SWINBURNE UNIVERSITY OF TECHNOLOGY

Research Report -

Global (Cloud) Databases vs Local Databases:

A General Overview

NAME:	LYAN HENG
ID:	102240691
DATE OF SUBMISSION:	05/11/2020
CLASS:	COS20015 –

FUNDAMENTAL OF DATA MANAGEMENT

Table of Content

I. Abstract

This research project is designed to study the possibility, advantages, and disadvantages of implementing a startup's database using a cloud-based database. This research analyses the advantages and disadvantages of using a cloud-based infrastructure, specifically a cloud-based database to improve the effectiveness and efficiency of startup companies. An interview with a game developing startup, in particular, will be studied to understand the advantages and challenges they faced in the use of a cloud-based database system from many aspects. Some aspects include classification of databases, the convenience of databases management, and speed accessing of databases. For efficiency, the report will use Microsoft's cloud product: Microsoft Azure as the primary cloud database to be studied and implemented. This report will also cover the similarities and differences of a global-accessed database and a locally-accessed database.

II. Project Scenario

Cloud Computing is a trending topic amongst the Startup technology world. With the introduction of cloud computing in the market, many startup companies have adopted cloud-based computer systems as their central ICT infrastructure (Ferri et al. 2017). A cloud-based database is one of the many products cloud service providers provide. Cloud-based databases present a more straightforward and cheaper approach to setting up a startup's ICT infrastructure. As stated in an interview conducted about AlphaGame, an Italian Game Development startup Company, AlphaGame's founder stated that the founders have "had this business idea several years before implementation, but it was not financially feasible." (Ferri et al. 2017).

The recent event of Covid-19 also presents a new perspective and challenge to any startup company. Due to the pandemic situation, all developers will have no access to the company's infrastructure, if they were built. Having the database server stored locally and bounded by the physical distance to the server, a new consideration was brought up within Softtech's management team to migrate all data to the cloud.

Due to the cost of the operation and presented risks, studies are carefully conducted, and careful considerations are being made in order to make sure every step of the operation goes smoothly and desirably.

III. Advantages of the Adaptation of Cloud-based Database

In the Journal Study titled "Analyzing Cloud-based Startups: Evidence from a Case Study in Italy" by the University of Naples Federico II, Italy, an interview of a company called AlphaGame was conducted to understand the advantages and disadvantages of the adaptation of a Cloud-based service as its ICT infrastructure.

Cloud computing is not ideal for all firms (Jorge Passos da Costa & Miguel Rosado da Cruz 2012). The advantages and disadvantages of cloud largely depend on the size and structure of the firm. However, a trend remains that for startup companies, adopting a cloud-based database or cloud computing, in general, can provide more advantages than disadvantages.

Even though brief, the research on AlphaGame showed that cloud-based computing brings up a total cut loss of up to 37% of ICT costs and 25% of maintenance operations (Ferri et al. 2017). A framework shown in the research paper shows the five main aspects that justify cloud computing. Those main aspects are “increased global collaboration, reduced opportunity costs, scalability, access to the global market, and access to international venture capital” (Ferri et al., 2017).

A further look into different aspects between the use of cloud-based databases and a locally-hosted database are discussed further in the section below.

IV. Cloud Service Provider - Microsoft Azure

Microsoft Azure is a cloud service provider that provides cloud computing services to any corporation or individual. The cloud computing products that Microsoft Azure provides include all sorts of computing that can be done in another server abroad (Hoffman, 2018). This includes products like Virtual Machine, App Storage, backups, and databases. Microsoft Azure was launched in 2008 (MSV 2020). When it was launched, it did not catch much attention as a cloud-based platform, due to the lack of popularity of a cloud-based platform. However, when Microsoft started to launch its “mobile-first cloud-first” strategy, Azure quickly became one of the giants of cloud computing services, second to only AWS, which is Amazon backed. Azure is also known for revolutionising the concept of Software as a Service (Saas) (MSV 2020). The focus of this research report is to look into the use of Azure’s third pillar - Azure SQL - which mostly resembles Microsoft SQL Server.

V. Global (Cloud) DB vs Local DB

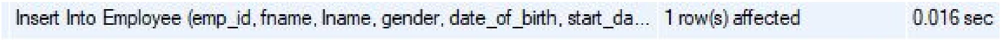
Cloud-based databases come with their pros and cons. Cloud-based databases can allow flexibility and room for growth, but if obstruction of daily operation occurs, the harm will overwrite the benefits. Any cloud-based databases have weaknesses; Azure is similar. Below are technical and financial comparisons between a cloud-based database and a local database, respectively using Microsoft Azure and MariaDB as prime examples.

A. Data Accessing Speed

A detailed data-accessing study dives into each query type provided. Some of the leading data accessing and manipulating methods are inserting, updating, and deletion. However, as this research focuses on the larger picture of the global and local system, a study of one can be used to infer the others. Insertion is a perfect candidate, as Sofftech's operation will rely mostly on insertion.

Any DLL statement even though small, takes time. According to MariaDB, a typical non-optimised instance of MariaDB can execute 476 insert queries in a second (MariaDB 2015?). This depends on multiple other factors; however, in general, the speed of MariaDB upserting into a table would be roughly 2ms per insert query.

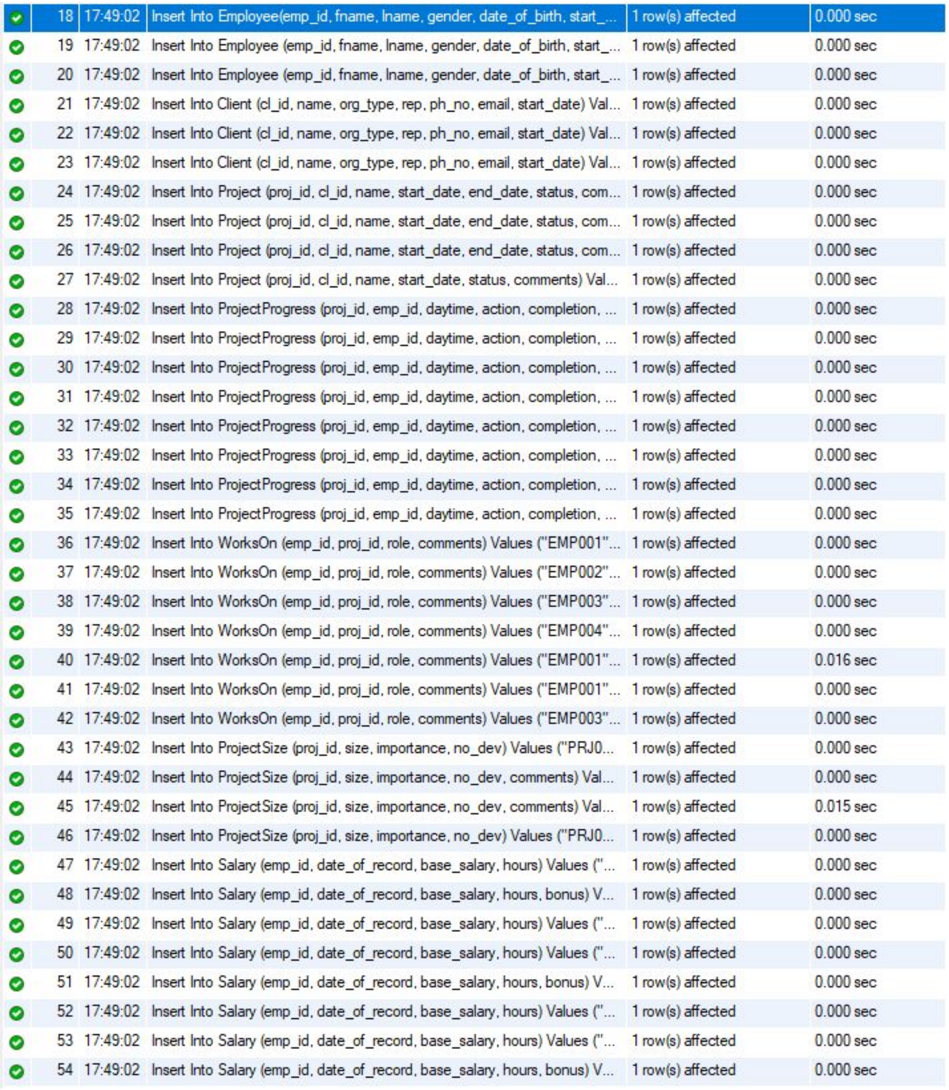
MariaDB inserting into a table:



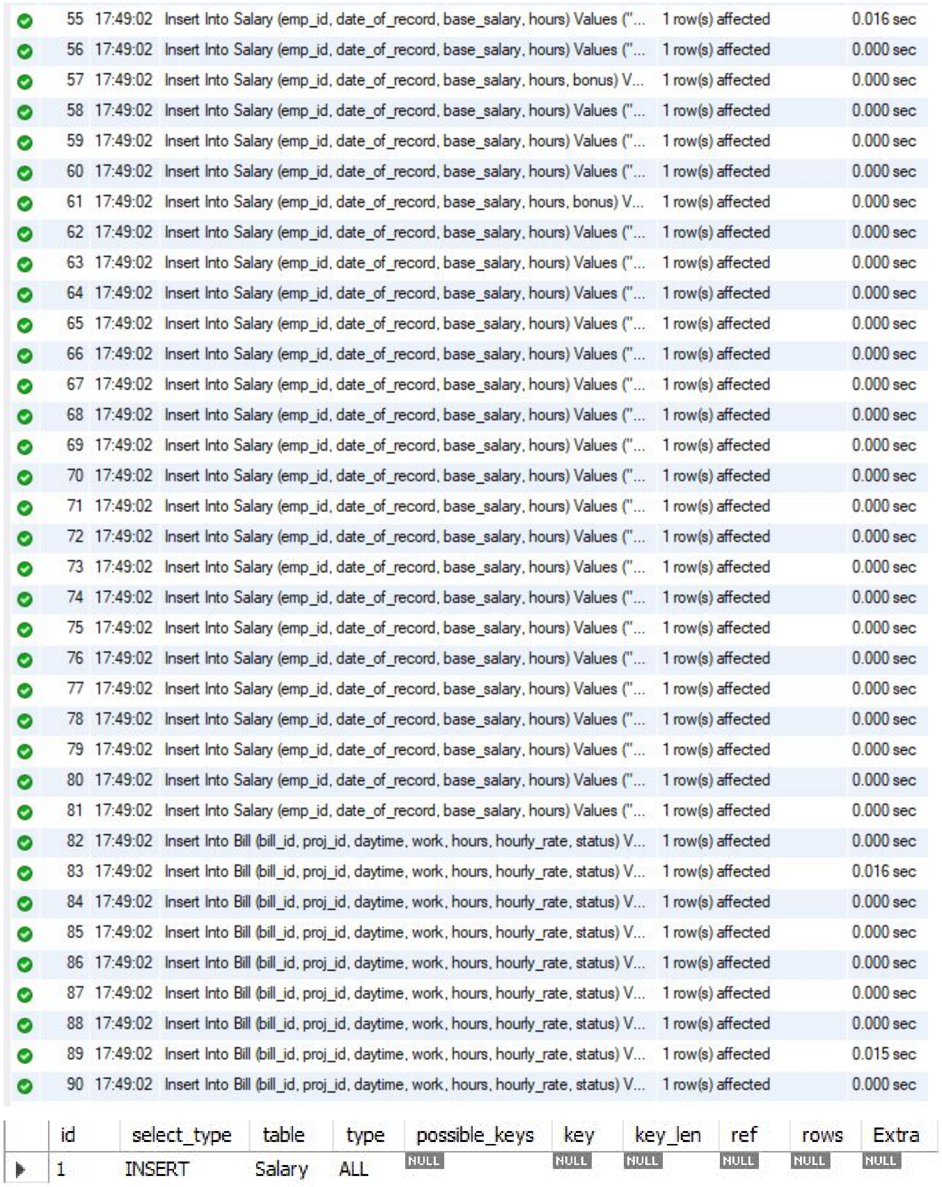
Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_da...	1 row(s) affected	0.016 sec

Figure 5.1 - Speed of 1 insert - Obtained from MySQL Workbench

The speed of the insertion decreases as the number of tasks increases or any lock modes are turned on. However, MariaDB was aware of the issue. Whenever a large chunk of updating is done, optimisation is applied. Below shows 73 insertions executed in a go. MariaDB optimises the process, keeping the insertion time roughly the same for every 20 queries.



