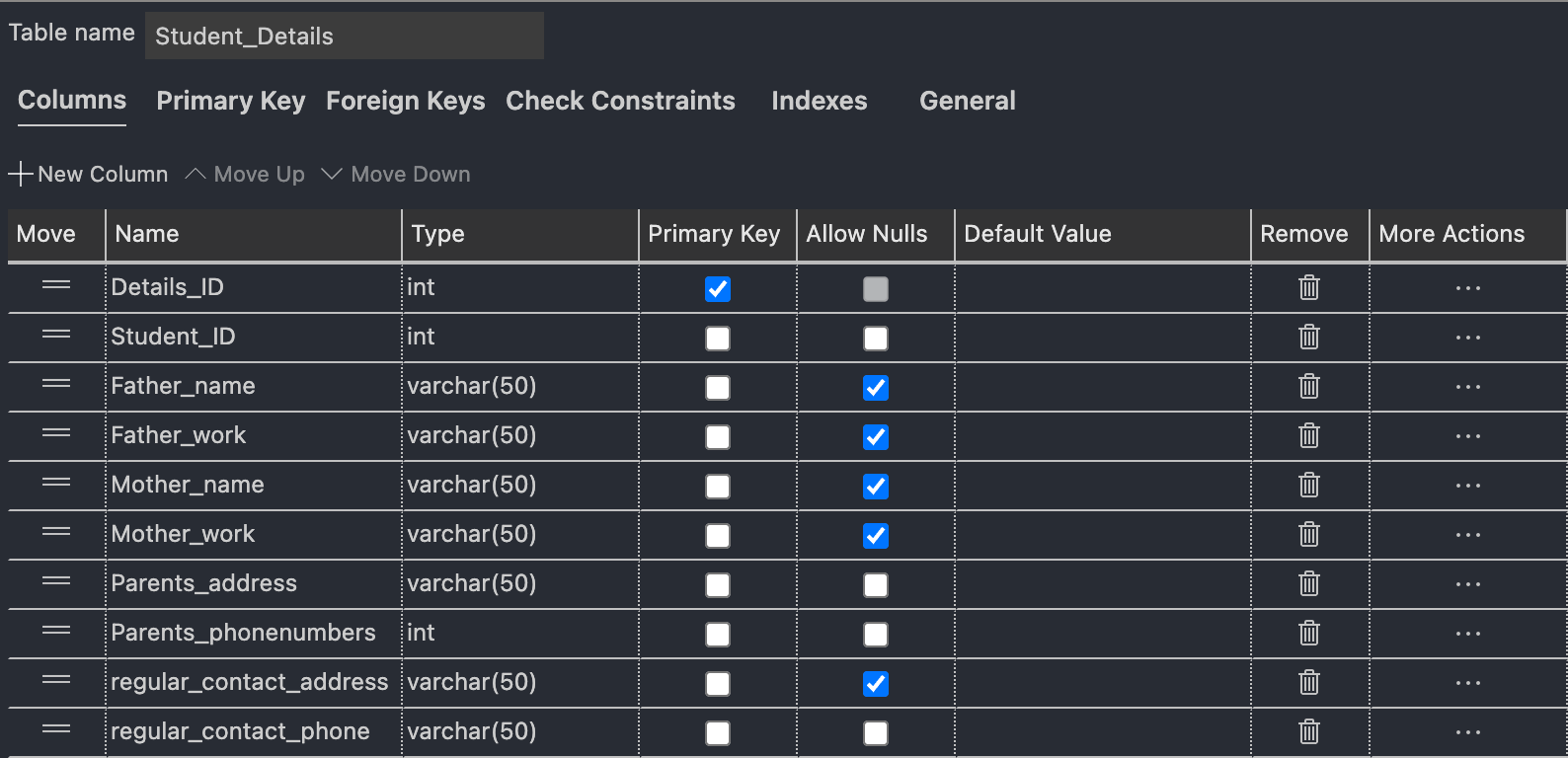
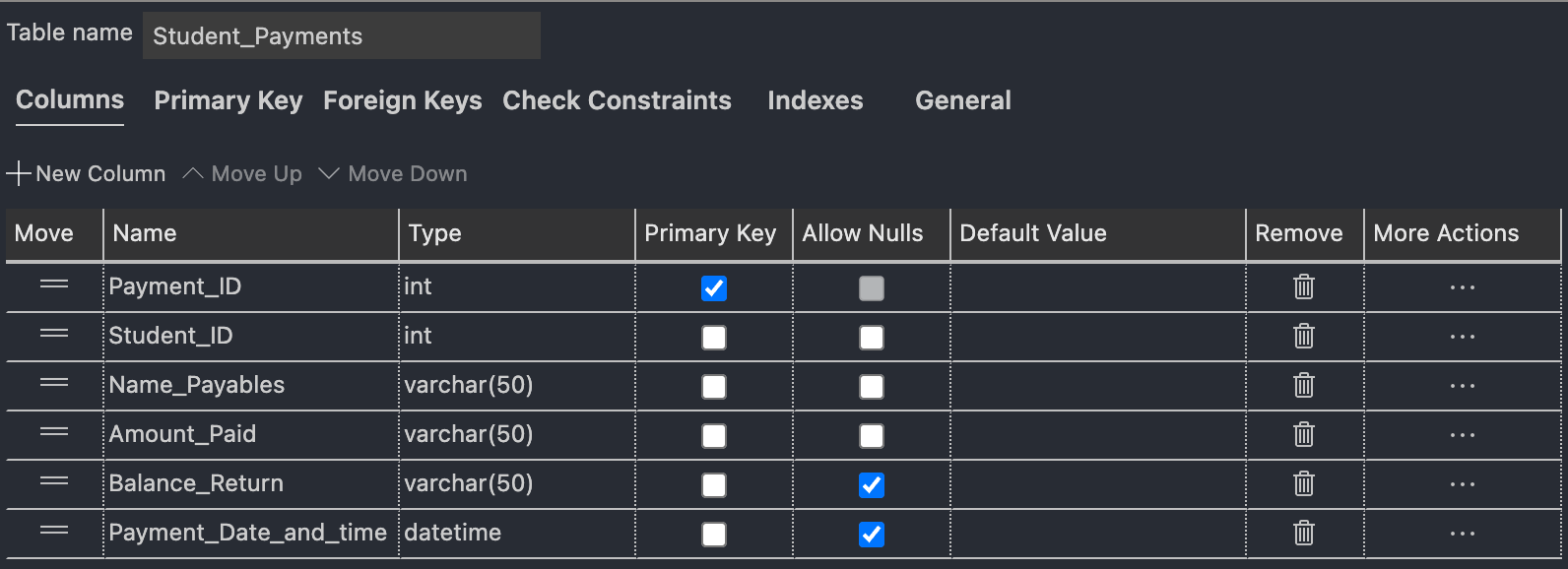
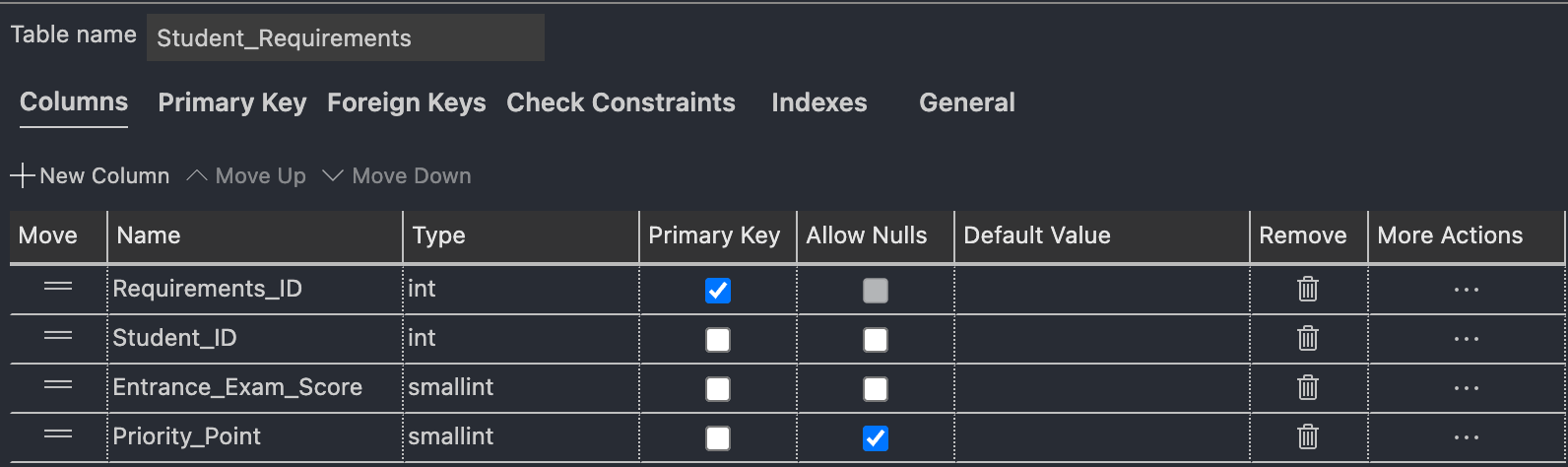
Để cài đặt và chạy thử, quản lý cơ sở dữ liệu quản lý tuyển sinh ta có thể thực hiện các bước sau:

