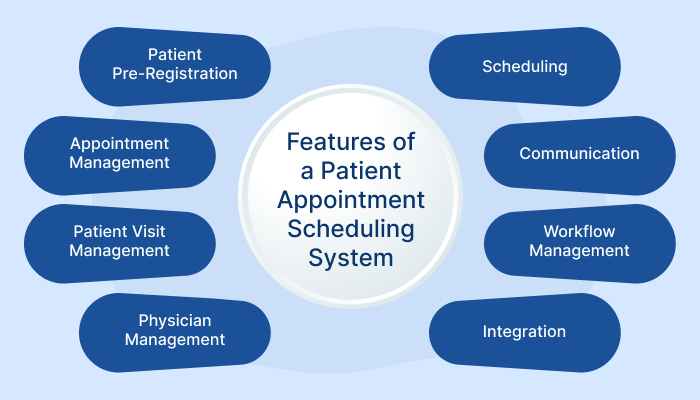
# **Hospital Information System**

A Patient Management System and an Appointment Management System are integral components of healthcare information technology that help healthcare providers streamline their operations, enhance patient care, and improve overall efficiency. Let's delve into each system.

# Appointment Management System

An Appointment Management System is a specialized component of a Patient Management System or a standalone software solution that focuses on efficiently handling the scheduling and management of patient appointments within a healthcare facility.



*Key Features:*

1. Appointment Scheduling: Allows staff to schedule, reschedule, and cancel appointments based on availability.
2. Automated Reminders: Sends automated reminders to patients about upcoming appointments through various communication channels (email, SMS, etc.).
3. Resource Allocation: Manages the allocation of resources such as examination rooms, equipment, and healthcare providers based on appointment schedules.
4. Patient Check-In/Check-Out: Facilitates the smooth check-in and check-out process for patients attending appointments.
5. Waitlist Management: Manages a waitlist for patients seeking earlier appointments or cancellations.
6. Integration with Other Systems: Integrates seamlessly with Electronic Health Record (EHR) systems and other components of the healthcare information infrastructure.

*Benefits:*

1. Optimized Scheduling: Improves the efficiency of appointment scheduling, reducing wait times and optimizing resource utilization.
2. Enhanced Patient Experience: Provides convenient and timely communication, reducing no-show rates and improving patient satisfaction.
3. Resource Efficiency: Ensures optimal use of healthcare facility resources, minimizing idle time and maximizing productivity.
4. Data Accuracy: Reduces errors associated with manual appointment scheduling and enhances the accuracy of patient records.

## Tables loaded

**1. Core Patients:**

- Contains detailed patient information such as age, gender, citizenship, insurance status, etc.

**2. Core Treatment Visits:**

- Includes comprehensive details of visits, including whether they are scheduled or walk-in, consultation name, paediatric or pregnant status, referral status, and eligibility for insurance schemes.

**3. Core Treatment Visit Status History:**

- Stores the history of visit statuses, providing a record of changes over time.

**4. Core Users:**

- Holds employee information for Healthcare Providers (HCP), including employee ID, identity ID, user type ID, super admin status, and user credentials.

**5. HR Departments:**

- Contains details of departments within an HCP, including department name, active status, clinic status, specialties ID, and eye care designation.

**6. HR Branches:**

- Stores data related to branches, including branch name, branch code, address (including country), default branch status, active status, phone, and email.

**7. HR Employees:**

- Provides comprehensive employee details such as name, date of joining (DOJ), basic salary, department ID, branch ID, country ID, employee ID, citizenship status, termination status, marital status, nationality, religion, passport number, and work type ID.

**8. Insurance InsuranceProvider:**

- Contains information about insurance companies, including company name, insurance provider ID, provider type, address (with phone number), TPA ID, active status, and policy number.

**9. Lookup Countries:**

- Stores information about countries, including country ID, country name, and country ISO number.

## Transformations Done

Core Patients

**1. Gender Description Column:**

- Add a new column named "Gender Description."

