**Technical Requirement Specification**

**For**

**OASiS Practice Growth Check Utility**

**Version <1.0>**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/09/2021 | 1.0 | Initial Draft | Sean Perera |
|  |  |  |  |
|  |  |  |  |

**Contents**

[1. GENERAL INFORMATION 6](#_Toc81572688)

[2. ABBREVIATIONS 6](#_Toc81572689)

[3. DEFINITIONS 6](#_Toc81572690)

[4. REQUIREMENT DESCRIPTION 7](#_Toc81572691)

[5. Graphical user interface 8](#_Toc81572692)

[6. import/export data descRIPTION 9](#_Toc81572693)

[7. MIMER SQL SYNTAX and EXAMPLES 10](#_Toc81572694)

[7.1. Database Connection 10](#_Toc81572695)

[7.2. Temporary objects 10](#_Toc81572696)

[7.2.1. First Invoice 10](#_Toc81572697)

[7.2.2. last invoice 10](#_Toc81572698)

[7.2.3. Future Appointment 11](#_Toc81572699)

[7.2.4. Appointment Books 11](#_Toc81572700)

[7.2.5. Billable Items 12](#_Toc81572701)

[7.2.6. Item Analysis 12](#_Toc81572702)

[7.2.7. Churn Dates 13](#_Toc81572703)

[7.2.8. Production amounts 14](#_Toc81572704)

[7.2.9. Unbooked recalls 14](#_Toc81572705)

[7.2.10. Lost Recalls 15](#_Toc81572706)

[7.2.11. Appointment book end time 15](#_Toc81572707)

[7.2.12. Appointment Duration 16](#_Toc81572708)

[7.2.13. Production calculation 17](#_Toc81572709)

[7.3. Global variables 18](#_Toc81572710)

[7.4. Data Collection 19](#_Toc81572711)

[7.4.1. Metric Types and Data Collection Methods 19](#_Toc81572712)

[7.4.2. Hours Worked 20](#_Toc81572713)

[7.4.3. hours cancelled 21](#_Toc81572714)

[7.4.4. Non-patient related hours 22](#_Toc81572715)

[7.4.5. Calendar hours 23](#_Toc81572716)

[7.4.6. Available Hours 23](#_Toc81572717)

[7.4.7. Utilisation 23](#_Toc81572718)

[7.4.8. Number of Appointments 24](#_Toc81572719)

[7.4.9. All Patients 24](#_Toc81572720)

[7.4.10. Active Patients 25](#_Toc81572721)

[7.4.11. New Patients 25](#_Toc81572722)

[7.4.12. Churned Patients 25](#_Toc81572723)

[7.4.13. Unique Patients 26](#_Toc81572724)

[7.4.14. Lapsed Patients 26](#_Toc81572725)

[7.4.15. New Patient % 26](#_Toc81572726)

[7.4.16. Net Patient Gain 27](#_Toc81572727)

[7.4.17. Unbooked Recalls 27](#_Toc81572728)

[7.4.18. Total Recalls 27](#_Toc81572729)

[7.4.19. Recall Effectiveness 28](#_Toc81572730)

[7.4.20. Lost Recalls 28](#_Toc81572731)

[7.4.21. Total Incomplete Treatment Value 28](#_Toc81572732)

[7.4.22. Treatment Plans Created 29](#_Toc81572733)

[7.4.23. Treatment Plans Not Accepted 29](#_Toc81572734)

[7.4.24. Treatment Plan Conversion Rate 29](#_Toc81572735)

[7.4.25. Total Production 30](#_Toc81572736)

[7.4.26. Production Per Available Hour 30](#_Toc81572737)

[7.4.27. Production Per Working Hour 30](#_Toc81572738)

[7.4.28. Production Per Patient 31](#_Toc81572739)

[7.4.29. Production Per Appointment 31](#_Toc81572740)

[7.4.30. Lost Revenue Through Cancellations 31](#_Toc81572741)

[7.4.31. Lost Revenue Through Patient Churn 32](#_Toc81572742)

[7.4.32. Debtors 32](#_Toc81572743)

[7.4.33. Top 10 Items By Value 32](#_Toc81572744)

[7.4.34. Top 10 Items By Count 33](#_Toc81572745)

[8. Other Outputs 34](#_Toc81572746)

[9. Temporary object clean up 35](#_Toc81572747)

[10. APPENDIX 36](#_Toc81572748)

1. GENERAL INFORMATION

|  |  |  |
| --- | --- | --- |
|  | **Requirement name** | OASiS Practice Growth Check Utility |
|  | **Purpose** | To provide Centaur staff a utility that automatically collects data from OASiS databases for a specified period. |
|  | **Priority** | High |
|  | **Cause of requirement appearing** | Commercial Need |
|  | **User** | Centaur Staff |
|  | **Data Source** | OASIS |
|  | **Resources supplied to developer by Centaur Software** | * Mimer SQL queries with examples to create temporary tables, functions, stored procedures. * Mimer SQL queries with examples to create extract data. * Access to a virtual environment with OASiS and Mimer installed, connected to a de-identified OASiS database. |

1. ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
|  | **Abbreviation** | Interpretation |
|  |  |  |

1. DEFINITIONS

|  |  |  |
| --- | --- | --- |
|  | **Definition** | Interpretation |
|  | **OASIS** | An Australian dental practice management software product developed by Henry Schien One |
|  | **Mimer** | The relational database management system that provides the database backend for OASIS |
|  | **DSN** | The ODBC DSN for OASiS Databases - OASIS |
|  | **D4W** | The flagship dental practice management software developed by Centaur |

1. REQUIREMENT DESCRIPTION

**Purpose**

OASiS Software was once a major competitor of the Centaur practice management system, D4W. However, in recent years any further developments to OASiS software have been ceased. Therefore, the dental practices that use OASiS has become the target of a specifically designed “Switch” campaign to encourage them to migrate to D4W.

