# Big Data Assignment-4

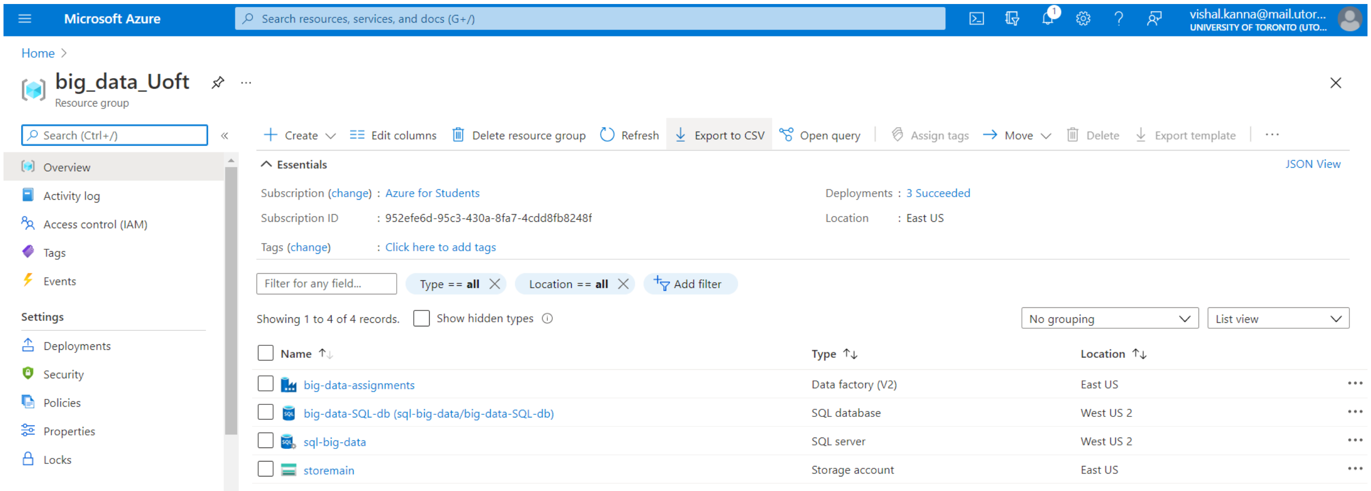
## Hruday Vishal Kanna Anand

## 1006874517

## Part-a

## 1.

## Resources created in resource group



big_data_Uoft

Resource group

Essentials

Subscription (change) : Azure for Students

Subscription ID : 952efe6d-95c3-430a-8fa7-4cdd8fb8248f

Tags (change) : Click here to add tags

Deployments : 3 Succeeded

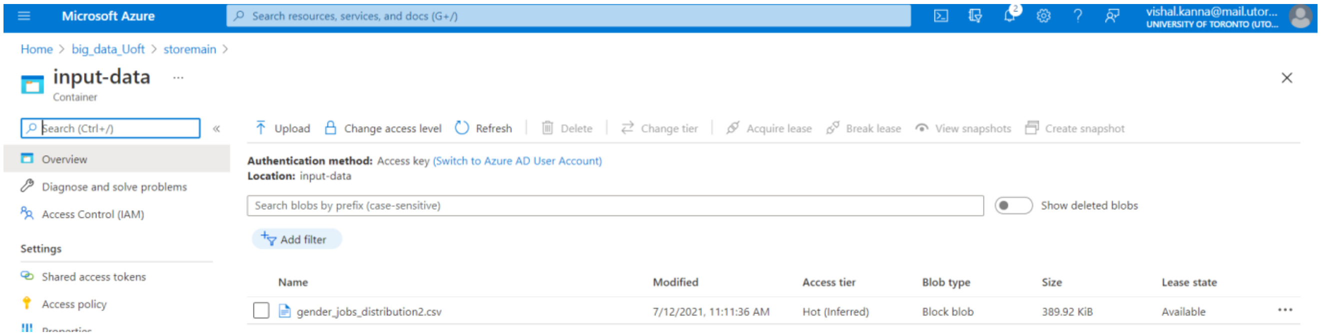
Location : East US

Showing 1 to 4 of 4 records. □ Show hidden types

| Name | Type | Location |
|---|---|---|
| big-data-assignments | Data factory (V2) | East US |
| big-data-SQL-db (sql-big-data/big-data-SQL-db) | SQL database | West US 2 |
| sql-big-data | SQL server | West US 2 |
| storemain | Storage account | East US |

## 2.

## The input data is copied into the blob storage



input-data

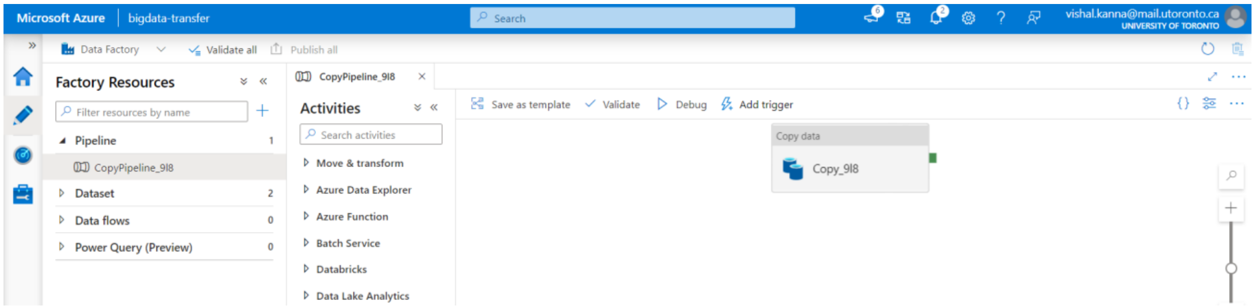
Container

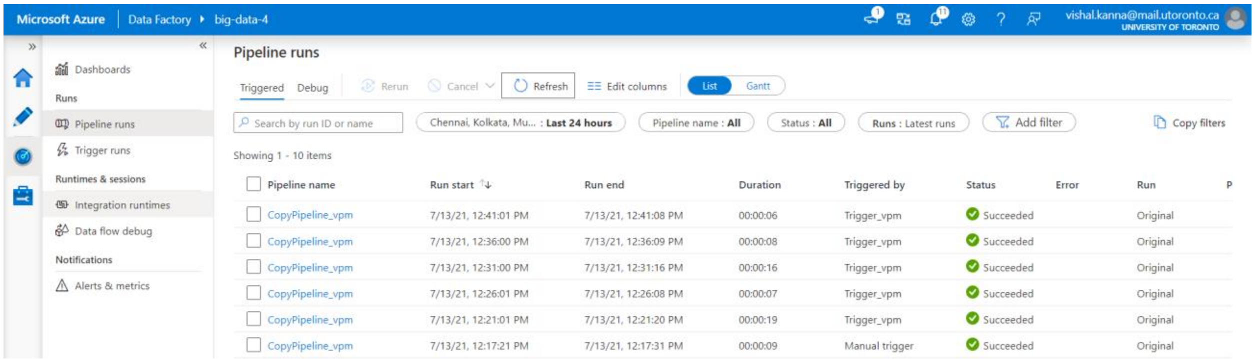
**Authentication method:** Access key (Switch to Azure AD User Account)

**Location:** input-data

| Name | Modified | Access tier | Blob type | Size | Lease state |
|---|---|---|---|---|---|
| gender_jobs_distribution2.csv | 7/12/2021, 11:11:36 AM | Hot (Inferred) | Block blob | 389.92 KiB | Available |

**The pipeline created in azure data factory is executed manually-** the name of the pipeline in the first picture and the pipeline executed in the monitor view is different as they were done at different times and I had deleted all my resources in between.