- Use the following logic:

- If Gender column value is 1, set "Male" in the Gender Description column.

- If Gender column value is 2, set "Female" in the Gender Description column.

- For any other values in the Gender column, set "Other" in the Gender Description column.

**2. Residence Type Description Column:**

- Add a new column named "Residence Type Description."

- Use the following logic:

- If Residence Type column value is 1, set "Citizen" in the Residence Type Description column.

- For any other values in the Residence Type column, set "Visitor" in the Residence Type Description column.

**3. Blood Group Description Column:**

- Add a new column named "Blood Group Description."

- Use the following logic:

- If Blood Group column value is 0, set "O -ve" in the Blood Group Description column.

- If Blood Group column value is 1, set "O +ve" in the Blood Group Description column.

- If Blood Group column value is 2, set "A -ve" in the Blood Group Description column.

- If Blood Group column value is 3, set "A +ve" in the Blood Group Description column.

- If Blood Group column value is 4, set "B -ve" in the Blood Group Description column.

- If Blood Group column value is 5, set "B +ve" in the Blood Group Description column.

- If Blood Group column value is 6, set "AB -Ve" in the Blood Group Description column.

- If Blood Group column value is 7, set "AB +ve" in the Blood Group Description column.

**4. Marital Status Description Column:**

- Add a new column named "Marital Status Description."

- Use the following logic:

- If Marital Status column value is 1, set "Single" in the Marital Status Description column.

- If Marital Status column value is 2, set "Married" in the Marital Status Description column.

- If Marital Status column value is 3, set "Divorced" in the Marital Status Description column.

- If Marital Status column value is 4, set "Widow" in the Marital Status Description column.

- If Marital Status column value is 5, set "Separated" in the Marital Status Description column.

- For any other values in the Marital Status column, set null in the Marital Status Description column.

**5. Remove Unnecessary Columns:**

- Remove any unnecessary columns from the dataset.

Core TreatmentVisits

**1. Visit Status Description Column:**

- Add a new column named "VisitStatusDescription."

- Use the following logic:

- If Visit Status column value is 1, set "Confirm" in the VisitStatusDescription column.

- If Visit Status column value is 2, set "Completed" in the VisitStatusDescription column.

- If Visit Status column value is 3, set "Cancelled" in the VisitStatusDescription column.

- If Visit Status column value is 4, set "Arrived" in the VisitStatusDescription column.

- If Visit Status column value is 5, set "Schedule" in the VisitStatusDescription column.

- If Visit Status column value is 6, set "NoShow" in the VisitStatusDescription column.

- If Visit Status column value is 7, set "Attended" in the VisitStatusDescription column.

- For any other values in the Visit Status column, set null in the VisitStatusDescription column.

**2. Visit Type Description Column:**

- Add a new column named "VisitTypeDescription."

- Use the following logic:

- If Visit Type column value is 1, set "New Patient" in the VisitTypeDescription column.

- If Visit Type column value is 2, set "New Visit" in the VisitTypeDescription column.

- If Visit Type column value is 3, set "Follow-up" in the VisitTypeDescription column.

- For any other values in the Visit Type column, set null in the VisitTypeDescription column.

**3. Arrival Status Description Column:**

- Add a new column named "ArrivalStatusDescription."

- Use the following logic:

- If Arrival Status column value is 1, set "On Time" in the ArrivalStatusDescription column.

- If Arrival Status column value is 2, set "Late" in the ArrivalStatusDescription column.

- If Arrival Status column value is 3, set "Early" in the ArrivalStatusDescription column.

- For any other values in the Arrival Status column, set null in the ArrivalStatusDescription column.

**4. EligibilityTypeID Status Column:**

- Add a new column named "EligibilityTypeIDStatus."

- Use the following logic:

- If EligibilityTypeID column value is 1, set "Cash" in the EligibilityTypeIDStatus column.

- If EligibilityTypeID column value is 2, set "Insurance" in the EligibilityTypeIDStatus column.