As an ethical software developer, Centaur Software plans to perform practice growth checks on potential customers’ OASiS databases and identify the areas that need improving and demonstrate that D4W can assist in practice growth.

The aim of the “OASiS Practice Growth Check Utility” is to provide Centaur sales team the ability to extract the relevant data from the OASiS database to perform a thorough review of the practice growth.

**The need for development of a utility**

Centaur software has technical staff who are specialists in the OASiS database structure and the Mimer database engine. They have developed extensive queries to extract the relevant data from OASiS databases. The developer’s task is to design the Graphical User Interface to run the queries supplied by Centaur software, capture the outputs, and output the results to a .csv file.

**Requirements**

1. The utility must be portable and self-contained and must run with no installation of additional files on the customer computer.
2. The utility to be named PGC.exe
3. The utility must connect to the OASIS DSN on the local computer.
4. A .csv output must be created in a newly created C:\OASIS\PGC folder.
5. The output file will have the timestamp included in the name.
6. The user will specify the desired date range in the interface and enter other data extraction parameters.
7. Upon acceptance of the final product, the source code must be handed to Centaur software (if developed by an external party) , and it will remain Centaur intellectual property

**Application**

The utility will connect to the OASiS database, create temporary tables, functions, and procedures, then run SELECT SQL queries to extract data from the temporary tables and other existing database tables. Finally, the tool will clean up the temporary objects by dropping them before exiting the database and closing the connection.

1. Graphical user interface

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen**  **Element** | **Type** | **Description** | **Source** |
| A | Date picker dialogue | Start date of the extraction period | Generic |
| B | Date picker dialogue | End date of the extraction period | Generic |
| C | Numeric dropdown | The number of production columns | Numeric values 1-15 |
| D | Multi select dropdown | The appointment books selection | Please refer to section 7.2.4 for the source query |
| E | Text Box | Practice Name | User enters the value |
| F | Button | Extraction command | Hard coded |
| G | Button | Cancel/Exit | Hard coded |
| H | Logo | Centaur Logo | Hard coded |

1. import/export data descRIPTION

* All the data captured should be output to a .csv file.
* Example naming convention – Practice\_Growth\_Check\_02092021.csv
* The sample format is found in the Appendix. A sample .csv file will be provided in addition to this document.

1. MIMER SQL SYNTAX and EXAMPLES

Database Connection

**DSN Name:** OASIS

**Username:** sysadm

**Password:** 12345pass

**Sample Connection String :** DSN=oasis;UID=sysadm;PWD=12345pass

Temporary objects

* + 1. First Invoice

|  |  |
| --- | --- |
| **Object Name** | FIRST\_INV |
| **Description** | First Invoice Date |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE FIRST\_INV( PTNO CHARACTER(6), INV DATE) |
| **Populate SQL** | INSERT INTO FIRST\_INV SELECT PATNUMBER , min(consultdate) from pbarcmas where deleted <> -1 and transtype =1 group by patnumber; |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE FIRST\_INV; |

* + 1. last invoice

|  |  |
| --- | --- |
| **Object Name** | LAST\_INV |
| **Description** | Last Invoice Date |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE LAST\_INV( PTNO CHARACTER(6), LASTVISITDATE DATE) ; |
| **Populate SQL** | INSERT INTO LAST\_INV SELECT PATNUMBER , max(consultdate) from pbarcmas where deleted <> -1 and transtype =1 group by patnumber; |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE LAST\_INV; |

* + 1. Future Appointment

|  |  |
| --- | --- |
| **Object Name** | FUT\_APP |
| **Description** | Future Appointment |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE FUT\_APP( PTNO CHARACTER(6)) ; |
| **Populate SQL** | INSERT INTO FUT\_APP SELECT DISTINCT PATNUMBER from PAAPPLNS where SKEY > SUBSTRING(CAST(CURRENT\_DATE AS CHAR(16)) FROM 6 FOR 4) || SUBSTRING(CAST(CURRENT\_DATE AS CHAR(16)) FROM 11 FOR 2) || SUBSTRING(CAST(CURRENT\_DATE AS CHAR(16)) FROM 14 FOR 2); |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE FUT\_APP; |

* + 1. Appointment Books

|  |  |
| --- | --- |
| **Object Name** | APP\_BOOK |
| **Description** | Appointment Books |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE APP\_BOOK (BOOKNO INT, BOOKNAME CHARACTER(100),PRIMARY KEY (BOOKNO)) ; |
| **Populate SQL** | INSERT INTO APP\_BOOK (BOOKNO,BOOKNAME) Select CAST(F1 AS INTEGER), F2 from SYTBLENT WHERE SKEY LIKE 'APPVIEWL%' AND SUBSTRING (SKEY FROM 10 FOR 4) = '0000' AND 'APPVIEWNE' || SUBSTRING (SKEY FROM 10 FOR 6) IN (SELECT SKEY FROM SYTBLENT) ; |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE APP\_BOOK; |

Please use this table when populating the multi-select dropdown (D) in the user interface.

SQL Syntax: SELECT BOOKNO, BOOKNAME from APP\_BOOK;

When the user has selected values in the dropdown box, index the value by BOOKNO. However, please ensure to convert to string and pad each BOOKNO value to four digits (2 becomes “00002”,13 becomes “0013”) for use in sections 7.4.1, 7.4.2, 7.4.3, and 7.4.7.

