Project-2 DBMS
Title: Recruitment Hospital

Name: Atharvaa Rane
SUID: 275976322

Table of Contents

ABSTRACT:

Background: In the dynamic environment of healthcare, the success of medical facilities hinges on the ability and readiness of their staff. Efficient and empathetic recruitment processes are vital to attract and maintain a workforce capable of delivering compassionate and high-quality patient care. The recruitment challenges in healthcare, characterized by the need for diverse roles ranging from administrative to clinical positions, demand an advanced and adaptable system to manage hiring intricacies.

Objective: This project seeks to develop a tailored database system for a University Medical Center, designed to nurture and enhance the recruitment journey. By focusing on a system that supports human resource professionals in nurturing potential employees from first contact through to hiring, we aim to foster a workplace that values thorough selection processes and thoughtful candidate engagement.

Methods: Employing SQL Server, we crafted a relational database to model the intricate web of interactions typical in healthcare recruitment. This database not only houses data but also mirrors the real-world processes of job applications, candidate evaluations, interviews, and final onboarding. Our approach emphasizes data integrity and user accessibility while ensuring sensitive information is safeguarded through robust security practices. Key elements such as stored procedures and triggers automate routine tasks and uphold data consistency, supporting staff in focusing more on candidate engagement and less on administrative burdens.

Implementation: Initial data populating the system reflects realistic recruitment scenarios, providing a foundation for rigorous testing and adjustments. The system is equipped with tools for generating insightful analytics, such as reports on time-to-hire metrics, effectiveness of recruitment channels, and interviewer performance, all aimed at refining the recruitment strategy continually.

Expected Outcomes: Anticipated benefits include a significant reduction in time and resources spent on recruitment activities, heightened transparency in recruitment statuses, and enriched decision-making through detailed analytical reports. By improving these aspects, the system will contribute to a more streamlined and candidate-focused recruitment process, enhancing staff satisfaction and retention rates

OVERVIEW OF THE PROJECT DESIGN AND IMPLEMENTATION:

- The project establishes a comprehensive recruitment database for a University Medical Center to streamline the hiring process.
- A relational database is set up in SQL Server, structured to facilitate detailed tracking of job applications and recruitment processes.
- The database includes key entities such as Candidates, Jobs, JobOpenings, Applications, Interviews, and related entities.
- Candidates table stores personal and contact details of applicants, facilitating a structured repository for candidate data.
- Jobs table records available positions, along with job descriptions, types, and requirements.
- JobOpenings links to Jobs, indicating the number of positions open for each job role.
- Applications table connects candidates to job openings, tracking the status and progress of each application.
- Interviews and Tests tables manage the scheduling and outcomes of interviews and assessments linked to specific applications.
- Interviewers, Documents, BackgroundChecks, DrugTests, and Reimbursements tables provide additional support data for the recruitment process.
- Comprehensive foreign key relationships ensure data integrity and enforce logical connections between related data entities.
- Stored procedures automate routine database operations such as adding new candidates or updating application statuses.
- Triggers are employed to maintain data consistency, such as updating job opening counts or logging interview details.
- Views are created for easily querying common data sets, such as current job openings or candidate status reports.
- Security roles and permissions are carefully implemented to safeguard sensitive data and control access based on user roles.
- The database design emphasizes scalability and efficiency, enabling the HR department to efficiently manage recruitment activities and provide timely reports for decision-making.
- This summary encapsulates the strategic approach to database management, emphasizing operational efficiency, data integrity, and system security within the recruitment context.

ER Diagram

Entities and Their Attributes
Here's a breakdown of the entities (tables) included in the ER diagram, along with their primary attributes:

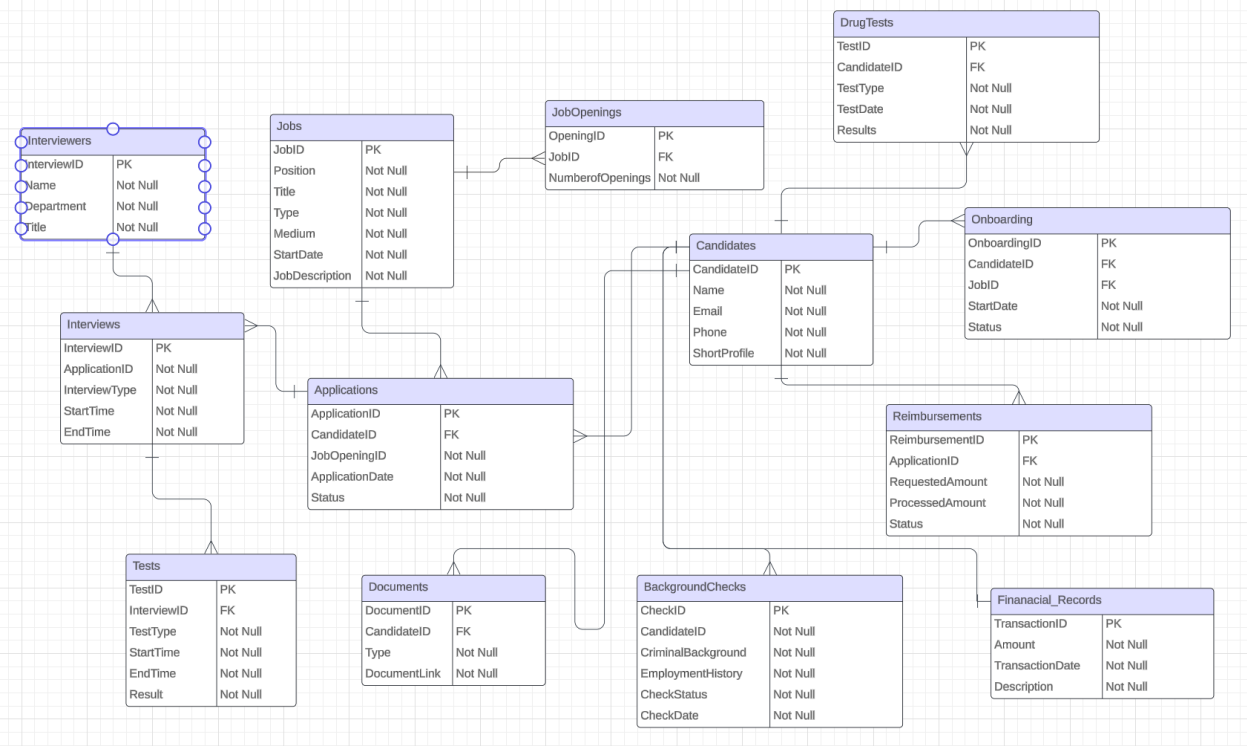
1. **Candidates**
    - **CandidateID** (Primary Key)
    - **Name**
    - **Email**
    - **Phone**
    - **ShortProfile**
2. **Jobs**
    - **JobID** (Primary Key)
    - **Position**
    - **Title**
    - **Type**
    - **Medium**
    - **StartDate**
    - **JobDescription**
3. **Job Openings**
    - **OpeningID** (Primary Key)
    - **JobID** (Foreign Key from Jobs)
    - **NumberOfPositions**
4. **Applications**
    - **ApplicationID** (Primary Key)
    - **CandidateID** (Foreign Key from Candidates)
    - **JobOpeningID** (Foreign Key from Job Openings)
    - **ApplicationDate**
    - **Status**
5. **Interviews**
    - **InterviewID** (Primary Key)
    - **ApplicationID** (Foreign Key from Applications)
    - **InterviewType**
    - **StartTime**
    - **EndTime**
6. **Tests**
    - **TestID** (Primary Key)
    - **InterviewID** (Foreign Key from Interviews)
    - **TestType**
    - **StartTime**
    - **EndTime**
    - **Result**

7. **Interviewers**
    - **InterviewerID** (Primary Key)
    - **Name**
    - **Department**
    - **Title**
8. **Documents**
    - **DocumentID** (Primary Key)
    - **CandidateID** (Foreign Key from Candidates)
    - **Type**
    - **DocumentLink**
9. **Background Checks**
    - **CheckID** (Primary Key)
    - **CandidateID** (Foreign Key from Candidates)
    - **CriminalBackground**
    - **EmploymentHistory**
    - **CheckStatus**
    - **CheckDate**
10. **Drug Tests**
    - **TestID** (Primary Key)
    - **CandidateID** (Foreign Key from Candidates)
    - **TestType**
    - **TestDate**
    - **Results**
11. **Reimbursements**
    - **ReimbursementID** (Primary Key)
    - **ApplicationID** (Foreign Key from Applications)
    - **RequestedAmount**
    - **ProcessedAmount**
    - **Status**
12. **Onboarding**
    - **OnboardingID** (Primary Key)
    - **CandidateID** (Foreign Key from Candidates)
    - **JobID** (Foreign Key from Jobs)
    - **StartDate**
    - **Status**

Relationships

- **Candidates** to **Applications**: One-to-Many (A candidate can have multiple applications)
- **Jobs** to **Job Openings**: One-to-Many (A job can have multiple openings)
- **Job Openings** to **Applications**: One-to-Many (A job opening can receive multiple applications)
- **Applications** to **Interviews**: One-to-Many (An application can lead to multiple interviews)
- **Interviews** to **Tests**: One-to-Many (An interview can have multiple tests)

- **Candidates** to **Documents**, **Background Checks**, **Drug Tests**, **Onboarding**: One-to-Many (A candidate can have multiple documents, background checks, drug tests, and one onboarding process)

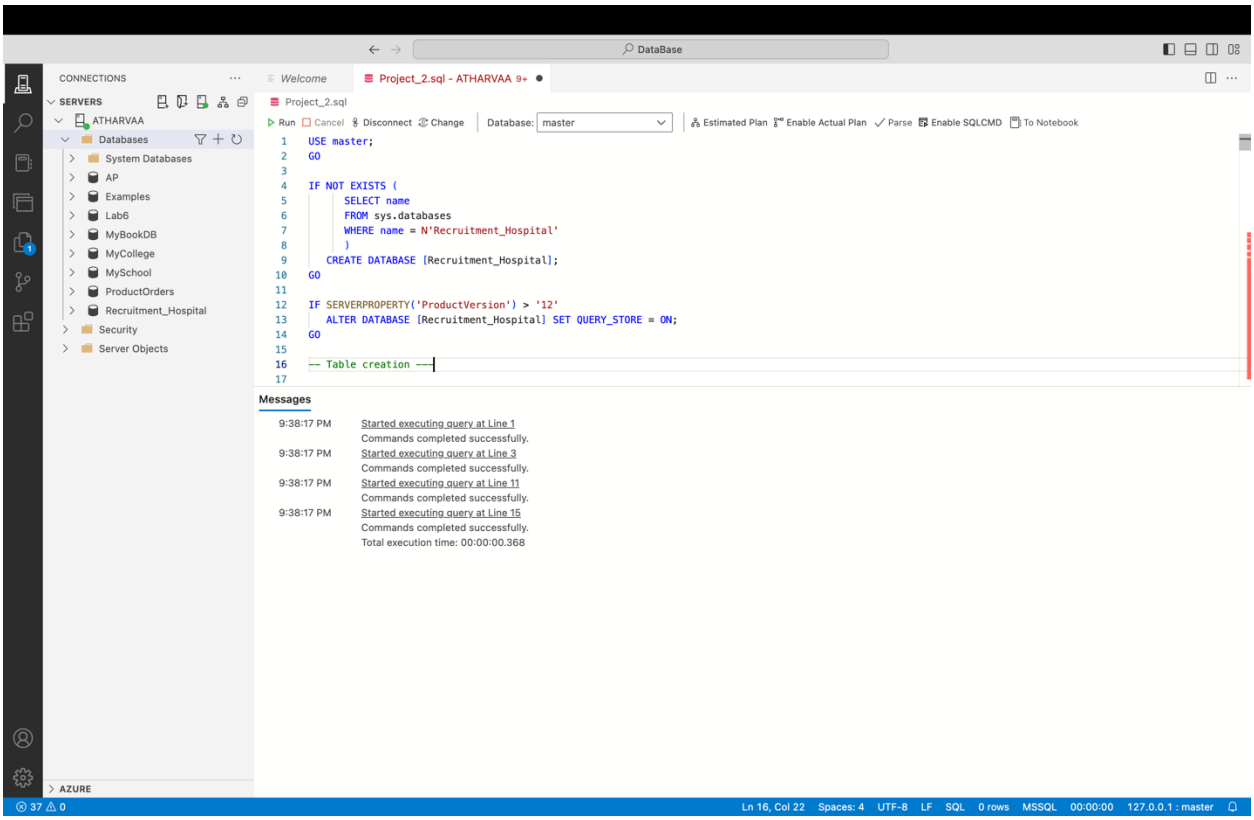


Create Commands

Query1:

```
USE master;
GO

IF NOT EXISTS (
      SELECT name
      FROM sys.databases
      WHERE name = N'Recruitment_Hospital'
      )
   CREATE DATABASE [Recruitment_Hospital];
GO

IF SERVERPROPERTY('ProductVersion') > '12'
   ALTER DATABASE [Recruitment_Hospital] SET QUERY_STORE = ON;
GO
```

This SQL script checks if the "Recruitment_Hospital" database exists in SQL Server; if not, it creates it. Additionally, it enables the Query Store feature for performance analysis if the server version is above 12.




```
-- We are creating a Table ---
```

Create Table:

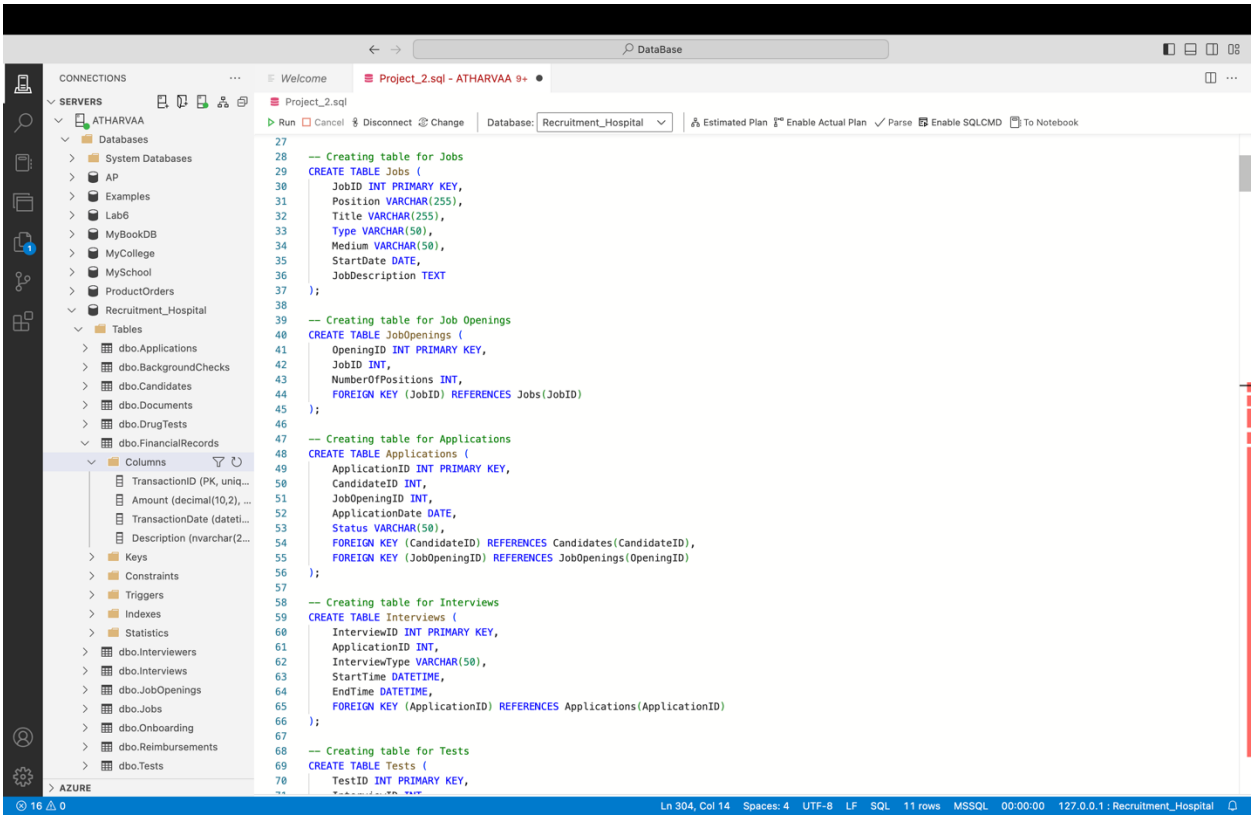
```
-- Creating table for Candidates
CREATE TABLE Candidates (
    CandidateID INT PRIMARY KEY,
    Name VARCHAR(255),
    Email VARCHAR(255) UNIQUE,
    Phone VARCHAR(50),
    ShortProfile TEXT
);

-- Creating table for Jobs
CREATE TABLE Jobs (
    JobID INT PRIMARY KEY,
    Position VARCHAR(255),
    Title VARCHAR(255),
    Type VARCHAR(50),
    Medium VARCHAR(50),
    StartDate DATE,
    JobDescription TEXT
);

-- Creating table for Job Openings
```

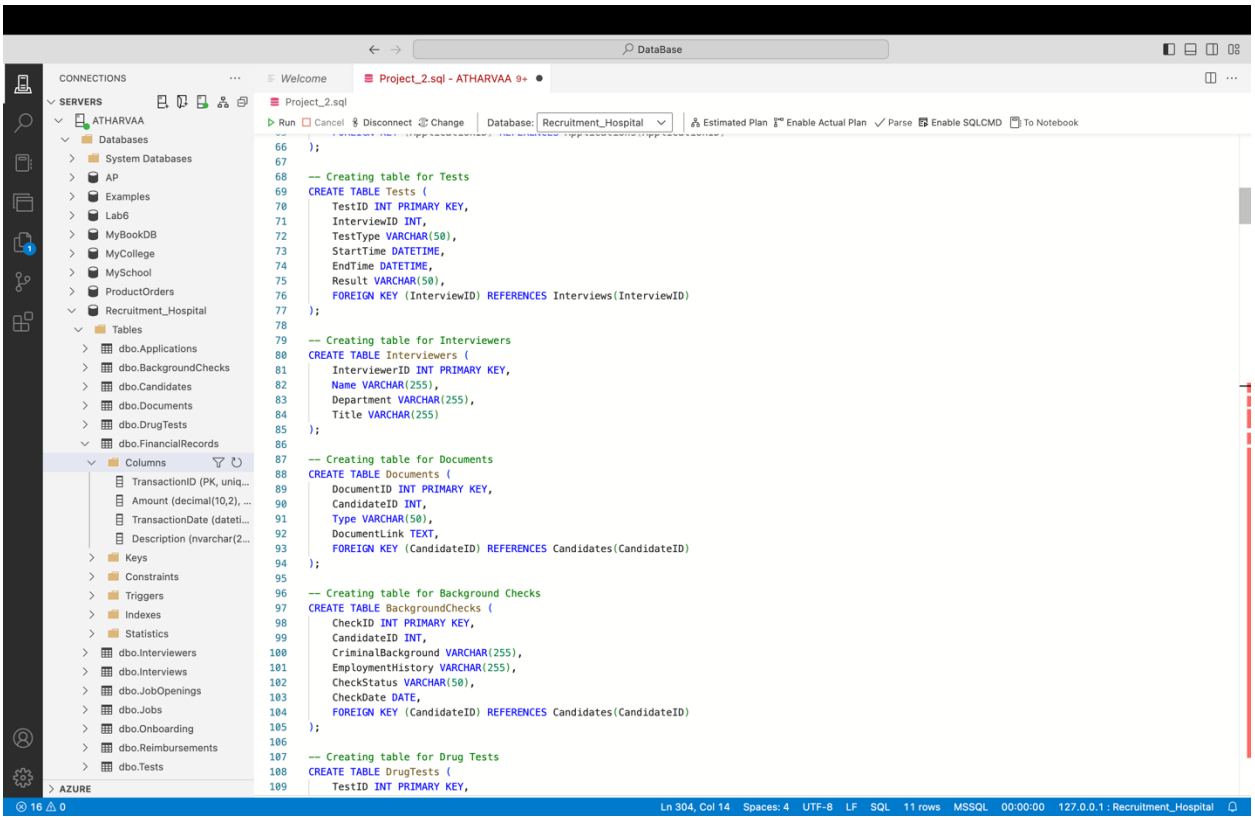
```sql
CREATE TABLE JobOpenings (
    OpeningID INT PRIMARY KEY,
    JobID INT,
    NumberOfPositions INT,
    FOREIGN KEY (JobID) REFERENCES Jobs(JobID)
);

-- Creating table for Applications
CREATE TABLE Applications (
    ApplicationID INT PRIMARY KEY,
    CandidateID INT,
    JobOpeningID INT,
    ApplicationDate DATE,
    Status VARCHAR(50),
    FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID),
    FOREIGN KEY (JobOpeningID) REFERENCES JobOpenings(OpeningID)
);

-- Creating table for Interviews
CREATE TABLE Interviews (
    InterviewID INT PRIMARY KEY,
    ApplicationID INT,
    InterviewType VARCHAR(50),
    StartTime DATETIME,
    EndTime DATETIME,
    FOREIGN KEY (ApplicationID) REFERENCES Applications(ApplicationID)
);

-- Creating table for Tests
CREATE TABLE Tests (
    TestID INT PRIMARY KEY,
    InterviewID INT,
    TestType VARCHAR(50),
    StartTime DATETIME,
    EndTime DATETIME,
    Result VARCHAR(50),
    FOREIGN KEY (InterviewID) REFERENCES Interviews(InterviewID)
);

-- Creating table for Interviewers
CREATE TABLE Interviewers (
    InterviewerID INT PRIMARY KEY,
    Name VARCHAR(255),
    Department VARCHAR(255),
    Title VARCHAR(255)
);

-- Creating table for Documents
```

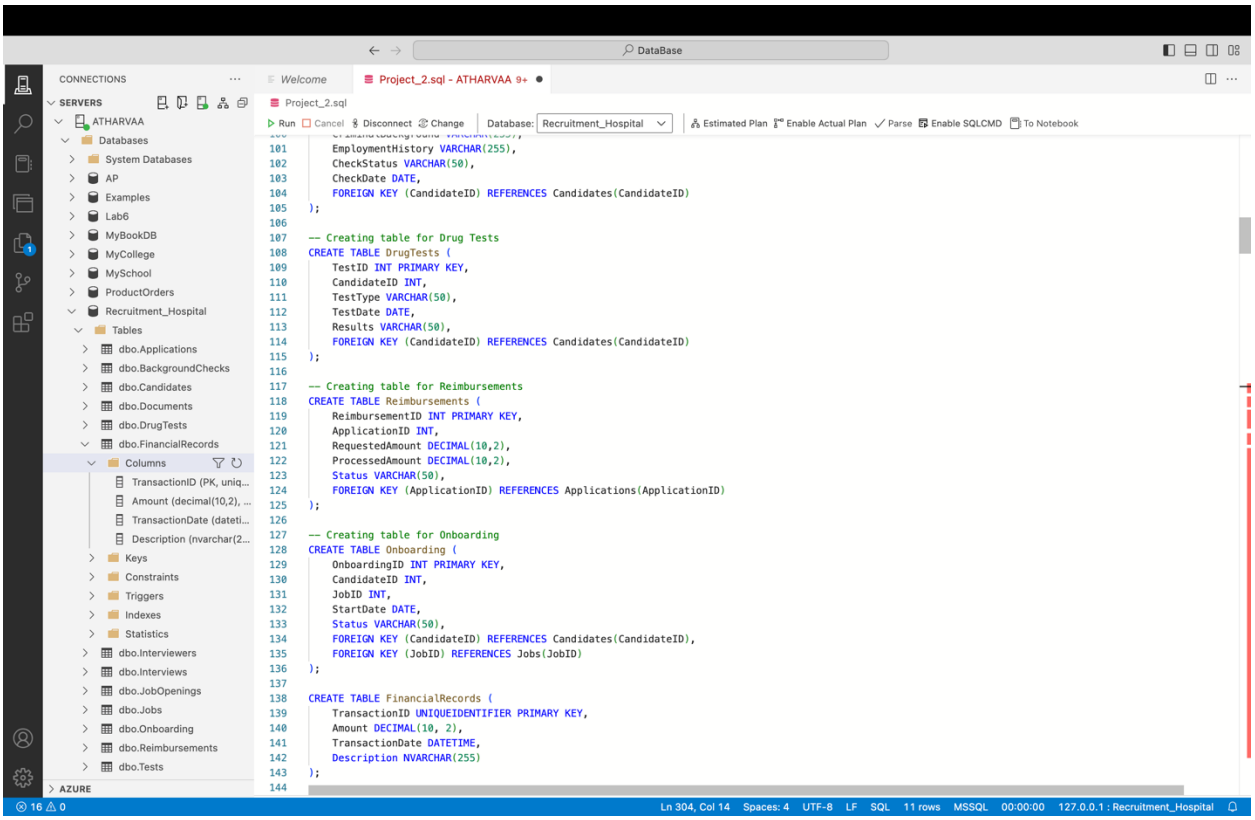
```sql
CREATE TABLE Documents (
    DocumentID INT PRIMARY KEY,
    CandidateID INT,
    Type VARCHAR(50),
    DocumentLink TEXT,
    FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
);

-- Creating table for Background Checks
CREATE TABLE BackgroundChecks (
    CheckID INT PRIMARY KEY,
    CandidateID INT,
    CriminalBackground VARCHAR(255),
    EmploymentHistory VARCHAR(255),
    CheckStatus VARCHAR(50),
    CheckDate DATE,
    FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
);

-- Creating table for Drug Tests
CREATE TABLE DrugTests (
    TestID INT PRIMARY KEY,
    CandidateID INT,
    TestType VARCHAR(50),
    TestDate DATE,
    Results VARCHAR(50),
    FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
);

-- Creating table for Reimbursements
CREATE TABLE Reimbursements (
    ReimbursementID INT PRIMARY KEY,
    ApplicationID INT,
    RequestedAmount DECIMAL(10,2),
    ProcessedAmount DECIMAL(10,2),
    Status VARCHAR(50),
    FOREIGN KEY (ApplicationID) REFERENCES Applications(ApplicationID)
);

-- Creating table for Onboarding
CREATE TABLE Onboarding (
    OnboardingID INT PRIMARY KEY,
    CandidateID INT,
    JobID INT,
    StartDate DATE,
    Status VARCHAR(50),
    FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID),
    FOREIGN KEY (JobID) REFERENCES Jobs(JobID)
```

```
);
```

This SQL script outlines the creation of a comprehensive database structure for managing a recruitment system. It includes tables for candidates, jobs, job openings, applications, interviews, and related entities like tests, interviewers, documents, background checks, drug tests, reimbursements, and onboarding. Each table is equipped with primary keys for identification and foreign keys to establish relationships between them, ensuring data integrity and relational management across the recruitment process.




DataBase
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
Columns
TransactionID (PK, uniq...
Amount (decimal(10,2), ...
TransactionDate (dateti...
Description (nvarchar(2...
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
66);
67
68 -- Creating table for Tests
69 CREATE TABLE Tests (
70 TestID INT PRIMARY KEY,
71 InterviewID INT,
72 TestType VARCHAR(50),
73 StartTime DATETIME,
74 EndTime DATETIME,
75 Result VARCHAR(50),
76 FOREIGN KEY (InterviewID) REFERENCES Interviews(InterviewID)
77);
78
79 -- Creating table for Interviewers
80 CREATE TABLE Interviewers (
81 InterviewerID INT PRIMARY KEY,
82 Name VARCHAR(255),
83 Department VARCHAR(255),
84 Title VARCHAR(255)
85);
86
87 -- Creating table for Documents
88 CREATE TABLE Documents (
89 DocumentID INT PRIMARY KEY,
90 CandidateID INT,
91 Type VARCHAR(50),
92 DocumentLink TEXT,
93 FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
94);
95
96 -- Creating table for Background Checks
97 CREATE TABLE BackgroundChecks (
98 CheckID INT PRIMARY KEY,
99 CandidateID INT,
100 CriminalBackground VARCHAR(255),
101 EmploymentHistory VARCHAR(255),
102 CheckStatus VARCHAR(50),
103 CheckDate DATE,
104 FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
105);
106
107 -- Creating table for Drug Tests
108 CREATE TABLE DrugTests (
109 TestID INT PRIMARY KEY,
Ln 304, Col 14 Spaces: 4 UTF-8 LF SQL 11 rows MSSQL 00:00:00 127.0.0.1 : Recruitment_Hospital


DataBase
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
Columns
TransactionID (PK, uniq...
Amount (decimal(10,2), ...
TransactionDate (dateti...
Description (nvarchar(2...
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
101 EmploymentHistory VARCHAR(255),
102 CheckStatus VARCHAR(50),
103 CheckDate DATE,
104 FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
105);
106
107 -- Creating table for Drug Tests
108 CREATE TABLE DrugTests (
109 TestID INT PRIMARY KEY,
110 CandidateID INT,
111 TestType VARCHAR(50),
112 TestDate DATE,
113 Results VARCHAR(50),
114 FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID)
115);
116
117 -- Creating table for Reimbursements
118 CREATE TABLE Reimbursements (
119 ReimbursementID INT PRIMARY KEY,
120 ApplicationID INT,
121 RequestedAmount DECIMAL(10,2),
122 ProcessedAmount DECIMAL(10,2),
123 Status VARCHAR(50),
124 FOREIGN KEY (ApplicationID) REFERENCES Applications(ApplicationID)
125);
126
127 -- Creating table for Onboarding
128 CREATE TABLE Onboarding (
129 OnboardingID INT PRIMARY KEY,
130 CandidateID INT,
131 JobID INT,
132 StartDate DATE,
133 Status VARCHAR(50),
134 FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID),
135 FOREIGN KEY (JobID) REFERENCES Jobs(JobID)
136);
137
138 CREATE TABLE FinancialRecords (
139 TransactionID UNIQUEIDENTIFIER PRIMARY KEY,
140 Amount DECIMAL(10, 2),
141 TransactionDate DATETIME,
142 Description NVARCHAR(255)
143);
144
Ln 304, Col 14 Spaces: 4 UTF-8 LF SQL 11 rows MSSQL 00:00:00 127.0.0.1 : Recruitment_Hospital

Insert Commands

This SQL script efficiently populates a recruitment system database by inserting detailed entries into tables that manage candidates, job openings, applications, interviews, and several supportive elements such as drug tests and reimbursements. The script meticulously establishes relationships using foreign keys, ensuring data consistency across different facets of the recruitment process. It covers the entire hiring lifecycle, from candidate entry, job listing, and application management to interview scheduling, background verification, and onboarding, demonstrating a thorough approach to recruitment data management.

