



SIT772

Database Fundamentals

Learning Summary Report

Isaac Han  
STUDENT ID: s224509577

**Self-Assessment Details**

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment				✓

Checklist	Included
Learning Summary Report	✓
All tasks required for the target grade completed	✓
Evidence of any additional task(s) or activities completed	✓

**Declaration**

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: **Isaac Han**

## Portfolio Overview

This portfolio includes work that demonstrates that I have achieved all unit learning outcomes for SIT772 Database Fundamentals to a High Distinction level.

Before this unit, I was only familiar with data languages like Pandas and R. Without much prior knowledge of structured query languages, I delved into database languages. I had understood the importance of data for statistics but did not realize how easily data relationships could be represented through entity-relationship diagrams. These diagrams enable an intuitive, higher-level understanding of data.

One of the key learning outcomes was interpreting and explaining concepts of data, information, and knowledge. The task *Reflection on Data-driven Systems* and *Entity-Relationship Diagrams* presented the idea that high-level reasoning can transform raw data into information or even knowledge – without complex calculations like covariance matrices. The task *Data Analysis and Visualization* also reinforced this by demonstrating that business intelligence and insights can emerge from simple visualizations rather than advanced statistics. I also noted that PK, FK, normalization of table, and database management can have data integrity, database rule reinforcement, automatic trigger, and handling large data faster unlike traditional file system and excel may not be capable of.

Regarding designing, implementing, evaluating, and maintaining databases through SQL, I gained practical experience during a two-month internship as a software engineer with a small company. I was able to design and implement a sample database, personally. For example, I linked customer inquiries with software, hardware, and manufacturing teams. The company had difficulties identifying the correct software version, so I introduced a collision-free GitHub has function as the primary key in the software table. Using Apache, HTML, PHP, and an SQL server, I was able to simulate customer inquiries via a website, automatically trigger database insertions, and send email notifications to employees. This was inspired through tasks 4.2, 7.2, 8.2, and 10.1. I also applied SQL to automate tasks, such as price reduction for products whose names contain “Univ” and creating a virtual view table for unshipped products.

Finally, I conceptually explored machine learning algorithms in a probing company and learned that large volumes of data do not always improve a learning algorithm’s accuracy. I applied active learning and probably approximately correct learning and learned from a book that some additional data may not contribute to better results. I also studied how image filtering and enhancement techniques can improve low-quality data, explained in OpenCV documentation under morphological operations and Gaussian filtering. I studied the Haar Cascade algorithm and Yolo object detection where large volumes of labeled data were fed into the algorithm. With external devices feeding buffered stream input frames, I saw how proper data pipelining is crucial for object detection. Techniques like storing the last detected cached frame can improvise non-continuous object detections and flickering.

## Reflections

The most important lesson I learned was that simple data relationships can be as effective as well-coordinated complex statistical calculations. I have been focused on using statistics for machine learning, I realized that designing databases and creating entity-relationship diagrams can provide a more intuitive and efficient approach. As a result, designing a table with well-planned relationships can save considerable time and effort.

### I feel I learnt these topics, concepts, and/or tools really well:

When I first encountered SQL, I found its structure quite similar to that of R and Pandas, which made learning SQL more comfortable for me.

### I found the following topics particularly challenging:

I found PSL and PL challenging because I was unfamiliar with it and did not have sufficient time to read the textbook due to my internship. The lack of study made it harder for me to grasp the new syntax.

### I found the following topics particularly interesting:

I found the trigger function particularly fascinating. Although I didn't fully implement the idea, I made a prototype that when a customer inquiry is made through HTML(GUI), a row is inserted into SQL server through PHP, and a trigger automatically email workers.

### I still need to work on the following areas:

I am not entirely comfortable using procedural query languages, such as PSL, as I often make mistakes. In one hand, I think integrating databases with procedural languages like Python could substitute the need for PSL in handling datasets, but I also recognize that at least knowing and being able to use PSL can be a great tool.

### The things that helped me most were:

The textbook was incredibly helpful in building my foundational knowledge of databases. Its case studies frequently helped me answer task questions and refine my approaches into another level.

My progress in this unit was ...:



I aimed to achieve a High Distinction in this unit, which was challenging due to the internship I got in the middle. I often worked on assignments after my internship hours, and I am proud to have completed most of the tasks. However, I regret not having more time to read the textbook once I started working.

If I did this unit again, I would do the following things differently:

If I were to take this unit again, I would dedicate more time to reading the textbook to reinforce my learning.

Other...:

Studying this unit while working at a company allowed me to visualize several database techniques that the company lacked. This helped me develop a deeper understanding of the course concepts by making a few database-related projects.

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ISSAC HAN

---

## Portfolio Submission

---

*Submitted By:*

Issac HAN  
s224509577

*Tutor:*

Iynkaran NATGUNANATHAN

October 2, 2024



---

## Contents

<b>1 Learning Summary Report</b>	<b>1</b>
<b>2 Overall Task Status</b>	<b>2</b>
<b>3 Learning Outcomes</b>	<b>3</b>
3.1 Fundamental concepts of database . . . . .	3
3.2 Relational Database Modelling . . . . .	3
3.3 Structured Query Language (SQL) . . . . .	3
3.4 Reflection . . . . .	4
3.5 Research and critical review . . . . .	4
<b>4 Additional Files</b>	<b>6</b>
FinalProj.pdf . . . . .	7
<b>5 Reflection on Data-driven Systems</b>	<b>8</b>
<b>6 Install MySQL</b>	<b>13</b>
<b>7 Database Modelling Tools</b>	<b>16</b>
<b>8 Entity Relationship Diagram</b>	<b>20</b>
<b>9 Database Normalisation</b>	<b>24</b>
<b>10 Miniproject Part-1 - Database Design and Normalisation</b>	<b>28</b>
<b>11 Basic SQL</b>	<b>33</b>
<b>12 Online Quiz 1</b>	<b>40</b>
<b>13 SELECT Queries with JOIN</b>	<b>43</b>
<b>14 Other DML and DDL Queries</b>	<b>49</b>
<b>15 Research Report and Presentation</b>	<b>55</b>
<b>16 Miniproject Part-2 - Database Implementation</b>	<b>73</b>
<b>17 Online Quiz 2</b>	<b>82</b>
<b>18 PL/PSQL - Trigger, Procedure and Function</b>	<b>85</b>
<b>19 Data Analysis and Visualization using Excel</b>	<b>92</b>
<b>20 Interaction with a database via an User Interface</b>	<b>99</b>
<b>21 Learning Summary Report</b>	<b>106</b>

---

## 2 Overall Task Status

Task	Status	Times Assessed
Reflection on Data-driven Systems	Complete	1
Install MySQL	Complete	1
Database Modelling Tools	Complete	2
Entity Relationship Diagram	Complete	1
Research Report and Presentation	Complete	1
Database Normalisation	Complete	1
Miniproject Part-1 - Database Design and Normalisation	Complete	1
Basic SQL	Complete	1
Online Quiz 1	Complete	1
SELECT Queries with JOIN	Complete	1
Other DML and DDL Queries	Complete	1
Miniproject Part-2 - Database Implementation	Complete	1
Online Quiz 2	Complete	1
PL/PSQL - Trigger, Procedure and Function	Complete	1
Data Analysis and Visualization using Excel	Complete	1
Interaction with a database via an User Interface	Complete	1
Learning Summary Report	Complete	1

### 3 Learning Outcomes

#### 3.1 Fundamental concepts of database

Interpret and explain fundamental concepts of data, information, and knowledge and demonstrate an understanding of differences between traditional file systems and databases.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦♦♦◊◊	Complete	1
Database Modelling Tools	♦♦♦◊◊	Complete	2
Entity Relationship Diagram	♦♦♦♦♦	Complete	1
Research Report and Presentation	♦♦♦◊◊	Complete	1
Database Normalisation	♦♦♦♦◊	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦◊◊◊	Complete	1
Online Quiz 2	♦♦♦♦♦	Complete	1
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦◊	Complete	1
Data Analysis and Visualization using Excel	♦♦◊◊◊	Complete	1
Interaction with a database via an User Interface	♦♦♦◊◊	Complete	1
Learning Summary Report	♦♦♦♦♦	Complete	1

#### 3.2 Relational Database Modelling

Analyse real-world problems to identify data requirements and apply data modelling concepts to design and develop Entity Relationship Diagrams for efficient data representation and storage.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦◊◊◊◊	Complete	1
Database Modelling Tools	♦♦♦♦◊	Complete	2
Entity Relationship Diagram	♦♦♦♦◊	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	1
Database Normalisation	♦♦♦♦◊	Complete	1
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦◊	Complete	1
Other DML and DDL Queries	♦♦◊◊◊	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Learning Summary Report	♦♦♦♦♦	Complete	1

#### 3.3 Structured Query Language (SQL)

Design, implement, evaluate and maintain relational database systems using SQL and Database Management Systems and explain the purpose of various SQL commands and operations.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦◊◊◊◊	Complete	1
Install MySQL	♦♦◊◊◊	Complete	1
Basic SQL	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦◊◊	Complete	1
Other DML and DDL Queries	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦◊◊◊	Complete	1
Learning Summary Report	♦♦♦♦♦	Complete	1

### 3.4 Reflection

Analyse and critique the achievements of learning outcomes and justify meeting specified outcomes through providing relevant evidence and evaluating the quality of that evidence against given criteria.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦♦♦	Complete	1
Install MySQL	♦◊◊◊◊	Complete	1
Database Modelling Tools	♦♦♦◊◊	Complete	2
Entity Relationship Diagram	♦♦♦◊◊	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	1
Database Normalisation	♦♦♦♦◊	Complete	1
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	1
Online Quiz 1	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦♦♦	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	1
Learning Summary Report	♦♦♦♦♦	Complete	1

### 3.5 Research and critical review

Conduct research on and critically evaluate tools and technologies used in contemporary business information systems to manage and analyse data.

Task	Rating	Status	Times Assessed
Reflection on Data-driven Systems	♦♦♦◊◊	Complete	1
Entity Relationship Diagram	♦♦♦♦◊	Complete	1
Research Report and Presentation	♦♦♦♦♦	Complete	1
Database Normalisation	♦♦♦♦◊◊	Complete	1
Miniproject Part-1 - Database Design and Normalisation	♦♦♦♦♦	Complete	1
SELECT Queries with JOIN	♦♦♦◊◊	Complete	1
Other DML and DDL Queries	♦♦◊◊◊	Complete	1
Miniproject Part-2 - Database Implementation	♦♦♦♦♦	Complete	1
PL/PSQL - Trigger, Procedure and Function	♦♦♦♦♦	Complete	1
Data Analysis and Visualization using Excel	♦♦♦♦♦	Complete	1
Interaction with a database via an User Interface	♦♦♦♦♦	Complete	1
Learning Summary Report	♦♦♦♦♦	Complete	1

---

## **4 Additional Files**

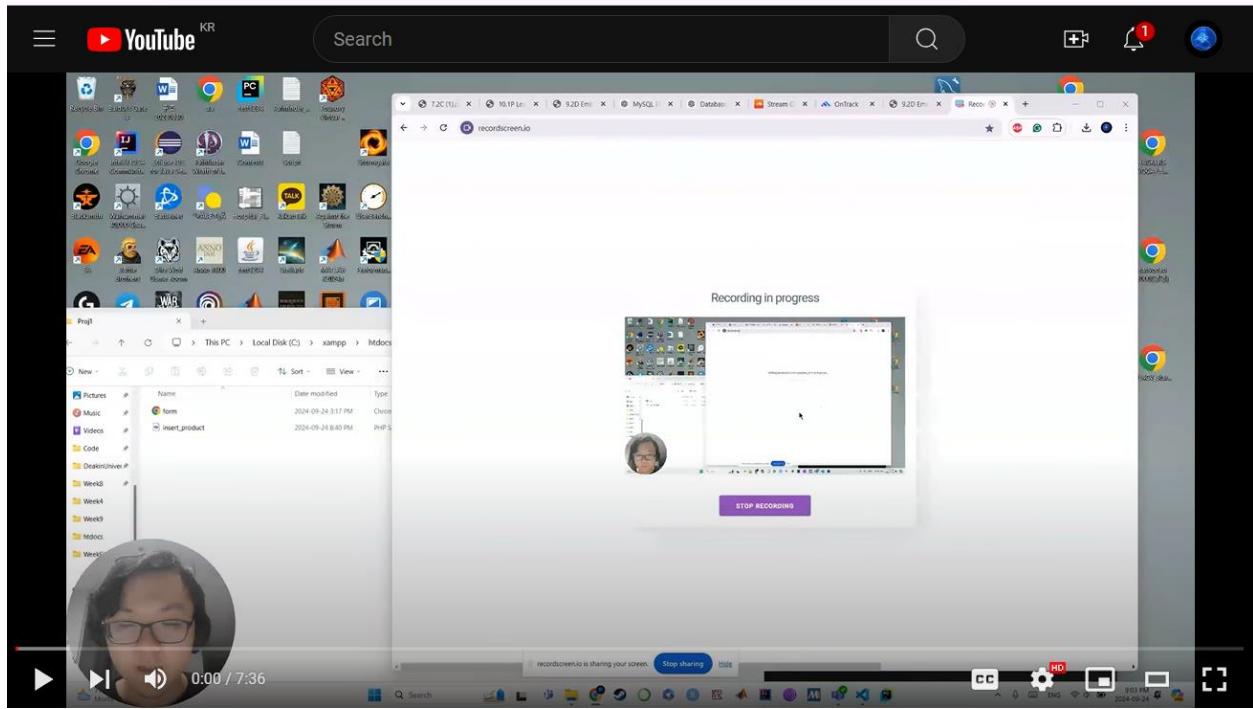
1. FinalProj.pdf

Name: Isaac Han

UID: s224509577

SIT772 Final Projects (HTML -> PHP -> SQL): [https://www.youtube.com/watch?v=Qbqm\\_xbcH8](https://www.youtube.com/watch?v=Qbqm_xbcH8)

I made explanation.



SIT772 Final

## 5 Reflection on Data-driven Systems

Reflect on three data-driven information systems that you use in your daily life.

Outcome	Weight
Fundamental concepts of database	◆◆◆◆

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized. Overall, it was time-consuming but refreshing to think of.

Outcome	Weight
Relational Database Modelling	◆◇◇◇

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized. Overall, it was time-consuming but refreshing to think of.

Outcome	Weight
Structured Query Language (SQL)	◆◇◇◇

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized. Overall, it was time-consuming but refreshing to think of.

Outcome	Weight
Reflection	◆◆◆◆

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized. Overall, it was time-consuming but refreshing to think of.

Outcome	Weight
Research and critical review	◆◆◆◇

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized. Overall, it was time-consuming but refreshing to think of.

Date	Author	Comment
2024/07/16 21:57	Issac Han	Need Help
2024/07/16 21:57	Issac Han	The assignment said to do Harvard-style citations, but I am quite exhausted and confused about how to do it. 1. Can I ask for a link to Turnitin? I want to cite the sources that it might think unoriginal.2. I would like to know the level of citation needed for this paper. Could you advise on it? Thank you.
2024/07/17 13:44	Issac Han	Ready to Mark
2024/07/17 13:44	Issac Han	I am familiar with standards with citations. If there is any problem with the essay or citation, please let me know. Thank you.
2024/07/18 07:05	Aditya Tarigopula	good work Issac, the link to turnintin should be in your cloud deakin site
2024/07/18 07:05	Aditya Tarigopula	Complete

---

## Reflection on Data-driven Systems

---

*Submitted By:*

Issac HAN  
s224509577  
2024/07/17 13:44

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	◆◆◆◆
Relational Database Modelling	◆◆◆◆
Structured Query Language (SQL)	◆◆◆◆
Reflection	◆◆◆◆
Research and critical review	◆◆◆◆

This task was rather thought-provoking. Initially, the journey began with how data is used and stored. Soon, software and DB techniques involved with automatically generating and exchanging data were considered. Finally, some physical devices engaged with data processing were realized.

Overall, it was time-consuming but refreshing to think of.

July 17, 2024



Name: Isaac Han

Class: SIT 772

Assignment: Submission#1

### **Software Maps:**

Daily, I use Maps to navigate from my house to the office and run errands, such as shopping and meeting people. I want to know the sequences of public transportation to take for a satisfactory path to the destination, and even sometimes, I want to find out the name of places with incomplete information, in hand. To find a path, places to wonder around, to know the time it takes for public transportation to arrive, and to reserve a seat in advance, I consume data: Sensor data such as GPS, the current location of public transportation, timetables for estimated bus arrivals, the average traffic volumes on certain boulevard, orientation measurements, seat numbers, and common keywords table that guesses what I meant with an incomplete set of inputs.

I use the service whenever I commute, even though I am familiar with the route. I partly do this to make sure that I am going in the right way and taking off at the right stop. Occasionally, I bypass the stop, at which point the software recommends a detour to get back on track. Without a map service, it can range from being a hassle to a total catastrophe; the software can be thought of as a simple integration of already available data. Bus information from bus corporations, optimal pathing search from engines, GPS from telephone companies, reservation information from the transportation company website. One cannot stress enough that even a tiny inefficiency in a navigational system, when combined, can lead to a billion-dollar economic problem. However, I think I will manage just fine; it will just mean spending a few more minutes and using more internet traffic. On the other hand, if we scale the unavailability of data to the extent that no one has any necessary data at all, it would be like WWIV, where we have no technology at all but stones and sticks. The extinction of GPS, time data, image processing techniques, and digital image data would mean a return to a primitive level of technology.