* + 1. Billable Items

|  |  |
| --- | --- |
| **Object Name** | ITEMS |
| **Description** | Billable Items |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE ITEMS ( ITEMCODE CHARACTER(5), DESCR CHARACTER (90)) ; |
| **Populate SQL** | INSERT INTO ITEMS (ITEMCODE,DESCR) SELECT SUBSTRING (SKEY FROM 10 FOR 5), SUBSTRING (F1 FROM 1 FOR 90) FROM SYTBLENT WHERE SUBSTRING (SKEY FROM 1 FOR 9) = 'ITEMNUMBE'; |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE ITEMS; |

* + 1. Item Analysis

|  |  |
| --- | --- |
| **Object Name** | ITEM\_ANALYSIS |
| **Description** | Item Analysis |
| **Object Type** | Table |
| **Create Event** | “Extract” Button Click |
| **Create SQL** | CREATE TABLE ITEM\_ANALYSIS ( ITEMCODE CHARACTER(5), USAGE BIGINT, AMOUNT BIGINT) ; |
| **Populate SQL** | INSERT INTO ITEM\_ANALYSIS (ITEMCODE,USAGE,AMOUNT) Select substring(itemcode from 1 for 5) as it, sum(frequency) as tms, sum(amount) as amt from pbarcmas where transtype = 1 and deleted <> -1 and entrydate between date'" & Format(**$startDate**, "YYYY-MM-DD") & "'" & " and date'" & Format(**$endDate**, "YYYY-MM-DD") & "'" & " group by itemcode ; |
| **Example (If Applicable)** | INSERT INTO ITEM\_ANALYSIS (ITEMCODE,USAGE,AMOUNT) Select substring(itemcode from 1 for 5) as it, sum(frequency) as tms, sum(amount) as amt from pbarcmas where transtype = 1 and deleted <> -1 and entrydate between DATE'2020-01-01' AND DATE'2020-12-31' group by itemcode ; |
| **Clean up SQL** | DROP TABLE ITEM\_ANALYSIS; |

* + 1. Churn Dates

|  |  |
| --- | --- |
| **Object Name** | CHURN |
| **Description** | Patient Churn Date |
| **Object Type** | Table |
| **Create Event** | Application Load |
| **Create SQL** | CREATE TABLE CHURN( PTNO CHARACTER(6), CHURNDATE DATE) ; |
| **Populate SQL** | INSERT INTO CHURN SELECT PTNO, LASTVISITDATE+INTERVAL'18'MONTH FROM LAST\_INV WHERE LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'60'MONTH AND CURRENT\_DATE-INTERVAL'48'MONTH ;  INSERT INTO CHURN SELECT PTNO, LASTVISITDATE+INTERVAL'18'MONTH FROM LAST\_INV WHERE LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'48'MONTH AND CURRENT\_DATE-INTERVAL'36'MONTH ;  INSERT INTO CHURN SELECT PTNO, LASTVISITDATE+INTERVAL'18'MONTH FROM LAST\_INV WHERE LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'36'MONTH AND CURRENT\_DATE-INTERVAL'24'MONTH ;  INSERT INTO CHURN SELECT PTNO, LASTVISITDATE+INTERVAL'18'MONTH FROM LAST\_INV WHERE LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'24'MONTH AND CURRENT\_DATE-INTERVAL'12'MONTH ;  INSERT INTO CHURN SELECT PTNO, LASTVISITDATE+INTERVAL'18'MONTH FROM LAST\_INV WHERE LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'12'MONTH AND CURRENT\_DATE ; |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE CHURN; |

* + 1. Production amounts

|  |  |
| --- | --- |
| **Object Name** | PRODUCTION |
| **Description** | Production Amounts |
| **Object Type** | Table |
| **Create Event** | “Extract” Button Click |
| **Create SQL** | CREATE TABLE Production( Prd DECIMAL (15,2)) ; |
| **Populate SQL** | N/A (Will be populated with a stored procedure) |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP TABLE PRODUCTION; |

* + 1. Unbooked recalls

|  |  |
| --- | --- |
| **Object Name** | unbkRecall |
| **Description** | Unbooked Recalls |
| **Object Type** | Function |
| **Create Event** | Application Load |
| **Create SQL** | @  CREATE FUNCTION unbkRecall(PT VARCHAR(6), RD DATE) RETURNS INTEGER  READS SQL DATA  BEGIN  DECLARE unbooked INTEGER;  DECLARE app INTEGER;  SELECT count(SKEY) INTO app from paapplns where patnumber = PT and entrydate between RD and RD+INTERVAL'30'DAY;  IF app > 0 then  SET unbooked = 0 ;  ELSE SET unbooked = 1 ;  END IF;  RETURN (Unbooked);  END  @ |
| **Populate SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP FUNCTION unbkRecall; |

* + 1. Lost Recalls

|  |  |
| --- | --- |
| **Object Name** | lostRecall |
| **Description** | Lost Recalls |
| **Object Type** | Function |
| **Create Event** | Application Load |
| **Create SQL** | @  CREATE FUNCTION lostRecall(PT VARCHAR(6), RD DATE) RETURNS INTEGER  READS SQL DATA  BEGIN  DECLARE lost INTEGER;  DECLARE app INTEGER;  SELECT count(SKEY) INTO app from paapplns where patnumber = PT and entrydate > RD-INTERVAL'1'DAY;  IF app > 0 then  SET lost = 0 ;  ELSE SET lost = 1 ;  END IF;  RETURN (lost);  END  @ |
| **Populate SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP FUNCTION lostRecall; |

* + 1. Appointment book end time

|  |  |
| --- | --- |
| **Object Name** | apptbookEnd |
| **Description** | Appointment Book End Time |
| **Object Type** | Function |
| **Create Event** | Application Load |
| **Create SQL** | @  CREATE FUNCTION apptbookEnd() RETURNS INTEGER  READS SQL DATA  BEGIN  DECLARE St INTEGER;  DECLARE Hrs INTEGER;  DECLARE En INTEGER;  SELECT CAST(F1 AS INTEGER) INTO St FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ;  SELECT CAST(F2 AS INTEGER) INTO Hrs FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ;  SET En = St + Hrs + 1;  RETURN (En);  END  @ |
| **Populate SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP FUNCTION apptbookEnd; |

