Insurance Claims for Vehicles that have been serviced at specific branches

1. Identify a business we are interested in.

https://www.kaggle.com/lukamauto/insurance-claim-info-for-vehicles-being-serviced

The data contains the information on the recorded insurance claims for vehicles that have been serviced at specific branches. It also contains turnaround times for each status a claim goes through.

This data is interesting because we got to know the number of insurance claim cases received by the insurance company in different branches, which model comes frequently for insurance claims and how much time they take to deal with these cases.

We have chosen this dataset because this data can give us insights on which type of vehicle category are more likely to claim insurance which can help you price your insurance for different categories.

2. Three business functions we would like to build a data warehouse

Business Functions:

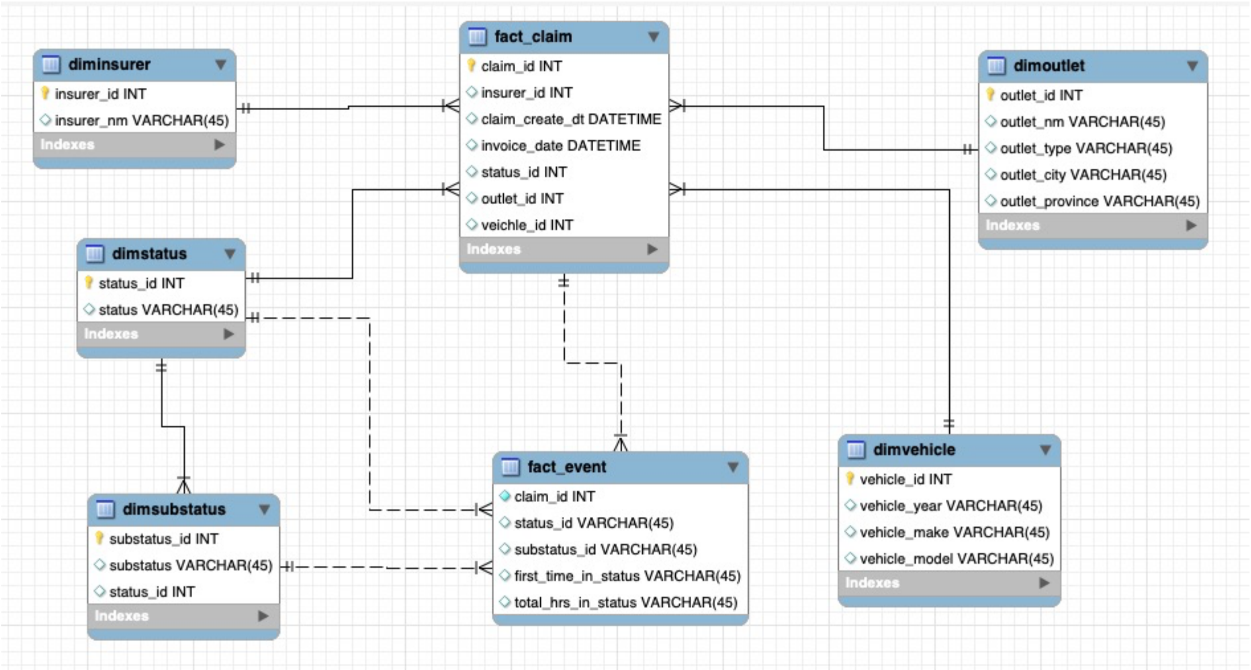
1) Showing the history status of each claim and spending time.
2) Showing the detailed status of the claim.
3) Showing the vehicle report of claim.

Our approach:

The dataset is in the data warehouse of recorded insurance claims for vehicles that have been serviced at specific branches. The dataset contains information regarding these vehicles (brand, model, and year) and the different stages each claim goes through for a particular vehicle before the claim is invoiced. The data also reflects the current status of each claim and sub statuses the claim has been through in its life cycle.

The original dataset is the basic fact table containing numeric measures produced by an operational measurement event. Then, we start changing the dimension of the dataset. Break into insurer, status, sub status, outlet, and vehicle dimension tables which contain only related features of the dimension. We observe that two columns(event status and event sub status) are the extension of two dimension tables(status and sub status). So we would like to build them as a consolidated fact table(event) to store the processing of each claim with (first time in status and total hours in status). Besides, we create another factless fact table (claim) that stores entities coming together at the moment in time with two features related to the claim (claim create date and invoice date).