✓	18	17:49:02	Insert Into Employee(emp_id, fname, lname, gender, date_of_birth, start_...	1 row(s) affected	0.000 sec
✓	19	17:49:02	Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_...	1 row(s) affected	0.000 sec
✓	20	17:49:02	Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_...	1 row(s) affected	0.000 sec
✓	21	17:49:02	Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Val...	1 row(s) affected	0.000 sec
✓	22	17:49:02	Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Val...	1 row(s) affected	0.000 sec
✓	23	17:49:02	Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Val...	1 row(s) affected	0.000 sec
✓	24	17:49:02	Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, com...	1 row(s) affected	0.000 sec
✓	25	17:49:02	Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, com...	1 row(s) affected	0.000 sec
✓	26	17:49:02	Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, com...	1 row(s) affected	0.000 sec
✓	27	17:49:02	Insert Into Project (proj_id, cl_id, name, start_date, status, comments) Val...	1 row(s) affected	0.000 sec
✓	28	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	29	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	30	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	31	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	32	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	33	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	34	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	35	17:49:02	Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, ...	1 row(s) affected	0.000 sec
✓	36	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001"...	1 row(s) affected	0.000 sec
✓	37	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP002"...	1 row(s) affected	0.000 sec
✓	38	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP003"...	1 row(s) affected	0.000 sec
✓	39	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP004"...	1 row(s) affected	0.000 sec
✓	40	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001"...	1 row(s) affected	0.016 sec
✓	41	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001"...	1 row(s) affected	0.000 sec
✓	42	17:49:02	Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP003"...	1 row(s) affected	0.000 sec
✓	43	17:49:02	Insert Into ProjectSize (proj_id, size, importance, no_dev) Values ("PRJ0...	1 row(s) affected	0.000 sec
✓	44	17:49:02	Insert Into ProjectSize (proj_id, size, importance, no_dev, comments) Val...	1 row(s) affected	0.000 sec
✓	45	17:49:02	Insert Into ProjectSize (proj_id, size, importance, no_dev, comments) Val...	1 row(s) affected	0.015 sec
✓	46	17:49:02	Insert Into ProjectSize (proj_id, size, importance, no_dev) Values ("PRJ0...	1 row(s) affected	0.000 sec
✓	47	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...	1 row(s) affected	0.000 sec
✓	48	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) V...	1 row(s) affected	0.000 sec
✓	49	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...	1 row(s) affected	0.000 sec
✓	50	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...	1 row(s) affected	0.000 sec
✓	51	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) V...	1 row(s) affected	0.000 sec
✓	52	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...	1 row(s) affected	0.000 sec
✓	53	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...	1 row(s) affected	0.000 sec
✓	54	17:49:02	Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) V...	1 row(s) affected	0.000 sec



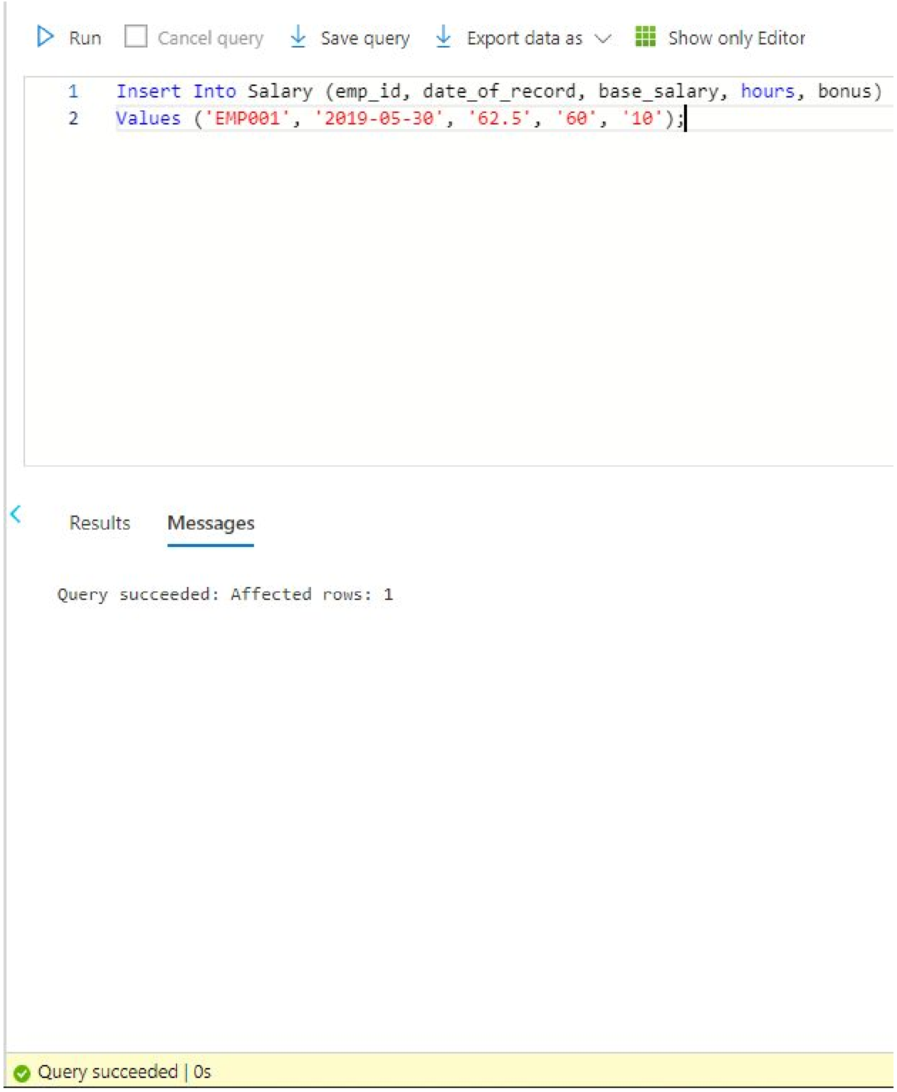
```
✓ 55 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.016 sec
✓ 56 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 57 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) V...  1 row(s) affected  0.000 sec
✓ 58 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 59 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 60 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 61 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) V...  1 row(s) affected  0.000 sec
✓ 62 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 63 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 64 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 65 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 66 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 67 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 68 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 69 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 70 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 71 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 72 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 73 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 74 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 75 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 76 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 77 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 78 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 79 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 80 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 81 17:49:02 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("...  1 row(s) affected  0.000 sec
✓ 82 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 83 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.016 sec
✓ 84 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 85 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 86 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 87 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 88 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
✓ 89 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.015 sec
✓ 90 17:49:02 Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) V...  1 row(s) affected  0.000 sec
```

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	INSERT	Salary	ALL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5.2 - Speed of Multiple Inserts with no Indexing - Obtained from MySQL Workbench

It is important to note that the speed of the insertions can be improved drastically by upgrading and improving hardware being used. However, the standard of the speed of insertion will be assumed to be 2ms without any hardware updates. It is also important to note that this was done without indexing. Indexing can improve the speed of insertion or selection by a considerate number.

Performing a similar operation on Azure, a different result can be seen. Azure's portal provides a service to run queries. The tool for measuring the speed of queries is similar to that of MySQL Workbench. However, the precision is much lower.



```
Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-05-30', '62.5', '60', '10');
```

```
Query succeeded: Affected rows: 1
```

Query succeeded | 0s

Figure 5.3 - Speed of Insert in Azure - Obtained from Microsoft Azure Portal Queries Section

The "work bench" tool in Azure only shows the speed of query up to the seconds. However, diving a bit deeper into Azure's Performance Tool, the following graph was shown:

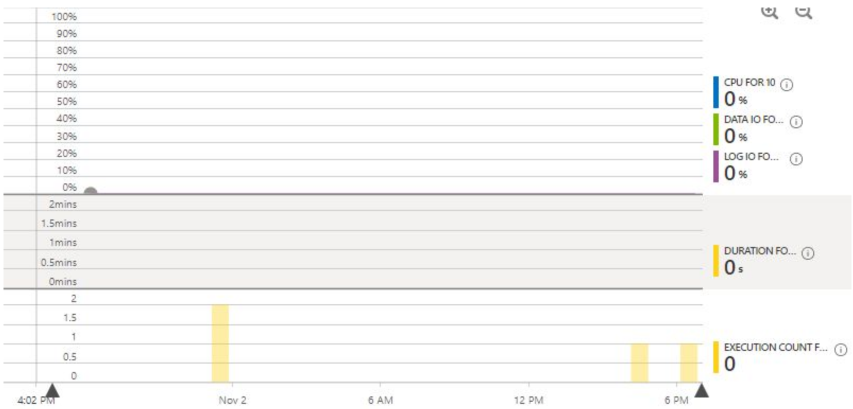
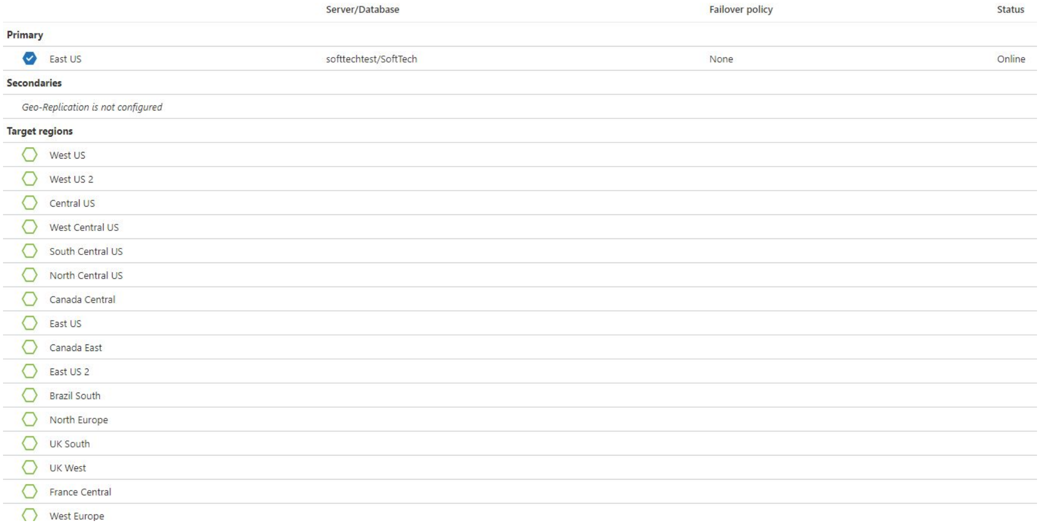


Figure 5.4 - Azure's Performance - Obtained from Microsoft Azure Portal

The lowest graph shows the speed of query execution up to the tenth of a second precision. The query executed above appears to take about 0.5s to 1s to execute.

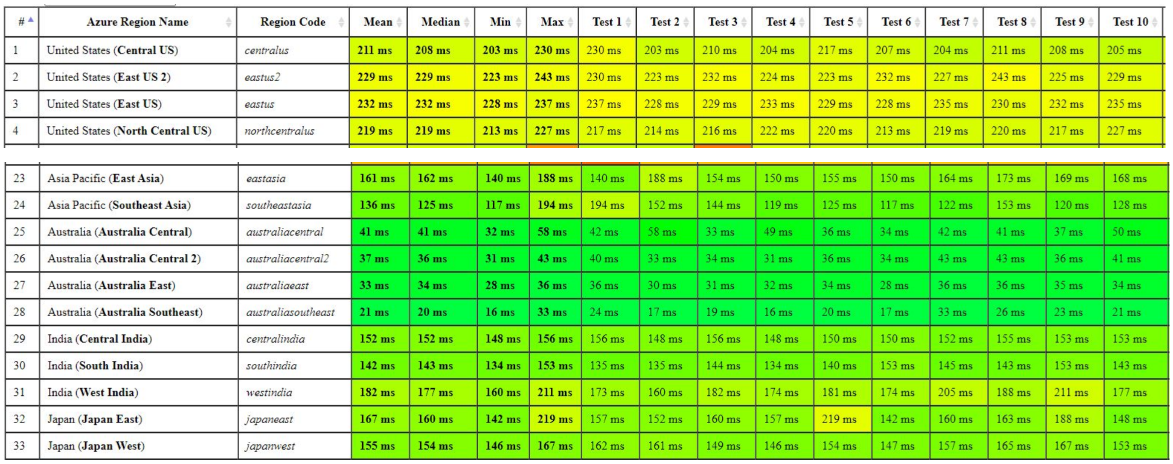
Compared to the 2ms query speed of MariaDB, it is no doubt that MariaDB execution's speed is faster. To put in comparison, a simple insert query in MariaDB is 375 faster than that of Azure. This does not come as a surprise. Being a cloud service, it takes time for any computer to send a patch of request to the selected data server. It also takes time for the data server to process the information and send it back.

It is important to note as well that Azure's free service only allows for the hosting of the server in the East USA region. The pinging of the East USA server is to be slower compared to databases being hosted in Australia or any of the Asia-Pacific Region. A test of 10 samples on the ping of the East USA region resulted in approximately 230 ms. The same test showed a ping of 20 ms - 40 ms ping in Azure's Australian Server and roughly 180 ms ping in the Asia-Pacific Area.