- If EligibilityTypeID column value is 3, set "Hospital" in the EligibilityTypeIDStatus column.

- If EligibilityTypeID column value is 4, set "Non-Insurance Companies" in the EligibilityTypeIDStatus column.

- For any other values in the EligibilityTypeID column, set null in the EligibilityTypeIDStatus column.

**5. Extracted Date\_only Column:**

- Add a new column named "Date\_only."

- Extract the date part from the "VisitDateTime" column and set it in the "Date\_only" column.

**6. Merged Tables:**

- Merge the "Core TreatmentVisits" table with the "Core Users" table based on the "UserID" using the left outer join method.

- Use the "EmployeeID" column from "Core Users."

**7. Join with HR Employees:**

- Join the resultant merged table with the "HR Employees" table using a left outer join to add the "FullNameEn" column.

**8. Visit Appointment Type Description Column:**

- Add a new column named "VisitAppointmentTypeDescription."

- Use the following logic:

- If Visit Appointment Type column value is 0, set "Appointment Visit" in the VisitAppointmentTypeDescription column.

- If Visit Appointment Type column value is 1, set "OR Appointment Visit" in the VisitAppointmentTypeDescription column.

- If Visit Appointment Type column value is 2, set "InPatient" in the VisitAppointmentTypeDescription column.

- If Visit Appointment Type column value is 3, set "Emergency Visit" in the VisitAppointmentTypeDescription column.

- If Visit Appointment Type column value is 4, set "OR Visit" in the VisitAppointmentTypeDescription column.

- If Visit Appointment Type column value is 5, set "HomeCare" in the VisitAppointmentTypeDescription column.

- For any other values in the Visit Appointment Type column, set "None."

**9. IsReferral Description Column:**

- Add a new column named "Referred\_."

- Use the following logic:

- If IsReferral column is TRUE, set "Referred" in the Referred\_ column.

- If IsReferral column is FALSE, set "Non Referred" in the Referred\_ column.

- For any other values in the IsReferral column, set null in the Referred\_ column.

- Order the dataset by descending order based on the "Referred\_" column.

Core TreatmentVisitStatusHistory

**Remove Unnecessary Columns:**

Identified columns in the dataset that were formed as a result of merging with another table.

Removed the identified unnecessary columns from the dataset to streamline it and reduce complexity.

Core Users

* No transformations done

HR\_Departments

* No transformations done

HR Branches

* No transformations done

HR Employees

* No transformations done

Insurance InsuranceProvider

**Remove Unnecessary Columns:**

Identified columns in the dataset that were formed as a result of merging with another table.

Removed the identified unnecessary columns from the dataset to streamline it and reduce complexity.

Lookup Countries

**Remove Unnecessary Columns:**

Identified columns in the dataset that were formed as a result of merging with another table.

Removed the identified unnecessary columns from the dataset to streamline it and reduce complexity.

## Data Modelling

**1. Core TreatmentVisits Connected with Calendar Table:**

- Connect the "Core TreatmentVisits" table with the "Calendar" table using a many-to-one relationship.

- Establish the connection based on the "Date" column in the "Calendar" table and date only column in the core treatmentvisits table.

**2. Core TreatmentVisits Connected with Core TreatmentStatusHistory:**

- Connect the "Core TreatmentVisits" table with the "Core TreatmentStatusHistory" table using a one-to-many relationship.

- Establish the relationship based on the common column "TreatmentVisitID."

**3. Core TreatmentVisits Connected with HR\_Department:**

- Connect the "Core TreatmentVisits" table with the "HR\_Department" table using a many-to-one relationship.

- Establish the relationship based on the common column "DepartmentID."

**4. HR\_Department Connected with HR Branches:**

- Connect the "HR\_Department" table with the "HR Branches" table using a many-to-one relationship.

- Establish the relationship based on the common column "BranchID."