ERD:



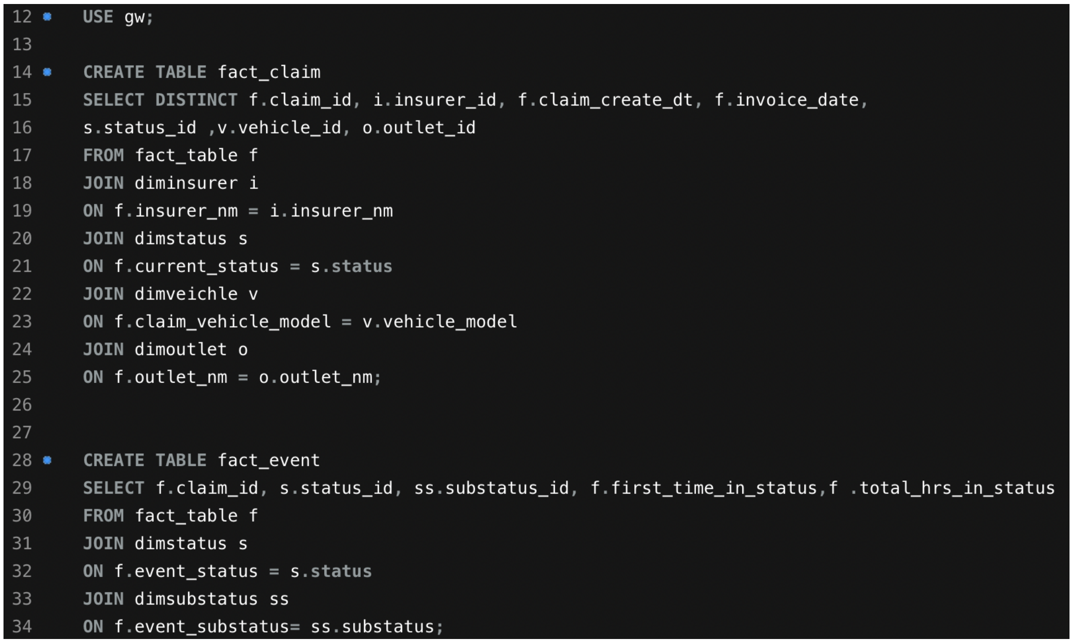
3. Value proposition

The value proposition for an insurance company is:

- Providing better Insurance premium to the customers
- Providing better claim status
- Reducing the turnaround time for vehicle
- Streamlining the claim process

4. Load schema with data

We break the original dataset into dimension tables (dimoutlet, dimstatus, dimsubstatus, and dimvehicle) and load table data by import wizard. Besides, we query from the original fact table to create two fact tables, fact_claim and fact_event.

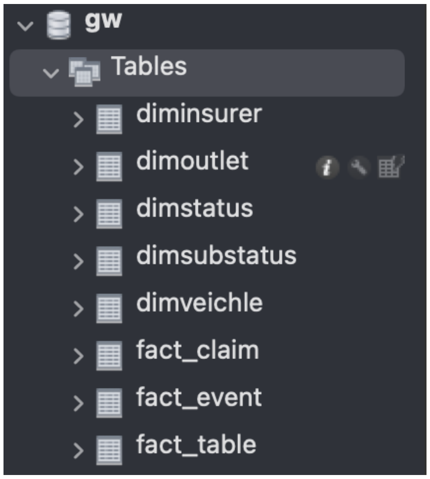


```sql
USE gw;

CREATE TABLE fact_claim
SELECT DISTINCT f.claim_id, i.insurer_id, f.claim_create_dt, f.invoice_date,
s.status_id ,v.vehicle_id, o.outlet_id
FROM fact_table f
JOIN diminsurer i
ON f.insurer_nm = i.insurer_nm
JOIN dimstatus s
ON f.current_status = s.status
JOIN dimveichle v
ON f.claim_vehicle_model = v.vehicle_model
JOIN dimoutlet o
ON f.outlet_nm = o.outlet_nm;


CREATE TABLE fact_event
SELECT f.claim_id, s.status_id, ss.substatus_id, f.first_time_in_status,f .total_hrs_in_status
FROM fact_table f
JOIN dimstatus s
ON f.event_status = s.status
JOIN dimsubstatus ss
ON f.event_substatus= ss.substatus;
```

