

Assignment 4 Report

Aditya Jain

1 Part A

1.1 Part A.1

Figure 1 shows the deployment of the three resources.

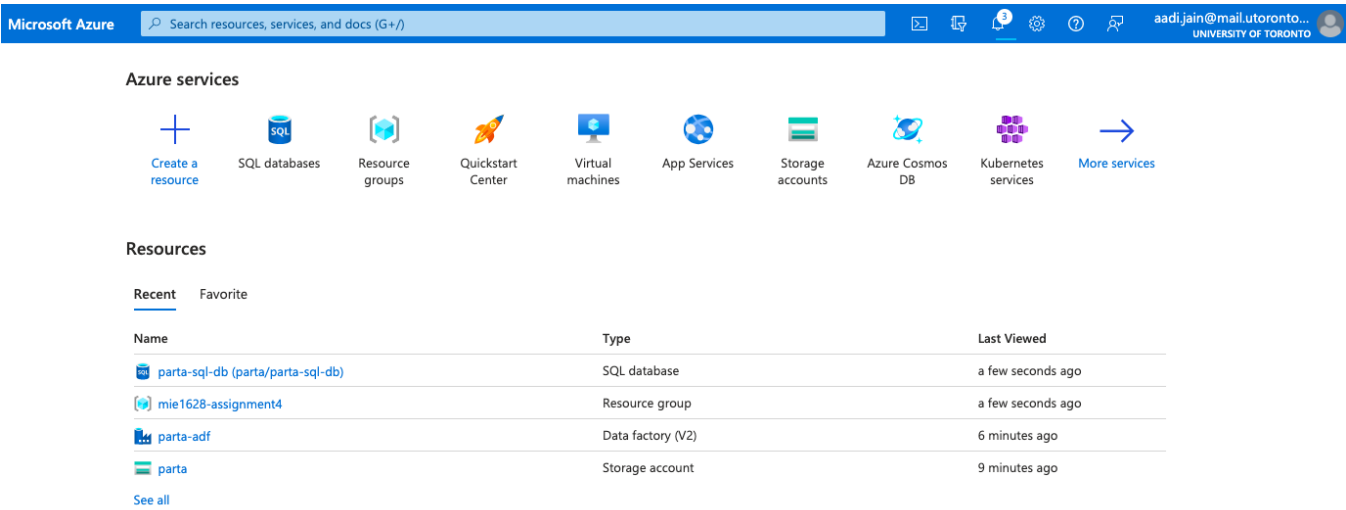
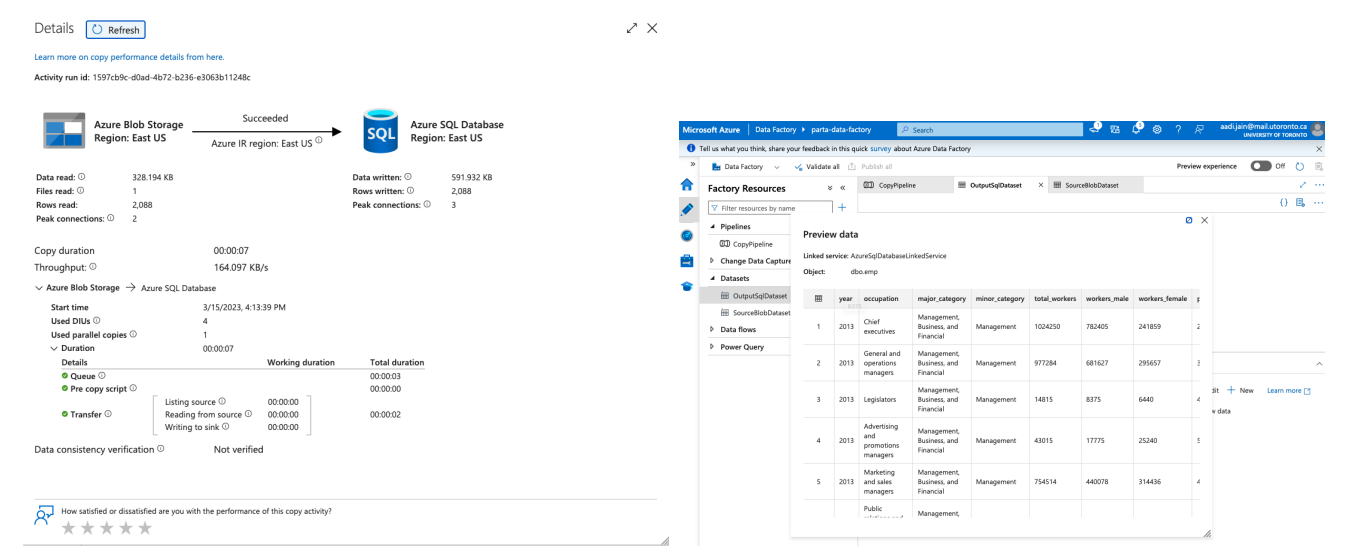