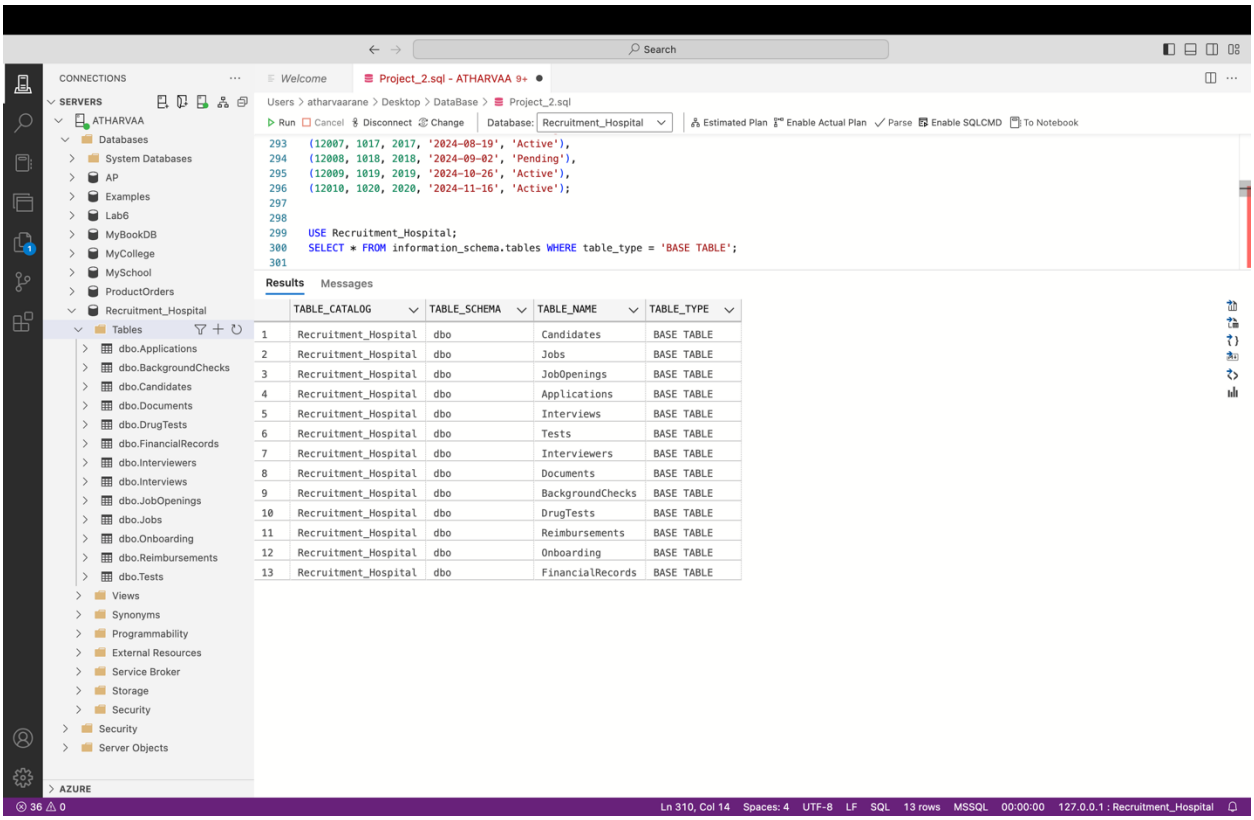
```
INSERT INTO Candidates (CandidateID, Name, Email, Phone, ShortProfile) VALUES
(1011, 'Atharvaa Rane', 'atharvaa.rane@syr.edu', '315-555-2974', 'Radiologist with a
special interest in musculoskeletal imaging and sports injuries.'),
(1012, 'Nolan Case', 'nolan.case@healthnet.com', '716-555-4521', 'Epidemiologist with
a strong background in infectious diseases and public health.'),
(1013, 'Lydia Bennett', 'lydia.bennett@clinic.org', '607-555-9832', 'Dermatologist
focusing on pediatric and adolescent skin conditions.'),
(1014, 'Reed Morales', 'reed.morales@pioneerhealth.com', '845-555-7391', 'Biochemist
with expertise in pharmaceutical developments and patient safety.'),
(1015, 'Eva Summers', 'eva.summers@welllife.com', '518-555-8264', 'Nutritionist
specializing in dietetics for chronic diseases and senior health.'),
(1016, 'Hugo Ross', 'hugo.ross@medicalcore.com', '914-555-4930', 'General practitioner
with a comprehensive approach to primary care and wellness.'),
(1017, 'Vivian Lacroix', 'vivian.lacroix@integrahealth.com', '347-555-1824', 'A
seasoned physical therapist dedicated to rehabilitation and recovery processes.'),
(1018, 'Miles Archer', 'miles.archer@lifeline.com', '631-555-5010', 'Clinical research
coordinator focusing on trials for cutting-edge cancer treatments.'),
(1019, 'Tara Flynn', 'tara.flynn@careplus.org', '212-555-2213', 'Anesthesiologist with
a focus on perioperative care and pain management.'),
(1020, 'Leo Carver', 'leo.carver@pharmtech.com', '646-555-6668', 'Pharmaceutical
technician engaged in medication therapy management and patient counseling.');

INSERT INTO Jobs (JobID, Position, Title, Type, Medium, StartDate, JobDescription)
VALUES
(2011, 'Cardiologist', 'Consultant Cardiologist', 'Full-time', 'Onsite', '2024-02-20',
'Diagnoses and treats patients with heart and cardiovascular conditions.'),
(2012, 'Medical Illustrator', 'Lead Medical Illustrator', 'Contract', 'Remote', '2024-
03-15', 'Creates detailed medical graphics to aid patient understanding and medical
education.'),
(2013, 'Genetic Counselor', 'Senior Genetic Counselor', 'Part-time', 'Onsite', '2024-
04-10', 'Provides genetic counseling and conducts risk assessments for patients.'),
(2014, 'Neurologist', 'Chief of Neurology', 'Full-time', 'Onsite', '2024-05-05',
'Heads the department of neurology and oversees treatment of neurological
disorders.'),
```

```
(2015, 'Medical Coder', 'Medical Coding Specialist', 'Full-time', 'Remote', '2024-06-
12', 'Ensures accurate coding of health services and procedures for billing and
documentation.'),
(2016, 'Pediatric Oncologist', 'Pediatric Cancer Specialist', 'Full-time', 'Onsite',
'2024-07-22', 'Specializes in diagnosing and treating cancers affecting children.'),
(2017, 'Speech Therapist', 'Speech-Language Pathologist', 'Full-time', 'Onsite',
'2024-08-18', 'Assists patients in improving speech disorders and swallowing
difficulties.'),
(2018, 'Health Data Analyst', 'Senior Data Analyst', 'Contract', 'Remote', '2024-09-
01', 'Analyzes health data to help improve hospital systems and patient care
outcomes.'),
(2019, 'Occupational Therapist', 'Lead Occupational Therapist', 'Part-time', 'Onsite',
'2024-10-25', 'Helps patients develop and recover daily living and work skills.'),
(2020, 'Clinical Psychologist', 'Behavioral Health Specialist', 'Full-time', 'Onsite',
'2024-11-15', 'Provides psychological assessments and therapy to patients dealing with
mental health issues.');


INSERT INTO JobOpenings (OpeningID, JobID, NumberOfPositions) VALUES
(3011, 2011, 2),  -- Cardiologist
(3012, 2012, 1),  -- Medical Illustrator
(3013, 2013, 1),  -- Genetic Counselor
(3014, 2014, 1),  -- Neurologist
(3015, 2015, 2),  -- Medical Coder
(3016, 2016, 2),  -- Pediatric Oncologist
(3017, 2017, 2),  -- Speech Therapist
(3018, 2018, 1),  -- Health Data Analyst
(3019, 2019, 2),  -- Occupational Therapist
(3020, 2020, 1);  -- Clinical Psychologist


-- Inserting application data into the 'Applications' table
INSERT INTO Applications (ApplicationID, CandidateID, JobOpeningID, ApplicationDate,
Status) VALUES
(4011, 1011, 3011, '2023-10-04', 'Under Review'),
(4012, 1012, 3012, '2023-09-20', 'Submitted'),
(4013, 1013, 3013, '2023-09-18', 'Under Review'),
(4014, 1014, 3014, '2023-09-15', 'Submitted'),
(4015, 1015, 3015, '2023-10-07', 'Under Review'),
(4016, 1016, 3016, '2023-09-22', 'Submitted'),
(4017, 1017, 3017, '2023-09-30', 'Under Review'),
(4018, 1018, 3018, '2023-09-21', 'Submitted'),
(4019, 1019, 3019, '2023-10-01', 'Submitted'),
(4020, 1020, 3020, '2023-09-19', 'Under Review');

-- Inserting data into the 'Interviews' table
INSERT INTO Interviews (InterviewID, ApplicationID, InterviewType, StartTime, EndTime)
VALUES
```

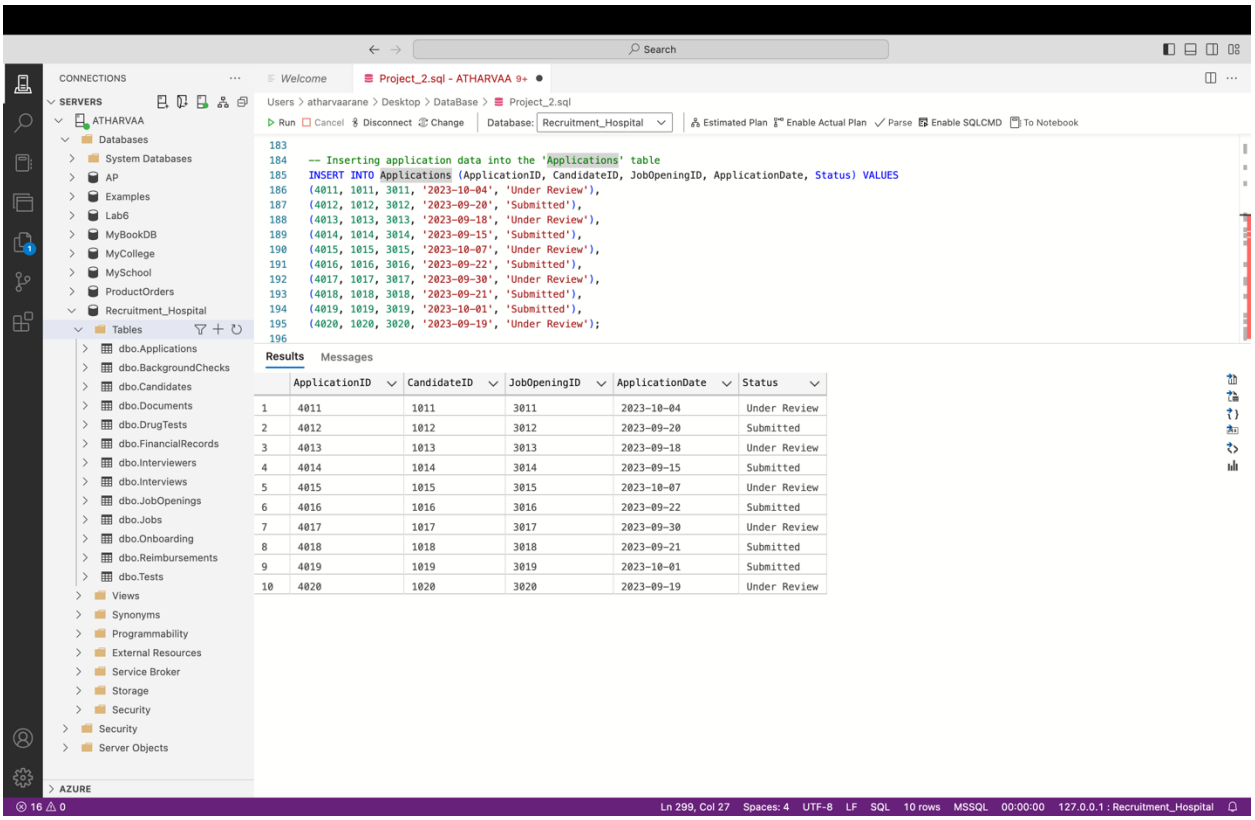
```
(5001, 4011, 'Initial Screening', '2023-10-05 09:00:00', '2023-10-05 10:00:00'),
(5002, 4012, 'Technical Interview', '2023-09-21 11:00:00', '2023-09-21 12:30:00'),
(5003, 4013, 'Panel Interview', '2023-09-19 14:00:00', '2023-09-19 15:30:00'),
(5004, 4014, 'HR Interview', '2023-09-16 10:00:00', '2023-09-16 11:00:00'),
(5005, 4015, 'Technical Interview', '2023-10-08 13:00:00', '2023-10-08 14:00:00'),
(5006, 4016, 'Panel Interview', '2023-09-23 10:00:00', '2023-09-23 11:30:00'),
(5007, 4017, 'HR Interview', '2023-09-01 09:00:00', '2023-09-01 10:00:00'),
(5008, 4018, 'Initial Screening', '2023-09-22 13:00:00', '2023-09-22 14:00:00'),
(5009, 4019, 'Technical Interview', '2023-10-02 10:00:00', '2023-10-02 11:00:00'),
(5010, 4020, 'HR Interview', '2023-09-20 09:00:00', '2023-09-20 10:00:00');


INSERT INTO Tests (TestID, InterviewID, TestType, StartTime, EndTime, Result) VALUES
(6001, 5001, 'Personality Test', '2023-10-05 11:00:00', '2023-10-05 12:00:00',
'Passed'),
(6002, 5002, 'Coding Test', '2023-09-21 13:30:00', '2023-09-21 15:30:00', 'Failed'),
(6003, 5003, 'Panel Review', '2023-09-19 15:00:00', '2023-09-19 16:00:00', 'Passed'),
(6004, 5004, 'Skill Assessment', '2023-09-16 10:30:00', '2023-09-16 11:30:00',
'Failed'),
(6005, 5005, 'Technical Quiz', '2023-10-08 14:30:00', '2023-10-08 15:30:00',
'Passed'),
(6006, 5006, 'Clinical Knowledge Test', '2023-09-23 11:00:00', '2023-09-23 12:00:00',
'Failed'),
(6007, 5007, 'Management Simulation', '2023-09-01 09:30:00', '2023-09-01 10:30:00',
'Passed'),
(6008, 5008, 'Communication Test', '2023-09-22 14:00:00', '2023-09-22 15:00:00',
'Failed'),
(6009, 5009, 'Problem Solving Test', '2023-10-02 11:00:00', '2023-10-02 12:00:00',
'Passed'),
(6010, 5010, 'Ethical Reasoning Test', '2023-09-20 09:30:00', '2023-09-20 10:30:00',
'Passed');

INSERT INTO Interviewers (InterviewerID, Name, Department, Title) VALUES
(7001, 'Dr. Lisa Ray', 'Cardiology', 'Senior Cardiologist'),
(7002, 'Mr. James Smith', 'IT', 'IT Manager'),
(7003, 'Ms. Nancy Drew', 'HR', 'HR Director'),
(7004, 'Dr. Alan Grant', 'Oncology', 'Lead Oncologist'),
(7005, 'Ms. Claire Dearing', 'Pediatrics', 'Pediatric Nurse Manager'),
(7006, 'Dr. Henry Wu', 'Genetics', 'Chief Geneticist'),
(7007, 'Mr. Owen Grady', 'Security', 'Security Supervisor'),
(7008, 'Ms. Ellie Sattler', 'Botany', 'Head of Botanical Research'),
(7009, 'Dr. Ian Malcolm', 'Mathematics', 'Statistical Analyst'),
(7010, 'Dr. Victor Fries', 'Cryogenics', 'Cryogenics Specialist');

INSERT INTO Documents (DocumentID, CandidateID, Type, DocumentLink) VALUES
(8001, 1011, 'Resume', 'https://docs.example.com/resumes/1011'),
(8002, 1012, 'Cover Letter', 'https://docs.example.com/coverletters/1012'),
(8003, 1013, 'Portfolio', 'https://docs.example.com/portfolios/1013'),
```

```
(8004, 1014, 'Resume', 'https://docs.example.com/resumes/1014'),
(8005, 1015, 'Cover Letter', 'https://docs.example.com/coverletters/1015'),
(8006, 1016, 'Portfolio', 'https://docs.example.com/portfolios/1016'),
(8007, 1017, 'Resume', 'https://docs.example.com/resumes/1017'),
(8008, 1018, 'Cover Letter', 'https://docs.example.com/coverletters/1018'),
(8009, 1019, 'Portfolio', 'https://docs.example.com/portfolios/1019'),
(8010, 1020, 'Resume', 'https://docs.example.com/resumes/1020');


INSERT INTO BackgroundChecks (CheckID, CandidateID, CriminalBackground,
EmploymentHistory, CheckStatus, CheckDate) VALUES
(9001, 1011, 'None', 'Verified', 'Completed', '2023-09-25'),
(9002, 1012, 'None', 'Verified', 'Completed', '2023-09-26'),
(9003, 1013, 'Minor', 'Verified', 'Pending', '2023-09-27'),
(9004, 1014, 'None', 'Verified', 'Completed', '2023-09-28'),
(9005, 1015, 'None', 'Verified', 'Completed', '2023-09-29'),
(9006, 1016, 'None', 'Verified', 'Completed', '2023-10-01'),
(9007, 1017, 'None', 'Verified', 'Completed', '2023-10-02'),
(9008, 1018, 'None', 'Verified', 'Completed', '2023-10-03'),
(9009, 1019, 'None', 'Verified', 'Completed', '2023-10-04'),
(9010, 1020, 'None', 'Verified', 'Completed', '2023-10-05');


INSERT INTO DrugTests (TestID, CandidateID, TestType, TestDate, Results) VALUES
(10001, 1011, 'Urine Test', '2023-09-30', 'Negative'),
(10002, 1012, 'Urine Test', '2023-10-01', 'Negative'),
(10003, 1013, 'Urine Test', '2023-10-02', 'Positive'),
(10004, 1014, 'Urine Test', '2023-10-03', 'Negative'),
(10005, 1015, 'Urine Test', '2023-10-04', 'Negative'),
(10006, 1016, 'Urine Test', '2023-10-05', 'Negative'),
(10007, 1017, 'Urine Test', '2023-10-06', 'Negative'),
(10008, 1018, 'Urine Test', '2023-10-07', 'Negative'),
(10009, 1019, 'Urine Test', '2023-10-08', 'Negative'),
(10010, 1020, 'Urine Test', '2023-10-09', 'Negative');


INSERT INTO Reimbursements (ReimbursementID, ApplicationID, RequestedAmount,
ProcessedAmount, Status) VALUES
(11001, 4011, 200.00, 200.00, 'Processed'),
(11002, 4012, 150.00, 150.00, 'Processed'),
(11003, 4013, 180.00, 180.00, 'Processed'),
(11004, 4014, 300.00, 290.00, 'Processed'),
(11005, 4015, 250.00, 250.00, 'Processed'),
(11006, 4016, 190.00, 180.00, 'Processed'),
(11007, 4017, 210.00, 200.00, 'Processed'),
(11008, 4018, 100.00, 100.00, 'Processed'),
(11009, 4019, 200.00, 190.00, 'Processed'),
(11010, 4020, 250.00, 250.00, 'Processed');
```

```
INSERT INTO Onboarding (OnboardingID, CandidateID, JobID, StartDate, Status) VALUES
(12001, 1011, 2011, '2024-02-21', 'Active'),
(12002, 1012, 2012, '2024-03-16', 'Active'),
(12003, 1013, 2013, '2024-04-11', 'Pending'),
(12004, 1014, 2014, '2024-05-06', 'Active'),
(12005, 1015, 2015, '2024-06-13', 'Active'),
(12006, 1016, 2016, '2024-07-23', 'Pending'),
(12007, 1017, 2017, '2024-08-19', 'Active'),
(12008, 1018, 2018, '2024-09-02', 'Pending'),
(12009, 1019, 2019, '2024-10-26', 'Active'),
(12010, 1020, 2020, '2024-11-16', 'Active');

INSERT INTO FinancialRecords (TransactionID, Amount, TransactionDate, Description)
VALUES
(NEWID(), 1500.00, '2024-05-01', 'Payment for medical equipment'),
(NEWID(), 200.00, '2024-05-02', 'Refund of overcharged fees'),
(NEWID(), 1200.00, '2024-05-03', 'Consultation fees for cardiology department'),
(NEWID(), 750.00, '2024-05-04', 'Purchase of new office furniture for clinic'),
(NEWID(), 350.00, '2024-05-05', 'Software license renewal for administrative
operations');
```

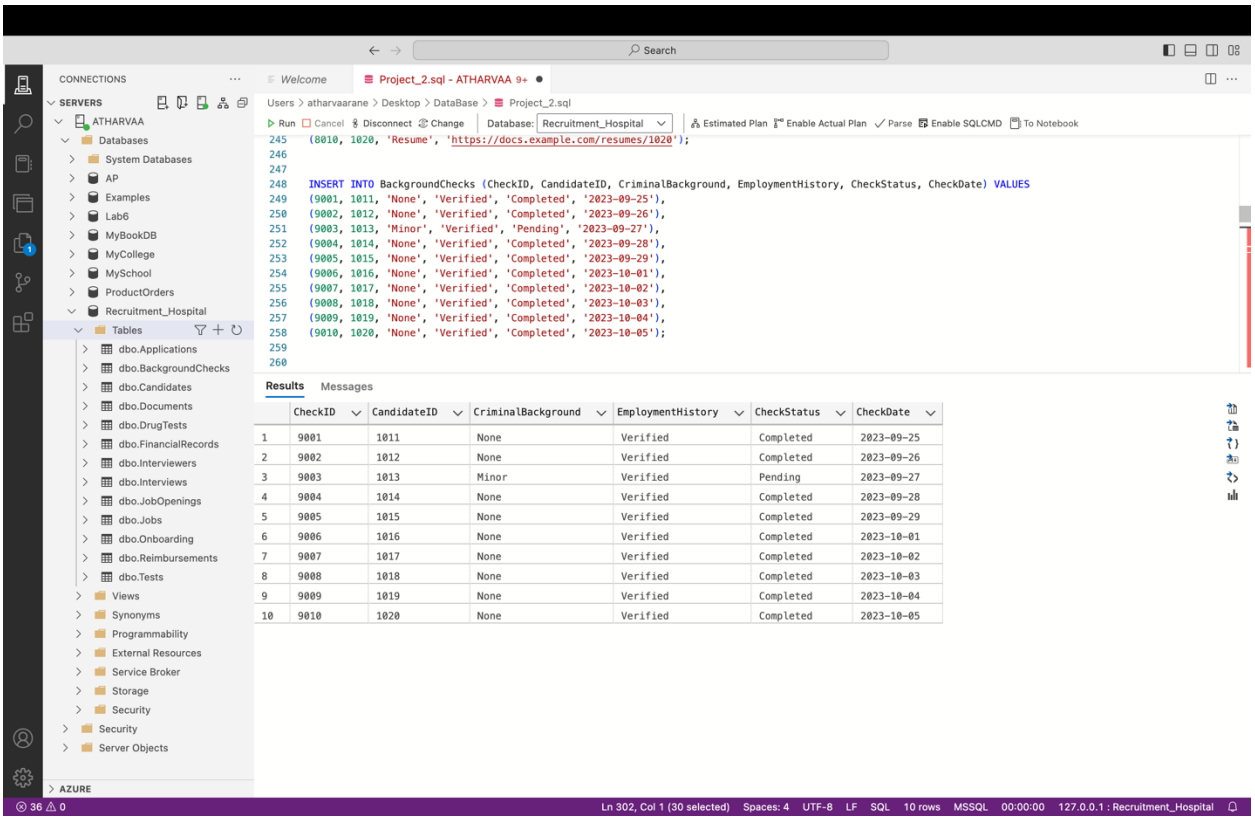


- This SQL command selects and lists all base tables from the specified database “Recruitment_Hospital". It uses the `information_schema.tables` to fetch table details, filtering to show only user-created tables.
- We can see that these are the tables that are available.



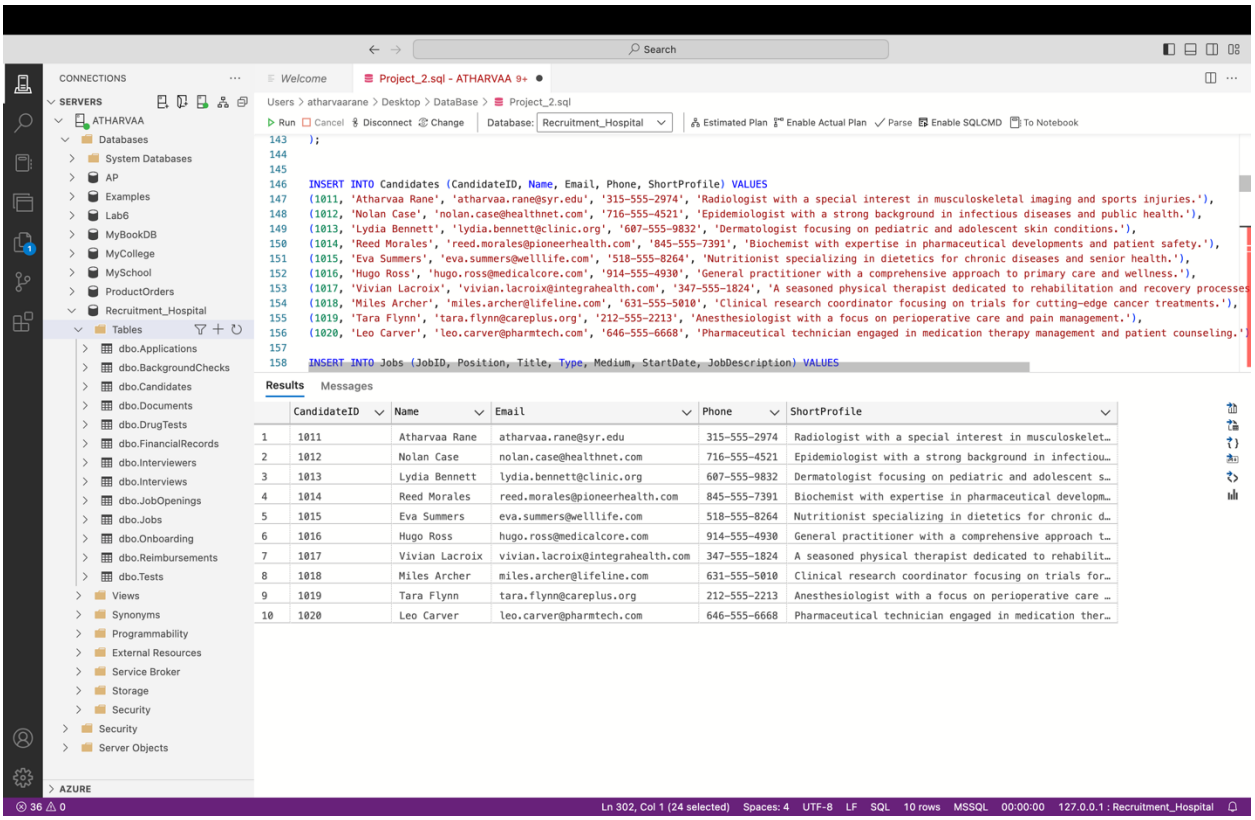
```
-- Inserting application data into the 'Applications' table
INSERT INTO Applications (ApplicationID, CandidateID, JobOpeningID, ApplicationDate, Status) VALUES
(4011, 1011, 3011, '2023-10-04', 'Under Review'),
(4012, 1012, 3012, '2023-09-20', 'Submitted'),
(4013, 1013, 3013, '2023-09-18', 'Under Review'),
(4014, 1014, 3014, '2023-09-15', 'Submitted'),
(4015, 1015, 3015, '2023-10-07', 'Under Review'),
(4016, 1016, 3016, '2023-09-22', 'Submitted'),
(4017, 1017, 3017, '2023-09-30', 'Under Review'),
(4018, 1018, 3018, '2023-09-21', 'Submitted'),
(4019, 1019, 3019, '2023-10-01', 'Submitted'),
(4020, 1020, 3020, '2023-09-19', 'Under Review');
```

	ApplicationID	CandidateID	JobOpeningID	ApplicationDate	Status
1	4011	1011	3011	2023-10-04	Under Review
2	4012	1012	3012	2023-09-20	Submitted
3	4013	1013	3013	2023-09-18	Under Review
4	4014	1014	3014	2023-09-15	Submitted
5	4015	1015	3015	2023-10-07	Under Review
6	4016	1016	3016	2023-09-22	Submitted
7	4017	1017	3017	2023-09-30	Under Review
8	4018	1018	3018	2023-09-21	Submitted
9	4019	1019	3019	2023-10-01	Submitted
10	4020	1020	3020	2023-09-19	Under Review



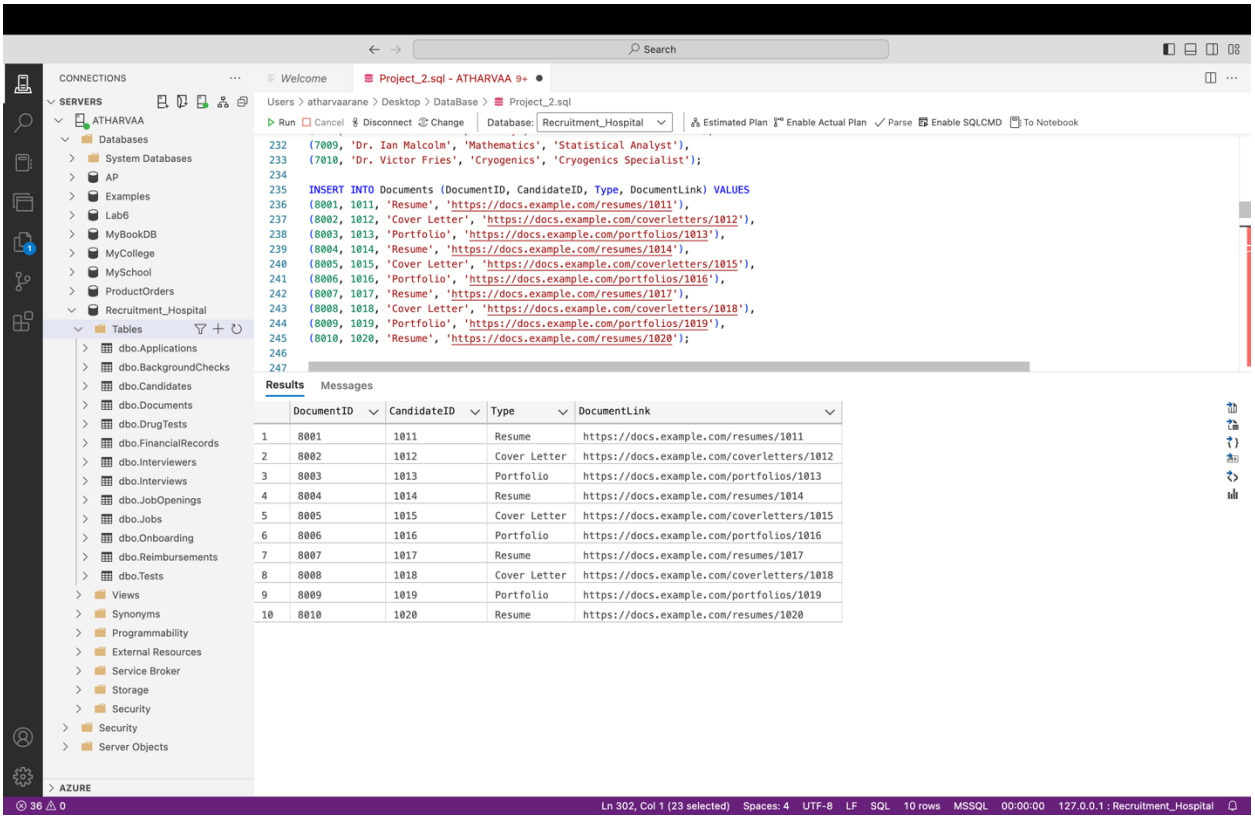
```
(8010, 1020, 'Resume', 'https://docs.example.com/resumes/1020');


INSERT INTO BackgroundChecks (CheckID, CandidateID, CriminalBackground, EmploymentHistory, CheckStatus, CheckDate) VALUES
(9001, 1011, 'None', 'Verified', 'Completed', '2023-09-25'),
(9002, 1012, 'None', 'Verified', 'Completed', '2023-09-26'),
(9003, 1013, 'Minor', 'Verified', 'Pending', '2023-09-27'),
(9004, 1014, 'None', 'Verified', 'Completed', '2023-09-28'),
(9005, 1015, 'None', 'Verified', 'Completed', '2023-09-29'),
(9006, 1016, 'None', 'Verified', 'Completed', '2023-10-01'),
(9007, 1017, 'None', 'Verified', 'Completed', '2023-10-02'),
(9008, 1018, 'None', 'Verified', 'Completed', '2023-10-03'),
(9009, 1019, 'None', 'Verified', 'Completed', '2023-10-04'),
(9010, 1020, 'None', 'Verified', 'Completed', '2023-10-05');
```

	CheckID	CandidateID	CriminalBackground	EmploymentHistory	CheckStatus	CheckDate
1	9001	1011	None	Verified	Completed	2023-09-25
2	9002	1012	None	Verified	Completed	2023-09-26
3	9003	1013	Minor	Verified	Pending	2023-09-27
4	9004	1014	None	Verified	Completed	2023-09-28
5	9005	1015	None	Verified	Completed	2023-09-29
6	9006	1016	None	Verified	Completed	2023-10-01
7	9007	1017	None	Verified	Completed	2023-10-02
8	9008	1018	None	Verified	Completed	2023-10-03
9	9009	1019	None	Verified	Completed	2023-10-04
10	9010	1020	None	Verified	Completed	2023-10-05