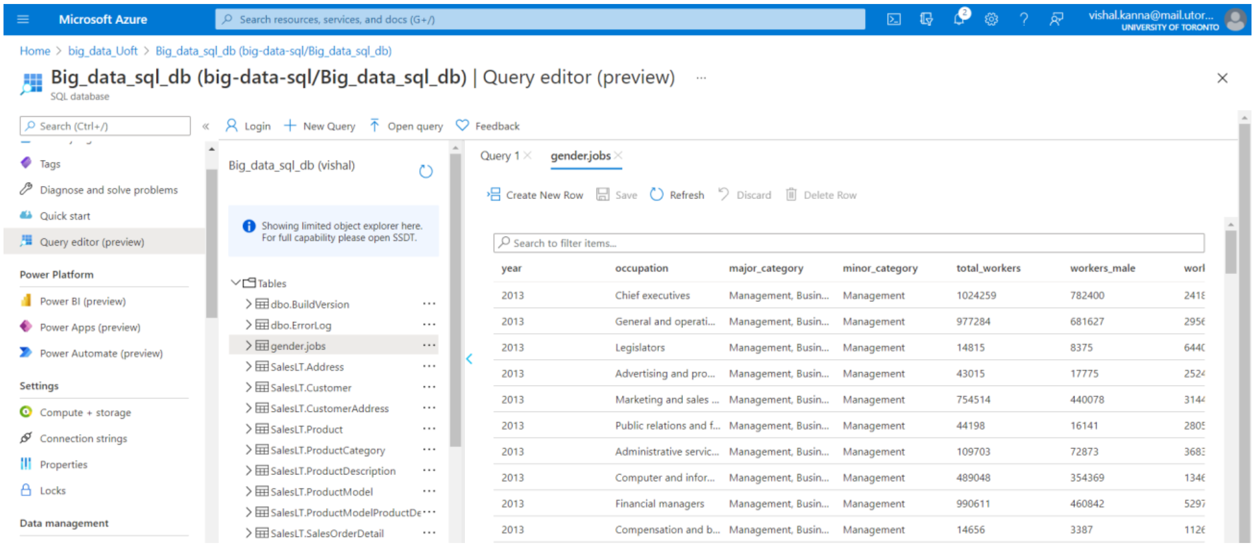




Showing 1 - 10 items

| ☐ Pipeline name | Run start | Run end | Duration | Triggered by | Status | Error | Run | P |
|---|---|---|---|---|---|---|---|---|
| ☐ CopyPipeline_vpm | 7/13/21, 12:41:01 PM | 7/13/21, 12:41:08 PM | 00:00:06 | Trigger_vpm | ✓ Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:36:00 PM | 7/13/21, 12:36:09 PM | 00:00:08 | Trigger_vpm | ✓ Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:31:00 PM | 7/13/21, 12:31:16 PM | 00:00:16 | Trigger_vpm | ✓ Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:26:01 PM | 7/13/21, 12:26:08 PM | 00:00:07 | Trigger_vpm | ✓ Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:21:01 PM | 7/13/21, 12:21:20 PM | 00:00:19 | Trigger_vpm | ✓ Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:17:21 PM | 7/13/21, 12:17:31 PM | 00:00:09 | Manual trigger | ✓ Succeeded | | Original | |

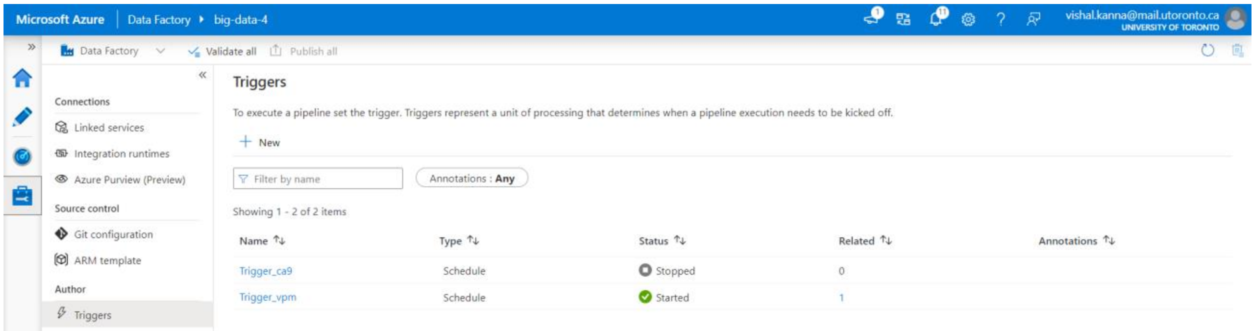
**The data is copied into SQL DB using azure data factory pipeline**



| year | occupation | major_category | minor_category | total_workers | workers_male | worl |
|---|---|---|---|---|---|---|
| 2013 | Chief executives | Management, Busin... | Management | 1024259 | 782400 | 241 |
| 2013 | General and operati... | Management, Busin... | Management | 977284 | 681627 | 295 |
| 2013 | Legislators | Management, Busin... | Management | 14815 | 8375 | 644 |
| 2013 | Advertising and pro... | Management, Busin... | Management | 43015 | 17775 | 252 |
| 2013 | Marketing and sales ... | Management, Busin... | Management | 754514 | 440078 | 314 |
| 2013 | Public relations and f... | Management, Busin... | Management | 44198 | 16141 | 280 |
| 2013 | Administrative servic... | Management, Busin... | Management | 109703 | 72873 | 368 |
| 2013 | Computer and infor... | Management, Busin... | Management | 489048 | 354369 | 134 |
| 2013 | Financial managers | Management, Busin... | Management | 990611 | 460842 | 529 |
| 2013 | Compensation and b... | Management, Busin... | Management | 14656 | 3387 | 112 |