**5. Core TreatmentVisits Connected with HR Employees:**

- Connect the "Core TreatmentVisits" table with the "HR Employees" table using a many-to-one relationship.

- Establish the relationship based on the common column "DoctorUsedID" in the "Core TreatmentVisits" table and "EmployeeID" in the "HR Employees" table.

- Set the cross-filter direction as both.

**6. HR Employees Connected to Core Users:**

- Connect the "HR Employees" table with the "Core Users" table using a one-to-many relationship.

- Establish the relationship based on the common column "EmployeeID."

**7. Core TreatmentVisits Connected with Core Patients:**

- Connect the "Core TreatmentVisits" table with the "Core Patients" table using a many-to-one relationship.

- Establish the relationship based on the common column "PatientID."

**8. Core Patients Connected with Lookups Countries:**

- Connect the "Core Patients" table with the "Lookups Countries" table using a many-to-one relationship.

- Establish the relationship based on the common column "NationalityID" in the "Core Patients" table and "CountryID" in the "Lookups Countries" table.

**9. Core Patients Connected with Insurance InsuranceProviders:**

- Connect the "Core Patients" table with the "Insurance InsuranceProviders" table using a many-to-one relationship.

- Establish the relationship based on the common column "InsuranceCompanyID" in the "Core Patients" table and "InsuranceProviderID" in the "Insurance InsuranceProviders" table.

## DAX Formulas Used:

|  |  |
| --- | --- |
| **Measure Name** | **DAX Formula** |
| Total Visits | [Walkins] + [Appoinment Visits] |
| Cancelled  Appointments | var result = CALCULATE( COUNT('Core TreatmentVisits'[TreatmentVisitID]),  'Core TreatmentVisits'[VisitStatus] = 3, -- Cancelled  'Core TreatmentVisits'[AddedManually] = FALSE() )  RETURN(COALESCE(result,0)) |
| % Cancellation Rate | VAR temp = [Cancelled Appointments] / [Appoinment Visits]  -- visits cancelled from total appointments RETURN   IF(ISBLANK(temp) || ISERROR(temp), 0, temp) |
| Walkins | CALCULATE (  COUNT ( 'Core TreatmentVisits'[TreatmentVisitID] ),  -- Walkins  (  'Core TreatmentVisits'[VisitAppoitmentType] = 0  || 'Core TreatmentVisits'[VisitAppoitmentType] = BLANK ()  )  && 'Core TreatmentVisits'[AddedManually] = TRUE () ) |
| No Show Appointments | CALCULATE( COUNT('Core TreatmentVisits'[TreatmentVisitID]),  'Core TreatmentVisits'[VisitStatus] =6, -- NoShow  'Core TreatmentVisits'[AddedManually] = FALSE() ) |
| Attended Appointments | var result = CALCULATE( COUNT('Core TreatmentVisits'[TreatmentVisitID]),  'Core TreatmentVisits'[VisitStatus] =7, -- Attended with the doctor  'Core TreatmentVisits'[AddedManually] = FALSE() )  RETURN(COALESCE(result,0)) |
| Completed Appointments | var result = CALCULATE( COUNT('Core TreatmentVisits'[TreatmentVisitID]),  'Core TreatmentVisits'[VisitStatus] =2 , -- Completed visit  'Core TreatmentVisits'[AddedManually] = FALSE() )  RETURN(COALESCE(result,0)) |
| % No Show Rate | COALESCE([No Show Appointments] / [Appoinment Visits] , 0)  -- poeple failed to turn up to the scheduled appointment |
| Appoinment Visits | CALCULATE (  COUNT ( 'Core TreatmentVisits'[TreatmentVisitID] ),   -- visits scheduled to meet up with doctors (in future)    (  'Core TreatmentVisits'[VisitAppoitmentType] = 0  || 'Core TreatmentVisits'[VisitAppoitmentType] = BLANK ()  )  && 'Core TreatmentVisits'[AddedManually] = FALSE() ) |
| visit\_Category | var cat1 = "new patient visit : initial or first time visits | "  var cat2 = " new visit : Subsequent visits"  var cat3 = "follow up visits | "  return lower(cat1 & cat3 & cat2) |
| Visit Adherence Rate | var \_sum = [Attended Appointments] + [Completed Appointments]  -- Sum of attended and completed  RETURN   COALESCE(DIVIDE(\_sum, [Total Visits],0),0) |
| Clinic\_time\_diff | DATEDIFF( CALCULATE(  MAX('Core TreatmentVisitStatusHistory'[CreationDate]),  -- Calculating time difference  'Core TreatmentVisitStatusHistory'[Status] = 4 ),CALCULATE(  MAX('Core TreatmentVisitStatusHistory'[CreationDate]),  'Core TreatmentVisitStatusHistory'[Status] = 7 ),HOUR) |
| Avg\_Wait\_time | var \_temp= CALCULATE(  AVERAGEX(  VALUES('Core TreatmentVisitStatusHistory'[TreatmentVisitID]),  / Average wait time for a patient to attend a doc after arriving /  [Clinic\_time\_diff]) )   RETURN   IF(ISBLANK(\_temp) || ISERROR(\_temp), 0, CONCATENATE(ROUND(\_temp,2)," Hrs")) |
| Selected Department  Name | VAR result =   IF (  ISFILTERED('HR\_Departments'[DepartmentTitleEn]),  SELECTEDVALUE('HR\_Departments'[DepartmentTitleEn]),  " " )  RETURN  result |
| visits tooltip | visits tooltip = "visits : " &FORMAT([Total Visits], "#,##" ) |
| cancellation rate  tooltip | cancellation rate tooltip = "cancellation rate : " &  ROUND(100 [% Cancellation Rate] , 2) & "%" |

