Assignment 1

B. Data Analytic Stage

Table of Contents

3. Create the following reports using OLAP queries. 2

a. Simple reports: 2

REPORT 1 (Top n report) 2

REPORT 2 (Top n% report) 3

b. Reports with proper sub-totals: 4

REPORT 3 (Cube) 4

REPORT 4 (Partial Cube) 5

REPORT 5 (Roll-up) 6

REPORT 6(Partial Roll-up) 7

c. Reports with moving and cumulative aggregates: 8

REPORT 7: 8

REPORT 8: 10

d. Reports with Partitions: 11

REPORT 9: 11

4. Business Intelligence (BI) Reports. 12

3. Create the following reports using OLAP queries.

a. Simple reports:

Produce two reports. Each report contains two attributes from two different dimensions, and one fact measurement.

For the report itself, the first report must be about Top n and the second report is Top n % .

REPORT 1 (Top n report)

(a) The query questions

Find out the top 10 equipment categories by total sales in 2020.

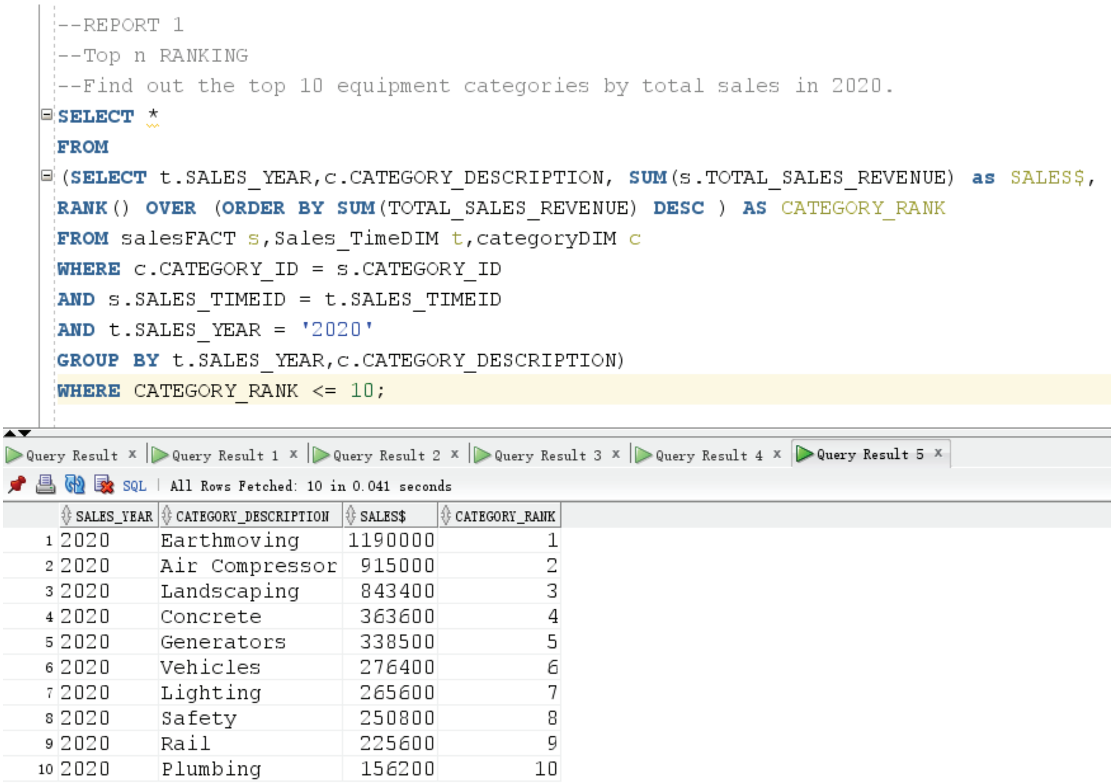
(b) Your explanation

Managers will be interested in the sales data of the past years, especially the most recent year, and listing the best-selling categories can help managers make decisions on how to place orders in the coming years.

(c) The SQL commands

```
SELECT *

FROM

(SELECT t.SALES_YEAR,c.CATEGORY_DESCRIPTION, SUM(s.TOTAL_SALES_REVENUE) as
SALES$,

RANK() OVER (ORDER BY SUM(TOTAL_SALES_REVENUE) DESC ) AS CATEGORY_RANK

FROM salesFACT s,Sales_TimeDIM t,categoryDIM c

WHERE c.CATEGORY_ID = s.CATEGORY_ID

AND s.SALES_TIMEID = t.SALES_TIMEID

AND t.SALES_YEAR = '2020'

GROUP BY t.SALES_YEAR,c.CATEGORY_DESCRIPTION)

WHERE CATEGORY_RANK <= 10;
```

(d) The screenshots of the query results



```
--REPORT 1
--Top n RANKING
--Find out the top 10 equipment categories by total sales in 2020.
SELECT *
FROM
(SELECT t.SALES_YEAR,c.CATEGORY_DESCRIPTION, SUM(s.TOTAL_SALES_REVENUE) as SALES$,
RANK() OVER (ORDER BY SUM(TOTAL_SALES_REVENUE) DESC ) AS CATEGORY_RANK
FROM salesFACT s,Sales_TimeDIM t,categoryDIM c
WHERE c.CATEGORY_ID = s.CATEGORY_ID
AND s.SALES_TIMEID = t.SALES_TIMEID
AND t.SALES_YEAR = '2020'
GROUP BY t.SALES_YEAR,c.CATEGORY_DESCRIPTION)
WHERE CATEGORY_RANK <= 10;
```

All Rows Fetched: 10 in 0.041 seconds

	SALES_YEAR	CATEGORY_DESCRIPTION	SALES$	CATEGORY_RANK
1	2020	Earthmoving	1190000	1
2	2020	Air Compressor	915000	2
3	2020	Landscaping	843400	3
4	2020	Concrete	363600	4
5	2020	Generators	338500	5
6	2020	Vehicles	276400	6
7	2020	Lighting	265600	7
8	2020	Safety	250800	8
9	2020	Rail	225600	9
10	2020	Plumbing	156200	10

REPORT 2 (Top n% report)

(a) The query questions

Find out the top 30% company branches by total sales in 2020.

(b) Your explanation

As a manager, he or she needs to know the branches with the top sales rankings. The company can reward these branches. At the same time, it can also allow these branches with good sales performance to pass on the experience to other branches with poor performance.

