Developing Standardized SMU Semester Calendar to Identify Trends in Library Behavior Post-COVID

Hollie Gardner: *hgardner@smu.edu*
Grace Lang: *graciel@smu.edu*
Rashmi Patel: *rashmip@smu.edu*

Abstract

Using past data, previous trends, and an ensemble model, the projected number of user engagement items for Southern Methodist University's (SMU) libraries was forecasted for the Fall 2020 and Spring 2021 semesters as if the COVID-19 pandemic had not happened. The forecasted data was compared to the actual numbers of user engagement items. This paper shares the methodology and comparisons of forecasts and actuals of SMU Libraries visitor counts, reference transactions, instructions sessions, research consultation appointments, and search session. In order to achieve a clearer forecast of user behavior, creating a semester calendar following SMU's academic calendar in the ETL step was necessary endeavor. Further discussion will take place on analysis of these differences and what this means for the future of the SMU Libraries and recommendations on improvements in data collection and analysis to manage future forecasting of user behavior to drive business decisions.

I. Introduction

The 2020 COVID-19 pandemic disrupted many institutions, causing significant changes to practices and services in response to heightened safety measures. An example of this would be the Southern Methodist University (SMU) Libraries. The SMU campus officially closed on March 16, 2020. On July 5, 2020, SMU reopened its doors to a limited number of students to attend classes in person during the second summer session.

As an integral part of research and academic activities, the SMU Libraries had to quickly respond to the changing environment to meet the needs of students and faculty during the Spring 2020 semester as they transitioned from fully in-person to fully remote. The SMU Libraries sought to serve the new digital needs of students with online advisements, virtual trainings to use the library database, and chat features to answer students' questions in real-time.

While the pandemic is still with potential variants looming on the horizon which threaten to send us back home, this project asks, "What user behavior trends do we see are part of a temporary response to the pandemic and what trends are part of the new normal of how our students and faculty use libraries and are now practices which may outlast the pandemic?" Some efforts to move fully online catalyzed changes to service models which were part of a larger strategic effort which would have taken more time. Therefore, it is important to understand how these changes in behavior should play into the larger direction SMU Libraries takes as it reimagines its future in a post-pandemic world.

This investigation seeks to understand the previous trends in library usage. While utilizing those trends, we will forecast the predicted usage of services and facilities that the SMU Libraries would have experienced had the COVID-19 pandemic not happened, as if the libraries were expected to continue in an alternate, COVID-19-free universe. Then, we will compare our forecasted numbers to the actual usage in the Fall 2020 and Spring 2021 semesters when the university reopened, and students began to return to their normal behaviors. This information will be used by library administration to answer questions about where the libraries should be allocating resources in a post-pandemic world.

II. Literature Review

In March 2020 COVID-19 halted many businesses and educational institutions in its' tracks, but there was an urgent need to continue "business as usual". However, the "normal" business that we knew would be transformed into even more heavily reliant on digital platform services. Higher educational libraries turned digital overnight as many social restrictions required libraries to close their doors. Virtual reference services (VRS) became a primary access for students and alumni's information needs [1]. This digital transformation was seen globally and because of the accelerate nature of the pandemic, many libraries saw significant challenges in providing "access to resources previously available in print" [2-3]. Converting textbooks from print to searchable pdf versions for students was one of the major challenges libraries had to overcome in a short amount of time. What was once a quick look up in a print table was now a painstaking process for a librarian to upload, convert, and reference documents for each area of study. To help in finding a solution to the barriers libraries were facing, the International Coalition of Library Consortia (ICOLC)

Commented [LG1]: Not a great title, but I wanted to slap something in here to start...

Commented [LG2]: do you mean accessing the library in-person? or did they participate in something?

Commented [GH3R2]: thanks for catching that. revised to clarify they attended classes in person.

preemptively lifted some digital restrictions to help libraries and students finish out the term with as little disruption as possible [5]. For example, on March 13, 2020, ICOLC made any relevant datasets and content surrounding COVID-19 available, lifted existing contractual limits on photocopying, waived limits to licensed digital content used for research, discovery, and learning.

Initially there was little published surrounding the effects of COVID-19 on higher education libraries. The first published articles focused on the early reactions of how some institutions were solving the barriers to education. However, new and more comprehensive studies are emerging with some of the trends seen over the past year. A study conducted by Louisiana State University saw that in the three libraries studied, the use of discovery tools for catalogs and major databases decreased, while there was a significant increase in virtual communication within the libraries [4]. This surprising shift in digital searches to focus on having a conversation with a librarian was one of the objectives we wanted to address in this analysis.

III. Data Sources

Data collection and operational data sources: The data for instruction sessions and reference transactions were collected using Springshare products. Librarians and library staff enter the data into forms within RefAnalytics or LibInsights. The screenshots of the forms that librarians use to enter the data are available in the appendix. There has been an evolution of the forms that librarians and library staff since a dramatic shift in organization structure for the SMU Libraries prompted the consolidation of data reporting structures.