The traffic data is autonomously generated; GPS devices in my phones, buses, and trains send signals to satellites, which then call back with time and distance metrics, later processed as coordinates. These data are aligned with graphs of roads. Then, it generates estimated arrival times, after being corrected based on algorithms utilizing history. Behind the scenes, there are software engineers who work to automatically generate data to communicate, and DB-related engineers to optimize the storage and flow of data (Carlos and Steven, 2020).

Since location-based services deal with sensitive information, there are plenty of concerns and high risks involved with personal data. For one, spoofing or data breach might reveal unpleasant information about individuals; it might be used to track the person in real life or used in scams, combined with other personal information gathered. Not only could there be illegal and unfortunate incidents, but the company providing the location-based service might also sell the data they collected, which could then be processed into targeted ads. This increases the scope of personal data breaches (Mosaic Theory of Fourth Amendments, 2024).

### **Steam Games:**

Steam is a global game distribution website where it hosts large numbers of games, transactions, and ad-hoc queries for users to download games and developers to upload their contents. From when I was in elementary school playing *The Binding of Isaac*, released in 2011, to the game *Against the Storm* that I just played yesterday, I have been playing games on steam, partly due to its wide range of selection, recommendation systems, and convenient digital ownerships of the games. For transactions, it stores the purchase history (date, amount, entity involved with transaction, reference number), the number of hours I played the game, similarity measurements between steam users for the recommendation system, copyrights, the version history of each game applications, etc.

I am mainly an end user in this database; abstraction layers hide the specific implementation of the database. However, I can make a few comments about how data might be processed in Steam.

First, Steam is a marketplace where sellers sell their digital assets and buyers purchase products. Steam must guarantee the safety of the products -like safety from malicious viruses and monetary transactions for sellers. Without a DB and legal protection safeguarding both parties, the enormous number of simultaneous transactions could not happen almost instantly. Ad-hoc queries, multi-user access, database access control, and the 3Vs (Volume, Velocity, Variety) of data are typical characteristics of DBSM and DB Modeling (Carlos and Steven, 2022). Second, the success of retail enterprises often relies on successful product suggestions to the potential consumers, thereby alleviating the stress for consumers choosing a satisfactory product and increasing sales. Modern machine learning Netflix recommendation system relies on constructing similar matrix whereby the correct classification of customers and filling sparse matrix is of upmost importance (Ethem 2020). Gaining customer data from the purchase history, the time spent in each category of games, the amount of money spent, and their declared interests are keys to successful recommendation. As a consumer, if there is no data available for developers and retail sellers, the quality of the product would decrease. However, end users might simply find another vendor or other forms of entertainment. In a sense, the unavailability of data would affect the whole market, but end users like me might be less sensitive to such changes.

Online transactions entail the exchange of data, typically in JSON format. This format allows companies to automatically process data, therefore, automatically creating data each time successful transactions occur. Developers create digital versions of their work, and this data might be handled by database specialists to be downloadable and accessible by qualified users all around the world. To denote the users who purchased the product, the transaction data extracted from JSON might be processed by algorithms. And the result, classified by the algorithm, would be stored in the database (Carlos and Steven, 2022).

Any service that includes credit information is followed by risks and concerns for the security and privacy of the data. However, cryptographic online transactions, scam detection, bank user management, and law enforcement seem to make online transactions safe enough that we don't need to worry too much about them. As shown in numerous accounts of Mosaic Theory, seemingly irrelevant data combined can reveal detrimental information, which is something to still be cautious about (Mosaic Theory of Fourth Amendments, 2024).

## Bibliography

Carlos, C. and Steven M. (2022) *Database Systems: Design, Implementation, & Management*. 14<sup>th</sup> edn. Boston, MA: Cengage

Wikipedia (2024) *Mosaic Theory of Fourth Amendment*. Available at:  
[https://en.wikipedia.org/wiki/Mosaic\\_theory\\_of\\_the\\_Fourth\\_Amendment](https://en.wikipedia.org/wiki/Mosaic_theory_of_the_Fourth_Amendment)

Ethem, A. (2020) *Introduction to Machine Learning*, 4<sup>th</sup> edn: the MIT Press

## 6 Install MySQL

Install the MySQL community server and workbench.

Outcome	Weight
Fundamental concepts of database	♦♦♦◊◊

Installation and playing with the program is the first step of learning a language, which is essential by itself.

Outcome	Weight
Structured Query Language (SQL)	♦♦◊◊◊

Installation and playing with the program is the first step of learning a language, which is essential by itself.

Outcome	Weight
Reflection	♦◊◊◊◊

Installation and playing with the program is the first step of learning a language, which is essential by itself.

Date	Author	Comment
2024/07/14 19:44	Issac Han	Ready to Mark
2024/07/14 20:33	Aditya Tarigopula	good work hope you got a basic understanding of SQL
2024/07/14 20:33	Aditya Tarigopula	Complete
2024/07/15 02:20	Issac Han	Thank you. I found some tutorials on SQL and see a few videos on it. :)

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Install MySQL

---

*Submitted By:*

Issac HAN  
s224509577  
2024/07/14 19:44

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

Installation and playing with the program is the first step of learning a language, which is essential by itself.

July 14, 2024



MySQL Workbench

LocalMySQL X

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Tables

```

1 -- create database SIT103;
2 • show databases;
3 • use SIT103;
4 • show tables;
5 -- create table student_test (ID int primary key, Name varchar(25));
6 • show tables;
7 • desc student_test;
8 -- insert into student_test values (1234, 'Peter John');
9 • select * from student_test;

```

Views

host\_summary

host\_summary\_by\_f

host\_summary\_by\_f

host\_summary\_by\_s

host\_summary\_by\_s

host\_summary\_by\_t

host\_summary\_by\_t

innodb\_buffer\_stats

innodb\_buffer\_stats

innodb\_lock\_waits

io\_by\_thread\_by\_Id

io\_global\_by\_file\_b

io\_global\_by\_file\_b

io\_global\_by\_file\_b

io\_global\_by\_wait\_t

Administration Schemas Information

Trigger: sys\_config\_update\_set\_user

Definition:

Event: UPDATE

Timing: BEFORE

Action Output

#	Time	Action	Message	Duration / Fetch
47	18:38:06	show databases	5 row(s) returned	0.000 sec / 0.000 sec
48	18:38:06	use SIT103	0 row(s) affected	0.000 sec
49	18:38:06	show tables	1 row(s) returned	0.000 sec / 0.000 sec
50	18:38:06	show tables	1 row(s) returned	0.000 sec / 0.000 sec
51	18:38:06	desc student_test	2 row(s) returned	0.000 sec / 0.000 sec
52	18:38:06	selected * from student_test LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content | Result Grid | Form Editor | Field Types | Query Stats | Contact Help | Snippets

## 7 Database Modelling Tools

Fundamental concepts of relational database modelling and tools for database modelling

Outcome	Weight
Fundamental concepts of database	♦♦♦◊◊

Learned a new tool that can facilitate the designing process; ERDs can help newcomers to onboard the ongoing project, and enable engineers to check whether relational requirements are met or not.

Outcome	Weight
Relational Database Modelling	♦♦♦♦◊

Learned a new tool that can facilitate the designing process; ERDs can help newcomers to onboard the ongoing project, and enable engineers to check whether relational requirements are met or not.

Outcome	Weight
Reflection	♦♦♦◊◊

Learned a new tool that can facilitate the designing process; ERDs can help newcomers to onboard the ongoing project, and enable engineers to check whether relational requirements are met or not.

Date	Author	Comment
2024/07/21 20:01	Issac Han	Ready to Mark
2024/07/21 20:01	Issac Han	Just wondering what are premium features of Lucid Chart, are and whether Deakin offers full access or not.
2024/07/22 05:26	Aditya Tarigopula	Fix and Resubmit
2024/07/22 05:26	Aditya Tarigopula	you need to draw two diagrams as given in the task sheet, please submit again
2024/07/22 18:41	Issac Han	Ready to Mark
2024/07/23 06:06	Aditya Tarigopula	Complete

---

## Database Modelling Tools

---

*Submitted By:*

Issac HAN  
s224509577  
2024/07/22 18:41

*Tutor:*

Aditya TARIGOPULA

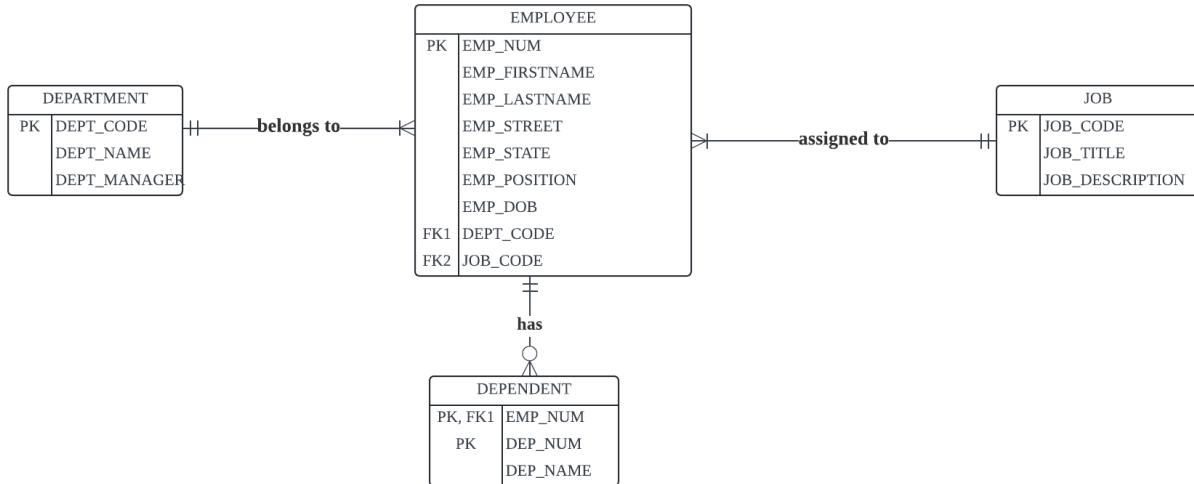
Outcome	Weight
Fundamental concepts of database	♦♦♦◊◊
Relational Database Modelling	♦♦♦♦♦
Reflection	♦♦♦◊◊

Learned a new tool that can facilitate the designing process; ERDs can help newcomers to onboard the ongoing project, and enable engineers to check whether relational requirements are met or not.

July 22, 2024



Name: Isaac Han  
Class: SIT772



- How many Entities are there and what are they?
- For the EMPLOYEE Entity:
  - (i) List all attributes;
  - (ii) What is the Primary Key and how do you know that? and
  - (iii) Are there any Foreign Key(s)? If yes, what are they and what are they referring to?
- Explain the relationship between EMPLOYEE and DEPENDENT. Your explanation should include how is EMPLOYEE related to DEPENDENT and vice-versa in terms of whether the relationship is Optional/Mandatory and their cardinalities. Also, discuss how is the relationship maintained in the ERD [Hint: Primary and Foreign Keys].
- What is the Primary Key of DEPENDENT? What type of key is it?

Q1: There are four entities, named Department, Employee, Job, and Dependent.

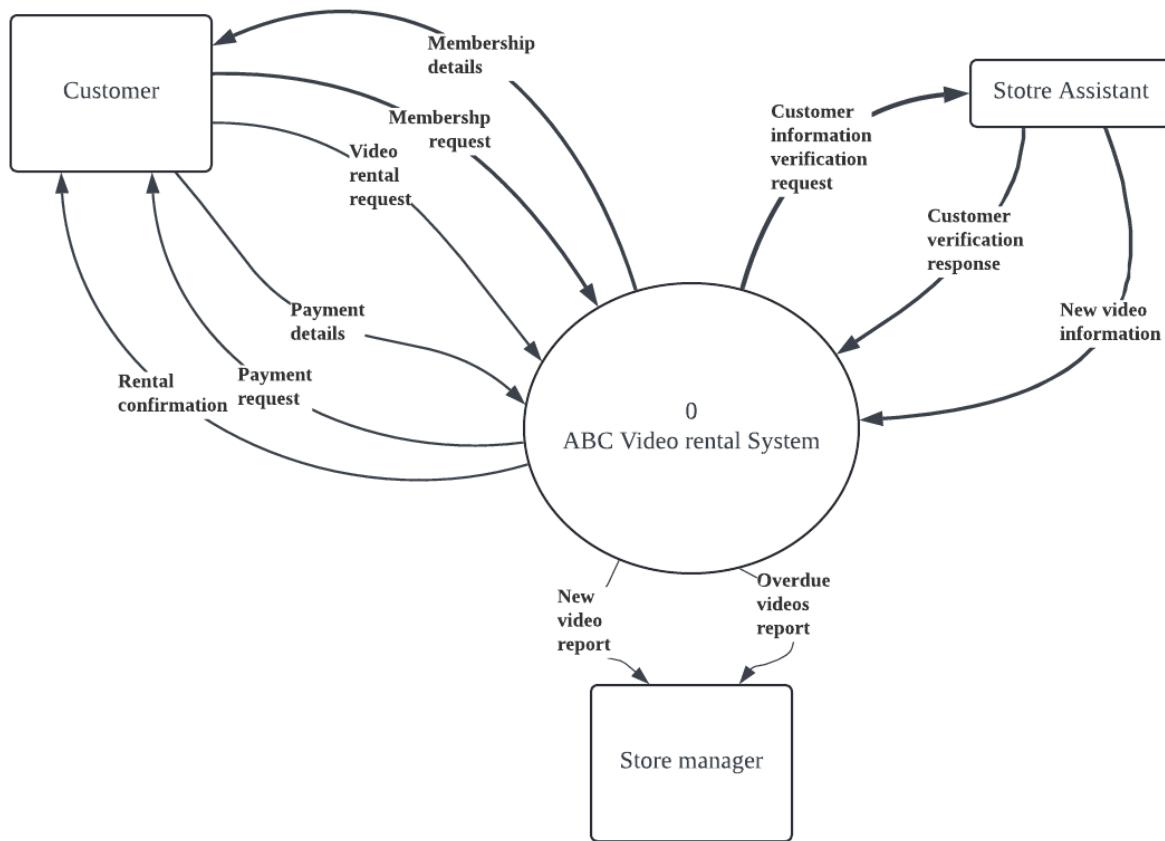
Q2: Employee has 9 attributes: EMP\_NUM, EMP\_FIRSTNAME, EMP\_LASTNAME, EMP\_STREET, EMP\_STATE, EMP\_POSITION, EMP\_DOB, DEPT\_CODE, and JOB\_CODE. The Primary Key is EMP\_NUM and is denoted by “PK” in the KEY field of entity. There are two Foreign Keys: DEPT\_CODE and JOB\_CODE. Foreign Keys are attributes that can uniquely identify rows in Entities outside; they reference instances in outside tables. Although I cannot find notion connecting which Foreign Keys are connected to which, the intuitive names indicate that DEPT\_CODE is for the DEPARTMENT table, and JOB\_CODE is for the JOB table.

Q3: Employees may have several dependents: from zero to many. Each dependent is associated with one employee at a time. Dependent may be null; an employee may not have it. But for all dependents, they must be associated with one and only one employee -personally, it may disregard the fact that both parents are employed at the same firm though – denoted by the double line

symbol. The empty circle with multiple line symbol represents it is optional and potential 1:M relations; the employ might not enter the value, 0, 1, or n. The relationship is maintained through the usage of a Foreign key and a Composite Key. The attribute, EMP\_NUM, is the Primary Key in both tables. Meaning, the Key can uniquely identify rows in both Employee and Dependents. Furthermore, by combining the key with DEP\_NUMBER, you can uniquely identify each dependents; one employee might have multiple dependents, and only having employee as a key in the Dependents table might not be uniquely identify each rows.

Q4. The Primary Key of DEPENDENT is a Composite Key, consisted of both EMP\_NUM and DEP\_NUM. Some additional details are already mentioned.

Draw the second Visio diagram:



## 8 Entity Relationship Diagram

Analyse data requirements of a given system and draw an ERD

Outcome	Weight
Fundamental concepts of database	◆◆◆◆◆

The situation was described rather simply—however, the actual interactions between entities required much more detailed analysis and individual research.

Outcome	Weight
Relational Database Modelling	◆◆◆◆◆

The situation was described rather simply—however, the actual interactions between entities required much more detailed analysis and individual research.

Outcome	Weight
Reflection	◆◆◆◆◆

The situation was described rather simply—however, the actual interactions between entities required much more detailed analysis and individual research.

Outcome	Weight
Research and critical review	◆◆◆◆◆

The situation was described rather simply—however, the actual interactions between entities required much more detailed analysis and individual research.

Date	Author	Comment
2024/07/28 00:32	Issac Han	Ready to Mark
2024/07/28 00:33	Issac Han	<a href="https://lucid.app/lucidchart/10d062e8-0ab4-426b-ac2a-7e3aa681a97f/edit?viewport_loc=-994%2C705%2C2502%2C1452%2C0_0&amp;invitationId=inv_a923ef38-cbb9-4422-bf3a-bde4431148fd">https://lucid.app/lucidchart/10d062e8-0ab4-426b-ac2a-7e3aa681a97f/edit?viewport_loc=-994%2C705%2C2502%2C1452%2C0_0&amp;invitationId=inv_a923ef38-cbb9-4422-bf3a-bde4431148fd</a>
2024/07/28 00:34	Issac Han	image comment
2024/07/28 00:34	Issac Han	image comment
2024/07/28 00:35	Issac Han	The resolution of the diagram is bad after conversion. So I attached both link to the diagram and png version downloaded directly from lucidchart.
2024/07/28 00:35	Issac Han	Have a great day!
2024/08/04 18:38	Aditya Tarigopula	Good work
2024/08/04 18:38	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Entity Relationship Diagram

---

*Submitted By:*

Issac HAN  
s224509577  
2024/07/28 00:32

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦◊

The situation was described rather simply—however, the actual interactions between entities required much more detailed analysis and individual research.