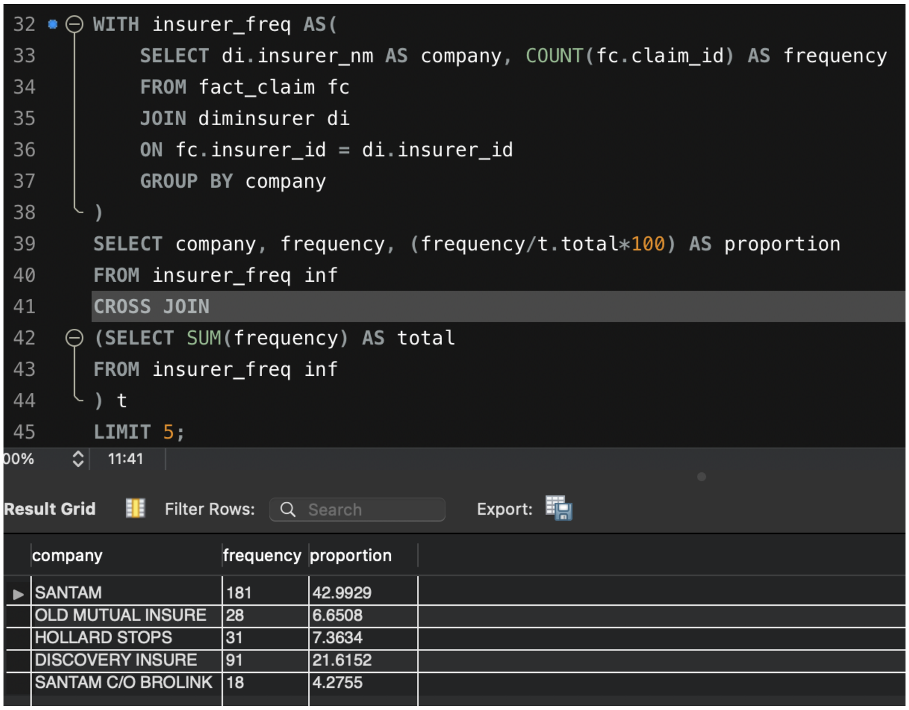
Here are the tables we have in our schema:



gw
- Tables
  - diminsurer
  - dimoutlet
  - dimstatus
  - dimsubstatus
  - dimveichle
  - fact_claim
  - fact_event
  - fact_table

5. 10 business question and queries (the use of lag, lead, join, rank, etc.)

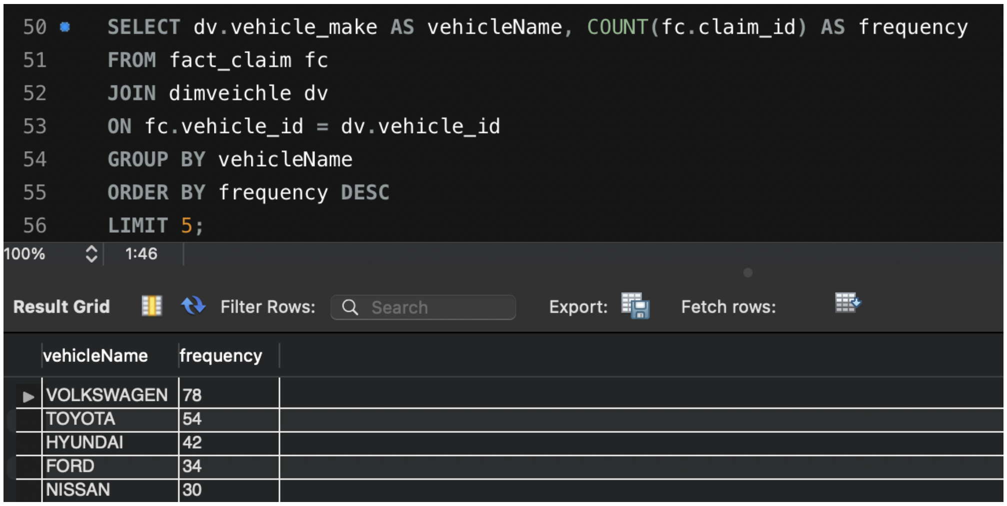
1) Getting the top five insurers and seeing how many cases they placed among the 996 unique records. (Knowing which insurance company holds the largest portion of dealing cases)



```
WITH insurer_freq AS(
    SELECT di.insurer_nm AS company, COUNT(fc.claim_id) AS frequency
    FROM fact_claim fc
    JOIN diminsurer di
    ON fc.insurer_id = di.insurer_id
    GROUP BY company
)
SELECT company, frequency, (frequency/t.total*100) AS proportion
FROM insurer_freq inf
CROSS JOIN
(SELECT SUM(frequency) AS total
FROM insurer_freq inf
) t
LIMIT 5;
```

company	frequency	proportion
SANTAM	181	42.9929
OLD MUTUAL INSURE	28	6.6508
HOLLARD STOPS	31	7.3634
DISCOVERY INSURE	91	21.6152
SANTAM C/O BROLINK	18	4.2755