(c) The SQL commands

SELECT *

FROM

(SELECT t.SALES_YEAR,c.COMPANY_BRANCH, SUM(s.TOTAL_SALES_REVENUE) as SALES$,

percent_rank() OVER (ORDER BY SUM(TOTAL_SALES_REVENUE) DESC) AS COMPANY_BRANCH_RANK

FROM salesFACT s,Sales_TimeDIM t,Company_BranchDIM c

WHERE c.COMPANY_BRANCH = s.COMPANY_BRANCH

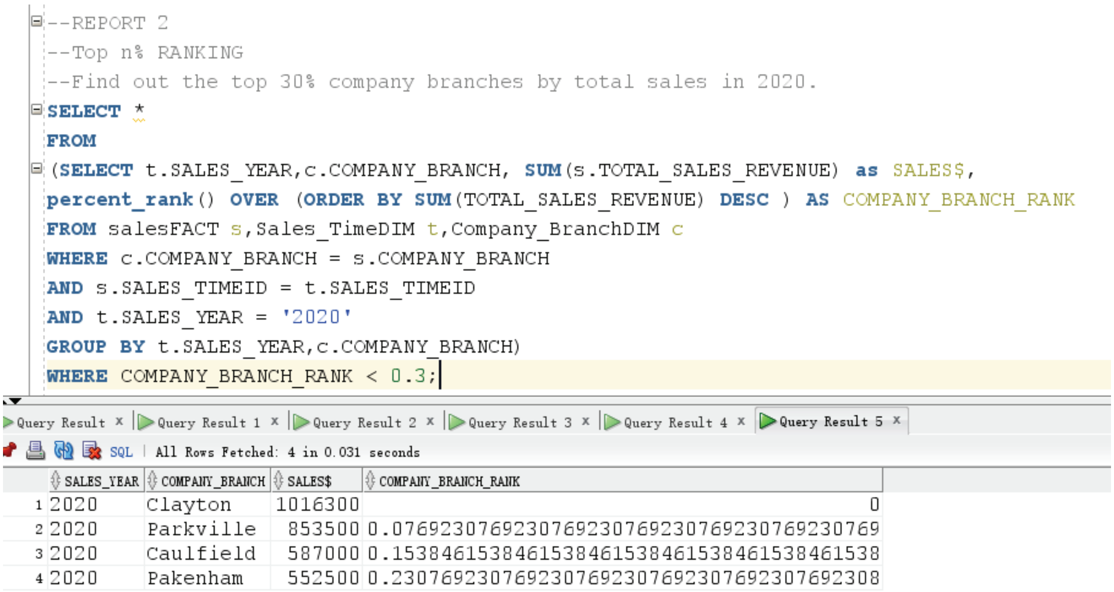
AND s.SALES_TIMEID = t.SALES_TIMEID

AND t.SALES_YEAR = '2020'

GROUP BY t.SALES_YEAR,c.COMPANY_BRANCH)

WHERE COMPANY_BRANCH_RANK < 0.3;

(d) The screenshots of the query results



```
--REPORT 2
--Top n% RANKING
--Find out the top 30% company branches by total sales in 2020.
SELECT *
FROM
(SELECT t.SALES_YEAR,c.COMPANY_BRANCH, SUM(s.TOTAL_SALES_REVENUE) as SALES$,
percent_rank() OVER (ORDER BY SUM(TOTAL_SALES_REVENUE) DESC ) AS COMPANY_BRANCH_RANK
FROM salesFACT s,Sales_TimeDIM t,Company_BranchDIM c
WHERE c.COMPANY_BRANCH = s.COMPANY_BRANCH
AND s.SALES_TIMEID = t.SALES_TIMEID
AND t.SALES_YEAR = '2020'
GROUP BY t.SALES_YEAR,c.COMPANY_BRANCH)
WHERE COMPANY_BRANCH_RANK < 0.3;
```

All Rows Fetched: 4 in 0.031 seconds

	SALES_YEAR	COMPANY_BRANCH	SALES$	COMPANY_BRANCH_RANK
1	2020	Clayton	1016300	0
2	2020	Parkville	853500	0.0769230769230769230769230769230769230769
3	2020	Caulfield	587000	0.1538461538461538461538461538461538461538
4	2020	Pakenham	552500	0.2307692307692307692307692307692307692308

b. Reports with proper sub-totals:

Produce four reports. These reports must include subtotals, using the Cube or Roll-up or Partial Cube/Roll-up operators

REPORT 3 (Cube)

select cb.COMPANY_BRANCH,ct.DESCRIPTION as customer_type,st.SALES_TIMEID as sales_time,

SUM(sf.TOTAL_SALES_REVENUE) as SALES$

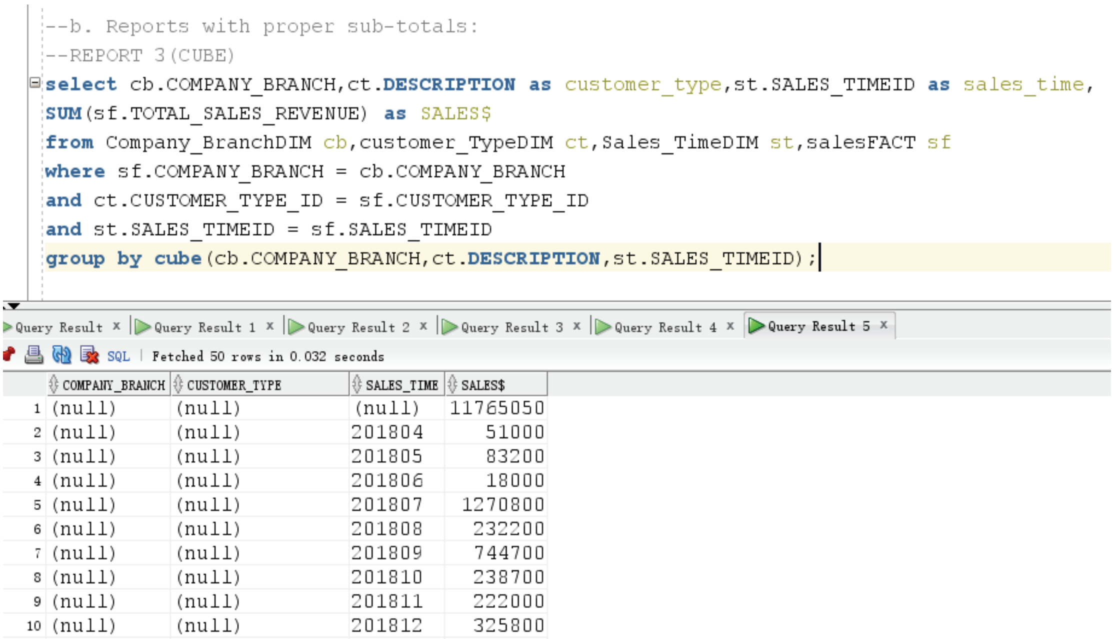
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf

where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH

and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID

and st.SALES_TIMEID = sf.SALES_TIMEID

group by cube(cb.COMPANY_BRANCH,ct.DESCRIPTION,st.SALES_TIMEID);



```
--b. Reports with proper sub-totals:
--REPORT 3(CUBE)
select cb.COMPANY_BRANCH,ct.DESCRIPTION as customer_type,st.SALES_TIMEID as sales_time,
SUM(sf.TOTAL_SALES_REVENUE) as SALES$
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf
where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH
and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID
and st.SALES_TIMEID = sf.SALES_TIMEID
group by cube(cb.COMPANY_BRANCH,ct.DESCRIPTION,st.SALES_TIMEID);
```