July 28, 2024

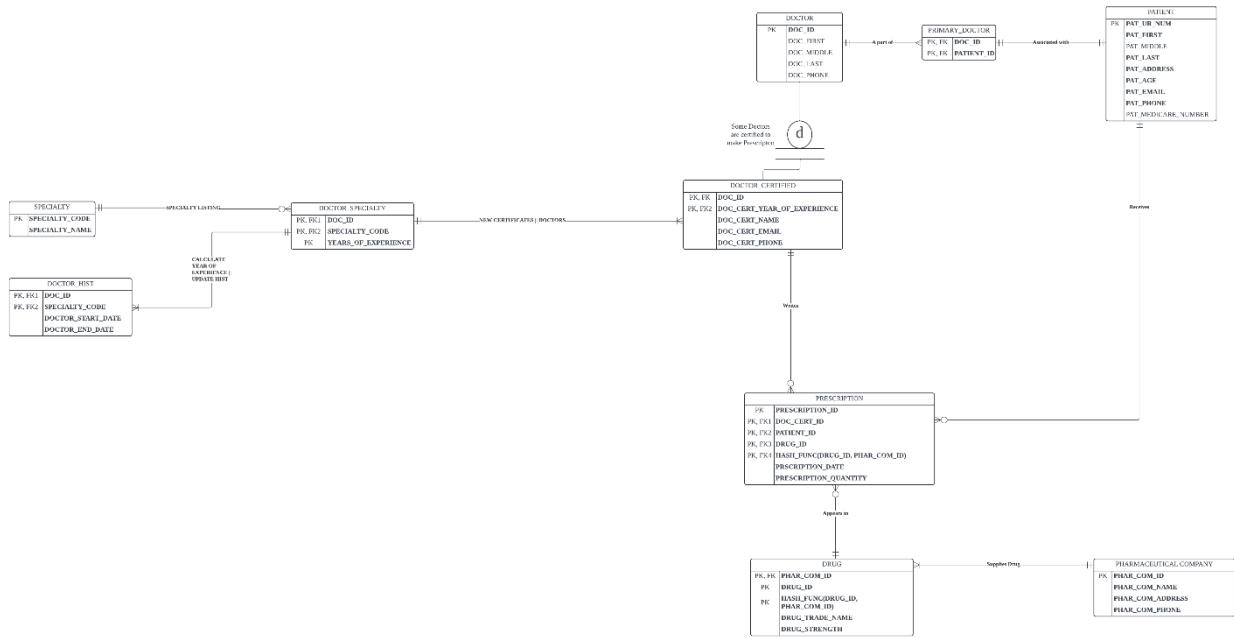


Name: Isaac Han

UID: s224509577

Class: SIT 772

Name: Isaac Han  
UID: s224509577  
Class: SIT 772



Prompts:

If you make any assumptions, do not forget to list them first in your submission.

Provide justifications (based on business rules or your assumptions) for entities, attributes, constraints, and relationships/cardinalities you considered

P1. Identify entities, attributes, constraints, and relationships with cardinalities:

All were identified and shown through the diagram; a few additional features would be discussed under assumptions.

P2. Identify Primary Key in all entities:

Identified and shown in the diagram.

P3 Implement and draw: Done

P4. Use inheritance and advanced diagramming techniques: Done

P5.

Assumptions & Justification:

1. The doctor entity only required DOC\_ID, but it doesn't make sense for subtypes to be more general than the parent type; so, basic attributes were made into optional.
2. I have considered Primary doctor as specialization(subtypes) of the Doctor entity. However, the Primary Doctor only interacts with patients and separates the M: N relationship between Doctors and Patients. Therefore, it was considered as an associative entity.
3. The year of experience is a time-dependent data, and doctors may be specialized in multiple disciplines. Based on the textbook's managing history data solution (p# 179), Specialty and Doctor Hist were created to calculate each doctor's lists of specialty and Year of experience. Additionally, doctors may discontinue their practices; so, each start date and end date were created to calculate precise year of experiences of doctors.
4. Based on the textbook's ternary relationship (p# 127), the relationship between doctors, patients, and drugs were derived by inserting the Prescription entity.
5. I have experience of working with a hospital managing patient data. It is very typical that multiple pharmaceutical companies to supply the same drugs (based on chemical compounds) with different branding names and prices. I created a hash function of drug identification number and pharmaceutical company identification numbers so that same drugs may be supplied by multiple companies. Moreso, the same drug may be supplied by the same pharmaceutical company under different branding names. Hash function chaining to solve such name collision problems.

## 9 Database Normalisation

Draw Dependency diagram and Normalise a given database table

Outcome	Weight
Fundamental concepts of database	♦♦♦♦◊

I studied the NP format and read the textbook to identify dependencies.

Outcome	Weight
Relational Database Modelling	♦♦♦♦◊

I studied the NP format and read the textbook to identify dependencies.

Outcome	Weight
Research and critical review	♦♦♦♦◊

I studied the NP format and read the textbook to identify dependencies.

Outcome	Weight
Reflection	♦♦♦♦◊

Normalization requires a deep reflection on the dependencies of variables.

Date	Author	Comment
2024/08/06 04:14	Issac Han	Ready to Mark
2024/08/06 07:08	Aditya Tarigopula	good work
2024/08/06 07:08	Aditya Tarigopula	Complete

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Database Normalisation

---

*Submitted By:*

Issac HAN

s224509577

2024/08/06 04:14

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦◊◊
Relational Database Modelling	♦♦♦◊◊
Research and critical review	♦♦♦♦♦

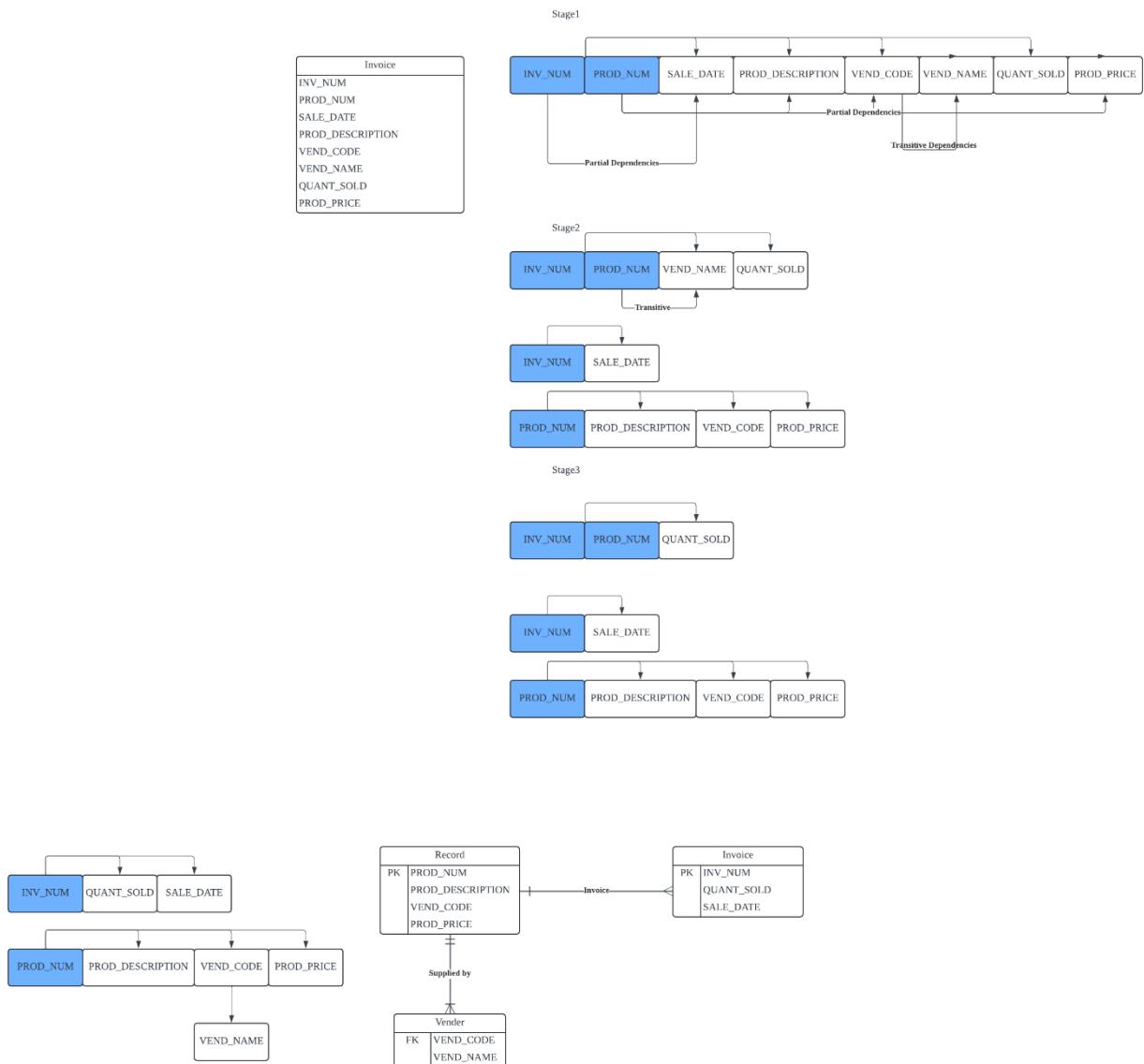
I studied the NP format and read the textbook to identify dependencies.

August 6, 2024



Name: Isaac Han

Class: SIT 772



Normalization: We started with a single-valued table.

At stage 2, apparent dependencies are taken out; Products are separated from the INVOICE. Since each row is single-valued and data is scalar, it is in the form of NP1. I additionally identified that the candidate key of a table; the sale date only requires invoice number to be correctly identified, instead of requiring invoice number and product number.

11	INVOICE						
12	INV_NUM	22347	211347	211347	211348	211349	
13	PROD_NUM	AA-E3422QW	QD-300932X	RU-995748G	AA-E3422QW	GH-778345P	
14	VEND_NAME	NeverFail, Inc.	NeverFail, Inc.	BeGood, Inc.	NeverFail, Inc.	ToughGo, Inc.	
15	QUANT SOLD	1	8	1	2	1	
16							
17	INVOICE2						
18	INV_NUM	22347	211347	211347	211348	211349	
19	SALE_DATE	15-Jan-04	15-Jan-04	15-Jan-04	15-Jan-04	16-Jan-04	
20							
21	PRODUCT						
22	PROD_NUM	AA-E3422QW	QD-300932X	RU-995748G	AA-E3422QW	GH-778345P	
23	VEND_CODE	211	211	309	211	157	
24	PROD_PRICE	\$49.95	\$3.45	\$39.99	\$49.95	\$87.75	
25	PROD_DESCRIPTION	Rotary sander	0.25-in. drill bit	Band saw	Rotary sander	Power drill	
--							

At stage 3, transitive dependencies are taken out and composite key (INV\_NUM, PROD\_NUM) are separated into two different tables. Since every non-candidate key attributes depends on the whole key, it is now in the NP3 format.

PRODUCT							
PROD_NUM	AA-E3422QW	QD-300932X	RU-995748G	AA-E3422QW	GH-778345P		
PROD_PRICE	\$49.95	\$3.45	\$39.99	\$49.95	\$87.75		
PROD_DESCRIPTION	Rotary sander	0.25-in. drill bit	Band saw	Rotary sander	Power drill		
VEND_CODE	211	211	309	211	157		
INVOICE							
INV_NUM	22347	211347	211347	211348	211349		
QUANT SOLD	1	8	1	2	1		
SALE_DATE	15-Jan-04	15-Jan-04	15-Jan-04	15-Jan-04	16-Jan-04		
Vendor							
VEND_CODE	211	211	309	211	157		
VEND_NAME	NeverFail, Inc.	NeverFail, Inc.	BeGood, Inc.	NeverFail, Inc.	ToughGo, Inc.		

## 10 Miniproject Part-1 - Database Design and Normalisation

Analyse data requirements of a company, design and normalised a database to meet their needs.

Outcome	Weight
Research and critical review	◆◆◆◆◆

By creating the entity diagram of the company that I am working, I was able to learn parameters that I was not familiar with: like TCR, Modbus, epc3016, RS485, and petier.

Outcome	Weight
Reflection	◆◆◆◆◆

Researched the product dynamics of the company and reflected on the better way by studying other dev ops online.

Outcome	Weight
Relational Database Modelling	◆◆◆◆◇

I created an entity-relationship diagram for relational database modeling.

Date	Author	Comment
2024/08/09 12:53	Issac Han	Ready to Mark
2024/08/11 17:04	Aditya Tarigopula	good work
2024/08/11 17:04	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Miniproject Part-1 - Database Design and Normalisation

---

*Submitted By:*

Issac HAN  
s224509577  
2024/08/09 12:53

*Tutor:*  
Aditya TARIGOPULA

Outcome	Weight
Research and critical review	♦♦♦♦

By creating the entity diagram of the company that I am working, I was able to learn parameters that I was not familiar with: like TCR, Modbus, epc3016, RS485, and petier.

August 9, 2024



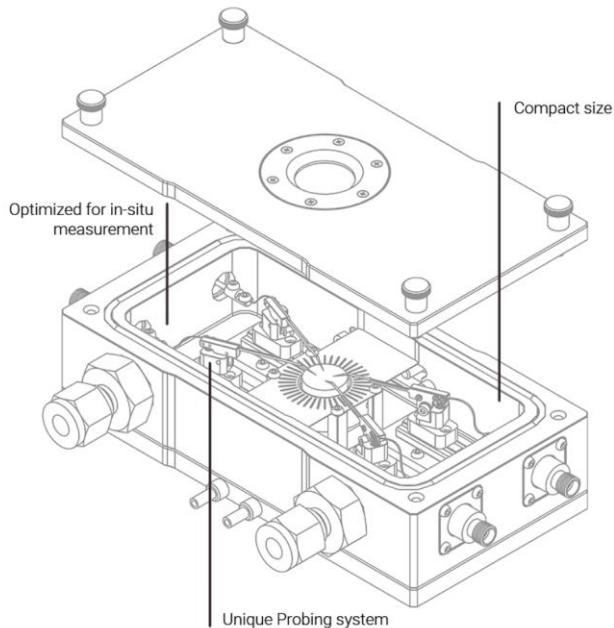
Name: Isaac Han

Class: SIT 772

Project: 4.2C Mini Project1

Subject: Nextron

Nextron is a hardware company that I am doing my internship at. Its main business is selling its products commercially named as “microprobe,” formally classified as probing stations: <https://www.microprobesystem.com/index.html>. The role of probing stations is to test, to analyze failures, and to meet the quality control requirements. Making small sized probing stations has benefits in certain fields such as semiconductor, IC, and circuits.



The company is divided into R&D, manufacturing, marketing, and administration. R&D is divided into PCB, software, and firmware whereas manufacturing consists of CAM and CNC machines. It sells specialized research devices with frequent customization inquiries, resulting in a rapid continuous development cycle based on order. The need for database arises in company's selling, cooperation, machine communication (Arduino and Raspberry Pi), machine learning, marketing, and A/S.

Lists of entities with attributes:

1. INVOICE(Invoice# (PK), Product ID (FK), Customer# (PK), BillOfMaterials# (FK), Price, Date)
2. PRODUCT[Product ID (PK), **Product Specification** (tip material, probe number, temperature range, software#, BillOfMaterials#), Inquiry# (FK), Customer number, Status]
3. INQUIRY(Customer# (PK, FK), Price, Deadline, customer requirements)
4. MARKETING(Product ID(PK), Customer# (PK), GoogleAds# (FK))

5. GOOGLEAD(GoogleAds#(PK), Statistics(Exposure, etc))
6. SOFTWARE(software# (PK), Division(PK))
7. DIVISION(DIVISION\_NAME(PK))
8. EMPLOYEE(EMP# (PK), EMP\_NAME(first, middle, last), Division)
9. Components(Components# (PK))
10. BillOfMaterials(Product ID (PK), Components# (PK))

Normalization:

1. INVOICE:
  - a. Created Product specification as a function of: inquiry and product\_base
  - b. BillOfMaterials are now dependent on specification,
  - c. Product ID is removed and added as Base\_Design\_Model under specification.

Now, Invoice, Specification, BillOfMaterials are single-valued, free of dependencies, and the primary key is the candidate key.