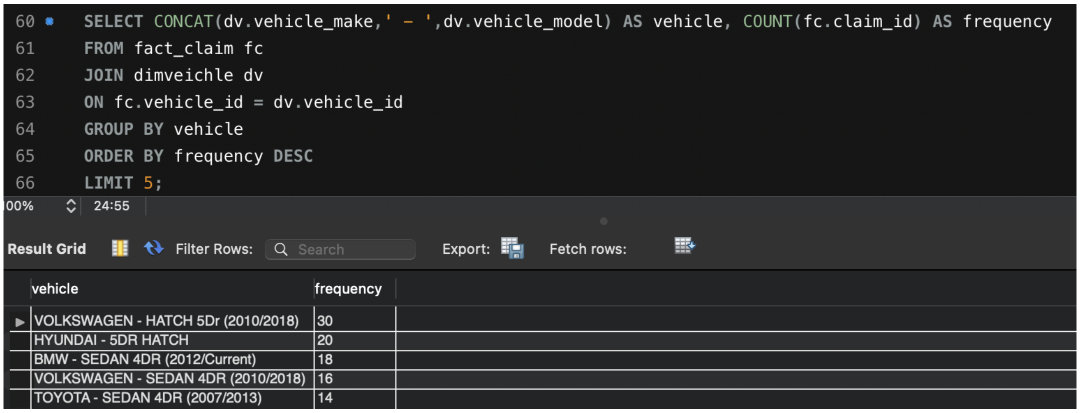
2) Getting the top five counts of vehicle_make to see which vehicle_make has the highest claim counts.



```
SELECT dv.vehicle_make AS vehicleName, COUNT(fc.claim_id) AS frequency
FROM fact_claim fc
JOIN dimveichle dv
ON fc.vehicle_id = dv.vehicle_id
GROUP BY vehicleName
ORDER BY frequency DESC
LIMIT 5;
```

vehicleName	frequency
VOLKSWAGEN	78
TOYOTA	54
HYUNDAI	42
FORD	34
NISSAN	30

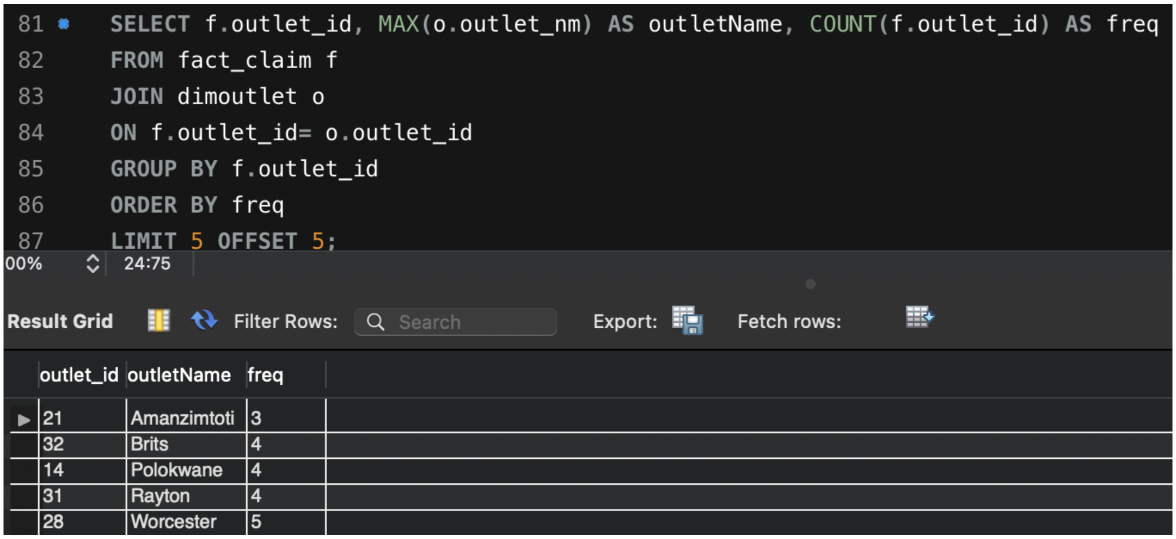
3) Getting the top five counts of claim_vehicle_model to see which vehicle MODEL has the highest claim counts.



```
SELECT CONCAT(dv.vehicle_make,' - ',dv.vehicle_model) AS vehicle, COUNT(fc.claim_id) AS frequency
FROM fact_claim fc
JOIN dimveichle dv
ON fc.vehicle_id = dv.vehicle_id
GROUP BY vehicle
ORDER BY frequency DESC
LIMIT 5;
```

vehicle	frequency
VOLKSWAGEN - HATCH 5Dr (2010/2018)	30
HYUNDAI - 5DR HATCH	20
BMW - SEDAN 4DR (2012/Current)	18
VOLKSWAGEN - SEDAN 4DR (2010/2018)	16
TOYOTA - SEDAN 4DR (2007/2013)	14