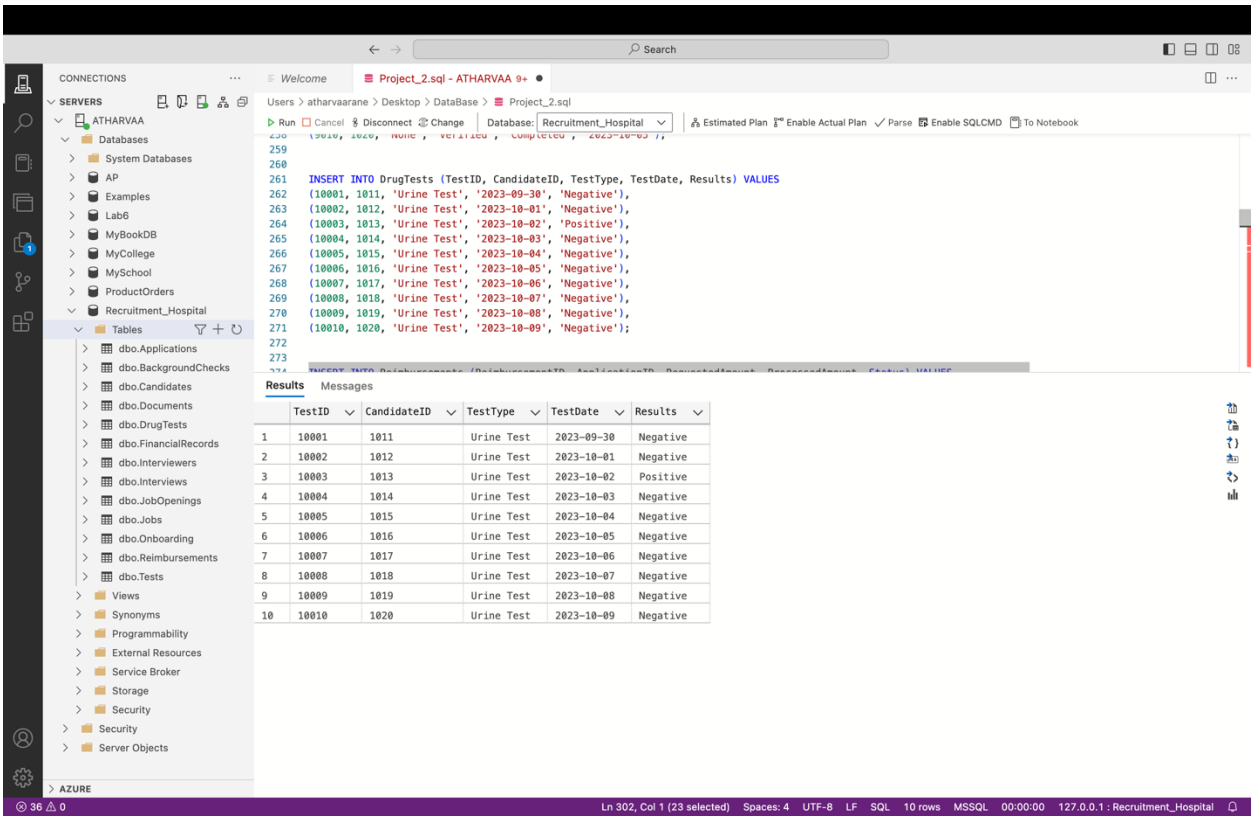
```
);


INSERT INTO Candidates (CandidateID, Name, Email, Phone, ShortProfile) VALUES
(1011, 'Atharvaa Rane', 'atharvaa.rane@syr.edu', '315-555-2974', 'Radiologist with a special interest in musculoskeletal imaging and sports injuries.'),
(1012, 'Nolan Case', 'nolan.case@healthnet.com', '716-555-4521', 'Epidemiologist with a strong background in infectious diseases and public health.'),
(1013, 'Lydia Bennett', 'lydia.bennett@clinic.org', '607-555-9832', 'Dermatologist focusing on pediatric and adolescent skin conditions.'),
(1014, 'Reed Morales', 'reed.morales@pioneerhealth.com', '845-555-7391', 'Biochemist with expertise in pharmaceutical developments and patient safety.'),
(1015, 'Eva Summers', 'eva.summers@welllife.com', '518-555-8264', 'Nutritionist specializing in dietetics for chronic diseases and senior health.'),
(1016, 'Hugo Ross', 'hugo.ross@medicalcore.com', '914-555-4930', 'General practitioner with a comprehensive approach to primary care and wellness.'),
(1017, 'Vivian Lacroix', 'vivian.lacroix@integrahealth.com', '347-555-1824', 'A seasoned physical therapist dedicated to rehabilitation and recovery processes'),
(1018, 'Miles Archer', 'miles.archer@lifeline.com', '631-555-5010', 'Clinical research coordinator focusing on trials for cutting-edge cancer treatments.'),
(1019, 'Tara Flynn', 'tara.flynn@careplus.org', '212-555-2213', 'Anesthesiologist with a focus on perioperative care and pain management.'),
(1020, 'Leo Carver', 'leo.carver@pharmtech.com', '646-555-6668', 'Pharmaceutical technician engaged in medication therapy management and patient counseling.')

INSERT INTO Jobs (JobID, Position, Title, Type, Medium, StartDate, JobDescription) VALUES
```

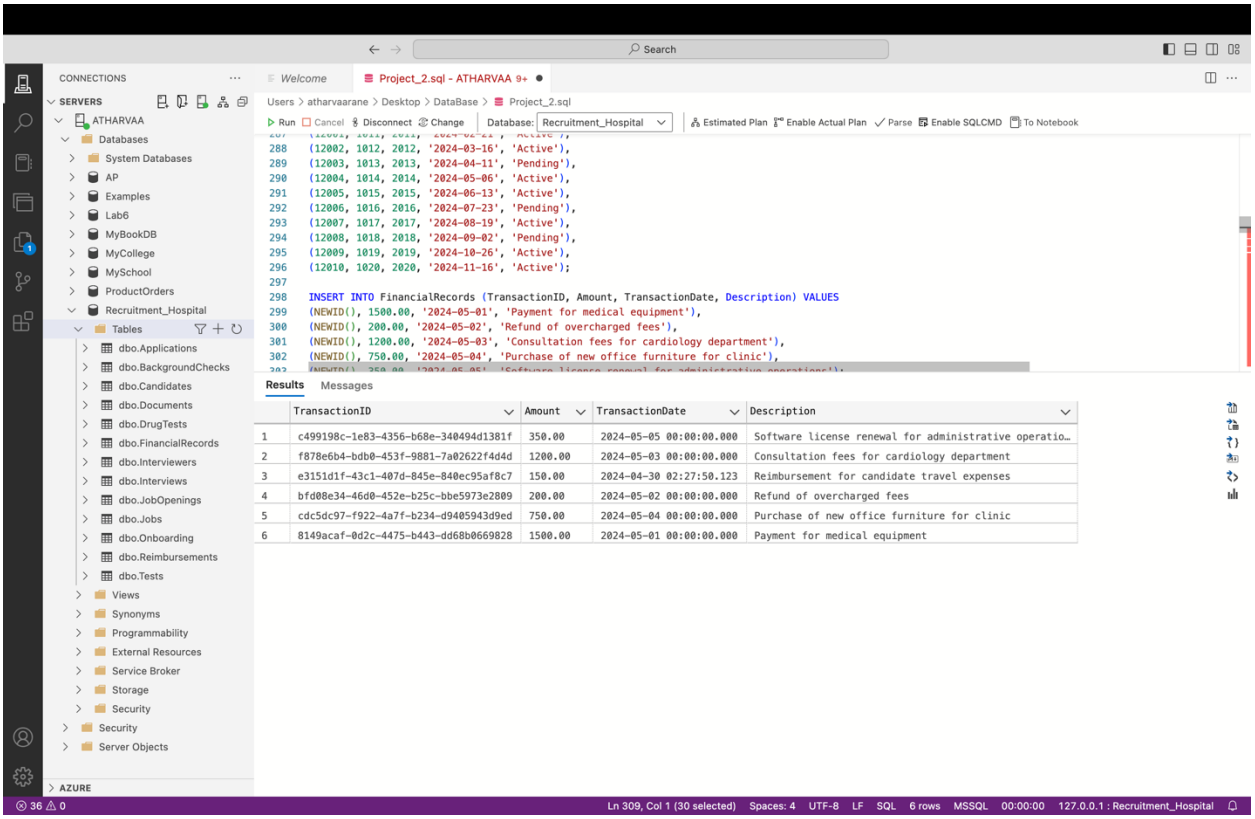
	CandidateID	Name	Email	Phone	ShortProfile
1	1011	Atharvaa Rane	atharvaa.rane@syr.edu	315-555-2974	Radiologist with a special interest in musculoskelet…
2	1012	Nolan Case	nolan.case@healthnet.com	716-555-4521	Epidemiologist with a strong background in infectiou…
3	1013	Lydia Bennett	lydia.bennett@clinic.org	607-555-9832	Dermatologist focusing on pediatric and adolescent s…
4	1014	Reed Morales	reed.morales@pioneerhealth.com	845-555-7391	Biochemist with expertise in pharmaceutical developm…
5	1015	Eva Summers	eva.summers@welllife.com	518-555-8264	Nutritionist specializing in dietetics for chronic d…
6	1016	Hugo Ross	hugo.ross@medicalcore.com	914-555-4930	General practitioner with a comprehensive approach t…
7	1017	Vivian Lacroix	vivian.lacroix@integrahealth.com	347-555-1824	A seasoned physical therapist dedicated to rehabilit…
8	1018	Miles Archer	miles.archer@lifeline.com	631-555-5010	Clinical research coordinator focusing on trials for…
9	1019	Tara Flynn	tara.flynn@careplus.org	212-555-2213	Anesthesiologist with a focus on perioperative care …
10	1020	Leo Carver	leo.carver@pharmtech.com	646-555-6668	Pharmaceutical technician engaged in medication ther…

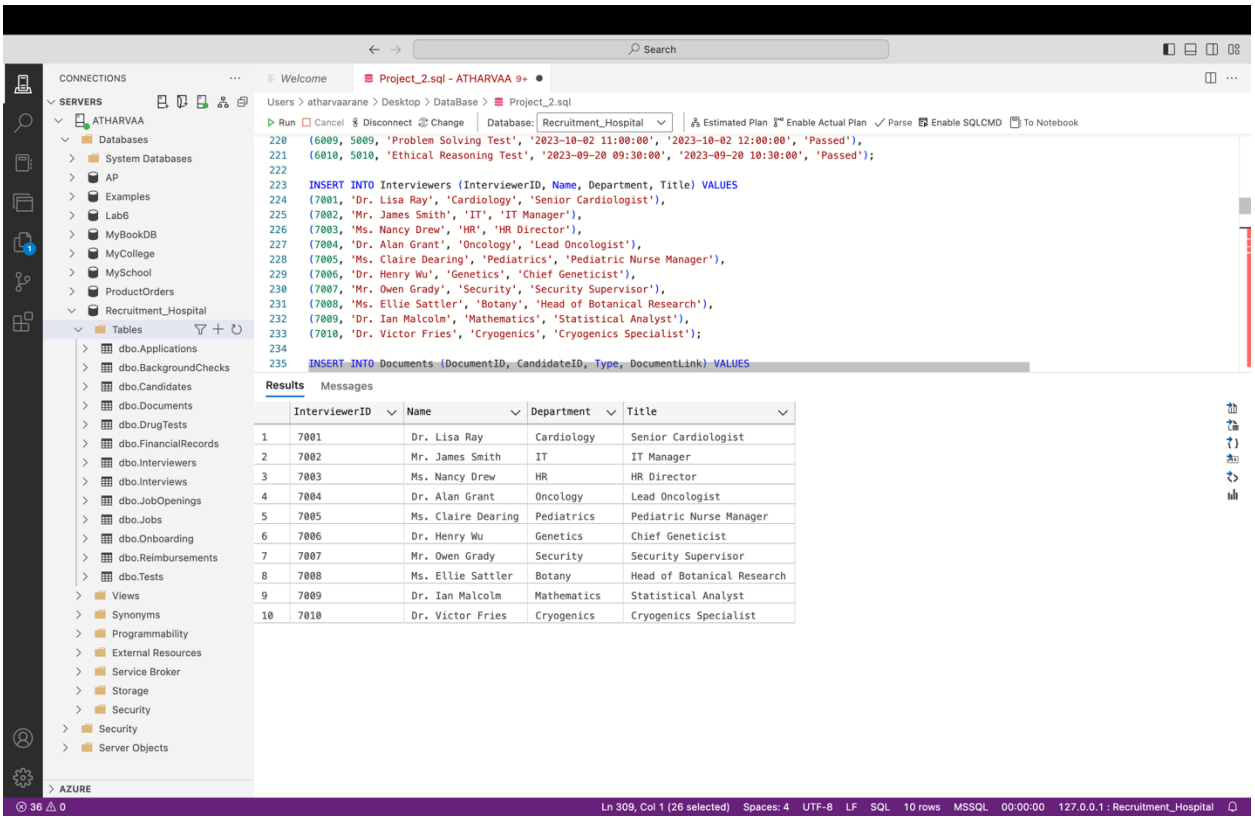


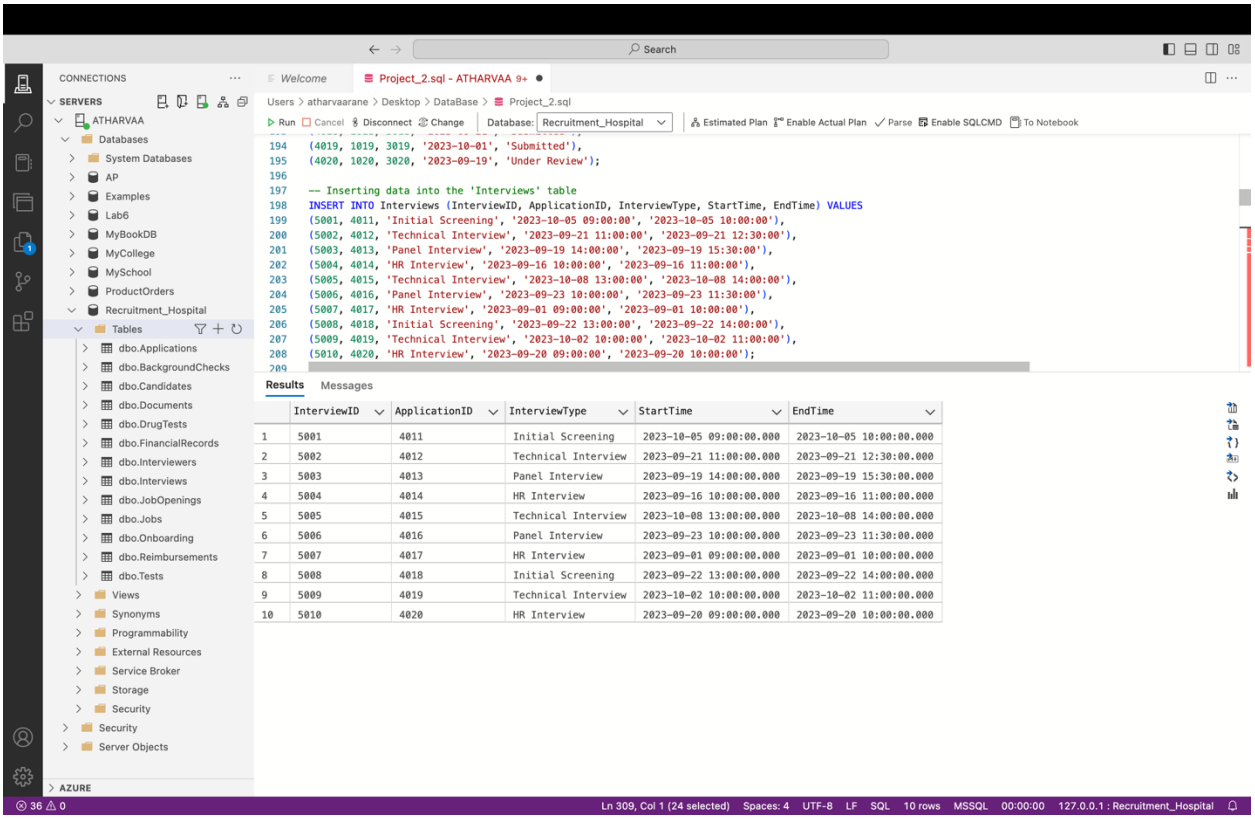
```
(7009, 'Dr. Ian Malcolm', 'Mathematics', 'Statistical Analyst'),
(7010, 'Dr. Victor Fries', 'Cryogenics', 'Cryogenics Specialist');

INSERT INTO Documents (DocumentID, CandidateID, Type, DocumentLink) VALUES
(8001, 1011, 'Resume', 'https://docs.example.com/resumes/1011'),
(8002, 1012, 'Cover Letter', 'https://docs.example.com/coverletters/1012'),
(8003, 1013, 'Portfolio', 'https://docs.example.com/portfolios/1013'),
(8004, 1014, 'Resume', 'https://docs.example.com/resumes/1014'),
(8005, 1015, 'Cover Letter', 'https://docs.example.com/coverletters/1015'),
(8006, 1016, 'Portfolio', 'https://docs.example.com/portfolios/1016'),
(8007, 1017, 'Resume', 'https://docs.example.com/resumes/1017'),
(8008, 1018, 'Cover Letter', 'https://docs.example.com/coverletters/1018'),
(8009, 1019, 'Portfolio', 'https://docs.example.com/portfolios/1019'),
(8010, 1020, 'Resume', 'https://docs.example.com/resumes/1020');
```

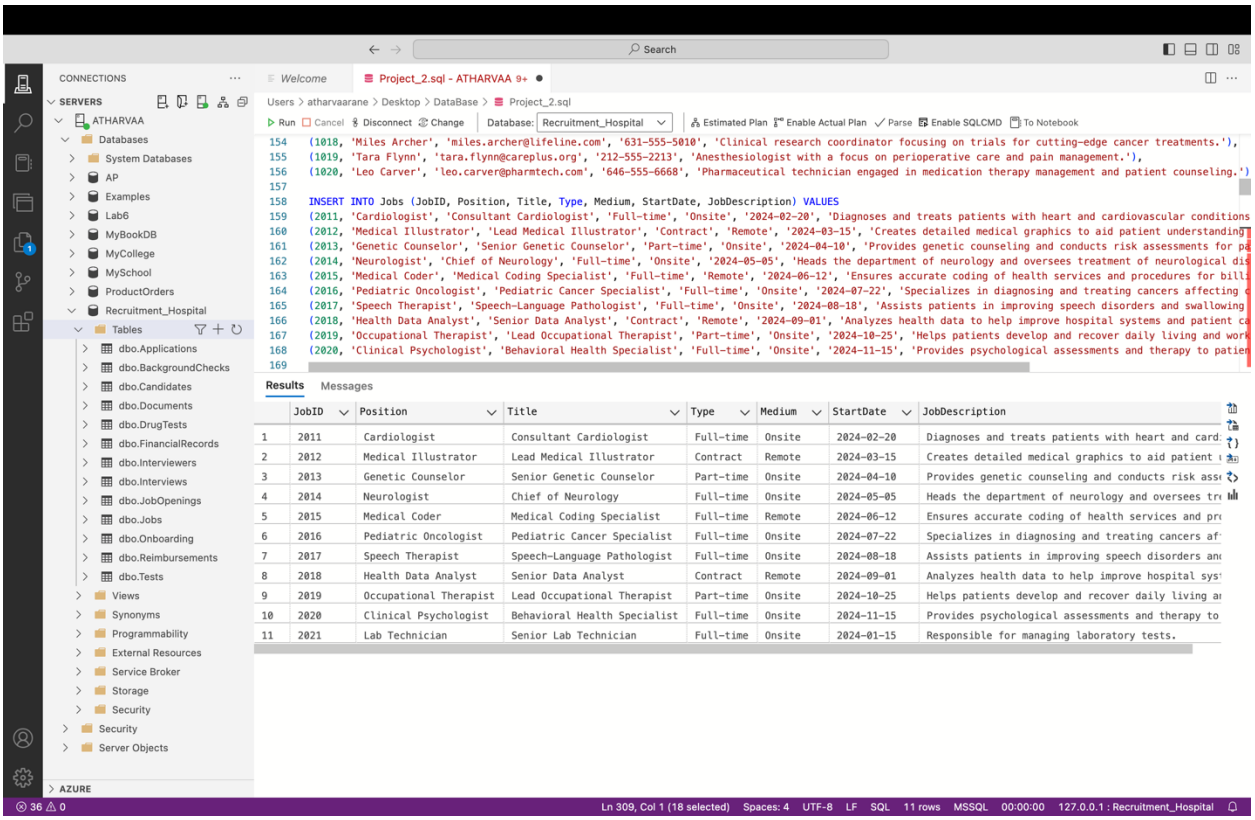
	DocumentID	CandidateID	Type	DocumentLink
1	8001	1011	Resume	https://docs.example.com/resumes/1011
2	8002	1012	Cover Letter	https://docs.example.com/coverletters/1012
3	8003	1013	Portfolio	https://docs.example.com/portfolios/1013
4	8004	1014	Resume	https://docs.example.com/resumes/1014
5	8005	1015	Cover Letter	https://docs.example.com/coverletters/1015
6	8006	1016	Portfolio	https://docs.example.com/portfolios/1016
7	8007	1017	Resume	https://docs.example.com/resumes/1017
8	8008	1018	Cover Letter	https://docs.example.com/coverletters/1018
9	8009	1019	Portfolio	https://docs.example.com/portfolios/1019
10	8010	1020	Resume	https://docs.example.com/resumes/1020

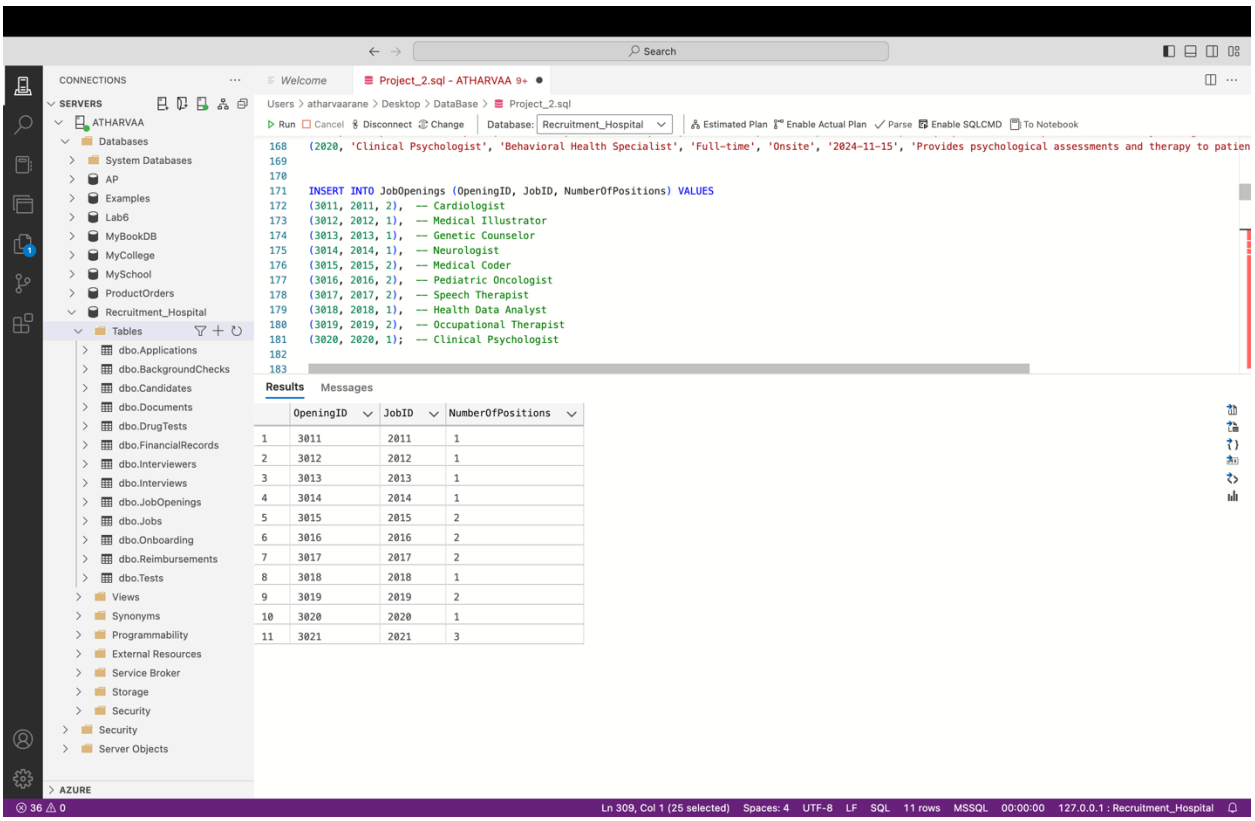

Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
INSERT INTO DrugTests (TestID, CandidateID, TestType, TestDate, Results) VALUES
(10001, 1011, 'Urine Test', '2023-09-30', 'Negative'),
(10002, 1012, 'Urine Test', '2023-10-01', 'Negative'),
(10003, 1013, 'Urine Test', '2023-10-02', 'Positive'),
(10004, 1014, 'Urine Test', '2023-10-03', 'Negative'),
(10005, 1015, 'Urine Test', '2023-10-04', 'Negative'),
(10006, 1016, 'Urine Test', '2023-10-05', 'Negative'),
(10007, 1017, 'Urine Test', '2023-10-06', 'Negative'),
(10008, 1018, 'Urine Test', '2023-10-07', 'Negative'),
(10009, 1019, 'Urine Test', '2023-10-08', 'Negative'),
(10010, 1020, 'Urine Test', '2023-10-09', 'Negative');
Results
Messages
TestID
CandidateID
TestType
TestDate
Results
1 10001 1011 Urine Test 2023-09-30 Negative
2 10002 1012 Urine Test 2023-10-01 Negative
3 10003 1013 Urine Test 2023-10-02 Positive
4 10004 1014 Urine Test 2023-10-03 Negative
5 10005 1015 Urine Test 2023-10-04 Negative
6 10006 1016 Urine Test 2023-10-05 Negative
7 10007 1017 Urine Test 2023-10-06 Negative
8 10008 1018 Urine Test 2023-10-07 Negative
9 10009 1019 Urine Test 2023-10-08 Negative
10 10010 1020 Urine Test 2023-10-09 Negative
36 0
Ln 302, Col 1 (23 selected)
Spaces: 4
UTF-8
LF
SQL
10 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital

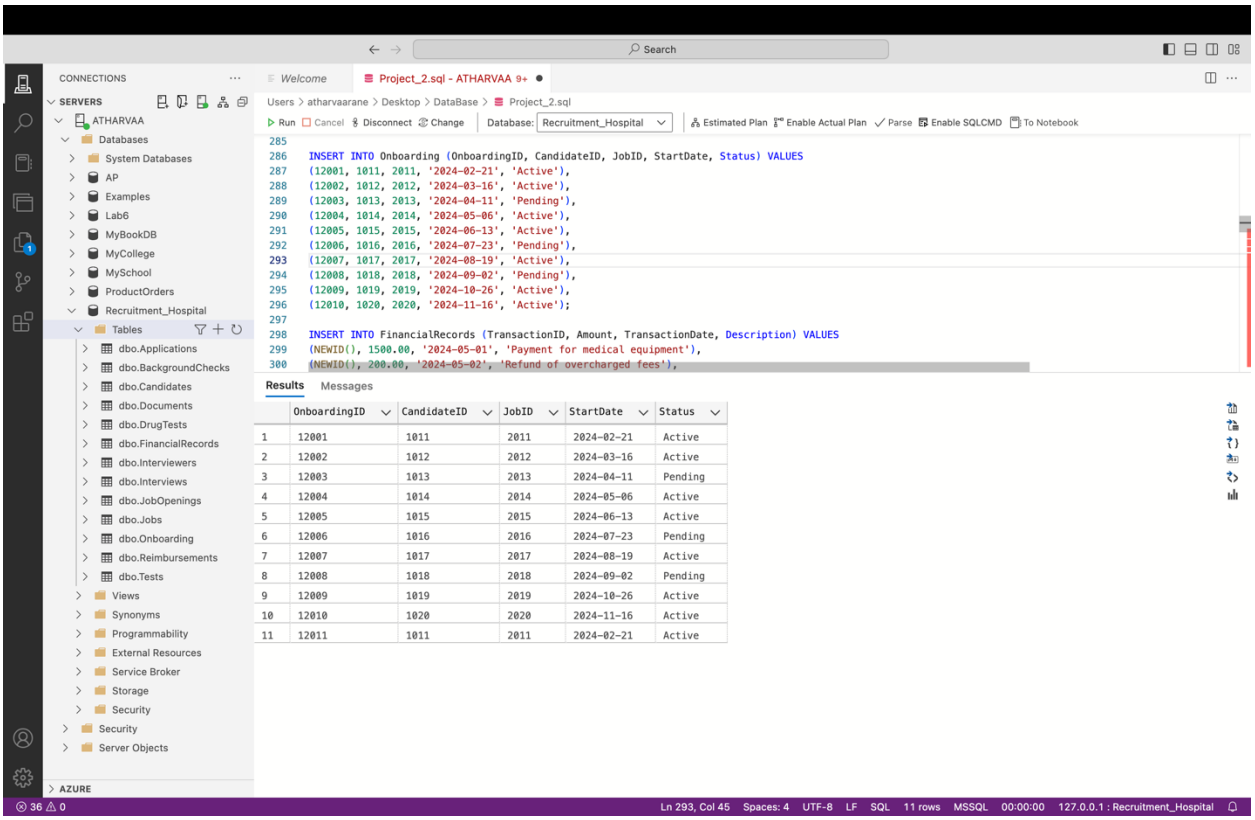

Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
(12002, 1012, 2012, '2024-03-16', 'Active'),
(12003, 1013, 2013, '2024-04-11', 'Pending'),
(12004, 1014, 2014, '2024-05-06', 'Active'),
(12005, 1015, 2015, '2024-06-13', 'Active'),
(12006, 1016, 2016, '2024-07-23', 'Pending'),
(12007, 1017, 2017, '2024-08-19', 'Active'),
(12008, 1018, 2018, '2024-09-02', 'Pending'),
(12009, 1019, 2019, '2024-10-26', 'Active'),
(12010, 1020, 2020, '2024-11-16', 'Active');
INSERT INTO FinancialRecords (TransactionID, Amount, TransactionDate, Description) VALUES
(NEWID(), 1500.00, '2024-05-01', 'Payment for medical equipment'),
(NEWID(), 200.00, '2024-05-02', 'Refund of overcharged fees'),
(NEWID(), 1200.00, '2024-05-03', 'Consultation fees for cardiology department'),
(NEWID(), 750.00, '2024-05-04', 'Purchase of new office furniture for clinic'),
Results
Messages
TransactionID
Amount
TransactionDate
Description
1 c499198c-1e83-4356-b68e-340494d1381f 350.00 2024-05-05 00:00:00.000 Software license renewal for administrative operatio…
2 f878e6b4-bdb0-453f-9881-7a02622f4d4d 1200.00 2024-05-03 00:00:00.000 Consultation fees for cardiology department
3 e3151d1f-43c1-407d-845e-840ec95af8c7 150.00 2024-04-30 02:27:50.123 Reimbursement for candidate travel expenses
4 bfd08e34-46d0-452e-b25c-bbe5973e2809 200.00 2024-05-02 00:00:00.000 Refund of overcharged fees
5 cdc5dc97-f922-4a7f-b234-d9405943d9ed 750.00 2024-05-04 00:00:00.000 Purchase of new office furniture for clinic
6 8149acaf-0d2c-4475-b443-dd68b0669828 1500.00 2024-05-01 00:00:00.000 Payment for medical equipment
36 0
Ln 309, Col 1 (30 selected)
Spaces: 4
UTF-8
LF
SQL
6 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital

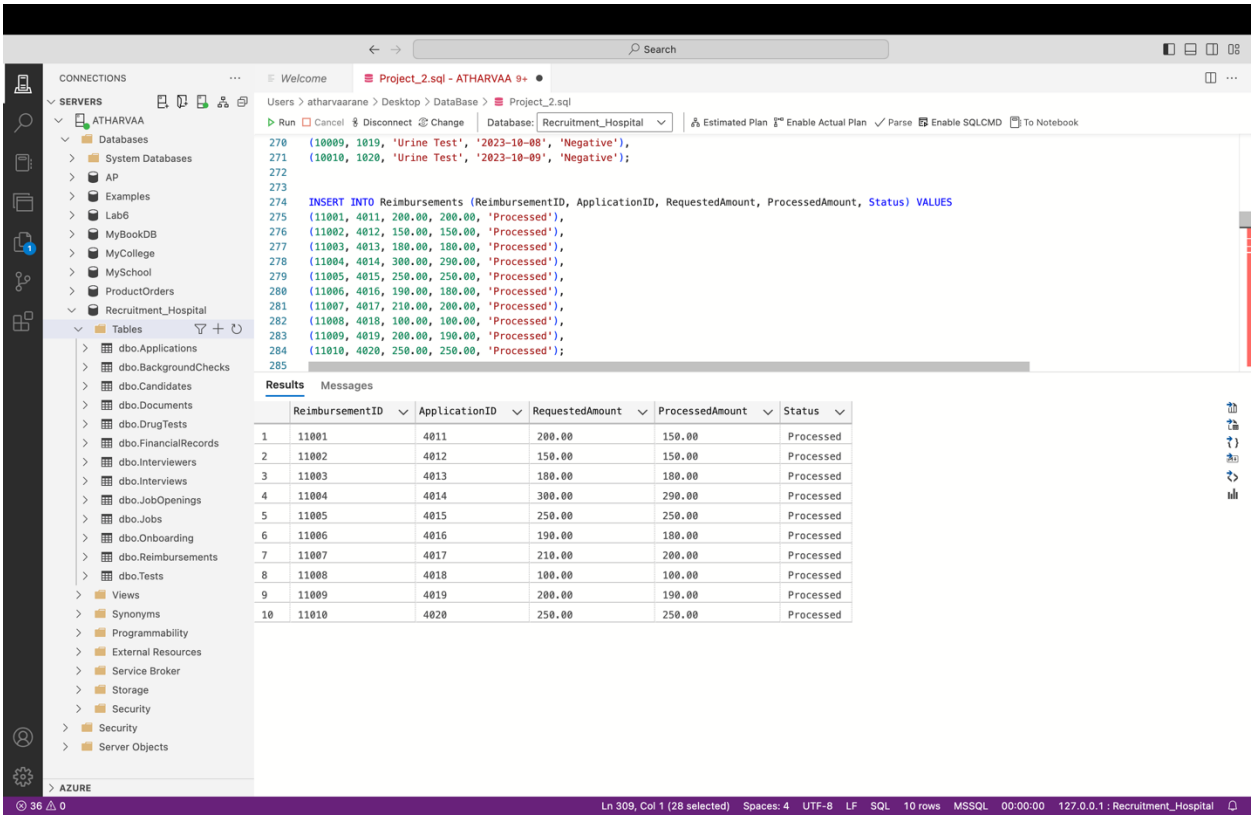

Project_2.sql - ATHARVAA
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Database: Recruitment_Hospital
(6009, 5009, 'Problem Solving Test', '2023-10-02 11:00:00', '2023-10-02 12:00:00', 'Passed'),
(6010, 5010, 'Ethical Reasoning Test', '2023-09-20 09:30:00', '2023-09-20 10:30:00', 'Passed');
INSERT INTO Interviewers (InterviewerID, Name, Department, Title) VALUES
(7001, 'Dr. Lisa Ray', 'Cardiology', 'Senior Cardiologist'),
(7002, 'Mr. James Smith', 'IT', 'IT Manager'),
(7003, 'Ms. Nancy Drew', 'HR', 'HR Director'),
(7004, 'Dr. Alan Grant', 'Oncology', 'Lead Oncologist'),
(7005, 'Ms. Claire Dearing', 'Pediatrics', 'Pediatric Nurse Manager'),
(7006, 'Dr. Henry Wu', 'Genetics', 'Chief Geneticist'),
(7007, 'Mr. Owen Grady', 'Security', 'Security Supervisor'),
(7008, 'Ms. Ellie Sattler', 'Botany', 'Head of Botanical Research'),
(7009, 'Dr. Ian Malcolm', 'Mathematics', 'Statistical Analyst'),
(7010, 'Dr. Victor Fries', 'Cryogenics', 'Cryogenics Specialist');
INSERT INTO Documents (DocumentID, CandidateID, Type, DocumentLink) VALUES
Results
Messages
InterviewerID | Name | Department | Title
1 | 7001 | Dr. Lisa Ray | Cardiology | Senior Cardiologist
2 | 7002 | Mr. James Smith | IT | IT Manager
3 | 7003 | Ms. Nancy Drew | HR | HR Director
4 | 7004 | Dr. Alan Grant | Oncology | Lead Oncologist
5 | 7005 | Ms. Claire Dearing | Pediatrics | Pediatric Nurse Manager
6 | 7006 | Dr. Henry Wu | Genetics | Chief Geneticist
7 | 7007 | Mr. Owen Grady | Security | Security Supervisor
8 | 7008 | Ms. Ellie Sattler | Botany | Head of Botanical Research
9 | 7009 | Dr. Ian Malcolm | Mathematics | Statistical Analyst
10 | 7010 | Dr. Victor Fries | Cryogenics | Cryogenics Specialist

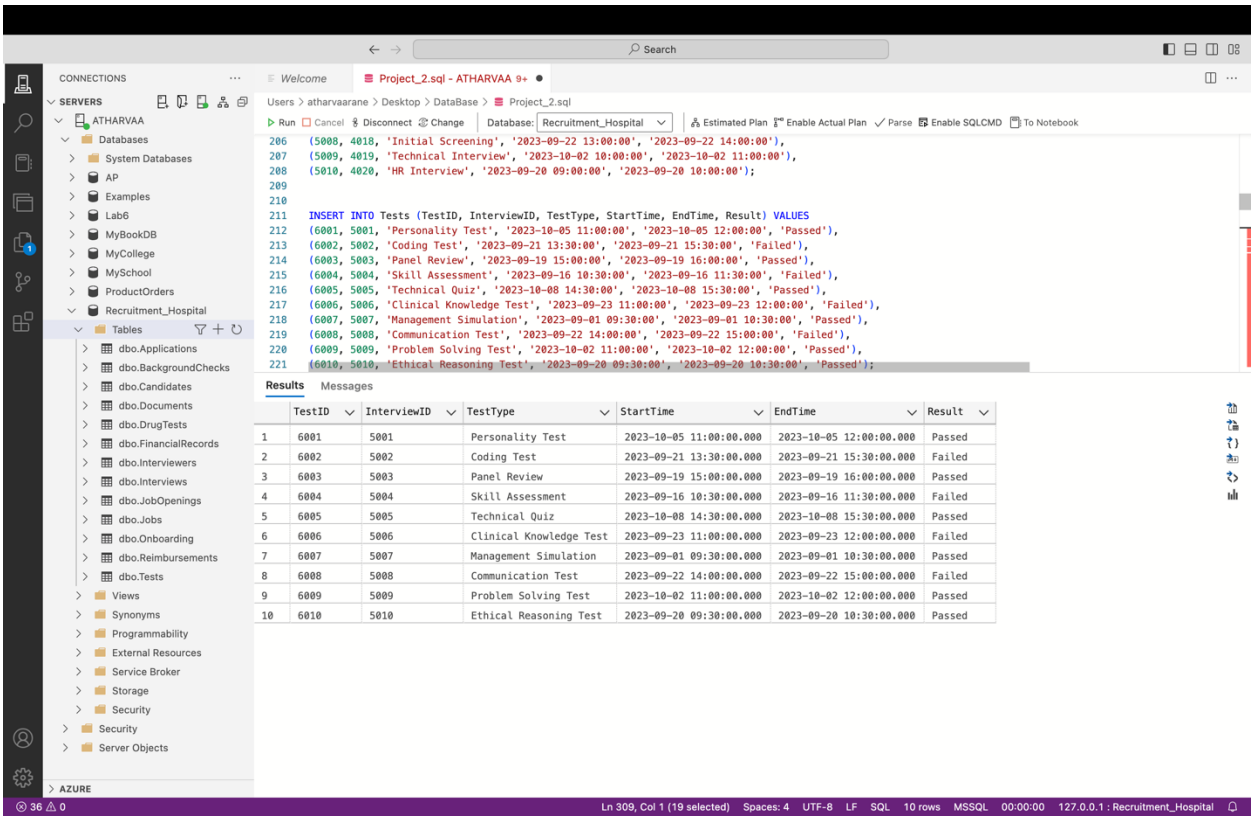

Project_2.sql - ATHARVAA
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Database: Recruitment_Hospital
(4019, 1019, 3019, '2023-10-01', 'Submitted'),
(4020, 1020, 3020, '2023-09-19', 'Under Review');
-- Inserting data into the 'Interviews' table
INSERT INTO Interviews (InterviewID, ApplicationID, InterviewType, StartTime, EndTime) VALUES
(5001, 4011, 'Initial Screening', '2023-10-05 09:00:00', '2023-10-05 10:00:00'),
(5002, 4012, 'Technical Interview', '2023-09-21 11:00:00', '2023-09-21 12:30:00'),
(5003, 4013, 'Panel Interview', '2023-09-19 14:00:00', '2023-09-19 15:30:00'),
(5004, 4014, 'HR Interview', '2023-09-16 10:00:00', '2023-09-16 11:00:00'),
(5005, 4015, 'Technical Interview', '2023-10-08 13:00:00', '2023-10-08 14:00:00'),
(5006, 4016, 'Panel Interview', '2023-09-23 10:00:00', '2023-09-23 11:30:00'),
(5007, 4017, 'HR Interview', '2023-09-01 09:00:00', '2023-09-01 10:00:00'),
(5008, 4018, 'Initial Screening', '2023-09-22 13:00:00', '2023-09-22 14:00:00'),
(5009, 4019, 'Technical Interview', '2023-10-02 10:00:00', '2023-10-02 11:00:00'),
(5010, 4020, 'HR Interview', '2023-09-20 09:00:00', '2023-09-20 10:00:00');
Results
Messages
InterviewID | ApplicationID | InterviewType | StartTime | EndTime
1 | 5001 | 4011 | Initial Screening | 2023-10-05 09:00:00.000 | 2023-10-05 10:00:00.000
2 | 5002 | 4012 | Technical Interview | 2023-09-21 11:00:00.000 | 2023-09-21 12:30:00.000
3 | 5003 | 4013 | Panel Interview | 2023-09-19 14:00:00.000 | 2023-09-19 15:30:00.000
4 | 5004 | 4014 | HR Interview | 2023-09-16 10:00:00.000 | 2023-09-16 11:00:00.000
5 | 5005 | 4015 | Technical Interview | 2023-10-08 13:00:00.000 | 2023-10-08 14:00:00.000
6 | 5006 | 4016 | Panel Interview | 2023-09-23 10:00:00.000 | 2023-09-23 11:30:00.000
7 | 5007 | 4017 | HR Interview | 2023-09-01 09:00:00.000 | 2023-09-01 10:00:00.000
8 | 5008 | 4018 | Initial Screening | 2023-09-22 13:00:00.000 | 2023-09-22 14:00:00.000
9 | 5009 | 4019 | Technical Interview | 2023-10-02 10:00:00.000 | 2023-10-02 11:00:00.000
10 | 5010 | 4020 | HR Interview | 2023-09-20 09:00:00.000 | 2023-09-20 10:00:00.000


Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
154 (1018, 'Miles Archer', 'miles.archer@lifeline.com', '631-555-5010', 'Clinical research coordinator focusing on trials for cutting-edge cancer treatments.'),
155 (1019, 'Tara Flynn', 'tara.flynn@careplus.org', '212-555-2213', 'Anesthesiologist with a focus on perioperative care and pain management.'),
156 (1020, 'Leo Carver', 'leo.carver@pharmtech.com', '646-555-6668', 'Pharmaceutical technician engaged in medication therapy management and patient counseling.')
157
158 INSERT INTO Jobs (JobID, Position, Title, Type, Medium, StartDate, JobDescription) VALUES
159 (2011, 'Cardiologist', 'Consultant Cardiologist', 'Full-time', 'Onsite', '2024-02-20', 'Diagnoses and treats patients with heart and cardiovascular conditions
160 (2012, 'Medical Illustrator', 'Lead Medical Illustrator', 'Contract', 'Remote', '2024-03-15', 'Creates detailed medical graphics to aid patient understanding
161 (2013, 'Genetic Counselor', 'Senior Genetic Counselor', 'Part-time', 'Onsite', '2024-04-10', 'Provides genetic counseling and conducts risk assessments for pa
162 (2014, 'Neurologist', 'Chief of Neurology', 'Full-time', 'Onsite', '2024-05-05', 'Heads the department of neurology and oversees treatment of neurological dis
163 (2015, 'Medical Coder', 'Medical Coding Specialist', 'Full-time', 'Remote', '2024-06-12', 'Ensures accurate coding of health services and procedures for billi
164 (2016, 'Pediatric Oncologist', 'Pediatric Cancer Specialist', 'Full-time', 'Onsite', '2024-07-22', 'Specializes in diagnosing and treating cancers affecting c
165 (2017, 'Speech Therapist', 'Speech-Language Pathologist', 'Full-time', 'Onsite', '2024-08-18', 'Assists patients in improving speech disorders and swallowing
166 (2018, 'Health Data Analyst', 'Senior Data Analyst', 'Contract', 'Remote', '2024-09-01', 'Analyzes health data to help improve hospital systems and patient ca
167 (2019, 'Occupational Therapist', 'Lead Occupational Therapist', 'Part-time', 'Onsite', '2024-10-25', 'Helps patients develop and recover daily living and work
168 (2020, 'Clinical Psychologist', 'Behavioral Health Specialist', 'Full-time', 'Onsite', '2024-11-15', 'Provides psychological assessments and therapy to patien
169
Results
Messages
JobID | Position | Title | Type | Medium | StartDate | JobDescription
1 | 2011 | Cardiologist | Consultant Cardiologist | Full-time | Onsite | 2024-02-20 | Diagnoses and treats patients with heart and card
2 | 2012 | Medical Illustrator | Lead Medical Illustrator | Contract | Remote | 2024-03-15 | Creates detailed medical graphics to aid patient
3 | 2013 | Genetic Counselor | Senior Genetic Counselor | Part-time | Onsite | 2024-04-10 | Provides genetic counseling and conducts risk ass
4 | 2014 | Neurologist | Chief of Neurology | Full-time | Onsite | 2024-05-05 | Heads the department of neurology and oversees tr
5 | 2015 | Medical Coder | Medical Coding Specialist | Full-time | Remote | 2024-06-12 | Ensures accurate coding of health services and pr
6 | 2016 | Pediatric Oncologist | Pediatric Cancer Specialist | Full-time | Onsite | 2024-07-22 | Specializes in diagnosing and treating cancers af
7 | 2017 | Speech Therapist | Speech-Language Pathologist | Full-time | Onsite | 2024-08-18 | Assists patients in improving speech disorders an
8 | 2018 | Health Data Analyst | Senior Data Analyst | Contract | Remote | 2024-09-01 | Analyzes health data to help improve hospital sys
9 | 2019 | Occupational Therapist | Lead Occupational Therapist | Part-time | Onsite | 2024-10-25 | Helps patients develop and recover daily living a
10 | 2020 | Clinical Psychologist | Behavioral Health Specialist | Full-time | Onsite | 2024-11-15 | Provides psychological assessments and therapy to
11 | 2021 | Lab Technician | Senior Lab Technician | Full-time | Onsite | 2024-01-15 | Responsible for managing laboratory tests.
36 0
Ln 309, Col 1 (18 selected)
Spaces: 4
UTF-8
LF
SQL
11 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital


Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
168 (2020, 'Clinical Psychologist', 'Behavioral Health Specialist', 'Full-time', 'Onsite', '2024-11-15', 'Provides psychological assessments and therapy to patien
169
170
171 INSERT INTO JobOpenings (OpeningID, JobID, NumberOfPositions) VALUES
172 (3011, 2011, 2), -- Cardiologist
173 (3012, 2012, 1), -- Medical Illustrator
174 (3013, 2013, 1), -- Genetic Counselor
175 (3014, 2014, 1), -- Neurologist
176 (3015, 2015, 2), -- Medical Coder
177 (3016, 2016, 2), -- Pediatric Oncologist
178 (3017, 2017, 2), -- Speech Therapist
179 (3018, 2018, 1), -- Health Data Analyst
180 (3019, 2019, 2), -- Occupational Therapist
181 (3020, 2020, 1); -- Clinical Psychologist
182
183
Results
Messages
OpeningID | JobID | NumberOfPositions
1 | 3011 | 2011 | 1
2 | 3012 | 2012 | 1
3 | 3013 | 2013 | 1
4 | 3014 | 2014 | 1
5 | 3015 | 2015 | 2
6 | 3016 | 2016 | 2
7 | 3017 | 2017 | 2
8 | 3018 | 2018 | 1
9 | 3019 | 2019 | 2
10 | 3020 | 2020 | 1
11 | 3021 | 2021 | 3
36 0
Ln 309, Col 1 (25 selected)
Spaces: 4
UTF-8
LF
SQL
11 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital


Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
285
286 INSERT INTO Onboarding (OnboardingID, CandidateID, JobID, StartDate, Status) VALUES
287 (12001, 1011, 2011, '2024-02-21', 'Active'),
288 (12002, 1012, 2012, '2024-03-16', 'Active'),
289 (12003, 1013, 2013, '2024-04-11', 'Pending'),
290 (12004, 1014, 2014, '2024-05-06', 'Active'),
291 (12005, 1015, 2015, '2024-06-13', 'Active'),
292 (12006, 1016, 2016, '2024-07-23', 'Pending'),
293 (12007, 1017, 2017, '2024-08-19', 'Active'),
294 (12008, 1018, 2018, '2024-09-02', 'Pending'),
295 (12009, 1019, 2019, '2024-10-26', 'Active'),
296 (12010, 1020, 2020, '2024-11-16', 'Active');
297
298 INSERT INTO FinancialRecords (TransactionID, Amount, TransactionDate, Description) VALUES
299 (NEWID(), 1500.00, '2024-05-01', 'Payment for medical equipment'),
300 (NEWID(), 200.00, '2024-05-02', 'Refund of overcharged fees');
Results
Messages
OnboardingID CandidateID JobID StartDate Status
1 12001 1011 2011 2024-02-21 Active
2 12002 1012 2012 2024-03-16 Active
3 12003 1013 2013 2024-04-11 Pending
4 12004 1014 2014 2024-05-06 Active
5 12005 1015 2015 2024-06-13 Active
6 12006 1016 2016 2024-07-23 Pending
7 12007 1017 2017 2024-08-19 Active
8 12008 1018 2018 2024-09-02 Pending
9 12009 1019 2019 2024-10-26 Active
10 12010 1020 2020 2024-11-16 Active
11 12011 1011 2011 2024-02-21 Active
36 0
Ln 293, Col 45
Spaces: 4
UTF-8
LF
SQL
11 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital


Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
270 (10009, 1019, 'Urine Test', '2023-10-08', 'Negative'),
271 (10010, 1020, 'Urine Test', '2023-10-09', 'Negative');
272
273
274 INSERT INTO Reimbursements (ReimbursementID, ApplicationID, RequestedAmount, ProcessedAmount, Status) VALUES
275 (11001, 4011, 200.00, 200.00, 'Processed'),
276 (11002, 4012, 150.00, 150.00, 'Processed'),
277 (11003, 4013, 180.00, 180.00, 'Processed'),
278 (11004, 4014, 300.00, 290.00, 'Processed'),
279 (11005, 4015, 250.00, 250.00, 'Processed'),
280 (11006, 4016, 190.00, 180.00, 'Processed'),
281 (11007, 4017, 210.00, 200.00, 'Processed'),
282 (11008, 4018, 100.00, 100.00, 'Processed'),
283 (11009, 4019, 200.00, 190.00, 'Processed'),
284 (11010, 4020, 250.00, 250.00, 'Processed');
285
Results
Messages
ReimbursementID ApplicationID RequestedAmount ProcessedAmount Status
1 11001 4011 200.00 150.00 Processed
2 11002 4012 150.00 150.00 Processed
3 11003 4013 180.00 180.00 Processed
4 11004 4014 300.00 290.00 Processed
5 11005 4015 250.00 250.00 Processed
6 11006 4016 190.00 180.00 Processed
7 11007 4017 210.00 200.00 Processed
8 11008 4018 100.00 100.00 Processed
9 11009 4019 200.00 190.00 Processed
10 11010 4020 250.00 250.00 Processed
36 0
Ln 309, Col 1 (28 selected)
Spaces: 4
UTF-8
LF
SQL
10 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital

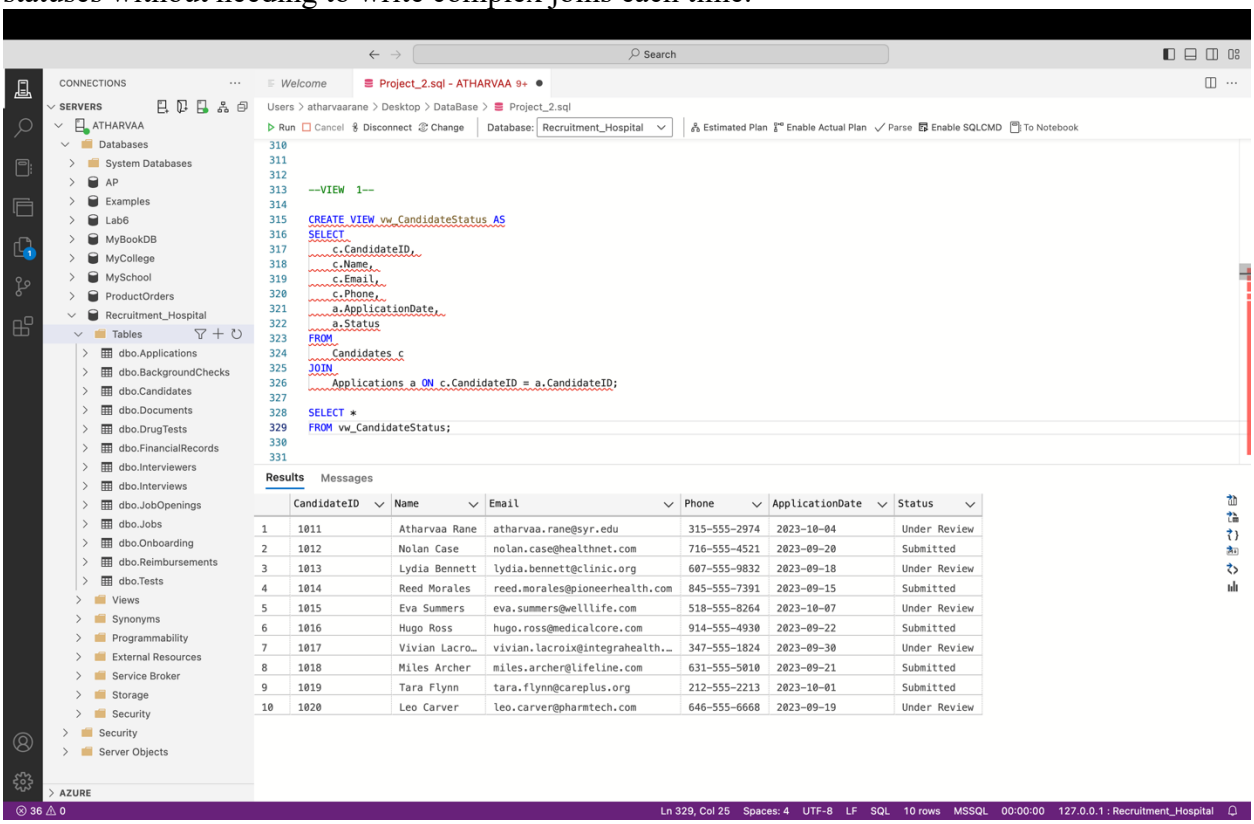

Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Server Objects
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
206 (5008, 4018, 'Initial Screening', '2023-09-22 13:00:00', '2023-09-22 14:00:00'),
207 (5009, 4019, 'Technical Interview', '2023-10-02 10:00:00', '2023-10-02 11:00:00'),
208 (5010, 4020, 'HR Interview', '2023-09-20 09:00:00', '2023-09-20 10:00:00');
209
210
211 INSERT INTO Tests (TestID, InterviewID, TestType, StartTime, EndTime, Result) VALUES
212 (6001, 5001, 'Personality Test', '2023-10-05 11:00:00', '2023-10-05 12:00:00', 'Passed'),
213 (6002, 5002, 'Coding Test', '2023-09-21 13:30:00', '2023-09-21 15:30:00', 'Failed'),
214 (6003, 5003, 'Panel Review', '2023-09-19 15:00:00', '2023-09-19 16:00:00', 'Passed'),
215 (6004, 5004, 'Skill Assessment', '2023-09-16 10:30:00', '2023-09-16 11:30:00', 'Failed'),
216 (6005, 5005, 'Technical Quiz', '2023-10-08 14:30:00', '2023-10-08 15:30:00', 'Passed'),
217 (6006, 5006, 'Clinical Knowledge Test', '2023-09-23 11:00:00', '2023-09-23 12:00:00', 'Failed'),
218 (6007, 5007, 'Management Simulation', '2023-09-01 09:30:00', '2023-09-01 10:30:00', 'Passed'),
219 (6008, 5008, 'Communication Test', '2023-09-22 14:00:00', '2023-09-22 15:00:00', 'Failed'),
220 (6009, 5009, 'Problem Solving Test', '2023-10-02 11:00:00', '2023-10-02 12:00:00', 'Passed'),
221 (6010, 5010, 'Ethical Reasoning Test', '2023-09-20 09:30:00', '2023-09-20 10:30:00', 'Passed');
Results
Messages
TestID
InterviewID
TestType
StartTime
EndTime
Result
1 6001 5001 Personality Test 2023-10-05 11:00:00.000 2023-10-05 12:00:00.000 Passed
2 6002 5002 Coding Test 2023-09-21 13:30:00.000 2023-09-21 15:30:00.000 Failed
3 6003 5003 Panel Review 2023-09-19 15:00:00.000 2023-09-19 16:00:00.000 Passed
4 6004 5004 Skill Assessment 2023-09-16 10:30:00.000 2023-09-16 11:30:00.000 Failed
5 6005 5005 Technical Quiz 2023-10-08 14:30:00.000 2023-10-08 15:30:00.000 Passed
6 6006 5006 Clinical Knowledge Test 2023-09-23 11:00:00.000 2023-09-23 12:00:00.000 Failed
7 6007 5007 Management Simulation 2023-09-01 09:30:00.000 2023-09-01 10:30:00.000 Passed
8 6008 5008 Communication Test 2023-09-22 14:00:00.000 2023-09-22 15:00:00.000 Failed
9 6009 5009 Problem Solving Test 2023-10-02 11:00:00.000 2023-10-02 12:00:00.000 Passed
10 6010 5010 Ethical Reasoning Test 2023-09-20 09:30:00.000 2023-09-20 10:30:00.000 Passed
36 0
Ln 309, Col 1 (19 selected)
Spaces: 4
UTF-8
LF
SQL
10 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital

View Commands

View1:

```
CREATE VIEW vw_CandidateStatus AS
SELECT
    c.CandidateID,
    c.Name,
    c.Email,
    c.Phone,
    a.ApplicationDate,
    a.Status
FROM
    Candidates c
JOIN
    Applications a ON c.CandidateID = a.CandidateID;

SELECT *
FROM vw_CandidateStatus;
```

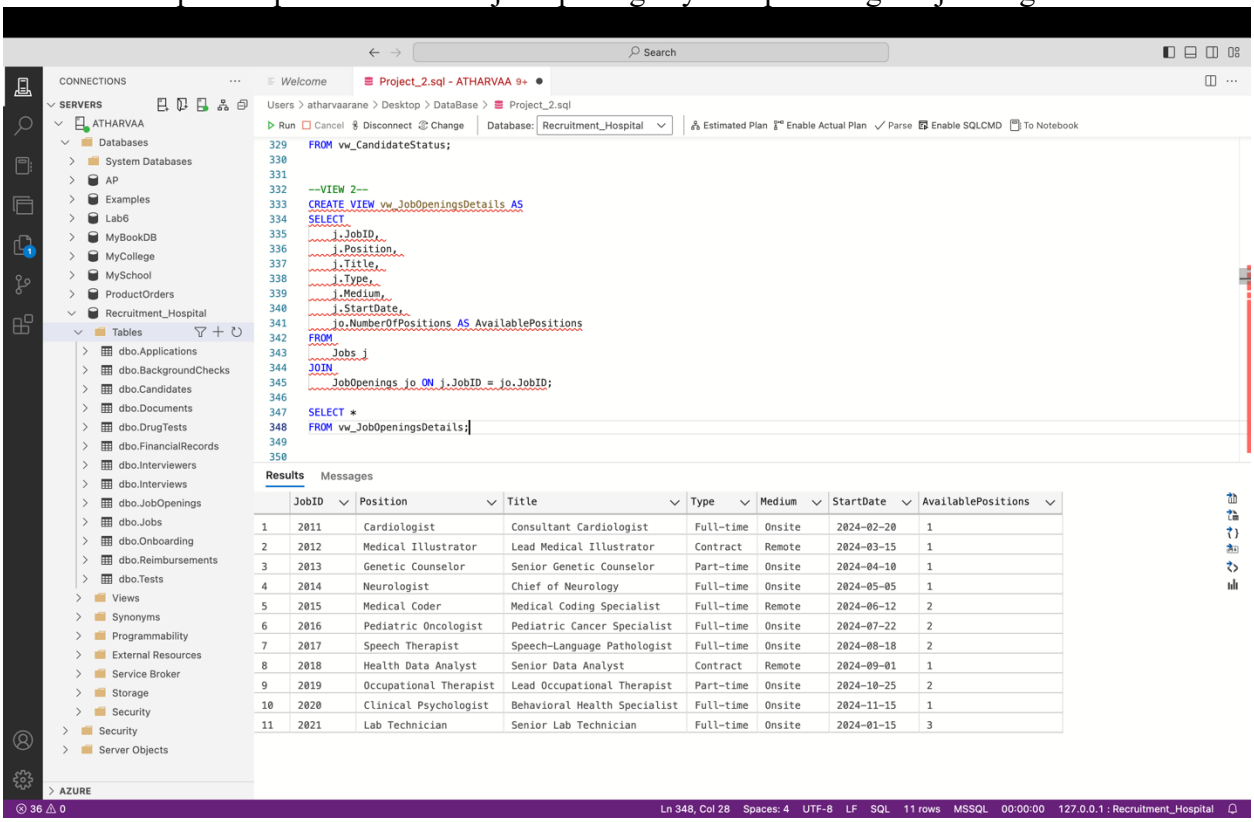
The SQL script creates a view named `vw_CandidateStatus`. This view consolidates data from two tables: `Candidates` and `Applications`. It joins these tables on the `CandidateID` to provide a combined dataset that includes each candidate's ID, name, email, phone number, application date, and application status. This view can be used to easily query and report on candidate statuses without needing to write complex joins each time.



View2:

```
CREATE VIEW vw_JobOpeningsDetails AS
SELECT
    j.JobID,
    j.Position,
    j.Title,
    j.Type,
    j.Medium,
    j.StartDate,
    jo.NumberOfPositions AS AvailablePositions
FROM
    Jobs j
JOIN
    JobOpenings jo ON j.JobID = jo.JobID;

SELECT *
FROM vw_JobOpeningsDetails;
```

The SQL script creates a view named `vw_JobOpeningsDetails`. This view is designed to provide detailed information about job openings by joining two tables: `Jobs` and `JobOpenings`. The view includes the job ID, position, title, type, medium, start date, and the number of available positions for each job. This facilitates easy access to comprehensive details about job openings, which can be particularly useful for HR departments or job listing platforms. The view simplifies queries related to job openings by encapsulating the join logic within it.