	Server/Database	Failover policy	Status
Primary			
✓ East US	softtechtest/SoftTech	None	Online
Secondaries			
Geo-Replication is not configured			
Target regions			
○ West US			
○ West US 2			
○ Central US			
○ West Central US			
○ South Central US			
○ North Central US			
○ Canada Central			
○ East US			
○ Canada East			
○ East US 2			
○ Brazil South			
○ North Europe			
○ UK South			
○ UK West			
○ France Central			
○ West Europe			

Figure 5.5 - Geo-Replication of Azure - Obtained from Microsoft Azure Portal



#	Azure Region Name	Region Code	Mean	Median	Min	Max	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
1	United States (**Central US**)	*centralus*	**211 ms**	**208 ms**	**203 ms**	**230 ms**	230 ms	203 ms	210 ms	204 ms	217 ms	207 ms	204 ms	211 ms	208 ms	205 ms
2	United States (**East US 2**)	*eastus2*	**229 ms**	**229 ms**	**223 ms**	**243 ms**	230 ms	223 ms	232 ms	224 ms	223 ms	232 ms	227 ms	243 ms	225 ms	229 ms
3	United States (**East US**)	*eastus*	**232 ms**	**232 ms**	**228 ms**	**237 ms**	237 ms	228 ms	229 ms	233 ms	229 ms	228 ms	235 ms	230 ms	232 ms	235 ms
4	United States (**North Central US**)	*northcentralus*	**219 ms**	**219 ms**	**213 ms**	**227 ms**	217 ms	214 ms	216 ms	222 ms	220 ms	213 ms	219 ms	220 ms	217 ms	227 ms
23	Asia Pacific (**East Asia**)	*eastasia*	**161 ms**	**162 ms**	**140 ms**	**188 ms**	140 ms	188 ms	154 ms	150 ms	155 ms	150 ms	164 ms	173 ms	169 ms	168 ms
24	Asia Pacific (**Southeast Asia**)	*southeastasia*	**136 ms**	**125 ms**	**117 ms**	**194 ms**	194 ms	152 ms	144 ms	119 ms	125 ms	117 ms	122 ms	153 ms	120 ms	128 ms
25	Australia (**Australia Central**)	*australiacentral*	**41 ms**	**41 ms**	**32 ms**	**58 ms**	42 ms	58 ms	33 ms	49 ms	36 ms	34 ms	42 ms	41 ms	37 ms	50 ms
26	Australia (**Australia Central 2**)	*australiacentral2*	**37 ms**	**36 ms**	**31 ms**	**43 ms**	40 ms	33 ms	34 ms	31 ms	36 ms	34 ms	43 ms	43 ms	36 ms	41 ms
27	Australia (**Australia East**)	*australiaeast*	**33 ms**	**34 ms**	**28 ms**	**36 ms**	36 ms	30 ms	31 ms	32 ms	34 ms	28 ms	36 ms	36 ms	35 ms	34 ms
28	Australia (**Australia Southeast**)	*australiasoutheast*	**21 ms**	**20 ms**	**16 ms**	**33 ms**	24 ms	17 ms	19 ms	16 ms	20 ms	17 ms	33 ms	26 ms	23 ms	21 ms
29	India (**Central India**)	*centralindia*	**152 ms**	**152 ms**	**148 ms**	**156 ms**	156 ms	148 ms	156 ms	148 ms	150 ms	150 ms	152 ms	155 ms	153 ms	153 ms
30	India (**South India**)	*southindia*	**142 ms**	**143 ms**	**134 ms**	**153 ms**	135 ms	135 ms	144 ms	134 ms	140 ms	153 ms	145 ms	143 ms	153 ms	143 ms
31	India (**West India**)	*westindia*	**182 ms**	**177 ms**	**160 ms**	**211 ms**	173 ms	160 ms	182 ms	174 ms	181 ms	174 ms	205 ms	188 ms	211 ms	177 ms
32	Japan (**Japan East**)	*japaneast*	**167 ms**	**160 ms**	**142 ms**	**219 ms**	157 ms	152 ms	160 ms	157 ms	219 ms	142 ms	160 ms	163 ms	188 ms	148 ms
33	Japan (**Japan West**)	*japanwest*	**155 ms**	**154 ms**	**146 ms**	**167 ms**	162 ms	161 ms	149 ms	146 ms	154 ms	147 ms	157 ms	165 ms	167 ms	153 ms

Figure 5.6 - Ping of Azure's Server WorldWide (Cloud Ping Test 2020?)

Inserting 33 rows result in an execution time of up to 3s:

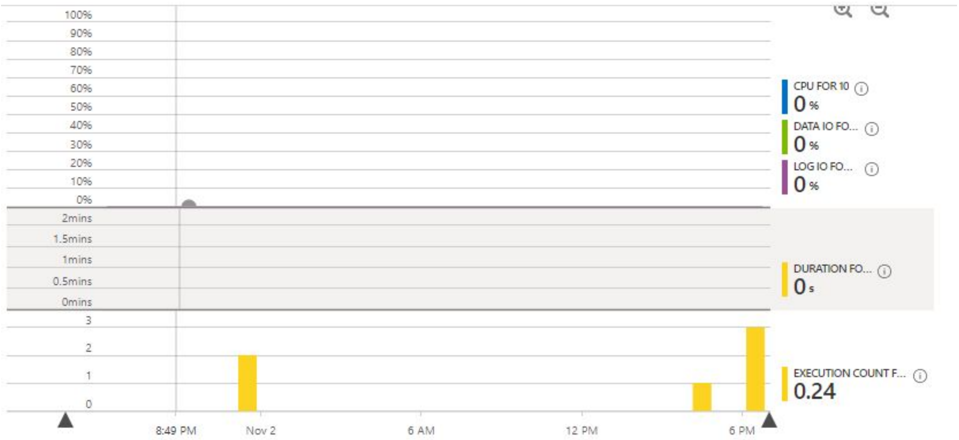
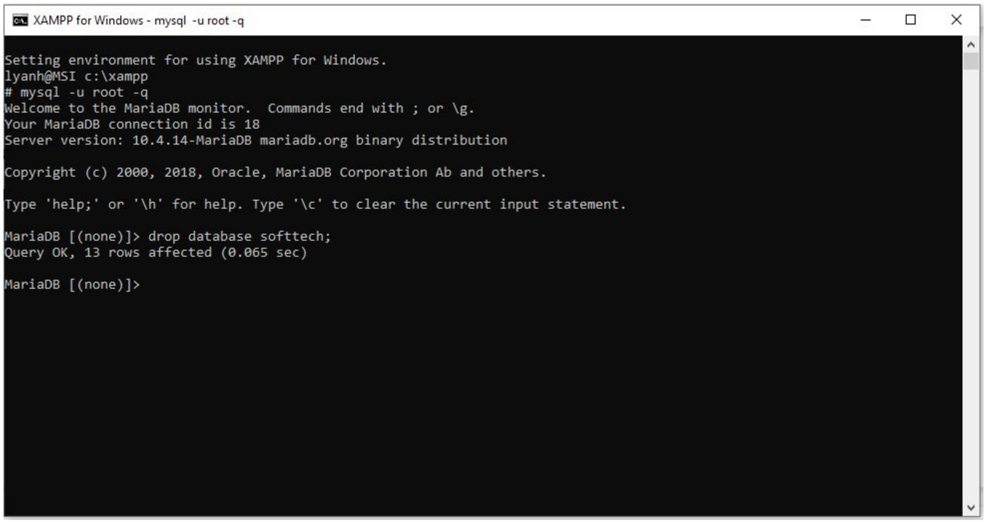


Figure 5.7 - 33 rows Inserted - Obtained from Microsoft Azure Portal

B. Tools and Support

MariaDB is an "interface of the MySQL relational database management" (Tutorialspoint 2020?). The developer of MySQL developed MariaDB. MariaDB itself does not have an IDE. Most database managers use the terminal to access MariaDB (Figure 5.6). The tools given by MariaDB are limited. In order for Softtech's IT department to be able to use a local database like MariaDB, Softtech must have a team of data managers that are proficient with the use of accessing the database with the terminal.



```
Setting environment for using XAMPP for Windows.
lyanh@MSI c:\xampp
# mysql -u root -q
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.4.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> drop database softtech;
Query OK, 13 rows affected (0.065 sec)

MariaDB [(none)]>
```

Figure 5.8 - Terminal Accessing MariaDB - Obtained from XAMPP's Shell

Alternative tools are available. Many companies decide to develop their database management tools. However, this only works if the companies are large and have enough resources (time and money) to spare. Many traditional-size companies adopt a third-party tool. MySQL Workbench is one of the tools used by many companies as a tool to ease data management operations.

MySQL Workbench "provides data modelling and SQL development tools for server configuration, user administration, backup, and much more" (MySQL 2020?). Figure representation of data is found to be useful in the data management aspect.

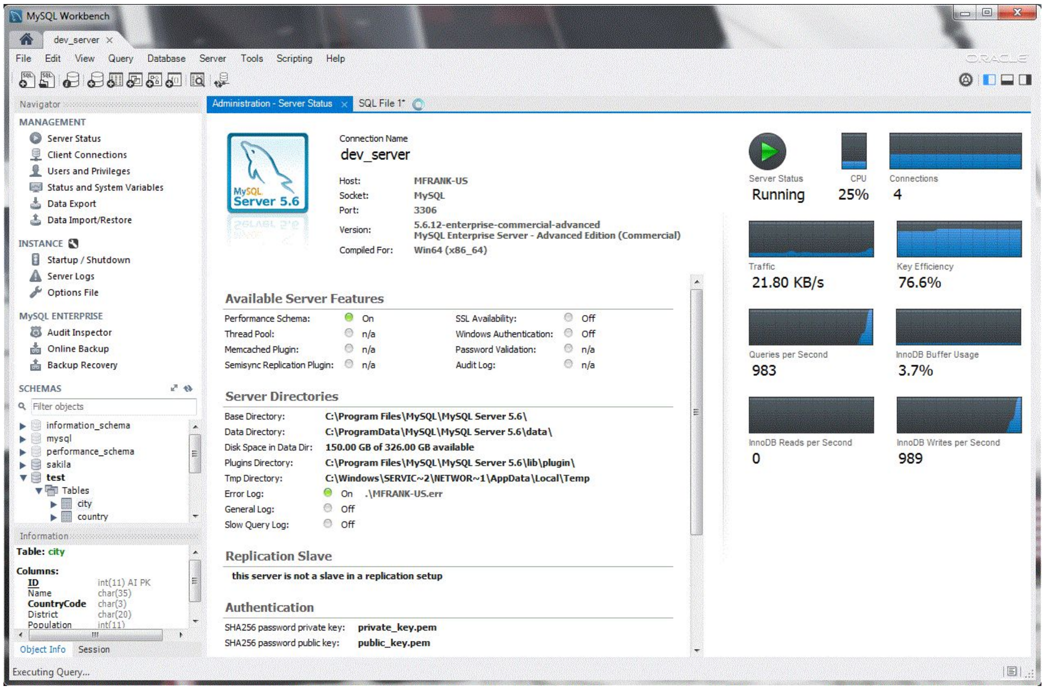


Table: city

Columns:

```
ID           int(11) AI PK
Name         char(35)
CountryCode  char(3)
District     char(20)
Population   int(11)
```

Connection Name
dev_server

Host: MFRANK-US
Socket: MySQL
Port: 3306
Version: 5.6.12-enterprise-commercial-advanced
MySQL Enterprise Server - Advanced Edition (Commercial)
Compiled For: Win64 (x86_64)

Available Server Features

Performance Schema:	● On	SSL Availability:	○ Off
Thread Pool:	○ n/a	Windows Authentication:	○ Off
Memcached Plugin:	○ n/a	Password Validation:	○ n/a
Semisync Replication Plugin:	○ n/a	Audit Log:	○ n/a

Server Directories

Base Directory:	C:\Program Files\MySQL\MySQL Server 5.6\
Data Directory:	C:\ProgramData\MySQL\MySQL Server 5.6\data\
Disk Space in Data Dir:	150.00 GB of 326.00 GB available
Plugins Directory:	C:\Program Files\MySQL\MySQL Server 5.6\lib\plugin\
Tmp Directory:	C:\Windows\SERVIC~2\NETWOR~1\AppData\Local\Temp
Error Log:	● On .\MFRANK-US.err
General Log:	○ Off
Slow Query Log:	○ Off

Replication Slave

this server is not a slave in a replication setup

Authentication

SHA256 password private key: private_key.pem
SHA256 password public key: public_key.pem

Server Status: Running
CPU: 25%
Connections: 4
Traffic: 21.80 KB/s
Key Efficiency: 76.6%
Queries per Second: 983
InnoDB Buffer Usage: 3.7%
InnoDB Reads per Second: 0
InnoDB Writes per Second: 989

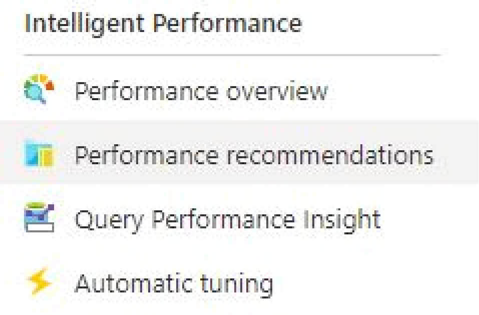
Executing Query...

Figure 5.9 - MySQL Workbench Home Page (MySQL 2020?)

Microsoft Azure provides nothing less. Even with the over-saturated amount of products provided in Azure, Azure's SQL Database provides a decent number of tools that allow for convenience for data management. Being backed by a giant company, Azure products go through constant improvement.

A study conducted by Dr Paulo Jorge Passos da Costa and Dr Antonio Miguel Rosado da Cruz in 2012 in regards to a paper called "Migration to Window Azure - Analysis and Comparison", shows the tools provided by Azure in the start of the launch period. SQL Azure database supports storing data from both on-premises applications and cloud-based applications (Jorge Passos da Costa & Miguel Rosado da Cruz 2012). It also had tools to "show reporting and business features."

Ever since then, Azure has continued to improve and expand its tools and support in SQL Azure Database. Azure provides a query editor tool that allows users to test queries rather than having to test the queries on Microsoft SQL Server. Azure also provides a range of tools to show queries and database performance (Figure 5.10).

