Individual Project :City of Toronto Data Store

Kartik Velede

8672807

Database Application– INFO8880

Douglas Hardie

February-13-2020

February-27-2020

**Table of Contents**

[**Basic Observations** 3](#_Toc33819316)

[**Advance Observations** 4](#_Toc33819317)

[**Key Performance Indicator** 5](#_Toc33819318)

[**Views** 5](#_Toc33819319)

[**Summary** 6](#_Toc33819320)

# **Basic Observations**

Firstly I have executed the two tables which are imported from the city of the Toronto data store and after that I have displayed the neighbourhood whose home price is less than 20000,displayed the columns from the table where household assisted is equal to 1 and used between operator to find the certain neighbourhoods, displayed the unique neighbourhood so that duplicate values get removed and displayed all the neighbourhood where name starts with a.

select \* from RawDataRef\_2011$

select \* from ['RawDataRef-Period2008$']

1st observations

Finding the neighbourhood where home price is less than 20000 from rawdataref\_2011$

select neighbourhood ,[Home Prices] prices from Rawdataref\_2011$ where [Home Prices] >20000

2nd observations

Select Neighbourhood,[Neighbourhood Id],[Households Assisted] from ['RawDataRef-Period2008$']

where [Households Assisted] =1

3rd Observation

select [Neighbourhood Id],Neighbourhood

from RawDataRef\_2011$

where [Neighbourhood Id] between 1 and 20

order by [Neighbourhood Id] desc

--4th observation

Select distinct Neighbourhood from RawDataRef\_2011$

--5th observation

Select Neighbourhood from ['RawDataRef-Period2008$']

where Neighbourhood like 'a%'

# **Advance Observations**

In advance observations I have used inner joins ,right joins to join the two tables by using the identical column in both tables and the subquery with where clause.

1st advance observation

select RawDataRef\_2011$.Neighbourhood, RawDataRef\_2011$.[Home Prices],['RawDataRef-Period2008$'].[Rent Bank Applicants]

from RawDataRef\_2011$ inner join ['RawDataRef-Period2008$'] on RawDataRef\_2011$.[Neighbourhood Id]=['RawDataRef-Period2008$'].[Neighbourhood Id]

2nd advance Observation

select \* from RawDataRef\_2011$

where [Neighbourhood Id] in

( select [Rent Bank Applicants] from ['RawDataRef-Period2008$']

where [Rent Bank Applicants]>=20)

3rd advance Observation

select RawDataRef\_2011$.Neighbourhood, RawDataRef\_2011$.[Home Prices],['RawDataRef-Period2008$'].[Rent Bank Applicants]

from RawDataRef\_2011$ right join ['RawDataRef-Period2008$'] on RawDataRef\_2011$.[Neighbourhood Id]=['RawDataRef-Period2008$'].[Neighbourhood Id]

# **Key Performance Indicator**

For the key performance indicator observations, I have used different aggregate operators on column of the table . Evaluated sum of applications received for the house rent, average housing turnover during the period 2008 and the count of households present in the year 2011.

1St KPI Observation

select [Households Assisted], sum([Rent Bank Applicants]) as Totalrentbank

from ['RawDataRef-Period2008$']

group by [Households Assisted]

order by [Households Assisted]

2nd KPI Observation

select Avg( [Social Housing Turnover]) as Avgscoialhouisngturnover

from ['RawDataRef-Period2008$']

where [Social Housing Turnover] is not null

3rd KPI Observation

Select count( [Mid-Century Highrise Households]) as totalpopulation

from RawDataRef\_2011$

# **Views**

Creating views for the above KPI’S. Views doesn’t take storage space in database.

1st view

Create view totalrentbank as

select [Households Assisted], sum([Rent Bank Applicants]) as Totalrentbank

from ['RawDataRef-Period2008$']

group by [Households Assisted]

select \* from totalrentbank

2nd View

Create view Avgsocialhousingturnover as

select Avg( [Social Housing Turnover]) as Avgscoialhouisngturnover

from ['RawDataRef-Period2008$']

where [Social Housing Turnover] is not null

select \* from avgsocialhousingturnover

3rd View

Create view totalpopulation as

Select count( [Mid-Century Highrise Households]) as totalpopulation from RawDataRef\_2011$

select \* from totalpopulation

# **Summary**

Overall, I have performed views ,joins, key performance indicators and some basic observations like group by, order by with ascending and descending order along with aggregate operators such as sum, count, average.