Course: IST-659
Name: Sharat Sripada
Homework #7
Date Submitted: 6/1/2020
Topic: Lab07 – Advanced Querying

Part 1 – Exploratory Data Analysis

Fig: All Users in the system

Fig: Basic Summaries

```sql
38  SELECT
39      vc_User.Username
40      , vc_User.EmailAddress
41      , COUNT(vc_VidCast.vc_VidCastID) as CountOfVidCasts
42  FROM vc_VidCast
43  RIGHT JOIN vc_User ON vc_User.vc_UserID = vc_VidCast.vc_UserID
44  GROUP BY
45      vc_User.Username
46      , vc_User.EmailAddress
47  ORDER BY vc_User.UserName
48
49  -- Order the results based on CountofVidCasts & Username(vc_User table)
50  SELECT
```

Username	EmailAddress	CountOfVidCasts
accurate	In@facilisiseget.co.uk	9
albite	nisl@vitaemauris.org	13
architect	a.dui.Cras@mi.edu	12
archives	ullamcorper.velit@interdumfeugiatSed.com	16
bedtime	enim.Etiam@egetmollislectus.edu	11
bewildered	Donec.porttitor.tellus@odioAliquamvulputate.edu	10
bicycle	Quisque.porttitor.eros@mi.net	8
bid	sed.turpis@hymenaeosMaurisut.co.uk	15
bittern	in.consequat@loremsemper.edu	14
camel	mauris@massanon.edu	10
canadian	Curabitur.dictum.Phasellus@eleifendnec.com	18
carpal	In.faucibus.Morbi@Mauris.ca	11
chef	ultricies.sem@estMauris.edu	14
console	tristique@justoeuarcu.com	6
darcy	urna.justo@orci.edu	9
dispatcher	quam@aptenttacitisociosqu.ca	8
DnDGal	dnd@nodomain.xyz	14
doughnut	ipsum.primis@Cumsociis.com	16
ecozone	placerat@netusetmalesuada.net	13

Query executed successfully.   ist-s-students.syr.edu (12.... AD\vssripad (71) IST659_M409_vssripad 00:00:00 68 rows

Fig: Using Group-by clause

```
56     GROUP BY
57         vc_User.Username
58         , vc_User.EmailAddress
59     ORDER BY CountofVidCasts DESC, vc_User.UserName
60
61  -- Stakeholders would like to their least prolific users using
62  -- the HAVING clause. Note that, the HAVING clause shall comprise
63  -- Aggregates and will follow the GROUP BY clause
64  SELECT
65         vc_User.Username
66         , vc_User.EmailAddress
67         , COUNT(vc_VidCast.vc_VidCastID) as CountOfVidCasts
68     FROM vc_VidCast
69     RIGHT JOIN vc_User ON vc_User.vc_UserID = vc_VidCast.vc_UserID
70     GROUP BY
71         vc_User.Username
72         , vc_User.EmailAddress
73     HAVING COUNT(vc_VidCast.vc_VidCastID) < 10
74     ORDER BY CountofVidCasts DESC, vc_User.UserName
75
76
77  -- Advanced Summaries
78  -- Actual duration of finished VidCasts
79  SELECT
80         vc_User.UserName
```

.00 %

Results Messages

	Username	EmailAddress	CountOfVidCasts
1	accurate	In@facilisiseget.co.uk	9
2	darcy	uma.justo@orci.edu	9
3	gum	ut@pharetraQuisqueac.com	9
4	spilling	ullamcorper@Mauris.net	9
5	bicycle	Quisque.porttitor.eros@mi.net	8
6	dispatcher	quam@aptenttacitisociosqu.ca	8
7	hygienist	magna.Ut@necumasuscipit.ca	7
8	stay	et.magnis@nonmagnaNam.co.uk	7
9	console	tristique@justoeuarcu.com	6
10	winter	accumsan@ascelerisque.net	6
11	embarrass	Nam.ligula@atfringilla.co.uk	0
12	prune	enim.sit.amet@aliquet.edu	0