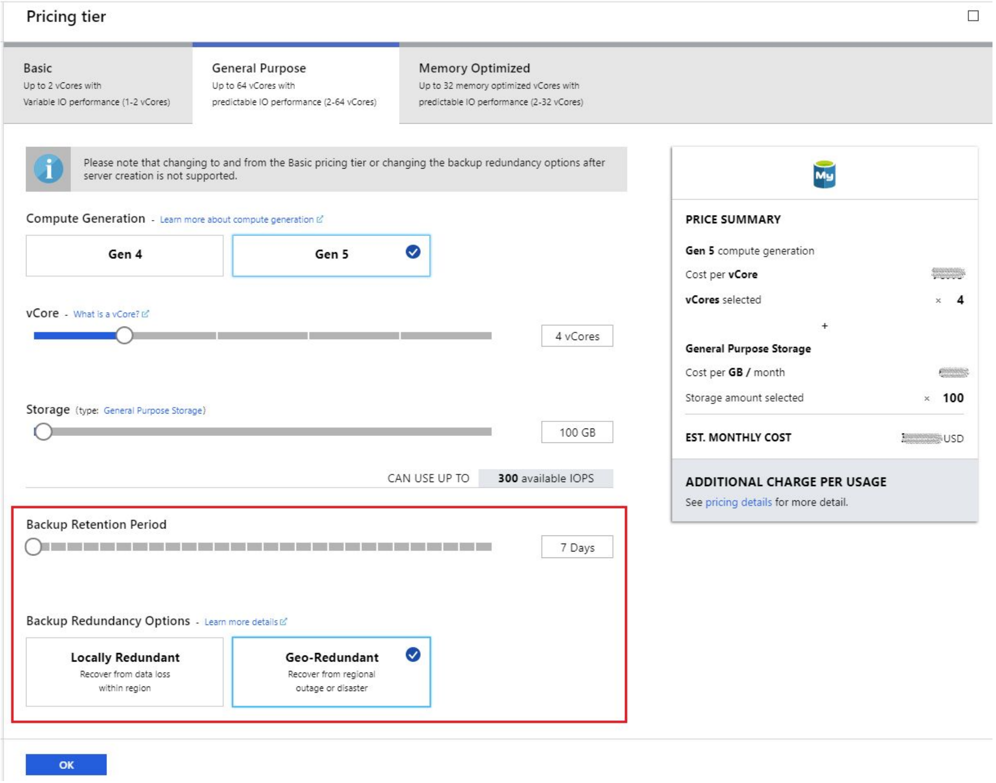


Intelligent Performance

- Performance overview
- Performance recommendations
- Query Performance Insight
- Automatic tuning

Figure 5.10 - Azure's Performance Tool - Obtained from Microsoft Azure Portal

Furthermore, According to Microsoft, Azure also backs up automatically to enable the restore feature (Microsoft Doc. c. 2020). Backups can be set in the Azure portal to be done periodically (Figure 5.11).



Pricing tier

Basic
Up to 2 vCores with
Variable IO performance (1-2 vCores)

General Purpose
Up to 64 vCores with
predictable IO performance (2-64 vCores)

Memory Optimized
Up to 32 memory optimized vCores with
predictable IO performance (2-32 vCores)

Please note that changing to and from the Basic pricing tier or changing the backup redundancy options after server creation is not supported.

Compute Generation - Learn more about compute generation

□ Gen 4 ☑ Gen 5

vCore - What is a vCore?

4 vCores

Storage (type: General Purpose Storage)

100 GB

CAN USE UP TO **300** available IOPS

Backup Retention Period

7 Days

Backup Redundancy Options - Learn more details

□ Locally Redundant
Recover from data loss within region

☑ Geo-Redundant
Recover from regional outage or disaster

OK

PRICE SUMMARY

Gen 5 compute generation	
Cost per **vCore**	[redacted]
vCores selected	× **4**
+	
General Purpose Storage	
Cost per **GB /** month	[redacted]
Storage amount selected	× **100**
EST. MONTHLY COST	[redacted] USD

ADDITIONAL CHARGE PER USAGE

See pricing details for more detail.

Figure 5.11 - Azure's Periodical Backup - Obtained from Microsoft Azure Portal

More than just having a single copy of the database like in a local database, Azure also allows the option to "restore to a different geographical region." (Microsoft Doc. c. 2020). Azure also takes security reasons seriously. Security tools are provided and easily accessible in the portal. If "Auditing" is turned on, Azure will automatically track any activity that goes on, and report those into the log section of the portal. Azure also comes with an advanced security setting powered by Azure Defender that analyses the database's vulnerability and threats to cyber attacks. It is important to note that this is common among most giant cloud service providers who have the resources to patch up advanced security measures quicker and better than most companies. A study showed that "most organisations take between 25 to 60 days to deploy security patches, but most CIOs admitted that it must take at least six months." Giants like Google or Microsoft, however, can "gather security intelligence billion transactions a day and apply it in real-time" (Sultan 2011). Azure's data are also encrypted and patched between transactions.

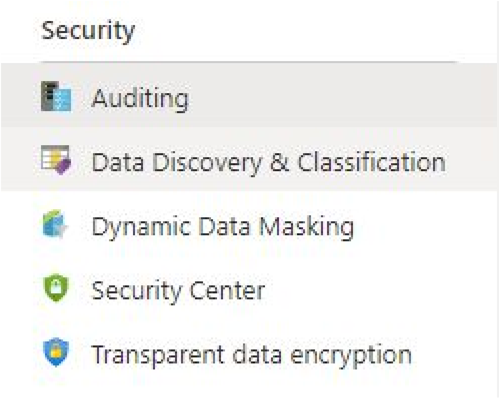
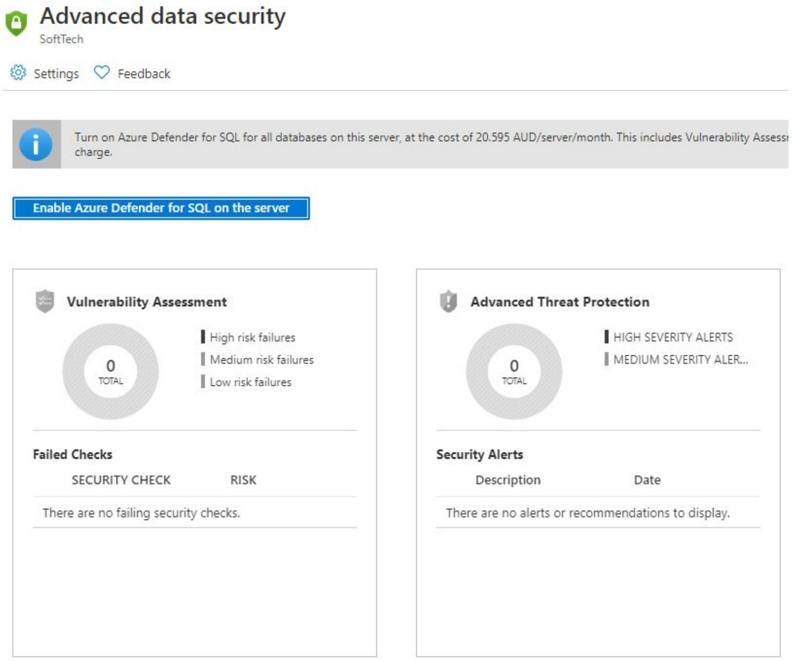


Figure 5.12 - Azure's Security Tools - Obtained from Microsoft Azure Portal



Turn on Azure Defender for SQL for all databases on this server, at the cost of 20.595 AUD/server/month. This includes Vulnerability Assess
charge.

There are no failing security checks.

There are no alerts or recommendations to display.

Figure 5.13 - Azure Defender - Obtained from Microsoft Azure Portal

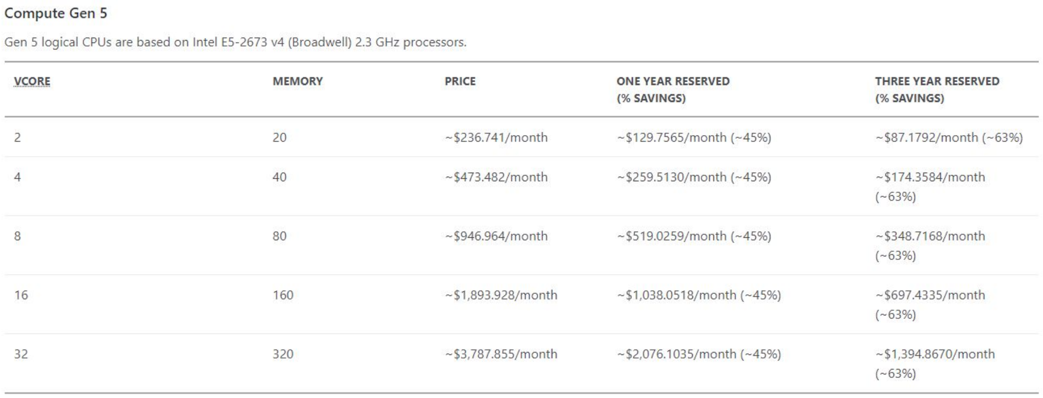
C. Cost

Although not technical, costs are aspects that are considered in each corporation before making any changes or big decisions,

When it comes to the traditional local database, the cost of hosting infrastructure and software can be high. The case study in Italy shows that most startup ideas are held back due to the cost of the ICT department infrastructure and the cost to maintain it (Ferri et al. 2017). AlphaGame's founder mentioned that the ideas had been brought up among the co-founders years before the actual launch year. However, due to the tremendous cost of renting a place, buying the servers, and hiring professionals to take care of those servers, executing the idea at that moment will result in a minimal profit margin.

Cloud-based databases provide a solution to such issues. The founder of AlphaGame mentioned that in a startup company, low investments in infrastructure investing provide more flexibility and agility so the focus of the company can be shifted to the core business (Ferri et al., 2017). AlphaGame chose to launch the company on Azure, providing them with many flexibilities and saving lots of resources. "Costs were constantly monitored provided by Microsoft Azure," the co-founder mentioned. Azure outsources all the activities that generally take lots of time and resources.

Azure also comes with a free trial period of 12 months. Small startup companies can use this opportunity to test out Azure's services. Although prices have a wide range among different products, the average price of a product available on Azure is around 100$ AUD per month (Figure 5.14). The cost of storage and backup is also less than 1$ AUD per month (Figure 5.15).

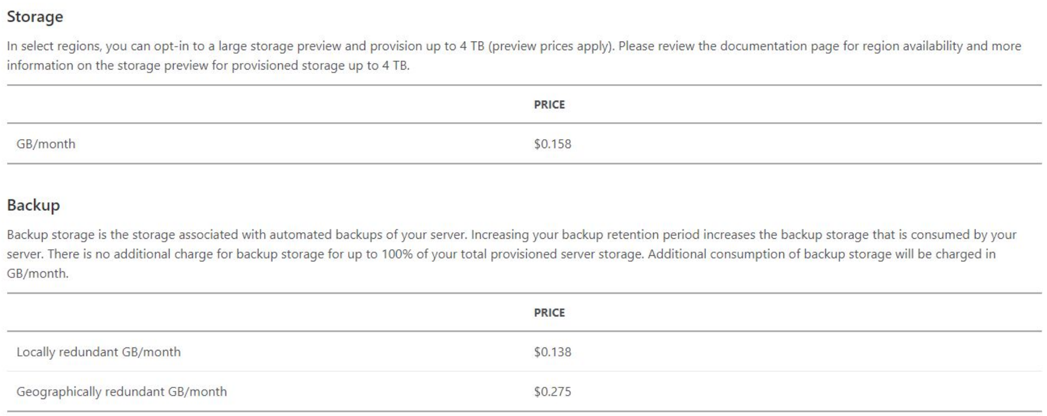


Compute Gen 5

Gen 5 logical CPUs are based on Intel E5-2673 v4 (Broadwell) 2.3 GHz processors.

VCORE	MEMORY	PRICE	ONE YEAR RESERVED (% SAVINGS)	THREE YEAR RESERVED (% SAVINGS)
2	20	~$236.741/month	~$129.7565/month (~45%)	~$87.1792/month (~63%)
4	40	~$473.482/month	~$259.5130/month (~45%)	~$174.3584/month (~63%)
8	80	~$946.964/month	~$519.0259/month (~45%)	~$348.7168/month (~63%)
16	160	~$1,893.928/month	~$1,038.0518/month (~45%)	~$697.4335/month (~63%)
32	320	~$3,787.855/month	~$2,076.1035/month (~45%)	~$1,394.8670/month (~63%)

Figure 5.14 - Database Price (Microsoft Azure 2020?)



Storage

In select regions, you can opt-in to a large storage preview and provision up to 4 TB (preview prices apply). Please review the documentation page for region availability and more information on the storage preview for provisioned storage up to 4 TB.

	PRICE
GB/month	$0.158

Backup

Backup storage is the storage associated with automated backups of your server. Increasing your backup retention period increases the backup storage that is consumed by your server. There is no additional charge for backup storage for up to 100% of your total provisioned server storage. Additional consumption of backup storage will be charged in GB/month.

	PRICE
Locally redundant GB/month	$0.138
Geographically redundant GB/month	$0.275

Figure 5.15 - Storage and Backup Cost (Microsoft Azure 2020?)

Furthermore, a cloud-based database also provides many options for upgrading. A traditional local database update requires the purchase of more servers and a hefty amount of operation. According to Sultan, the cloud's main benefit is its efficiency in matching the supply to demand the use of cloud services (Sultan 2011). With Azure pay-as-you-go payment model, the cost of hosting an online database system is as low as any traditional database maintenance can get. Based on the same paper, Sultan mentioned that the costs to a company's cloud database system are "roughly 20% of traditional hosting" (Sultan 2011).