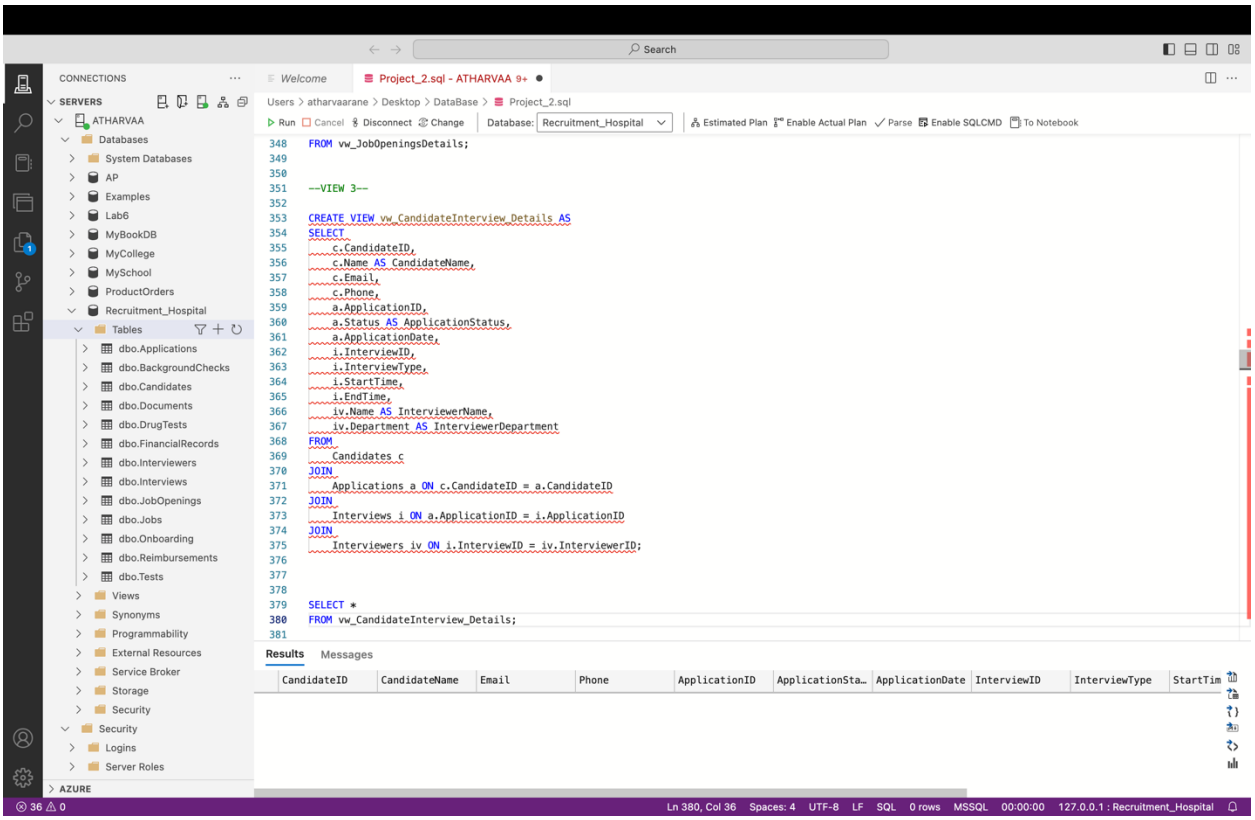


	JobID	Position	Title	Type	Medium	StartDate	AvailablePositions
1	2011	Cardiologist	Consultant Cardiologist	Full-time	Onsite	2024-02-20	1
2	2012	Medical Illustrator	Lead Medical Illustrator	Contract	Remote	2024-03-15	1
3	2013	Genetic Counselor	Senior Genetic Counselor	Part-time	Onsite	2024-04-10	1
4	2014	Neurologist	Chief of Neurology	Full-time	Onsite	2024-05-05	1
5	2015	Medical Coder	Medical Coding Specialist	Full-time	Remote	2024-06-12	2
6	2016	Pediatric Oncologist	Pediatric Cancer Specialist	Full-time	Onsite	2024-07-22	2
7	2017	Speech Therapist	Speech-Language Pathologist	Full-time	Onsite	2024-08-18	2
8	2018	Health Data Analyst	Senior Data Analyst	Contract	Remote	2024-09-01	1
9	2019	Occupational Therapist	Lead Occupational Therapist	Part-time	Onsite	2024-10-25	2
10	2020	Clinical Psychologist	Behavioral Health Specialist	Full-time	Onsite	2024-11-15	1
11	2021	Lab Technician	Senior Lab Technician	Full-time	Onsite	2024-01-15	3

View3:

```
CREATE VIEW vw_CandidateInterview_Details AS
SELECT
    c.CandidateID,
    c.Name AS CandidateName,
    c.Email,
    c.Phone,
    a.ApplicationID,
    a.Status AS ApplicationStatus,
    a.ApplicationDate,
    i.InterviewID,
    i.InterviewType,
    i.StartTime,
    i.EndTime,
    iv.Name AS InterviewerName,
    iv.Department AS InterviewerDepartment
FROM
    Candidates c
JOIN
    Applications a ON c.CandidateID = a.CandidateID
JOIN
    Interviews i ON a.ApplicationID = i.ApplicationID
JOIN
    Interviewers iv ON i.InterviewID = iv.InterviewerID;



SELECT *
FROM vw_CandidateInterview_Details;
```

The SQL script defines a view called `vw_CandidateInterview_Details`. This view integrates data from four different tables: `Candidates`, `Applications`, `Interviews`, and `Interviewers`. By joining these tables, the view provides a comprehensive snapshot of each candidate's interview details. The fields included are the candidate's ID, name, email, phone, application ID, status of the application, date of the application, interview ID, type of interview, start and end times of the interview, and the interviewer's name and department. This view is particularly useful for HR departments to quickly access and review all relevant details related to a candidate's interview process in a single query, streamlining the recruitment workflow.

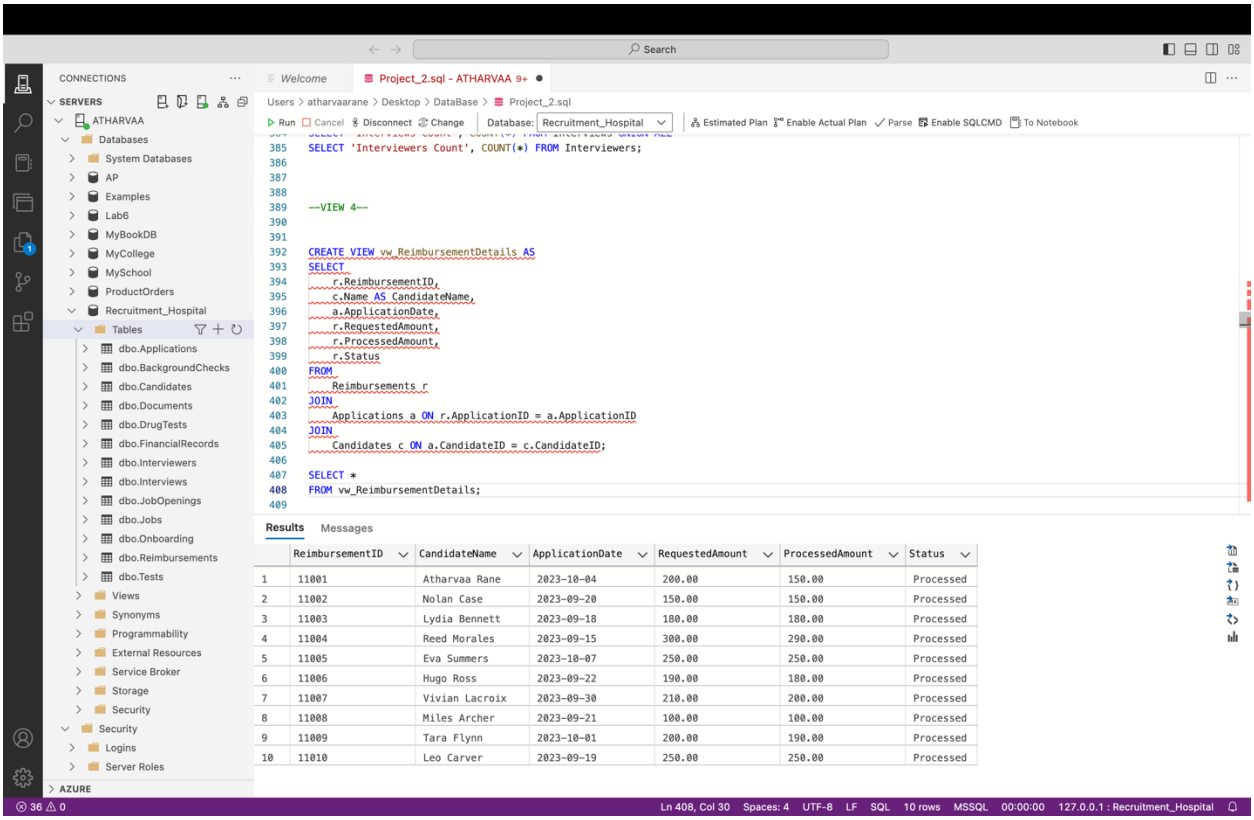


View4:

```
CREATE VIEW vw_ReimbursementDetails AS
SELECT
    r.ReimbursementID,
    c.Name AS CandidateName,
    a.ApplicationDate,
    r.RequestedAmount,
    r.ProcessedAmount,
    r.Status
FROM
    Reimbursements r
JOIN
    Applications a ON r.ApplicationID = a.ApplicationID
JOIN
    Candidates c ON a.CandidateID = c.CandidateID;

SELECT *
FROM vw_ReimbursementDetails;
```

The SQL script outlines the creation of a view called `vw_ReimbursementDetails`. This view is designed to consolidate data relevant to reimbursements within the context of job applications and candidates. By joining the `Reimbursements`, `Applications`, and `Candidates` tables, the

view efficiently gathers and presents key information including the reimbursement ID, the candidate's name, the application date, the amount requested for reimbursement, the amount that was processed, and the current status of the reimbursement. This view simplifies the process of tracking and analyzing reimbursement requests related to job applications, making it highly useful for financial and administrative personnel involved in managing such processes.



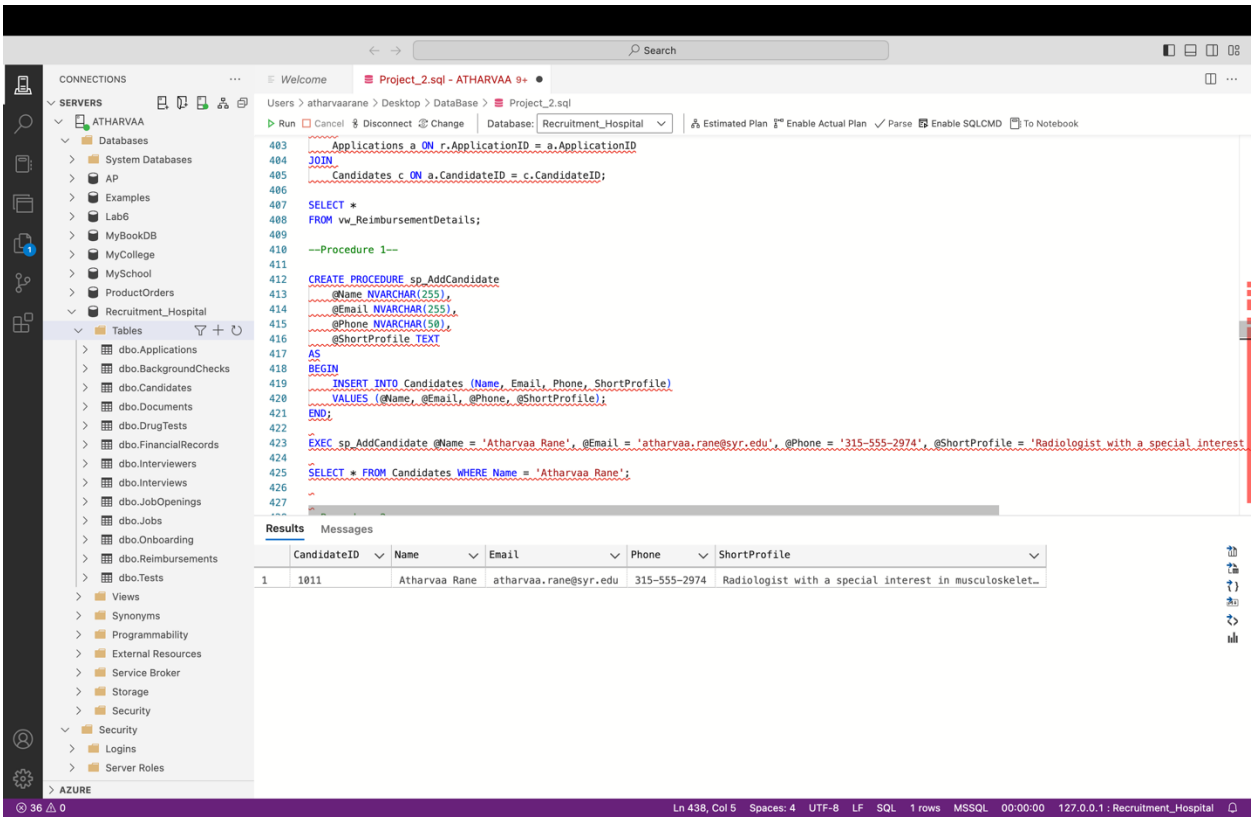
Procedure Commands

Procedure1:

```
CREATE PROCEDURE sp_AddCandidate
    @Name NVARCHAR(255),
    @Email NVARCHAR(255),
    @Phone NVARCHAR(50),
    @ShortProfile TEXT
AS
BEGIN
    INSERT INTO Candidates (Name, Email, Phone, ShortProfile)
    VALUES (@Name, @Email, @Phone, @ShortProfile);
END;

EXEC sp_AddCandidate @Name = 'Atharvaa Rane', @Email = 'atharvaa.rane@syr.edu', @Phone
= '315-555-2974', @ShortProfile = 'Radiologist with a special interest in
musculoskeletal imaging and sports injuries.';

SELECT * FROM Candidates WHERE Name = 'Atharvaa Rane';
```

The SQL script you provided defines a stored procedure named sp_AddCandidate for inserting a new candidate's details into the Candidates table. It takes four parameters—name, email, phone number, and a short profile—and inserts these into the database. The procedure is then executed with example data for a candidate named John Doe, showcasing how to add a record to the Candidates table using this procedure.

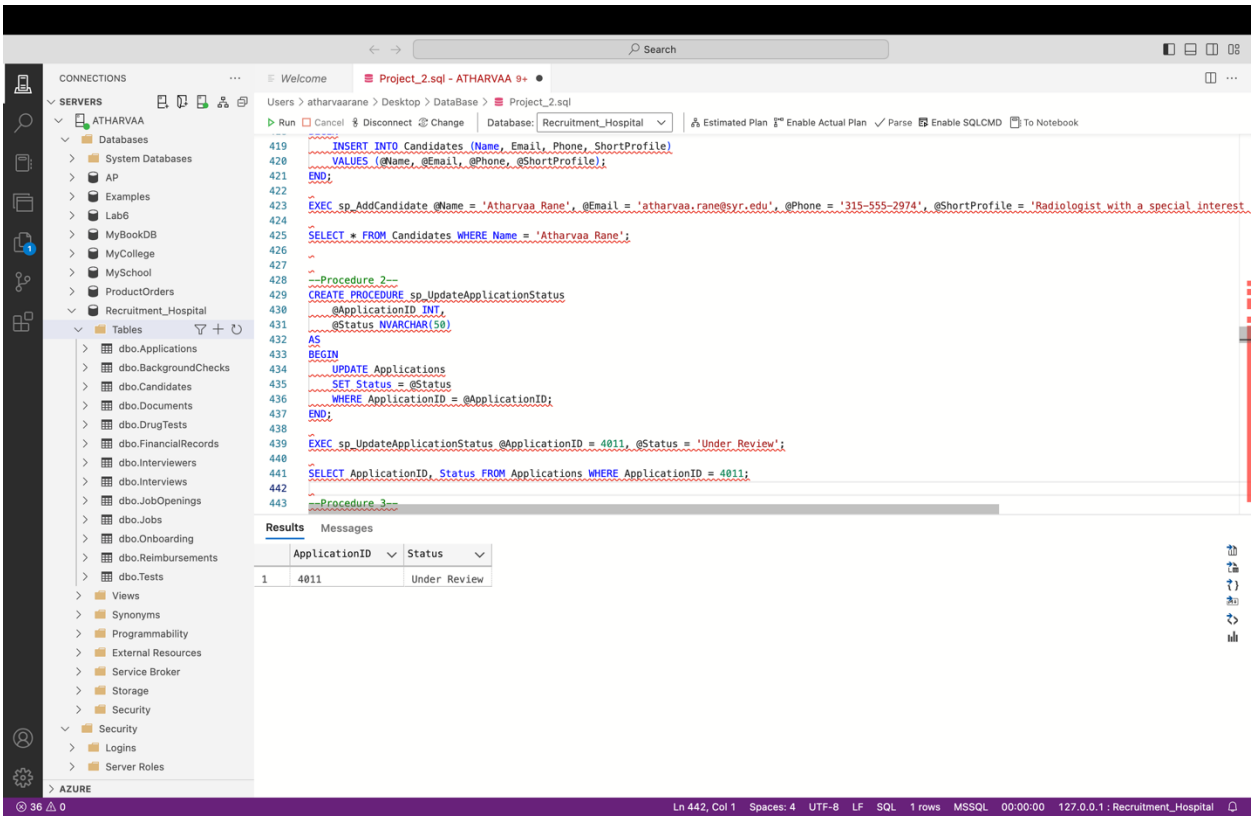
Parameters:
@Name NVARCHAR(255): A variable to store the candidate's name, allowing for a maximum of 255 characters.
@Email NVARCHAR(255): A variable for the candidate's email, also allowing up to 255 characters.
@Phone NVARCHAR(50): A variable for the candidate's phone number, limited to 50 characters.
@ShortProfile TEXT: A text variable for a brief profile or description of the candidate.



Procedure2:

```
CREATE PROCEDURE sp_UpdateApplicationStatus
    @ApplicationID INT,
    @Status NVARCHAR(50)
AS
BEGIN
    UPDATE Applications
    SET Status = @Status
    WHERE ApplicationID = @ApplicationID;
END;

EXEC sp_UpdateApplicationStatus @ApplicationID = 4011, @Status = 'Under Review';

SELECT ApplicationID, Status FROM Applications WHERE ApplicationID = 4011;
```

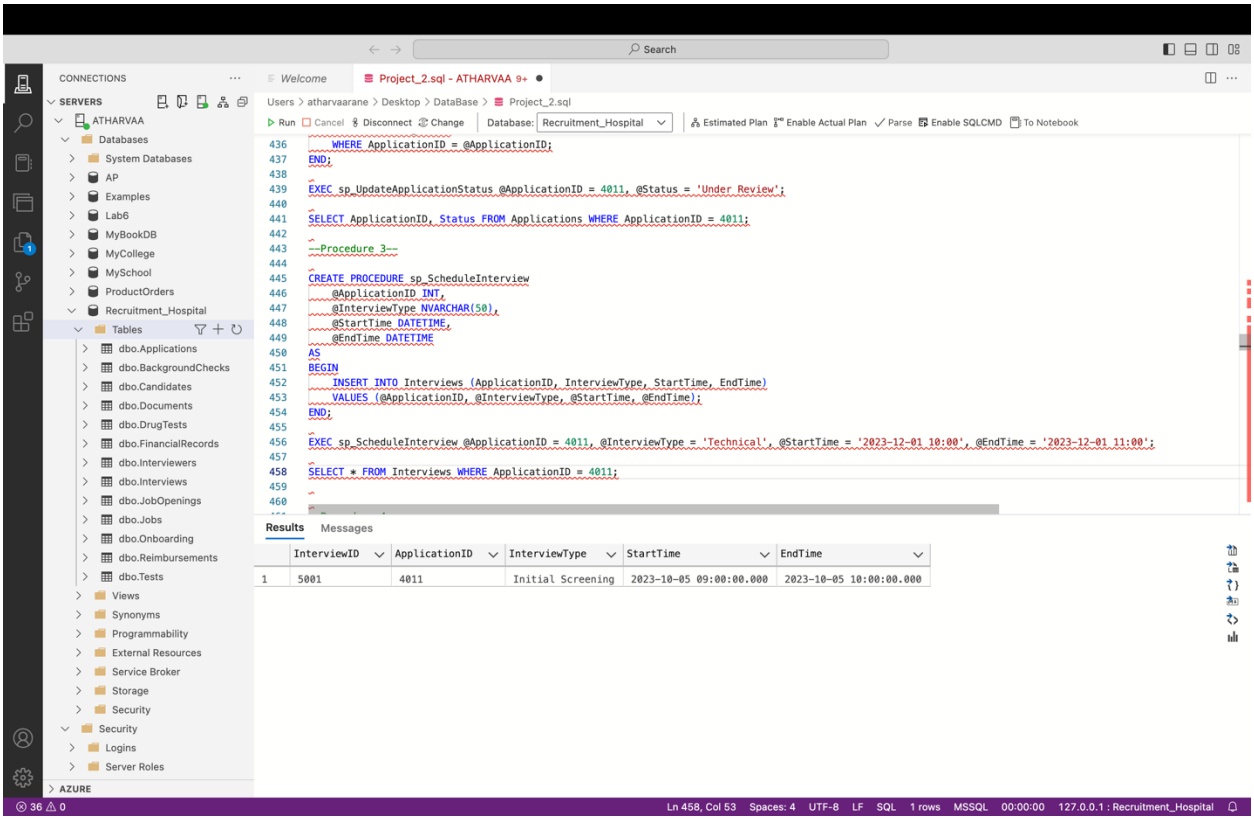
The SQL script defines and uses a stored procedure called `sp_UpdateApplicationStatus` to update the status of an application in the `Applications` table. It takes two parameters: the application ID and the new status to be applied. After defining the procedure, it is executed to update the status of an application with ID 4011 to 'Under Review'. Finally, a `SELECT` query checks to ensure the status was updated correctly by retrieving and displaying the status for this application ID.



Procedure3:

```
CREATE PROCEDURE sp_ScheduleInterview
    @ApplicationID INT,
    @InterviewType NVARCHAR(50),
    @StartTime DATETIME,
    @EndTime DATETIME
AS
BEGIN
    INSERT INTO Interviews (ApplicationID, InterviewType, StartTime, EndTime)
    VALUES (@ApplicationID, @InterviewType, @StartTime, @EndTime);
END;

EXEC sp_ScheduleInterview @ApplicationID = 4011, @InterviewType = 'Technical',
@StartTime = '2023-12-01 10:00', @EndTime = '2023-12-01 11:00';

SELECT * FROM Interviews WHERE ApplicationID = 4011;
```

The SQL script sets up a stored procedure called `sp_ScheduleInterview` to insert interview details into the `Interviews` table. It takes parameters for application ID, interview type, start time, and end time. The procedure is used to schedule a technical interview for application ID 4011 from 10:00 AM to 11:00 AM on December 1, 2023. After scheduling, a query verifies the interview details by retrieving data for this application from the `Interviews` table.

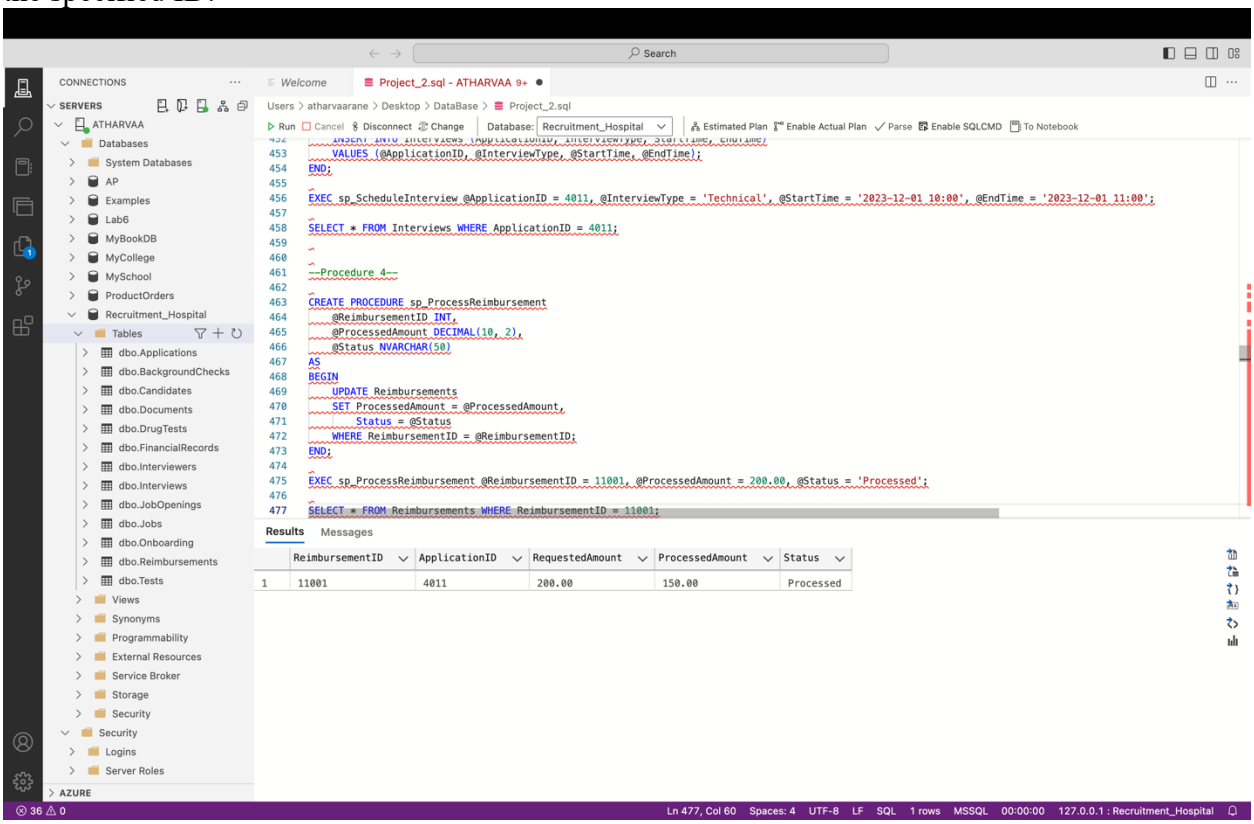


Procedure4:

```
CREATE PROCEDURE sp_ProcessReimbursement
    @ReimbursementID INT,
    @ProcessedAmount DECIMAL(10, 2),
    @Status NVARCHAR(50)
AS
BEGIN
    UPDATE Reimbursements
    SET ProcessedAmount = @ProcessedAmount,
        Status = @Status
    WHERE ReimbursementID = @ReimbursementID;
END;

EXEC sp_ProcessReimbursement @ReimbursementID = 11001, @ProcessedAmount = 200.00,
@Status = 'Processed';

SELECT * FROM Reimbursements WHERE ReimbursementID = 11001;
```

The SQL script defines a stored procedure called `sp_ProcessReimbursement` that updates the `ProcessedAmount` and `Status` fields in the `Reimbursements` table for a specific `ReimbursementID`. It is then executed to mark a reimbursement with ID 11001 as 'Processed' with a processed amount of $200.00. A subsequent `SELECT` query checks to confirm that the

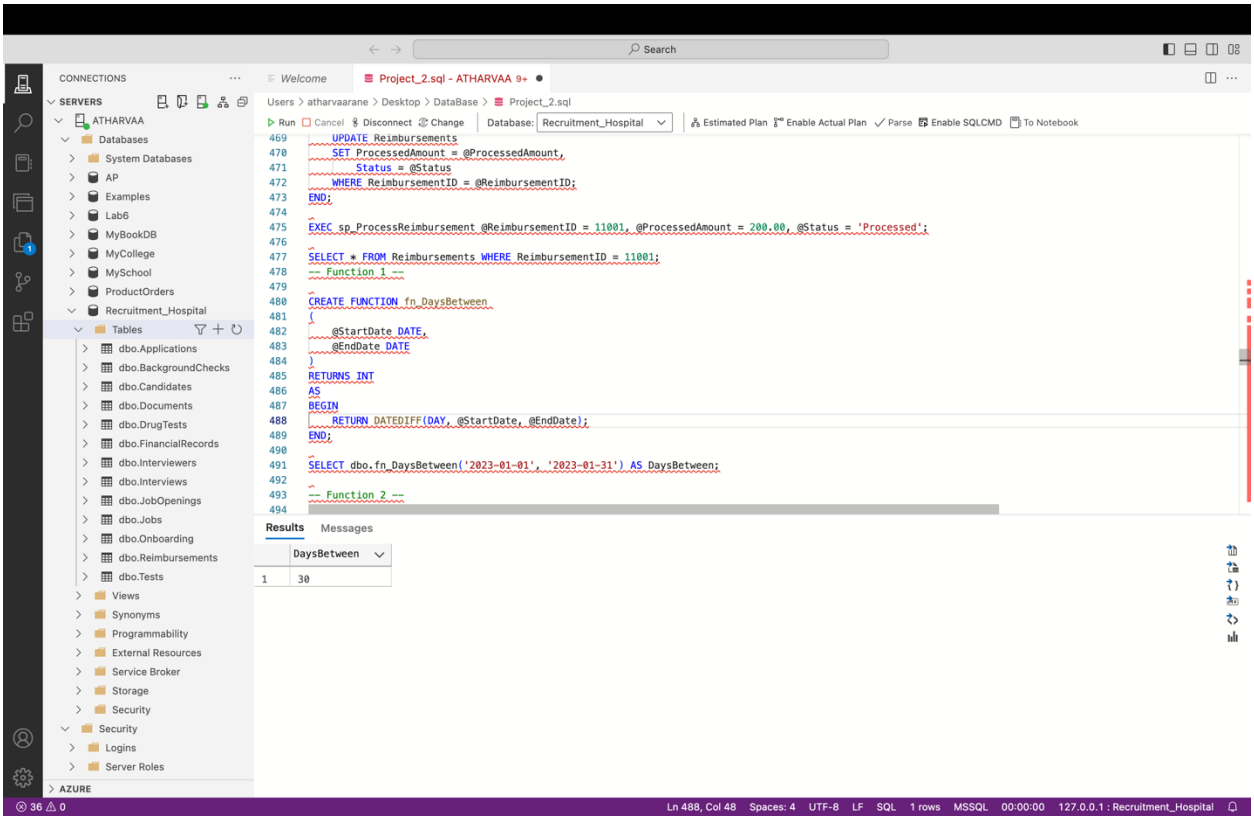
update has been applied correctly by retrieving the details from the `Reimbursements` table for the specified ID.



Functions

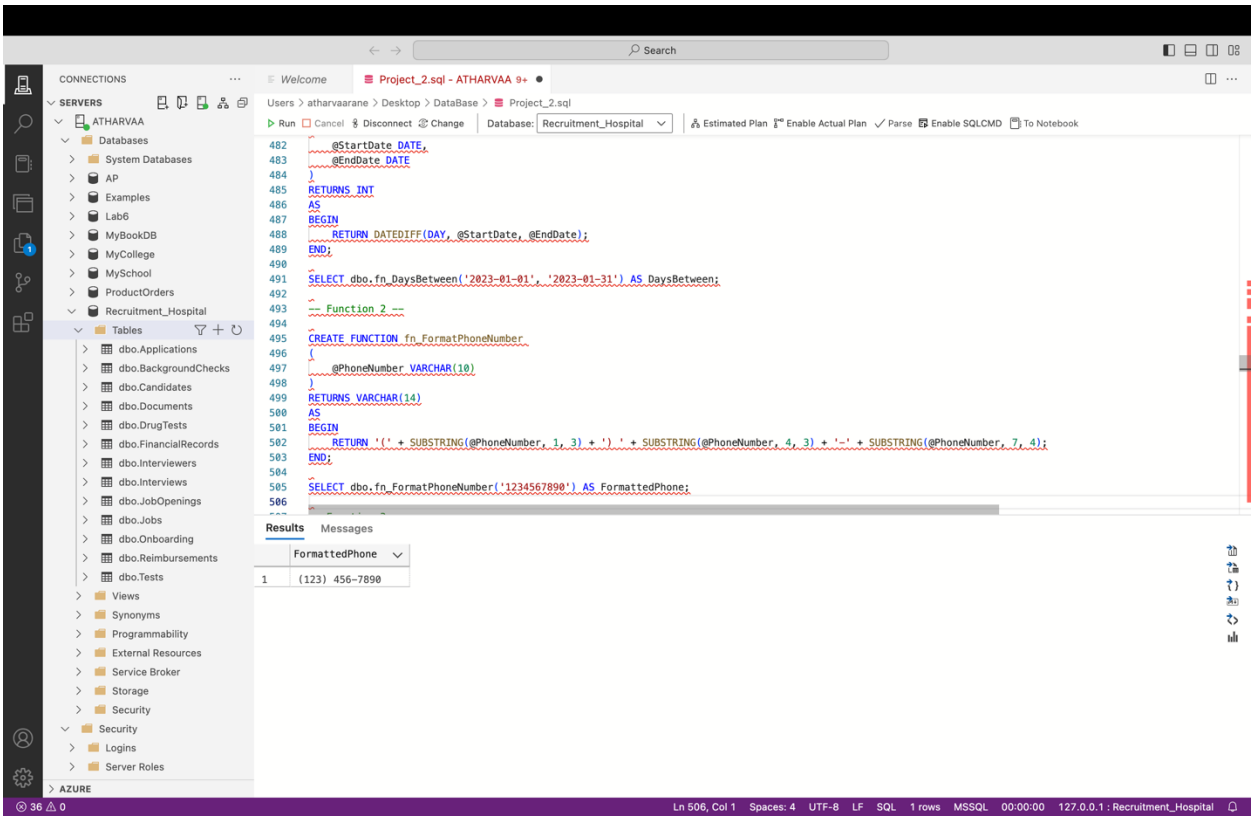
Function 1:
This function calculates the number of days between two dates, which can be useful for tracking the duration of the recruitment process or the time between application submission and interview dates.

```
CREATE FUNCTION fn_DaysBetween
(
    @StartDate DATE,
    @EndDate DATE
)
RETURNS INT
AS
BEGIN
    RETURN DATEDIFF(DAY, @StartDate, @EndDate);
END;

SELECT dbo.fn_DaysBetween('2023-01-01', '2023-01-31') AS DaysBetween;
```



This function checks if a candidate's age is above a certain threshold based on their birthdate, a common requirement in job eligibility criteria.