Figure 1: Result of Part A.1

1.2 Part A.2

Figure 2 shows successful copy pipeline from Blob storage to SQL database.



(a) Successful copy pipeline

(b) Preview of data in Azure SQL DB after copy

Figure 2: Result of Part A.2

1.3 Part A.3

Figure 3 shows five successful runs of scheduled trigger.

The screenshot displays the 'Trigger runs' section of the Microsoft Azure Data Factory portal. It shows a table with five rows, each representing a successful run of a 'RunEveryThreeMinutes' trigger. The columns include Trigger name, Trigger type, Trigger time, Status, Pipelines, Run, Message, Properties, and Run ID. All runs are marked as 'Succeeded'.

Trigger name	Trigger type	Trigger time	Status	Pipelines	Run	Message	Properties	Run ID
RunEveryThreeMinutes	Schedule trigger	3/15/2023, 4:35:00 PM	Succeeded	1	Original			085852269438537158079302
RunEveryThreeMinutes	Schedule trigger	3/15/2023, 4:32:00 PM	Succeeded	1	Original			08585226945651524019866
RunEveryThreeMinutes	Schedule trigger	3/15/2023, 4:29:00 PM	Succeeded	1	Original			085852269474545178339638
RunEveryThreeMinutes	Schedule trigger	3/15/2023, 4:26:00 PM	Succeeded	1	Original			085852269492471440908821
RunEveryThreeMinutes	Schedule trigger	3/15/2023, 4:23:00 PM	Succeeded	1	Original			085852269510507605162811

Figure 3: Result of Part A.3

There are three different types of triggers available in ADF:

- **Schedule trigger:** A trigger that invokes a pipeline at a scheduled time on the clock.
- **Tumbling window trigger:** A type of trigger that fires at a periodic interval from a specified start time, while retaining state. This trigger supports only one-to-one relationship and can be scheduled for windows in the past, unlike schedule trigger.
- **Event-based trigger:** A trigger that is executed in response to an event, such as arrival or deletion of a file in Azure Blob Storage account.

1.4 Part A.4

ADLS Gen 2 is built on top of Azure Blob Storage, hence, the process of replicating data will be quite similar to the one we implemented in the previous part. Using Azure Data Factory, we can create a copy pipeline (similar to Part A.2) to transfer data from ADLS Gen 2 in Canada Central to ADLS Gen 2 in West Europe but using an event-based trigger this time. When a new object be uploaded to Canada Central or West Europe, the pipeline will be triggered and the data will be copied to other other storage.

2 Part B

The file being used for this part is *gender_jobs_distribution2.csv*. Figure 4 shows the result for part B1.