VI. "Data Migration" Process (From Local to Global)

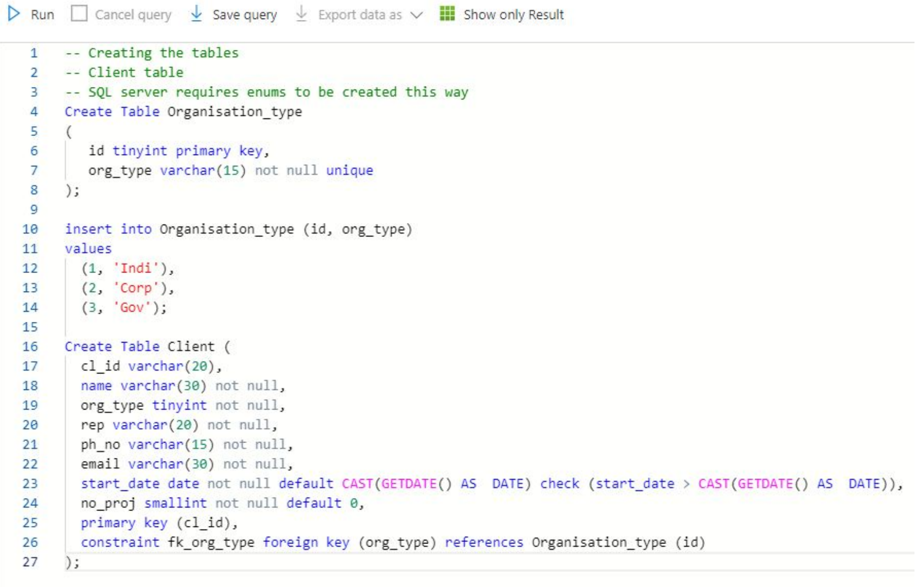
Data Migration is a topic brought upon the startup community when companies start migrating to the cloud. Although many experts suggested startups to start with the use of cloud (Ferri et al., 2017), there is a higher demand for startups with the traditional database set up to convert to a cloud-based database.

Most large companies contain large amounts of data. Data migration processes in large companies are more complex and require many steps to make sure that the data are secured. However, due to the nature of the research being applied to a small startup that can start by cloud, the migration process will rely more on manual migration.

An important aspect to look into is the type of database MariaDB and Azure implement. MariaDB uses MySQL as its database management system. Azure uses its version of SQL Server Database Management system.

MySQL and SQL Server are both relational database systems that are not very different structurally or in syntax. This allows the "migration" process to be more straightforward than most cases. Most of the basic DLLs involving creating tables and insertions of data are similar. Most of the structural changes that have to be done are in regards to the flexibility of each language.

Azure SQL Server relies on manual query designs. An enumeration in MySQL does not exist in SQL Server. To be able to create a list of options in SQL Server's attribute, a new table must be created. Figure 6.1 shows the creation of a table called Organisation_type that contains the enumeration options. The Table Client uses that information in the Client table to refer to the type of organisation. This ensures a normalised database.



```
1   -- Creating the tables
2   -- Client table
3   -- SQL server requires enums to be created this way
4   Create Table Organisation_type
5   (
6      id tinyint primary key,
7      org_type varchar(15) not null unique
8   );
9
10  insert into Organisation_type (id, org_type)
11  values
12    (1, 'Indi'),
13    (2, 'Corp'),
14    (3, 'Gov');
15
16  Create Table Client (
17    cl_id varchar(20),
18    name varchar(30) not null,
19    org_type tinyint not null,
20    rep varchar(20) not null,
21    ph_no varchar(15) not null,
22    email varchar(30) not null,
23    start_date date not null default CAST(GETDATE() AS  DATE) check (start_date > CAST(GETDATE() AS  DATE)),
24    no_proj smallint not null default 0,
25    primary key (cl_id),
26    constraint fk_org_type foreign key (org_type) references Organisation_type (id)
27  );
```

Figure 6.1 - Enumeration Creation in Azure SQL Server - Obtained from Microsoft Azure Portal

SQL Server also does not block unpredictable constants from being a part of clauses. This means all the triggers set up in MariaDB can be eliminated and added as a check clause in the table creation itself. Shown in Figure 6.1, Line 23 shows that the check_start_date trigger can be done as a standard check clause.

VII. Implementation

This section implements the knowledge of data migration to migrate data from a startup company - Softtech - MariaDB Database to Microsoft Azure's SQL Database. This aims to implement a simple database and migrate the data with simple insert queries.

[The feature and data of the database are featured in the previous tasks attached with this research report].

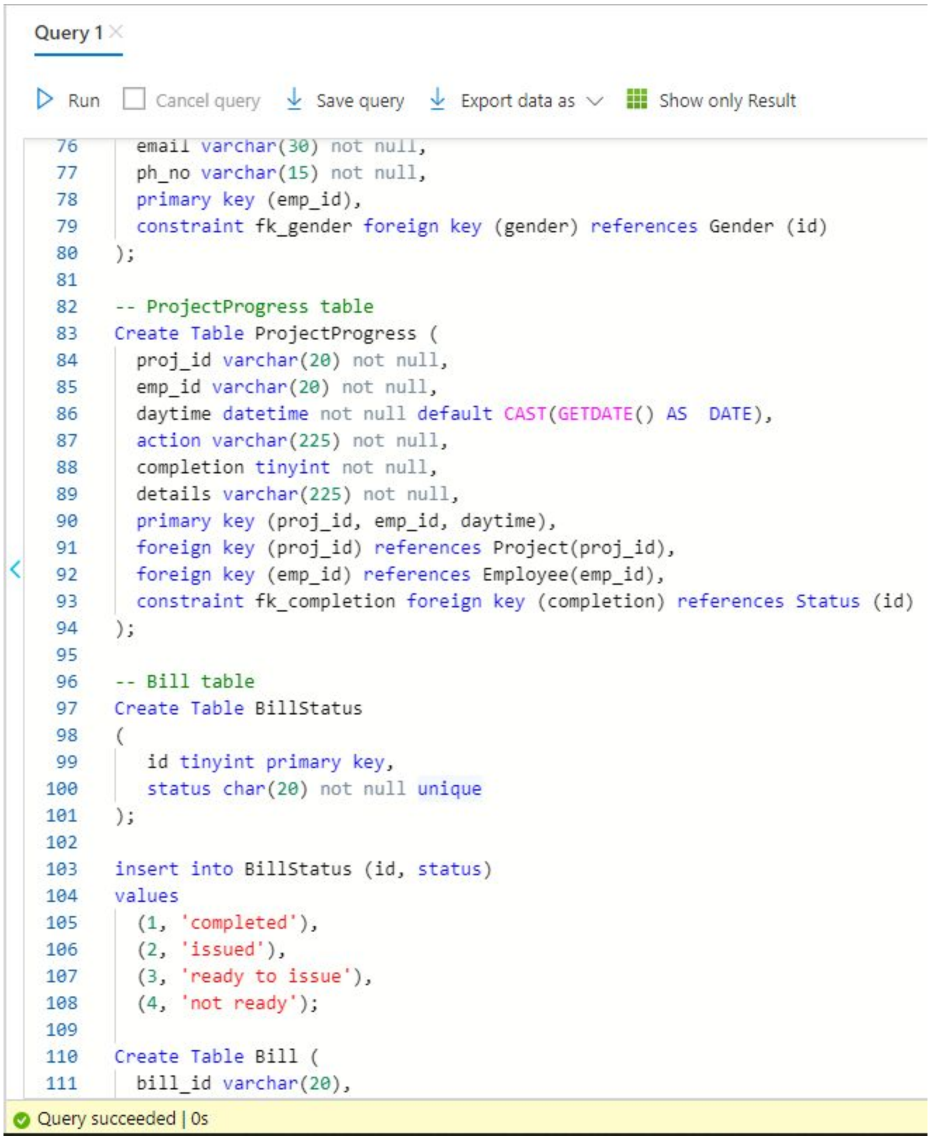
Table Creation queries:

```
-- Creating the tables
-- Client table
-- SQL server requires enums to be created this way
Create Table Organisation_type
(
   id tinyint primary key,
   org_type varchar(15) not null unique
);

insert into Organisation_type (id, org_type)
values
  (1, 'Indi'),
  (2, 'Corp'),
  (3, 'Gov');

Create Table Client (
  cl_id varchar(20),
  name varchar(30) not null,
  org_type tinyint not null,
  rep varchar(20) not null,
  ph_no varchar(15) not null,
  email varchar(30) not null,
  start_date date not null default CAST(GETDATE() AS  DATE) check
(start_date < CAST(GETDATE() AS  DATE)),
  no_proj smallint not null default 0,
  primary key (cl_id),
  constraint fk_org_type foreign key (org_type) references
Organisation_type (id)
);

-- Project table
Create Table Status
(
   id tinyint primary key,
   status varchar(15) not null unique
);

insert into Status (id, status)
values
  (1, 'completed'),
  (2, 'on-progress'),
  (3, 'not started');

Create Table Project (
  proj_id varchar(20) not null,
  cl_id varchar(20) not null,
  name varchar(50) not null,
  start_date date not null default CAST(GETDATE() AS  DATE) check
(start_date < CAST(GETDATE() AS  DATE)),
  end_date date,
  status tinyint not null default 3,
  comments varchar(225),
  primary key (proj_id),
  foreign key (cl_id) references Client(cl_id),
  constraint fk_status foreign key (status) references Status (id)
);

-- Employee table
Create Table Gender
(
   id tinyint primary key,
   gender char(1) not null unique
);
```

```
insert into Gender (id, gender)
values
  (1, 'M'),
  (2, 'F'),
  (3, 'U');

Create Table Employee (
  emp_id varchar(20) not null,
  fname varchar(20) not null,
  lname varchar(20) not null,
  gender tinyint not null,
  date_of_birth date not null,
  start_date date not null check (start_date < CAST(GETDATE() AS  DATE)),
  address varchar(150) not null,
  email varchar(30) not null,
  ph_no varchar(15) not null,
  primary key (emp_id),
  constraint fk_gender foreign key (gender) references Gender (id)
);

-- ProjectProgress table
Create Table ProjectProgress (
  proj_id varchar(20) not null,
  emp_id varchar(20) not null,
  daytime datetime not null default CAST(GETDATE() AS  DATE),
  action varchar(225) not null,
  completion tinyint not null,
  details varchar(225) not null,
  primary key (proj_id, emp_id, daytime),
  foreign key (proj_id) references Project(proj_id),
  foreign key (emp_id) references Employee(emp_id),
  constraint fk_completion foreign key (completion) references Status
(id)
);

-- Bill table
Create Table BillStatus
(
   id tinyint primary key,
   status char(20) not null unique
);

insert into BillStatus (id, status)
values
  (1, 'completed'),
  (2, 'issued'),
  (3, 'ready to issue'),
  (4, 'not ready');

Create Table Bill (
  bill_id varchar(20),
  proj_id varchar(20) not null,
  daytime datetime not null,
  work varchar(50) not null,
  hours int not null check (hours >= 0),
  hourly_rate decimal(7, 2) not null check (hourly_rate >= 0),
  total as (hours * hourly_rate),
  status tinyint not null,
  primary key (bill_id),
  foreign key (proj_id) references Project(proj_id)
);

Create Table Salary (
  emp_id varchar(20) not null,
  date_of_record date not null,
  base_salary decimal(6, 2) not null check (base_salary > 0),
  hours smallint not null check (hours > 0),
  bonus decimal(5, 2) not null default 0.00 check (bonus >= 0),
  total as (base_salary * hours + bonus),
  primary key (emp_id, date_of_record),
  foreign key (emp_id) references Employee(emp_id)
);

-- WorksOn table
Create Table WorksOn (
  emp_id varchar(20) not null,
  proj_id varchar(20) not null,
  role varchar(25) not null,
  comments varchar(225),
  primary key (proj_id, emp_id),
  foreign key (proj_id) references Project(proj_id),
  foreign key (emp_id) references Employee(emp_id)
);

-- AnnualBillReport table
Create Table AnnualBillReport (
  cl_id varchar(20) not null,
  year smallint not null,
  income decimal(13, 2) not null,
  no_proj smallint not null default 0 check (no_proj > 0),
  primary key (cl_id, year),
  foreign key (cl_id) references Client(cl_id)
);

-- ProjectSize table
Create Table Size
(
   id tinyint primary key,
   size char(1) not null unique
);

insert into Size (id, size)
values
  (1, 'S'),
  (2, 'M'),
  (3, 'L');

Create Table Importance
(
   id tinyint primary key,
   importance char(10) not null unique
);
```

```
Insert into Importance (id, importance)
values
  (1, 'low'),
  (2, 'medium'),
  (3, 'high');

Create Table ProjectSize (
  proj_id varchar(20) not null,
  size tinyint not null,
  importance tinyint not null,
  no_dev tinyint not null check (no_dev > 0),
  comments varchar(225),
  primary key (proj_id),
  foreign key (proj_id) references Project(proj_id),
  constraint fk_size foreign key (size) references Size(id),
  constraint fk_importance foreign key (importance) references
Importance(id)
);
```