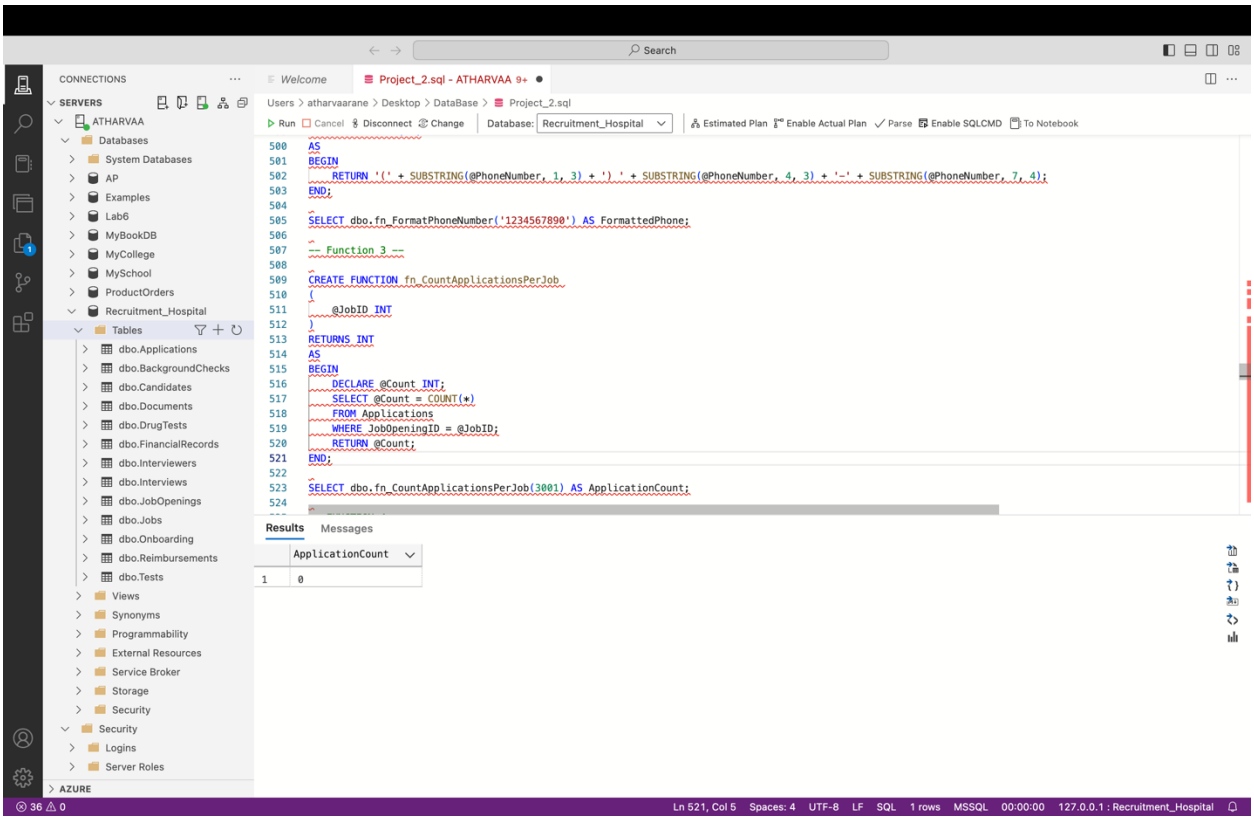
```
CREATE FUNCTION fn_FormatPhoneNumber
(
    @PhoneNumber VARCHAR(10)
)
RETURNS VARCHAR(14)
AS
BEGIN
    RETURN '(' + SUBSTRING(@PhoneNumber, 1, 3) + ') ' + SUBSTRING(@PhoneNumber, 4, 3)
+ '-' + SUBSTRING(@PhoneNumber, 7, 4);
END;

SELECT dbo.fn_FormatPhoneNumber('1234567890') AS FormattedPhone;
```



Function 3 :

Function fn_CountApplicationsPerJob that returns the number of applications for a specified job ID by querying the Applications table, and it is then executed to get the application count for job ID 3001.
Top of Form

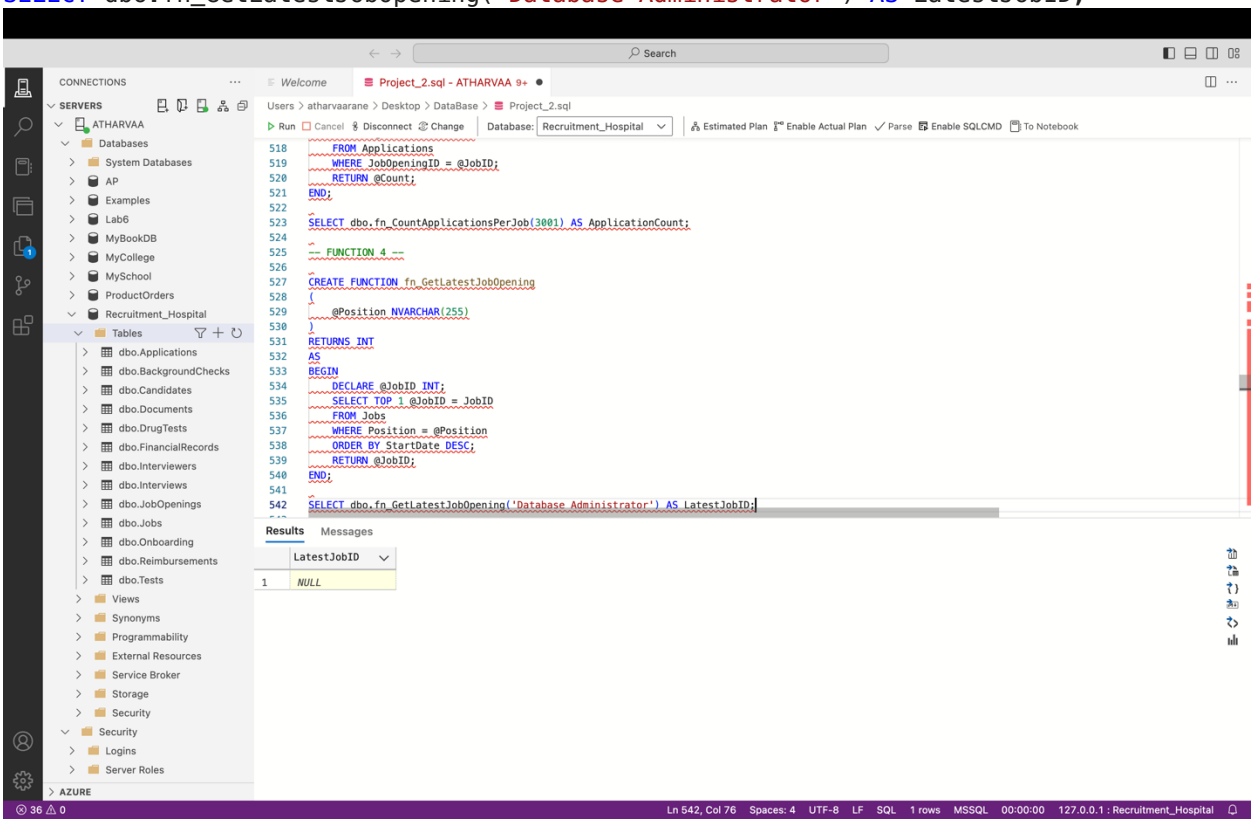
```sql
CREATE FUNCTION fn_CountApplicationsPerJob
(
    @JobID INT
)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;
    SELECT @Count = COUNT(*)
    FROM Applications
    WHERE JobOpeningID = @JobID;
    RETURN @Count;
END;

SELECT dbo.fn_CountApplicationsPerJob(3001) AS ApplicationCount;
```



Function4:

Function fn_GetLatestJobOpening that retrieves the most recent job ID for a specified position from the Jobs table, ordered by start date, and executes it to find the latest job ID for the position "Database Administrator".
Top of Form

```
CREATE FUNCTION fn_GetLatestJobOpening
(
    @Position NVARCHAR(255)
)
RETURNS INT
AS
BEGIN
    DECLARE @JobID INT;
    SELECT TOP 1 @JobID = JobID
    FROM Jobs
    WHERE Position = @Position
    ORDER BY StartDate DESC;
    RETURN @JobID;
END;

SELECT dbo.fn_GetLatestJobOpening('Database Administrator') AS LatestJobID;
```



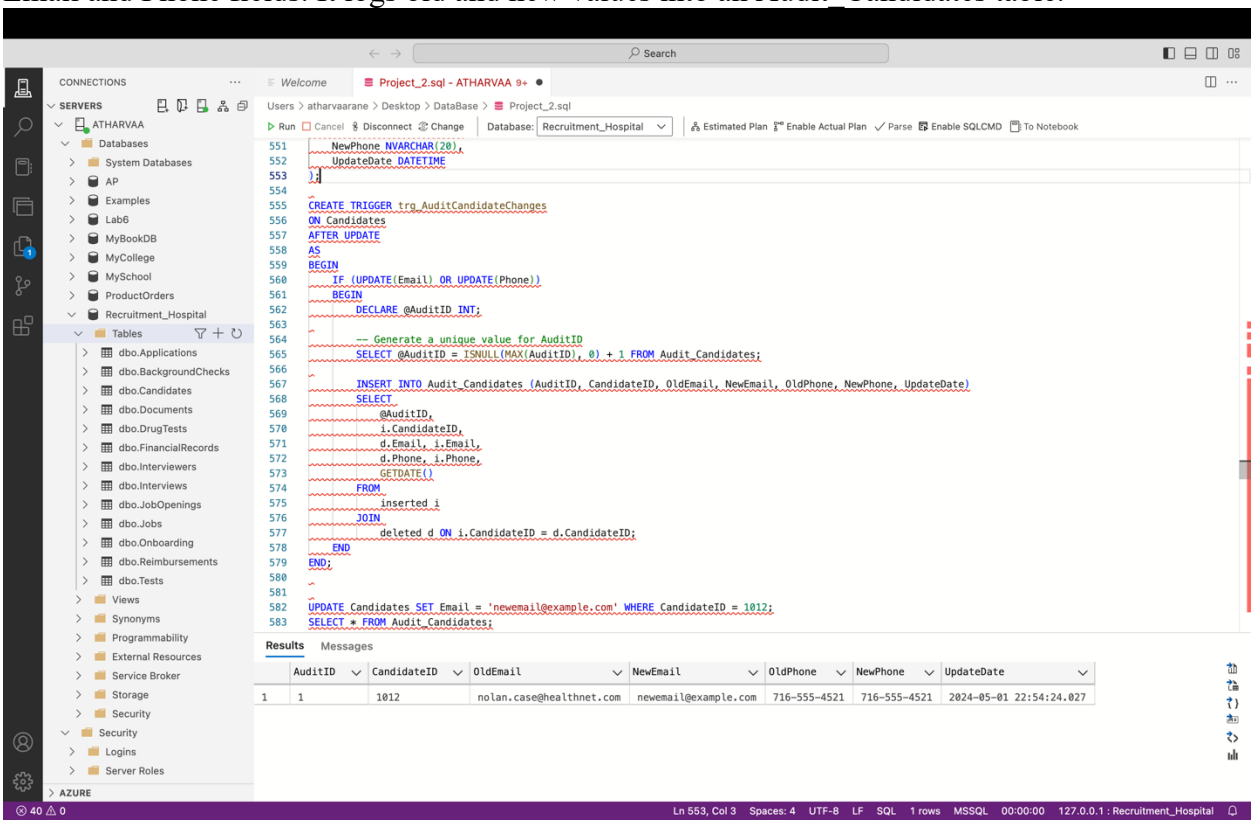
Triggers

Trigger1:

```
CREATE TABLE Audit_Candidates (
    AuditID INT PRIMARY KEY,
    CandidateID INT,
    OldEmail NVARCHAR(255),
    NewEmail NVARCHAR(255),
    OldPhone NVARCHAR(20),
    NewPhone NVARCHAR(20),
    UpdateDate DATETIME
);

CREATE TRIGGER trg_AuditCandidateChanges
ON Candidates
AFTER UPDATE
AS
BEGIN
    IF (UPDATE(Email) OR UPDATE(Phone))
    BEGIN
        DECLARE @AuditID INT;

        -- Generate a unique value for AuditID
        SELECT @AuditID = ISNULL(MAX(AuditID), 0) + 1 FROM Audit_Candidates;

        INSERT INTO Audit_Candidates (AuditID, CandidateID, OldEmail, NewEmail,
OldPhone, NewPhone, UpdateDate)
        SELECT
            @AuditID,
            i.CandidateID,
            d.Email, i.Email,
            d.Phone, i.Phone,
            GETDATE()
        FROM
            inserted i
        JOIN
            deleted d ON i.CandidateID = d.CandidateID;
    END
END;


UPDATE Candidates SET Email = 'newemail@example.com' WHERE CandidateID = 1012;
SELECT * FROM Audit_Candidates;
```

Trigger to Audit Candidate Updates

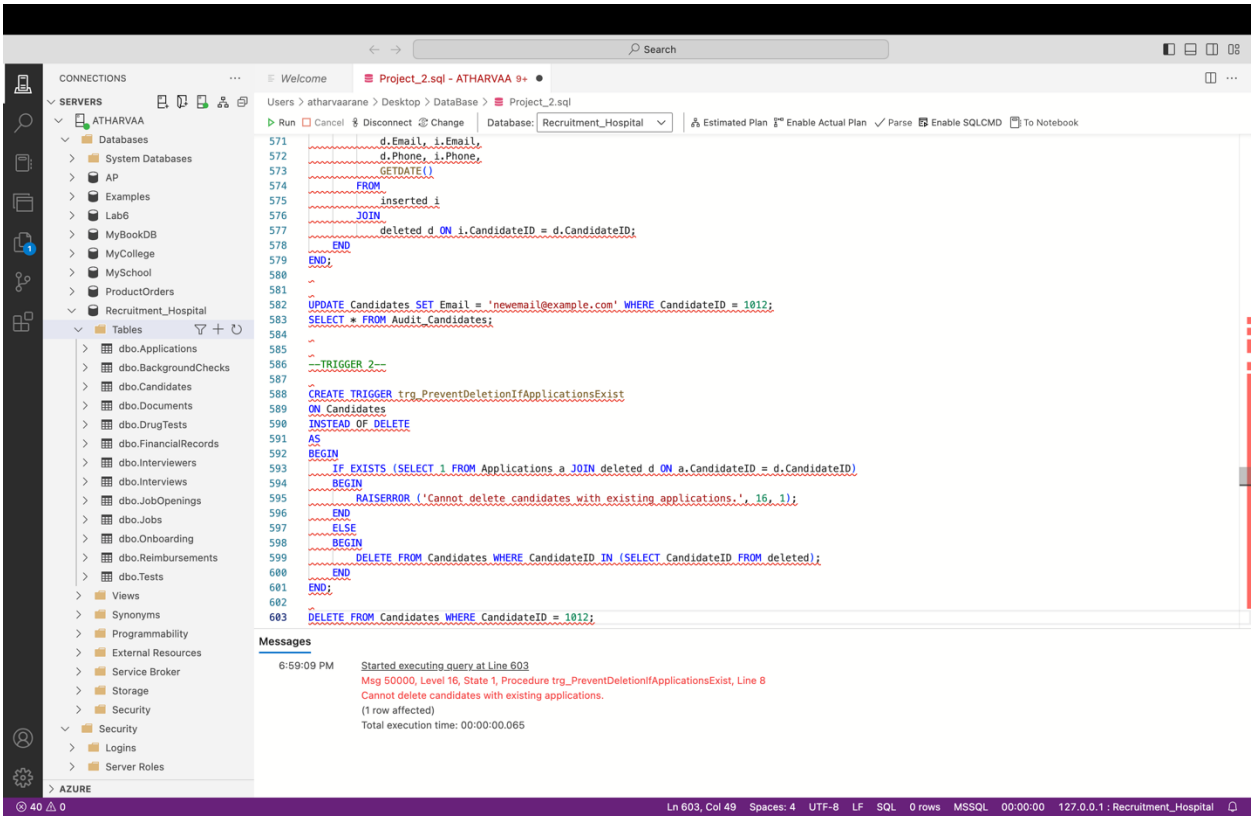
This trigger will record changes to candidate information, specifically tracking updates to the Email and Phone fields. It logs old and new values into an Audit_Candidates table.



Trigger 2 :

```
CREATE TRIGGER trg_PreventDeletionIfApplicationsExist
ON Candidates
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Applications a JOIN deleted d ON a.CandidateID =
d.CandidateID)
    BEGIN
        RAISERROR ('Cannot delete candidates with existing applications.', 16, 1);
    END
    ELSE
    BEGIN
        DELETE FROM Candidates WHERE CandidateID IN (SELECT CandidateID FROM deleted);
    END
END;

DELETE FROM Candidates WHERE CandidateID = 1012;
```

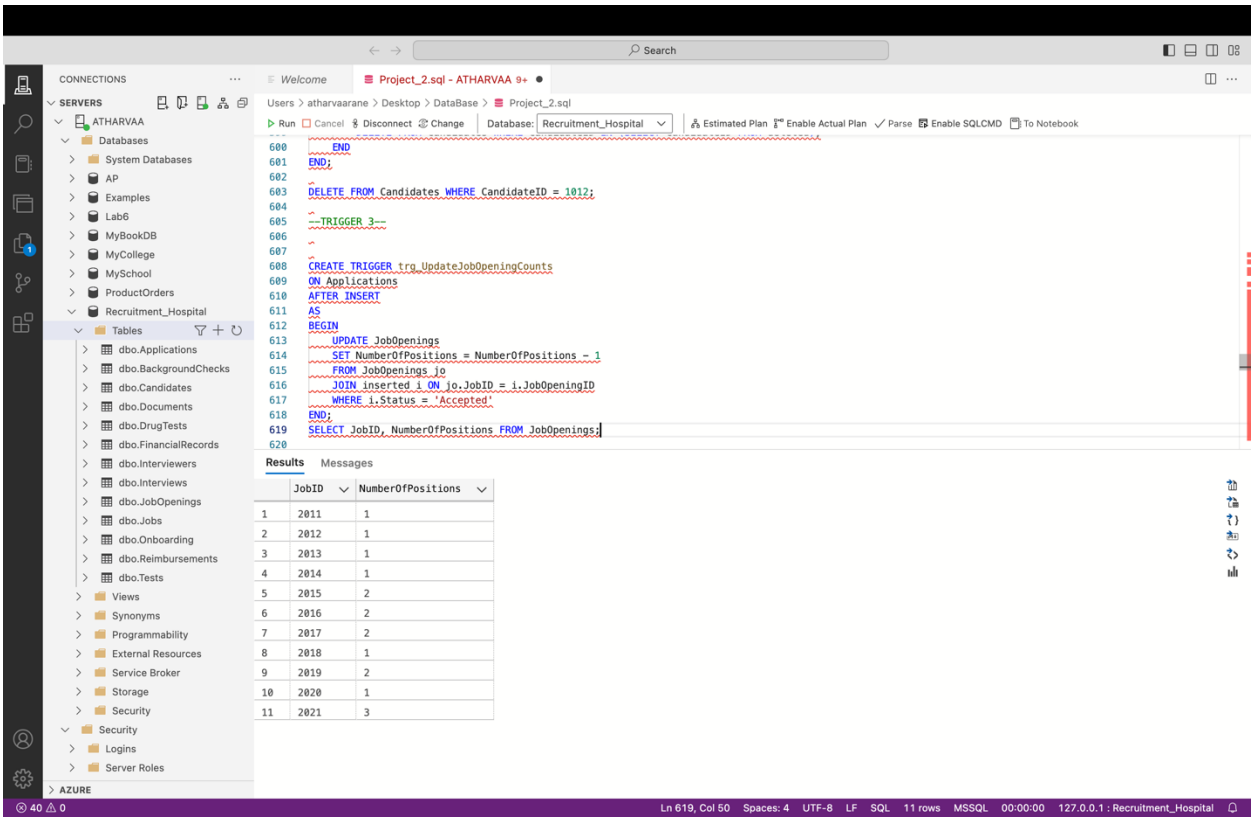
Trigger to Prevent Deletion of Candidates with Applications
This trigger prevents the deletion of candidate records if they have any associated applications, ensuring data integrity across related tables.



Trigger3:

Trigger to Automatically Update Job Opening Counts
This trigger adjusts the number of positions available in the JobOpenings table whenever a new application is inserted and marked as Accepted.

```
CREATE TRIGGER trg_UpdateJobOpeningCounts
ON Applications
AFTER INSERT
AS
BEGIN
    UPDATE JobOpenings
    SET NumberOfPositions = NumberOfPositions - 1
    FROM JobOpenings jo
    JOIN inserted i ON jo.JobID = i.JobOpeningID
    WHERE i.Status = 'Accepted'
END;
SELECT JobID, NumberOfPositions FROM JobOpenings;
```

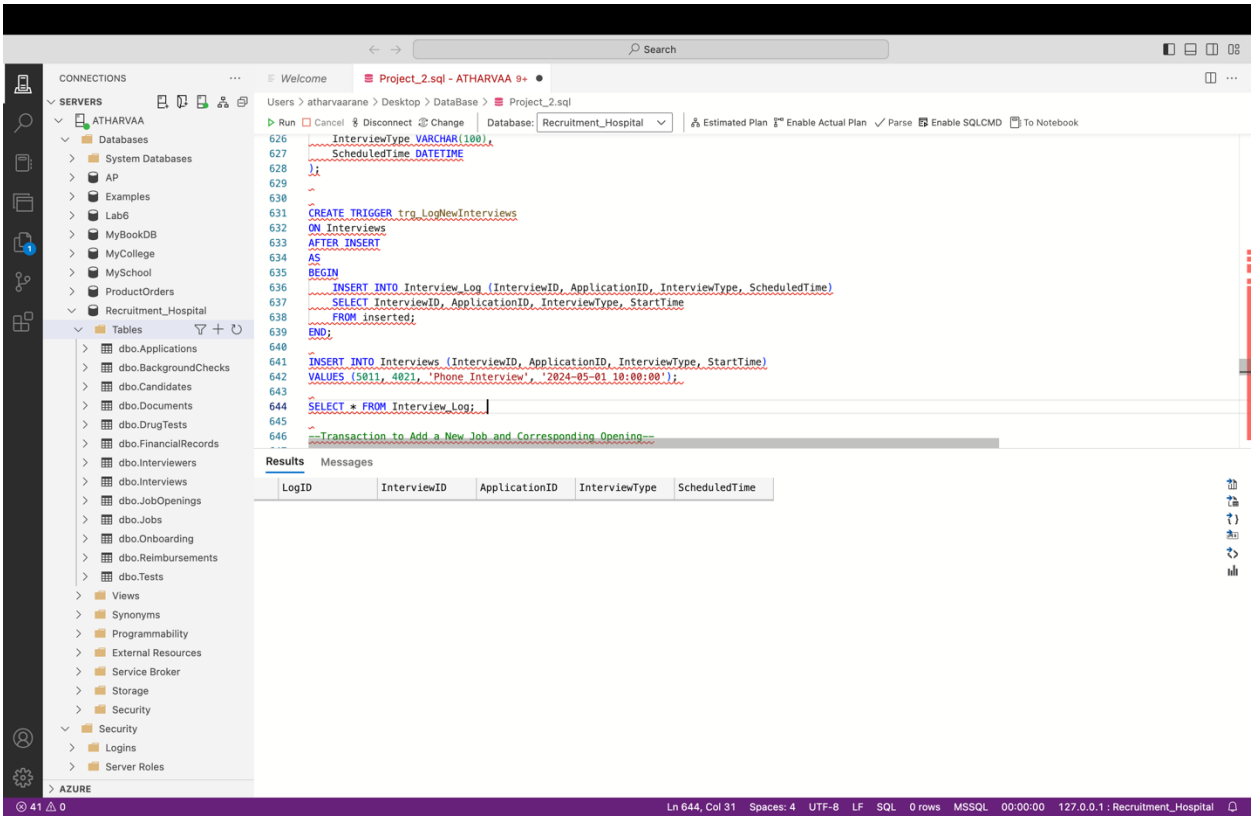


Trigger 4 :

Trigger to Log Interview Scheduling
This trigger logs every new interview scheduled into an Interview_Log table, capturing the datetime and details of the interview.

```
CREATE TABLE Interview_Log (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    InterviewID INT,
    ApplicationID INT,
    InterviewType VARCHAR(100),
    ScheduledTime DATETIME
);


CREATE TRIGGER trg_LogNewInterviews
ON Interviews
AFTER INSERT
AS
BEGIN
```

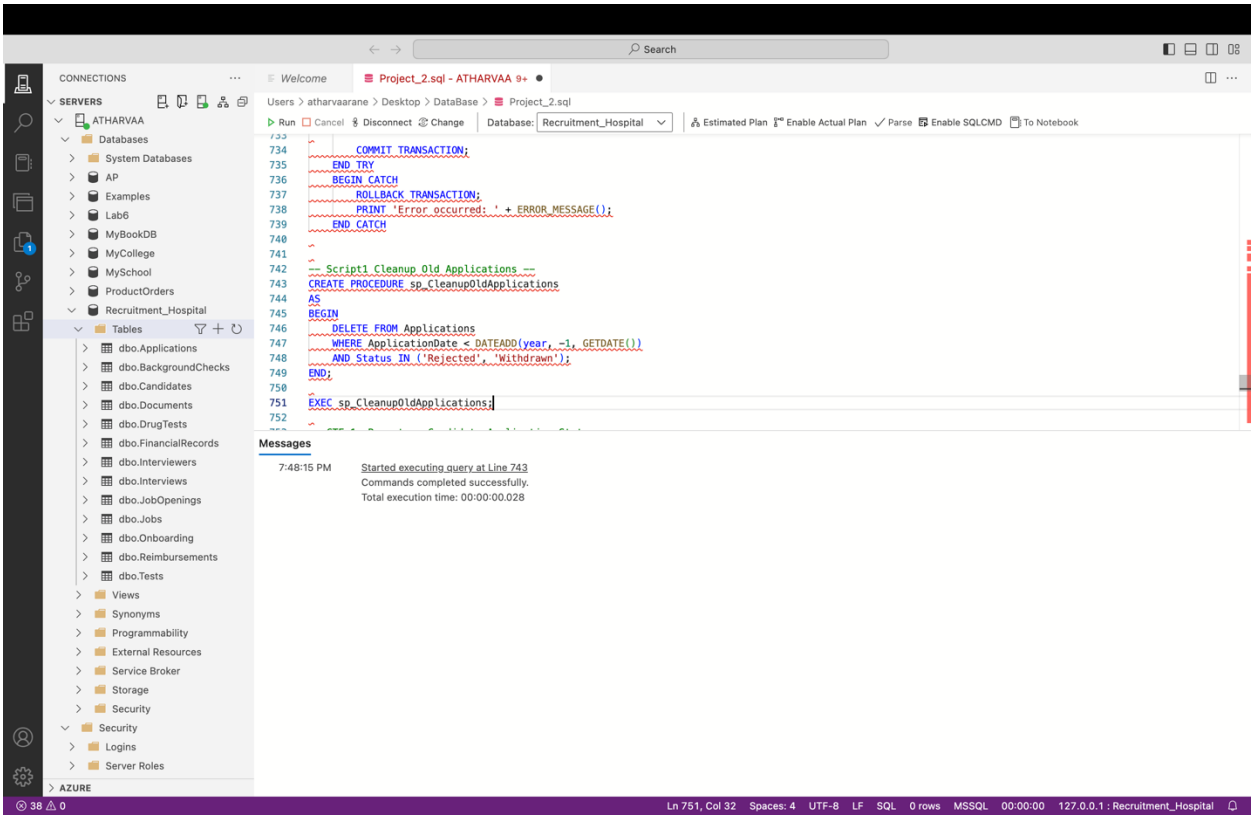
```
    INSERT INTO Interview_Log (InterviewID, ApplicationID, InterviewType,
ScheduledTime)
    SELECT InterviewID, ApplicationID, InterviewType, StartTime
    FROM inserted;
END;

INSERT INTO Interviews (InterviewID, ApplicationID, InterviewType, StartTime)
VALUES (5011, 4021, 'Phone Interview', '2024-05-01 10:00:00');

SELECT * FROM Interview_Log;
```



Scripts

Script1:Automated Data Cleanup Script
This script is designed to remove outdated job applications from the database that are older than a certain threshold (e.g., one year) and have been marked as either "Rejected" or "Withdrawn". This helps in keeping the database size manageable and ensures data relevancy.