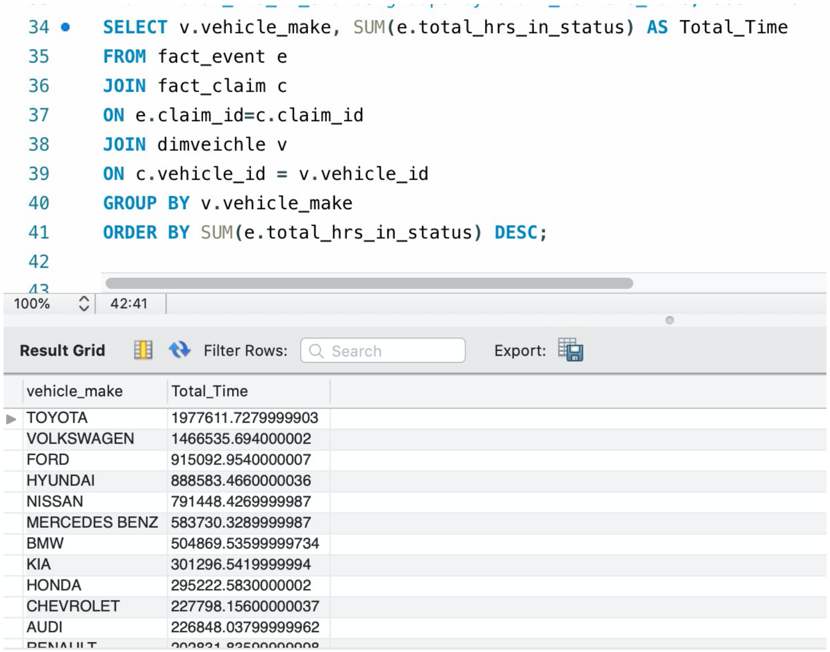
4) Getting the top 5-10 counts of outlet_name to see what the distinguishing outlet are dominating the insurance claims.



```
SELECT f.outlet_id, MAX(o.outlet_nm) AS outletName, COUNT(f.outlet_id) AS freq
FROM fact_claim f
JOIN dimoutlet o
ON f.outlet_id= o.outlet_id
GROUP BY f.outlet_id
ORDER BY freq
LIMIT 5 OFFSET 5;
```

outlet_id	outletName	freq
21	Amanzimtoti	3
32	Brits	4
14	Polokwane	4
31	Rayton	4
28	Worcester	5

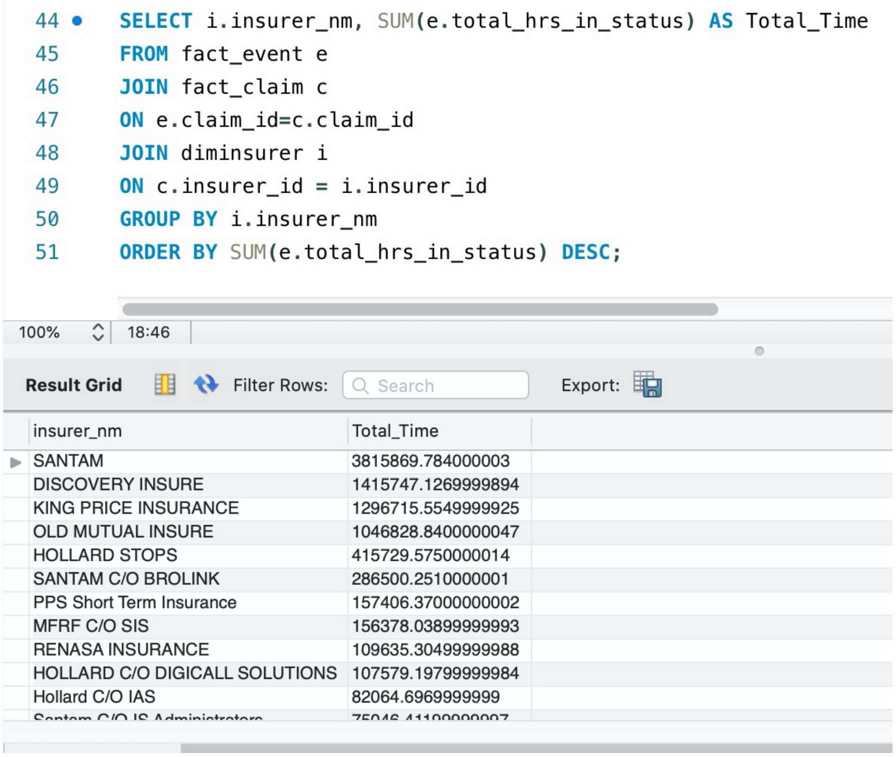
5) Rank total_hrs_in_status groups by claim_vehicle_make, see which car consumes the most time to repair.



```sql
SELECT v.vehicle_make, SUM(e.total_hrs_in_status) AS Total_Time
FROM fact_event e
JOIN fact_claim c
ON e.claim_id=c.claim_id
JOIN dimveichle v
ON c.vehicle_id = v.vehicle_id
GROUP BY v.vehicle_make
ORDER BY SUM(e.total_hrs_in_status) DESC;
```

vehicle_make	Total_Time
TOYOTA	1977611.7279999903
VOLKSWAGEN	1466535.694000002
FORD	915092.9540000007
HYUNDAI	888583.4660000036
NISSAN	791448.4269999987
MERCEDES BENZ	583730.3289999987
BMW	504869.53599999734
KIA	301296.5419999994
HONDA	295222.5830000002
CHEVROLET	227798.15600000037
AUDI	226848.03799999962
RENAULT	202831.82599999998