Fetched 50 rows in 0.032 seconds

	COMPANY_BRANCH	CUSTOMER_TYPE	SALES_TIME	SALES$
1	(null)	(null)	(null)	11765050
2	(null)	(null)	201804	51000
3	(null)	(null)	201805	83200
4	(null)	(null)	201806	18000
5	(null)	(null)	201807	1270800
6	(null)	(null)	201808	232200
7	(null)	(null)	201809	744700
8	(null)	(null)	201810	238700
9	(null)	(null)	201811	222000
10	(null)	(null)	201812	325800

REPORT 4 (Partial Cube)

select ct.DESCRIPTION as customer_type,cb.COMPANY_BRANCH,st.SALES_TIMEID as sales_time,

SUM(sf.TOTAL_SALES_REVENUE) as SALES$

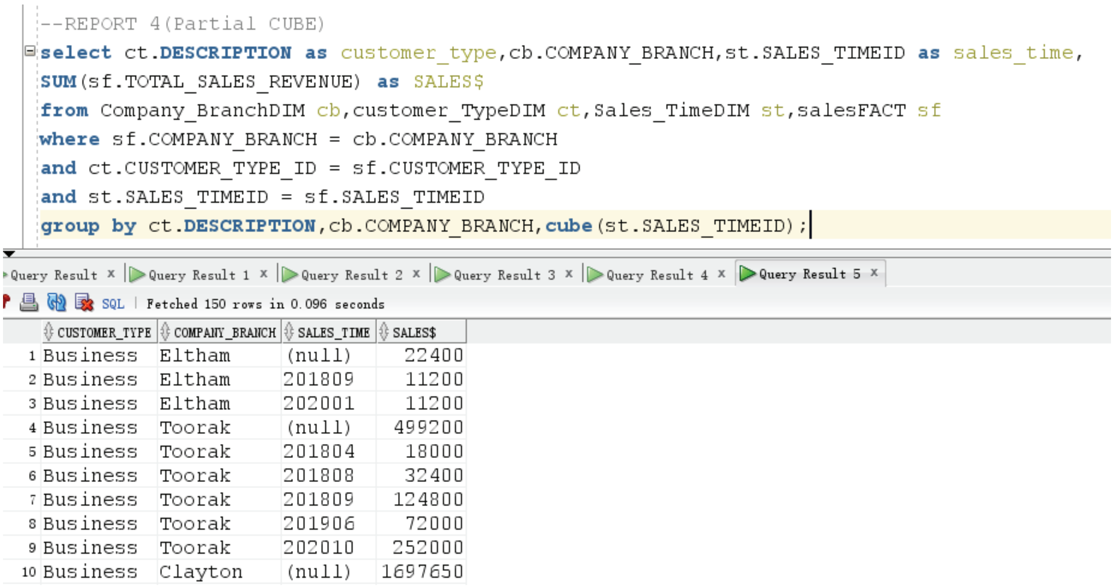
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf

where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH

and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID

and st.SALES_TIMEID = sf.SALES_TIMEID

group by ct.DESCRIPTION,cb.COMPANY_BRANCH,cube(st.SALES_TIMEID);



```
--REPORT 4(Partial CUBE)
select ct.DESCRIPTION as customer_type,cb.COMPANY_BRANCH,st.SALES_TIMEID as sales_time,
SUM(sf.TOTAL_SALES_REVENUE) as SALES$
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf
where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH
and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID
and st.SALES_TIMEID = sf.SALES_TIMEID
group by ct.DESCRIPTION,cb.COMPANY_BRANCH,cube(st.SALES_TIMEID);
```

Fetched 150 rows in 0.096 seconds

	CUSTOMER_TYPE	COMPANY_BRANCH	SALES_TIME	SALES$
1	Business	Eltham	(null)	22400
2	Business	Eltham	201809	11200
3	Business	Eltham	202001	11200
4	Business	Toorak	(null)	499200
5	Business	Toorak	201804	18000
6	Business	Toorak	201808	32400
7	Business	Toorak	201809	124800
8	Business	Toorak	201906	72000
9	Business	Toorak	202010	252000
10	Business	Clayton	(null)	1697650

REPORT 5 (Roll-up)

(a) The query questions

Find out the total sales of the business and individual customer type in each company branch, and from the year 2018 to 2020.

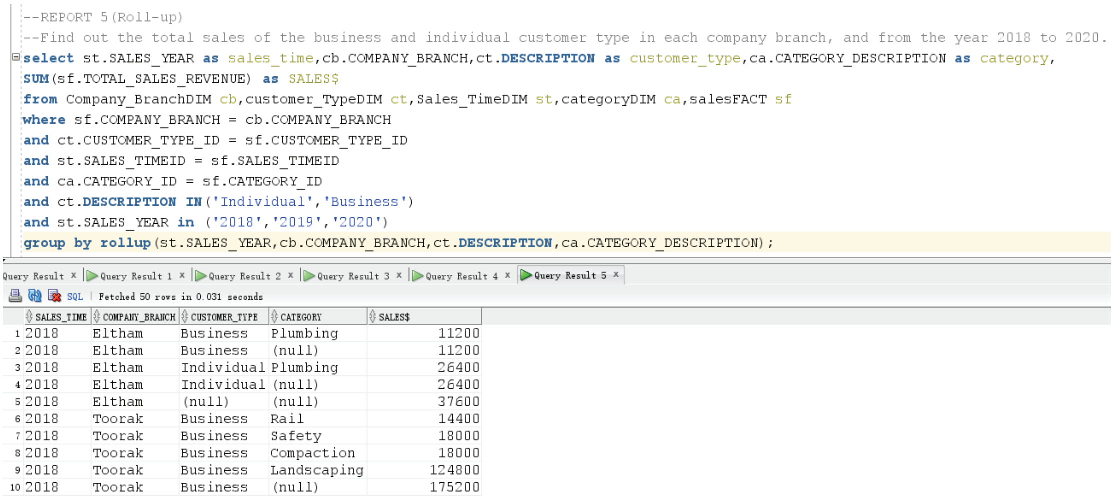
(b) Your explanation

This query question is more comprehensive. First of all, it covers the sales data for the last three years regarding the time dimension. Second, it covers all branches. Third, it covers both business and individual customer types. Last, it also contains all the equipment categories. In this way, the management can see the data information of these dimensions more comprehensively, helping them make better business decisions.