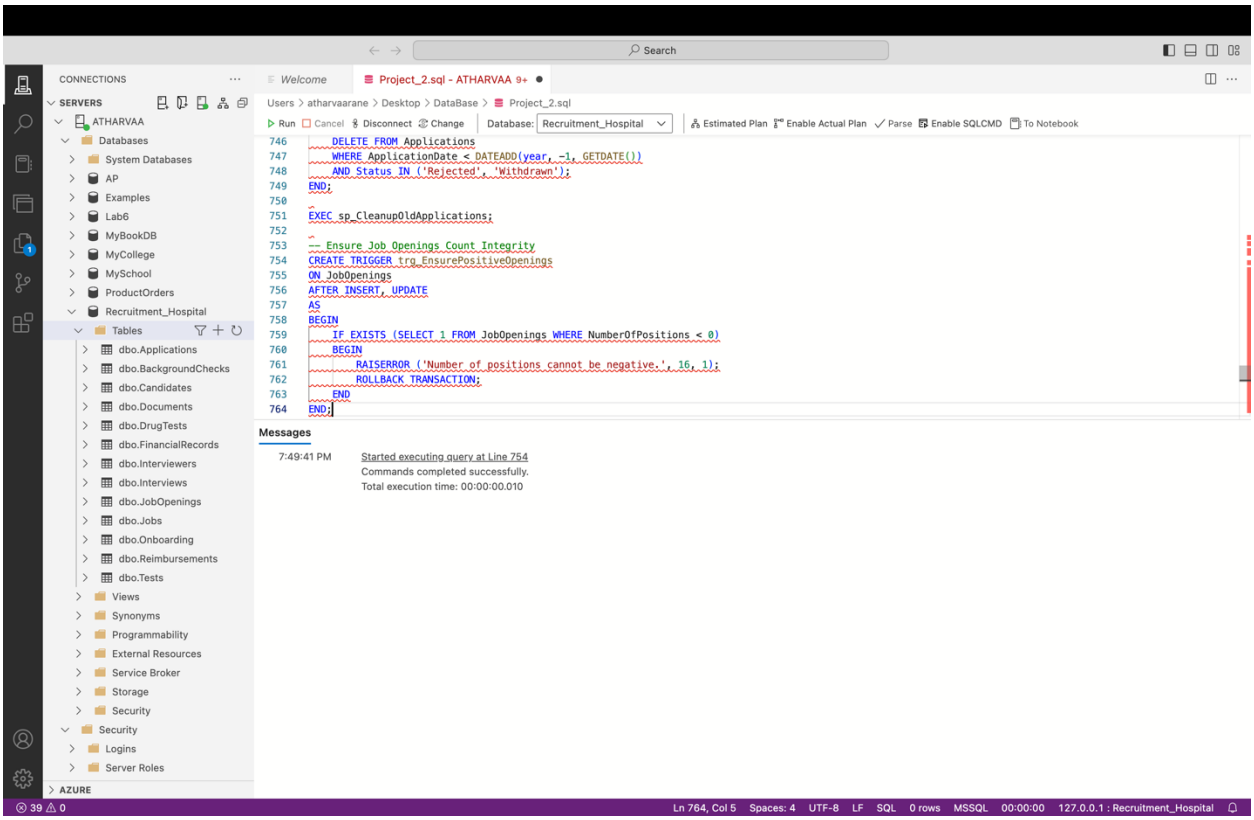
```
CREATE PROCEDURE sp_CleanupOldApplications
AS
BEGIN
    DELETE FROM Applications
    WHERE ApplicationDate < DATEADD(year, -1, GETDATE())
    AND Status IN ('Rejected', 'Withdrawn');
END;

EXEC sp_CleanupOldApplications;
```



Script2:Data Integrity Script
This script ensures that the number of positions available in job openings is never negative, which could occur due to data entry errors or system glitches.

```
-- Ensure Job Openings Count Integrity
```

```
CREATE TRIGGER trg_EnsurePositiveOpenings
ON JobOpenings
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM JobOpenings WHERE NumberOfPositions < 0)
    BEGIN
        RAISERROR ('Number of positions cannot be negative.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

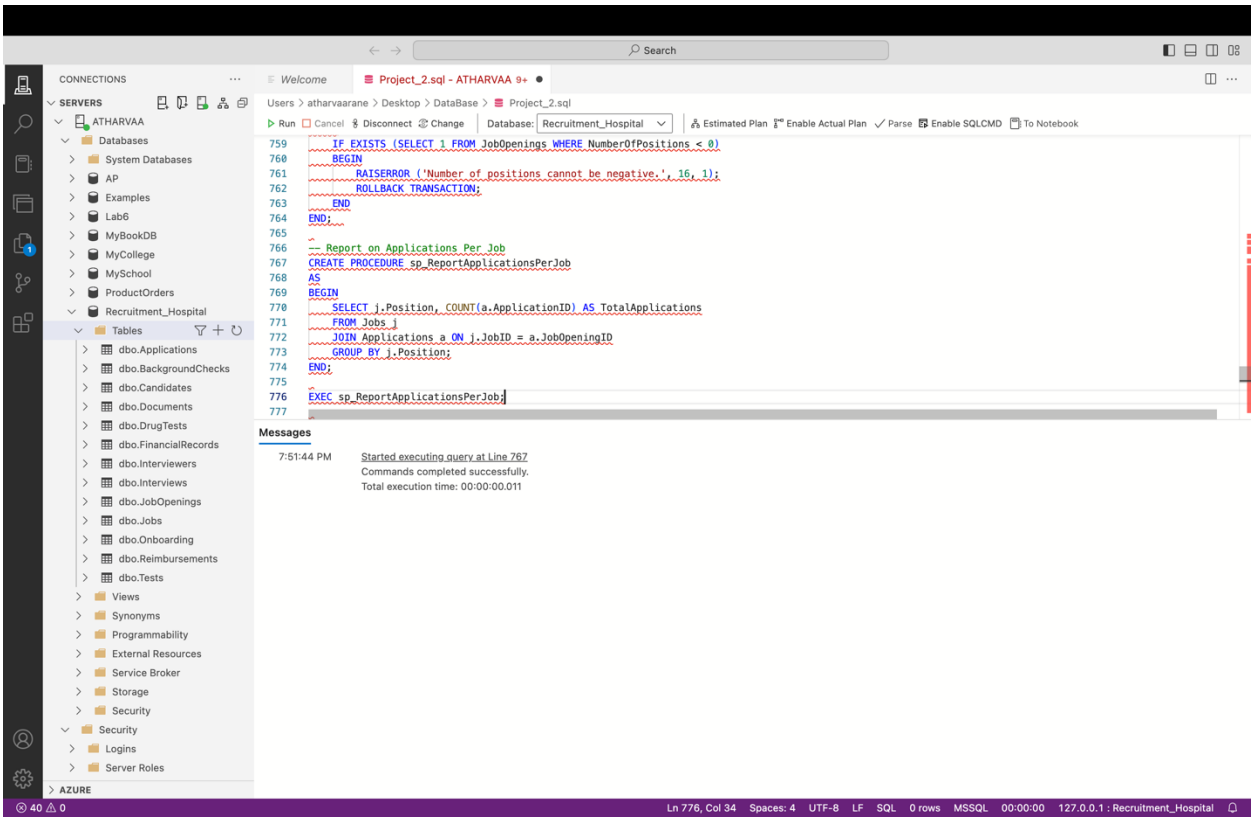


Script3: Reporting Script

This stored procedure generates a report showing the number of applications per job position, which can help in understanding which positions are most or least popular among candidates.

```
-- Report on Applications Per Job --
CREATE PROCEDURE sp_ReportApplicationsPerJob
AS
BEGIN
    SELECT j.Position, COUNT(a.ApplicationID) AS TotalApplications
    FROM Jobs j
    JOIN Applications a ON j.JobID = a.JobOpeningID
    GROUP BY j.Position;
```

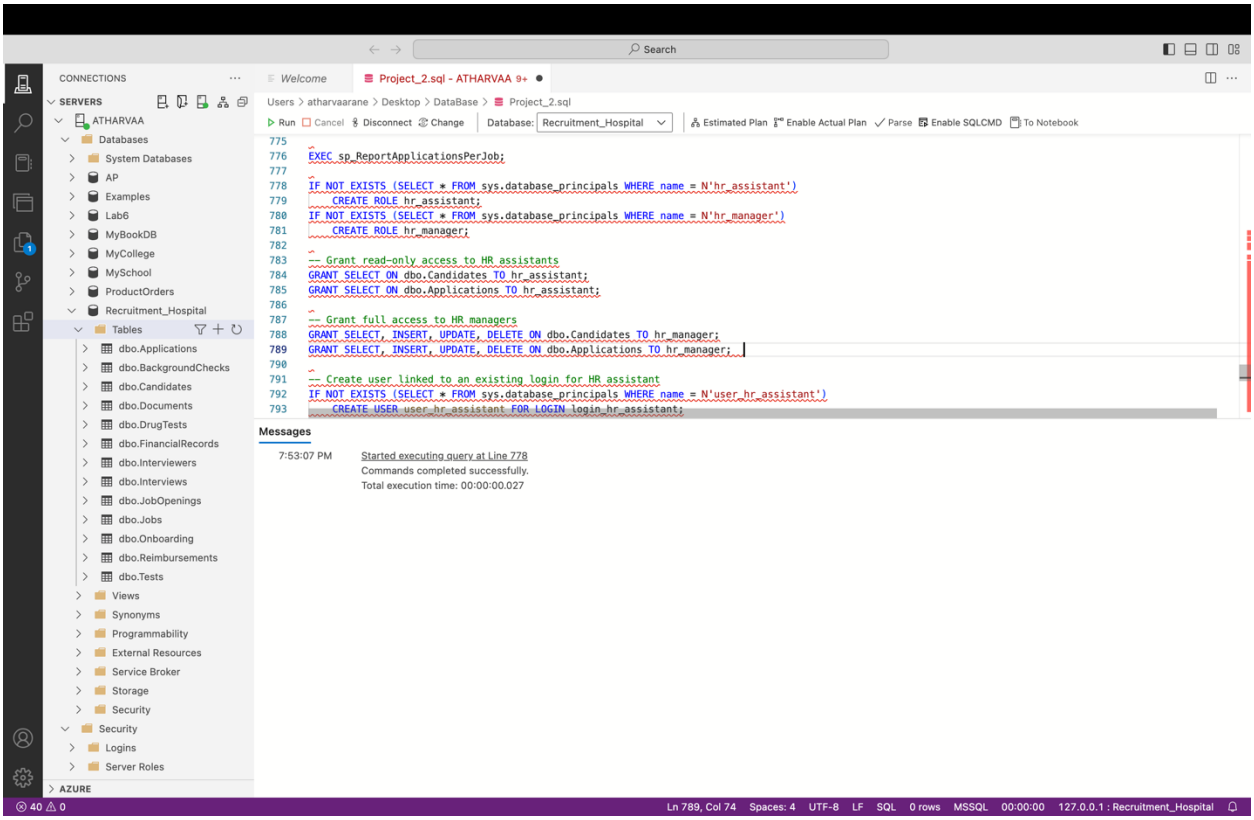
```
END;

EXEC sp_ReportApplicationsPerJob;
```



Script4: Security Script (User Access Management)
This script is used to manage user roles and permissions, ensuring that only authorized users can view, insert, or update data in sensitive tables like Candidates and Applications.

```
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'hr_assistant')
    CREATE ROLE hr_assistant;
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'hr_manager')
    CREATE ROLE hr_manager;

-- Grant read-only access to HR assistants
GRANT SELECT ON dbo.Candidates TO hr_assistant;
GRANT SELECT ON dbo.Applications TO hr_assistant;

-- Grant full access to HR managers
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Candidates TO hr_manager;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Applications TO hr_manager;

-- Create user linked to an existing login for HR assistant
```

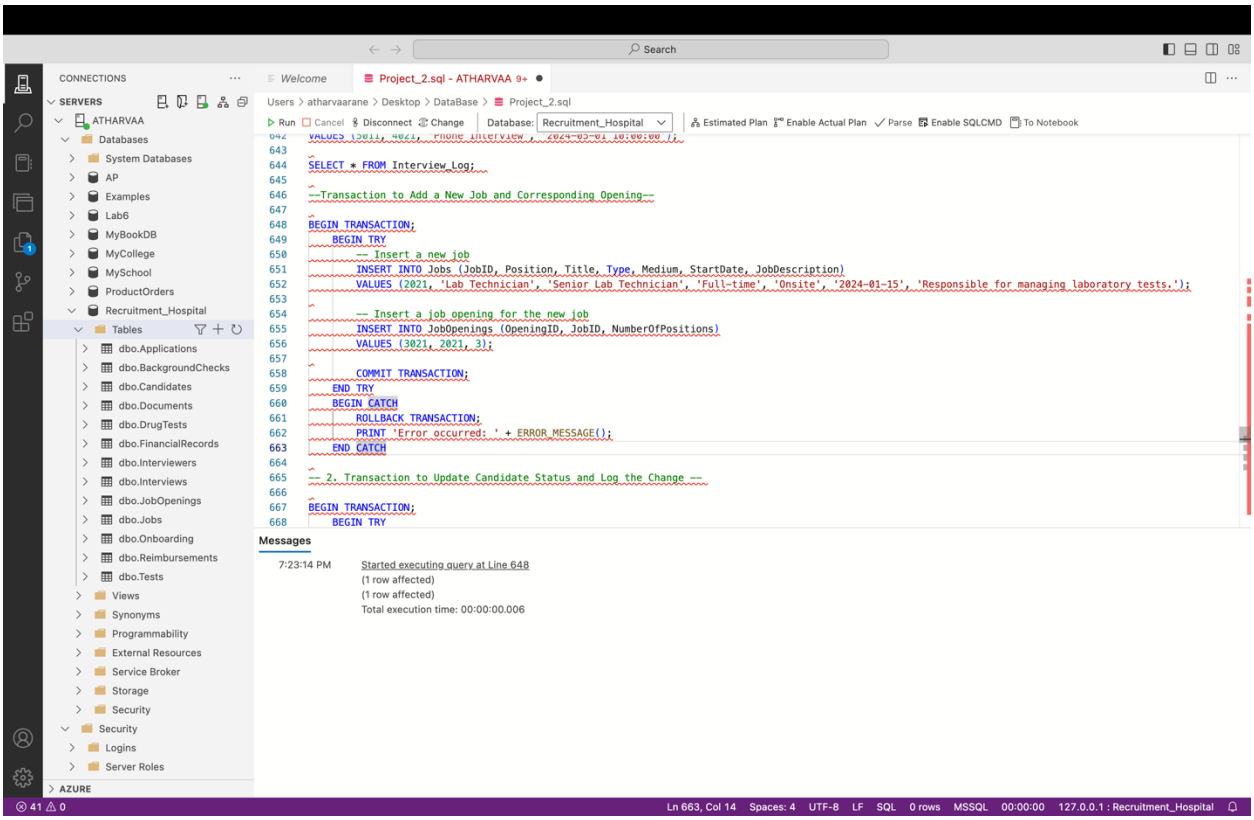
```
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name =
N'user_hr_assistant')
    CREATE USER user_hr_assistant FOR LOGIN login_hr_assistant;
ALTER ROLE hr_assistant ADD MEMBER user_hr_assistant;

-- Create user linked to an existing login for HR manager
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = N'user_hr_manager')
    CREATE USER user_hr_manager FOR LOGIN login_hr_manager;
ALTER ROLE hr_manager ADD MEMBER user_hr_manager;
```



Transactions

Transaction1:
Transaction to Add a New Job and Corresponding Opening
This transaction ensures that when a new job is added, its corresponding job opening is also created correctly.

```
BEGIN TRANSACTION;
    BEGIN TRY
        -- Insert a new job
        INSERT INTO Jobs (JobID, Position, Title, Type, Medium, StartDate,
JobDescription)
        VALUES (2021, 'Lab Technician', 'Senior Lab Technician', 'Full-time',
'Onsite', '2024-01-15', 'Responsible for managing laboratory tests.');

        -- Insert a job opening for the new job
        INSERT INTO JobOpenings (OpeningID, JobID, NumberOfPositions)
        VALUES (3021, 2021, 3);

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'Error occurred: ' + ERROR_MESSAGE();
    END CATCH
```

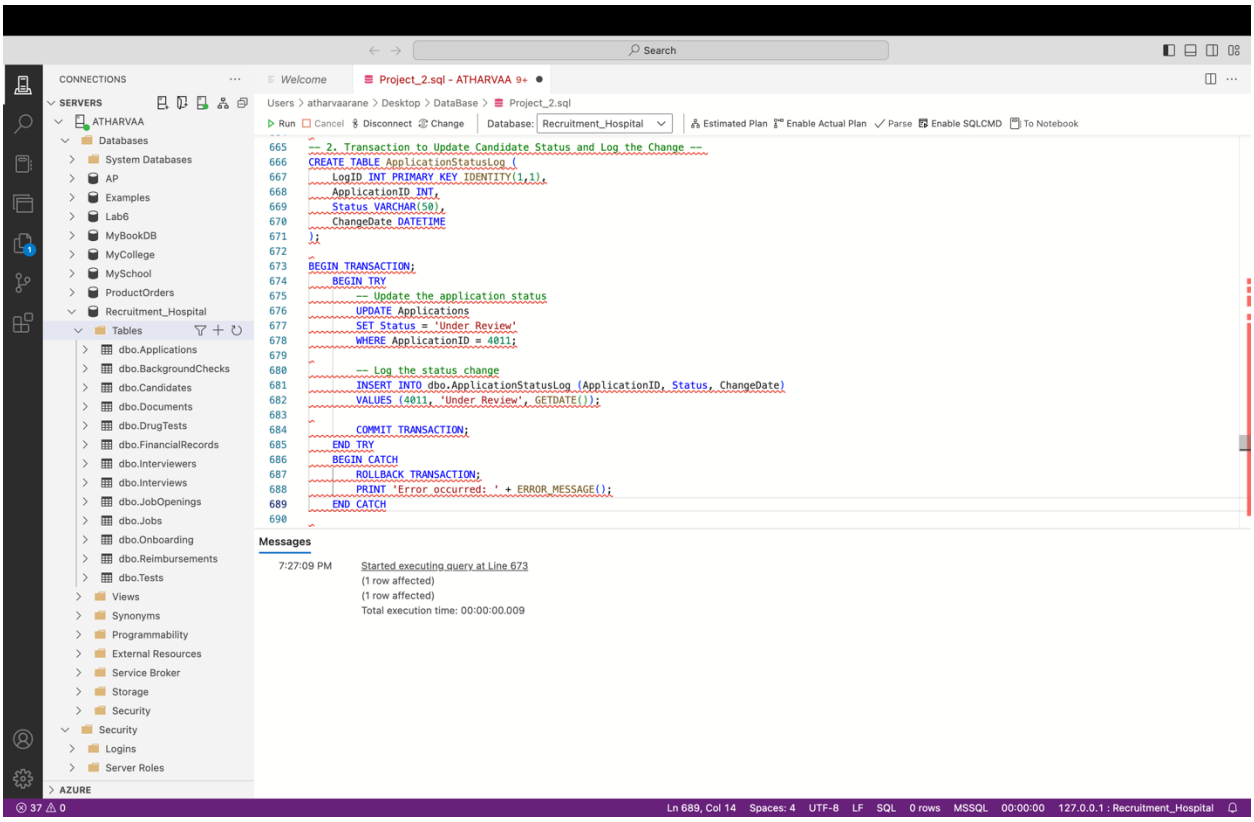


Transaction2:

Transaction to Update Candidate Status and Log the Change
This transaction updates a candidate's application status and logs the update for auditing purposes.

```
    LogID INT PRIMARY KEY IDENTITY(1,1),
    ApplicationID INT,
    Status VARCHAR(50),
    ChangeDate DATETIME
);

BEGIN TRANSACTION;
    BEGIN TRY
        -- Update the application status
        UPDATE Applications
        SET Status = 'Under Review'
        WHERE ApplicationID = 4011;

        -- Log the status change
        INSERT INTO dbo.ApplicationStatusLog (ApplicationID, Status, ChangeDate)
        VALUES (4011, 'Under Review', GETDATE());

        COMMIT TRANSACTION;
```

```
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'Error occurred: ' + ERROR_MESSAGE();
    END CATCH
```

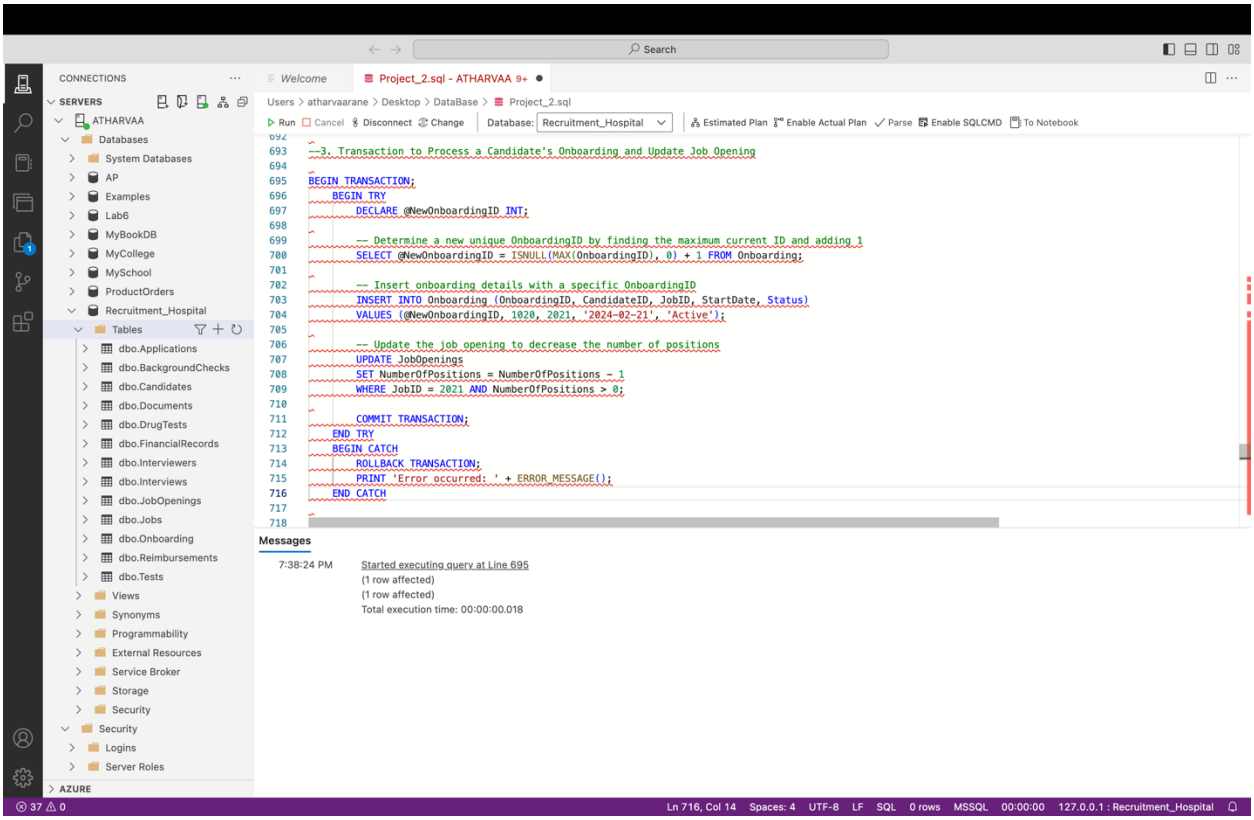


Transaction3:

Transaction to Process a Candidate's Onboarding and Update Job Opening
This transaction processes a candidate's onboarding and updates the job opening count to reflect the filled position.

```
BEGIN TRANSACTION;
    BEGIN TRY
        DECLARE @NewOnboardingID INT;

        -- Determine a new unique OnboardingID by finding the maximum current ID and
adding 1
        SELECT @NewOnboardingID = ISNULL(MAX(OnboardingID), 0) + 1 FROM Onboarding;

        -- Insert onboarding details with a specific OnboardingID
        INSERT INTO Onboarding (OnboardingID, CandidateID, JobID, StartDate, Status)
        VALUES (@NewOnboardingID, 1020, 2021, '2024-02-21', 'Active');
```

```
        -- Update the job opening to decrease the number of positions
        UPDATE JobOpenings
        SET NumberOfPositions = NumberOfPositions - 1
        WHERE JobID = 2021 AND NumberOfPositions > 0;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'Error occurred: ' + ERROR_MESSAGE();
    END CATCH
```

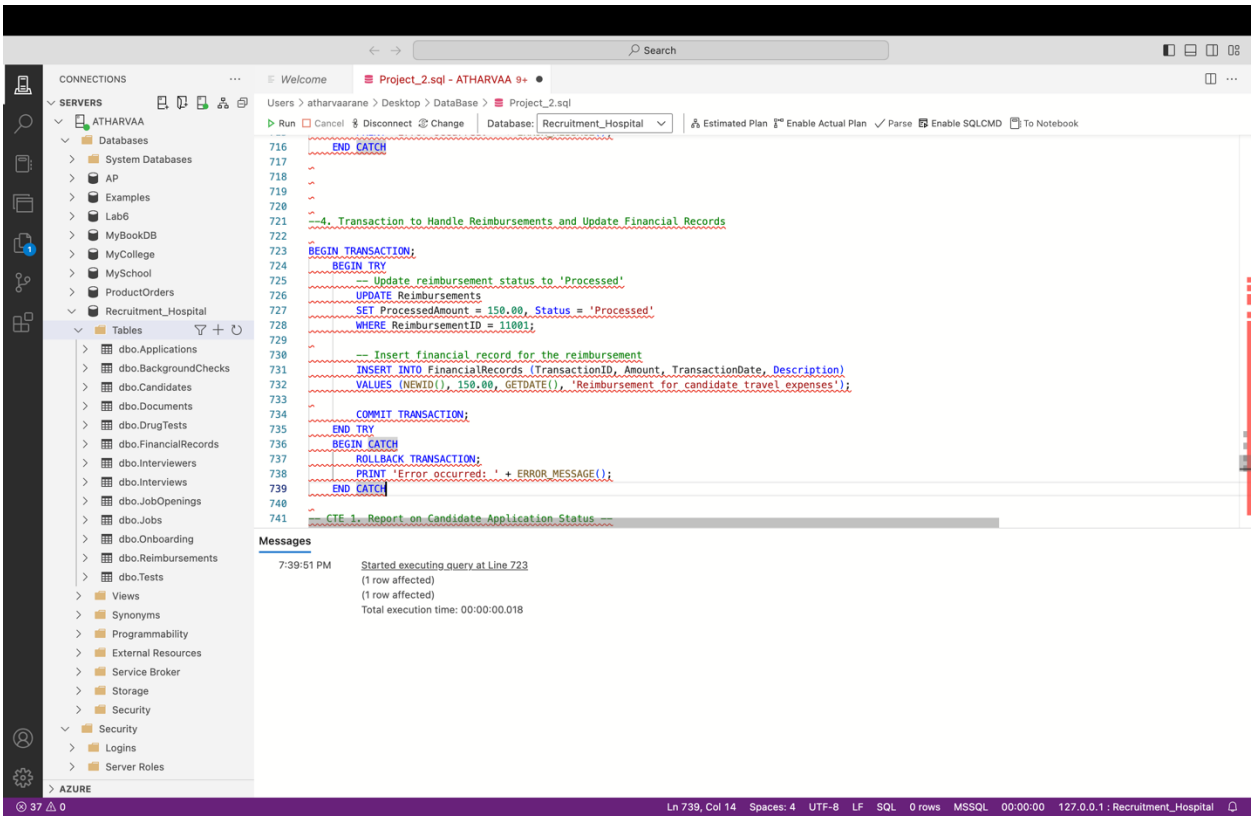


Transaction4:

Transaction to Handle Reimbursements and Update Financial Records
This transaction processes a reimbursement and updates financial records to reflect the payout.

```
BEGIN TRANSACTION;
    BEGIN TRY
        -- Update reimbursement status to 'Processed'
        UPDATE Reimbursements
        SET ProcessedAmount = 150.00, Status = 'Processed'
```

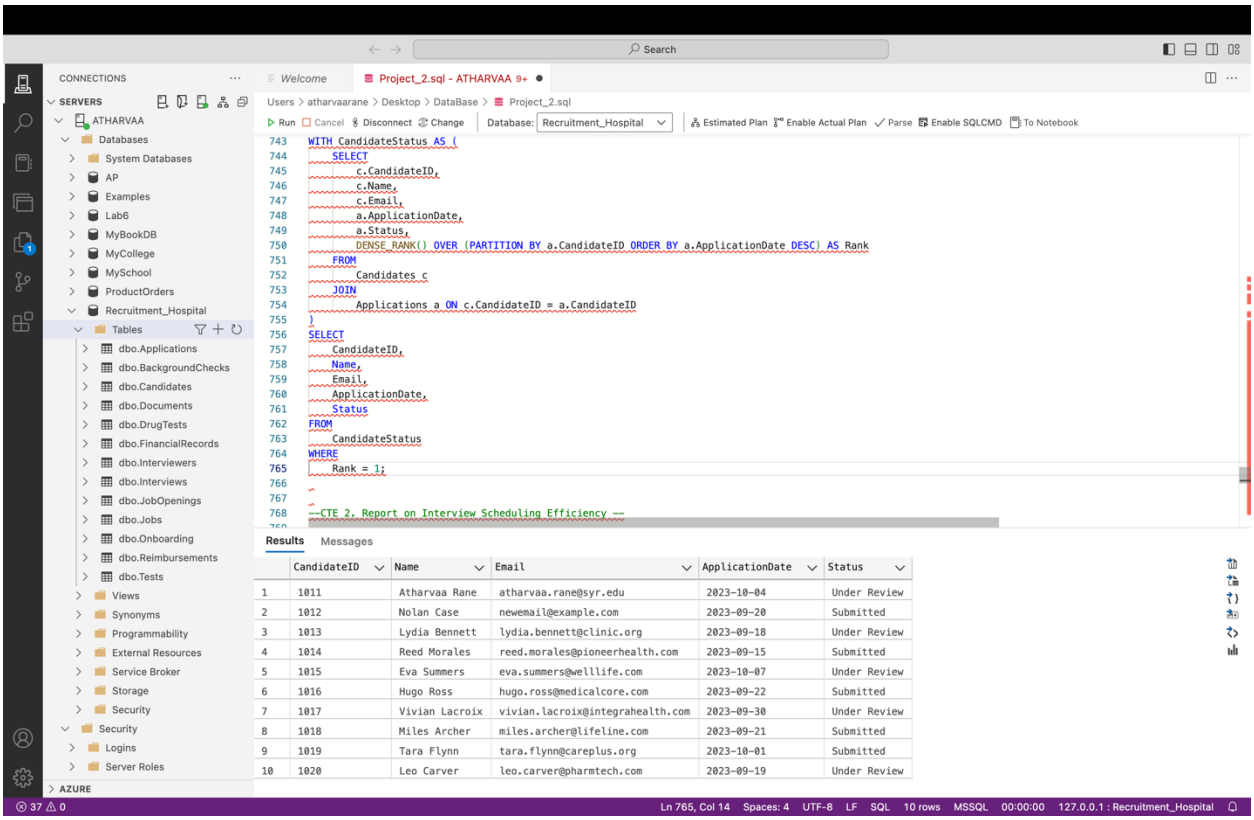
```
        WHERE ReimbursementID = 11001;

        -- Insert financial record for the reimbursement
        INSERT INTO FinancialRecords (TransactionID, Amount, TransactionDate,
Description)
        VALUES (NEWID(), 150.00, GETDATE(), 'Reimbursement for candidate travel
expenses');

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        PRINT 'Error occurred: ' + ERROR_MESSAGE();
    END CATCH
```



Common Table Expressions

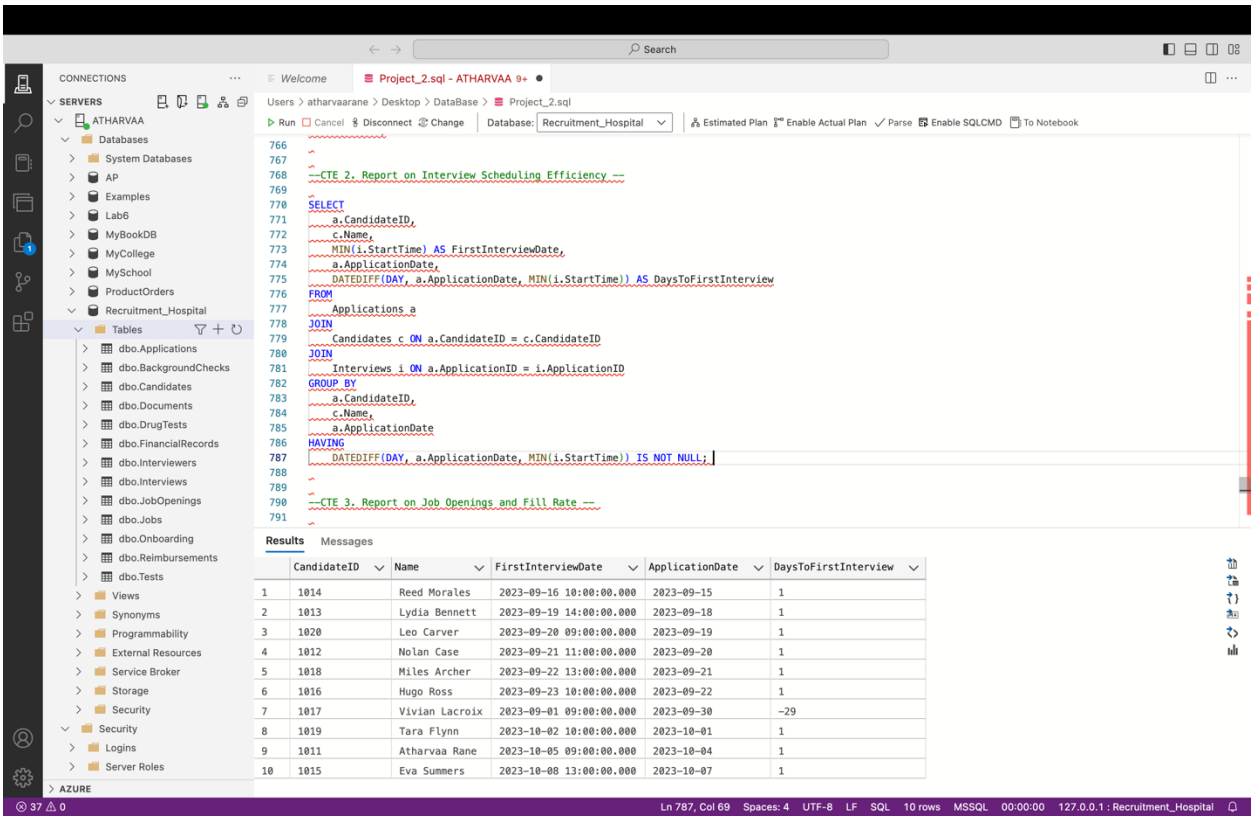
CTE1:
The SQL script uses named CandidateStatus to retrieve the most recent application details for each candidate from the Candidates and Applications tables, ranked by the latest application date, and selects the details of the candidates with the most recent application.