Fig: Using the Having clause
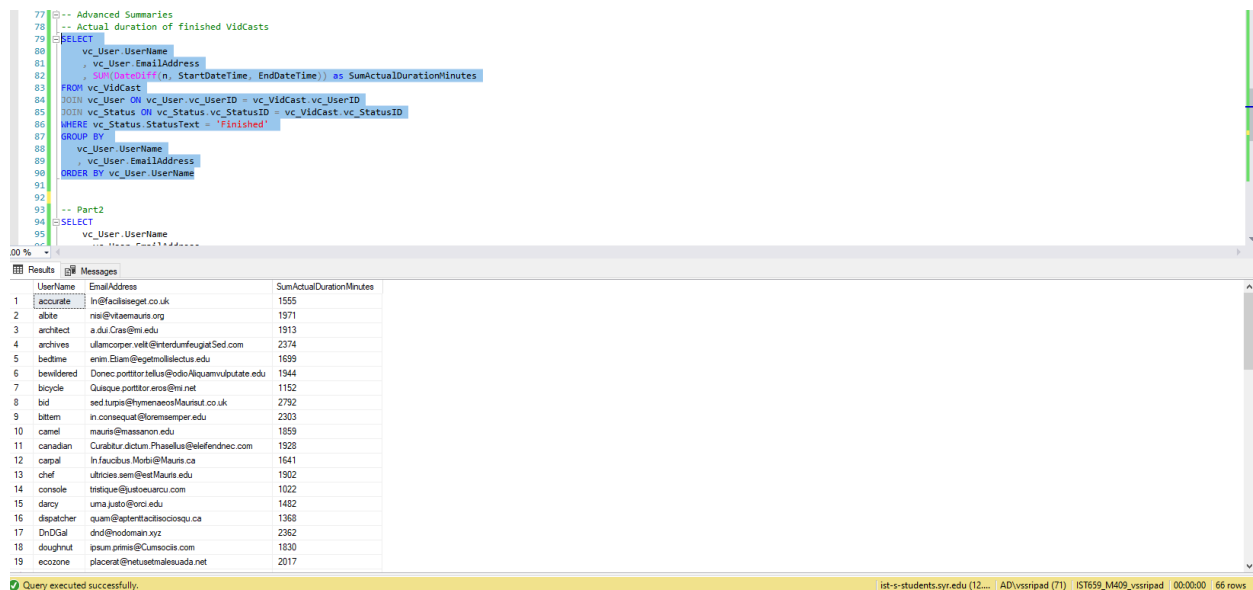
Part 2 – Putting all together
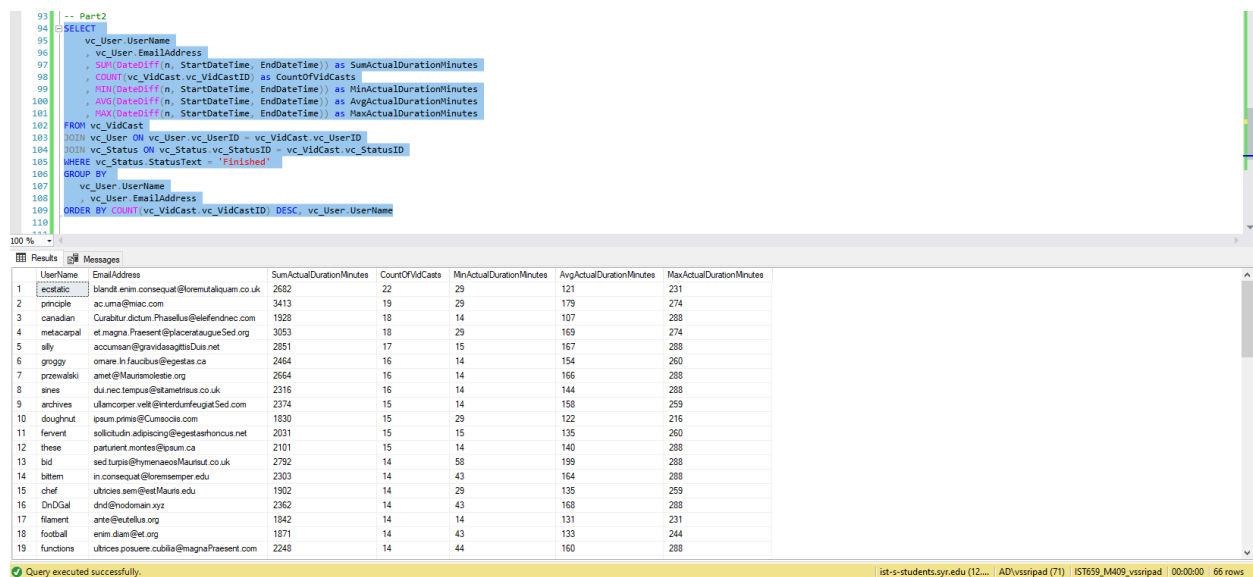
Appendix – SQL Queries Lab07

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

```sql
-- 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
```

```sql
-- 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
```