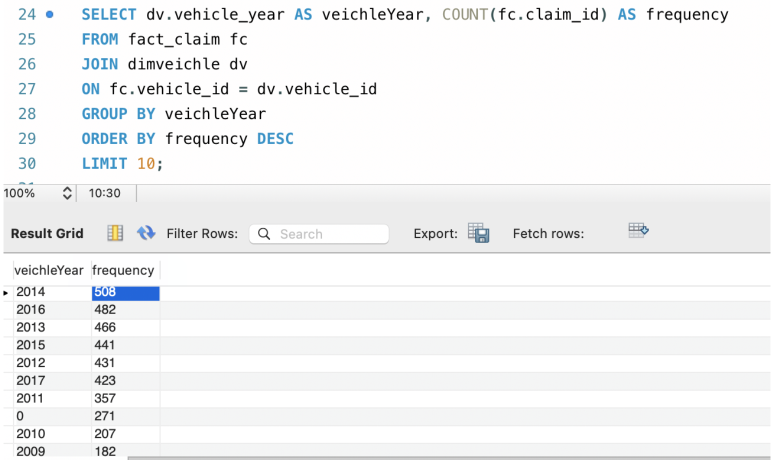
6) Sum total_hrs_in_status groups by insurer_nm, see which insurer contributed the most time of the claim.



```sql
SELECT i.insurer_nm, SUM(e.total_hrs_in_status) AS Total_Time
FROM fact_event e
JOIN fact_claim c
ON e.claim_id=c.claim_id
JOIN diminsurer i
ON c.insurer_id = i.insurer_id
GROUP BY i.insurer_nm
ORDER BY SUM(e.total_hrs_in_status) DESC;
```

insurer_nm	Total_Time
SANTAM	3815869.784000003
DISCOVERY INSURE	1415747.1269999894
KING PRICE INSURANCE	1296715.5549999925
OLD MUTUAL INSURE	1046828.8400000047
HOLLARD STOPS	415729.5750000014
SANTAM C/O BROLINK	286500.2510000001
PPS Short Term Insurance	157406.37000000002
MFRF C/O SIS	156378.03899999993
RENASA INSURANCE	109635.30499999988
HOLLARD C/O DIGICALL SOLUTIONS	107579.19799999984
Hollard C/O IAS	82064.6969999999
Santam C/O IS Administrators	75046.41199999997

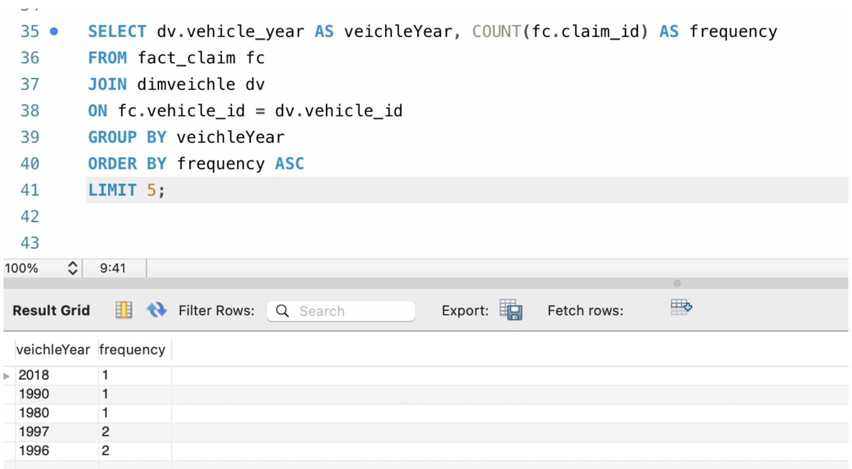
7) Count which claim_vehicle_year has the most claim case



```sql
SELECT dv.vehicle_year AS veichleYear, COUNT(fc.claim_id) AS frequency
FROM fact_claim fc
JOIN dimveichle dv
ON fc.vehicle_id = dv.vehicle_id
GROUP BY veichleYear
ORDER BY frequency DESC
LIMIT 10;
```

veichleYear	frequency
2014	508
2016	482
2013	466
2015	441
2012	431
2017	423
2011	357
0	271
2010	207
2009	182