**3.**

## The schedule trigger is created

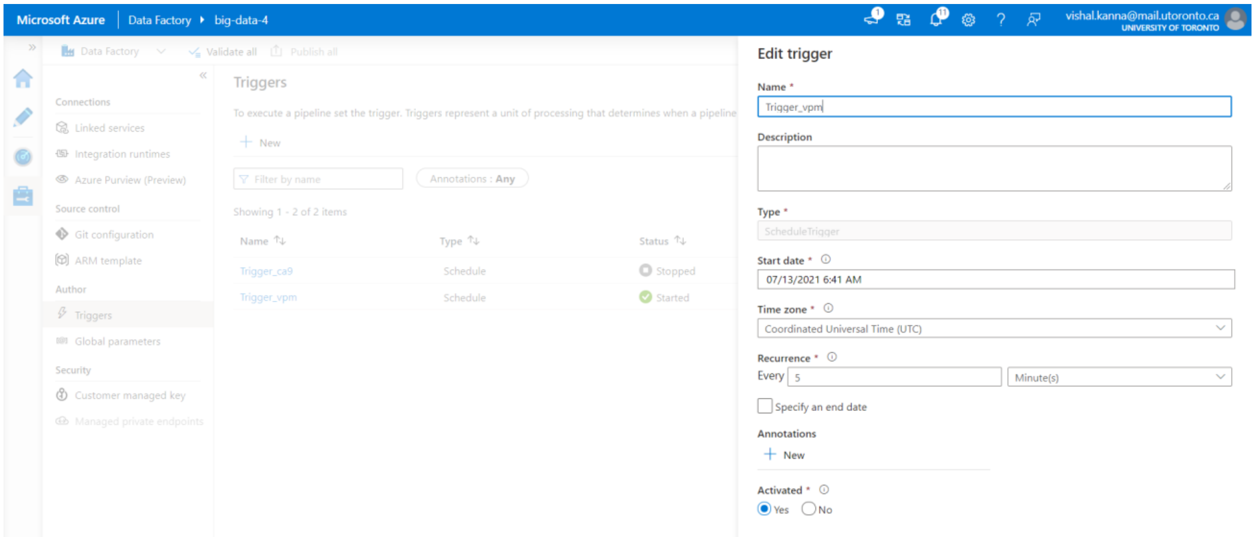


Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

Showing 1 - 2 of 2 items

| Name | Type | Status | Related | Annotations |
|---|---|---|---|---|
| Trigger_ca9 | Schedule | Stopped | 0 | |
| Trigger_vpm | Schedule | Started | 1 | |



Edit trigger

Name * Trigger_vpm

Description

Type * ScheduleTrigger

Start date * 07/13/2021 6:41 AM

Time zone * Coordinated Universal Time (UTC)

Recurrence * Every 5 Minute(s)

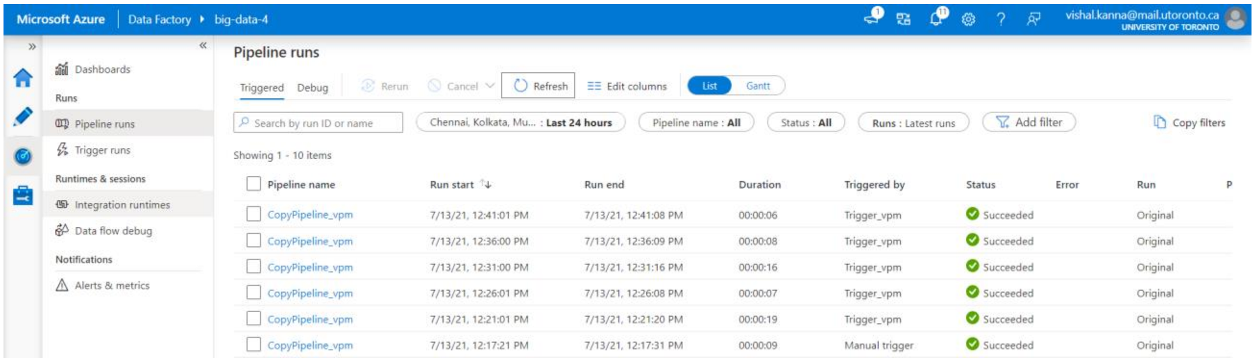
☐ Specify an end date

Annotations

New

Activated * ● Yes ○ No

## The trigger is executed 5 times



Pipeline runs

Chennai, Kolkata, Mu... : Last 24 hours · Pipeline name : All · Status : All · Runs : Latest runs

Showing 1 - 10 items

| ☐ Pipeline name | Run start | Run end | Duration | Triggered by | Status | Error | Run | P |
|---|---|---|---|---|---|---|---|---|
| ☐ CopyPipeline_vpm | 7/13/21, 12:41:01 PM | 7/13/21, 12:41:08 PM | 00:00:06 | Trigger_vpm | Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:36:00 PM | 7/13/21, 12:36:09 PM | 00:00:08 | Trigger_vpm | Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:31:00 PM | 7/13/21, 12:31:16 PM | 00:00:16 | Trigger_vpm | Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:26:01 PM | 7/13/21, 12:26:08 PM | 00:00:07 | Trigger_vpm | Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:21:01 PM | 7/13/21, 12:21:20 PM | 00:00:19 | Trigger_vpm | Succeeded | | Original | |
| ☐ CopyPipeline_vpm | 7/13/21, 12:17:21 PM | 7/13/21, 12:17:31 PM | 00:00:09 | Manual trigger | Succeeded | | Original | |

There are mainly 4 types of triggers in data factory-

1. Manual trigger
2. Schedule trigger
3. Tumbling window trigger

4. Event based trigger

Manual trigger is a trigger that activated by the user manually by either using the UI or a Powershell script.

Schedule trigger runs a pipeline using generic clock timing such as hours, minutes, weeks, etc. It can be a simple timing such as weekly or more complex such as every Monday at 2:00 P.M. There can also be many schedule triggers for a single pipeline.

Tumbling window trigger is similar to a schedule trigger as it also runs the pipeline at specific time intervals but with the inclusion of state. So, in case of failure it can automatically try again as it stores past states making this more reliable. There can only be one tumbling window trigger for a pipeline.

Event based trigger runs a pipeline in response to an event. There are 2 types of event triggers-

1. storage based- triggered when a change occurs in a storage account
2. custom event

**4.**

They would first need to create ADLS gen 2 accounts in both Canada central and west Europe regions. This can be done very similar to creating storage accounts. We will specify the region as either Canada central or west Europe and we will enable hierarchical namespace for both the accounts. This will make them ADLS gen 2 accounts.

We will then have to create data factory account to facilitate the data transfer from one ADLS to another. The data factory can communicate with the ADLS's with the help of a linked service that must be created to both the data lakes.