## Page Layout:

Height : 1230

Width : 1500

## Calculated Columns:

**Calendar table**

Calendar =

    CALENDAR(

        MIN('Core TreatmentVisits'[VistDateTime].[Date]),

        MAX('Core TreatmentVisits'[VistDateTime].[Date]))

Day = DAY('Calendar'[Date])

DayType = IF(WEEKDAY(DATEVALUE('Calendar'[Date])) <= 5, "weekday", "weekend")

Month = MONTH('Calendar'[Date])

Month\_name = FORMAT('Calendar'[Date],"mmm")

Quarter = "Q"&QUARTER('Calendar'[Date])

WeekNo = WEEKNUM('Calendar'[Date])

WeekOfMonth =

"w" &

FORMAT(

INT((DAY('calendar'[Date]) - 1 + WEEKDAY(DATE(YEAR('calendar'[Date]), MONTH('calendar'[Date]), 1), 2)) / 7) + 1, "0" )

Year = YEAR('Calendar'[Date])

**Core Patients**

Age = ABS('Core Patients'[CreationDate].[Year] - 'Core Patients'[DateOfBirth].[Year])

**Core TreatmentVisitStatusHistory**

Status Description = LOOKUPVALUE('Core TreatmentVisits'[VisitStatusDescription],'Core TreatmentVisits'[TreatmentVisitID],'Core TreatmentVisitStatusHistory'[TreatmentVisitID],Blank())

## Visualization:

Title card:

**1. Text Card Configuration:**

- Select a text card visualization.

- Insert a title with the following settings:

- Font Family: Segoe UI

- Font Size: 18

- Font Color: Black

- Bold Setting: Enabled

- Add a report description below the title with the following settings:

- Font Family: DIN

- Font Size: 8

- Font Color: Black

- Italic Setting: Enabled

**2. Title Card Size Adjustment:**

- Adjust the size of the title card to the following dimensions:

- Height: 96

- Width: 656

Slicers:

**1. Branch and Year Slicers Configuration:**

- Add Branch name column from HR branches and Year column from calendar table to the slicer fields.

- Set the slicer style to "Dropdown."