* + 1. Appointment Duration

|  |  |
| --- | --- |
| **Object Name** | apptUsed |
| **Description** | Appointment Duration |
| **Object Type** | Function |
| **Create Event** | Application Load |
| **Create SQL** | @  CREATE FUNCTION apptUsed(SKEY VARCHAR(20),usd INTEGER, appEnd INTEGER) RETURNS INTEGER  READS SQL DATA  BEGIN  DECLARE stHr INTEGER;  DECLARE maxDur INTEGER;  DECLARE usedTime INTEGER;  SET stHr = CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) ;  SET maxDur = (appEnd-StHr) \* 60;  IF usd > maxDur then  SET usedTime = maxDur ;  ELSE SET usedTime = usd ;  END IF;  RETURN (usedTime);  END  @ |
| **Populate SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP FUNCTION apptUsed; |

* + 1. Production calculation

|  |  |
| --- | --- |
| **Object Name** | Production |
| **Description** | Production Calculation |
| **Object Type** | Stored Procedure |
| **Create Event** | Application Load |
| **Create SQL** | @  CREATE PROCEDURE Production (IN A DATE, IN B DATE)  MODIFIES SQL DATA  BEGIN  DECLARE invoice DECIMAL (15,2);  DECLARE Adjust DECIMAL (15,2);  DECLARE discount DECIMAL (15,2);  DECLARE Writeoff DECIMAL (15,2);  DECLARE Prod DECIMAL (15,2);  DELETE FROM PRODUCTION;  SELECT Sum(Amount) INTO invoice from pbarcmas where transtype = 1 and deleted <> -1 and entrydate between A and B;  SELECT Sum(Amount) INTO Adjust from pbarcmas where transtype = 2 and deleted <> -1 and entrydate between A and B;  SELECT Sum(Amount) INTO discount from pbarcmas where transtype = 4 and deleted <> -1 and entrydate between A and B;  SELECT Sum(Amount) INTO Writeoff from pbarcmas where transtype = 5 and deleted <> -1 and entrydate between A and B;  IF invoice is null then  set invoice = 0;  END IF;  IF Adjust is null then  set Adjust= 0;  END IF;  IF discount is null then  set discount = 0;  END IF;  IF Writeoff is null then  set Writeoff = 0;  END IF;  SET Prod = invoice+Adjust-discount-Writeoff;  INSERT INTO PRODUCTION (Prd) VALUES (Prod);  END  @ |
| **Populate SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Clean up SQL** | DROP PROCEDURE Production ; |

Global variables

|  |  |
| --- | --- |
| **Global Variable** | **Source** |
| **$startDate** (Reporting period start date) | **$startDate** = GUI Input A |
| **$endDate** (Reporting period end date) | **$endDate** = GUI Input B |
| **$bookLength** (Hours worked in a day) | **$bookLength** = **SELECT CAST(F2 AS INTEGER) +1 AS appbooklength FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ;)** |
| **$prodCol** (No of Production Columns) | **$prodCol =** GUI Input C |
| **$incCol** (Included Columns) | **$incCol** = GUI Input D  (Select from drop down and format as "0000") |
| **$appslot**  (appointment slot length) | **SELECT CAST(F3 AS INTEGER) as Slot FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ; $appslot** = 60/Slot |
| **$iconCan** (Cancellation Icon) | **Select CAST(F5 AS INTEGER) as CAN from sytblent where substring (skey from 1 for 9 ) = 'APPSIXRFE' ; $iconCan** = CAN |
| **$iconNS** (Noshow Icon) | **Select CAST(F6 AS INTEGER) as NS from sytblent where substring (skey from 1 for 9 ) = 'APPSIXRFE' ;** **$iconNS** = NS |
| **$toDay** (Today's Date) | **CURRENTDATE;** |
| **$appStart** (Appointment Book Start Hour) | **SELECT CAST(F1 AS INTEGER) as appStart FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ;** |
| **$appEnd** (Appointment Book End Hour) | **Select apptbookend() FROM SYTBLENT WHERE SKEY = 'PAOPTIONE0000' ;** |

Data Collection

Please note the following notations in the syntax tables below.

* + 1. Metric Types and Data Collection Methods

|  |  |
| --- | --- |
| **Metric Type** | |
| **Monthly** | 1. Count the months between A and B 2. Produce the monthly statistics for each month in the period   Consider the following example where the user selected 01/01/2020 as A and 31/12/2020 as B.  Number of months between A and B: 12  Produce Monthly calculations and statistics for each month between month 1 and month 12. |
| **Snapshot** | Snapshot of the system at the time  In Snapshot calculations the date range that the user entered is not used for the calculation. |
| **Period** | Calculate the total for the entire period. Do not separate by months.  Consider the following example where the user selected 01/01/2020 as A and 31/12/2020 as B.  All period calculations and statistics should be done for 01/01/2020-31/12/2020 |
| **User Input** | Value entered by user |

|  |  |
| --- | --- |
| **Data Collection Method** | |
| **Query Output** | Use the output of the query |
| **Calculation** | Use the given formula to calculate value |
| **User Input** | Value entered by user |