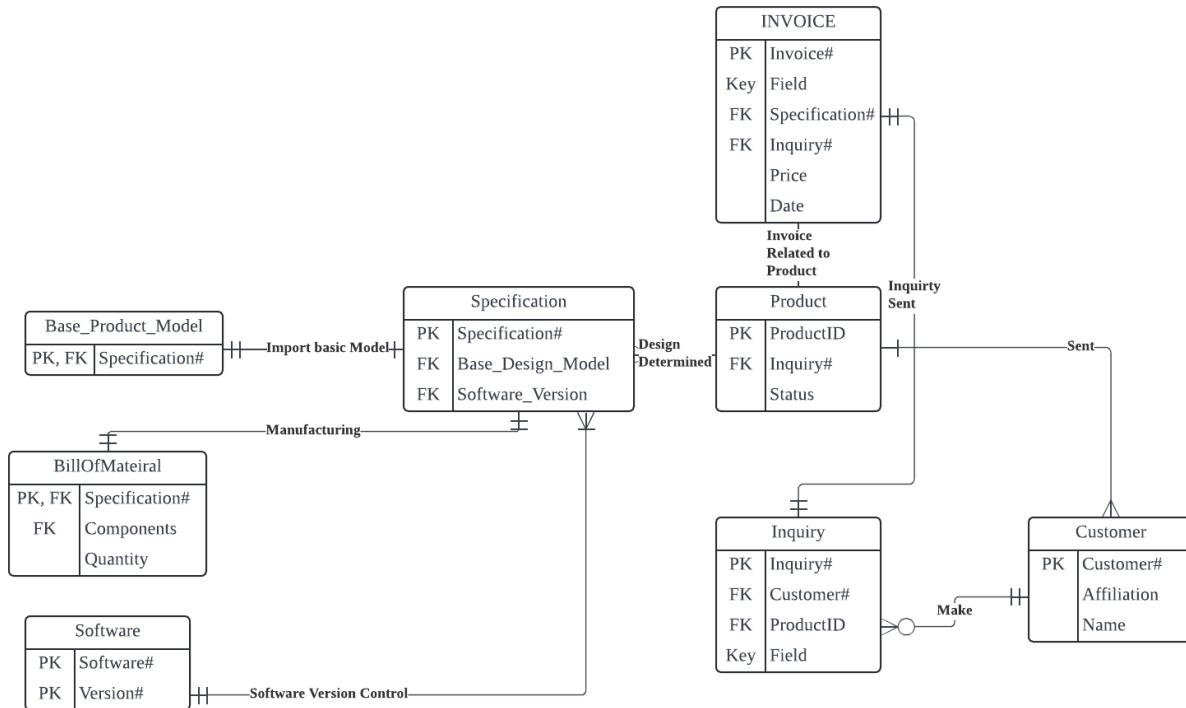
2. Product:
  - a. Product specification depends on consumer's request; it should have been taken into separate table, converted into a Foreign Key.
  - b. Similarly, software version and customer also depend on customer# and the invoice.
  - c. BillsOfMaterials as a child of Product Specification.
  - d. The customer number is moved to the Invoice table.

Now, the Product class is in NP3 for the same reasons as stated before.

3. INQUIRY:
  - a. Customer# is no longer a FK.
  - b. Ambiguous Price became Cus\_Suggested\_Price, and Cus\_suggest\_price can be multi-valued; taken into the same table and deadline.
  - c. Deadline is multi-valued as the deadline can frequently change. Separated into a different table.
  - d. Customer requirements are multi-valued, separated into a different table.
4. MARKETING: \\\commented out
5. GOOGLEAD: \\\ Commented out
6. SOFTWARE:
  - a. Updated Product\_Specification to have the software attribute.
7. DIVISION:
  - a. Manufacturing, programming, designing, etc.
8. EMPLOYEE: \\\ commented out
9. Components:
  - a. Added Manufacturer
10. BillOfMaterials:
  - a. Made into sub-class of prod\_specification.

Fixed:

1. INVOICE(Invoice# (PK), Specification#(FK), Customer# (FK), Price, Date)
  - a. Specification(Specification# (PK), Base\_Design\_Model(FK), software(FK))
    - i. DC, RF
    - b. BillOfMaterials(Specification# (PF, FK), material, quantity)
    - c. Base\_Design\_Model(Specification (PK, FK))
2. PRODUCT[Product ID (PK), Inquiry#(FK), Status]
3. INQUIRY(Customer# (PK), Cus\_Suggested\_Price, Deadline# (FK), specification(FK))
  - a. Deadline(Customer# (PK, FK), Deadline(PK), Cus\_Suggested\_Price(PK))
4. MARKETING(Product ID(PK), Customer# (PK), GoogleAds# (FK))
5. GOOGLEAD(GoogleAds#(PK), Statistics(Exposure, etc))
6. SOFTWARE(software# (PK), Division(PK))
7. DIVISION(DIVISION\_NAME(PK))
  - a. i.e. Java, Python, Arduino, CAD, Solidworks, etc.
8. EMPLOYEE(EMP# (PK), EMP\_NAME(Init, middle, last), Division)
9. Components(Components# (PK), Manufacturer#(FK))
  - a. Dino-lite
  - b. PogoPins
  - c. Petier Stage



## 11 Basic SQL

Learn basic SQL commands

Outcome	Weight
Structured Query Language (SQL)	◆◆◆◆◆

Learning MySql

Date	Author	Comment
2024/08/12 01:09	Issac Han	Ready to Mark
2024/08/13 06:06	Aditya Tarigopula	good work
2024/08/13 06:06	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Basic SQL

---

*Submitted By:*

Issac HAN  
s224509577  
2024/08/12 01:09

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Structured Query Language (SQL)	◆◆◆◆

Learning MySql

August 12, 2024



```

create database AssignmentWeek5;
show databases;
use AssignmentWeek5;

CREATE TABLE employee (
    employeeName CHAR(15) NOT NULL,
    street CHAR(15),
    city CHAR(10),
    PRIMARY KEY (employeeName)
);

CREATE TABLE works(
    employeeName CHAR(15) NOT NULL,
    companyName CHAR(15) NOT NULL,
    salary NUMERIC(7),
    PRIMARY KEY (employeeName, companyName),
    CHECK (salary >=0)
);

CREATE TABLE manages (
    employeeName CHAR(15) NOT NULL,
    managerName CHAR(15),
    PRIMARY KEY (employeeName)
);

CREATE TABLE company (
    companyName CHAR(15) NOT NULL,
    city CHAR(10) NOT NULL,
    PRIMARY KEY (companyName, city)
);

-- 1. Write a SQL query to retrieve names (displayed as "Employee Name") and salary of employees. [Relevant
-- table: Works]
select employeeName, salary from works;

-- 2. Write a SQL query to list name, street, and city of employees in descending order by their names. [Relevant
-- table: Employee]
select employeeName, street, city from employee order by employeeName desc;

-- 3. Write a SQL query to get a list of unique streets from the Employee table. [Relevant table: Employee]
select distinct street from employee;

-- 4. Write a SQL query to list all records in the works table in descending order of company names and within a
-- company in ascending order by employee name. [Relevant table: Works]
select* from works order by companyName desc, employeeName;

-- 5. Write a SQL query to list name and salary of all employees who work in Meyer and sort the records in
-- ascending order by their incomes. [Relevant table: Works]
select employeeName, salary from Works where companyName = "Meyer" order by salary;

-- 6. Assuming that the salary in the Works table is annual salary, write a SQL query to retrieve names (displayed
-- as "Employee Name") and monthly salary as "Monthly Salary" of employees. [Relevant table: Works]
select employeeName, salary / 12 as "Monthly Salary" from works;

-- 7. Write a SQL query to list names and salaries of all employees who work in Meyer and earn more than
-- 20000. [Relevant table: Works]
select employeeName, salary from works where companyName = "Meyer" and salary > 20000;

-- 8. Write a SQL query to list names and companies of the employees who earn in the range of 20000 to 25000
-- (inclusive). [Relevant table: Works]
select employeeName, companyName from works where 20000 <= salary <= 25000;

-- 9. Write a SQL query to list names of employees whose managers have "ll" (double ls) in their names.
-- [Relevant table: Manages]
select employeeName from manages where managerName like "%ll%";

-- 10. Write a SQL query to list company names and the average salary of their employees. [Relevant table:
-- Works]
select distinct companyName, avg(salary) from works group by companyName;

-- 11. Write a SQL query to list the name of the companies with average salary of employees more than or equal
-- to 20000. [Relevant table: Works]
select companyName from works group by companyName having avg(salary) >= 20000;

-- 12. Write a SQL query to select details of the employees who works in companies located in Rye. [Relevant
-- tables: Works and Company; Hint: use a subquery]
select employeeName from works natural join company where city = "Rye";

-- 13. Write a SQL query find the number of rows in the Manages table. [Relevant tables: Manages; Hint: use
-- COUNT()]
-- select * from manages;
select count(employeeName) from manages;

-- 14. Write a SQL query to find the name and company of the employee earning the highest salary. [Relevant
-- tables: Works; Hint: use a subquery using max() to find the highest salary. Please do not use 'WHERE
-- salary=25000' as it is the highest salary in this case. Hope you can understand that it is not possible to know
-- the highest value easily if there are millions of records. We want you to learn how to find it with a query.]
select employeeName, companyName from works where salary = (select max(salary) from works);

```

Output List:

1.

	employeeName	salary
▶	Adams	22000
	Curry	25000
	Hayes	19000
	Jones	21000
	Lindsay	9000
	Smith	22000
	Turner	20000
	Williams	18000

2.

	employeeName	street	city
▶	Williams	Nassus	Princeton
	Turner	Putname	Stamford
	Smith	North	Rye
	Lindsay	Park	Pittsfield
	Jones	Main	Harrison
	Hayes	Main	Harrison
	Curry	North	Rye
	Adams	Spring	Pittsfield
*	NULL	NULL	NULL

3.

	street
▶	Spring
	North
	Main
	Park
	Putname
	Nassus

4.

	employeeName	companyName	salary
▶	Hayes	Woolworths	19000
	Smith	Waltons	22000
	Jones	Tweeties	21000
	Williams	Tweeties	18000
	Adams	Meyer	22000
	Curry	Meyer	25000
	Lindsay	Meyer	9000
	Turner	Firebrand	20000
*	NULL	NULL	NULL

5.

	employeeName	salary
▶	Lindsay	9000
	Adams	22000
	Curry	25000

6.

	employeeName	Monthly Salary
▶	Adams	1833.3333
	Curry	2083.3333
	Hayes	1583.3333
	Jones	1750.0000
	Lindsay	750.0000
	Smith	1833.3333
	Turner	1666.6667
	Williams	1500.0000

7.

	employeeName	salary
▶	Adams	22000
	Curry	25000

8.

	employeeName	companyName
▶	Adams	Meyer
	Curry	Meyer
	Hayes	Woolworths
	Jones	Tweeties
	Lindsay	Meyer
	Smith	Waltons
	Turner	Firebrand
	Williams	Tweeties
*	NULL	NULL

9.

	employeeName
▶	Curry
	Hayes
	Jones
	Smith
*	NULL

10.

	companyName	avg(salary)
▶	Meyer	18666.6667
	Woolworths	19000.0000
	Tweeties	19500.0000
	Waltons	22000.0000
	Firebrand	20000.0000

11.

	companyName
▶	Waltons
	Firebrand

12.

	employeeName
▶	Adams
	Curry
	Lindsay
	Smith

13.

	count(employeeName)
▶	8

14.

	employeeName	companyName
▶	Curry	Meyer
*	NULL	NULL

## 12 Online Quiz 1

Complete Online Quiz 1 available on the unit site

Outcome	Weight
Reflection quiz	◆◆◆◆◆

Date	Author	Comment
2024/08/12 01:46	Issac Han	Ready to Mark
2024/08/13 06:02	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Online Quiz 1

---

*Submitted By:*

Issac HAN  
s224509577  
2024/08/12 01:46

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Reflection	♦♦♦♦

quiz

August 12, 2024



**Task 5.2C - Online Quiz 1**

Your work has been saved and submitted

Written 12 August, 2024 1:30 AM - 12 August, 2024 1:45 AM • Attempt 2 of 3

Attempt Score 19 / 20 - 95 %

Overall Grade (Highest Attempt) 19 / 20 - 95 %

[Done](#)

## 13 SELECT Queries with JOIN

Write different types of SELECT queries to extract required information from a database

Outcome	Weight
Fundamental concepts of database	◆◆◆◆◆

Join operators are one of the most fundamental set operators essential to database manipulation and creating normalization.

Outcome	Weight
Structured Query Language (SQL)	◆◆◆◇◇

I spent quite a time with ch7 of the book to figure out the syntax of MySQL.

Outcome	Weight
Reflection	◆◆◆◆◆

I spent quite a time with ch7 of the book to figure out the syntax of MySQL.

Outcome	Weight
Research and critical review	◆◆◆◇◇

I spent quite a time with ch7 of the book to figure out the syntax of MySQL.

Date	Author	Comment
2024/08/25 19:04	Issac Han	Ready to Mark
2024/09/01 15:47	Aditya Tarigopula	Good work
2024/09/01 15:47	Aditya Tarigopula	Complete

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## SELECT Queries with JOIN

---

*Submitted By:*

Issac HAN  
s224509577  
2024/08/25 19:04

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	◆◆◆◆
Relational Database Modelling	◆◆◆◆
Structured Query Language (SQL)	◆◆◆◆
Reflection	◆◆◆◆
Research and critical review	◆◆◆◆

I spent quite a time with ch7 of the book to figure out the syntax of MySQL.

August 25, 2024



Name: Isaac Han

Class: SIT 772

Assignment: Week6

P1:

```
1 -- Task1
2 -- Write a SQL statement to list invoices with their INV_NUMBER, INV_DATE along with the CUS_FNAME
3 -- and CUS_LNAME of the customers they belong to.
4
5 • use assignmentweek6;
6 • select INV_NUMBER, INV_DATE, CUS_FNAME, CUS_LNAME
7   from invoice join customer on invoice.CUS_CODE = customer.CUS_CODE;
```

Result Grid				
	INV_NUMBER	INV_DATE	CUS_FNAME	CUS_LNAME
▶	1001	2018-01-16 00:00:00	Myron	Orlando
	1002	2018-01-16 00:00:00	Leona	Dunne
	1003	2018-01-16 00:00:00	Kathy	Smith
	1004	2018-01-17 00:00:00	Leona	Dunne
	1005	2018-01-17 00:00:00	Anne	Farriss
	1006	2018-01-17 00:00:00	Myron	Orlando
	1007	2018-01-17 00:00:00	Amy	O'Brian
	1008	2018-01-17 00:00:00	Leona	Dunne

P2:

```

8
9    -- Task2
10   -- Write a SQL statement to list V_CODE and V_NAME along with the number of products they supply (column
11   -- name as 'NUM_PRODS') and the average price of product they supply (column name as
12   -- 'AVG_PROD_PRICE').
13
14 • select vendor.V_CODE, vendor.V_NAME, count(vendor.V_CODE) as NUM_PRODS, sum(product.P_PRICE) / count(vendor.V_CODE) as AVG_PROD_PRICE
15   from vendor join product on vendor.V_CODE = product.V_CODE
16   group by vendor.V_CODE;
17
18 • select * from products;
19 • select * from vendor;

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

The screenshot shows a database result grid with the following columns: V\_CODE, V\_NAME, NUM\_PRODS, and AVG\_PROD\_PRICE. The data is as follows:

V_CODE	V_NAME	NUM_PRODS	AVG_PROD_PRICE
21225	Bryson, Inc.	2	8.470000
21231	D&E Supply	1	8.450000
21344	Gomez Bros.	3	12.490000
23119	Randssets Ltd.	2	41.970000
24288	ORDVA, Inc.	3	155.593333
25595	Rubicon Systems	3	89.630000

Result 20 x Read Only

P3:

```

19
20   -- task 3
21   -- Write a SQL statement to list P_CODE and P_DESCRPT of all products along with their vendors' V_CODE
22   -- and V_NAME if available. Note that your results must include all products regardless of whether vendor
23   -- information is available or not. For products where vendor information is not available, V_CODE and
24   -- V_NAME columns will be empty or NULL in the result (Hint: use outer join).
25
26 • select P.P_CODE, P.P_DESCRPT, V.V_CODE, V.V_NAME
27   from product as P left outer join vendor as V on V.V_CODE = P.V_CODE;

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

The screenshot shows a database result grid with the following columns: P\_CODE, P\_DESCRPT, V\_CODE, and V\_NAME. The data is as follows:

P_CODE	P_DESCRPT	V_CODE	V_NAME
11QER/31	Power painter, 15 psi., 3-nozzle	25595	Rubicon Systems
13-Q2/P2	7.25-in. pwr. saw blade	21344	Gomez Bros.
14-Q1/L3	9.00-in. pwr. saw blade	21344	Gomez Bros.
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	23119	Randssets Ltd.
1558-QW1	Hrd. cloth, 1/2-in., 3x50	23119	Randssets Ltd.
2232/QTY	B&D jigsaw, 12-in. blade	24288	ORDVA, Inc.
2232/QWE	B&D jigsaw, 8-in. blade	24288	ORDVA, Inc.
2238/QPD	B&D cordless drill, 1/2-in.	25595	Rubicon Systems
23109-HB	Claw hammer	21225	Bryson, Inc.
23114-AA	Sledge hammer, 12 lb.	<b>NUL</b>	<b>NUL</b>
54778-ZT	Rat-tail file, 1/8-in. fine	21344	Gomez Bros.
89-WRE-Q	Hicut chain saw, 16 in.	24288	ORDVA, Inc.
PVC23DRT	PVC pipe, 3.5-in., 8-ft	<b>NUL</b>	<b>NUL</b>
SM-18277	1.25-in. metal screw, 25	21225	Bryson, Inc.
SW-23116	2.5-in. wd. screw, 50	21231	D&E Supply
WR3/TT3	Steel matting, 4x8'x1/6", .5" m...	25595	Rubicon Systems