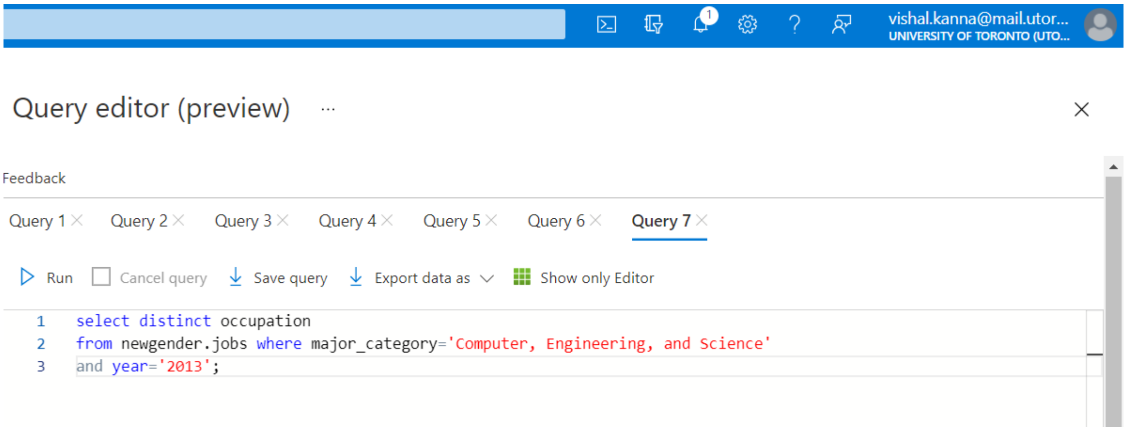
Now we must create a pipeline for transferring files (objects) from one ADLS to another ADLS. Let's say we transfer from Canada central to west Europe. this pipeline will be run with a storage event trigger; hence it will be triggered when a new file is added to the Canada central ADLS. This trigger will have parameters of the file that caused trigger with the help of @triggerbody().folderpath and @triggerbody().filename. Using these parameters, we run the first activity of our pipeline which is validation. This is done to check if west Europe ADLS has the folder and file path that triggered the pipeline. If the folder and file is found in west Europe ADLS then the pipeline proceeds to do nothing (wait activity) as we don't want replicate files to be created by multiple copies. If it is not found, then a copy data activity is done where data is copied from Canada central ADLS using the trigger properties above to the same folder and file as it was in Canada central to west Europe. We can see this method prevents unnecessary copies of files if the file already exists in the other ADLS.

As we want this to be bi directional we set up a similar pipeline but this time it will be from west Europe ADLS to Canada central ADLS following the same steps as above only switching the data flow path(west Europe to Canada central) and the validation will be done on Canada central and the storage event trigger will be based on west Europe ADLS.

**Part-b**

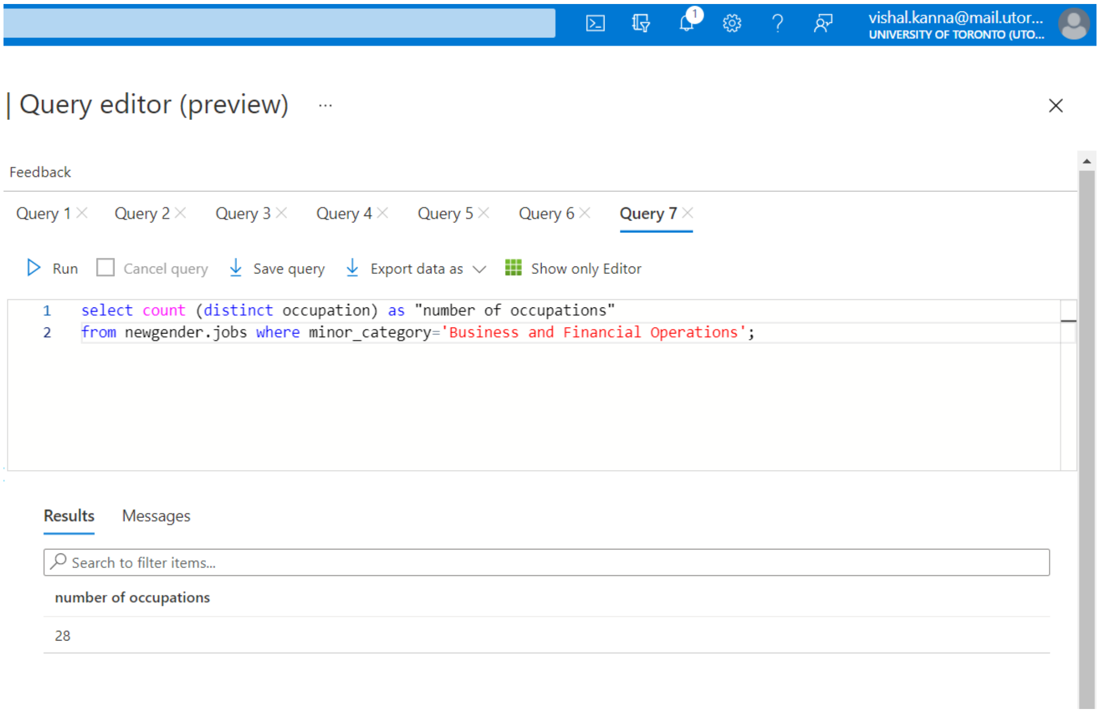
**1.**

**The output is attached in a csv file**



```sql
select distinct occupation
from newgender.jobs where major_category='Computer, Engineering, and Science'
and year='2013';
```

**2.**

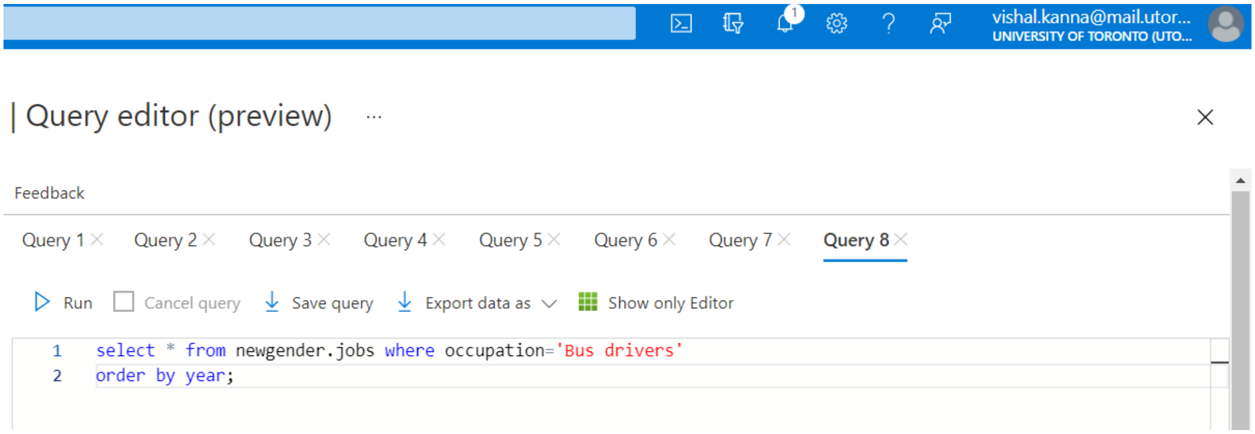


```sql
select count (distinct occupation) as "number of occupations"
from newgender.jobs where minor_category='Business and Financial Operations';
```

