**Medical Appointment System Analysis**

**1. Downloading Dataset**

The dataset was downloaded from Kaggle, specifically from the following link: [Kaggle - Exploratory Data Analysis of Medical Appointment System Dataset](https://www.kaggle.com/code/carogonzalezgaltier/eda-medical-appointment-system-dataset#Exploratory-Data-Analysis-of-Medical-Appointment-System-Dataset). This dataset provides information about medical appointments, including patient details, appointment dates, and attendance status. The dataset is in CSV format and contains multiple fields such as appointment dates, patient IDs, and no-show indicators.

**2. Server and Table Creation in SSMS**

A SQL Server database was configured using SQL Server Management Studio (SSMS). The following steps were performed:

**2.1. Database Creation**

A new database named MedicalAppointments was created in SSMS.

**2.2. Table Creation**

The database was structured to include the following tables:

1. **Appointment Table**:
   * **Columns**:
     + AppointmentID (Primary Key, INT, Auto Increment)
     + PatientID (Foreign Key, INT)
     + SlotID (Foreign Key, INT)
     + AppointmentDate (DATETIME)
     + Status (VARCHAR(50))
   * **Purpose**: Stores details of appointments, including their date, slot, and status (e.g., "Attended," "Cancelled").
2. **Patients Table**:
   * **Columns**:
     + PatientID (Primary Key, INT, Auto Increment)
     + Name (VARCHAR(255))
     + Age (INT)
     + Gender (CHAR(1))
     + Insurance (VARCHAR(255))
   * **Purpose**: Maintains patient demographic information and contact details.
3. **Slots Table**:
   * **Columns**:
     + SlotID (Primary Key, INT, Auto Increment)
     + SlotTime (TIME)
     + IsAvailable (BIT)
   * **Purpose**: Tracks the availability of time slots for appointments.

**3. Importing Dataset into SSMS**

The downloaded dataset was processed and imported into the SQL Server tables. The following steps were taken:

**3.1. Data Transformation**

The CSV file was preprocessed to align with the structure of the database tables. This included:

* Splitting data into multiple files to map to Patients, Appointment, and Slots tables.
* Standardizing column names to match the database schema.
* Handling null values by replacing them with appropriate defaults.

**3.2. Data Import**

The SSMS Import and Export Wizard was utilized to load data into the respective tables. During the import:

* Primary and foreign key constraints were maintained.
* Data types were validated to ensure compatibility with table schemas.
* Error logs were monitored to address data inconsistencies.

**4. Connecting SSMS to Power BI**

The database was connected to Power BI for advanced analysis and visualization. Steps involved:

**4.1. Data Source Connection**

* Selected **SQL Server** as the data source.
* Entered server details and selected the MedicalAppointments database.
* Chose **ImportMode** mode for data analysis.

**4.2. Data Loading**

* Imported the Appointment, Patients, and Slots tables into Power BI.
* Validated data load and refreshed the connection to ensure accuracy.

**. Data Modeling and Data Type Assignment**

In Power BI, a comprehensive data model was created:

**5.1. Relationships**

* Defined relationships as follows:
  + Patients.PatientID ⟶ Appointments.PatientID (One-to-Many)
  + Slots.SlotID ⟶ Appointments.SlotID (One-to-Many)

**5.2. Data Type Assignment**

* AppointmentDate: Changed to DATETIME format.
* SlotTime: Assigned TIME data type.
* Age: Ensured as an INTEGER.
* Status: Set as STRING/VARCHAR for categorical representation.

**6. Pages and Visualizations in Power BI**

Three analytical pages were designed:

**a. Slot Availability**

* **Objective**: Provide an overview of available slots and their timings.
* **Technical Details**:
  + **Visuals**:
    - Bar Chart: Slots grouped by SlotTime with IsAvailable as the measure.
    - Table: Detailed listing of slot availability.
  + **Filters**:
    - Date filter to limit results to a specific day.
    - Availability filter to show only available slots.

**b. Upcoming Appointments**

* **Objective**: Show insights into future appointments and cancellations.
* **Technical Details**:
  + **Visuals**:
    - Pie Chart: Proportion of appointment statuses (attended vs cancelled).
    - Table: Detailed view of upcoming and cancelled appointments.
  + **Filters**:
    - Date range for upcoming days.

**c. Forecast**

* **Objective**: Predict the expected flow of patients over the next 10 days.
* **Technical Details**:
  + **Visuals**:
    - Line Chart: Time series forecast using Power BI’s inbuilt ARIMA forecasting model.
    - Confidence Intervals: Displayed using shaded regions on the chart.
  + **Parameters**:
    - Confidence level set to 95%.
    - Historical data range used for prediction: Last 30 days.

**7. Importing Dataset into Python for EDA**

The dataset was imported into Python for comprehensive exploratory data analysis.

**7.1. Data Import**

* **Libraries Used**: Pandas, NumPy.
* **File Format**: CSV.

**7.2. Data Cleaning**

* Checked for missing values using data.isnull().sum().
* Handled missing values by imputing or dropping rows as needed.
* Standardized column names to lowercase for consistency.

**7.3. Exploratory Analysis**

* **Distribution Analysis**
* **Trend Analysis**
* **Age Group Analysis**