P4:

```

40
29
30    -- task 4
31    -- Write a SQL statement to retrieve INV_DATE and the list of products with P_CODE and P_DESCRIFT of
32    -- the invoice with INV_NUMBER = 1008 (Hint: you may have to join three tables).
33
34 • select I.INV_DATE, P.P_CODE, P.P_DESCRIFT
35   from product as P join Line as L on P.P_CODE = L.P_CODE join invoice as I on I.INV_NUMBER = L.InV_NUMBER
36   where I.INV_NUMBER = 1008;

```

Result Grid			
	INV_DATE	P_CODE	P_DESCRIFT
▶	2018-01-17 00:00:00	PVC23DRT	PVC pipe, 3.5-in., 8-ft
	2018-01-17 00:00:00	WR3/TT3	Steel matting, 4x8'x1/6", .5" mesh
	2018-01-17 00:00:00	23109-HB	Claw hammer

P5:

```

37
38    -- task 5
39    -- Write a SQL statement to list full names (by concatenating EMP_FNAME and EMP_LNAME separated by
40    -- a space) of employees AS 'Employee' with their managers' full name (again by concatenating
41    -- EMP_FNAME and EMP_LNAME separated by a space) AS 'Manager' (Hint: recursive/self join and string
42    -- concatenation).
43
44 • select concat(E.EMP_FNAME, E.EMP_LNAME) as Employee, concat(M.EMP_FNAME, M.EMP_LNAME) as Manager
45   from emp E join emp M on E.EMP_MGR = M.EMP_NUM;
46
47 • select * from emp;

```

Result Grid	
Employee	Manager
RhondaLewis	GeorgeKolmycz
RhettVandam	GeorgeKolmycz
AnneJones	GeorgeKolmycz
JohnLange	RobertWilliams
JeanineSmith	RobertWilliams
JorgeDante	RobertWilliams
GeorgeSmith	PaulWiesenbach
LeighaGenkazi	PaulWiesenbach
RupertWashington	RobertWilliams
EdwardJohnson	GeorgeKolmycz
MelanieSmythe	RobertWilliams
MarieBrandon	PaulWiesenbach
HermineSaranda	RobertWilliams
GeorgeSmith	PaulWiesenbach

P6:

```
48
49      -- task 6
50      -- Write a SQL statement to list P_CODE and P_DESCRIP of products that came in store (INDATE) in the
51      -- month of February of 2018.
52
53 •  select P_CODE, P_DESCRIP
54   from product
55   where year(P_INDATE) = 2018 and month(P_INDATE) = 2;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
P_CODE	P_DESCRIP				
89-WRE-Q	Hicut chain saw, 16 in.				
PVC23DRT	PVC pipe, 3.5-in., 8-ft				
SW-23116	2.5-in. wd. screw, 50				
NULL	NULL				
*					

## 14 Other DML and DDL Queries

Write other data manipulation and data definition queries

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Data definition and manipulations are fundamental steps in database management.

Outcome	Weight
Research and critical review	♦♦◊◊◊

I studied with online tutorials to complete the tasks.

Outcome	Weight
Relational Database Modelling	♦♦◊◊◊

There wasn't a strong relational modeling besides a few potential join operations.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Manually creating data and defining the characteristics of rows are essential in SQL.

Date	Author	Comment
2024/09/06 21:28	Issac Han	Ready to Mark
2024/09/08 16:03	Aditya Tarigopula	good work
2024/09/08 16:03	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Other DML and DDL Queries

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/06 21:28

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	◆◆◆◆

Learning Basic Syntax of mySQL

September 6, 2024



Name: Isaac Han

Class: SIT 772

Q1:

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below the toolbar, the SQL editor contains the following SQL script:

```
4 ●    create database SIT772_DDL;
5 ●    show databases;
6 ●    use SIT772_DDL;
7
8    -- Write an SQL statement create the EMPLOYEE table with the following attributes and constraints.
9    -- EMP_NUM (PK) number,
10   -- EMP_TITLE Characters of length up to 5,
11   -- EMP_LNAME Characters of length up to 25 (can't be NULL),
12   -- EMP_FNAME Characters of length up to 25 (can't be NULL),
13   -- EMP_INITIAL Only one character, (can't be NULL)
14   -- EMP_DOB Date,
15   -- EMP_HIRE_DATE Date,
16   -- EMP_AREACODE Characters of length 3 (default value of '001')
17   -- EMP_PHONE Characters of length 8,
18   -- EMP_MGR number
19
20 ●  create Table EMPLOYEE (
21      EMP_NUM numeric primary key,
22      EMP_TITLE varchar(5),
23      EMP_LNAME varchar(25) Not Null,
24      EMP_FNAME varchar(25) Not Null,
25      EMP_INITIAL CHAR(1) Not Null,
26      EMP_DOB DATE,
27      EMP_HIRE_DATE DATE,
28      EMP_AREACODE char(3) Default "001",
29      EMP_PHONE char(8),
30      EMP_MGR numeric
31 );
32
33 ●  show tables;
34 ●  DESC EMPLOYEE;
35
```

Below the SQL editor is the Result Grid, which displays the structure of the EMPLOYEE table:

Field	Type	Null	Key	Default	Extra
EMP_NUM	decimal(10,0)	NO	PRI	NULL	
EMP_TITLE	varchar(5)	YES		NULL	
EMP_LNAME	varchar(25)	NO		NULL	
EMP_FNAME	varchar(25)	NO		NULL	
EMP_INITIAL	char(1)	NO		NULL	
EMP_DOB	date	YES		NULL	
EMP_HIRE_DATE	date	YES		NULL	
EMP_AREACODE	char(3)	YES		001	
EMP_PHONE	char(8)	YES		NULL	
EMP_MGR	decimal(10,0)	YES		NULL	

Q2:

```
32  
33 • show tables;  
34 • DESC EMPLOYEE;  
35  
36 -- Write an SQL statement to change the column name 'EMP_MGR' to 'EMP_MGR_NUM'. [Hint: ALTER  
37 -- TABLE to rename column, discussed in the class].  
38 -- As your answer to this question, please include (i) your SQL statement to alter the EMPLOYEE table, and  
39 -- (ii) a screenshot of the result of the following SQL statement:  
40 -- DESC EMPLOYEE;  
41  
42 • alter Table EMPLOYEE rename Column EMP_MGR to EMP_MGR_NUM;  
43 • desc EMPLOYEE;  
44  
45
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>					
Field	Type	Null	Key	Default	Extra
EMP_LNAME	varchar(25)	NO		HULL	
EMP_FNAME	varchar(25)	NO		HULL	
EMP_INITIAL	char(1)	NO		HULL	
EMP_DOB	date	YES		HULL	
EMP_HIRE_DATE	date	YES		HULL	
EMP_AREACODE	char(3)	YES		001	
EMP_PHONE	char(8)	YES		HULL	
EMP_MGR_NUM	decimal(10,0)	YES		HULL	

Result 9

Q3:

```
44  
45 -- Write an SQL statement to ADD FOREIGN KEY CONSTRAINT to EMP_MGR_NUM to reference to  
46 -- EMP_NUM (self-reference). This column indicates who is the manager (EMP_NUM of the manager) of an  
47 -- employee. [Hint: ALTER TABLE to add constraint, discussed in the class]  
48  
49 • alter table EMPLOYEE  
50   add constraint FK_MANAGER  
51   foreign key (EMP_MGR_NUM)  
52   references EMPLOYEE (EMP_NUM);  
53  
54 • select COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_COLUMN_NAME,  
55   REFERENCED_TABLE_NAME from information_schema.KEY_COLUMN_USAGE where TABLE_NAME  
56 = 'EMPLOYEE';
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>			
COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_COLUMN_NAME	REFERENCED_TABLE_NAME
employeeName	PRIMARY	HULL	HULL
EMP_NUM	PRIMARY	HULL	HULL
employeeName	PRIMARY	HULL	HULL
EMP_NUM	PRIMARY	HULL	HULL
EMP_MGR_NUM	FK_MANAGER	EMP_NUM	employee
employeeName	PRIMARY	HULL	HULL

Result 10

Q4:

```

58 • truncate table EMPLOYEE;
59 • -- Write SQL statements to insert the following records in the EMPLOYEE TABLE.
60 • -- Insert all employees from the table
62
63 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
64 VALUES (100, 'Mr', 'Kolmycz', 'George', 'D', '1942-06-15', '1985-03-15', '615', '324-5456', NULL);
65 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
66 VALUES (101, 'Ms', 'Lewis', 'Rhonda', 'G', '1965-03-19', '1986-04-25', '615', '324-4472', 100);
67 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
68 VALUES (102, 'Mr', 'Vandam', 'Rhett', 'N', '1958-11-14', '1990-12-20', '901', '675-3456', 100);
69 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
70 VALUES (103, 'Ms', 'Jones', 'Anne', 'M', '1974-10-16', '1994-08-28', '615', '898-3456', 100);
71 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
72 VALUES (104, 'Mr', 'Lange', 'John', 'P', '1971-08-08', '1994-10-20', '901', '504-4430', 100);
73 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
74 VALUES (105, 'Mr', 'Williams', 'Robert', 'D', '1975-03-14', '1998-11-08', '615', '890-3220', NULL);
75 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
76 VALUES (106, 'Mrs', 'Smith', 'Jeanine', 'K', '1968-02-12', '1995-08-16', '615', '324-7883', 105);

```

EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR_NUM
100	Mr	Kolmycz	George	D	1942-06-15	1985-03-15	615	324-5456	NULL
101	Ms	Lewis	Rhonda	G	1965-03-19	1986-04-25	615	324-4472	100
102	Mr	Vandam	Rhett	N	1958-11-14	1990-12-20	901	675-3456	100
103	Ms	Jones	Anne	M	1974-10-16	1994-08-28	615	898-3456	100
104	Mr	Lange	John	P	1971-08-08	1994-10-20	901	504-4430	100
105	Mr	Williams	Robert	D	1975-03-14	1998-11-08	615	890-3220	NULL
106	Mrs	Smith	Jeanine	K	1968-02-12	1995-08-16	615	324-7883	105
107	Mr	Diantre	Jorge	D	1974-08-21	1994-07-02	615	890-4567	105
108	Mr	Wiesenbach	Paul	R	1966-02-14	1992-11-18	615	897-4358	NULL
109	Mr	Smith	George	K	1961-06-18	1996-04-14	901	504-3339	108
110	Mrs	Genkazi	Leighla	W	1970-05-19	1990-12-01	901	569-0093	108
111	Mr	Washington	Rupert	E	1966-01-09	1999-06-21	615	324-9981	108
112	Mr	Johnson	Edward	E	1961-05-14	1996-12-01	615	324-8899	108
113	Ms	Smythe	Melanie	P	1970-09-15	1999-05-11	615	324-9006	108
114	Ms	Brandon	Marie	O	1956-11-20	1979-11-15	901	882-0845	108
115	Mrs	Saranda	Hermine	R	1972-07-25	1998-04-23	615	324-5505	105
116	Mr	Smith	George	A	1965-11-09	1988-12-10	615	890-2984	108

EMPLOYEE 13 x

```

79 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
80 VALUES (108, 'Mr', 'Wiesenbach', 'Paul', 'R', '1966-02-14', '1992-11-18', '615', '897-4358', NULL);
81 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
82 VALUES (109, 'Mr', 'Smith', 'George', 'K', '1961-06-18', '1996-04-14', '901', '504-3339', 108);
83 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
84 VALUES (110, 'Mrs', 'Genkazi', 'Leighla', 'W', '1970-05-19', '1990-12-01', '901', '569-0093', 108);
85 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
86 VALUES (111, 'Mr', 'Washington', 'Rupert', 'E', '1966-01-09', '1999-06-21', '615', '324-9981', 108);
87 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
88 VALUES (112, 'Mr', 'Johnson', 'Edward', 'E', '1961-05-14', '1996-12-01', '615', '324-8899', 108);
89 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
90 VALUES (113, 'Ms', 'Smythe', 'Melanie', 'P', '1970-09-15', '1999-05-11', '615', '324-9006', 108);
91 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
92 VALUES (114, 'Ms', 'Brandon', 'Marie', 'O', '1956-11-20', '1979-11-15', '901', '882-0845', 108);
93 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
94 VALUES (115, 'Mrs', 'Saranda', 'Hermine', 'R', '1972-07-25', '1998-04-23', '615', '324-5505', 105);
95 • INSERT INTO EMPLOYEE (EMP_NUM, EMP_TITLE, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB, EMP_HIRE_DATE, EMP_AREACODE, EMP_PHONE, EMP_MGR_NUM)
96 VALUES (116, 'Mr', 'Smith', 'George', 'A', '1965-11-08', '1988-12-10', '615', '890-2984', 108);
97 • SELECT * FROM EMPLOYEE;

```

Q5:

```
98
99 -- Write an SQL statement to set the manager of Paul Wiesenbach (EMP_NUM = 108) as George Kolmycz
100 -- (EMP_NUM = 100
101
102 • update EMPLOYEE
103   set EMP_MGR_NUM = 100
104   where EMP_NUM = 108;
105
106 • SELECT * FROM EMPLOYEE WHERE EMP_NUM = 108;
107
108
```

Result Grid		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
		EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR_NUM
▶	*	108	Mr	Wiesenbach	Paul	R	1966-02-14	1992-11-18	615	897-4358	100
		NULL	NULL	NULL	NULL	HULL	NULL	NULL	NULL	NULL	HULL

Q6:

```
108 -- Write an SQL statement to delete records of all employees managed by Paul Wiesenbach (EMP_NUM =  
109 -- 108).  
110 • DELETE FROM EMPLOYEE  
111 WHERE EMP_MGR_NUM = 108;  
112 • SELECT * FROM EMPLOYEE;
```

## 15 Research Report and Presentation

Prepare a research report on one emerging trend or issue related to database.

Outcome	Weight
Fundamental concepts of database	◆◆◆◇◇

A very challenging assignment to explore.

Outcome	Weight
Relational Database Modelling	◆◆◆◆◆

A very challenging assignment to explore.

Outcome	Weight
Reflection	◆◆◆◆◆

A very challenging assignment to explore.

Outcome	Weight
Research and critical review	◆◆◆◆◆

A very challenging assignment to explore.

Date	Author	Comment
2024/09/13 23:44	Issac Han	Ready to Mark
2024/09/13 23:57	Issac Han	In overleaf, conclusion part, I includded why this project is important for the company that I chose.
2024/09/13 23:57	Issac Han	And here is the link for the video.
2024/09/13 23:59	Issac Han	<a href="https://youtu.be/rN98cFdErHQ">https://youtu.be/rN98cFdErHQ</a>
2024/09/23 06:05	Aditya Tarigopula	good work
2024/09/23 06:05	Aditya Tarigopula	Complete

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Research Report and Presentation

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/13 23:44

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

A very challenging assignment to explore.

September 13, 2024



**Research: Machine Learning in Probing Station**

Using the Database to feed Yolo object detection and coordinate

**Author: Isaac Han**

**Class: SIT772**

A thesis presented for the class SIT772

Deakin University  
September 13, 2024

## 1 Abstract

As computational power becomes cheaper and more accessible, the applications of computational agents and the demand for processing capabilities continue to grow. The sheer strength of computational resources has proven advantageous in a variety of areas: brute-force algorithms for solving least-time searches, robot pathfinding, and even cryptography are just a few examples highlighting the importance of computational power. In recent years, the need for enhanced computational ability has become even more critical with the rise of artificial intelligence (AI). Contrary to its name, AI does not involve intelligence, but rather learning algorithms. These algorithms have shown themselves to be versatile and applicable across a wide range of fields, as many students might have experienced firsthand with tools like ChatGPT. The potential benefits of machine learning are now more apparent than ever.

However, the advantages of machine learning come with costs—specifically, the need for greater computational power and large datasets to train these algorithms effectively. In my work at Nextron, a company that manufactures probing stations used to test the electrical, material, and magnetic properties of materials, I encountered a particular industrial challenge. As I developed software prototypes to control this research equipment, I identified a major opportunity: automating the probing process. This automation would increase operational efficiency and reduce the burden on operators.

Such automation relies heavily on image processing techniques. During the design of these procedures, two data-intensive algorithms stood out. The first is the YOLO object detection algorithm, and the second is a regression technique used to correct image distortions caused by heat and camera angles. This paper will explore the role of YOLO object detection and regression techniques, focusing on their reliance on large datasets and computational power.

## 2 Introduction

The overall procedure involves feeding real-time camera images to guide and correct the probing motion, utilizing computational agents for vector calculations and image training, and ultimately employing the Piezo program for precise motion control based on the received coordinate data.

List of the overview:

1. Camera Image Feeding
2. Computer Workspace
3. Image Processing
4. Piezo Program

## 2.1 Camera Image Feeding

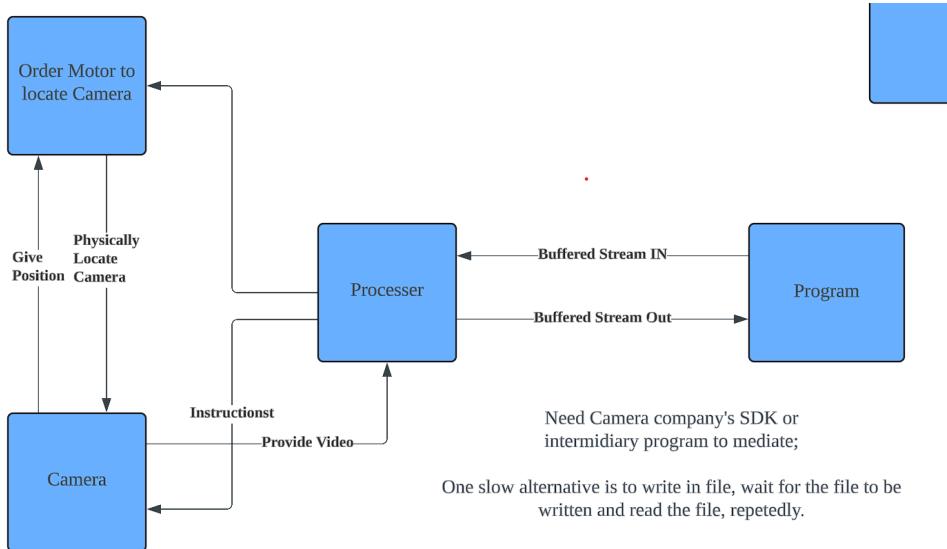


Figure 1: Diagram showing the interactions between camera and computer, from MATLAB [1]

The diagram highlights the critical need for separating the processor and the program. In this setup, the program sends queries to the processor, which returns the camera images as a buffered stream output. Managing the incoming data efficiently is essential, and it falls upon the program's protocol to decide when and how to flush out excess buffered stream input. A common approach is to clear the buffer once it accumulates enough data to ensure that the most recent frame from the camera is being processed, thus optimizing real-time responsiveness.

## 2.2 Computer Work Space

This is the stage where the computer processes and outputs the essential characteristics required for image processing. The system may incorporate a graphical user interface (GUI) for ease of interaction, allowing users to save and read files, and manage pre-trained models. These pre-trained models are vital for tasks such as object detection and correction algorithms, ensuring that the image processing is both efficient and accurate. The integration of these components enables the user to fine-tune the system, adapting it to various probing tasks or environments.

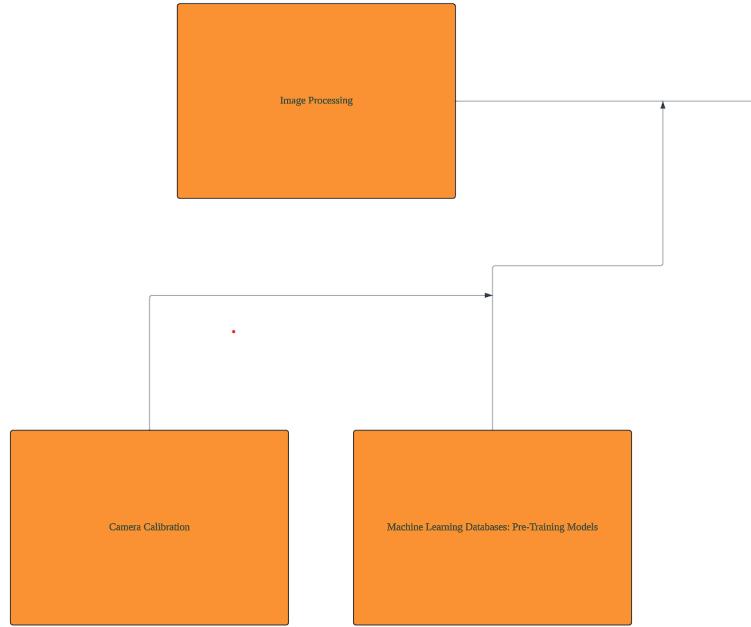


Figure 2: Preliminary Image Processing Steps. Created by the author.

### 2.3 Image Processing

This preliminary stage involves determining the extrinsic and intrinsic parameters of the camera, along with the trained model. Geometric camera calibration is performed at this layer and is essential for tasks such as perspective transformation and world coordinate mapping. The camera matrix is defined as follows:

$$w \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where w represents the scale factor, (x, y) represents the image coordinate, P is the camera matrix(projection of 3D into 2D image), and (X, Y, Z) are the real-world coordinates.

The Camera Matrix(a projection matrix) can be further represented by a linear transformation of matrixes.

$$P = K \cdot [Rt]$$

where P represents the camera matrix, K is an intrinsic matrix, and Rt is for Extrinsic (rotations and translation) [2] [1].

## 2.4 Image Processing: Main1

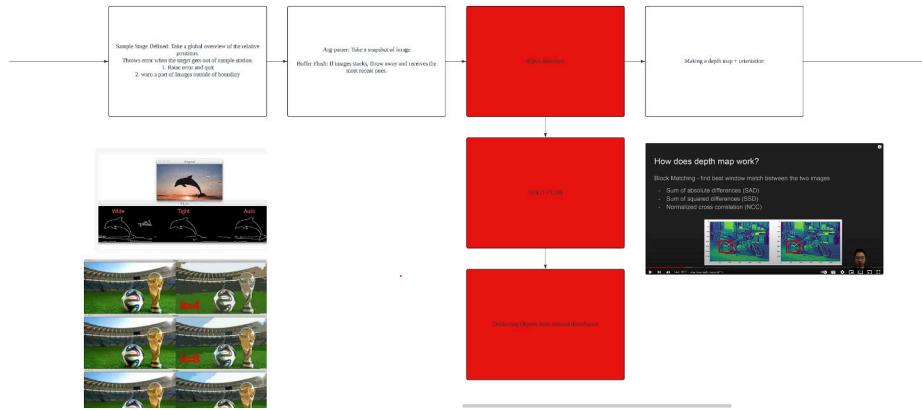


Figure 3: The main Image Processing Steps. Some images are taken from Kelvin Wood's YouTube channel. [3].

Applying filters and checking the readability of the image from the skimage.exposure function. Edge detection and segmentation create the object to be detected by the YOLO object detection.

## 2.5 Image Processing: Main2

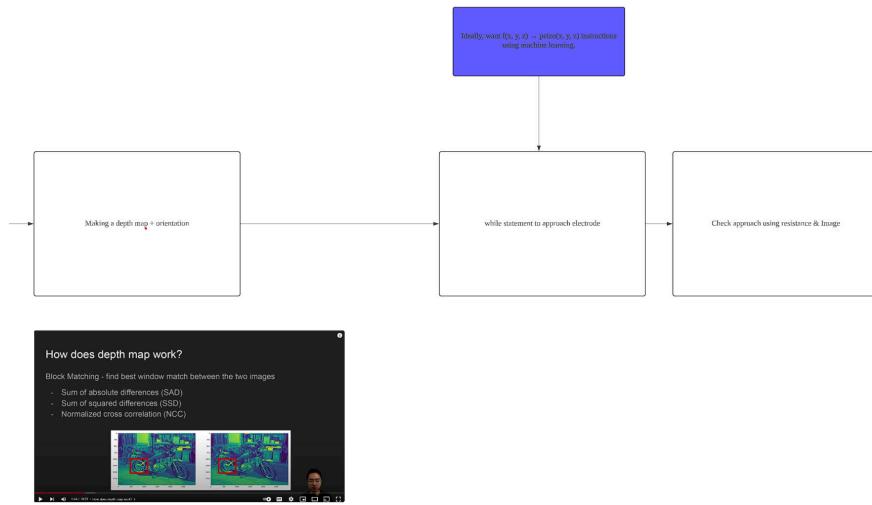


Figure 4: The main Image Processing Steps. Some images are taken from Kelvin Wood's YouTube channel. [3].

After the Yolo object detection layers detect the object and destination, I want to determine the direction and the size of the vector to move.

## 2.6 Piezo Actuator and Signal Distance Estimation

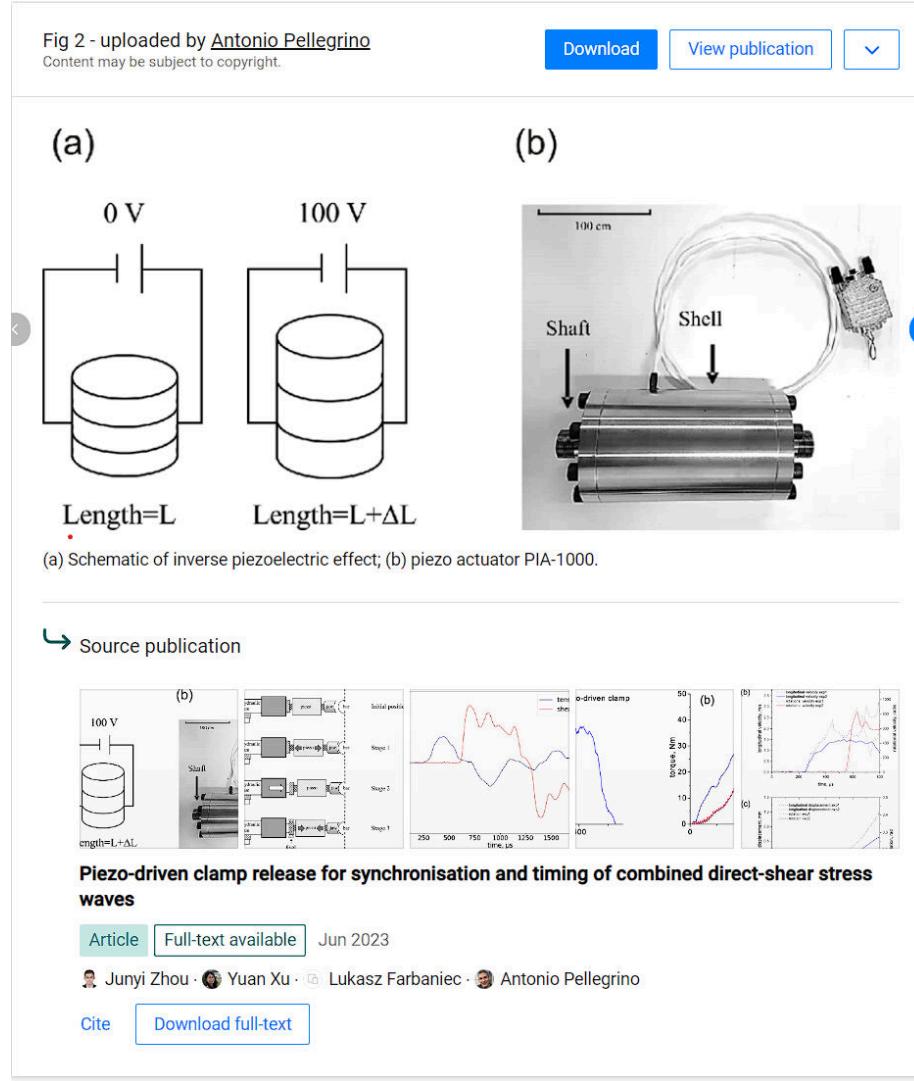


Figure 5: Piezo Actuator [4].

The simplified working mechanism of the Piezo actuator is illustrated in the picture. When an external voltage is applied to the Piezo material, it undergoes the Piezoelectric effect, which causes the material to deform or bend. When

Piezo materials are connected in series and voltage is applied across them, their total length either increases or decreases. Due to the high precision of Piezo materials, which are governed by physical laws, their responses are rapid and involve no moving parts, making them ideal for precise guidance.

However, guiding probing tips using Piezo actuators presents a challenge: the extent of expansion or contraction of the Piezo material is not solely dependent on the applied voltage. The behavior of the Piezo actuator can vary with different environmental conditions, such as pressure and temperature. Consequently, to accurately guide probing tips to their destination, it is essential to implement a control logic that adjusts the signal strength based on parameter estimation.

### 3 Main

Again, the overall process of the autonomous probing guiding consists of image feeding, computer processing, image processing, and contacting the API of the software that moves Piezo actuators. Among those steps, some are particularly heavy with data and therefore, require data management. One is the YOLO object detection, another is noise reduction, and the last is parameter estimation and learning.

#### 3.1 Fundamentals of Object Detection

R-CNN, a recurrent convoluted neural network, is an algorithm that utilizes a kernel (a feature patch) to detect whether the desired feature can be found in the local viewing window or not.

Unlike traditional feature mapping, R-CNN's local pooling enables finding desired features in the local frame, which helps to detect objects in the presence of background noises, different light contrasts, and scale variation.

Feeding correct data is important; there are variations in subtypes (like Chihuahua and Golden Retriever), lights, viewing angle distortions, and object deformations. To encounter such variations and correctly assign the object type, feeding correctly pre-process data is important.

The diagram Figure: 7 explains the importance of data; even in the most simple box model with linearly separable examples, the absence of data (the gap between negatives and positives) creates the shaded region to be undetermined.

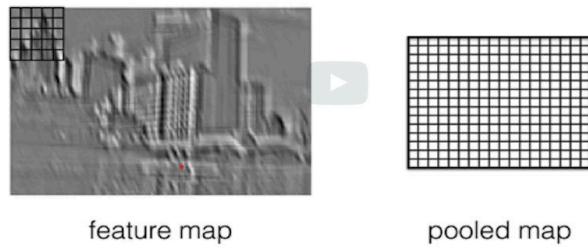
Additionally, adding data outside of the shaded region or inside the positive sample does not improve the quality of the classification algorithm. Only additional data within the shaded region would improve the accuracy of the algorithm. In modern machine learning algorithms where it can actively request the data, it would request samples within shaded regions.

It is often the case that hand-written feature maps are not readily available. Humans can distinguish the building in Figure: 6 but are unable to provide a discriminant that distinguishes buildings from all others. Machine learning might be used to automatically extract the feature maps; it is not magic that



## Pooling

- We wish to know whether a feature was there but not exactly where it was



feature map

pooled map

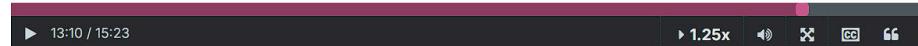


Figure 6: Principle of Image Detection: Pooling [5].

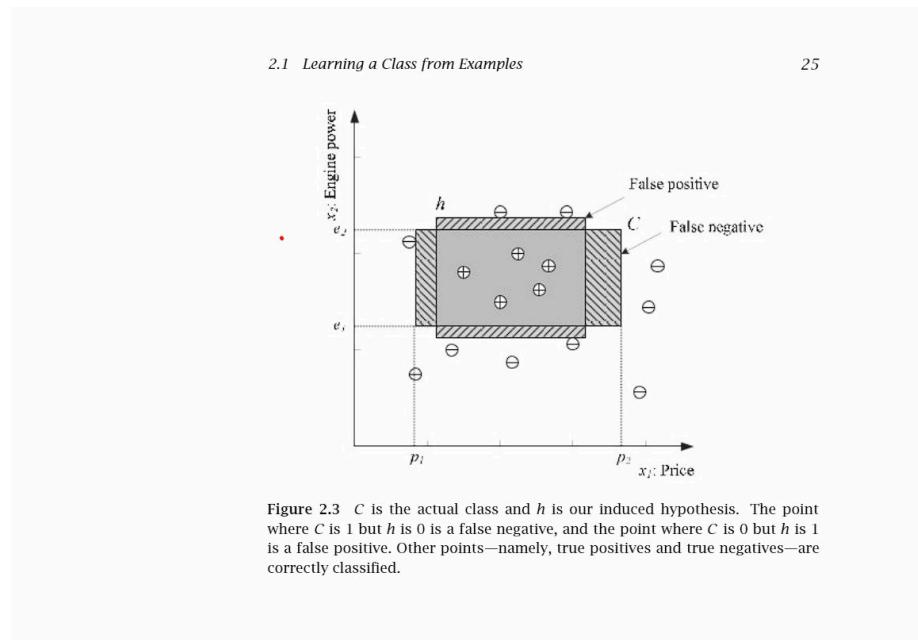


Figure 7: Diagram Explaining the importance of data [6].

provides the feature map but an abundance of data and processing power that tests candidate discriminates.

### 3.2 Fundamental of Object Detection - Detailed

The specifics of algorithms mentioned previously can be summarized as follows;  
**Overview:**

1. Kernels
2. Sliding Window
3. Image Pyramids
4. Non-maxima suppression

Taken from A. Rosebrock [7].

#### Mechanisms

1. Kernels: Kernels, also known as convolution matrix or mask, is a small matrix used for convolution calculation. Based on the characteristics of a matrix, it can perform different tasks. [8]
2. Sliding Window: The kernel gets sweep from left-to-right and top-to-bottom, classifying each Region of Interest(ROI) along the way. This helps detect objects in noisy and obscured environments.
3. Image Pyramid: This is the process where images are convoluted and reduced in size. Finally, the image may be flattened for final classification.
4. Non-maxima suppression: Having a small sliding window means sweeping parts of the same object multiple times. This creates detecting objects multiple times. It is necessary to apply non-maxima suppression to keep the most probable one.

### 3.3 Data in object detection

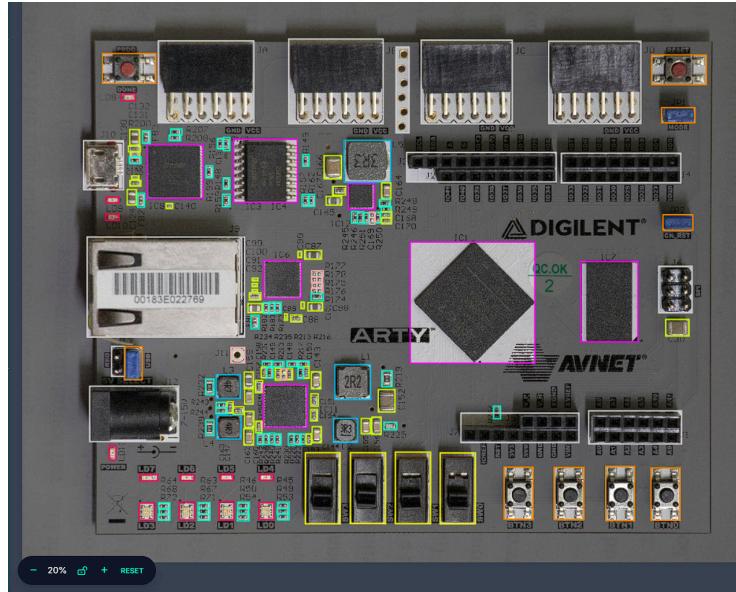


Figure 8: Pre-trained Yolo model from Roboflow [9].

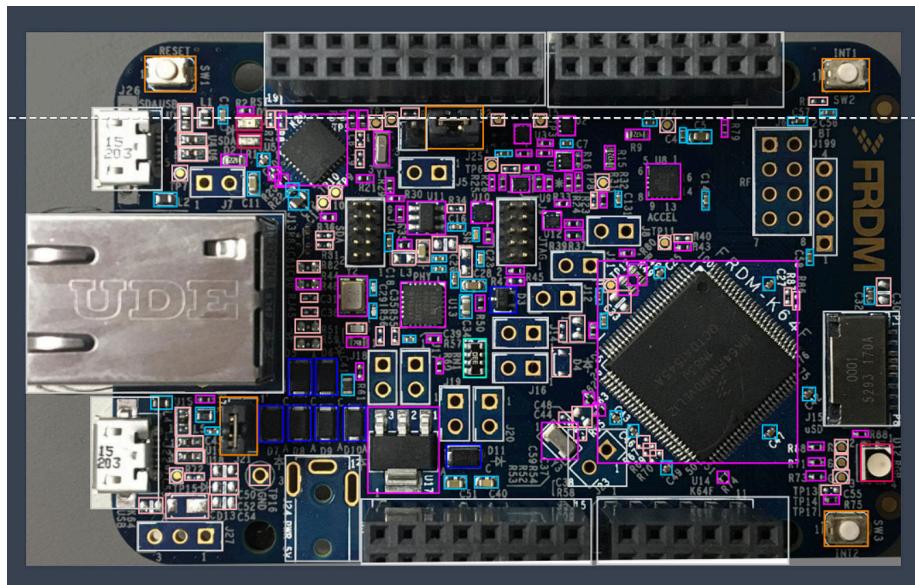


Figure 9: Pre-trained Yolo model from Roboflow [9].

Typical Yolo Models are trained with images, surrounding boxes, and image labels. Multiple discriminants require significant amounts of training data. The coordinate data for the viewing angle might be also considered if the viewing angle is inconsistent and obscured.

As it is typically recommended to add a few hundred images for each category in multi-class classification, making the data set would take a significant amount of time to even collect data, perform perspective transformation, mark surrounding boxes, and label each box, it is desirable to search a data set online. Even though there is no direct correspondence, a similar study was PCB components detections.

### 3.4 Denoising

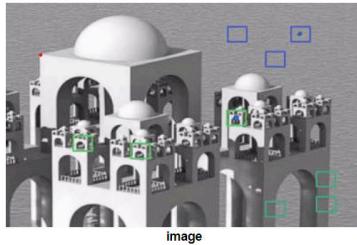


Figure 10: Denoising Example1 [10].

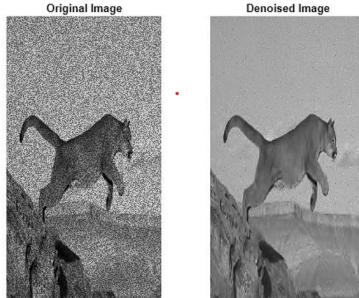


Figure 11: Denoising Example2 [11].

The presence of noise may be of concern for classification accuracy. In the case of the probing chamber, the temperature can range from  $-196^{\circ}\text{C}$  to about  $1000^{\circ}\text{C}$ . Meaning, a visual distortion may be created due to temperature gradient. Similar visual distortion may be created due to the camera being out of focus. To distinguish the difference between the out-of-focus objects and the image distortion created by temperature, applying a blurring filter and correcting for noise can improve the accuracy of detecting a real-time object.

Since there is a single heat source, the Gaussian noise reduction technique in the Region of Interest may suppress the image distortion caused by the temperature. To achieve such goals, it is necessary to estimate the parameters of the Gaussian noise,  $\mu$  and  $\sigma^2$ , from data.

### 3.5 Polar Coordinate

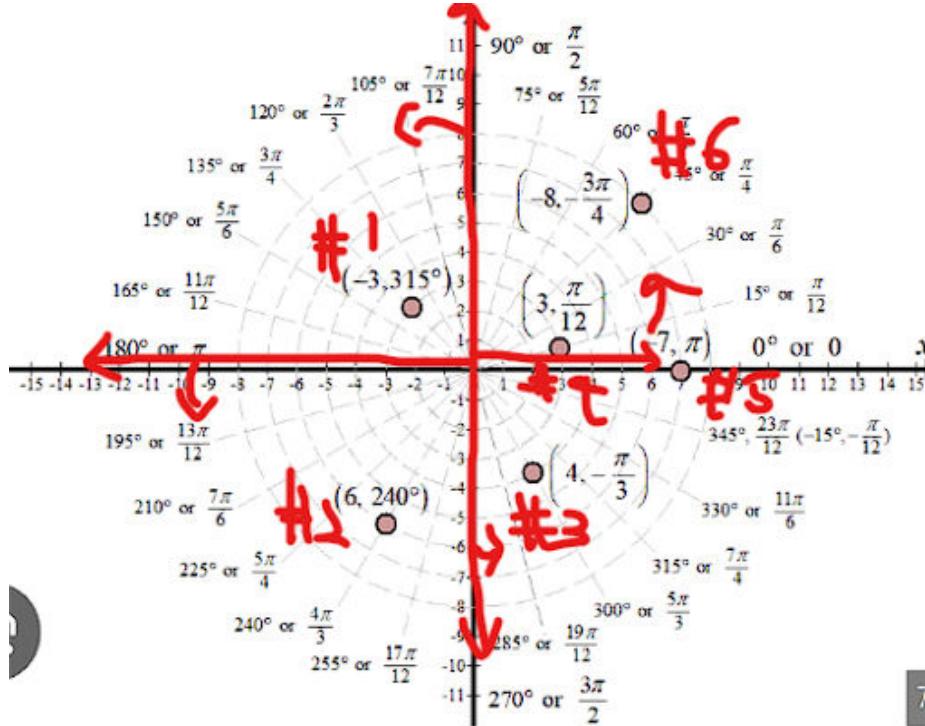


Figure 12: Numbering the order of contact using polar coordinate

There are four probes and four contacts to be made. One particular problem is mapping a probe to an electrode without crossing over. One algorithm that I thought was to number electrodes in order of angle and radius from the center. Figure: ?? demonstrates the ordering. Draw a line and rotate the line to linearize the ordering. Using a stack, the last in is the first out to prevent tips from being cross-covered.

## 4 Data in ML

Data is important

## 5 Main Discussion - the product



Figure 13: Nextron Piezo Micro-probing System. The picture was taken by Isaac(the author) and permission to use was granted on Sep 13, 2024.

The Micro Probing Station by Nextron is a commercial product designed for manual probing. In its basic configuration, the connection between the sample and the probes is achieved manually by human operators. This process involves pressing a spring and releasing it to establish contact. However, the limitation of manual probing lies in its usability, as it is effective only with samples that have large electrodes. When it comes to smaller electrodes or nanoscale measurements, manual probing is no longer viable, prompting the need for a more precise solution. To address these limitations, Nextron has equipped its probing station with Piezo actuators.

Piezoelectric material has the unique property of bending when subjected to an external electric potential, and this characteristic can be exploited to control the probing tips with nanoscale precision. Piezo actuators are renowned for their ability to move with extreme consistency, enabling highly accurate control of probe tips. However, the advanced Piezo modules capable of this precision are expensive, often costing at least three times as much as the more basic Piezo modules currently used in Nextron's system. In opting for the more affordable Piezo modules, Nextron sacrifices some of the precision and consistency. Specifically, these lower-cost modules do not offer consistent movement, as they possess

a frequency cycle and exhibit an undetermined travel distance in response to external voltage. The travel speed of these modules can also vary, even when the applied voltage remains constant.

To reduce production costs and still enhance functionality, one potential solution is to control the probing tips using image processing rather than relying solely on the absolute coordinate system. By integrating image processing, it may be possible to compensate for the imprecise movement of the Piezo actuators. Although full automation through image processing may not be achievable, semi-automation would still represent a significant improvement. Currently, human operators are required to manually control the microscope and the probing tips via joystick-like controls. If the field of view of the microscope does not cover the probing tips and the sample simultaneously, the operator must manually adjust the microscope—a task that can become burdensome and introduce potential sources of error into the experiment.

In contrast, large corporations such as Samsung, which operates in the semiconductor industry, have developed fully autonomous probing stations at a much larger scale. These advanced systems can cost up to \$10 million, making them inaccessible for many smaller entities. Ph.D. students, university departments, and small research laboratories often lack the necessary funding to purchase such expensive equipment, leaving them in search of alternatives. For these groups, a small, affordable, and semi-automated probing station could offer significant benefits. Not only would this help them to produce prototypes for their research, but it could also assist in convincing investors to support their projects. In some cases, these affordable alternatives might suffice for research needs, while in other cases, they could act as a stepping stone toward the eventual acquisition of more advanced machinery.

At its core, a probing station is a measurement device. When connected to instruments such as the Keithley 2450, it can measure a variety of electrical characteristics, including resistance, voltage, and current. Additionally, if connected to environmental control devices like the FB100 or Alicat, the station can regulate the conditions surrounding the sample, including temperature, pressure, and humidity. This flexibility makes the probing station an invaluable tool for measuring samples under specialized conditions. Moreover, the compact nature of the station, combined with its ability to operate in a small chamber, allows for the adjustment of environmental variables even under extreme conditions.

Given these factors, the potential for semi-automation via image processing in a low-cost probing station could bridge the gap between large-scale industrial automation and the more modest needs of academic or research-based environments. This solution would empower smaller teams to engage in cutting-edge research without the prohibitive costs of high-end systems, opening the door for innovation and discovery in a more accessible manner.

## 6 Conclusion

Machine learning, with its growing accessibility and increasing computational power, has transformed numerous industries, offering innovative solutions to complex problems. At its core, there is an algorithm for processing, data to extract useful information from, and hardware to actualize the physical actions. In this study, a probing station was studied in detail.

The integration of real-time camera images for guiding and correcting probing motions, alongside computational agents for gradient vector calculations are all possible thanks to algorithms, data, and hardware. By leveraging YOLO for object detection and employing regression techniques to address image distortions, one can advance the functionality of the probing station one step further.

The exploration of Piezo actuators highlights their potential for precise control and image processing for reducing the cost of actuators. Instead of using expensive and steady traditional actuators, cheaper and unsteady actuator motions might be corrected with image learning even in environmental variability and signal strength. The proposed solution of using image processing to account for these variations offers a practical approach to semi-automate probing tasks, balancing cost and performance.

Overall, the findings underscore the importance of high-quality data and robust algorithms in machine learning applications. While full automation may be out of reach, solving smaller problems along the way significantly boosts the effectiveness of the probing solution. The incremental advancements achieved through semi-automation can significantly improve operational runtime and reduce the burden on human operators.

In summary, the application of machine learning in probing stations exemplifies how data acquisition processing and machine learning can be integrated all together to solve a complex real-world problem. .

## References

- [1] MATLAB, *24.1.0.2628055 (R2024a) Update 4*. Natick, Massachusetts: The MathWorks Inc., 2024.
- [2] OpenCV, “Geometric transformations of images - opencv documentation,” 2024, accessed: 2024-09-13. [Online]. Available: [https://docs.opencv.org/4.x/d6e/tutorial\\_py\\_geometric\\_transformations.html](https://docs.opencv.org/4.x/d6e/tutorial_py_geometric_transformations.html)
- [3] K. Wood, “Robotics & ai,” <https://www.youtube.com/@kevinwoodrobotics/videos>, 2024.
- [4] J. Zhou, Y. Xu, L. Farbaniec, and A. Pellegrino, “Piezo-driven clamp release for synchronisation and timing of combined direct-shear stress waves,” *International Journal of Impact Engineering*, vol. 180, p. 104672, Jun. 2023.
- [5] e. a. Tommi S. Jaakkola, “Lecture 10: Recurrent neural networks1: Gating and lstm,” in *Machine Learning with Python-From Linear Models to Deep*

- Learning—MITx Course*, Cambridge MA, 2013, MIT OpenCourseWare. [Online]. Available: <https://learning.edx.org/course/course-v1:MITx+6.86x+3T2023/home>
- [6] E. Alpaydin, *Introduction to Machine Learning*, ser. Adaptive computation and machine learning. MIT Press, 2010. [Online]. Available: <https://books.google.co.kr/books?id=4j9GAQAAIAAJ>
  - [7] A. Rosebrock, *Deep Learning for Computer Vision with Python: Practitioner bundle*. Adrian Rosebrock, PyImageSearch.com, 2019, no. v. 2. [Online]. Available: <https://books.google.co.kr/books?id=oYljzgEACAAJ>
  - [8] ———, *Deep Learning for Computer Vision with Python: Starter Bundle*. PyImageSearch, 2017. [Online]. Available: <https://books.google.co.kr/books?id=9Ul-tgEACAAJ>
  - [9] Roboflow, “Printed circuit board dataset - roboflow universe,” 2024, accessed: 2024-09-13. [Online]. Available: <https://universe.roboflow.com/roboflow-100/printed-circuit-board/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true>
  - [10] OpenCV, “Non-local means denoising - opencv documentation,” 2024, accessed: 2024-09-13. [Online]. Available: [https://docs.opencv.org/3.4/d5/d69/tutorial\\_py\\_non\\_local\\_means.html](https://docs.opencv.org/3.4/d5/d69/tutorial_py_non_local_means.html)
  - [11] MathWorks, “Wavelet denoising - matlab & simulink documentation,” 2024, accessed: 2024-09-13. [Online]. Available: <https://www.mathworks.com/help/wavelet/ug/wavelet-denoising.html>

## 16 Miniproject Part-2 - Database Implementation

Implement and manage database designed in Task 4.2C

Outcome	Weight
Fundamental concepts of database	♦♦◊◊◊

It is hardly fundamental as it is more advanced.

Outcome	Weight
Relational Database Modelling	♦♦♦♦♦

Including several departments resulted in relationships interdependent.

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

Based on assignments, lectures, and textbook contents, I implemented a database for a pseudo-real case.

Outcome	Weight
Reflection	♦♦♦♦♦

I researched and reflected deeply on the dynamics of the team and imagined a few improvements. And made a few details as I implemented the language.

Outcome	Weight
Research and critical review	♦♦♦♦♦

I researched and reflected deeply on the dynamics of the team and imagined a few improvements. And made a few details as I implemented the language.

Date	Author	Comment
2024/09/12 14:28	Issac Han	Ready to Mark
2024/09/15 15:56	Aditya Tarigopula	Complete
2024/09/15 15:56	Aditya Tarigopula	Good Work on developing queries.

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Miniproject Part-2 - Database Implementation

---

*Submitted By:*

Issac HAN

s224509577

2024/09/12 14:28

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	◆◆◆◆
Relational Database Modelling	◆◆◆◆
Structured Query Language (SQL)	◆◆◆◆
Reflection	◆◆◆◆

Tried to reflect the need for real-life query from my working experiences.

September 12, 2024



Name: Isaac Han

CLASS: SIT772

## 1. Creating tables