| number of occupations |
|---|
| 28 |

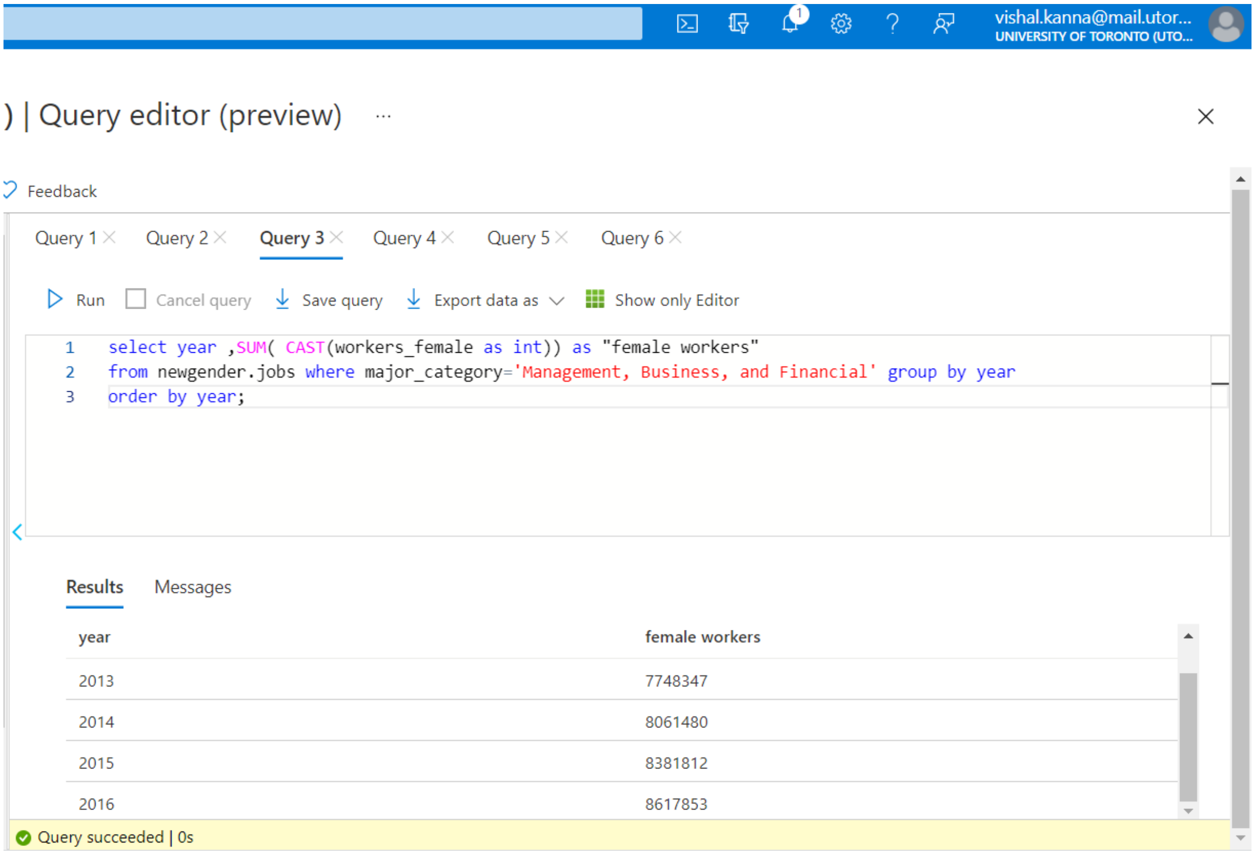
## 3.

**The output is attached in a csv file**



```
select * from newgender.jobs where occupation='Bus drivers'
order by year;
```

## 4.

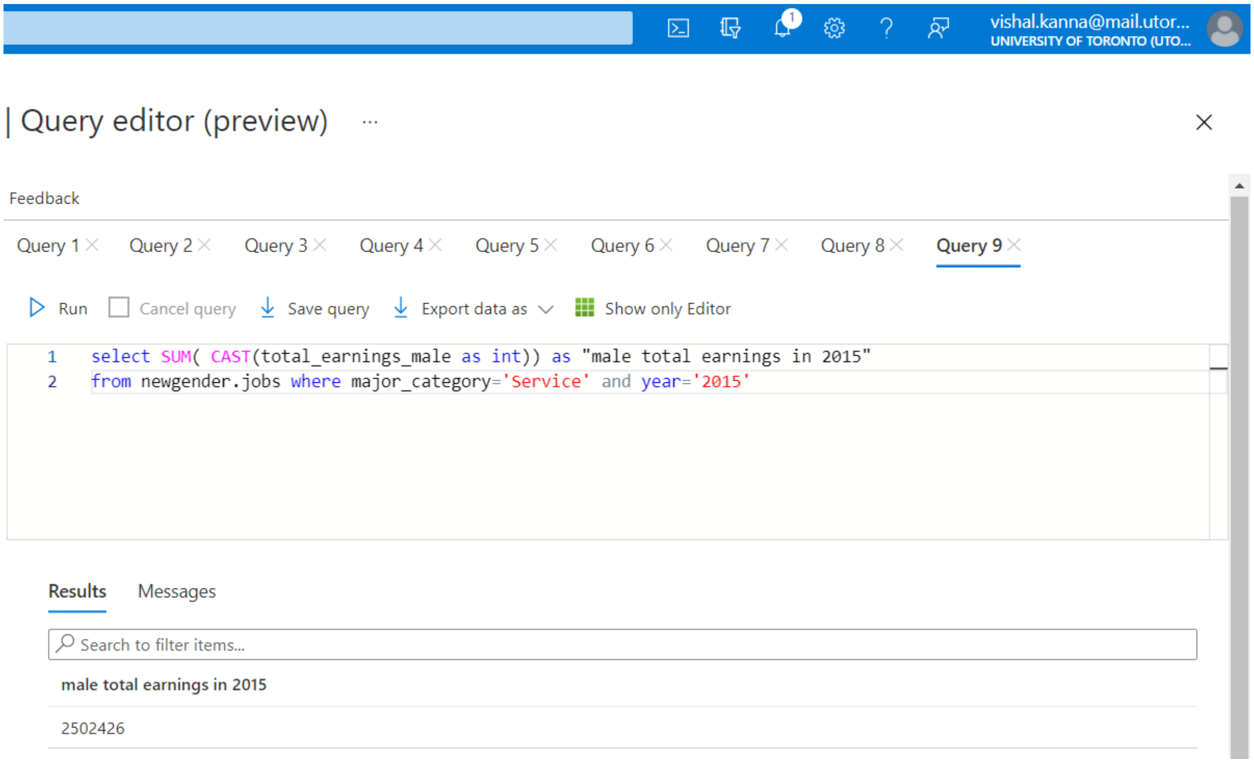


```
select year ,SUM( CAST(workers_female as int)) as "female workers"
from newgender.jobs where major_category='Management, Business, and Financial' group by year
order by year;
```