- Configure slicer header with the following settings:

- Font Family: Segoe UI

- Font Size: 9

- Font Color: Black

- Bold Setting: Enabled

**2. Visit Day Slicer Configuration:**

- Add date column from the calendar table to the slicer field.

- Set the slicer style to "Between" with slider enabled.

- Configure slicer header and values with the following settings:

- Font Family: Segoe UI

- Font Size: 9

- Font Color: Black

- Bold Setting: Enabled

**3. Slicer Size Adjustment:**

- Adjust the size of the Year and Month slicer to the following dimensions:

- Height: 64

- Width: 169

- Adjust the size of the Visit Day slicer to the following dimensions:

- Height: 98

- Width: 265

KPI Cards:

**1. KPI Card Configuration:**

- Add four KPI cards to the report: Adherence Rate, Cancellation Rate, No Show Rate, and Patient Wait Time.

- Set the size of each KPI card to the following dimensions:

- Height: 150

- Width: 210

- Add the respective measures to the KPI card fields.

**2. Text Configuration:**

- Configure the call-out value text for each KPI card with the following settings:

- Font Family: DIN

- Font Size: 25

- Font Color: Black

**3. Title Configuration:**

- Turn on titles for the KPI cards.

- Set the title font to the following settings:

- Font Family: Segoe UI

- Font Size: 13

- Font Color: Black

- Bold and Italic Setting: Enabled

- Enable text wrapping for the titles.

**4. Subtitle Configuration:**

- Set the subtitle font to the following settings:

- Font Family: Segoe UI

- Font Size: 8

- Font Color: #6C6966

**5. Background Configuration:**

- Turn on the background for each KPI card.

- Set the background color to white.

Trendline:

**1. Trendline Plot Configuration:**

- Add a line chart visualization to the report.

- Set the x-axis to include Year, Month, Week, Weekend | Weekday, and Day fields from the dataset.

- Set the y-axis to include the "Total Visits" measure.

- Assign the "Visit Type" column from the "Core TreatmentVisit" table as the legend.

**2. Visual Size Adjustment:**

- Adjust the size of the visual to the following dimensions:

- Height: 311

- Width: 901

**3. Axis Title Configuration:**

- Turn off the x-axis title.

- Turn off both the y-axis title and values.

**4. Legend Configuration:**

- Select the legend position as "Top Center."

**5. Markers and Data Labels Configuration:**

- Turn on markers and set their size to 4.

- Turn on data labels.

**6. Title Configuration:**

- Turn on the title.

- Set the title font to the following settings:

- Font Family: Segoe UI

- Font Size: 13

- Font Color: Black

- Bold and Italic Setting: Enabled

- Enable text wrapping for the title.

**7. Subtitle Configuration:**

- Turn on the subtitle.

- Set the subtitle font to the following settings:

- Font Family: Segoe UI

- Font Size: 9

- Font Color: #666666 (Hex Code)

- Enable the divider.

**8. Tooltip Configuration:**

- Turn on tooltips.

**9. Background Configuration:**

- Turn on the background for the visual.

- Set the background color to white.

Pie Chart:

**1. First Pie Chart Configuration (Category Pie Chart):**

- Add a pie chart visualization to the report.

- Set the size of the visual to the following dimensions:

- Height: 320

- Width: 440

- Assign the "VisitTypeDescription" column from the dataset as the legend.

- Add the "Total Visits" measure to the values field.

- Customize the slice colors as follows:

- New Patient: #FFCDB2

- Follow up: #FFB4A2

- New Visit: #E5989B

- Blank: #E0E2DB

- Turn on the legend and set its position to "Top Center."

- Turn on detail labels and set their position to "Outside."

- Set label contents to "Data Value" and "Percent of Total."

- Configure the font for labels to:

- Font Family: Segoe UI

- Font Size: 9

- Turn on title and subtitle for the chart.

- Set the title font to:

- Font Family: Segoe UI

- Font Size: 13

- Font Color: Black

- Bold and Italic Setting: Enabled

- Turn on text wrapping for the title.