(c) The SQL commands

```
select st.SALES_YEAR as sales_time,cb.COMPANY_BRANCH,ct.DESCRIPTION as
customer_type,ca.CATEGORY_DESCRIPTION as category,

SUM(sf.TOTAL_SALES_REVENUE) as SALES$

from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,categoryDIM
ca,salesFACT sf

where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH

and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID

and st.SALES_TIMEID = sf.SALES_TIMEID

and ca.CATEGORY_ID = sf.CATEGORY_ID

and ct.DESCRIPTION IN('Individual','Business')

and st.SALES_YEAR in ('2018','2019','2020')

group by
rollup(st.SALES_YEAR,cb.COMPANY_BRANCH,ct.DESCRIPTION,ca.CATEGORY_DESCRIPTION);
```

(d) The screenshots of the query results



```
--REPORT 5(Roll-up)
--Find out the total sales of the business and individual customer type in each company branch, and from the year 2018 to 2020.
select st.SALES_YEAR as sales_time,cb.COMPANY_BRANCH,ct.DESCRIPTION as customer_type,ca.CATEGORY_DESCRIPTION as category,
SUM(sf.TOTAL_SALES_REVENUE) as SALES$
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,categoryDIM ca,salesFACT sf
where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH
and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID
and st.SALES_TIMEID = sf.SALES_TIMEID
and ca.CATEGORY_ID = sf.CATEGORY_ID
and ct.DESCRIPTION IN('Individual','Business')
and st.SALES_YEAR in ('2018','2019','2020')
group by rollup(st.SALES_YEAR,cb.COMPANY_BRANCH,ct.DESCRIPTION,ca.CATEGORY_DESCRIPTION);
```

Fetched 50 rows in 0.031 seconds

```
   SALES_TIME COMPANY_BRANCH CUSTOMER_TYPE CATEGORY     SALES$
 1 2018       Eltham         Business      Plumbing      11200
 2 2018       Eltham         Business      (null)        11200
 3 2018       Eltham         Individual    Plumbing      26400
 4 2018       Eltham         Individual    (null)        26400
 5 2018       Eltham         (null)        (null)        37600
 6 2018       Toorak         Business      Rail          14400
 7 2018       Toorak         Business      Safety        18000
 8 2018       Toorak         Business      Compaction    18000
 9 2018       Toorak         Business      Landscaping  124800
10 2018       Toorak         Business      (null)       175200
```

REPORT 6(Partial Roll-up)

(a) The query questions

Find out the total sales of the individual and business category in each company branch of 2020.

(b) Your explanation

This query will narrow the scope and focus the data analysis on the most recent year 2020. At the same time, only the customer type and all branches are selected, so that the management can understand each branch's performance in the past year.

(c) The SQL commands

select st.SALES_YEAR as sales_time,cb.COMPANY_BRANCH,ct.DESCRIPTION as customer_type,

SUM(sf.TOTAL_SALES_REVENUE) as SALES$

from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf

where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH

and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID

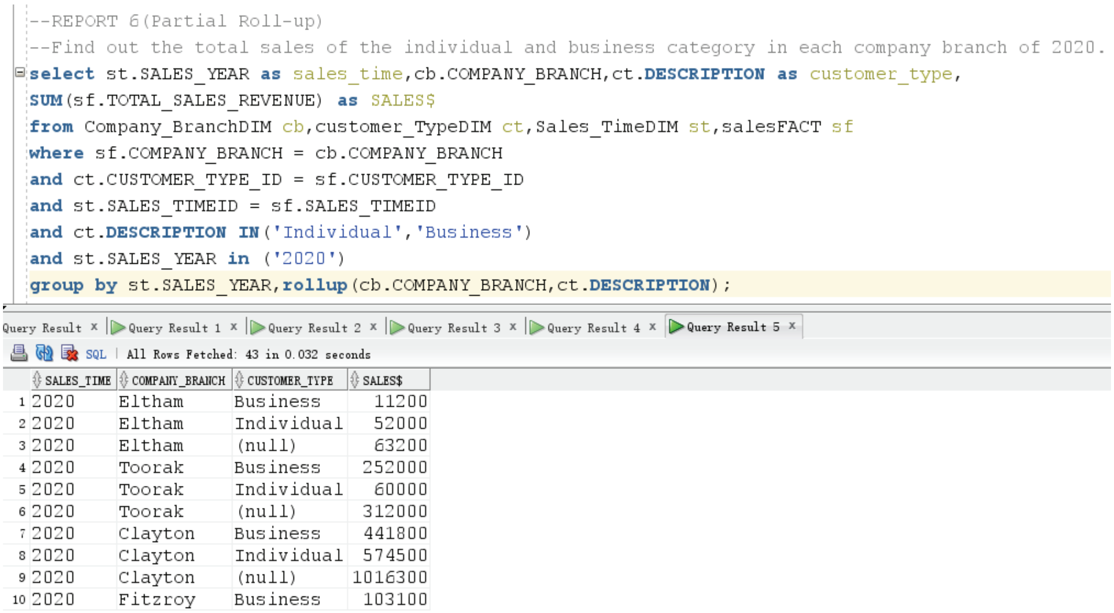
and st.SALES_TIMEID = sf.SALES_TIMEID

and ct.DESCRIPTION IN('Individual','Business')

and st.SALES_YEAR in ('2020')

group by st.SALES_YEAR,rollup(cb.COMPANY_BRANCH,ct.DESCRIPTION);

(d) The screenshots of the query results



```
--REPORT 6(Partial Roll-up)
--Find out the total sales of the individual and business category in each company branch of 2020.
select st.SALES_YEAR as sales_time,cb.COMPANY_BRANCH,ct.DESCRIPTION as customer_type,
SUM(sf.TOTAL_SALES_REVENUE) as SALES$
from Company_BranchDIM cb,customer_TypeDIM ct,Sales_TimeDIM st,salesFACT sf
where sf.COMPANY_BRANCH = cb.COMPANY_BRANCH
and ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID
and st.SALES_TIMEID = sf.SALES_TIMEID
and ct.DESCRIPTION IN('Individual','Business')
and st.SALES_YEAR in ('2020')
group by st.SALES_YEAR,rollup(cb.COMPANY_BRANCH,ct.DESCRIPTION);
```

All Rows Fetched: 43 in 0.032 seconds

```
   SALES_TIME COMPANY_BRANCH CUSTOMER_TYPE SALES$
 1 2020       Eltham         Business        11200
 2 2020       Eltham         Individual      52000
 3 2020       Eltham         (null)          63200
 4 2020       Toorak         Business       252000
 5 2020       Toorak         Individual      60000
 6 2020       Toorak         (null)         312000
 7 2020       Clayton        Business       441800
 8 2020       Clayton        Individual     574500
 9 2020       Clayton        (null)        1016300
10 2020       Fitzroy        Business       103100
```