Query result:



```
  email varchar(30) not null,
  ph_no varchar(15) not null,
  primary key (emp_id),
  constraint fk_gender foreign key (gender) references Gender (id)
);

-- ProjectProgress table
Create Table ProjectProgress (
  proj_id varchar(20) not null,
  emp_id varchar(20) not null,
  daytime datetime not null default CAST(GETDATE() AS  DATE),
  action varchar(225) not null,
  completion tinyint not null,
  details varchar(225) not null,
  primary key (proj_id, emp_id, daytime),
  foreign key (proj_id) references Project(proj_id),
  foreign key (emp_id) references Employee(emp_id),
  constraint fk_completion foreign key (completion) references Status (id)
);

-- Bill table
Create Table BillStatus
(
   id tinyint primary key,
   status char(20) not null unique
);

insert into BillStatus (id, status)
values
  (1, 'completed'),
  (2, 'issued'),
  (3, 'ready to issue'),
  (4, 'not ready');

Create Table Bill (
  bill_id varchar(20),
```

Query succeeded | 0s

Figure 7.1 - Table Creation

Insertion Queries:

```
-- Adding records into the database (minimum 25 records & 5 JOINs)
-- Insert into Employees
Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ('EMP001', 'Lyan', 'Heng', 1, '1995-04-19', '2019-03-19', '32
Jones Road WESTLAKE, Queensland(QLD), 4074', 'lyanh@softtech.com',
'+61403193849');
```

```
Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ('EMP002', 'John', 'Wit', 1, '1999-12-07', '2019-04-03', '95 Zipfs
Road WEST IPSWICH, Queensland(QLD), 4305', 'johnw@softtech.com',
'+61401820391');

Insert Into Employee(emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ('EMP003', 'Jim', 'Tim', 1, '1997-10-04', '2019-05-01', '42 Sale
Street LIDSTER, New South Wales(NSW), 2800', 'jimt@softtech.com',
'+61419203920');

Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ('EMP004', 'Ashley', 'Nguyen', 2, '2001-02-01', '2019-07-29', '15
Tennyson Road LAKEMBA, New South Wales(NSW), 2195',
'ashleyn@softtech.com', '+61401923019');

Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ('EMP005', 'Natalie', 'Wilkinson', 3, '2001-02-01', '2020-01-09',
'41 Shirley Street HOLMVIEW, Queensland(QLD), 4207',
'nataliew@softtech.com', '+61412938210');

-- Insert into Client
Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date,
no_proj)
Values ('CLI001', 'Jay Tim Technology Co. Ltd', 2, 'May Dimson',
'0347538293', 'buspartner@jaytim.com', '2019-02-05', 1);

Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date,
no_proj)
Values ('CLI002', 'MacroHard Technology Co. Ltd', 2, 'Tom Polland',
'03419230291', 'contact@mt.com', '2019-12-03', 2);

Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date,
no_proj)
Values ('CLI003', 'Australian Taxation Office', 3, 'Jim Carter',
'03029103192', 'jc@ato.au.gov', '2019-02-05', 1);

-- Insert into Project
Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
Values ('PRJ001', 'CLI001', 'Jay Tims Database System', '2019-02-05',
'2019-08-04', 1, 'Create a database system for Jay Tim.');

Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
```

```
Values ('PRJ002', 'CLI002', 'Microcontroller Design', '2019-05-02',
'2019-08-01', 1, 'Microcontroller for Door System');

Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
Values ('PRJ003', 'CLI003', 'Website Design', '2019-07-02', '2019-08-10',
1, 'Small section of website for ATO');

Insert Into Project (proj_id, cl_id, name, start_date, status, comments)
Values ('PRJ004', 'CLI002', 'Door Safety System', '2019-08-02', 2, 'A
sequel after the door micro-design');

-- Insert into ProjectProgress
Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ001', 'EMP001', '2019-03-10 11:45:03', 'Completed Design', 1,
'After a long study of company structure, design is completed.');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ002', 'EMP003', '2019-05-02 08:34:53', 'Project started', 1,
'Contract Signed. Project proceeded right away.');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ001', 'EMP002', '2019-05-22 18:07:37', 'Database Draft
Completed', 1, 'A draft of the database completed. Proceed to testing');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ002', 'EMP003', '2019-06-01 08:12:34', 'Microcontroller
Finalisation', 2, 'Report on finalisation progress.');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ003', 'EMP001', '2019-07-02 12:30:22', 'Project started', 1,
'Contract signed. Project started to be built.');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ002', 'EMP004', '2019-08-01 08:00:00', 'Project finalised',
1, 'Project has been finalised.');

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ004', 'EMP003', '2019-08-02 08:32:12', 'Project started', 1,
'Contract signed.');
```

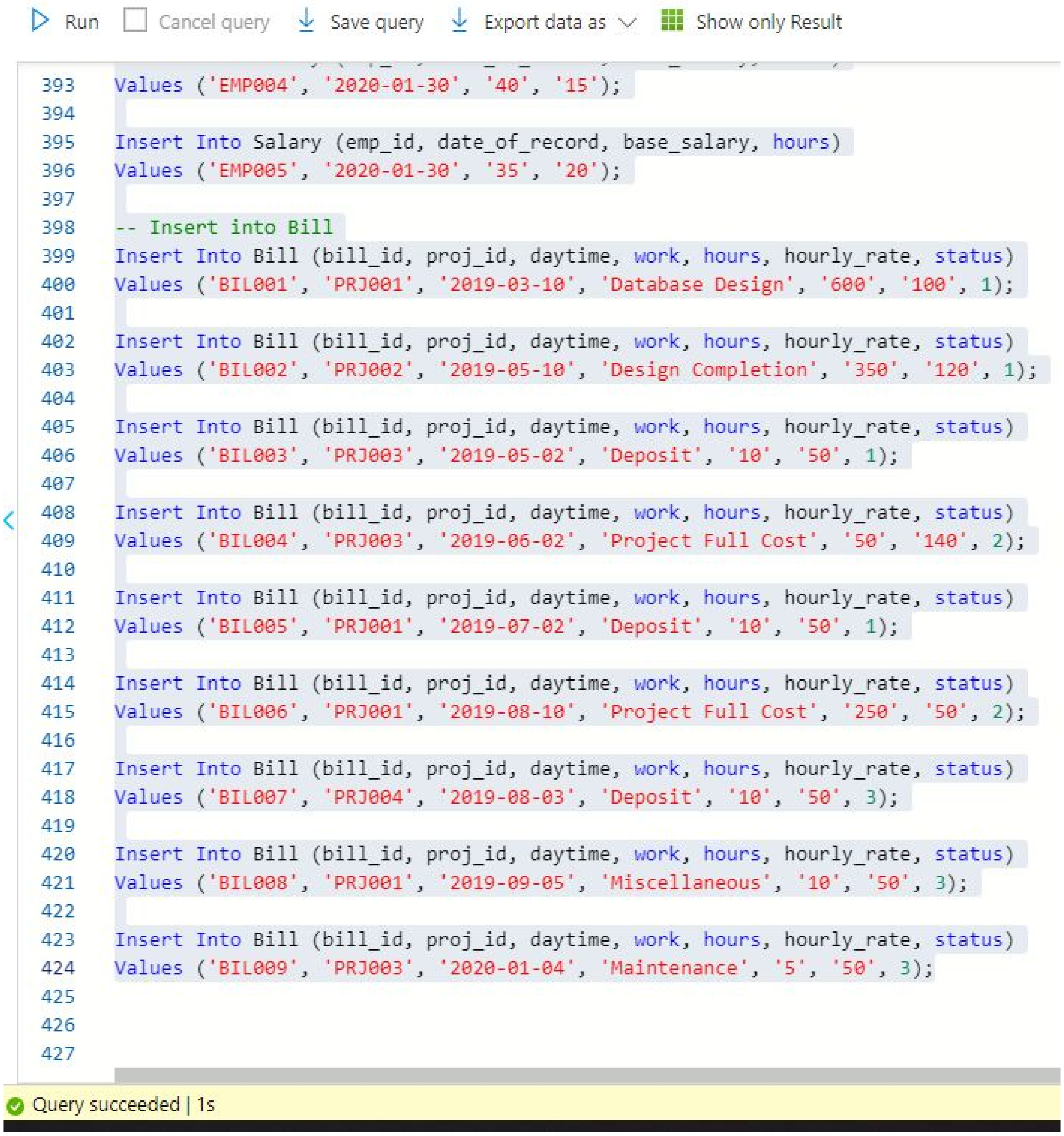
```sql
Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ('PRJ003', 'EMP001', '2019-08-10 12:30:00', 'Project completed',
1, 'Website built to satisfaction');

-- Insert into WorksOn
Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP001', 'PRJ001', 'Project Manager', 'Responsible for the
progress and completion of the project');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP002', 'PRJ001', 'Developer', 'Develop the database system');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP003', 'PRJ002', 'Project Manager', 'Responsible for the
progress and completion of the project');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP004', 'PRJ002', 'Developer', 'Microcontroller Designer');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP001', 'PRJ002', 'Tester', 'Responsible for the product being
up to par');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP001', 'PRJ003', 'Sole Developer', 'Responsible for the
project');

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ('EMP003', 'PRJ004', 'Sole Developer', 'Responsible for the
project');

-- Insert into ProjectSize
Insert Into ProjectSize (proj_id, size, importance, no_dev)
Values ('PRJ001', 2, 2, '2');

Insert Into ProjectSize (proj_id, size, importance, no_dev, comments)
Values ('PRJ002', 3, 2, '3', 'Finish in Three Months');

Insert Into ProjectSize (proj_id, size, importance, no_dev, comments)
Values ('PRJ003', 1, 2, '1', 'Government query');

Insert Into ProjectSize (proj_id, size, importance, no_dev)
Values ('PRJ004', 1, 1, '1');

-- Insert into Salary
Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP001', '2019-04-30', '62.5', '29');
```

```
Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-05-30', '62.5', '60', '10');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-05-30', '35.2', '12');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-05-30', '35', '35');

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-06-30', '62.5', '60', '10');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-06-30', '35.2', '50');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-06-30', '35', '30');

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-07-30', '62.5', '60', '20');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-07-30', '35.2', '45');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-07-30', '35', '35');

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-08-30', '62.5', '58', '5');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-08-30', '35.2', '50');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-08-30', '35', '35');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP004', '2019-08-30', '40', '5');

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ('EMP001', '2019-09-30', '62.5', '60', '10');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-09-30', '35.2', '45');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-09-30', '35', '20');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
```

```
Values ('EMP004', '2019-09-30', '40', '25');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP001', '2019-10-30', '62.5', '62');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-10-30', '35.2', '55');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-10-30', '35', '20');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP004', '2019-10-30', '40', '15');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP001', '2019-11-30', '62.5', '62');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-11-30', '35.2', '60');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-11-30', '35', '20');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP004', '2019-11-30', '40', '15');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP001', '2019-12-30', '62.5', '62');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2019-12-30', '35.2', '60');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2019-12-30', '35', '30');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP004', '2019-12-30', '40', '15');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP001', '2020-01-30', '65', '55');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP002', '2020-01-30', '40', '50');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP003', '2020-01-30', '35', '25');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP004', '2020-01-30', '40', '15');
```

```
Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP005', '2020-01-30', '35', '20');

-- Insert into Bill
Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL001', 'PRJ001', '2019-03-10', 'Database Design', '600',
'100', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL002', 'PRJ002', '2019-05-10', 'Design Completion', '350',
'120', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL003', 'PRJ003', '2019-05-02', 'Deposit', '10', '50', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL004', 'PRJ003', '2019-06-02', 'Project Full Cost', '50',
'140', 2);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL005', 'PRJ001', '2019-07-02', 'Deposit', '10', '50', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL006', 'PRJ001', '2019-08-10', 'Project Full Cost', '250',
'50', 2);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL007', 'PRJ004', '2019-08-03', 'Deposit', '10', '50', 3);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL008', 'PRJ001', '2019-09-05', 'Miscellaneous', '10', '50',
3);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ('BIL009', 'PRJ003', '2020-01-04', 'Maintenance', '5', '50', 3);
```

Queries Result:

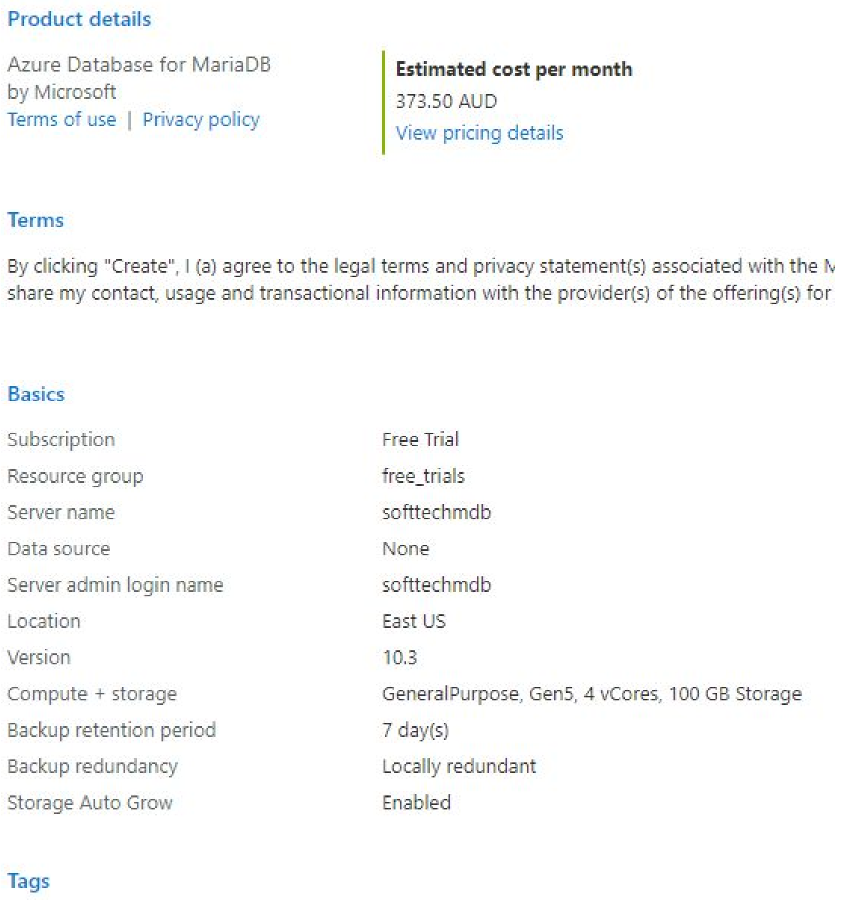


```
Values ('EMP004', '2020-01-30', '40', '15');

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ('EMP005', '2020-01-30', '35', '20');

-- Insert into Bill
Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL001', 'PRJ001', '2019-03-10', 'Database Design', '600', '100', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL002', 'PRJ002', '2019-05-10', 'Design Completion', '350', '120', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL003', 'PRJ003', '2019-05-02', 'Deposit', '10', '50', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL004', 'PRJ003', '2019-06-02', 'Project Full Cost', '50', '140', 2);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL005', 'PRJ001', '2019-07-02', 'Deposit', '10', '50', 1);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL006', 'PRJ001', '2019-08-10', 'Project Full Cost', '250', '50', 2);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL007', 'PRJ004', '2019-08-03', 'Deposit', '10', '50', 3);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL008', 'PRJ001', '2019-09-05', 'Miscellaneous', '10', '50', 3);

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status)
Values ('BIL009', 'PRJ003', '2020-01-04', 'Maintenance', '5', '50', 3);
```

Query succeeded | 1s

Figure 7.2 - Data Insertion - Obtained from Microsoft Azure Portal

It is important to note that Azure also provides the option to host a MariaDB server on the cloud. The data migration for Softtech will be smoother. The cost of the service per month is 373.50$ AUD per month. However, due to the lack of resources sponsored or available for this service (Figure 7.3), the research is not able to dive into the data migration from MariaDB into the MariaDB Azure Service.



Product details

Azure Database for MariaDB
by Microsoft
Terms of use | Privacy policy

Estimated cost per month
373.50 AUD
View pricing details

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the
share my contact, usage and transactional information with the provider(s) of the offering(s) for

Basics

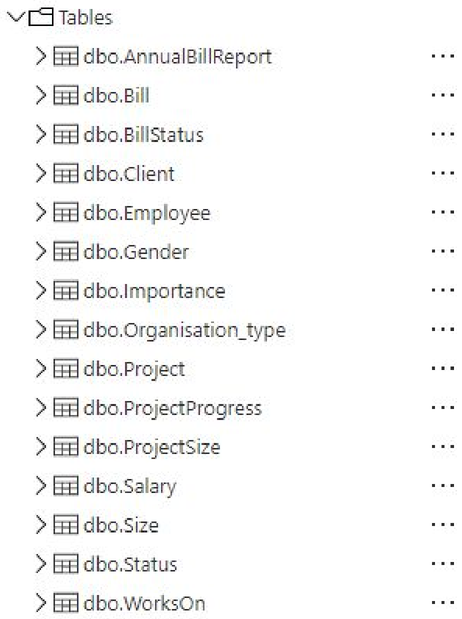
Subscription	Free Trial
Resource group	free_trials
Server name	softtechmdb
Data source	None
Server admin login name	softtechmdb
Location	East US
Version	10.3
Compute + storage	GeneralPurpose, Gen5, 4 vCores, 100 GB Storage
Backup retention period	7 day(s)
Backup redundancy	Locally redundant
Storage Auto Grow	Enabled

Tags

Figure 7.3 - Cost to Migrate to Azure MariaDB Service - Obtained from Microsoft Azure Portal

VIII. Results

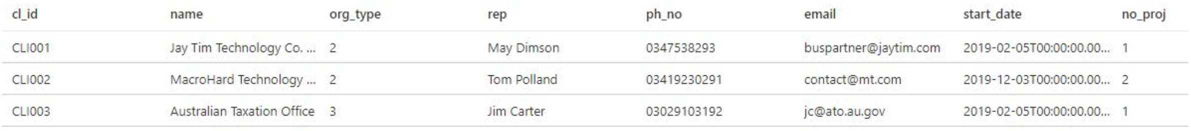
Tables created:



Tables

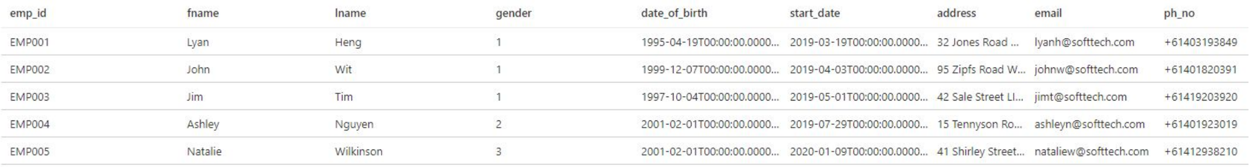
- dbo.AnnualBillReport
- dbo.Bill
- dbo.BillStatus
- dbo.Client
- dbo.Employee
- dbo.Gender
- dbo.Importance
- dbo.Organisation_type
- dbo.Project
- dbo.ProjectProgress
- dbo.ProjectSize
- dbo.Salary
- dbo.Size
- dbo.Status
- dbo.WorksOn

Figure 8.1 - Tables Created - Obtained from Microsoft Azure Portal



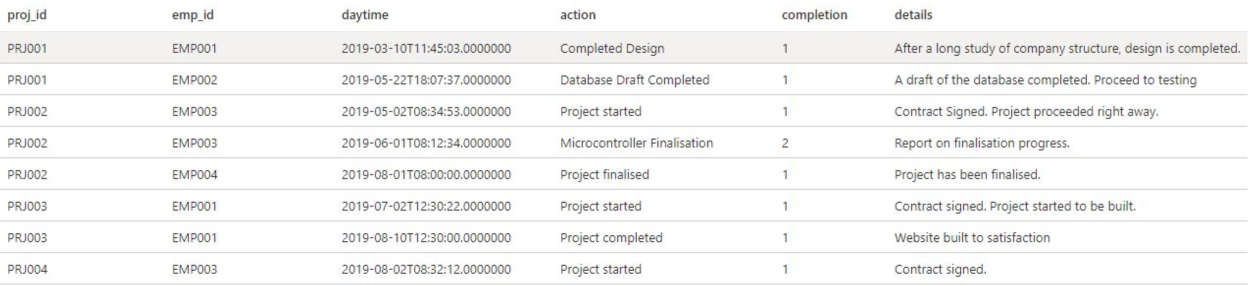
cl_id	name	org_type	rep	ph_no	email	start_date	no_proj
CLI001	Jay Tim Technology Co. ...	2	May Dimson	0347538293	buspartner@jaytim.com	2019-02-05T00:00:00.00...	1
CLI002	MacroHard Technology ...	2	Tom Polland	03419230291	contact@mt.com	2019-12-03T00:00:00.00...	2
CLI003	Australian Taxation Office	3	Jim Carter	03029103192	jc@ato.au.gov	2019-02-05T00:00:00.00...	1

Figure 8.2 - Client Table - Obtained from Microsoft Azure Portal



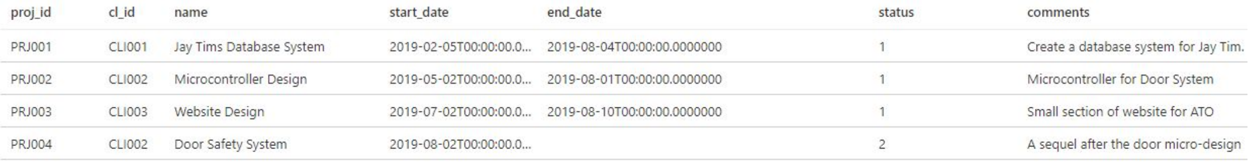
emp_id	fname	lname	gender	date_of_birth	start_date	address	email	ph_no
EMP001	Lyan	Heng	1	1995-04-19T00:00:00.0000...	2019-03-19T00:00:00.0000...	32 Jones Road ...	lyanh@softtech.com	+61403193849
EMP002	John	Wit	1	1999-12-07T00:00:00.0000...	2019-04-03T00:00:00.0000...	95 Zipfs Road W...	johnw@softtech.com	+61401820391
EMP003	Jim	Tim	1	1997-10-04T00:00:00.0000...	2019-05-01T00:00:00.0000...	42 Sale Street Ll...	jimt@softtech.com	+61419203920
EMP004	Ashley	Nguyen	2	2001-02-01T00:00:00.0000...	2019-07-29T00:00:00.0000...	15 Tennyson Ro...	ashleyn@softtech.com	+61401923019
EMP005	Natalie	Wilkinson	3	2001-02-01T00:00:00.0000...	2020-01-09T00:00:00.0000...	41 Shirley Street...	nataliew@softtech.com	+61412938210

Figure 8.3 - Employee Table - Obtained from Microsoft Azure Portal



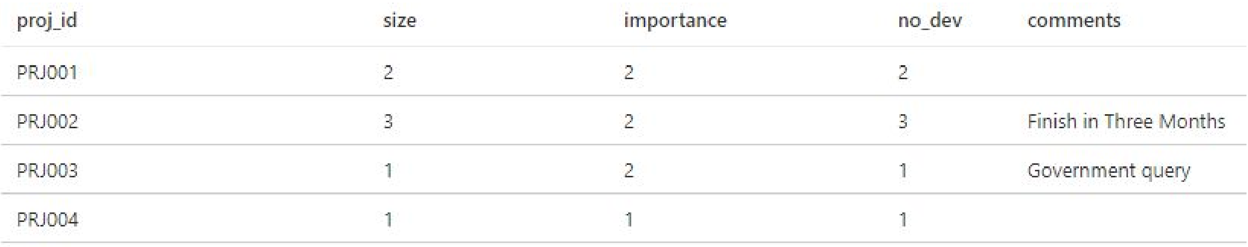
proj_id	emp_id	daytime	action	completion	details
PRJ001	EMP001	2019-03-10T11:45:03.0000000	Completed Design	1	After a long study of company structure, design is completed.
PRJ001	EMP002	2019-05-22T18:07:37.0000000	Database Draft Completed	1	A draft of the database completed. Proceed to testing
PRJ002	EMP003	2019-05-02T08:34:53.0000000	Project started	1	Contract Signed. Project proceeded right away.
PRJ002	EMP003	2019-06-01T08:12:34.0000000	Microcontroller Finalisation	2	Report on finalisation progress.
PRJ002	EMP004	2019-08-01T08:00:00.0000000	Project finalised	1	Project has been finalised.
PRJ003	EMP001	2019-07-02T12:30:22.0000000	Project started	1	Contract signed. Project started to be built.
PRJ003	EMP001	2019-08-10T12:30:00.0000000	Project completed	1	Website built to satisfaction
PRJ004	EMP003	2019-08-02T08:32:12.0000000	Project started	1	Contract signed.