Gate counts were recently collected using LibInsights form (see *Appendix Image 1*). Prior to this time, library staff used an Excel document. All past data has been migrated into LibInsights. Gate counts are collected using 3M door security devices that also measure the laser beam interruptions for each entry and exit. Every morning, library staff collect the number from the five different entrances and record as the previous day's end and the current day's beginning number. The LibInsights system calculates the difference and divides in half since the counting system is bi-directional (measures both in and out traffic). Some counting errors in the data may relate to patrons entering in a dense grouping, beam interruption due to carts or other baggage entering the facility, or mechanical failures of the 3M devices. Other inaccuracies in data collection may have resulted from human errors such as data entry errors or failure to collect staff not collecting gate counts, counting errors. Due to the disruption of COVID-19 and staff turnover, gate count daily data was lost for Spring 2020. To resolve this issue, Spring 2020 gate counts were excluded from forecasting analysis. The sum of the semester gate count was retrieved through

Study room reservations were collected using Springshare's LibCal booking system. Study room booking calendars were created within LibCal and students, faculty, or staff can use SMU credentials to make reservations. These reservations were recorded by the system and then exported. This data does not capture organic usage that comes from occupancy of the rooms by passerby's who find the room vacant. Nor does it capture reservations that were made, but not were not occupied by the patron. The system sets 3-hour limits on reservations and only allows for one back-to-back reservation by the same user.

Library Search Sessions are recorded within the library's discovery tool, Ex Libris Primo. According to the Ex Libris Knowledge Center, a session is "a measurement field that tracks the number of sessions in which an action was taken." If a user ran multiple searches in a period, only one session would be counted.

Ask Us is the branded reference service for the SMU Libraries. Reference transactions are recorded within Springshare's RefAnalytics product (see *Appendix Image 2*). Some of the transactions are recorded by the system, since LibAnswers is being used for email, chat, and SMS transactions, whereas in-person and phone transactions rely on library staff to record the interactions. Inaccuracies may occur if staff forget to record in-person or phone transactions or if they record email, chat, and SMS demographic information incorrectly. Data prior to FY21 was captured through multiple forms since each individual library was able to develop their own prior to the organization structure changes. Beginning June 1, 2020, all libraries were expected to record their reference transactions in the same form within RefAnalytics. Some exceptions include special collections entities since the nature of their reference transactions requires them to also record material requests. Extensive efforts were made to consolidate all reference transaction forms into a single dataset. In addition to all Ask Us Transactions being analyzed, both *In Person* and *Chat* modalities were analyzed separately since user behavior could have changed significantly in these areas due to the pandemic.

Instruction Sessions include one-shot library instruction sessions where librarians visit a course session to provide instruction on various topics including information literacy, data literacy, and research skills training. Librarians record the data in the Course-Related Instruction form within Springshare's LibInsight (see *Appendix Image 3*). It should be noted that other external factors include instruction sessions began to take place online using Zoom in March 2020 and remained online through the end of Spring 2021.

Librarian Appointments are one-on-one appointments that students and faculty can schedule with a librarian who specializes in their subject area. Appointments typically last anywhere from 30 minutes to 1 hour and can be scheduled using the Springshare LibCal system or via email depending on the librarian. Depending on the nature of the topic, some librarians can provide consultation via email if the support provided is up to the same level that would be provided during a meeting. It should be noted that other external factors include that librarian appointments were offered solely online using Zoom beginning in March 2020 and remained online through the end of Spring 2021.

Extraction & Data Storage: Many of the source systems do not offer a way to perform automated extraction. For the purposes of this project, we relied on manual extraction and upload batch files to OneDrive. Formalized efforts at retrieving

all library data from the numerous systems have not been made by the organization. Also, not all data was recorded and available for extraction based on the library and their systems. For the purposes of this project, Microsoft OneDrive was utilized to serve as a data lake to hold files and permissions were shared with project team members.

Transformation: When utilizing PowerBI the time frames were aggregated at the calendar year, quarter, and week level. However, in the context of SMU's semesters it was difficult to parse out when a semester started or ended within a quarter or at the week level. This governance of this analysis and future analysis represented by this data was reliant on a manual process of entering in what semester, fiscal year, or semester week a specific date was within the SMU calendar. This database project allowed us to develop a specific calendar table that can be used across multiple analysis for the purpose of aligning it with SMU's specific calendar. These customized calendar fields will allow us to see patterns within the data that we weren't able to view at the level of detail before.

The calendar table is initially built from an automatic incrementing procedure which fills out consecutive days within a timeframe. Then using case statements and nested groupings we can properly identify how an individual date falls within the SMU fiscal year, semester, and week of the semester. Standard transformation functions are also utilized to pull out the day of the week name, the calendar week, and year.

