



## The Requirement

Create a report that gives aggregate monthly statistics on employer accounts and their superannuation payments, per month and per “employer tier”.

The report takes the form of a table, with the following columns:

1. **Tier** – the employer tier (i.e. segmentation category) of this row
2. **Month end date** – the end date for the month of this row
3. **Num payments** – the total number of superannuation contributions paid for this month/tier
4. **Amount of payments** – the total amount of superannuation contributions paid for this month/tier
5. **New employers** – newly open employer accounts introduced within this month at this tier
6. **Open employers at EOM (end of month)** – number of employer accounts open at the month end date at this tier

The report is to cover **January 2018 to December 2018** and ordered by “Tier” and “Month end date”.

A partial example of how the result table is to be structured:

Tier	Month end date	Num payments	Amount of payments	New employers	Open employers at EOM
...	...	...	...	...	...
1	30/04/2018	45	77125.12	1	50
1	31/05/2018	38	100502.32	0	50
1	30/06/2018	37	94994.46	1	51
...	...	...	...	...	...

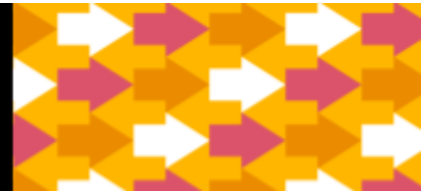
The process to create the report must be easily repeatable and as automated as possible. You do not have to automate your import of the data provided or your export of final results.

## Data provided

Two files are provided:

- **Employer master.csv** – Historical employer account data in the form of a slowly changing dimension
- **Payment transactions.csv** – Superannuation contribution transactions

***N.B. All data supplied is dummy data (it is purely fictional) - it does not contain any personal identifiable information.***



## Technology instructions

It is preferable if you use Microsoft SQL Server and its associated languages and tools (SQL, Transact-SQL, SSIS if you need it). Microsoft SQL Server Express Edition and SQL Server Data Tools are available free download from their website. The use of Python will also be accepted, which is also free to download.

## Approach and Guidelines

Our purpose for this test is to gain insight into:

- Skill and efficiency with data and with solving a problem of some complexity
- How you decide on the correct logic/approach and how rigorously you validate your assumptions and conclusions
- What insights you might gain about solving the problem from the data itself
- How you write, structure and organise your code for a task with some complexity
- How you validate and ensure the correctness of your results

Please design your work to showcase these aspects.

Due to time constraints, we are unable to answer questions regarding this test. If you have any questions, the intention is that you should be able to derive the answers from the data with confidence and explain how you have done so. If any ambiguities remain, please document them, and make the assumptions that seem most reasonable to you.

## Required deliverables

Please provide us with three files, one for each of the following deliverables:

1. **Results** - The full results table, formatted as a CSV file.
2. **Code** - All of your code (with comments) that has been written to complete the task (including any additional code that does not contribute to the final output).
3. **Assumptions and approach** - Any notes detailing your approach and decisions made based on any assumptions you have relied on.

## Test time

Upon receiving this test you, the testee, will have 8 hours to complete and to submit your deliverables. Deliverables received outside of the 8 hour test period will be disqualified at our discretion, and may **only** be evaluated at our discretion if there are any mitigating circumstances.