Figure 8.4 - ProjectProgress Table - Obtained from Microsoft Azure Portal



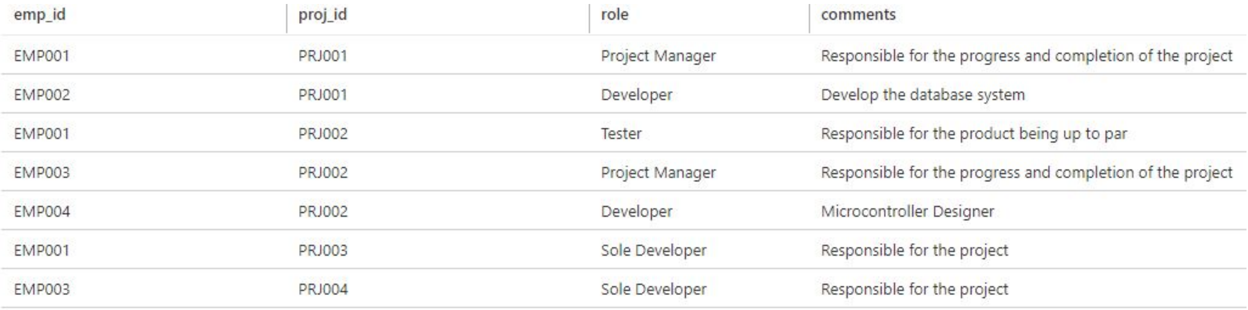
proj_id	cl_id	name	start_date	end_date	status	comments
PRJ001	CLI001	Jay Tims Database System	2019-02-05T00:00:00.0...	2019-08-04T00:00:00.0000000	1	Create a database system for Jay Tim.
PRJ002	CLI002	Microcontroller Design	2019-05-02T00:00:00.0...	2019-08-01T00:00:00.0000000	1	Microcontroller for Door System
PRJ003	CLI003	Website Design	2019-07-02T00:00:00.0...	2019-08-10T00:00:00.0000000	1	Small section of website for ATO
PRJ004	CLI002	Door Safety System	2019-08-02T00:00:00.0...		2	A sequel after the door micro-design

Figure 8.5 - Project Table - Obtained from Microsoft Azure Portal



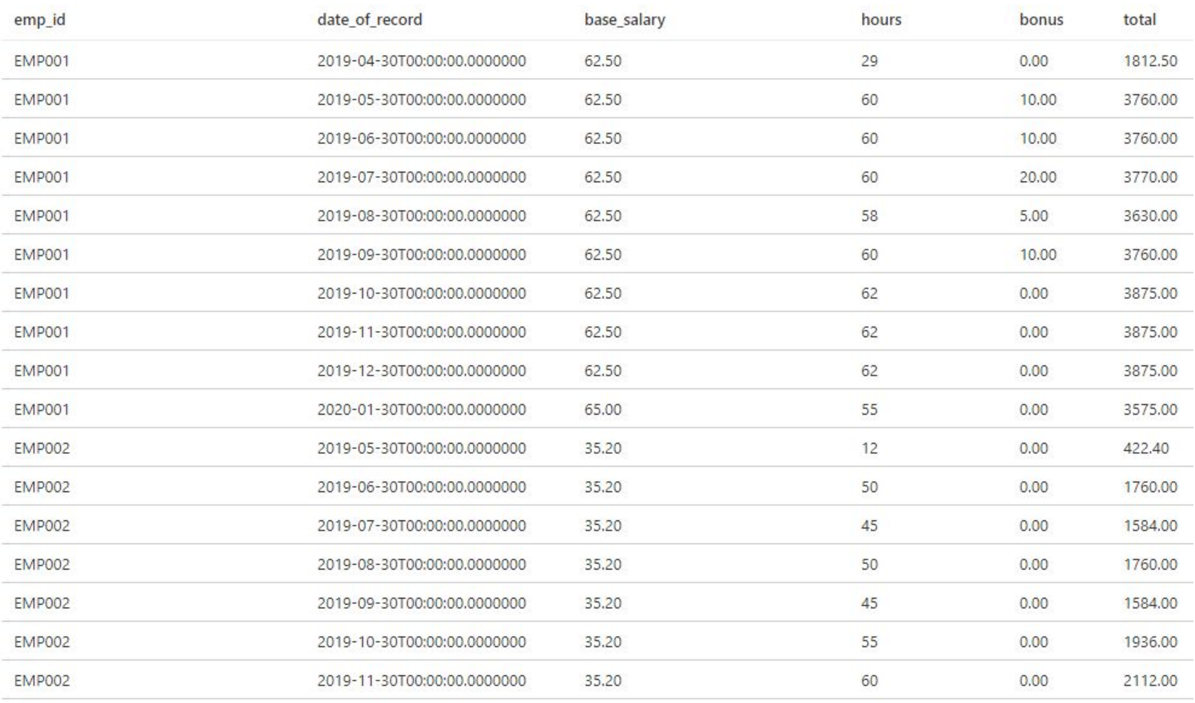
proj_id	size	importance	no_dev	comments
PRJ001	2	2	2	
PRJ002	3	2	3	Finish in Three Months
PRJ003	1	2	1	Government query
PRJ004	1	1	1	

Figure 8.6 - ProjectSize Table - Obtained from Microsoft Azure Portal



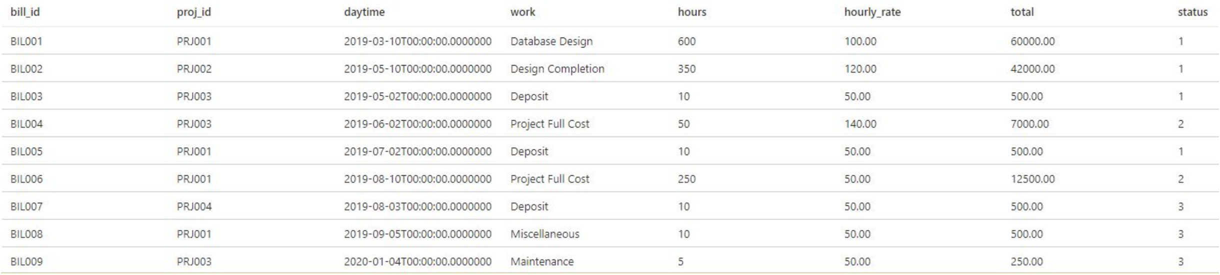
emp_id	proj_id	role	comments
EMP001	PRJ001	Project Manager	Responsible for the progress and completion of the project
EMP002	PRJ001	Developer	Develop the database system
EMP001	PRJ002	Tester	Responsible for the product being up to par
EMP003	PRJ002	Project Manager	Responsible for the progress and completion of the project
EMP004	PRJ002	Developer	Microcontroller Designer
EMP001	PRJ003	Sole Developer	Responsible for the project
EMP003	PRJ004	Sole Developer	Responsible for the project

Figure 8.7 - WorksOn Table - Obtained from Microsoft Azure Portal



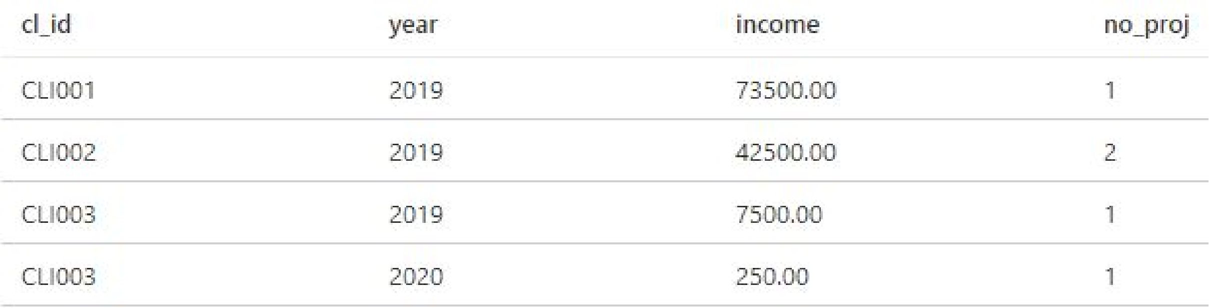
emp_id	date_of_record	base_salary	hours	bonus	total
EMP001	2019-04-30T00:00:00.0000000	62.50	29	0.00	1812.50
EMP001	2019-05-30T00:00:00.0000000	62.50	60	10.00	3760.00
EMP001	2019-06-30T00:00:00.0000000	62.50	60	10.00	3760.00
EMP001	2019-07-30T00:00:00.0000000	62.50	60	20.00	3770.00
EMP001	2019-08-30T00:00:00.0000000	62.50	58	5.00	3630.00
EMP001	2019-09-30T00:00:00.0000000	62.50	60	10.00	3760.00
EMP001	2019-10-30T00:00:00.0000000	62.50	62	0.00	3875.00
EMP001	2019-11-30T00:00:00.0000000	62.50	62	0.00	3875.00
EMP001	2019-12-30T00:00:00.0000000	62.50	62	0.00	3875.00
EMP001	2020-01-30T00:00:00.0000000	65.00	55	0.00	3575.00
EMP002	2019-05-30T00:00:00.0000000	35.20	12	0.00	422.40
EMP002	2019-06-30T00:00:00.0000000	35.20	50	0.00	1760.00
EMP002	2019-07-30T00:00:00.0000000	35.20	45	0.00	1584.00
EMP002	2019-08-30T00:00:00.0000000	35.20	50	0.00	1760.00
EMP002	2019-09-30T00:00:00.0000000	35.20	45	0.00	1584.00
EMP002	2019-10-30T00:00:00.0000000	35.20	55	0.00	1936.00
EMP002	2019-11-30T00:00:00.0000000	35.20	60	0.00	2112.00

Figure 8.8 - Salary Table - Obtained from Microsoft Azure Portal



bill_id	proj_id	daytime	work	hours	hourly_rate	total	status
BIL001	PRJ001	2019-03-10T00:00:00.0000000	Database Design	600	100.00	60000.00	1
BIL002	PRJ002	2019-05-10T00:00:00.0000000	Design Completion	350	120.00	42000.00	1
BIL003	PRJ003	2019-05-02T00:00:00.0000000	Deposit	10	50.00	500.00	1
BIL004	PRJ003	2019-06-02T00:00:00.0000000	Project Full Cost	50	140.00	7000.00	2
BIL005	PRJ001	2019-07-02T00:00:00.0000000	Deposit	10	50.00	500.00	1
BIL006	PRJ001	2019-08-10T00:00:00.0000000	Project Full Cost	250	50.00	12500.00	2
BIL007	PRJ004	2019-08-03T00:00:00.0000000	Deposit	10	50.00	500.00	3
BIL008	PRJ001	2019-09-05T00:00:00.0000000	Miscellaneous	10	50.00	500.00	3
BIL009	PRJ003	2020-01-04T00:00:00.0000000	Maintenance	5	50.00	250.00	3

Figure 8.9 - Bill Table - Obtained from Microsoft Azure Portal



cl_id	year	income	no_proj
CLI001	2019	73500.00	1
CLI002	2019	42500.00	2
CLI003	2019	7500.00	1
CLI003	2020	250.00	1

Figure 8.10 - AnnualBillReport Table - Obtained from Microsoft Azure Portal

IX. Discussion & Conclusion

Even though cloud-based databases might present issues with queries speed and performance, cloud-based databases are shown in case of studies and through the experiment of this research that it provides other benefits that outweigh the disadvantages. Cloud-based databases provide more flexibility when it comes to the scalability of the databases. Startups can choose to start with a small database for a smaller cost and upgrade the size along the way as the company grows. This option provides a benefit compared to the traditional database infrastructure where the companies have to estimate the size of the database correctly, and the price of upgrading database infrastructure will be costly. Companies will also be saving the revenue on the expense of IT infrastructure and IT professionals needed to maintain the database systems, as the cloud service provider will outsource this.

Furthermore, most cloud service providers also charge on a pay-as-you-go payment, where cost can be minimized. The process of data migration can also be minimalistic if the companies possess enough resources; namely, time and money. In general, starting or migrating a startup company's database system to cloud-based database system provides a larger benefit to the future of the company's infrastructure.

The proposed company, Softtech, is not an exception. By migrating the Softtech database system to a cloud service provider such as Microsoft Azure or AWS, Softtech will be able to benefit from the tools, services, and efficiency as discussed above. Based on various studies and this research, the construction or build up of the process of Softtech's database in a cloud-based database is highly encouraged.

X. References

- ‘Azure Database for MariaDB pricing’ [image], Microsoft Azure 2020?, viewed 03 November 2020, <https://azure.microsoft.com/en-us/pricing/details/mariadb/>.

- ‘Azure Ping Test (Latency)’ [image], Cloud Ping Test 2020?, viewed 04 November 2020, <https://cloudpingtest.com/azure>.

- Ferri, L, Maffei, M, Mangia, G & Tomo, A 2017, Analyzing Cloud-based Startups: Evidence from a Case Study in Italy, Canadian Center of Science and Education, Naples, pp. 73, 74, 79, 80, viewed 2 November, 2020, <https://pdfs.semanticscholar.org/3bae/92b417440b3808444efc41969063b279ccb0.pdf>.

- Hoffman, C 2018, What Is Microsoft Azure, Anyway?, viewed 2 November, 2020, <https://www.howtogeek.com/337961/what-is-microsoft-azure/>.

- Jorge Passos da Costa, P & Miguel Rosado da Cruz, A 2012, "Migration to Windows Azure – Analysis and Comparison", CENTERIS 2012 - Conference on ENTERprise Information Systems / HCIST 2012 - International Conference on Health and Social Care Information Systems and Technologies, Procedia Technology, Valença, pp. 94, 98, viewed 3 November, 2020, <https://www.sciencedirect.com/science/article/pii/S2212017312004422>.

- MariaDB 2015?, Poor performance on inserts MariaDB, viewed 03 November 2020, <https://mariadb.com/kb/en/poor-performance-on-inserts-mariadb/>.

- Microsoft Doc. c. 2020, How to backup and restore a server in Azure Database for MySQL using the Azure portal, Microsoft Doc., viewed 03 November 2020, <https://docs.microsoft.com/en-us/azure/mysql/howto-restore-server-portal>.

- MSV, J 2020, A Look Back At Ten Years Of Microsoft Azure, viewed 2 November, 2020, <https://www.forbes.com/sites/janakirammsv/2020/02/03/a-look-back-at-ten-years-of-microsoft-azure/?sh=2629544e4929>.

- MySQL 2020?, MySQL Enterprise Edition, viewed 03 November 2020, <https://www.mysql.com/products/workbench/>.

- ‘MySQL Workbench Home’ [image], MySQL 2020?, viewed 03 November 2020, <https://www.mysql.com/common/images/products/MySQL_Workbench_Mainscreen_Windows.gif>.

- Sultan, N 2011, "Reaching for the “cloud”: How SMEs can manage", International Journal of Information Management, vol. 31, no. 3, pp. 273 - 276, viewed 1 November, 2020, <https://www.sciencedirect.com/science/article/pii/S0268401210001143>.

- Tutorialspoint 2020?, MariaDB Tutorial, viewed 03 November 2020, <https://www.tutorialspoint.com/mariadb/index.htm>.