Load: The data sources listed above were imported and cleaned using MySQL Workbench. The queries and procedures created in this process will then be leveraged in the PowerBI deployment. After transformation using MySQL Workbench, we loaded the data back into OneDrive and set permissions on the file for use.

Data analysis: Simple descriptive statistics were developed within MySQL Workbench to answer some of the overall trend insights that were needed before completing any time series analysis.

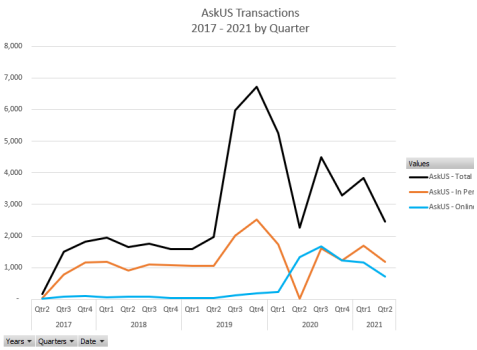


Figure 1.0: The Ask US library feature was picking up momentum in 2019. COVID hurt overall usage of the tool; however, it did increase the Online Chat platform of the tool.

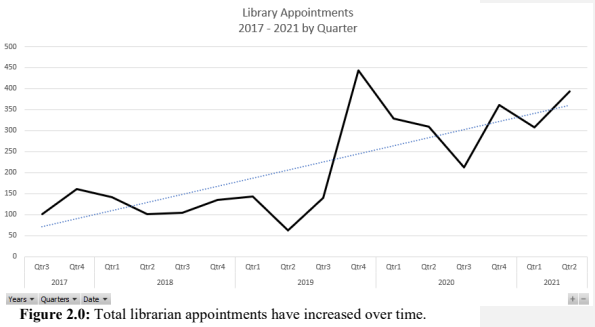


Figure 2.0: Total librarian appointments have increased over time.

Additional views by specific library are also available with this dataset upon filtering. These trends by library will be used in discussion with all SMU library directors when discussing fresh marketing or information sessions.

TS Analysis: The first step in the analysis involved plotting a simple time series of all data to ensure data was complete. Next, a signal-plus noise model was used to determine if there was an existing trend in the pre-COVID data. Only Fall and Spring semester data was retained for the model since those populations of dates were most similar in terms of the conditions set by SMU with the traditional academic calendar that most students follow. Including the winter and summer semesters would have drastically altered the mean, therefore our predictions for Fall and Spring semesters would have had a much higher ASE. Spring break was also excluded from the data, allowing the Fall and Spring semester week numbers to better align.

The existing trend was calculated for each dataset using a signal-plus noise model and then verified using the Cochrane-Orcutt test. Following evaluation of the trends, a vector-autoregressive model (VAR) was used to perform predictions. "Day of the week" was discovered to be a significant variable and was used to facilitate lower ASE rate, in addition to the lag of 7. Furthermore, a multi-layer perceptron (MLP) neural network model was used and included the "day of the week" and "semester week number" variables. It was discovered that the ensemble model of using the VAR (7) and MLP neural network produced the lowest ASE, therefore the forecasts were produced using this model. In the interest of time, the same variables and lag were used in the ensemble model across all datasets without any additional transformations. Some forecasts could benefit from additional transformations in future iterations of the models to improve predictions.

The last step included a comparison taking the sum of the forecasts for each Spring 2020, Fall 2020, and Spring 2021 and comparing to the percentage change to the actual for each semester.

Front-end Application: Power BI is a business intelligence platform that allows users which are from non-technical

Commented [GH4]: Changed the SQL methodology to align with course-language.

background to gather, analyze, visualize, and share the data reports. The SMU Library data was loaded onto the Power BI platform and different tools were used to analyze the data and visualize the interesting facts and trends. The reports using the data are dynamic where specific filters can be performed to see trends by fiscal year, semester, month, or any specific time range. The reports contain the data visualization for the Ask Us Service, Instruction Session, Search Session, Library Appointments, and Gate Counts datasets. Tools like cards, clustered column chart, donut chart, table and slicer are used to generate the reports.

IV. Results

The signal-plus noise model resulted in a trend equation for each variable, which the equation is shown below. All the variables were increasing at various degrees, except for instruction sessions, which was experiencing a slight daily decline.

Variable	Equation
Gate Counts	= 3069.94 + 2.19(days)
Instruction Sessions	= 1.22 - 0.00037(days)
Ask Us Transactions	= 7.12 + 0.096(days)
- In Person	= 11.63+ 0.022(days)
- Chat	= 0.58 + 0.0016(days)
Librarian Appts	= 1.47 + 0.0035(days)
Search Sessions	= 732.28 + 0.20(days)

After conducting analysis on this topic, the research shows that some items experienced a steep decline in what was forecasted while others experienced great increases. The following table shows the trend equation produced using the signal-plus noise model and the percent change from the forecasted amount using the ensemble model.