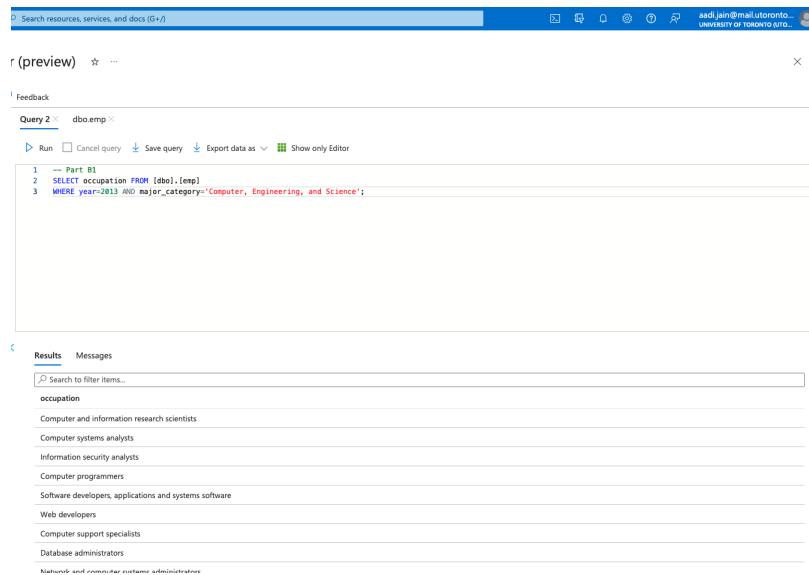


Figure 4: Result of Part B.1

Figure 5 shows the result for part B2.

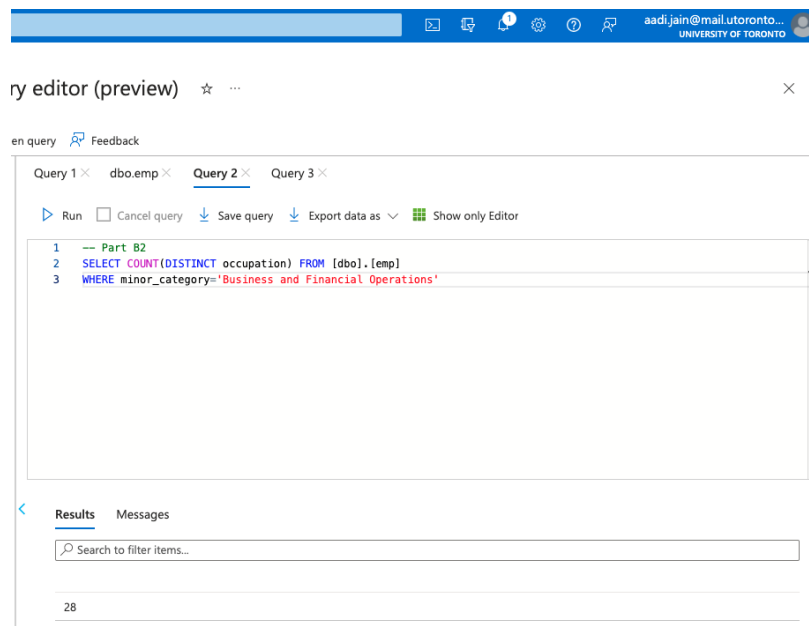
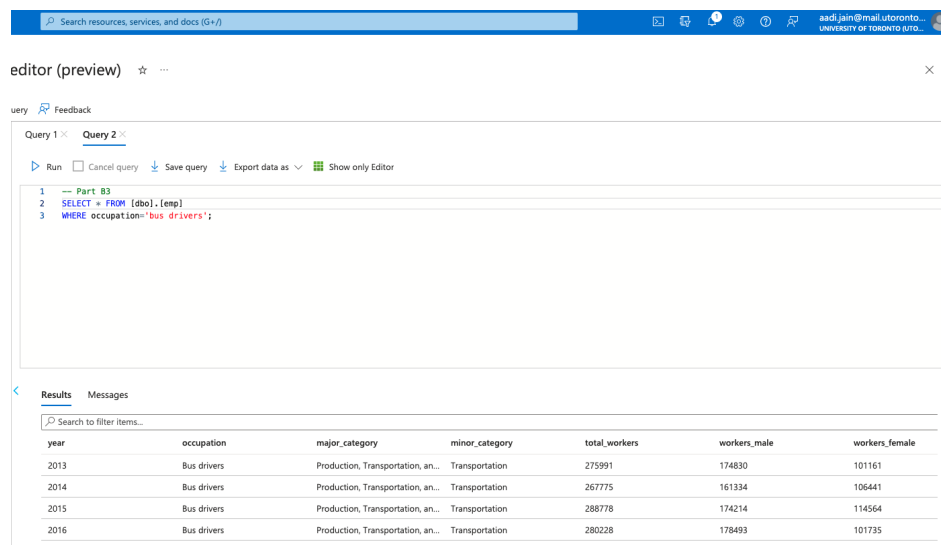


Figure 5: Result of Part B.2

Figure 6 shows the result for part B3.



editor (preview) ☆ ...