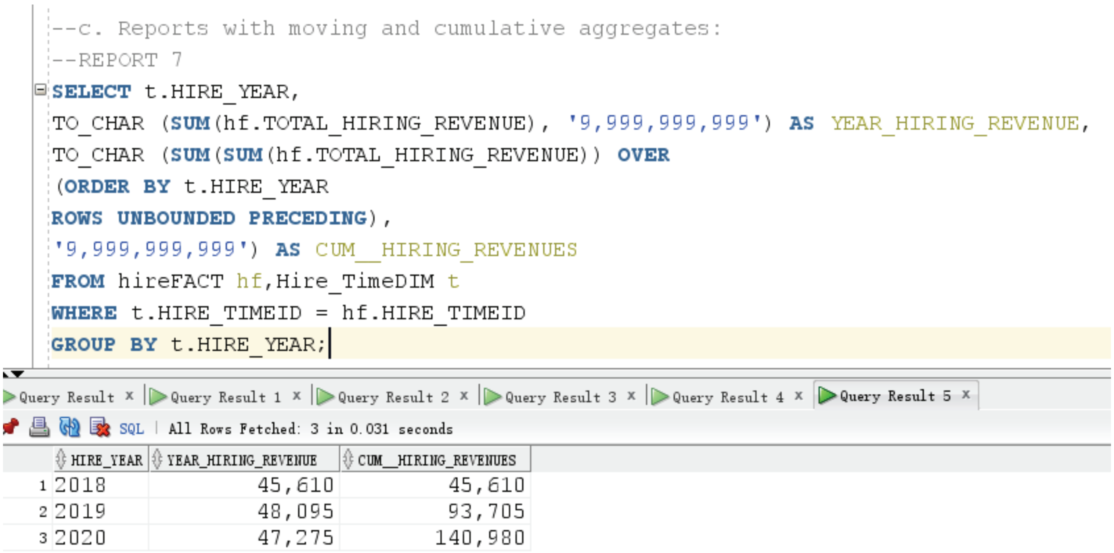
c. Reports with moving and cumulative aggregates:

REPORT 7:

What are the total revenue for hiring equipment and cumulative total hiring revenue for Site Equipment in each year?

Total revenue for hiring equipment

```
SELECT t.HIRE_YEAR,

TO_CHAR (SUM(hf.TOTAL_HIRING_REVENUE), '9,999,999,999') AS YEAR_HIRING_REVENUE,

TO_CHAR (SUM(SUM(hf.TOTAL_HIRING_REVENUE)) OVER

(ORDER BY t.HIRE_YEAR

ROWS UNBOUNDED PRECEDING),

'9,999,999,999') AS CUM__HIRING_REVENUES

FROM hireFACT hf,Hire_TimeDIM t

WHERE t.HIRE_TIMEID = hf.HIRE_TIMEID

GROUP BY t.HIRE_YEAR;
```



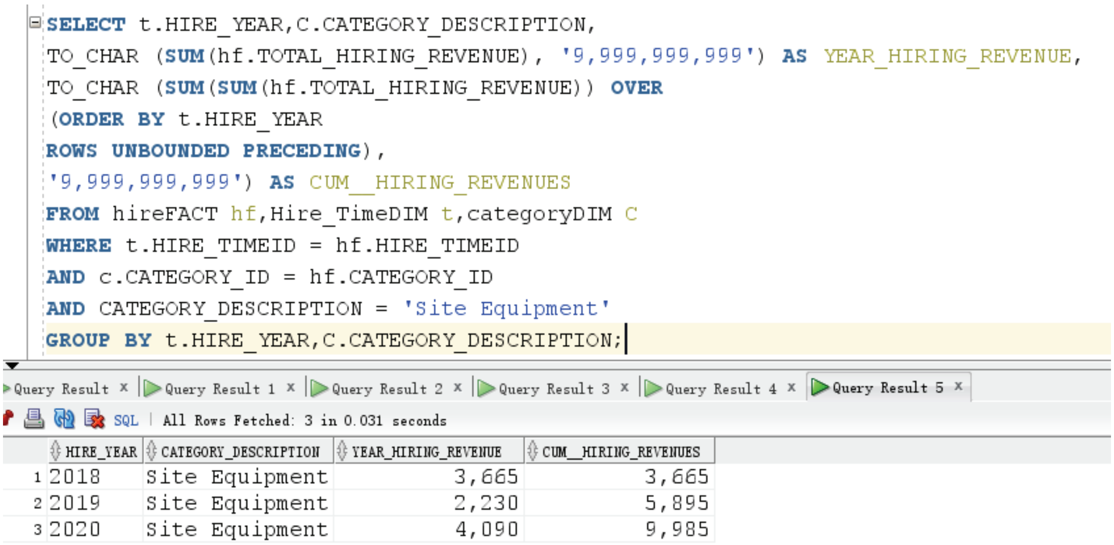
```
--c. Reports with moving and cumulative aggregates:
--REPORT 7
SELECT t.HIRE_YEAR,
TO_CHAR (SUM(hf.TOTAL_HIRING_REVENUE), '9,999,999,999') AS YEAR_HIRING_REVENUE,
TO_CHAR (SUM(SUM(hf.TOTAL_HIRING_REVENUE)) OVER
(ORDER BY t.HIRE_YEAR
ROWS UNBOUNDED PRECEDING),
'9,999,999,999') AS CUM__HIRING_REVENUES
FROM hireFACT hf,Hire_TimeDIM t
WHERE t.HIRE_TIMEID = hf.HIRE_TIMEID
GROUP BY t.HIRE_YEAR;
```

All Rows Fetched: 3 in 0.031 seconds

	HIRE_YEAR	YEAR_HIRING_REVENUE	CUM__HIRING_REVENUES
1	2018	45,610	45,610
2	2019	48,095	93,705
3	2020	47,275	140,980

Cumulative total hiring revenue for Site Equipment in each year

```
SELECT t.HIRE_YEAR,C.CATEGORY_DESCRIPTION,

TO_CHAR (SUM(hf.TOTAL_HIRING_REVENUE), '9,999,999,999') AS YEAR_HIRING_REVENUE,

TO_CHAR (SUM(SUM(hf.TOTAL_HIRING_REVENUE)) OVER

(ORDER BY t.HIRE_YEAR

ROWS UNBOUNDED PRECEDING),

'9,999,999,999') AS CUM__HIRING_REVENUES

FROM hireFACT hf,Hire_TimeDIM t,categoryDIM C

WHERE t.HIRE_TIMEID = hf.HIRE_TIMEID

AND c.CATEGORY_ID = hf.CATEGORY_ID

AND CATEGORY_DESCRIPTION = 'Site Equipment'

GROUP BY t.HIRE_YEAR,C.CATEGORY_DESCRIPTION;
```



```
SELECT t.HIRE_YEAR,C.CATEGORY_DESCRIPTION,
TO_CHAR (SUM(hf.TOTAL_HIRING_REVENUE), '9,999,999,999') AS YEAR_HIRING_REVENUE,
TO_CHAR (SUM(SUM(hf.TOTAL_HIRING_REVENUE)) OVER
(ORDER BY t.HIRE_YEAR
ROWS UNBOUNDED PRECEDING),
'9,999,999,999') AS CUM__HIRING_REVENUES
FROM hireFACT hf,Hire_TimeDIM t,categoryDIM C
WHERE t.HIRE_TIMEID = hf.HIRE_TIMEID
AND c.CATEGORY_ID = hf.CATEGORY_ID
AND CATEGORY_DESCRIPTION = 'Site Equipment'
GROUP BY t.HIRE_YEAR,C.CATEGORY_DESCRIPTION;
```

All Rows Fetched: 3 in 0.031 seconds

	HIRE_YEAR	CATEGORY_DESCRIPTION	YEAR_HIRING_REVENUE	CUM__HIRING_REVENUES
1	2018	Site Equipment	3,665	3,665
2	2019	Site Equipment	2,230	5,895
3	2020	Site Equipment	4,090	9,985

REPORT 8:

Produce one other moving/cumulative aggregate report that is useful for management.