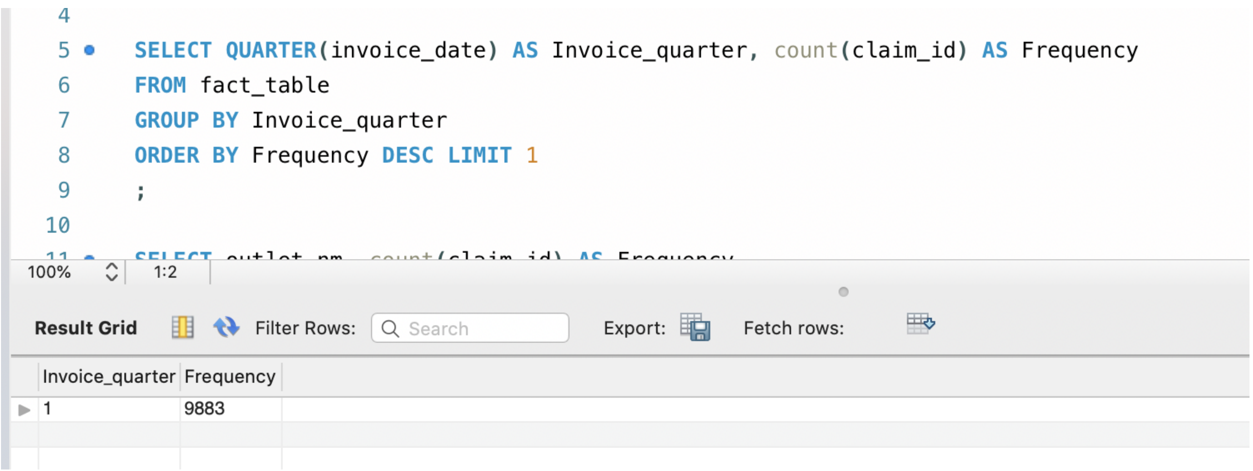
8) Get the 5 lowest claim count vehicle_year to see which vehicle years have the lowest claim cases.



```sql
SELECT dv.vehicle_year AS veichleYear, COUNT(fc.claim_id) AS frequency
FROM fact_claim fc
JOIN dimveichle dv
ON fc.vehicle_id = dv.vehicle_id
GROUP BY veichleYear
ORDER BY frequency ASC
LIMIT 5;
```

veichleYear	frequency
2018	1
1990	1
1980	1
1997	2
1996	2

9) Check invoice_date to see which quarter of the year has the most claim cases.

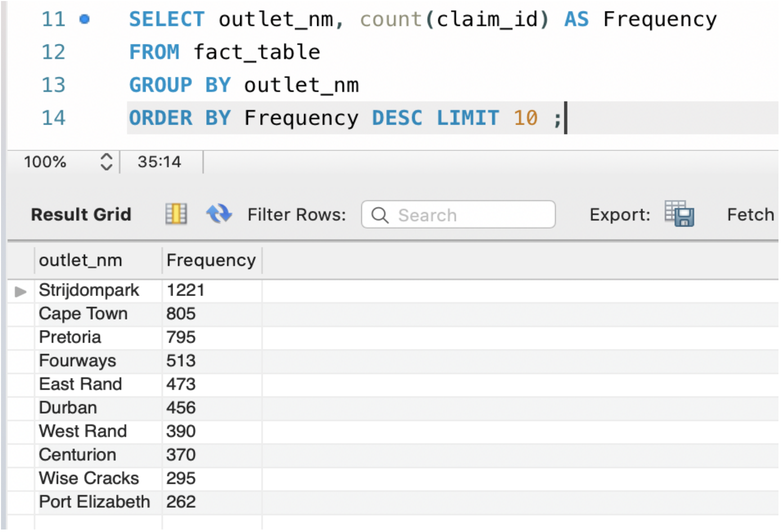


```sql
SELECT QUARTER(invoice_date) AS Invoice_quarter, count(claim_id) AS Frequency
FROM fact_table
GROUP BY Invoice_quarter
ORDER BY Frequency DESC LIMIT 1
;

SELECT outlet_nm, count(claim_id) AS Frequency
```