| year | female workers |
|---|---|
| 2013 | 7748347 |
| 2014 | 8061480 |
| 2015 | 8381812 |
| 2016 | 8617853 |

Query succeeded | 0s

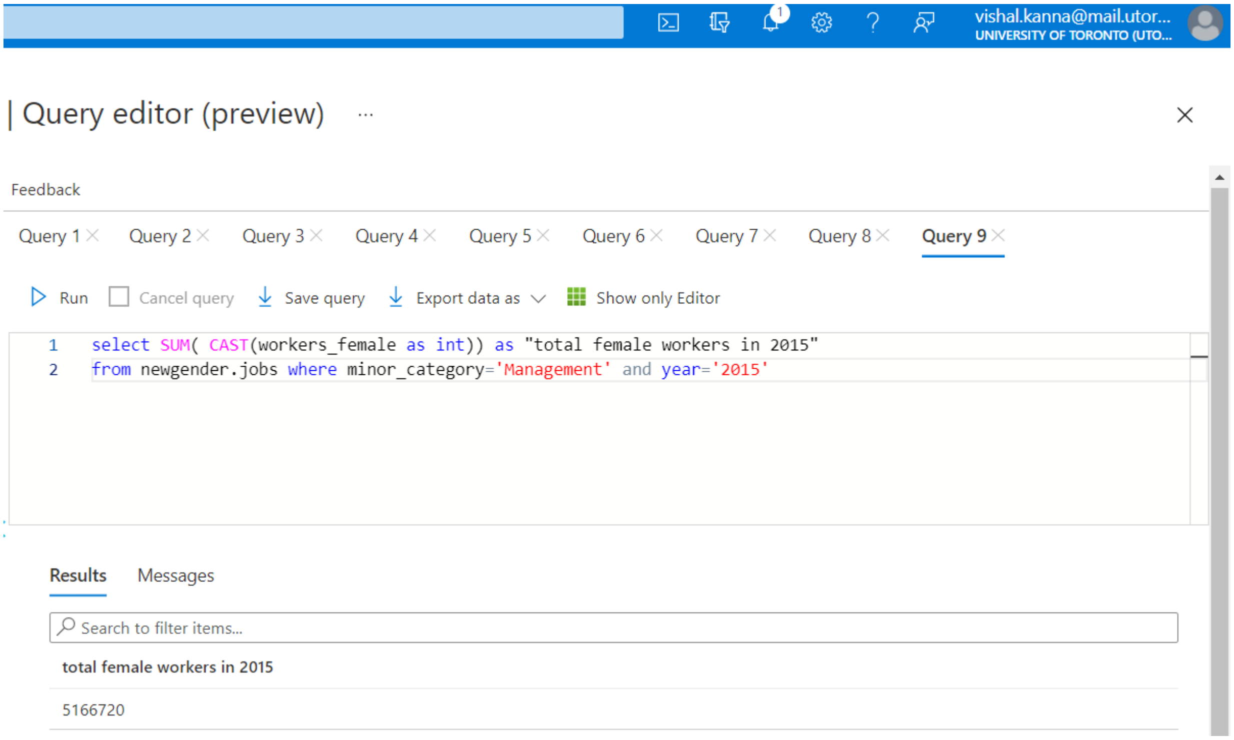
**5.**



```
select SUM( CAST(total_earnings_male as int)) as "male total earnings in 2015"
from newgender.jobs where major_category='Service' and year='2015'
```

```
male total earnings in 2015
2502426
```