The outputs of this task are:

(a) The query questions

For individual and business customer types, find out the three-month moving average of total sales revenue.

(b) Your explanation

Managers will be interested in the sales revenue of different types of customers. So I choose to analyse the three-month moving average of the two customer types. Therefore, Managers can easily see the growth of sales of different kind of customers.

(c) The SQL commands

```
SELECT ct.DESCRIPTION as customer_type,s.SALES_TIMEID,

TO_CHAR (SUM(sf.TOTAL_SALES_REVENUE), '9,999,999,999') AS Month_SALES_REVENUE,

TO_CHAR (AVG(SUM(sf.TOTAL_SALES_REVENUE)) OVER

(ORDER BY s.SALES_TIMEID

ROWS 2 PRECEDING),

'9,999,999,999') AS MOVING_3_MONTH_AVG

FROM salesFACT sf,Sales_TimeDIM s,customer_TypeDIM ct
```

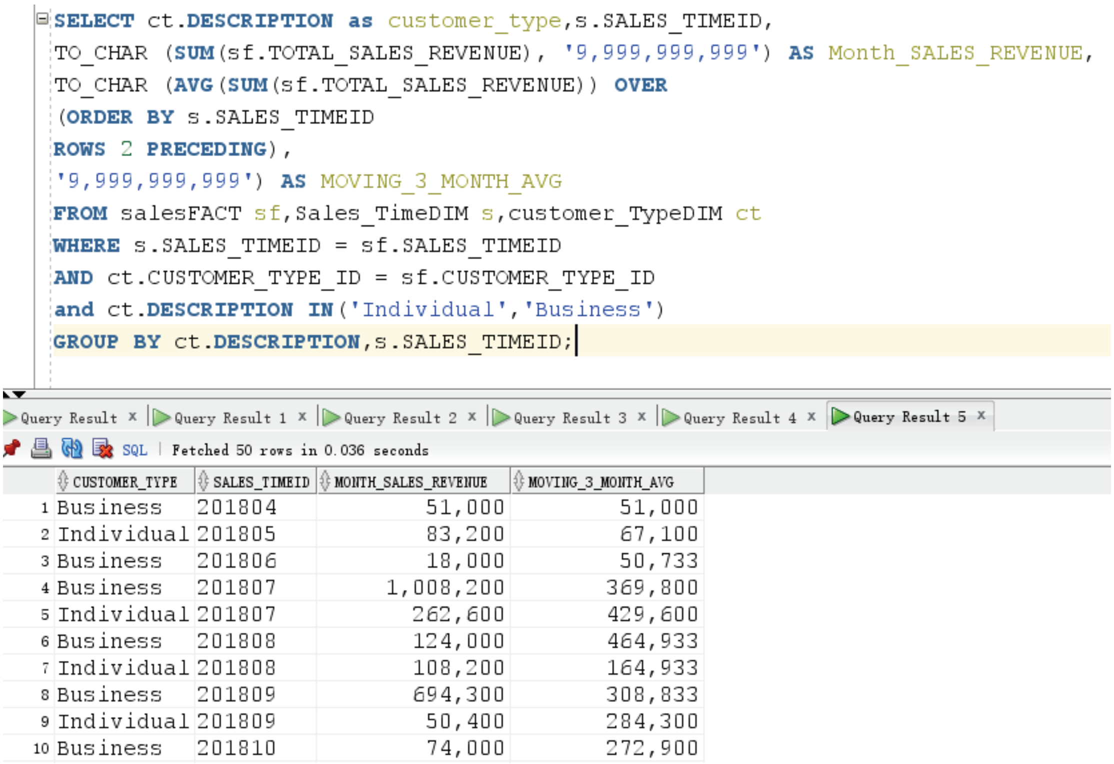
WHERE s.SALES_TIMEID = sf.SALES_TIMEID

AND ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID

and ct.DESCRIPTION IN('Individual','Business')

GROUP BY ct.DESCRIPTION,s.SALES_TIMEID;

(d) The screenshots of the query results



```
SELECT ct.DESCRIPTION as customer_type,s.SALES_TIMEID,
TO_CHAR (SUM(sf.TOTAL_SALES_REVENUE), '9,999,999,999') AS Month_SALES_REVENUE,
TO_CHAR (AVG(SUM(sf.TOTAL_SALES_REVENUE)) OVER
(ORDER BY s.SALES_TIMEID
ROWS 2 PRECEDING),
'9,999,999,999') AS MOVING_3_MONTH_AVG
FROM salesFACT sf,Sales_TimeDIM s,customer_TypeDIM ct
WHERE s.SALES_TIMEID = sf.SALES_TIMEID
AND ct.CUSTOMER_TYPE_ID = sf.CUSTOMER_TYPE_ID
and ct.DESCRIPTION IN('Individual','Business')
GROUP BY ct.DESCRIPTION,s.SALES_TIMEID;
```

Fetched 50 rows in 0.036 seconds

	CUSTOMER_TYPE	SALES_TIMEID	MONTH_SALES_REVENUE	MOVING_3_MONTH_AVG
1	Business	201804	51,000	51,000
2	Individual	201805	83,200	67,100
3	Business	201806	18,000	50,733
4	Business	201807	1,008,200	369,800
5	Individual	201807	262,600	429,600
6	Business	201808	124,000	464,933
7	Individual	201808	108,200	164,933
8	Business	201809	694,300	308,833
9	Individual	201809	50,400	284,300
10	Business	201810	74,000	272,900

d. Reports with Partitions:

REPORT 9:

Show ranking of each equipment category based on the monthly total number of equipment hired and the ranking of each branch based on the monthly total number of equipment hired.