Variable	Spring 2020	Fall 2020	Spring 2021
Gate Counts	Up 17%	Down 64%	Down 63%
Instruction Sessions	Down 16%	Up 35%	Up 91%
Ask Us Transactions	Up 12%	Down 3%	Down 17%
- In Person	Down 38%	Down 8%	Down 6%
- Chat	Up 417%	Up 1066%	Up 760%
Librarian Appts	Up 64%	Up 44%	Up 66%
Search Sessions	Up 7%	Up 426%	Up 376%

V. Deployment

Data Warehouse Recommendations: The SMU Libraries has a vast amount of data that should be utilized in this manner to track user behaviors and make strategic investments, especially given the disruption of the pandemic and the accelerated changes in how users perform academic work. The first recommendation would be to undertake a project that would bring together all data streams where possible. Given the proprietary nature or limitations that vendors have on the data being recorded, not all processes can be automated, so designated staff should be made available to conduct regular extractions. This will allow for library staff to make use of the data without everyone having to go through the same process of collecting every single time any data analysis wants to be performed. If automation cannot be achieved and close to real-time data is desired, it is potentially worth investigating if Springshare products will continue to be the right platform to use for data collection.

Future Scaling: The SMU library data stored in open-source relational database can be connected to Power BI desktop only through in-built connector which helps for easy integration, analysis, and visualization of relational data. So, for using the data in Power BI as Service, the library data needs to be stored in cloud like Azure, AWS RDS, Amazon Redshift, GCP, etc.

VI. Conclusion

Major disruption catalyzes change. Out of the 1918 pandemic, lending libraries were shuttered, and items were quarantined at patron homes as they were instructed. Upon reopening, there was a shift in the mindset and approaches to librarianship as evidenced by the dialogue and discussions taking place among American Libraries Association leadership where ideas were proposed of a single, large catalog database of every book published and a dramatic shift in the idea of unifying the librarians under a profession[citation].

It would be naïve to assume that such a catalytic event as the current pandemic would not result in major shifts to user behaviors and the approaches the profession takes to address those shifts in the present time. From this analysis of the existing data at SMU, locally there have been some changes in user behavior such as increase in reliance on digital modes, such as the increase in chat, more consultation usage once it moved virtually, and higher instruction sessions and it could be recommended to the library administration to invest in systems and training which facilitate better online usage. On the other hand, we see gate counts dropping which may have impacts on future space planning and may indicate that space resources can be redirected towards other initiatives.

It is impossible to determine that the pandemic is the cause of these changes, but we can observe the changes that are happening and how we are being responsive to those changes. There is an opportunity to track these changes in the library profession so we can continue to be nimble and more responsive in our service model shifts. Further opportunities exist to examine these patterns with a more systematic approach, perhaps while using the Association of College and Research Libraries' annual survey data collection to see if the same changes are occurring at other institutions with their campus populations.

Appendix

Image 1: Daily gate count for Fondren Library is recorded within a LibInsights form