Query 1 × Query 2 ×

Run Cancel query Save query Export data as Show only Editor

```

1 -- Part B3
2 SELECT * FROM [dbo].[emp]
3 WHERE occupation='Bus drivers';

```

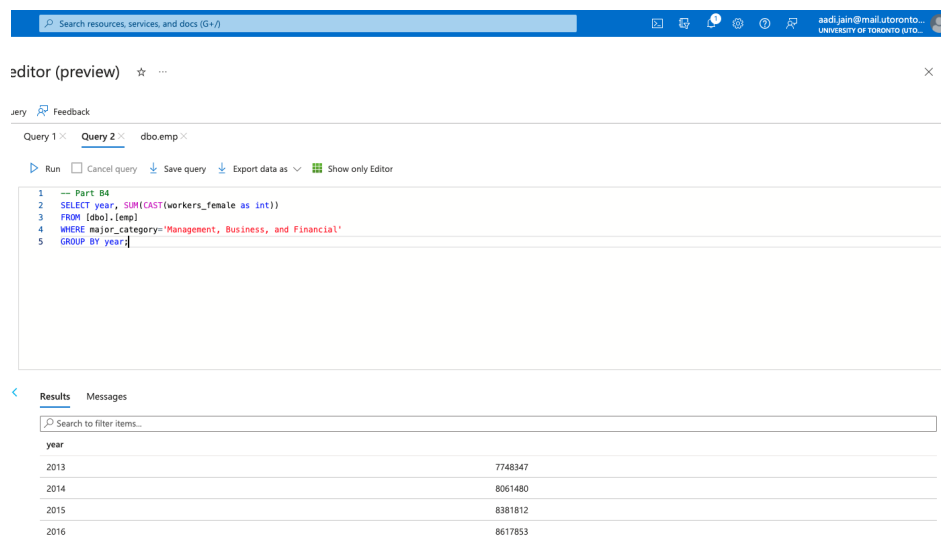
Results Messages

Search to filter items...

year	occupation	major_category	minor_category	total_workers	workers_male	workers_female
2013	Bus drivers	Production, Transportation, an...	Transportation	275991	174830	101161
2014	Bus drivers	Production, Transportation, an...	Transportation	267775	161334	106441
2015	Bus drivers	Production, Transportation, an...	Transportation	288778	174214	114564
2016	Bus drivers	Production, Transportation, an...	Transportation	280228	178493	101735

Figure 6: Result of Part B.3

Figure 7 shows the result for part B4.



editor (preview) ☆ ...

Query 1 × Query 2 × dbo.emp ×

Run Cancel query Save query Export data as Show only Editor

```

1 -- Part B4
2 SELECT year, SUM(CAST(workers_female as int))
3 FROM [dbo].[emp]
4 WHERE major_category='Management, Business, and Financial'
5 GROUP BY year;

```

Results Messages

Search to filter items...

year	
2013	7748347
2014	8061480
2015	8381812
2016	8617853

Figure 7: Result of Part B.4

Figure 8 shows the result for part B5.