```
1 ●  create database AssignmentWeek7;
2 ●  show databases;
3 ●  use AssignmentWeek7;
4
5 ●  ○  create table INVOICE (
6     INVOICE_ID int primary KEY,
7     PRODUCT_ID varchar(10) default("0000000000") references PRODUCT(PRODUCT_ID),
8     CUSTOMER_ID varchar(5) default("00000") references CUSTOMER(CUSTOMER_ID),
9     PRICE int,
10    INVOICE_DATE date
11 );
12
13 ●  ○  create table CUSTOMER (
14     CUSTOMER_ID varchar(5) default("00000") PRIMARY KEY,
15     AFFILIATION varchar(50),
16     CUSTOMER_LNAME varchar(30),
17     CUSTOMER_MNAME varchar(5),
18     CUSTOMER_FNAME varchar(30)
19 );
20
21 ●  ○  create table PRODUCT (
22     PRODUCT_ID char(7) PRIMARY KEY,
23     PRODUCT_STATUS char(3),
24     SOFTWARE_GITHUB_HASHFUN char(10) references SPECIFICATION(SOFTWARE_GITHUB_HASHFUN),
25     check (PRODUCT_STATUS in ("FIN", "CON", "DEV", "CAN"))
26 );
27
28 ●  ○  create table SPECIFICATION (
29     SPECIFICATION_ID char(5) PRIMARY KEY references BILLOMATERIAL(SPECIFICATION_ID),
30     SOFTWARE_GITHUB_HASHFUN char(10) references SOFTWARE(SOFTWARE_GITHUB_HASHFUN),
31     BOM_SPECIFICATION int references BILLOMATERIAL(BOM_SPECIFICATION)
32 );
33
34 ●  ○  create table BILLOMATERIAL (
35     SPECIFICATION_ID char(5) PRIMARY KEY,
36     COMPONENT_NAME varchar(30),
37     QUANTITY int default 0,
38     check (QUANTITY >= 0)
39 );
40
41 ●  ○  create table SOFTWARE (
42     SOFTWARE_GITHUB_HASHFUN varchar(10) PRIMARY KEY,
43     VERSION_NUMBER decimal(3, 2)
44 );
```

Tables_in_assignmentweek7
billomaterial
customer
invoice
product
software
specification

## 2. Inserting Rows into Tables

