Course: IST-659

Name: Sharat Sripada

Homework #7

Date Submitted: 6/1/2020

Topic: Lab07 – Advanced Querying

## Part 1 – Exploratory Data Analysis

A screenshot of a social media post

Description automatically generated

**Fig: All Users in the system**

A screenshot of a social media post

Description automatically generated

**Fig: Basic Summaries**

**A screenshot of a social media post

Description automatically generated**

**Fig: Using Group-by clause**

**A screenshot of a cell phone

Description automatically generated**

**Fig: Using the Having clause**

A screenshot of a social media post

Description automatically generated

**Fig: Advanced Summaries**

## Part 2 – Putting all together

A screenshot of a computer

Description automatically generated

**Fig: Descriptive Stats**

## Appendix – SQL Queries Lab07

-- Business question:

-- Our stakeholders would like to know some things about how the users are using the system.

-- INNER-JOIN only gives vc\_UserID common in both tables:

-- - vc\_User

-- - vc\_Vidcast (834 rows)

SELECT

vc\_User.Username

, vc\_User.EmailAddress

, vc\_VidCast.vc\_VidCastID

FROM vc\_VidCast

JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

ORDER BY vc\_User.UserName

-- Below query helps identity users that were left out (2 rows)

SELECT \* FROM vc\_User

WHERE vc\_UserID NOT IN (SELECT vc\_UserID FROM vc\_Vidcast)

-- Let's use a RIGHT JOIN instead to get all users from the vc\_User table (836 rows)

SELECT

vc\_User.Username

, vc\_User.EmailAddress

, vc\_VidCast.vc\_VidCastID

FROM vc\_VidCast

RIGHT JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

ORDER BY vc\_User.UserName

-- Basic summaries

SELECT

COUNT(vc\_VidcastID) as NumberofVidCasts

, SUM(ScheduleDurationMinutes) as TotalScheduledMinutes

, MIN(ScheduleDurationMinutes) as MinScheduledMinutes

, AVG(ScheduleDurationMinutes) as AvgScheduledMinutes

, MAX(ScheduleDurationMinutes) as MaxScheduledMinutes

FROM vc\_VidCast

-- Using GROUP BY in queries

SELECT

vc\_User.Username

, vc\_User.EmailAddress

, COUNT(vc\_VidCast.vc\_VidCastID) as CountOfVidCasts

FROM vc\_VidCast

RIGHT JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

GROUP BY

vc\_User.Username

, vc\_User.EmailAddress

ORDER BY vc\_User.UserName

-- Order the results based on CountofVidCasts & Username(vc\_User table)

SELECT

vc\_User.Username

, vc\_User.EmailAddress

, COUNT(vc\_VidCast.vc\_VidCastID) as CountOfVidCasts

FROM vc\_VidCast

RIGHT JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

GROUP BY

vc\_User.Username

, vc\_User.EmailAddress

ORDER BY CountofVidCasts DESC, vc\_User.UserName

-- Stakeholders would like to their least prolific users using

-- the HAVING clause. Note that, the HAVING clause shall comprise

-- Aggregates and will follow the GROUP BY clause

SELECT

vc\_User.Username

, vc\_User.EmailAddress

, COUNT(vc\_VidCast.vc\_VidCastID) as CountOfVidCasts

FROM vc\_VidCast

RIGHT JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

GROUP BY

vc\_User.Username

, vc\_User.EmailAddress

HAVING COUNT(vc\_VidCast.vc\_VidCastID) < 10

ORDER BY CountofVidCasts DESC, vc\_User.UserName

-- Advanced Summaries

-- Actual duration of finished VidCasts

SELECT

vc\_User.UserName

, vc\_User.EmailAddress

, SUM(DateDiff(n, StartDateTime, EndDateTime)) as SumActualDurationMinutes

FROM vc\_VidCast

JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

JOIN vc\_Status ON vc\_Status.vc\_StatusID = vc\_VidCast.vc\_StatusID

WHERE vc\_Status.StatusText = 'Finished'

GROUP BY

vc\_User.UserName

, vc\_User.EmailAddress

ORDER BY vc\_User.UserName

-- Part2

SELECT

vc\_User.UserName

, vc\_User.EmailAddress

, SUM(DateDiff(n, StartDateTime, EndDateTime)) as SumActualDurationMinutes

, COUNT(vc\_VidCast.vc\_VidCastID) as CountOfVidCasts

, MIN(DateDiff(n, StartDateTime, EndDateTime)) as MinActualDurationMinutes

, AVG(DateDiff(n, StartDateTime, EndDateTime)) as AvgActualDurationMinutes

, MAX(DateDiff(n, StartDateTime, EndDateTime)) as MaxActualDurationMinutes

FROM vc\_VidCast

JOIN vc\_User ON vc\_User.vc\_UserID = vc\_VidCast.vc\_UserID

JOIN vc\_Status ON vc\_Status.vc\_StatusID = vc\_VidCast.vc\_StatusID

WHERE vc\_Status.StatusText = 'Finished'

GROUP BY

vc\_User.UserName

, vc\_User.EmailAddress

ORDER BY COUNT(vc\_VidCast.vc\_VidCastID) DESC, vc\_User.UserName