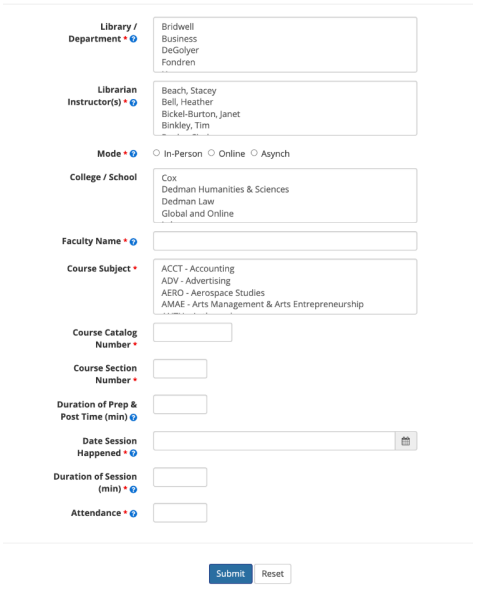


Image 2: Ask Us transaction form used by library staff to record data

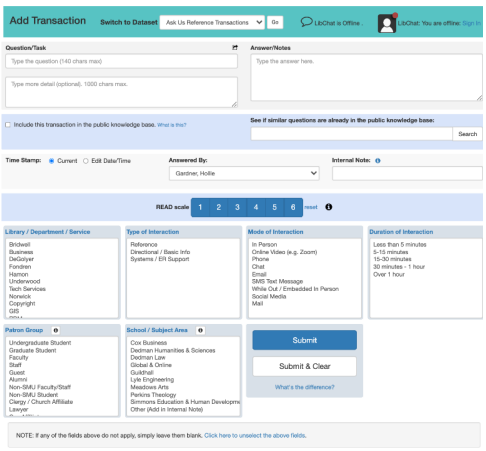
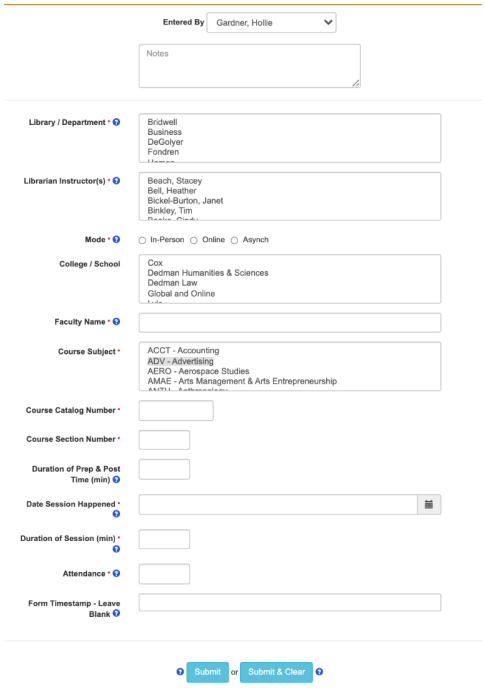


Image 3: Library instruction sessions are recorded within the LibInsights form by teaching librarians



References

[1]
K. A. Muhammad, "Implementation and Use of Virtual Reference Services in Academic Libraries during and post COVID-19 Pandemic: A Necessity for Developing Countries," pp. 1–18, 2021.

[2]
R. Orcutt, L. Campbell, M. Gervits, B. Opar, and K. Edwards, "COVID-19 Pandemic," vol. 40, no. 1, pp. 123–140, 2021, doi: 10.1086/714593.

[3]
A. França, "Transforming library collections in a pandemic: the perspective from Edge Hill University," vol. 34, no. 1, pp. 1–6, 2021, doi: 10.1629/uksg.536.

[4]
R. S. Connell, L. C. Wallis, and D. Comeaux, "The Impact of COVID-19 on the Use of Academic Library Resources," vol. 40, no. 2, 2021, doi: 10.6017/ital.v40i2.12629.

[5]
P. Johnson, "Libraries During and After the Pandemic," vol. 40, no. 4, pp. 2–8, 2020.