* + 1. Hours Worked

|  |  |
| --- | --- |
| **Metric Name** | Hours Worked |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | **[Update 13/10/2021 SP]**  Select sum(timeused \* **5**)/60 as ptHr from paapplns where picturenumber <> $iconCan and picturenumber <> $iconNS AND SKEY BETWEEN " & "'" & Format($startDate, "YYYYMMDD") & "%' AND " & "'" & Format($endDate, "YYYYMMDD") & "%' and patnumber <> '000000' and (substring(skey from 13 for 4) = '$incCol1' OR substring(skey from 13 for 4) = '$incCol2' OR substring(skey from 13 for 4) = '$incCol3') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > $appStart -1 ; |
| **Example (If Applicable)** | Select sum(timeused \* 5)/60 as ptHr from paapplns where picturenumber <> 93 and picturenumber <> 92 AND SKEY BETWEEN '20200101%' AND '20201231%' and patnumber <> '000000' and (substring(skey from 13 for 4) = '0001' OR substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0005') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > 5; |
| **Notes** | $incCol1, $inCol2…, are the inputs in screen prompt **D.** The user may select 1 or more appointment books. All selected columns must be included in the query.  For example if the user selects appointment book 1 only  substring(skey from 13 for 4) = '0001'  If the user selects appointment books 2 and 13  substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0013'  Please note all the user inputs from D should be padded to four digits. |

* + 1. hours cancelled

|  |  |
| --- | --- |
| **Metric Name** | Hours Cancelled |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | **[Update 13/10/2021 SP]**  Select sum(timeused \* **5**)/60 as cnHr from paapplns where (picturenumber = $iconCan OR picturenumber = $iconNS) AND SKEY BETWEEN " & "'" & Format($startDate, "YYYYMMDD") & "%' AND " & "'" & Format($endDate, "YYYYMMDD") & "%' and patnumber <> '000000' and (substring(skey from 13 for 4) = '$incCol1' OR substring(skey from 13 for 4) = '$incCol2' OR substring(skey from 13 for 4) = '$incCol3') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > $appStart -1 ; |
| **Example (If Applicable)** | Select sum(timeused \* 5)/60 as cnHr from paapplns where (picturenumber = 93 OR picturenumber = 92) AND SKEY BETWEEN '20200101%' AND '20201231%' and patnumber <> '000000' and (substring(skey from 13 for 4) = '0001' OR substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0005') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > 5; |
| **Notes** | $incCol1, $inCol2…, are the inputs in screen prompt **D.** The user may select 1 or more appointment books. All selected columns must be included in the query.  For example if the user selects appointment book 1 only  substring(skey from 13 for 4) = '0001'  If the user selects appointment books 2 and 13  substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0013'  Please note all the user inputs from D should be padded to four digits. |

* + 1. Non-patient related hours

|  |  |
| --- | --- |
| **Metric Name** | Non-Patient Related Hours |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | **[Update 13/10/2021 SP]**  Select sum(apptUsed(SKEY,timeused \* **5**,$appEnd))/60 as nonptHr from paapplns where SKEY BETWEEN " & "'" & Format($startDate, "YYYYMMDD") & "%' AND " & "'" & Format($endDate, "YYYYMMDD") & "%' and (patnumber = '000000' or patnumber ='') and and (substring(skey from 13 for 4) = '$incCol1' OR substring(skey from 13 for 4) = '$incCol2' OR substring(skey from 13 for 4) = '$incCol3') AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > $appStart -1 ; |
| **Example (If Applicable)** | Select sum(  apptUsed(SKEY,timeused \*5, 21)) /60 as nonptHr from paapplns where SKEY BETWEEN '20200101%' AND '20201231%' and (patnumber = '000000' or patnumber ='' )and (substring(skey from 13 for 4) = '0001' OR substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0005') AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > 5 ; |
| **Notes** | $incCol1, $inCol2…, are the inputs in screen prompt **D.** The user may select 1 or more appointment books. All selected columns must be included in the query.  For example if the user selects appointment book 1 only  substring(skey from 13 for 4) = '0001'  If the user selects appointment books 2 and 13  substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0013'  Please note all the user inputs from D should be padded to four digits. |

* + 1. Calendar hours

|  |  |
| --- | --- |
| **Metric Name** | Calendar Hours |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Calendar Hours = Months($endDate -$startDate) \* 30 \* $bookLength\* $prodCol |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | The value 30 always remains consistent regardless of the other variables.  Example:  If the period of start and end date was 01/01/2020-31/12/2020, the number of months is **12**.  If the appointment book length is **10** and production column is **5**  The calculation is:  Calendar Hours = **12** \* 30 \* **10**\* **5** |

* + 1. Available Hours

|  |  |
| --- | --- |
| **Metric Name** | Available Hours |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Available Hours = Calendar Hours - Non Patient Related Hours |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** |  |

* + 1. Utilisation

|  |  |
| --- | --- |
| **Metric Name** | Utilisation |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Utilisation = (Hours worked/Available hours) % |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** |  |

* + 1. Number of Appointments

|  |  |
| --- | --- |
| **Metric Name** | Number of Appointments |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count(\*) as ptac from paapplns where picturenumber <> $iconCan and picturenumber <> $iconNS AND SKEY BETWEEN " & "'" & Format($startDate, "YYYYMMDD") & "%' AND " & "'" & Format($endDate, "YYYYMMDD") & "%' and patnumber <> '000000' and and (substring(skey from 13 for 4) = '$incCol1' OR substring(skey from 13 for 4) = '$incCol2' OR substring(skey from 13 for 4) = '$incCol3') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > $appStart -1; |
| **Example (If Applicable)** | Select count(\*) as ptac from paapplns where picturenumber <> 93 and picturenumber <> 92 AND SKEY BETWEEN '20200101%' AND '20201231%' and patnumber <> '000000' and (substring(skey from 13 for 4) = '0001' OR substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0005') and patnumber <>'' AND CAST(SUBSTRING(SKEY FROM 17 FOR 2) AS INTEGER) > 5 |
| **Notes** | $incCol1, $inCol2…, are the inputs in screen prompt **D.** The user may select 1 or more appointment books. All selected columns must be included in the query.  For example if the user selects appointment book 1 only  substring(skey from 13 for 4) = '0001'  If the user selects appointment books 2 and 13  substring(skey from 13 for 4) = '0002' OR substring(skey from 13 for 4) = '0013'  Please note all the user inputs from D should be padded to four digits. |