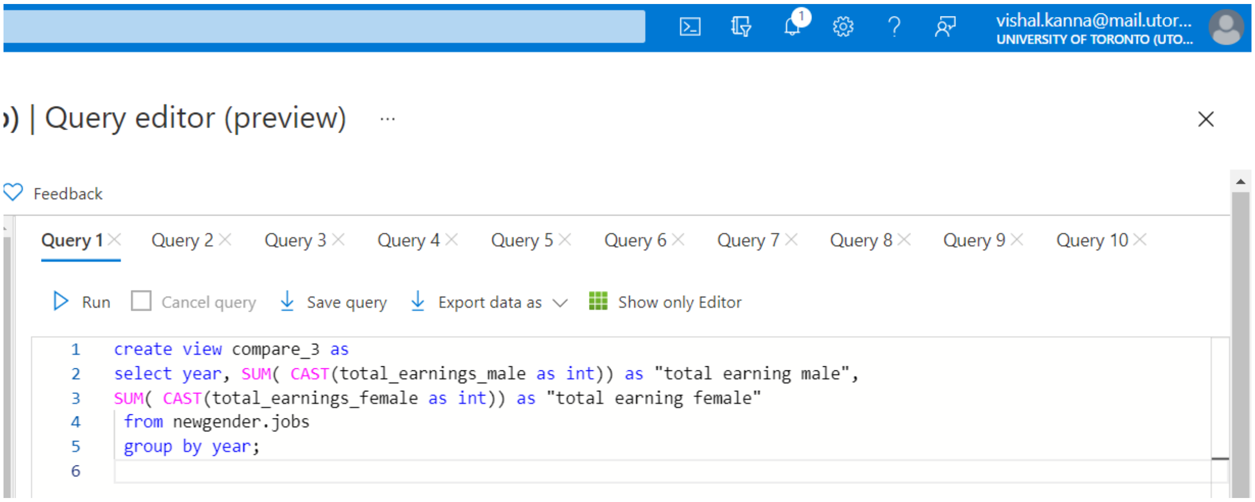
**6.**



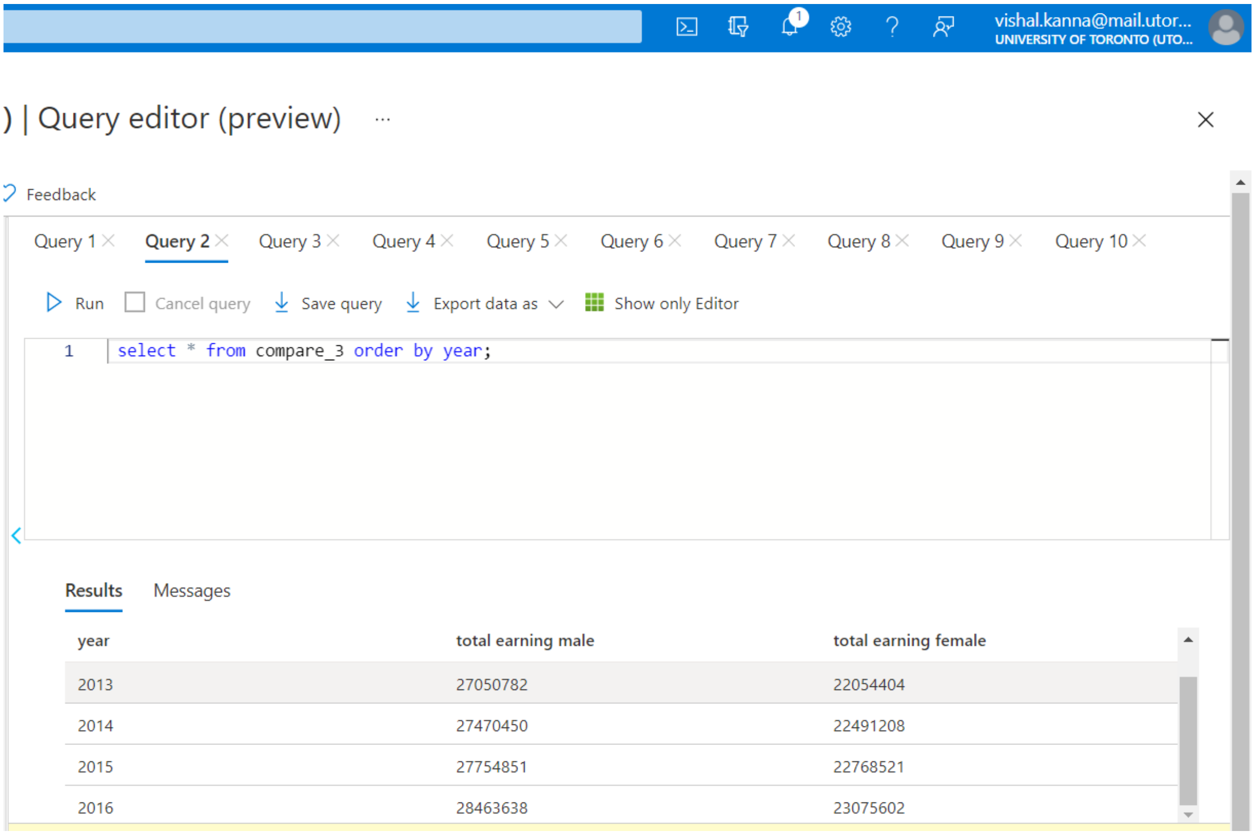
```
select SUM( CAST(workers_female as int)) as "total female workers in 2015"
from newgender.jobs where minor_category='Management' and year='2015'
```

```
total female workers in 2015
5166720
```

**7.**



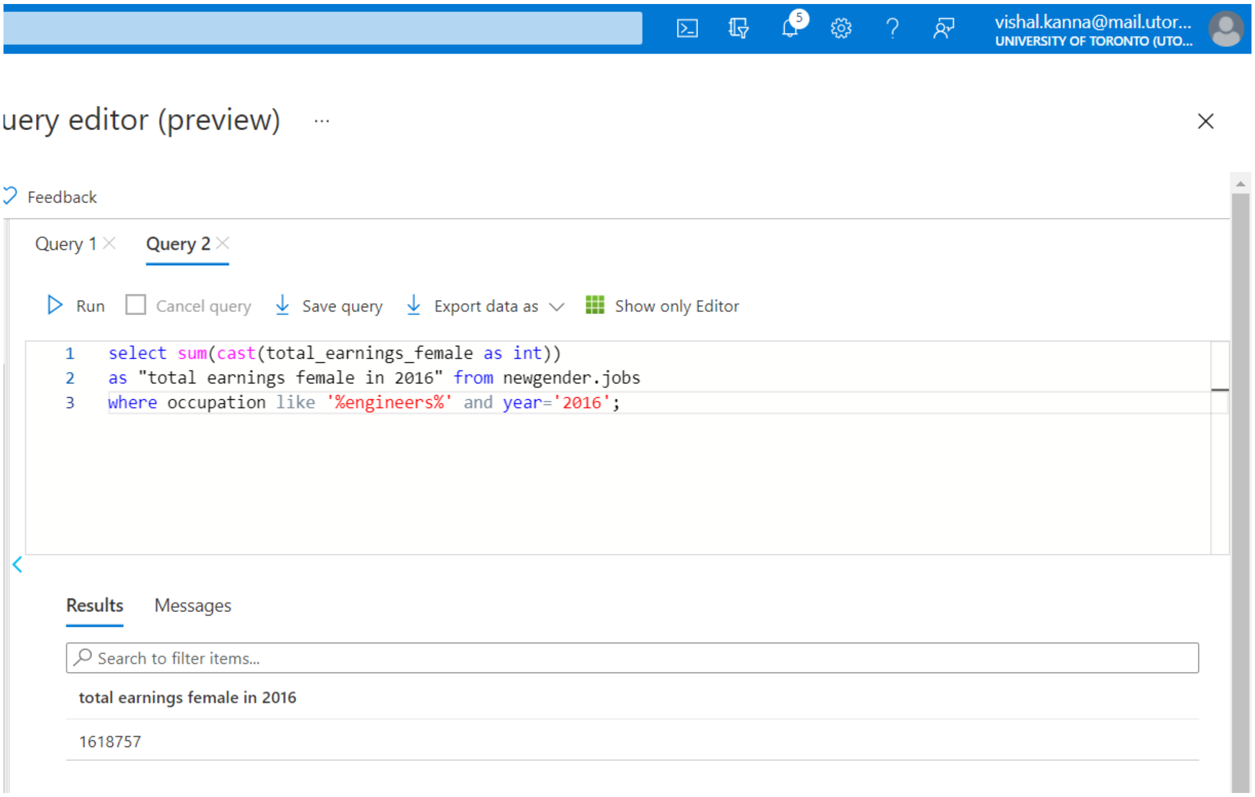
```
create view compare_3 as
select year, SUM( CAST(total_earnings_male as int)) as "total earning male",
SUM( CAST(total_earnings_female as int)) as "total earning female"
 from newgender.jobs
 group by year;
```



```
select * from compare_3 order by year;
```

| year | total earning male | total earning female |
|---|---|---|
| 2013 | 27050782 | 22054404 |
| 2014 | 27470450 | 22491208 |
| 2015 | 27754851 | 22768521 |
| 2016 | 28463638 | 23075602 |