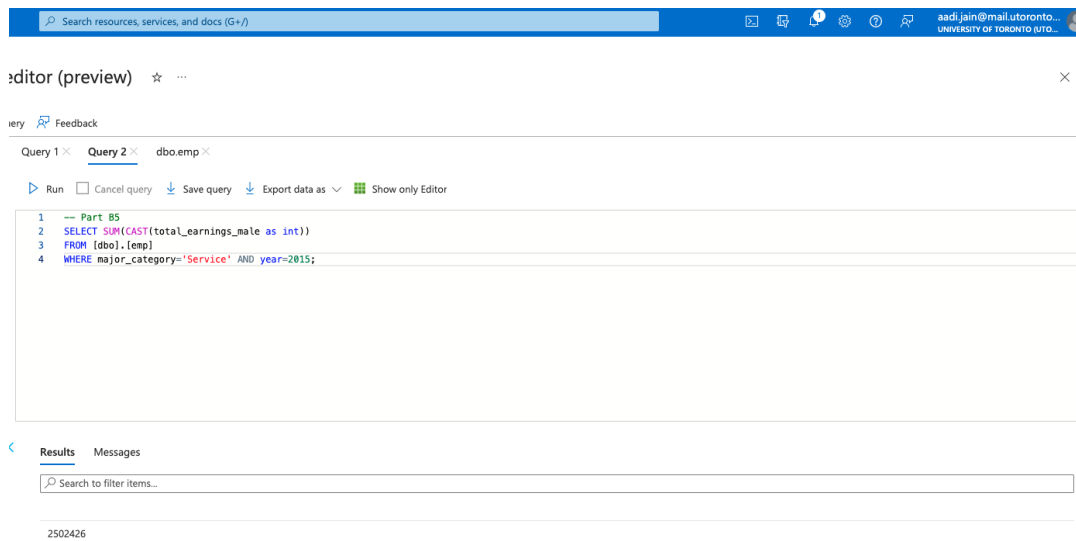


Figure 8: Result of Part B.5

Figure 9 shows the result for part B6.

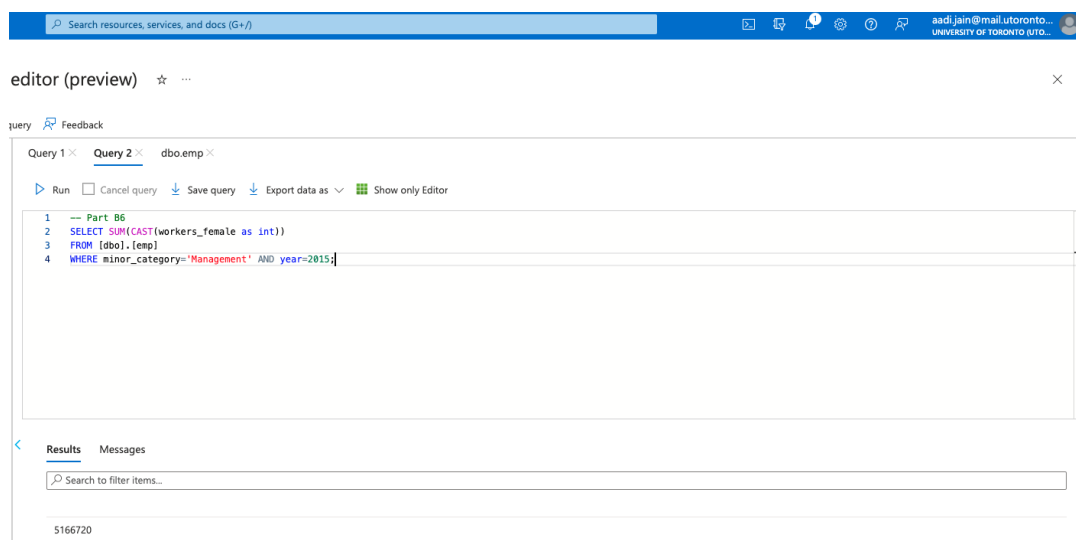
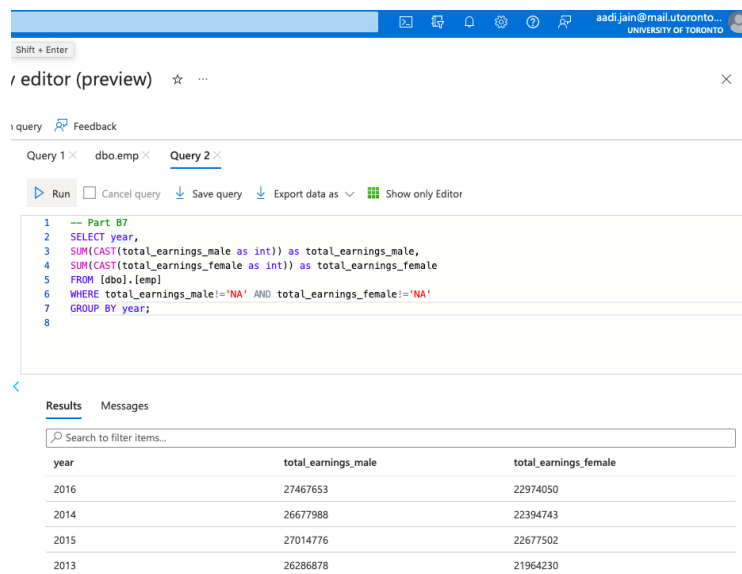