* + 1. All Patients

|  |  |
| --- | --- |
| **Metric Name** | All Patients |
| **Type** | Snapshot |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count(\*) as tot from pbpatmas where patnumber < '900000' and inactiveflag <> -1 ; |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Active Patients

|  |  |
| --- | --- |
| **Metric Name** | Active Patients |
| **Type** | Snapshot |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count(\*) as actPat from LAST\_INV where LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'18'MONTH and CURRENT\_DATE; |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. New Patients

|  |  |
| --- | --- |
| **Metric Name** | New Patients |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count (\*) as newPat from FIRST\_INV where INV between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | Select count (\*) as newPat from FIRST\_INV where INV between date'2020-01-01' and date'2020-12-31'; |
| **Notes** | N/A |

* + 1. Churned Patients

|  |  |
| --- | --- |
| **Metric Name** | Churned Patients |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count (\*) as churnPat from CHURN where CHURNDATE between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " and PTNO NOT IN (SELECT PTNO FROM FUT\_APP) ; |
| **Example (If Applicable)** | Select count (\*) as churnPat from CHURN where CHURNDATE between date'2020-01-01' and date'2020-12-31' and PTNO NOT IN (SELECT PTNO FROM FUT\_APP) ; |
| **Notes** | N/A |

* + 1. Unique Patients

|  |  |
| --- | --- |
| **Metric Name** | Unique Patients |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count (distinct patnumber) as uniqPt from pbarcmas where transtype = 1 and deleted <> -1 and entrydate between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | Select count ( distinct patnumber) as uniqPt from pbarcmas where transtype = 1 and deleted <> -1 and entrydate between date'2020-01-01' and date'2020-12-31'; |
| **Notes** | N/A |

* + 1. Lapsed Patients

|  |  |
| --- | --- |
| **Metric Name** | Unique Patients |
| **Type** | Snapshot |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count(\*) as lapsedpt from LAST\_INV where LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'36'MONTH and CURRENT\_DATE-INTERVAL'18'MONTH |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. New Patient %

|  |  |
| --- | --- |
| **Metric Name** | New Patient % |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | NP% = (New Patients/Unique Patients) % |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Net Patient Gain

|  |  |
| --- | --- |
| **Metric Name** | Net Patient Gain |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | NPG = New Patients-Churned Patients |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Unbooked Recalls

|  |  |
| --- | --- |
| **Metric Name** | Unbooked Recalls |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select sum(unBKRecall(patnumber,datecreated)) as unbooked from ptpatnts where details like 'Recall%' and datecreated between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | Select sum(unBKRecall(patnumber,datecreated)) as unbooked from ptpatnts where details like 'Recall%' and datecreated between date'2021-01-01' and date'2021-12-31' ; |
| **Notes** | N/A |

* + 1. Total Recalls

|  |  |
| --- | --- |
| **Metric Name** | Total Recalls |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select count(\*) as totRecall from ptpatnts where details like 'Recall%' and datecreated between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | Select count(\*) as totRecall from ptpatnts where details like 'Recall%' and datecreated between date'2021-01-01' and date'2021-12-31' ; |
| **Notes** | N/A |

* + 1. Recall Effectiveness

|  |  |
| --- | --- |
| **Metric Name** | Recall Effectiveness |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Recall Effectiveness = (Total Recalls-unbooked recalls)/Total Recalls % |
| **SQL** | N/A |
| **Example (If Applicable)** |  |
| **Notes** | N/A |

* + 1. Lost Recalls

|  |  |
| --- | --- |
| **Metric Name** | Lost Recalls |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select sum(lostRecall(patnumber,datecreated)) as lost from ptpatnts where details like 'Recall%' and datecreated between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | Select sum(lostRecall(patnumber,datecreated)) as lost from ptpatnts where details like 'Recall%' and datecreated between date'2021-01-01' and date'2021-12-31' ; |
| **Notes** | N/A |

* + 1. Total Incomplete Treatment Value

|  |  |
| --- | --- |
| **Metric Name** | Total Incomplete Treatment Value |
| **Type** | Snapshot |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | SELECT SUM(AMOUNT) FROM PTPATTPN WHERE HEADER <> -1 AND VISITHEADER <> -1 AND COMPLETED <> -1 AND PATNUMBER IN (SELECT PTNO FROM LAST\_INV where LASTVISITDATE BETWEEN CURRENT\_DATE-INTERVAL'18'MONTH and CURRENT\_DATE); |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Treatment Plans Created

|  |  |
| --- | --- |
| **Metric Name** | Treatment Plans Created |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | SELECT SUM(AMOUNT) FROM PTPATTPN WHERE HEADER <> -1 AND VISITHEADER <> -1 AND DATECREATED between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " ; |
| **Example (If Applicable)** | SELECT SUM(AMOUNT) FROM PTPATTPN WHERE HEADER <> -1 AND VISITHEADER <> -1 AND DATECREATED BETWEEN DATE'2020-01-01' AND DATE'2020-12-31'; |
| **Notes** | N/A |

* + 1. Treatment Plans Not Accepted

|  |  |
| --- | --- |
| **Metric Name** | Treatment Plans Not Accepted |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | SELECT SUM(AMOUNT) FROM PTPATTPN WHERE HEADER <> -1 AND VISITHEADER <> -1 AND COMPLETED <> -1 AND DATECREATED between date'" & Format($startDate, "YYYY-MM-DD") & "'" & " and date'" & Format($endDate, "YYYY-MM-DD") & "'" & " and PATNUMBER NOT IN (SELECT PTNO FROM FUT\_APP) ; |
| **Example (If Applicable)** | SELECT SUM(AMOUNT) FROM PTPATTPN WHERE HEADER <> -1 AND VISITHEADER <> -1 AND COMPLETED <> -1 AND DATECREATED BETWEEN DATE'2020-01-01' AND DATE'2020-12-31' and PATNUMBER NOT IN (SELECT PTNO FROM FUT\_APP) ; |
| **Notes** | N/A |