- Set the subtitle font to:

- Font Family: Segoe UI

- Font Size: 9

- Font Color: #6C6966 (Hex Code)

- Turn on the divider.

- Turn on the background for the chart and set it to white.

**2. Second Pie Chart Configuration (Gender Pie Chart):**

- Follow the same steps as above but with the following adjustments:

- Assign the "Gender" column from the dataset as the legend.

- Customize the slice colors as follows:

- Female: #F3D9DC

- Male: #B0D0D3

- Other: #E27396

Donut Chart:

**Donut Chart Configuration:**

- Add a donut chart visualization to the report.

- Set the size of the visual to the following dimensions:

- Height: 320

- Width: 440

- Assign the "Visit\_Type" column from the dataset as the legend.

- Add the "Total Visit" measure to the values field.

- Customize the slice colors as follows:

- Scheduled: #70BBFF

- Walk – in: #12239E

- Turn on the legend and set its position to "Top Center."

- Turn on detail labels and set their position to "Outside."

- Set label contents to "Data Value" and "Percent of Total."

- Configure the font for labels to:

- Font Family: Segoe UI

- Font Size: 9

- Font Color: #6C6966 (Hex Code)

- Turn on title for the chart.

- Set the title font to:

- Font Family: Segoe UI

- Font Size: 13

- Font Color: Black

- Bold and Italic Setting: Enabled

- Turn on the divider.

- Turn on tooltips.

- Turn on the background for the chart and set it to white.

Stacked Bar charts:

**1. Stacked Bar Chart Configuration (Doctor):**

- Add a stacked bar chart visualization to the report.

- Set the size of the visual to the following dimensions:

- Height: 320

- Width: 440

- Assign the "Total Visit" measure to the x-axis.

- Assign the "Doctor" column from the dataset to the y-axis.

- Turn on only values for both x-axis and y-axis.

- Set the font for both axes to Segoe UI, size 9, and color #6C6966 (Hex Code).

- Turn on data labels with the following settings:

- Position: Auto

- Font: Segoe UI, size 9, color #6C6966

- Turn on tooltips.

- Turn on title and subtitle for the chart.

- Set the title font to Segoe UI, size 13, with bold and italic settings.

- Set the subtitle font to Segoe UI, size 10, and color #6C6966.

- Enable text wrapping and the divider.

- Turn on the background for the chart and set it to white.

**2. Stacked Bar Chart Configuration (Department):**

- Follow the same steps as above, but assign the "Department" column from the dataset to the y-axis.

- Assign the tooltip page for the department stacked bar chart.

**3. Stacked Bar Chart Configuration (Country):**

- Follow the same steps as above, but assign the "Country" column from the dataset to the y-axis.

Filter on the report:

**1. IsClinic Filter:**

- Set the filter for the "IsClinic" column to include only TRUE values.

**2. UserTypeID Filter:**

- Set the filter for the "UserTypeID" column to include values where UserTypeID is equal to 1 or 2.

**3. VisitStatusDescription Filter:**

- Set the filter for the "VisitStatusDescription" column to exclude values where VisitStatusDescription is equal to "Scheduled".

## Descriptions:

ResidenceTypeEnum { CitizenOrResidence = 1, Visitor = 2 }

EligibilityTypes { Cash = 1, Insurance = 2, Hospital = 3, NonInsuranceCompanies = 4 }

TreatmentVisitStatus { Confirmed = 1, Completed = 2, Canceled = 3, Arrived = 4, Scheduled = 5,

NoShow = 6, Attended = 7 }

VisitType { NewPatientVisitFee = 1, NewVisitFee = 2, FollowUp = 3 }

VisitAppoitmentTypeEnum { AppointmentVisit = 0, ORAppointmentVisit = 1, InPatient = 2,

EmergencyVisit = 3, ORVisit = 4, HomeCare = 5 }

User Type ID 1 is Physician, 2 is dentist

isclinic = TRUE means the departments of the hospital