Code used for MySQL Workbench

```
##########################################
##############################
```

```
#Create new Schema for SMU Library data
  #create schema smu_library;
use smu_library;
#####################################
############################
-- drop table IF EXISTS instruction;
-- drop table IF EXISTS gatecounts;
-- drop table IF EXISTS
instruction_sessions;
-- drop table IF EXISTS
consultation_sessions;
#####################################
############################
CREATE TABLE
`library_smu`.`instruction`
(`id` int
  , `Form Timestamp - Leave Blank` text
  , `IP Address` text
  , `Notes` text
  , `Entered By` text
  , `Library / Department` text
  , `Librarian Instructor(s)` text
  , `Mode` text
  , `College / School` text
  , `Faculty Name` text
  , `Course Subject` text
  , `Course Catalog Number` int
  , `Course Section Number` int
  , `Duration of Prep & Post Time (min)`
int
  , `Date Session Happened` text
  , `Duration of Session (min)` int
  , `Attendance` int);
create table gatecounts
  (location  int(2),
   lib_name  varchar(75),
   gate_day  date,
     gate_start   timestamp,
     gate_end   timestamp,
     visitor_count int,
     recording  varchar(25),
     NOTES     VARCHAR(25) DEFAULT NULL,
   primary key (location,gate_day)
  );

create table instruction_sessions
  (id  int(5),
   gate_start  datetime,
     ip_address   int(50) DEFAULT NULL,
     notes     VARCHAR(25) DEFAULT NULL,
     entered_by   varchar(50),
     library   varchar(25),
     librarian_inst  varchar(25),
     class_mode   varchar(25),
     school    varchar(75),
     faculty   varchar(75),
     course    varchar(75),
     course_cat_numfloat,
     course_sec_num float,
     duration_preppost  float,
   inst_session_date datetime,
     duration_session   float,
     attendence   float,
   primary key (id,gate_start)
  );
#####################################
####################
#Data Cleanup
ALTER TABLE library_smu.askus RENAME
COLUMN `ï»¿Id` TO `ID_ASKUS`;
ALTER TABLE library_smu.askus RENAME
COLUMN `Entered By` TO `Entered_By`;
ALTER TABLE library_smu.askus RENAME
COLUMN `Patron group served` TO
`GROUP_SERVED`;
ALTER TABLE library_smu.askus RENAME
COLUMN `Number of Interactions` TO
`NUM_INTERACTIONS`;
UPDATE library_smu.askus set `Date` =
STR_TO_DATE(`Date`, '%m/%d/%Y');
UPDATE library_smu.askus set `id_askus`
=  right(id_askus,8);
UPDATE `library_smu`.`instruction` SET
`Form Timestamp - Leave Blank` =
str_to_date( `Form Timestamp - Leave
Blank`, '%d-%m-%Y' );
ALTER TABLE library_smu.consultations
RENAME COLUMN `ï»¿"Id"` TO `Id`;
UPDATE library_smu.consultations set
`id` =  right(id,8);
#####################################
####################
#    Creating some individual rollup
views
#####################################
####################
#Creating rollup by library
Create table Gate_Count as
select location
  , case when name like '%Fondren%' then
'Fondren Library' else name end as
Lib_name
    , date
    , sum(visitors) as visitor_cnt
from library_smu.gatecounts  a
group by 1,2,3
;
#Creating rollup by Instruction Session
Create table Session_Count as
SELECT
  `Date Session Happened`
  , `Library / Department`
    , Mode
    , sum(Attendance)
FROM library_smu.instruction
group by 1,2,3;
```

```
;
#Rollup of semester dates, having a
case statement within a table creation
can sometimes be easer than a join to
another table
  #Since the other table had some
cleanup work, I just did a case
statement to be comprehensive
Select
lib_name
, sum(visitor_cnt) as gate_cnt
, case when cast(cast(z.cal_date as
date) as date) between '2017-01-22' and
'2017-05-31' then 'Spring'
  when cast(cast(`date` as date) as
date) between '2017-06-01' and '2017-
08-19' then 'Summer'
  when cast(cast(`date` as date) as
date) between '2017-08-20' and '2017-
12-16' then 'Fall'
  when cast(cast(`date` as date) as
date) between '2017-12-17' and '2017-
12-31' then 'Winter'
  when cast(cast(`date` as date) as
date) between '2018-01-01' and '2018-
05-15' then 'Spring'
  when cast(cast(`date` as date) as
date) between '2018-05-16' and '2018-
08-19' then 'Summer'
  when cast(cast(`date` as date) as
date) between '2018-08-20' and '2018-
12-12' then 'Fall'
  when cast(cast(`date` as date) as
date) between '2018-12-13' and '2019-
01-17' then 'Winter'
  when cast(cast(`date` as date) as
date) between '2019-01-18' and '2019-
05-14' then 'Spring'
  when cast(cast(`date` as date) as
date) between '2019-05-15' and '2019-
08-25' then 'Summer'
  when cast(cast(`date` as date) as
date) between '2019-08-26' and '2019-
12-18' then 'Fall'
  when cast(cast(`date` as date) as
date) between '2018-12-19' and '2020-
01-16' then 'Winter'
  when cast(cast(`date` as date) as
date) between '2020-01-17' and '2020-
03-15' then 'Spring'
  when cast(cast(`date` as date) as
date) between '2020-03-16' and '2020-
07-05' then 'COVID'
  when cast(cast(`date` as date) as
date) between '2020-07-06' and '2020-
08-23' then 'Summer'
  when cast(cast(`date` as date) as
date) between '2020-08-24' and '2020-
12-04' then 'Fall'
  when cast(cast(`date` as date) as
date) between '2020-12-05' and '2020-
12-31' then 'Winter'
  when cast(cast(`date` as date) as
date) between '2021-01-01' and '2021-
01-23' then 'Winter'
  when cast(cast(`date` as date) as
date) between '2021-01-24' and '2021-
05-12' then 'Spring'
  when cast(cast(`date` as date) as
date) between '2021-05-13' and '2021-
08-21' then 'Summer'
  when cast(cast(`date` as date) as
date) between '2021-08-22' and '2021-
12-04' then 'Fall'
  when cast(cast(`date` as date) as
date) between '2021-12-05' and '2021-
12-31' then 'Winter'