* + 1. Treatment Plan Conversion Rate

|  |  |
| --- | --- |
| **Metric Name** | Treatment Plan Conversion Rate |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | CR = (plans created - not completed)/plans created% |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Total Production

|  |  |
| --- | --- |
| **Metric Name** | Total Production |
| **Type** | Monthly |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Call Production ($startDate,$endDate); Select prd from production; |
| **Example (If Applicable)** | CALL Production (date'2020-01-01', date'2020-12-31') ;  Select prd from production; |
| **Notes** | N/A |

* + 1. Production Per Available Hour

|  |  |
| --- | --- |
| **Metric Name** | Production Per Available Hour |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Total production/ available hours |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Total Production is calculated in 7.4.24  Available Hours is calculated in 7.4.5 |

* + 1. Production Per Working Hour

|  |  |
| --- | --- |
| **Metric Name** | Production Per Working Hour |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Total production/Hours Worked |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Total Production is calculated in 7.4.24  Hours Worked is calculated in 7.4.1 |

* + 1. Production Per Patient

|  |  |
| --- | --- |
| **Metric Name** | Production Per Patient |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Total production/Unique Patients |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Total Production is calculated in 7.4.24  Unique Patients is calculated in 7.4.12 |

* + 1. Production Per Appointment

|  |  |
| --- | --- |
| **Metric Name** | Production Per Appointment |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Total production/Number of Appointments |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Total Production is calculated in 7.4.24  Number of Appointments is calculated in 7.4.7 |

* + 1. Lost Revenue Through Cancellations

|  |  |
| --- | --- |
| **Metric Name** | Lost Revenue Through Cancellations |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Production per working hour \* Hours Cancelled |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Production per working hour is calculated in 7.4.26  Hours Cancelled is calculated in 7.4.1 |

* + 1. Lost Revenue Through Patient Churn

|  |  |
| --- | --- |
| **Metric Name** | Lost Revenue Through Patient Churn |
| **Type** | Monthly |
| **Method** | Calculation |
| **Calculation** | Production per patient \* Churned Patients |
| **SQL** | N/A |
| **Example (If Applicable)** | N/A |
| **Notes** | Production per patient calculated in 7.4.28  Churned Patients is calculated in 7.4.11 |

* + 1. Debtors

|  |  |
| --- | --- |
| **Metric Name** | Debtors |
| **Type** | Snapshot |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | Select Sum(invoices-credits) as patdebt from pbaccbal where drnumber = 0; |
| **Example (If Applicable)** | N/A |
| **Notes** | N/A |

* + 1. Top 10 Items By Value

|  |  |
| --- | --- |
| **Metric Name** | Top 10 Items by Value |
| **Type** | Period |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | select itemcode,amount from item\_analysis order by amount desc ; |
| **Example (If Applicable)** | N/A |
| **Notes** | Please note that the Mimer Database Engine does not have a command to select top n records. Therefore, read the output of the above query to a recordset or array and select the top 10.  Please output the result in the first three columns of the .CSV file in the following format.  Top 10 items by value,,,,,,,,,,,,,  1,688,"$551,700",,,,,,,,,,,  2,613,"$533,715",,,,,,,,,,,  3,114,"$513,077",,,,,,,,,,,  4,661,"$225,595",,,,,,,,,,,  5,012,"$205,188",,,,,,,,,,,  6,022,"$176,422",,,,,,,,,,,  7,735,"$150,150",,,,,,,,,,,  8,965,"$126,760",,,,,,,,,,,  9,324,"$120,375",,,,,,,,,,,  10,672,"$119,645",,,,,,,,,,, |

* + 1. Top 10 Items By Count

|  |  |
| --- | --- |
| **Metric Name** | Top 10 Items by Count |
| **Type** | Period |
| **Method** | Query Output |
| **Calculation** | N/A |
| **SQL** | select itemcode,usage from item\_analysis order by usage desc ; |
| **Example (If Applicable)** | N/A |
| **Notes** | Please note that the Mimer Database Engine does not have a command to select top n records. Therefore, read the output of the above query to a recordset or array and select the top 10.  Please output the result in the first three columns of the .CSV file in the following format.  Top 10 items by Count,,,,,,,,,,,,,  1,688,234,,,,,,,,,,,  2,613,291,,,,,,,,,,,  3,114,3346,,,,,,,,,,,  4,661,206,,,,,,,,,,,  5,012,3259,,,,,,,,,,,  6,022,3374,,,,,,,,,,,  7,735,151,,,,,,,,,,,  8,965,213,,,,,,,,,,,  9,324,281,,,,,,,,,,,  10,672,67,,,,,,,,,,, |

1. Other Outputs

Diagram

Description automatically generated

The following user inputs captured from the Graphical User Interface must also be output the .csv file in the following format. In addition, the date and time of the data extraction (system date and time) must also be output. This information to be added at the end of the file.

|  |  |
| --- | --- |
| A | Start Date,1/07/2020,,,,,,,,,,,, |
| B | End Date,30/06/2020,,,,,,,,,,,, |
| C | Production Columns,5,,,,,,,,,,,, |
| D | Appointment Books,"0200,0005,0013",,,,,,,,,,,, |
| E | Practice Name,XYZ Dental,,,,,,,,,,,, |
| Date | Date of Extraction,1/09/2021,,,,,,,,,,,, |
| Time | Time of Extraction,13:55:22,,,,,,,,,,,, |

1. Temporary object clean up

Each time an extraction is run the following Mimer SQL queries should be used to clean up the temporary objects created during the extraction.