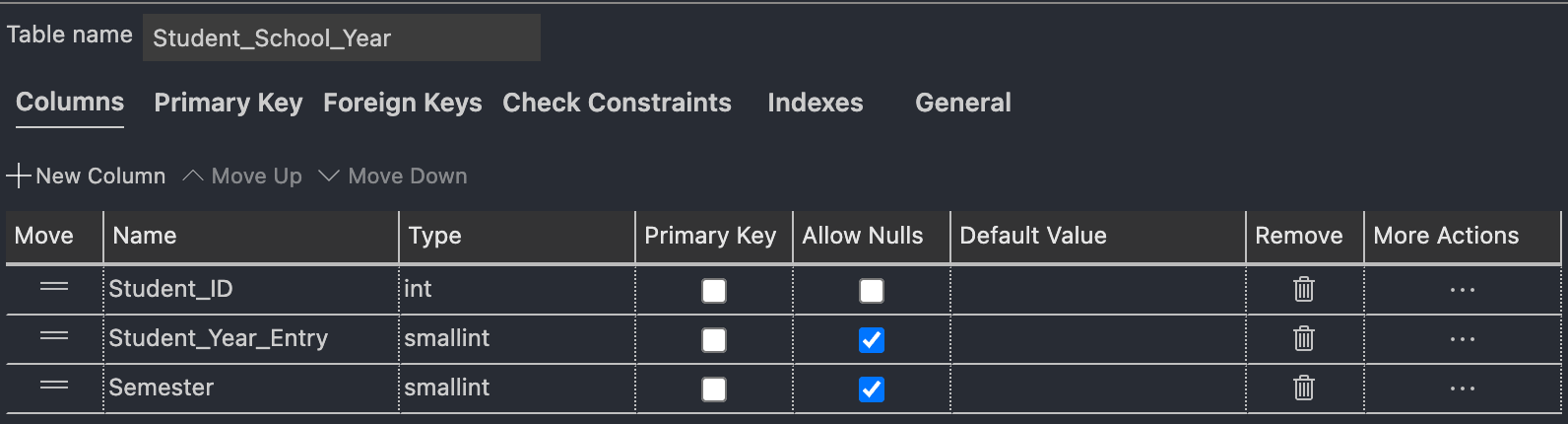
* Cài chương trình quản lý cơ sở dữ liệu như: Microsoft sql server,Azure Data Studio,…
* Cập nhật các thông tin liên quan của cơ sở dữ liệu
* Truy vấn cơ sở dữ liệu để tìm thông tin