   else 'before scope' end as Semester
, year(date) as Years
from library_smu.gate_count
where cast(`date` as date) > '2017-08-
20'
group by 1,3,4
########################################
############################
    # SQL to develop calendar table for
SMU Library
########################################
############################
#https://stackoverflow.com/questions/10
132024/how-to-populate-a-table-with-a-
range-of-dates
########################################
############################
## Creating a table with just the dates
needed
-- DROP TABLE library_smu.incr;
CREATE TABLE library_smu.`incr` (
  `cal_date` date,
  PRIMARY KEY  (`cal_date`)
) ;
#Creating a procedure to auto increment
dates so we dont miss any dates
DROP PROCEDURE IF EXISTS filldates;
DELIMITER |
CREATE PROCEDURE filldates(dateStart
DATE, dateEnd DATE)
BEGIN
  WHILE dateStart <= dateEnd DO
    INSERT INTO incr (cal_date) VALUES
(dateStart);
    SET dateStart = date_add(dateStart,
INTERVAL 1 DAY);
  END WHILE;
```

```
END;
|
DELIMITER ;
#######################################
############################
#Call the procedure to fill out the
table
 CALL filldates('2017-06-01','2021-12-
31');
#######################################
############################
#Double check to make sure all of the
dates loaded as desired
select min(cal_date), max(cal_date)
from incr;
#######################################
############################
#Create the calendar view that will be
used on all table joins
CREATE VIEW Calendar_Reference as
    select y.cal_date
   , dayofweek(cast(y.cal_date as
date)) as Day_of_Week
   , case when
dayofweek(cast(y.cal_date as date)) = 1
then 'Sunday'
     when dayofweek(cast(y.cal_date as
date)) = 2 then 'Monday'
     when dayofweek(cast(y.cal_date as
date)) = 3 then 'Tuesday'
     when dayofweek(cast(y.cal_date as
date)) = 4 then 'Wednesday'
     when dayofweek(cast(y.cal_date as
date)) = 5 then 'Thursday'
     when dayofweek(cast(y.cal_date as
date)) = 6 then 'Friday'
     when dayofweek(cast(y.cal_date as
date)) = 7 then 'Saturday'
       else 'Error' end as Day_Name
   , week(cast(y.cal_date as date)) as
calendar_Week
   , year(cast(y.cal_date as date)) as
cal_Year
   , case when cast(y.cal_date as date)
between '2017-06-01' and '2018-05-31'
then 'FY18'
   when cast(y.cal_date as date)
between '2018-06-01' and '2019-05-31'
then 'FY19'
        when cast(y.cal_date as date)
between '2019-06-01' and '2020-05-31'
then 'FY20'
        when cast(y.cal_date as date)
between '2020-06-01' and '2021-05-31'
then 'FY21'
        when cast(y.cal_date as date)
between '2021-06-01' and '2022-05-31'
then 'FY22'
     else 'Error' end as Fiscal_Year
        , x.Semester
        -- , x.calendar_Week_1
        , x.Week_in_Semester
     from library_smu.incr y
   left join (
       select
                week(cast(cal_date as
date)) as calendar_Week_1
       , year(cast(cal_date as date))
as cal_Year
                , case when
cast(cast(z.cal_date as date) as date)
between '2017-01-22' and '2017-05-31'
then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2017-06-01' and
'2017-08-19' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2017-08-20' and
'2017-12-16' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2017-12-17' and
'2017-12-31' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2018-01-01' and
'2018-05-15' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2018-05-16' and
'2018-08-19' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2018-08-20' and
'2018-12-12' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2018-12-13' and
'2019-01-17' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2019-01-18' and
'2019-05-14' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2019-05-15' and
'2019-08-25' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2019-08-26' and
'2019-12-18' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2018-12-19' and
'2020-01-16' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2020-01-17' and
'2020-03-15' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2020-03-16' and
'2020-07-05' then 'COVID'
        when cast(cast(z.cal_date as
date) as date) between '2020-07-06' and
'2020-08-23' then 'Summer'
```

```
        when cast(cast(z.cal_date as
date) as date) between '2020-08-24' and
'2020-12-04' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2020-12-05' and
'2020-12-31' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2021-01-01' and
'2021-01-23' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2021-01-24' and
'2021-05-12' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2021-05-13' and
'2021-08-21' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2021-08-22' and
'2021-12-04' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2021-12-05' and
'2021-12-31' then 'Winter'
        else 'before scope' end as
Semester
                    , row_number()
over(partition by z1.Semester) as
Week_in_Semester
       from library_smu.incr z
                 left join
                 (select case when
cast(cast(cal_date as date) as date)
between '2017-01-22' and '2017-05-31'
then 'Spring'
        when cast(cast(cal_date as
date) as date) between '2017-06-01' and
'2017-08-19' then 'Summer'
        when cast(cast(cal_date as
date) as date) between '2017-08-20' and
'2017-12-16' then 'Fall'
        when cast(cast(cal_date as
date) as date) between '2017-12-17' and
'2017-12-31' then 'Winter'
        when cast(cast(cal_date as
date) as date) between '2018-01-01' and
'2018-05-15' then 'Spring'
        when cast(cast(cal_date as
date) as date) between '2018-05-16' and
'2018-08-19' then 'Summer'
        when cast(cast(cal_date as