Figure 9: Result of Part B.6

Figure 10 shows the result for part B7.



Query editor (preview) ☆ ...

Query 1 × dbo.emp × Query 2 ×

Run Cancel query Save query Export data as Show only Editor

```

1 -- Part B7
2 SELECT year,
3 SUM(CAST(total_earnings_male as int)) as total_earnings_male,
4 SUM(CAST(total_earnings_female as int)) as total_earnings_female
5 FROM [dbo].[emp]
6 WHERE total_earnings_male != 'NA' AND total_earnings_female != 'NA'
7 GROUP BY year;
8

```

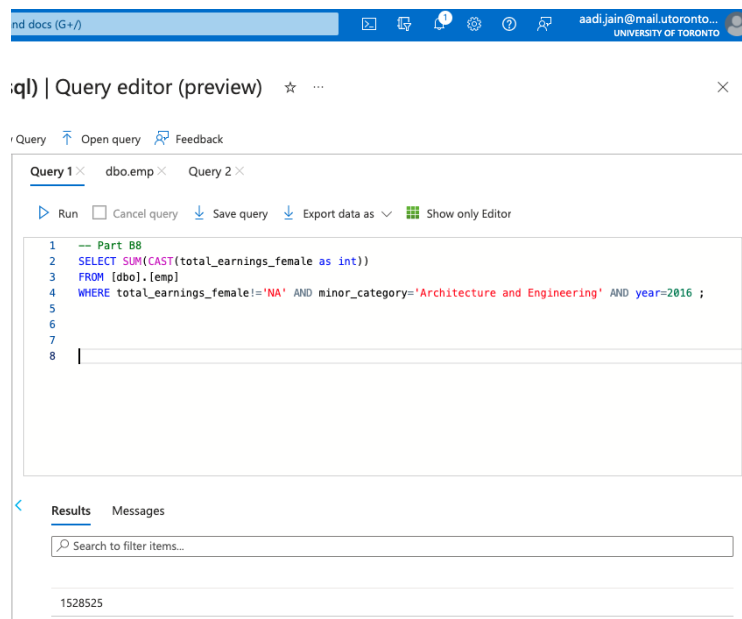
Results Messages

Search to filter items...

year	total_earnings_male	total_earnings_female
2016	27467653	22974050
2014	26677988	22394743
2015	27014776	22677502
2013	26286878	21964230

Figure 10: Result of Part B.7

Figure 12 shows the result for part B8.



nd docs (G+)

Query editor (preview) ☆ ...

Query Open query Feedback

Query 1 × dbo.emp × Query 2 ×

Run Cancel query Save query Export data as Show only Editor

```

1 -- Part B8
2 SELECT SUM(CAST(total_earnings_female as int))
3 FROM [dbo].[emp]
4 WHERE total_earnings_female != 'NA' AND minor_category='Architecture and Engineering' AND year=2016 ;
5
6
7
8