1. **Kết luận**
   1. **Tóm tắt công việc làm được:**

* Thiết kế và triển khai được một cơ sở dữ liệu quản lý tuyển sinh bằng ngôn ngữ SQL.
* Trong cơ sở dữ liệu có rõ các thông tin cần thiết cho một đợt tuyển sinh cũng như các thông tin chi tiết của học sinh
  1. **Đánh giá chất lượng sản phẩm và hiệu quả của chương trình**
* Cơ sở dữ liệu đáp ứng đủ các như cầu như thông tin học sinh, thông tin liên lạc của học sinh cũng như các thông tin liên quan khác đối với việc quản lý tuyển sinh

1. **CƠ SỞ DỮ LIỆU QUẢN LÝ TUYỂN SINH:**
2. **Đoạn code chương trình:**

* Code cũng như khoá chính liên quan đến table Student:
* CREATE TABLE [dbo].[Student] (
* [Student\_ID] INT NOT NULL,
* [First\_Middle\_Name] VARCHAR (50) NOT NULL,
* [Last\_Name] VARCHAR (50) NOT NULL,
* [Age] SMALLINT NOT NULL,
* [Gender] VARCHAR (50) NOT NULL,
* [Address] VARCHAR (50) NULL,
* [Date\_Of\_Birth] DATE NOT NULL,
* [Student\_Phone\_Numbers] VARCHAR (50) NOT NULL,
* CONSTRAINT [PK\_Student] PRIMARY KEY CLUSTERED ([Student\_ID] ASC)
* );
* Code cũng như khoá chính, khoá phụ liên quan đến table Student\_Details:
* CREATE TABLE [dbo].[Student\_Details] (
* [Details\_ID] INT NOT NULL,
* [Student\_ID] INT NOT NULL,
* [Father\_name] VARCHAR (50) NULL,
* [Father\_work] VARCHAR (50) NULL,
* [Mother\_name] VARCHAR (50) NULL,
* [Mother\_work] VARCHAR (50) NULL,
* [Parents\_address] VARCHAR (50) NOT NULL,
* [Parents\_phonenumbers] INT NOT NULL,
* [regular\_contact\_address] VARCHAR (50) NULL,
* [regular\_contact\_phone] VARCHAR (50) NOT NULL,
* CONSTRAINT [PK\_Student\_Details] PRIMARY KEY CLUSTERED ([Details\_ID] ASC),
* CONSTRAINT [Student\_ID] FOREIGN KEY ([Student\_ID]) REFERENCES [dbo].[Student] ([Student\_ID])
* );
* Code cũng như khoá chính, khoá phụ liên quan đến table Student\_Payments:
* CREATE TABLE [dbo].[Student\_Payments] (
* [Payment\_ID] INT NOT NULL,
* [Student\_ID] INT NOT NULL,
* [Name\_Payables] VARCHAR (50) NOT NULL,
* [Amount\_Paid] VARCHAR (50) NOT NULL,
* [Balance\_Return] VARCHAR (50) NULL,
* [Payment\_Date\_and\_time] DATETIME NULL,
* CONSTRAINT [PK\_Student\_Payments] PRIMARY KEY CLUSTERED ([Payment\_ID] ASC),
* CONSTRAINT [FK\_Student\_Payments\_Student] FOREIGN KEY ([Student\_ID]) REFERENCES [dbo].[Student] ([Student\_ID])
* );
* Code cũng như khoá chính, khoá phụ liên quan đến table Student\_Requirements:
* CREATE TABLE [dbo].[Student\_Requirements] (
* [Requirements\_ID] INT NOT NULL,
* [Student\_ID] INT NOT NULL,
* [Entrance\_Exam\_Score] SMALLINT NOT NULL,
* [Priority\_Point] SMALLINT NULL,
* CONSTRAINT [PK\_Student\_Requirements] PRIMARY KEY CLUSTERED ([Requirements\_ID] ASC),
* CONSTRAINT [FK1] FOREIGN KEY ([Student\_ID]) REFERENCES [dbo].[Student] ([Student\_ID])
* );
* Code cũng như các khoá chính, khoá phụ liên quan đến table Student\_School\_Year:
* CREATE TABLE [dbo].[Student\_School\_Year] (
* [Student\_ID] INT NOT NULL,
* [Student\_Year\_Entry] SMALLINT NULL,
* [Semester] SMALLINT NULL,
* CONSTRAINT [FK2] FOREIGN KEY ([Student\_ID]) REFERENCES [dbo].[Student] ([Student\_ID])
* );
* Code cũng như các khoá chính, khoá phụ liên quan đến table Employees\_Accounts:
* CREATE TABLE [dbo].[Employees\_Accounts] (
* [Employee\_ID] INT NOT NULL,
* [Email] VARCHAR (50) NOT NULL,
* [FIrst\_Middle\_Name] VARCHAR (50) NULL,
* [Last\_Name] VARCHAR (50) NULL,
* CONSTRAINT [PK\_Employees\_Accounts] PRIMARY KEY CLUSTERED ([Employee\_ID] ASC)
* );
* Từ đó hình thành các table tương ứng như sau:
* ****
* ****
* ****
* ****
* **Table