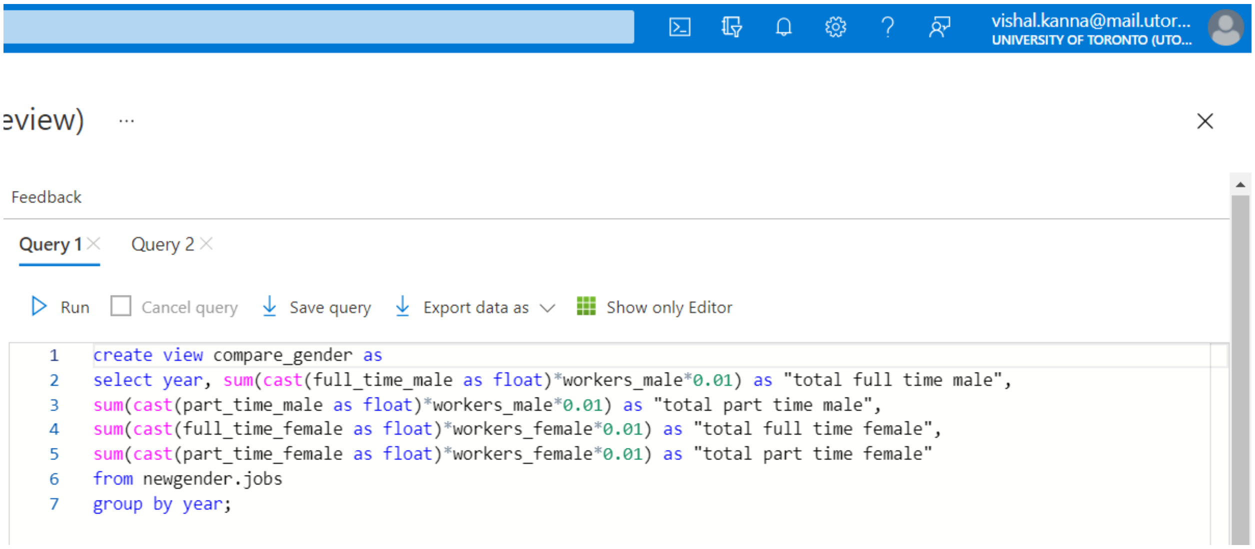
**8.**



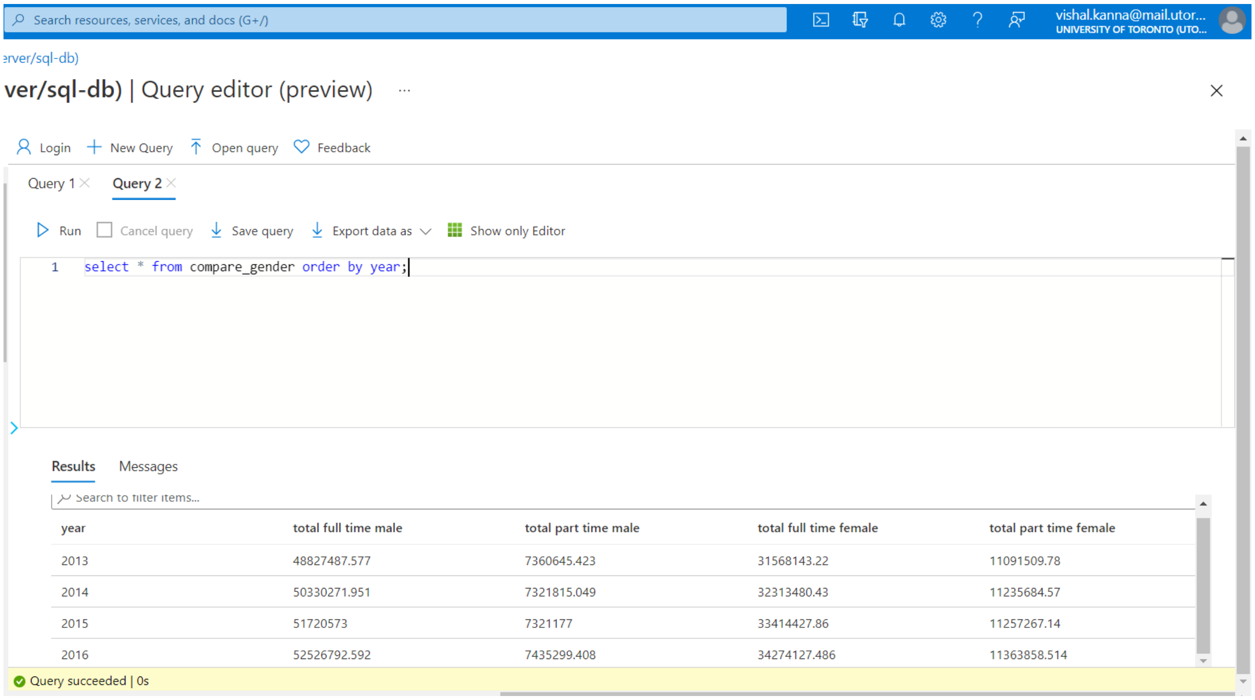
```sql
select sum(cast(total_earnings_female as int))
as "total earnings female in 2016" from newgender.jobs
where occupation like '%engineers%' and year='2016';
```

| total earnings female in 2016 |
|---|
| 1618757 |

**9.**



```sql
create view compare_gender as
select year, sum(cast(full_time_male as float)*workers_male*0.01) as "total full time male",
sum(cast(part_time_male as float)*workers_male*0.01) as "total part time male",
sum(cast(full_time_female as float)*workers_female*0.01) as "total full time female",
sum(cast(part_time_female as float)*workers_female*0.01) as "total part time female"
from newgender.jobs
group by year;
```


Search resources, services, and docs (G+/)
vishal.kanna@mail.utor...
UNIVERSITY OF TORONTO (UTO...
erver/sql-db)
ver/sql-db) | Query editor (preview)
Login
New Query
Open query
Feedback
Query 1
Query 2
Run
Cancel query
Save query
Export data as
Show only Editor
Results
Messages
Search to filter items...

```
select * from compare_gender order by year;
```

| year | total full time male | total part time male | total full time female | total part time female |
|---|---|---|---|---|
| 2013 | 48827487.577 | 7360645.423 | 31568143.22 | 11091509.78 |
| 2014 | 50330271.951 | 7321815.049 | 32313480.43 | 11235684.57 |
| 2015 | 51720573 | 7321177 | 33414427.86 | 11257267.14 |
| 2016 | 52526792.592 | 7435299.408 | 34274127.486 | 11363858.514 |

Query succeeded | 0s