```
WITH CandidateStatus AS (
    SELECT
        c.CandidateID,
        c.Name,
        c.Email,
        a.ApplicationDate,
        a.Status,
        DENSE_RANK() OVER (PARTITION BY a.CandidateID ORDER BY a.ApplicationDate DESC)
AS Rank
    FROM
        Candidates c
    JOIN
        Applications a ON c.CandidateID = a.CandidateID
)
SELECT
    CandidateID,
    Name,
    Email,
    ApplicationDate,
    Status
FROM
    CandidateStatus
WHERE
    Rank = 1;
```



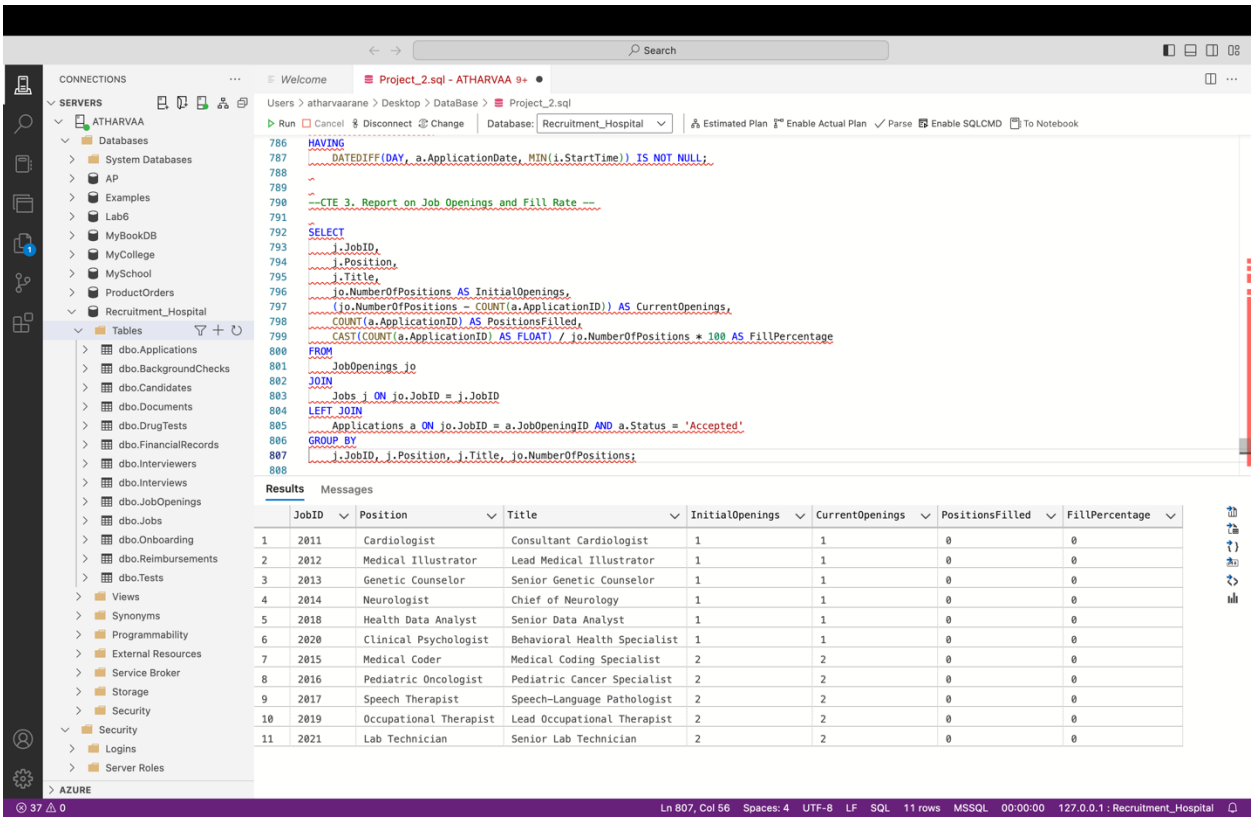
CTE2:
Calculates the days between each candidate's application date and their first interview, displaying the candidate ID, name, earliest interview date, application date, and days to the first interview. It filters to include only those records where the first interview date is not null.

```
SELECT
    a.CandidateID,
    c.Name,
    MIN(i.StartTime) AS FirstInterviewDate,
    a.ApplicationDate,
    DATEDIFF(DAY, a.ApplicationDate, MIN(i.StartTime)) AS DaysToFirstInterview
FROM
    Applications a
JOIN
    Candidates c ON a.CandidateID = c.CandidateID
JOIN
    Interviews i ON a.ApplicationID = i.ApplicationID
GROUP BY
    a.CandidateID,
    c.Name,
    a.ApplicationDate
HAVING
    DATEDIFF(DAY, a.ApplicationDate, MIN(i.StartTime)) IS NOT NULL;
```



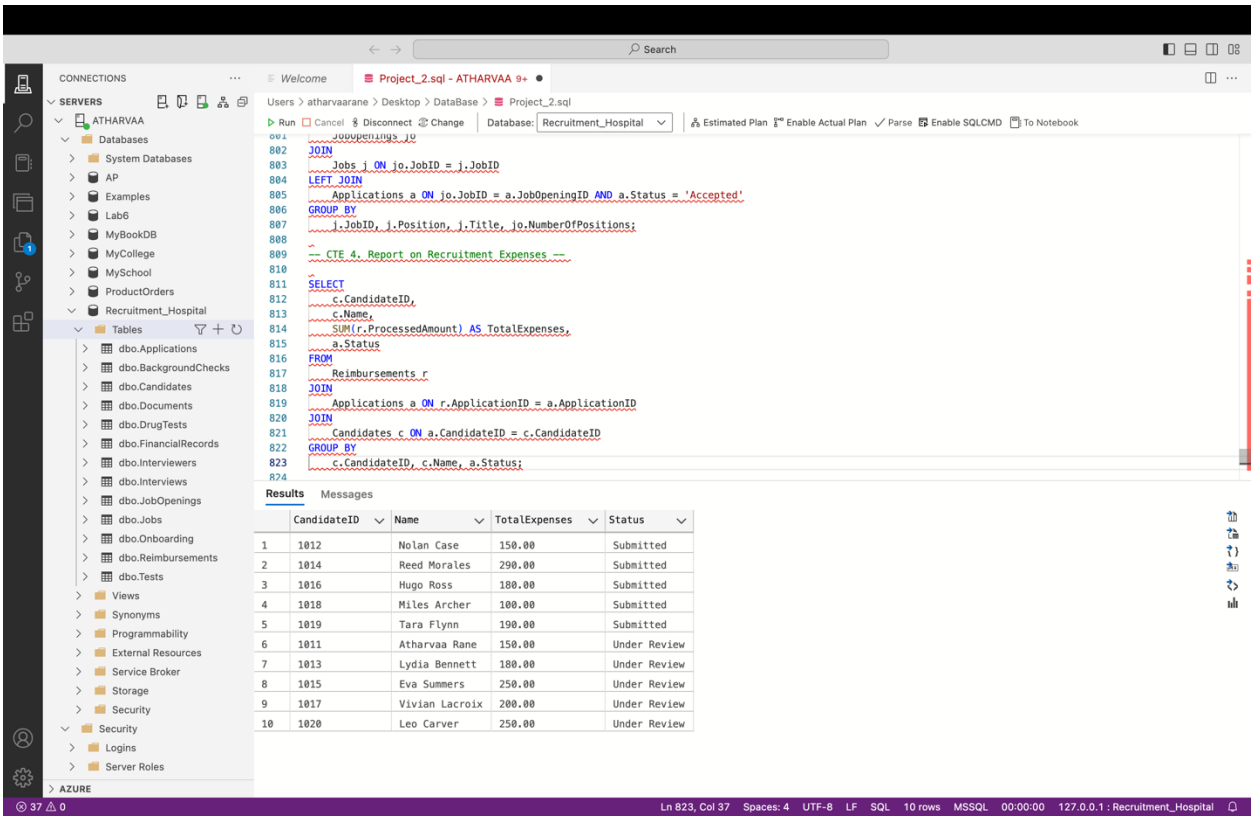
CTE3:
The SQL query aggregates job opening data by calculating initial openings, current openings, positions filled, and the fill percentage for each job. It joins the `JobOpenings`, `Jobs`, and `Applications` tables, filtering by applications with an 'Accepted' status, and groups the results by job-specific attributes to provide insights into the hiring progress for each position.

```
SELECT
    j.JobID,
    j.Position,
    j.Title,
    jo.NumberOfPositions AS InitialOpenings,
    (jo.NumberOfPositions - COUNT(a.ApplicationID)) AS CurrentOpenings,
    COUNT(a.ApplicationID) AS PositionsFilled,
    CAST(COUNT(a.ApplicationID) AS FLOAT) / jo.NumberOfPositions * 100 AS
FillPercentage
FROM
    JobOpenings jo
JOIN
    Jobs j ON jo.JobID = j.JobID
LEFT JOIN
    Applications a ON jo.JobID = a.JobOpeningID AND a.Status = 'Accepted'
GROUP BY
    j.JobID, j.Position, j.Title, jo.NumberOfPositions;
```



CTE4:
The SQL query aggregates total reimbursement expenses for each candidate by joining the `Reimbursements`, `Applications`, and `Candidates` tables, grouping the results by candidate ID, name, and application status to display expenses associated with each application status.

```
SELECT
    c.CandidateID,
    c.Name,
    SUM(r.ProcessedAmount) AS TotalExpenses,
    a.Status
FROM
    Reimbursements r
JOIN
    Applications a ON r.ApplicationID = a.ApplicationID
JOIN
    Candidates c ON a.CandidateID = c.CandidateID
GROUP BY
    c.CandidateID, c.Name, a.Status;
```


Search
CONNECTIONS
SERVERS
ATHARVAA
Databases
System Databases
AP
Examples
Lab6
MyBookDB
MyCollege
MySchool
ProductOrders
Recruitment_Hospital
Tables
dbo.Applications
dbo.BackgroundChecks
dbo.Candidates
dbo.Documents
dbo.DrugTests
dbo.FinancialRecords
dbo.Interviewers
dbo.Interviews
dbo.JobOpenings
dbo.Jobs
dbo.Onboarding
dbo.Reimbursements
dbo.Tests
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
Security
Logins
Server Roles
AZURE
Welcome
Project_2.sql - ATHARVAA 9+
Users > atharvaarane > Desktop > DataBase > Project_2.sql
Run
Cancel
Disconnect
Change
Database: Recruitment_Hospital
Estimated Plan
Enable Actual Plan
Parse
Enable SQLCMD
To Notebook
JOIN
Jobs j ON jo.JobID = j.JobID
LEFT JOIN
Applications a ON jo.JobID = a.JobOpeningID AND a.Status = 'Accepted'
GROUP BY
j.JobID, j.Position, j.Title, jo.NumberOfPositions;
-- CTE 4. Report on Recruitment Expenses --
SELECT
c.CandidateID,
c.Name,
SUM(r.ProcessedAmount) AS TotalExpenses,
a.Status
FROM
Reimbursements r
JOIN
Applications a ON r.ApplicationID = a.ApplicationID
JOIN
Candidates c ON a.CandidateID = c.CandidateID
GROUP BY
c.CandidateID, c.Name, a.Status;
Results
Messages
CandidateID
Name
TotalExpenses
Status
1 1012 Nolan Case 150.00 Submitted
2 1014 Reed Morales 290.00 Submitted
3 1016 Hugo Ross 180.00 Submitted
4 1018 Miles Archer 100.00 Submitted
5 1019 Tara Flynn 190.00 Submitted
6 1011 Atharvaa Rane 150.00 Under Review
7 1013 Lydia Bennett 180.00 Under Review
8 1015 Eva Summers 250.00 Under Review
9 1017 Vivian Lacroix 200.00 Under Review
10 1020 Leo Carver 250.00 Under Review
37 0
Ln 823, Col 37
Spaces: 4
UTF-8
LF
SQL
10 rows
MSSQL
00:00:00
127.0.0.1 : Recruitment_Hospital

BUSINESS REPORTS

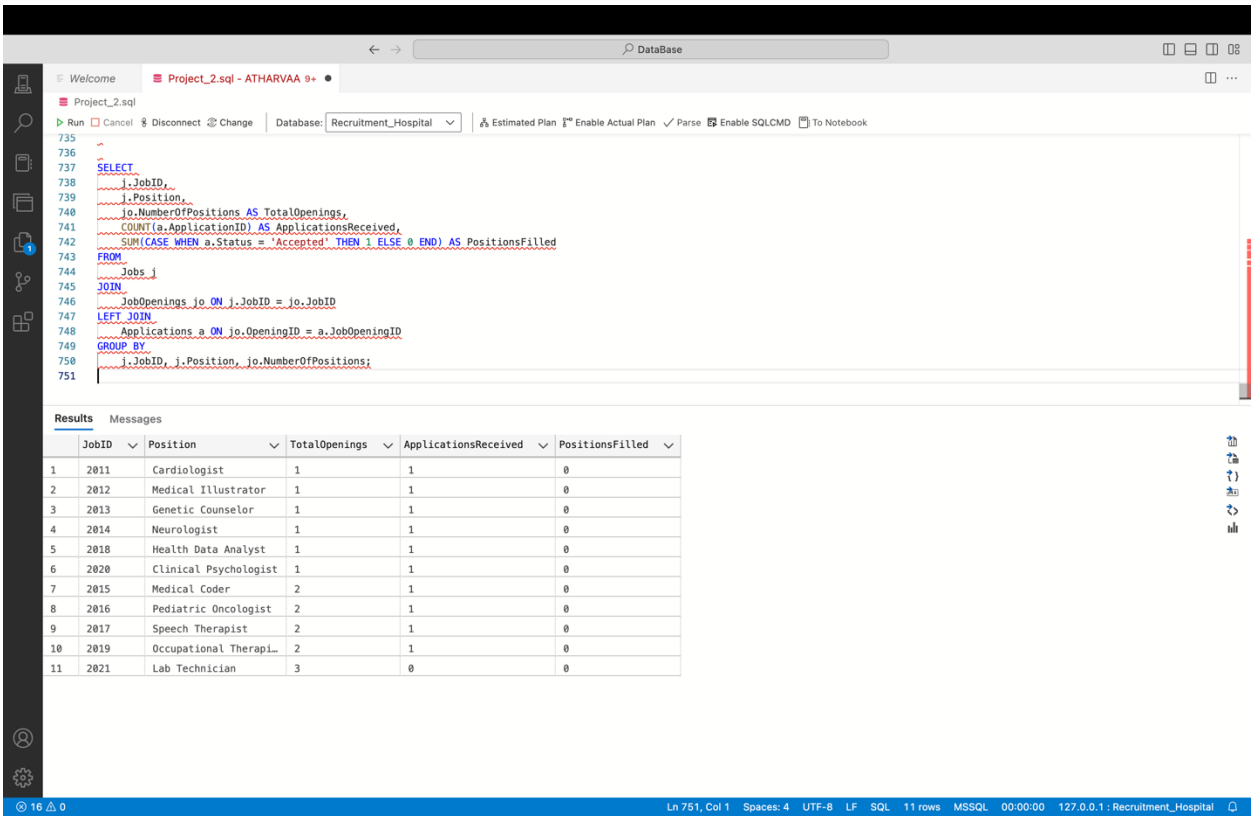
Business Report 1:

Job Opening Fulfillment Rate Report
This report shows the number of openings versus how many positions have been filled, providing insights into how well different job positions are being filled.

```
SELECT
    j.JobID,
    j.Position,
    jo.NumberOfPositions AS TotalOpenings,
    COUNT(a.ApplicationID) AS ApplicationsReceived,
    SUM(CASE WHEN a.Status = 'Accepted' THEN 1 ELSE 0 END) AS PositionsFilled
FROM
    Jobs j
JOIN
    JobOpenings jo ON j.JobID = jo.JobID
LEFT JOIN
    Applications a ON jo.OpeningID = a.JobOpeningID
GROUP BY
    j.JobID, j.Position, jo.NumberOfPositions;
```

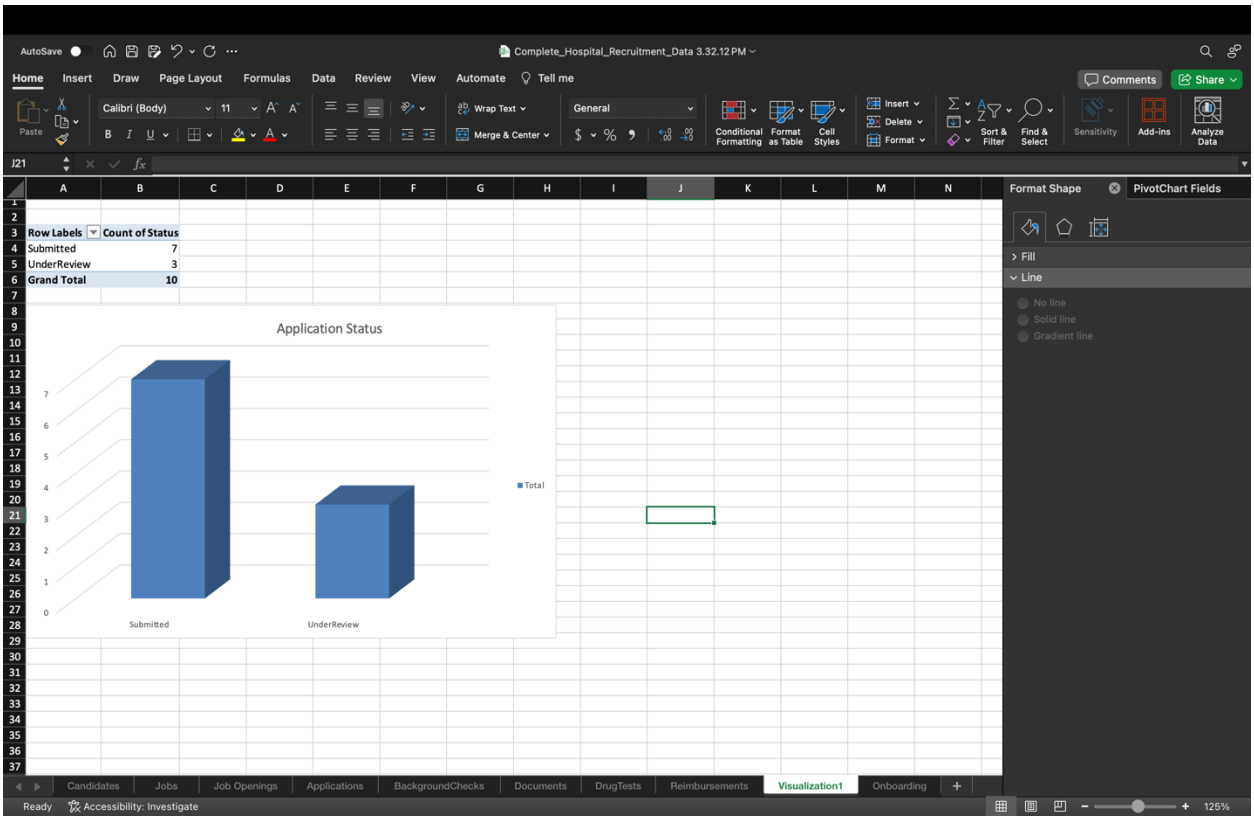
Description:

- **JobID** and **Position**: Identify the job and its title.
- **TotalOpenings**: Total number of job openings available.
- **ApplicationsReceived**: Total applications received for each job.
- **PositionsFilled**: Number of positions filled (where application status is 'Accepted').



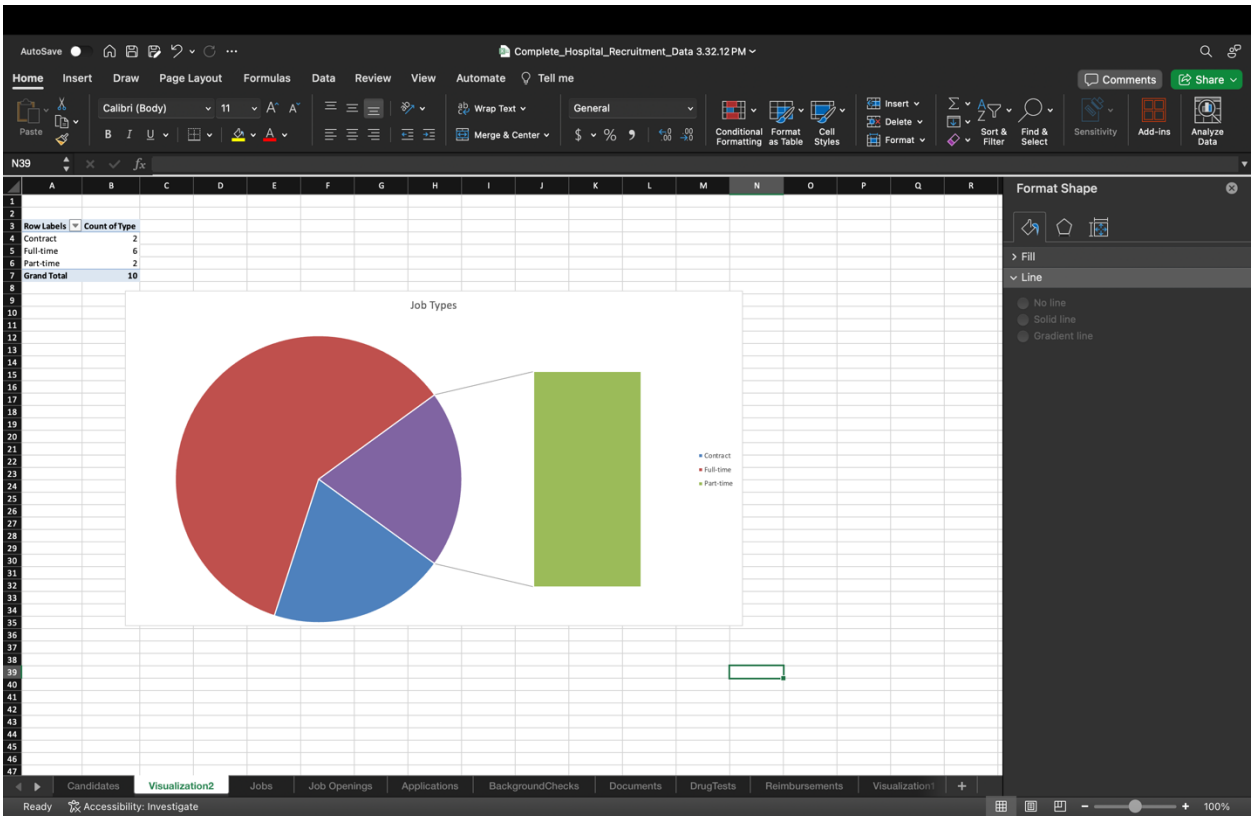
Business Report2:

The graph displayed provides a visual representation of the status of job applications within the recruitment database. It shows that there are a total of 10 applications, with 7 of them in the 'Submitted' stage and 3 in the 'Under Review' stage. This suggests that the majority of applications are still awaiting initial review, indicating either a backlog or a recent influx of applications. The substantial difference between the two statuses might necessitate a review of staffing levels or process efficiency to ensure timely processing of applications. This data can be critical for HR to optimize their workflow and reduce time-to-hire, thereby improving the overall recruitment process efficiency. The graph, included as part of a dashboard in Excel, serves as a practical tool for real-time monitoring and management of application statuses.



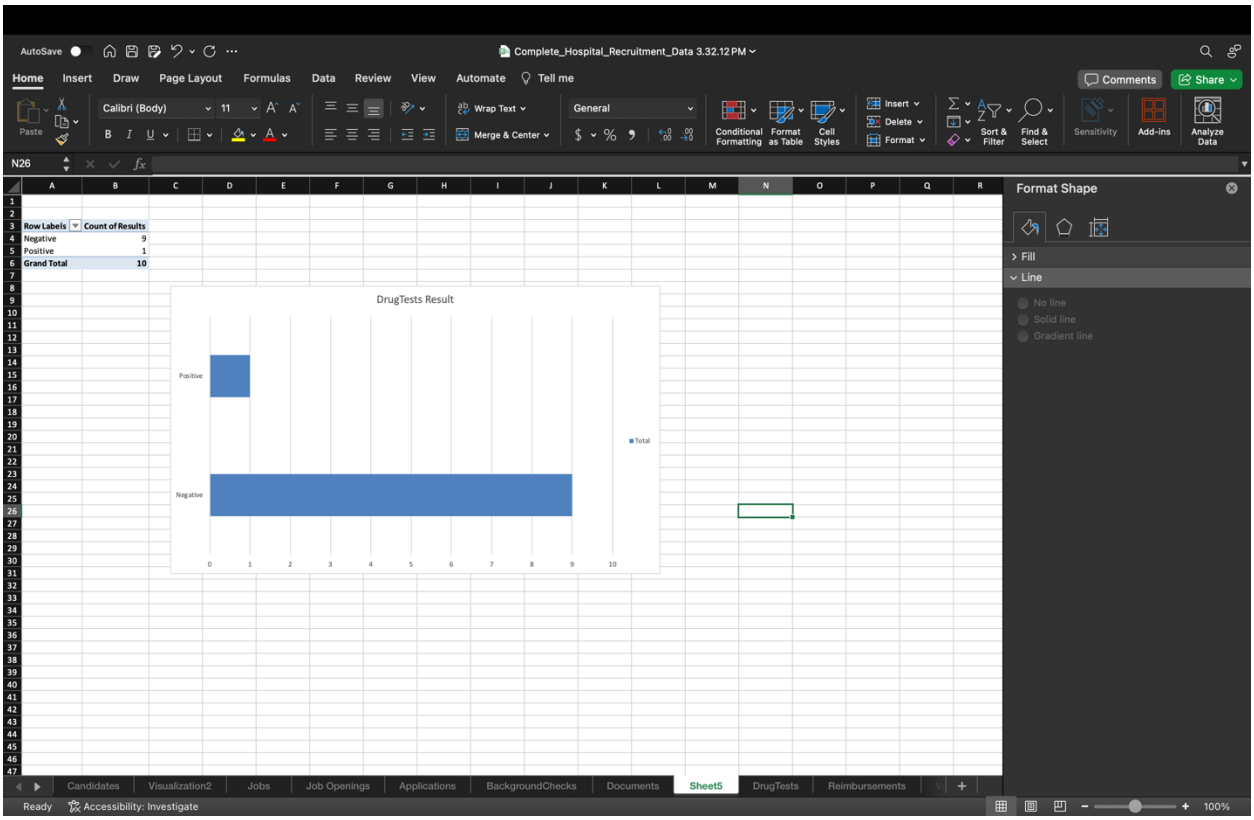
Business Report3:

The pie chart provides a distribution of job types within the healthcare recruitment system. It reveals that full-time positions are the most prevalent, constituting 6 out of 10 total job openings, indicating a strong demand for permanent staff. Contract roles account for 2 positions, reflecting a moderate reliance on temporary staffing solutions, which might be used to address seasonal fluctuations or project-specific needs. Part-time roles make up the remaining 2 positions, suggesting limited opportunities for flexible working arrangements. This data could help HR to understand current hiring trends and possibly reevaluate job offerings to align with organizational needs and market conditions. The visualization assists in quickly identifying which job types dominate the recruitment efforts, enabling strategic planning and resource allocation.



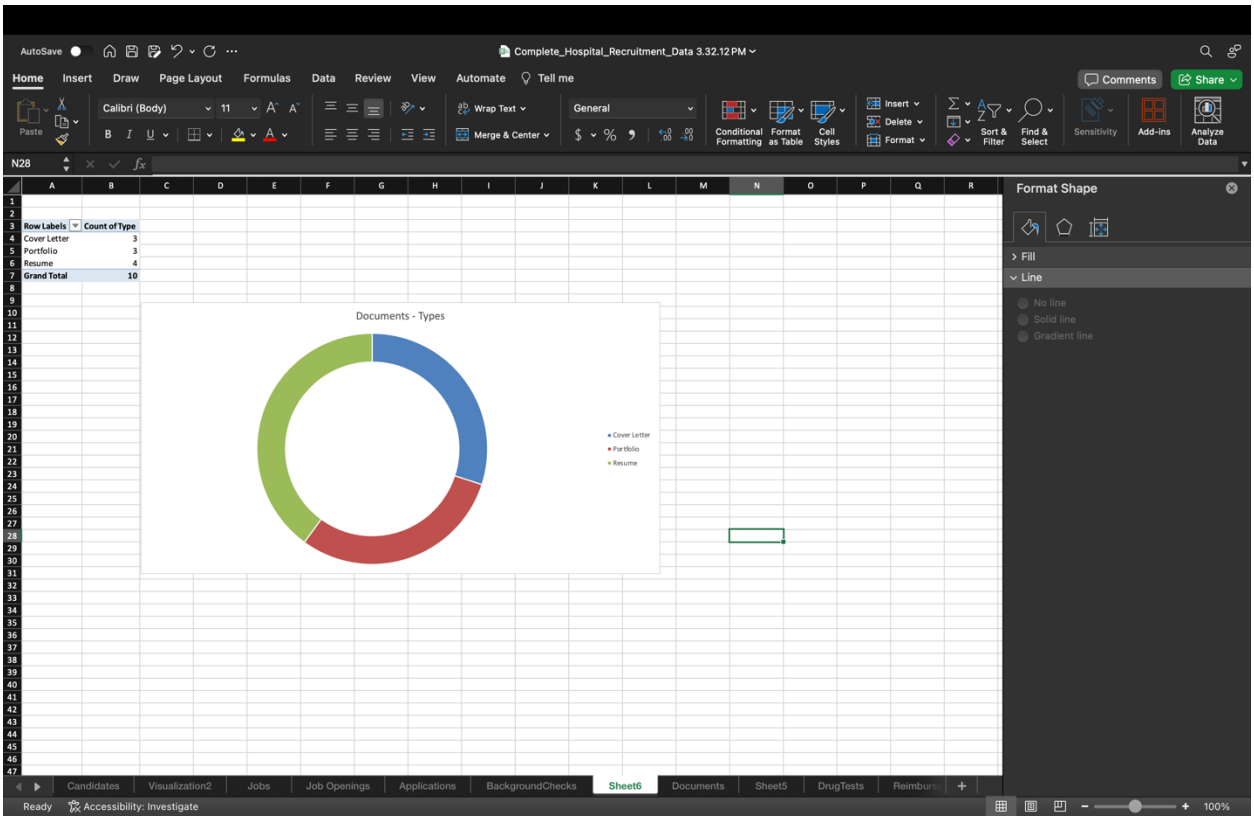
Business Report4:

The bar chart depicts the results of drug tests conducted during the recruitment process, displaying a significant disparity between negative and positive results. Out of 10 recorded tests, 9 results are negative, underscoring a predominantly drug-free candidate pool, which is reassuring for maintaining workplace safety and compliance standards. Only 1 result returned positive, indicating a potential concern that may require further investigation or immediate attention to ensure adherence to the organization's health and safety policies. This low incidence of positive results may reflect effective pre-screening or deterrence measures but suggests that continuous monitoring and a stringent drug policy are essential. The graph is a valuable tool for HR to monitor the efficacy of current drug testing protocols and maintain high standards of employee health and safety within the organization.



Business Report 5:

The donut chart illustrates the types of documents submitted by candidates during the recruitment process. It highlights that resumes constitute half of all documents, emphasizing their importance in initial candidate assessments. Cover letters account for 30% of submissions, supporting the comprehensive evaluation of candidates by providing additional context to their applications. The remaining 20% consists of portfolios, likely specific to roles requiring demonstrable work, such as in design or project management. This distribution underscores the importance of a varied document submission strategy that allows HR to gain a holistic view of applicants' qualifications and experiences. Such insights are crucial for effective screening and selecting the right candidates for interviews, ensuring the recruitment process aligns with organizational needs and job specifications.


Complete_Hospital_Recruitment_Data 3.32.12 PM
Row Labels
Count of Type
Cover Letter 3
Portfolio 3
Resume 4
Grand Total 10
Documents - Types
Cover Letter
Portfolio
Resume
Format Shape
Fill
Line
No line
Solid line
Gradient line
Candidates
Visualization2
Jobs
Job Openings
Applications
BackgroundChecks
Sheet6
Documents
Sheet5
DrugTests
Ready
Accessibility: Investigate
100%

CONCLUSION

This project successfully developed a tailored recruitment database for a University Medical Center, significantly enhancing the efficiency and transparency of the recruitment process. By automating key tasks and centralizing data management, the system not only streamlines operations but also empowers HR professionals to focus more on strategic engagement and less on administrative burdens. The implementation of robust security measures ensures that sensitive information remains protected, supporting the organization's commitment to privacy and compliance. Overall, the new database system is poised to significantly improve the recruitment lifecycle, contributing to better hiring decisions and fostering a skilled healthcare workforce.