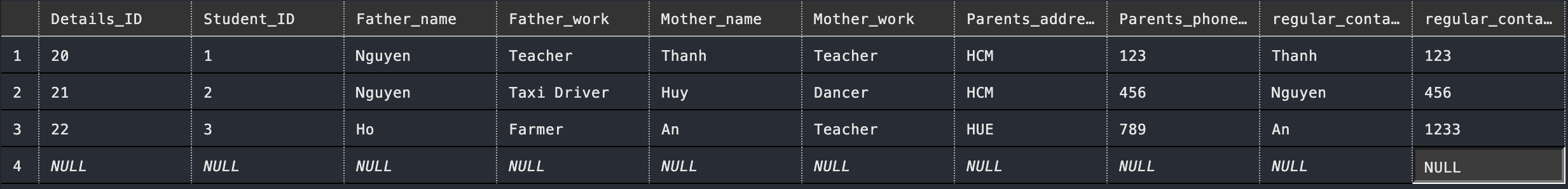
  Description automatically generated with medium confidence**
* ****

1. **Thêm dữ liệu và thao tác với cơ sở dữ liệu:**

* Ta tiến hành thêm dữ liệu của từng bảng như sau:
* Student:

Graphical user interface

Description automatically generated

* Student\_Details
* Student\_PaymentsTable

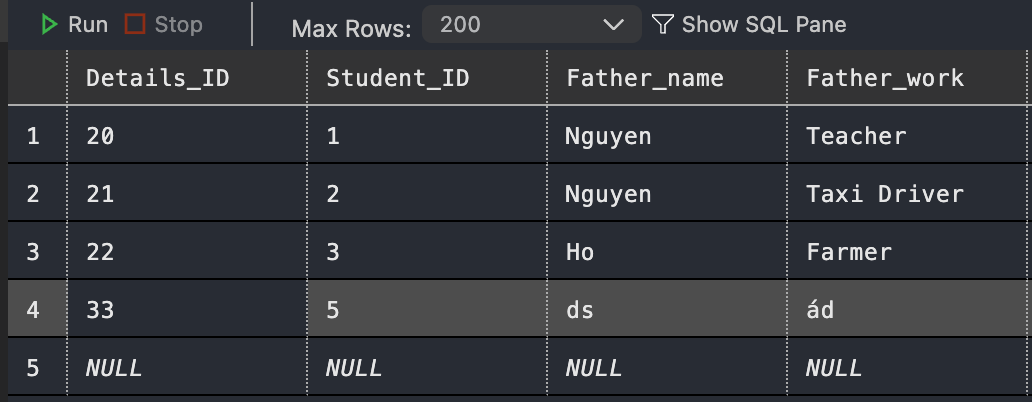
  Description automatically generated with medium confidence
* Student\_RequirementsTable