date) as date) between '2018-08-20' and
'2018-12-12' then 'Fall'
        when cast(cast(cal_date as
date) as date) between '2018-12-13' and
'2019-01-17' then 'Winter'
        when cast(cast(cal_date as
date) as date) between '2019-01-18' and
'2019-05-14' then 'Spring'
        when cast(cast(cal_date as
date) as date) between '2019-05-15' and
'2019-08-25' then 'Summer'
        when cast(cast(cal_date as
date) as date) between '2019-08-26' and
'2019-12-18' then 'Fall'
        when cast(cast(cal_date as
date) as date) between '2018-12-19' and
'2020-01-16' then 'Winter'
        when cast(cast(cal_date as
date) as date) between '2020-01-17' and
'2020-03-15' then 'Spring'
        when cast(cast(cal_date as
date) as date) between '2020-03-16' and
'2020-07-05' then 'COVID'
        when cast(cast(cal_date as
date) as date) between '2020-07-06' and
'2020-08-23' then 'Summer'
        when cast(cast(cal_date as
date) as date) between '2020-08-24' and
'2020-12-04' then 'Fall'
        when cast(cast(cal_date as
date) as date) between '2020-12-05' and
'2020-12-31' then 'Winter'
        when cast(cast(cal_date as
date) as date) between '2021-01-01' and
'2021-01-23' then 'Winter'
        when cast(cast(cal_date as
date) as date) between '2021-01-24' and
'2021-05-12' then 'Spring'
        when cast(cast(cal_date as
date) as date) between '2021-05-13' and
'2021-08-21' then 'Summer'
        when cast(cast(cal_date as
date) as date) between '2021-08-22' and
'2021-12-04' then 'Fall'
        when cast(cast(cal_date as
date) as date) between '2021-12-05' and
'2021-12-31' then 'Winter'
        else 'before scope' end as
Semester
                    from
library_smu.incr
                    group by 1
                    ) z1 on z1.Semester
= case when cast(cast(z.cal_date as
date) as date) between '2017-01-22' and
'2017-05-31' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2017-06-01' and
'2017-08-19' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2017-08-20' and
'2017-12-16' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2017-12-17' and
'2017-12-31' then 'Winter'
```

```
        when cast(cast(z.cal_date as
date) as date) between '2018-01-01' and
'2018-05-15' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2018-05-16' and
'2018-08-19' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2018-08-20' and
'2018-12-12' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2018-12-13' and
'2019-01-17' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2019-01-18' and
'2019-05-14' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2019-05-15' and
'2019-08-25' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2019-08-26' and
'2019-12-18' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2018-12-19' and
'2020-01-16' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2020-01-17' and
'2020-03-15' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2020-03-16' and
'2020-07-05' then 'COVID'
        when cast(cast(z.cal_date as
date) as date) between '2020-07-06' and
'2020-08-23' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2020-08-24' and
'2020-12-04' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2020-12-05' and
'2020-12-31' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2021-01-01' and
'2021-01-23' then 'Winter'
        when cast(cast(z.cal_date as
date) as date) between '2021-01-24' and
'2021-05-12' then 'Spring'
        when cast(cast(z.cal_date as
date) as date) between '2021-05-13' and
'2021-08-21' then 'Summer'
        when cast(cast(z.cal_date as
date) as date) between '2021-08-22' and
'2021-12-04' then 'Fall'
        when cast(cast(z.cal_date as
date) as date) between '2021-12-05' and
'2021-12-31' then 'Winter'
        else 'before scope' end
       group by 1,2
     ) x
     on x.cal_Year =
year(cast(y.cal_date as date)) AND
x.calendar_Week_1 =
week(cast(y.cal_date as date))
     group by 1,2,3,4,5
     order by 1 asc;

#######################################
############################
   # Join Calendar Reference Table to
Library Database to create one
aggregated table view
#######################################
############################
select
  a.cal_date
    , a.Day_of_Week
    , a.Day_Name
    , a.Calendar_Week
    , a.Cal_Year
    , a.Fiscal_Year
    , a.semester
    , a.Week_in_Semester
  -- , b.Library
    , sum(Total_Gate) as
Gate_Count_Total_Lib
    , c.Total_Ask_Us
    , c.Chat_Ask_Us
    , c.InPerson_Ask_Us
    , c.EmailPhone_Ask_Us
    , d.Instruction_Sessions
    , d.Inst_Session_Vol_by_Day
    , e.Consultation_Cnt_by_Day
from library_smu.calendar_reference a
  left join
    (select `date`
     , sum(visitor_cnt) as Total_Gate
        from library_smu.gate_count
        group by 1) b
   on a.cal_date = b.`date`
  left join
    (select `Date`
     , sum(num_interactions) as
Total_Ask_Us
           , sum(case when `Mode of
Interaction` = 'Chat' then
num_interactions else 0 end) as
Chat_Ask_Us
     , sum(case when `Mode of
Interaction` = '' then num_interactions
else 0 end) as InPerson_Ask_Us
     , sum(case when `Mode of
Interaction` in ('Email or
Phone','Email') then num_interactions
else 0 end) as EmailPhone_Ask_Us
        from library_smu.askus
        group by 1
        )c
```

```
   on a.cal_date = c.`Date`
  left join (
   select `Date Session Happened`
     , count(id) as Instruction_Sessions
     , sum(`Attendance`) as
Inst_Session_Vol_by_Day
        from library_smu.`instruction`
        group by 1) d
   on a.cal_date = d.`Date Session
Happened`
  left join (select `Date`
   , count(id) as
Consultation_Cnt_by_Day
   from library_smu.consultations
   group by 1 )e
   on a.cal_date = e.`Date`
group by 1;
```