(a) The SQL commands that contain partitions

SELECT cb.COMPANY_BRANCH, ht.HIRE_TIMEID AS month,

TO_CHAR(SUM(hf.NUM_EQUIPMENT_HIRED)) AS number_of_equipment_hired,

RANK() OVER (PARTITION BY cb.COMPANY_BRANCH

ORDER BY SUM(hf.NUM_EQUIPMENT_HIRED) DESC) AS RANK_BY_COMPANY_BRANCH,

RANK() OVER (PARTITION BY ht.HIRE_TIMEID

ORDER BY SUM(hf.NUM_EQUIPMENT_HIRED) DESC) AS RANK_BY_MONTH

FROM Hire_TimeDIM ht,Company_BranchDIM cb,hireFACT hf

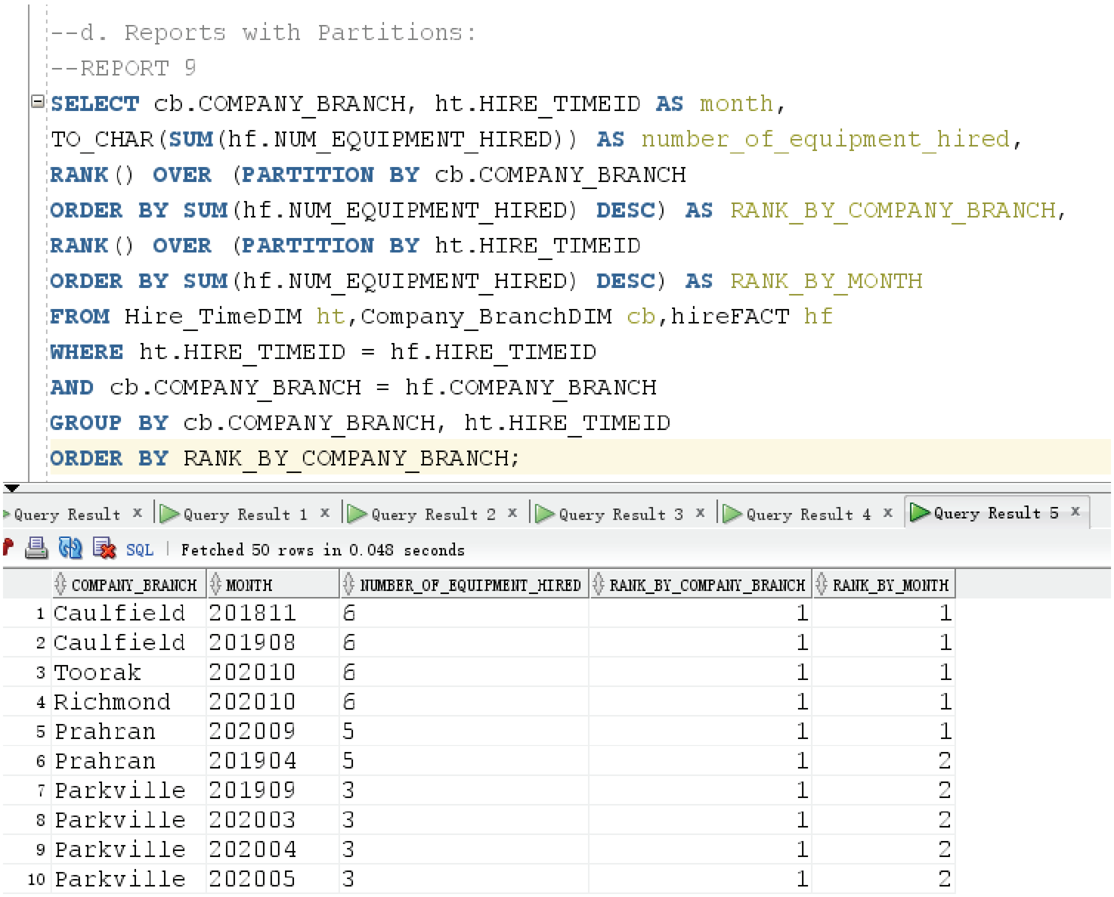
WHERE ht.HIRE_TIMEID = hf.HIRE_TIMEID

AND cb.COMPANY_BRANCH = hf.COMPANY_BRANCH

GROUP BY cb.COMPANY_BRANCH, ht.HIRE_TIMEID

ORDER BY RANK_BY_COMPANY_BRANCH;

(b) The screenshots of the query results



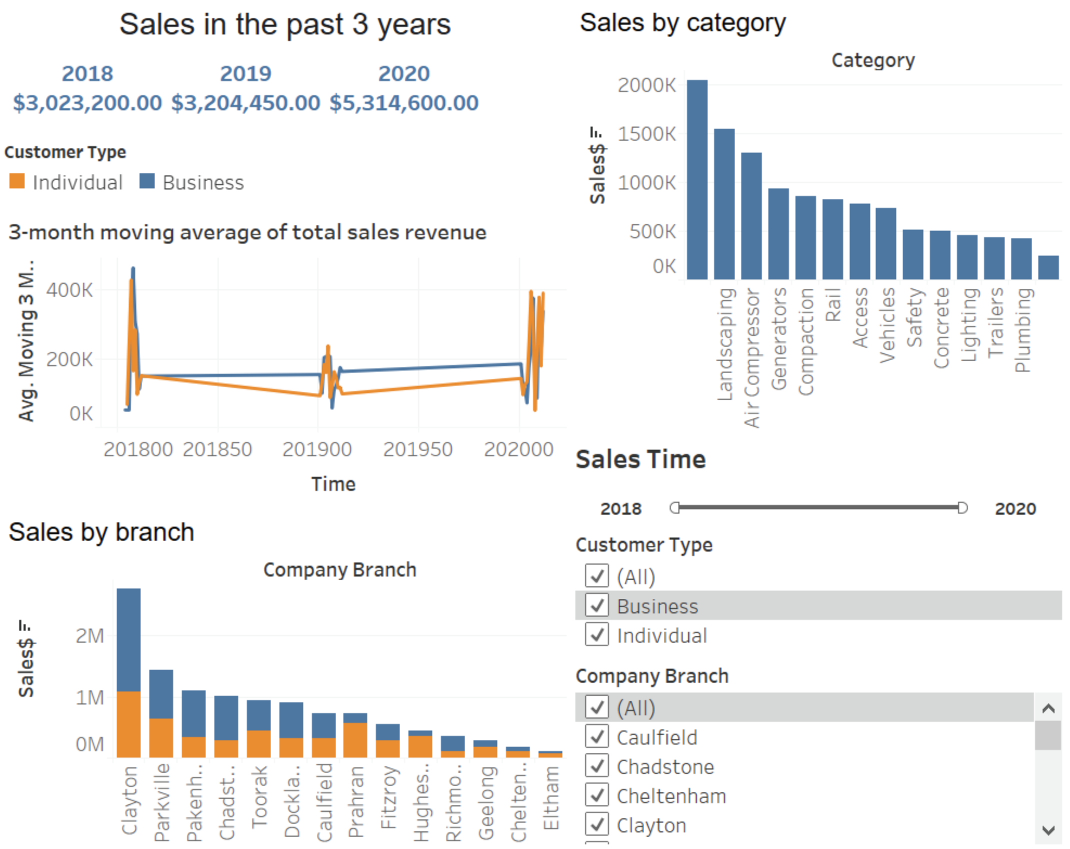
```
--d. Reports with Partitions:
--REPORT 9
SELECT cb.COMPANY_BRANCH, ht.HIRE_TIMEID AS month,
TO_CHAR(SUM(hf.NUM_EQUIPMENT_HIRED)) AS number_of_equipment_hired,
RANK() OVER (PARTITION BY cb.COMPANY_BRANCH
ORDER BY SUM(hf.NUM_EQUIPMENT_HIRED) DESC) AS RANK_BY_COMPANY_BRANCH,
RANK() OVER (PARTITION BY ht.HIRE_TIMEID
ORDER BY SUM(hf.NUM_EQUIPMENT_HIRED) DESC) AS RANK_BY_MONTH
FROM Hire_TimeDIM ht,Company_BranchDIM cb,hireFACT hf
WHERE ht.HIRE_TIMEID = hf.HIRE_TIMEID
AND cb.COMPANY_BRANCH = hf.COMPANY_BRANCH
GROUP BY cb.COMPANY_BRANCH, ht.HIRE_TIMEID
ORDER BY RANK_BY_COMPANY_BRANCH;
```