  Description automatically generated
* Student\_School\_YearTable

  Description automatically generated
* Employees\_Accounts

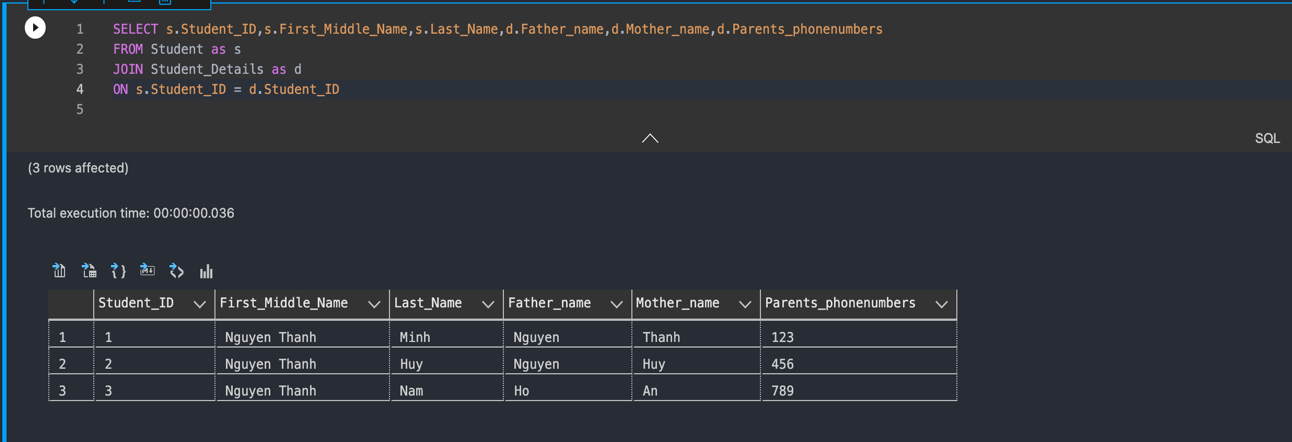
Table

Description automatically generated

* Vì student là khoá phụ của các table còn lại ( trừ Table :Employees\_Accounts), nên mối liên quan rất chặt chẽ đến thành phân Student\_ID
* Thử insert data bất kì không phải trong table Student, cơ sở dữ liệu liền báo lỗi ngay:
* Lỗi xung đột giữa khoá ngoại nên chương trình không cho insert thêm data

Text

Description automatically generated

* Thực thi các câu lệnh đối với cơ sở dữ liệu:
* 
* Giải thích:

Mã này là một truy vấn SQL để lấy thông tin sinh viên và chi tiết sinh viên từ hai bảng khác nhau: "Student" và "Student\_Details".

* Câu lệnh SELECT được sử dụng để chọn các cột cần lấy dữ liệu từ bảng.
* s.Student\_ID, s.First\_Middle\_Name và s.Last\_Name là các cột được chọn từ bảng "Student".
* d.Father\_name, d.Mother\_name và d.Parents\_phonenumbers là các cột được chọn từ bảng "Student\_Details".
* FROM được sử dụng để chỉ định bảng chứa dữ liệu cần được truy xuất.
* JOIN được sử dụng để kết hợp dữ liệu từ hai bảng trên cùng một dòng dữ liệu, bằng cách sử dụng cột Student\_ID làm điểm kết nối (join key) giữa hai bảng.
* ON được sử dụng để chỉ định điều kiện kết hợp dữ liệu giữa hai bảng, trong trường hợp này là s.Student\_ID = d.Student\_ID.
* Vì cả hai bảng đều chứa cột Student\_ID, chúng ta sử dụng tên bảng để định rõ cột nào được sử dụng trong điều kiện JOIN.

Kết quả truy vấn này sẽ trả về các cột: Student\_ID, First\_Middle\_Name, Last\_Name, Father\_name, Mother\_name và Parents\_phonenumbers, và sẽ hiển thị thông tin chi tiết của sinh viên từ bảng "Student\_Details" khi nó khớp với bản ghi tương ứng trong bảng "Student".