Invoice_quarter	Frequency
1	9883

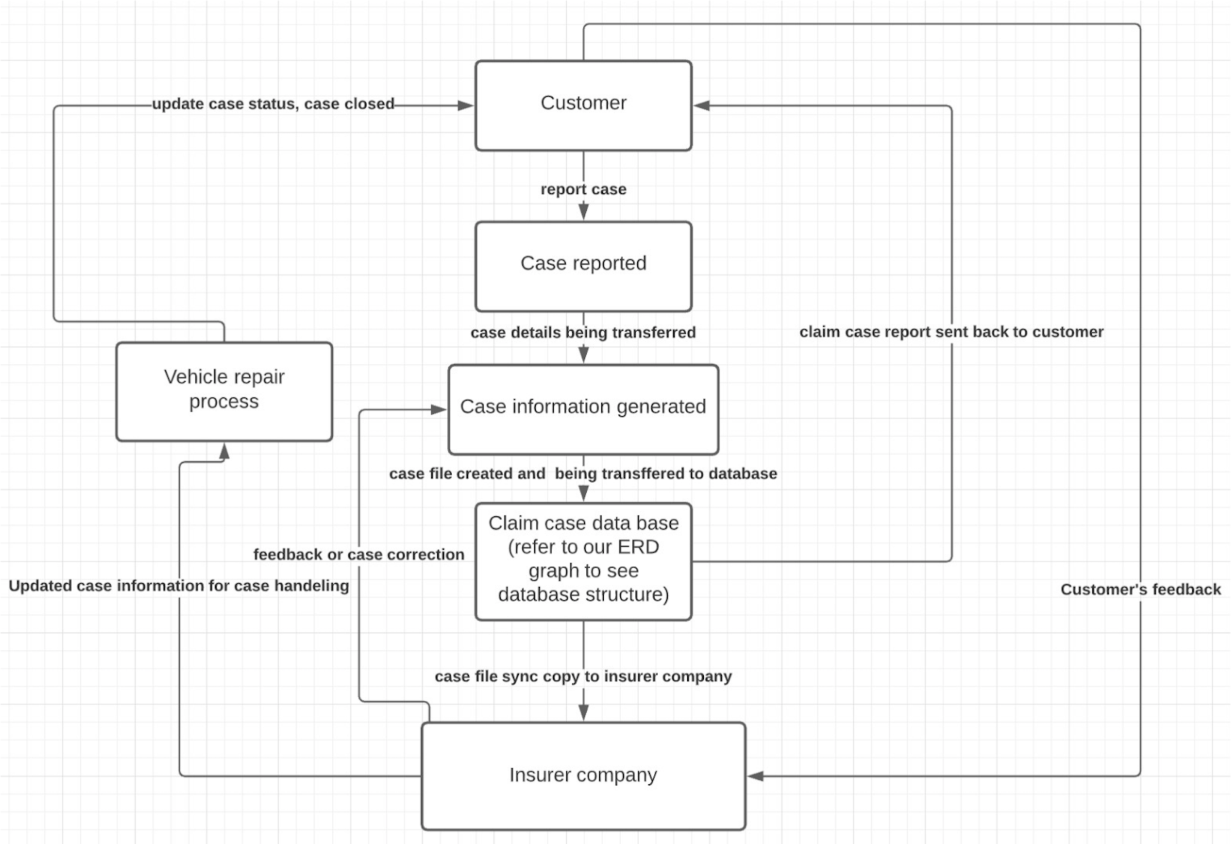
10) Check which outlet_nm has the most cases, count cases by outlet_nm group



```sql
SELECT outlet_nm, count(claim_id) AS Frequency
FROM fact_table
GROUP BY outlet_nm
ORDER BY Frequency DESC LIMIT 10 ;
```

outlet_nm	Frequency
Strijdompark	1221
Cape Town	805
Pretoria	795
Fourways	513
East Rand	473
Durban	456
West Rand	390
Centurion	370
Wise Cracks	295
Port Elizabeth	262

6. Reference section



Customer
update case status, case closed
report case
Case reported
case details being transferred
claim case report sent back to customer
Vehicle repair process
Case information generated
case file created and being transffered to database
Claim case data base (refer to our ERD graph to see database structure)
feedback or case correction
Updated case information for case handeling
Customer's feedback
case file sync copy to insurer company
Insurer company

This flow graph shows the processes of how claim cases and their data are reported, processed, stored, and synced to the insurance company and the clients of the insurance company. At first, when a claim case is being reported by the customer, the system will collect necessary data from the customers’ report, for example, the insurer’s id, case status, vehicle info, and the fact claim details. When each report is generated, the case report will be transferred to our database system and stored according to our database structure indicated on our ERD graph. Then the system will sync the case report to both the insurance company (for further claim process) and the customer (for case report copy). When the claim process is updated, the system will send a notification to the customer. At the same time, the customer could provide any feedback or case correction info to the insurer company, which insurance company can update feedback or case correction to the database anytime. According to our database system, the data types will be collected. We can optimize the resources we have to provide for our customers and get our customers lower prices of the insurance coverage. And also shorten the duration of claim status. Most importantly, we can make sure the claim process is simple enough for our customers.