DROP TABLE FIRST\_INV;

DROP TABLE LAST\_INV;

DROP TABLE FUT\_APP;

DROP TABLE APP\_BOOK;

DROP TABLE ITEMS;

DROP TABLE ITEM\_ANALYSIS;

DROP TABLE CHURN;

DROP TABLE Production;

DROP FUNCTION unbkRecall;

DROP FUNCTION lostRecall;

DROP FUNCTION apptbookEnd;

DROP FUNCTION apptUsed;

DROP PROCEDURE Production ;

1. APPENDIX

The following is the expected output in the .csv file.

Headers,Snapshot,Jul-20,Aug-20,Sep-20,Oct-20,Nov-20,Dec-20,Jan-21,Feb-21,Mar-21,Apr-21,May-21,Jun-21

Hours Worked,,123,213,206,213,127,256,192,198,156,185,210,220

Hours Cancelled,,19,16,22,12,19,15,14,16,11,22,13,12

Non-Patient Related Hours,,375,154,120,218,355,165,260,199,345,360,215,95

Calendar Hours,,600,600,600,600,600,600,600,600,600,600,600,600

Available Hours,,225,446,480,382,245,435,340,401,255,240,385,505

Utilisation,,55%,48%,43%,56%,52%,59%,56%,49%,61%,77%,55%,44%

Number of appointments,,391,375,324,324,322,400,426,255,270,401,430,335

All Patients,16445,,,,,,,,,,,,

Active Patients,4508,,,,,,,,,,,,

Inactive Patients,11937,,,,,,,,,,,,

New Patients,,53,48,49,39,40,46,56,34,46,58,70,43

Churned Patients,,55,76,62,74,62,66,46,62,61,68,79,89

Unique,,580,543,461,198,433,630,602,301,338,608,592,531

Lapsed Patients,1605,,,,,,,,,,,,

New Patient %,,9%,9%,11%,20%,9%,7%,9%,11%,14%,10%,12%,8%

Net patient gain ,,-2,-28,-13,-35,-22,-20,10,-28,-15,-10,-9,-46

Unbooked Recalls,,446,318,183,286,441,441,381,372,298,163,266,421

All Recalls,,541,384,231,373,526,509,443,418,364,211,353,506

Recall Effectiveness,,18%,17%,21%,23%,16%,13%,14%,11%,18%,23%,25%,17%

Lost Recalls,,274,225,141,195,347,389,368,372,254,205,121,175

Total Incomplete Treatment Value,"$690,527",,,,,,,,,,,,

Treatment plans created ,,"$123,860","$191,945","$142,103","$116,226","$151,791","$133,696","$123,606","$155,306","$128,561","$115,161","$136,498","$113,133"

Treatment plans not accepted,,"$64,103","$127,580","$87,714","$74,675","$79,632","$85,844","$93,743","$90,328","$74,931","$56,030","$86,398","$65,567"

Treatment conversion rate,,48%,34%,38%,36%,48%,36%,24%,42%,42%,51%,37%,42%

Total Production,,"$83,004","$126,726","$116,742","$122,673","$122,676","$110,611","$95,406","$93,684","$141,511","$121,720","$139,123","$105,958"

Production per available hour,,$138,$211,$195,$204,$204,$184,$159,$156,$236,$203,$232,$177

Production per working hour,,$675,$595,$567,$576,$966,$432,$497,$473,$907,$658,$662,$482

Production per patient,,$143,$233,$253,$620,$283,$176,$158,$311,$419,$200,$235,$200

Production per appointment,,$212,$338,$360,$379,$381,$277,$224,$367,$524,$304,$324,$316

Lost revenue through cancellations,,"$12,822","$9,519","$12,468","$6,911","$18,353","$6,481","$6,957","$7,570","$9,978","$14,475","$8,612","$5,780"

Lost revenue though patient churn,,"$37,116","$45,217","$35,136","$42,619","$59,889","$28,517","$22,858","$29,335","$55,334","$44,740","$52,337","$42,865"

Debtors,"$284,279",,,,,,,,,,,,

Top 10 items by value,,,,,,,,,,,,,

1,688,"$551,700",,,,,,,,,,,

2,613,"$533,715",,,,,,,,,,,

3,114,"$513,077",,,,,,,,,,,

4,661,"$225,595",,,,,,,,,,,

5,12,"$205,188",,,,,,,,,,,

6,22,"$176,422",,,,,,,,,,,

7,735,"$150,150",,,,,,,,,,,

8,965,"$126,760",,,,,,,,,,,

9,324,"$120,375",,,,,,,,,,,

10,672,"$119,645",,,,,,,,,,,

Top 10 items by Count,,,,,,,,,,,,,

1,688,234,,,,,,,,,,,

2,613,291,,,,,,,,,,,

3,114,3346,,,,,,,,,,,

4,661,206,,,,,,,,,,,

5,12,3259,,,,,,,,,,,

6,22,3374,,,,,,,,,,,

7,735,151,,,,,,,,,,,

8,965,213,,,,,,,,,,,

9,324,281,,,,,,,,,,,

10,672,67,,,,,,,,,,,

Practice Name,XYZ Dental,,,,,,,,,,,,

Start Date,1/07/2020,,,,,,,,,,,,

End Date,30/06/2020,,,,,,,,,,,,

Production Columns,5,,,,,,,,,,,,

Appointment Books,"20,000,050,013",,,,,,,,,,,,

Date of Extraction,1/09/2021,,,,,,,,,,,,

Time of Extraction,13:55:22,,,,,,,,,,,,