```

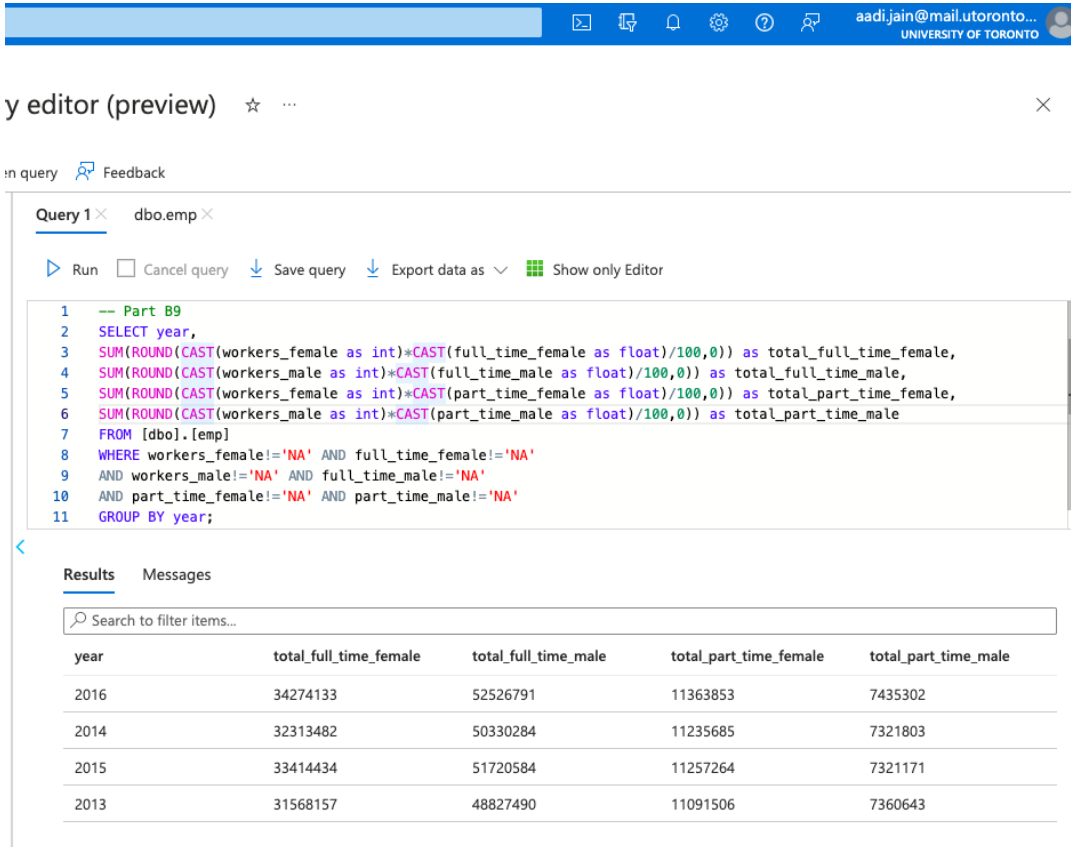
Results Messages

Search to filter items...

1528525

Figure 11: Result of Part B.8

Figure 12 shows the result for part B9.



The screenshot shows a SQL query editor interface. At the top, there's a blue header bar with user information: "aadi.jain@mail.utoronto... UNIVERSITY OF TORONTO". Below the header, the editor is titled "y editor (preview)". The query is named "Query 1" and is associated with the database "dbo.emp". The query text is as follows:

```
1  -- Part B9
2  SELECT year,
3  SUM(ROUND(CAST(workers_female as int)*CAST(full_time_female as float)/100,0)) as total_full_time_female,
4  SUM(ROUND(CAST(workers_male as int)*CAST(full_time_male as float)/100,0)) as total_full_time_male,
5  SUM(ROUND(CAST(workers_female as int)*CAST(part_time_female as float)/100,0)) as total_part_time_female,
6  SUM(ROUND(CAST(workers_male as int)*CAST(part_time_male as float)/100,0)) as total_part_time_male
7  FROM [dbo].[emp]
8  WHERE workers_female!='NA' AND full_time_female!='NA'
9  AND workers_male!='NA' AND full_time_male!='NA'
10 AND part_time_female!='NA' AND part_time_male!='NA'
11 GROUP BY year;
```

Below the query editor, there are tabs for "Results" and "Messages". The "Results" tab is active, showing a table with the following data:

year	total_full_time_female	total_full_time_male	total_part_time_female	total_part_time_male
2016	34274133	52526791	11363853	7435302
2014	32313482	50330284	11235685	7321803
2015	33414434	51720584	11257264	7321171
2013	31568157	48827490	11091506	7360643

Figure 12: Result of Part B.9