Fetched 50 rows in 0.048 seconds

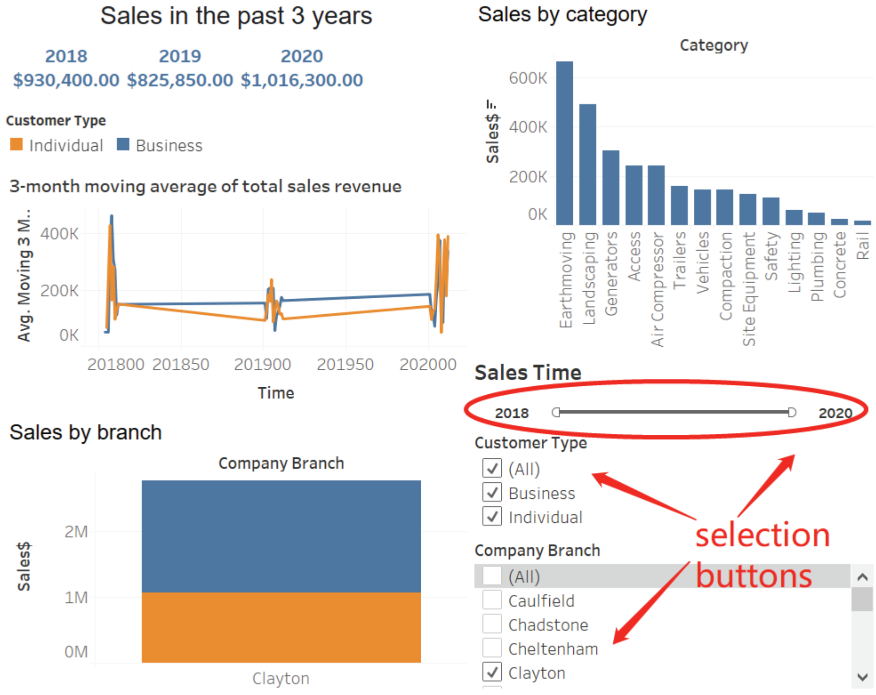
	COMPANY_BRANCH	MONTH	NUMBER_OF_EQUIPMENT_HIRED	RANK_BY_COMPANY_BRANCH	RANK_BY_MONTH
1	Caulfield	201811	6	1	1
2	Caulfield	201908	6	1	1
3	Toorak	202010	6	1	1
4	Richmond	202010	6	1	1
5	Prahran	202009	5	1	1
6	Prahran	201904	5	1	2
7	Parkville	201909	3	1	2
8	Parkville	202003	3	1	2
9	Parkville	202004	3	1	2
10	Parkville	202005	3	1	2

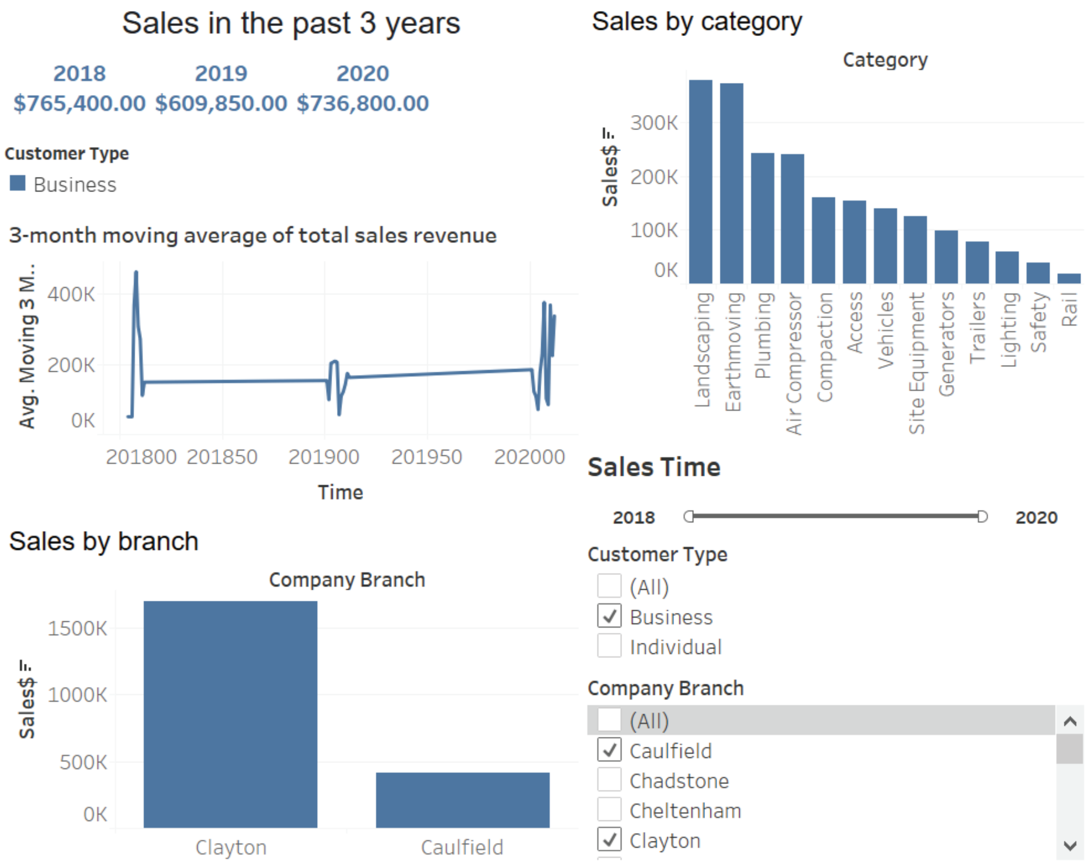
4. Business Intelligence (BI) Reports.

About the BI report, I choose to use the report5, 6 and 8 to do the presentation., and use the Tableau put above reports into the dashboard. Furthermore, this presentation can show the sales fact to the management. Managers can directly see the 3 moths moving average sales data of the past three years, the sales revenue of various branches, and the revenue of different equipment category on the dashboard (see the figure below).



This dashboard provides users with three selection buttons: time, company branch, and customer type. Users can easily choose the data they want to know and use these data for comparison (see the figure below).





Sales in the past 3 years

2018	2019	2020
$765,400.00	$609,850.00	$736,800.00

Customer Type

■ Business

3-month moving average of total sales revenue

Sales by branch

Sales by category

Sales Time

2018 — 2020

Customer Type

- □ (All)
- ☑ Business
- □ Individual

Company Branch

- □ (All)
- ☑ Caulfield
- □ Chadstone
- □ Cheltenham
- ☑ Clayton