```

51    VALUES
52        ('C0001', 'Yale Univ.', 'Zu', 'A', 'John'),
53        ('C0002', 'MIT Univ.', 'Doe', 'B', 'Jane'),
54        ('C0003', 'CNU Univ.', 'U', 'C', 'N'),
55        ('C0004', 'POSCO', 'Dock', 'D', 'Builder'),
56        ('C0005', 'DEAKIN Univ.', 'Isaac', 'G', 'HAN');

57
58 • select * from CUSTOMER;
59
60 • INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_STATUS, SOFTWARE_GITHUB_HASHFUN)
61     VALUES
62        ('MPS-CHU', 'FIN', 'abcdefgij'),
63        ('MPS-CHH', 'CON', 'bcdefghijk'),
64        ('MPS-PTH', 'DEV', 'ccdefghijk'),

```

	CUSTOMER_ID	AFFILIATION	CUSTOMER_LNAME	CUSTOMER_MNAME	CUSTOMER_FNAME
▶	C0001	Yale Univ.	Zu	A	John
	C0002	MIT Univ.	Doe	B	Jane
	C0003	CNU Univ.	U	C	N
	C0004	POSCO	Dock	D	Builder
*	C0005	DEAKIN Univ.	Isaac	G	HAN
*	NULL	NULL	NULL	NULL	NULL

a)

```

58 • INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_STATUS, SOFTWARE_GITHUB_HASHFUN)
59     VALUES
60        ('MPS-CHU', 'FIN', 'abcdefgij'),
61        ('MPS-CHH', 'CON', 'bcdefghijk'),
62        ('MPS-PTH', 'DEV', 'ccdefghijk'),
63        ('MPS-PHL', 'CAN', 'defghijkln'),
64        ('MPS-PLN', 'FIN', 'efghijklnm');
65
66 • select* from PRODUCT;
67
68

```

	PRODUCT_ID	PRODUCT_STATUS	SOFTWARE_GITHUB_HASHFUN
▶	MPS-CHH	CON	bcdefghijk
	MPS-CHU	FIN	abcdefgij
	MPS-PHL	CAN	defghijkln
	MPS-PLN	FIN	efghijklnm
	MPS-PTH	DEV	ccdefghijk
*	NULL	NULL	NULL

b)

```

71 • INSERT INTO INVOICE (INVOICE_ID, PRODUCT_ID, CUSTOMER_ID, PRICE, INVOICE_DATE)
72     VALUES
73         (1, 'MPS-CHU', 'C0001', 75000, '2024-01-01'),
74         (2, 'MPS-CHH', 'C0002', 60000, '2024-01-02'),
75         (3, 'MPS-PTH', 'C0003', 65000, '2024-01-03'),
76         (4, 'MPS-PHL', 'C0004', 70000, '2024-01-04'),
77         (5, 'MPS-PLN', 'C0005', 80000, '2024-01-05');
78
79 • select* from INVOICE;
80
81

```

Result Grid					
	INVOICE_ID	PRODUCT_ID	CUSTOMER_ID	PRICE	INVOICE_DATE
▶	1	MPS-CHU	C0001	75000	2024-01-01
	2	MPS-CHH	C0002	60000	2024-01-02
	3	MPS-PTH	C0003	65000	2024-01-03
	4	MPS-PHL	C0004	70000	2024-01-04
	5	MPS-PLN	C0005	80000	2024-01-05
*	NUL	NUL	NUL	NUL	NUL

c)

```

79
80 • INSERT INTO SPECIFICATION (SPECIFICATION_ID, SOFTWARE_GITHUB_HASHFUN)
81     VALUES
82     ('S001', 'abcdefg hij'),
83     ('S002', 'bcdefghijk'),
84     ('S003', 'ccdefghijk'),
85     ('S004', 'defghijkln'),
86     ('S005', 'efghijklnm');
87
88 • select* from SPECIFICATION;
89

```

Result Grid		
	SPECIFICATION_ID	SOFTWARE_GITHUB_HASHFUN
▶	S001	abcdefg hij
	S002	bcdefghijk
	S003	ccdefghijk
	S004	defghijkln
	S005	efghijklnm
*	NUL	NUL

d)

e)

```

101 • INSERT INTO SOFTWARE (SOFTWARE_GITHUB_HASHFUN, VERSION_NUMBER)
102     VALUES
103         ('abcdefghij', 2.00),
104         ('bcdefghijk', 2.01),
105         ('ccdefghijk', 3.00),|
106         ('defghijkln', 3.01),
107         ('efghijklnm', 3.02);
108
109 • select* from SOFTWARE;
110

```

	SOFTWARE_GITHUB_HASHFUN	VERSION_NUMBER
▶	abcdefghij	2.00
	bcdefghijk	2.01
	ccdefghijk	3.00
	defghijkln	3.01
	efghijklnm	3.02
*	NULL	NULL

f)

```

90 • INSERT INTO BILLOFMATERIAL (SPECIFICATION_ID, COMPONENT_NAME, QUANTITY)
91     VALUES
92         ('S001', 'XRD_DORM', 3),
93         ('S002', 'PALTIER', 7),
94         ('S003', 'CERAMIC', 20),
95         ('S004', 'PIEZO', 15),
96         ('S005', 'probe_tip', 12);
97
98 • select* from BILLOFMATERIAL;
99
100 • INSERT INTO SOFTWARE (SOFTWARE_GITHUB_HASHFUN, VERSION_NUMBER)
101     VALUES
102         ('ccdefghijk', 3.00)

```

	SPECIFICATION_ID	COMPONENT_NAME	QUANTITY
▶	S001	XRD_DORM	3
	S002	PALTIER	7
	S003	CERAMIC	20
	S004	PIEZO	15
	S005	probe_tip	12
*	NULL	NULL	NULL

g)

3. Inserting a new column discount into invoice

```
110 • alter table INVOICE  
111     add column DISCOUNT decimal(2, 2) default 0.00;  
112  
113 • select* from INVOICE;
```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	INVOICE_ID	PRODUCT_ID	CUSTOMER_ID	PRICE	INVOICE_DATE	DISCOUNT
▶	1	MPS-CHU	C0001	75000	2024-01-01	0.00
	2	MPS-CHH	C0002	60000	2024-01-02	0.00
	3	MPS-PTH	C0003	65000	2024-01-03	0.00
	4	MPS-PHL	C0004	70000	2024-01-04	0.00
*	5	MPS-PLN	C0005	80000	2024-01-05	0.00
*	NULL	NULL	NULL	NULL	NULL	NULL

4. Change value based on query

```
115 • SET SQL_SAFE_UPDATES = 0;  
116 • update INVOICE  
117     set DISCOUNT = 0.15  
118     where CUSTOMER_ID in (  
119         select CUSTOMER_ID  
120         from CUSTOMER  
121         WHERE AFFILIATION LIKE "%Univ.%"  
122     );  
123  
124 • select* from INVOICE;
```

```
125 • DESC CUSTOMER;  
126 • SET SQL_SAFE_UPDATES = 1;
```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	INVOICE_ID	PRODUCT_ID	CUSTOMER_ID	PRICE	INVOICE_DATE	DISCOUNT
▶	1	MPS-CHU	C0001	75000	2024-01-01	0.15
	2	MPS-CHH	C0002	60000	2024-01-02	0.15
	3	MPS-PTH	C0003	65000	2024-01-03	0.15
	4	MPS-PHL	C0004	70000	2024-01-04	0.00
*	5	MPS-PLN	C0005	80000	2024-01-05	0.15
*	NULL	NULL	NULL	NULL	NULL	NULL

5. .

- a. Write an SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY.

-I want to find VIP by finding who spent the most money on the product.

```
128 •   SELECT i.INVOICE_ID, c.CUSTOMER_FNAME, c.CUSTOMER_LNAME, i.PRICE, i.INVOICE_DATE  
129     FROM INVOICE i  
130    INNER JOIN CUSTOMER c ON i.CUSTOMER_ID = c.CUSTOMER_ID  
131    ORDER BY i.PRICE DESC;  
132  
133
```

	INVOICE_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	PRICE	INVOICE_DATE
▶	5	HAN	Isaac	80000	2024-01-05
1	John	Zu		75000	2024-01-01
4	Builder	Dock		70000	2024-01-04
3	N	U		65000	2024-01-03
2	Jane	Doe		60000	2024-01-02

- b. Write an SQL query to demonstrate the use of SELECT with WHERE and IN

-Fetching software version from github (github hash fun) whose software version is 2.

```
133 •   select SOFTWARE_GITHUB_HASHFUN, VERSION_NUMBER  
134     from SOFTWARE  
135    where VERSION_NUMBER like ("2.%");  
136  
137 •   desc SOFTWARE;  
138 •   select* from SOFTWARE;  
139  
140
```

	SOFTWARE_GITHUB_HASHFUN	VERSION_NUMBER
▶	abcdefghijkl	2.00
	bcdefghijk	2.01
*	NULL	NULL

- c. Write an SQL query to demonstrate the use of at least one DATE function.

```
140 •  SELECT INVOICE_ID, PRICE, INVOICE_DATE  
141      FROM INVOICE  
142      WHERE DAY(INVOICE_DATE) in (1, 2) AND YEAR(INVOICE_DATE) = 2024;
```

	INVOICE_ID	PRICE	INVOICE_DATE
▶	1	75000	2024-01-01
	2	60000	2024-01-02
*	NULL	NULL	NULL

- d. Write an SQL statement to create a VIEW using a SELECT statement with a JOIN  
Created virtual view where we need to send the product as they are not sent to customers yet.

```
144 •  CREATE VIEW PRODUCT_TO_SEND AS  
145      SELECT i.INVOICE_ID, i.PRICE, i.INVOICE_DATE, p.PRODUCT_ID, p.PRODUCT_STATUS  
146      FROM INVOICE i  
147      JOIN PRODUCT p ON i.PRODUCT_ID = p.PRODUCT_ID  
148      Where p.PRODUCT_STATUS in ("CON", "CAN");  
149  
150 •  select* from PRODUCT_TO_SEND;  
151
```

	INVOICE_ID	PRICE	INVOICE_DATE	PRODUCT_ID	PRODUCT_STATUS
▶	2	60000	2024-01-02	MPS-CHH	CON
	4	70000	2024-01-04	MPS-PHL	CAN

## 17 Online Quiz 2

Complete Online Quiz 2 available on the unit site

Outcome	Weight
Fundamental concepts of database	◆◆◆◆◆

I felt the Quiz was rather basic compared to the contents of the textbook I read by far.

Date	Author	Comment
2024/09/13 23:11	Issac Han	Ready to Mark
2024/09/15 15:34	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Online Quiz 2

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/13 23:11

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦

Quiz testing basics.

September 13, 2024



SIT103\_SIT772 - Data and Information Management     ISSAC HAN

Home Content Discussions Assessment ▾ Tools ▾ 2024 T2

Quiz List > Submissions

## Quiz Submissions - Task 8.1C - Online Quiz 2 ▾

Add to ePortfolio

ISSAC HAN (username: s224509577)

Individual Attempts	Grade
Attempt 1	18 / 20 - 90 %
Overall Grade (highest attempt):	18 / 20 - 90 %

Add to ePortfolio - Google Chrome  
https://d2l.deakin.edu.au/d2l/common/popup/popup.d2l?ou=1521432&queryString=

Add to ePortfolio

Artifact created successfully

Site  
SIT103\_SIT772 - Data and Information Management

Tool  
Quiz

Item  
Task 8.1C - Online Quiz 2

## 18 PL/PSQL - Trigger, Procedure and Function

Write procedure and Cursors to manipulate SQL results

Outcome	Weight
Research and critical review	♦♦♦♦♦

I learned Oracle Syntax for the first time and it was also the time that I spent many nights studying the basics due to a software internship.

Outcome	Weight
Fundamental concepts of database	♦♦♦♦◊

It extends the functionality of database management language into another step; it combines procedural languages with data. I felt the functionality is somewhat similar to R but the contents and syntax are quite different.

Date	Author	Comment
2024/09/20 23:16	Issac Han	Ready to Mark
2024/09/23 06:29	Aditya Tarigopula	good work
2024/09/23 06:29	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## PL/PSQL - Trigger, Procedure and Function

---

*Submitted By:*

Issac HAN

s224509577

2024/09/20 23:16

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Research and critical review	◆◆◆◆

Learned ORACLE syntax the first time.

September 20, 2024



Name: Isaac Han

Unit: SIT 772

Assignment: 8.2D

Beginning of the table:

```
interactive-access-2019082100-prod.aa-global.deakin.edu.au - PuTTY
select* from INVOICE:
          *
ERROR at line 1:
ORA-00933: SQL command not properly ended

SQL> select* from CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
-----  -----
    1000  Smith           Jeanne          1050.11
    1001  Ortega          Juan            840.92

SQL> select* from INVOICE;

  INV_NUM   CUST_NUM INV_DATE     INV_AMOUNT
-----  -----
    8000       1000 23-MAR-16    235.89
    8001       1001 23-MAR-16    312.82
    8002       1001 30-MAR-16    528.1
    8003       1000 12-APR-16    194.78
    8004       1000 23-APR-16    619.44

SQL>
```

1. Task1

```
Procedure created.

SQL> select * from CUSTOMER;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
-----  -----
  1000 Smith          Jeanne        1050.11
  1001 Ortega         Juan         840.92

SQL> EXEC PRC_ADD_CUSTOMER(1002, 'Rauthor', 'Peter', .8);

PL/SQL procedure successfully completed.

SQL> select * from customer;

  CUST_NUM CUST_LNAME      CUST_FNAME      CUST_BALANCE
-----  -----
  1000 Smith          Jeanne        1050.11
  1001 Ortega         Juan         840.92
  1002 Rauthor        Peter          .8

SQL>
```

2

```
SQL> createt or replace function get_invoice_count(
SP2-0734: unknown command beginning "createt or..." - rest of line ignored.
SQL> create or replace FUNCTION get_invoice_count (
 2  p_cust_num IN INVOICE.CUST_NUM%TYPE
 3  ) RETURN NUMBER AS
 4  v_count NUMBER;
 5  BEGIN
 6    SELECT COUNT(*)
 7    INTO v_count
 8    FROM INVOICE
 9    WHERE CUST_NUM = p_cust_num;
10   RETURN v_count;
11 END;
12 /
```

Function created.

```
GET_INVOICE_COUNT(1000)
-----
3

SQL> SELECT get_invoice_count(0) FROM DUAL;

GET_INVOICE_COUNT(0)
-----
0

SQL> SELECT get_invoice_count(1000) FROM DUAL;

GET_INVOICE_COUNT(1000)
-----
3

SQL>
      3
SQL> SP2-0226: Invalid line number
SQL>
SQL> SELECT get_invoice_count(1005) FROM DUAL;

GET_INVOICE_COUNT(1005)
-----
0

SQL> select * from INVOICE;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-16	235.89
8001	1001	23-MAR-16	312.82
8002	1001	30-MAR-16	528.1
8003	1000	12-APR-16	194.78
8004	1000	23-APR-16	619.44

```
SQL> █
```

3.

```
SQL> create or replace trigger TRG_UPDATE_CUST_BALANCE
  2  after INSERT on INVOICE
  3  for each ROW
  4  BEGIN
  5    UPDATE CUSTOMER
  6    SET CUST_BALANCE = CUST_BALANCE + :NEW.INV_AMOUNT
  7    WHERE CUST_NUM = :NEW.CUST_NUM;
  8  END;
  9 /
```

Trigger created.

```
SQL> select * from CUSTOMER;
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	840.92
1002	Rauthor	Peter	.8

```
SQL> INSERT INTO INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  2  VALUES(8005, 1001, TO_DATE('27-Apr-2016', 'DD-Mon-YYYY'), 225.40)
  3 ;
```

ERROR:

ORA-01756: quoted string not properly terminated

```
SQL> INSERT into INVOICE (INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  2  VALUES (8005, 1001, TO_DATE('27-Apr-2016', 'DD-Mon-YYYY'), 225.40);
```

1 row created.

```
SQL> select * from CUSTOMER;
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	.8

```
SQL> █
```

4.

```
SQL> create or replace procedure PRC_ADD_INVOICE(
  2  p_inv_num in number,
  3  p_cust_num in number,
  4  p_inv_date in DATE,
  5  p_inv_amount in number
  6  ) AS
  7  BEGIN
  8      INSERT INTO INVOICE(INV_NUM, CUST_NUM, INV_DATE, INV_AMOUNT)
  9      VALUES(p_inv_num, p_cust_num, p_inv_date, p_inv_amount);
10  END;
11 /
```

Procedure created.

```
SQL> select* from customer;
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	.8

```
SQL> select* from Invoice;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-16	235.89
8001	1001	23-MAR-16	312.82
8002	1001	30-MAR-16	528.1
8003	1000	12-APR-16	194.78
8004	1000	23-APR-16	619.44
8005	1001	27-APR-16	225.4

6 rows selected.

```
SQL> EXEC PRC_ADD_INVOICE(8006, 1002, TO_DATE('29-Apr-2016', 'DD-Mon-YYYY'), 175.85);
```

PL/SQL procedure successfully completed.

```
SQL> select* from INVOICE;
```

INV_NUM	CUST_NUM	INV_DATE	INV_AMOUNT
8000	1000	23-MAR-16	235.89
8001	1001	23-MAR-16	312.82
8002	1001	30-MAR-16	528.1
8003	1000	12-APR-16	194.78
8004	1000	23-APR-16	619.44
8005	1001	27-APR-16	225.4
8006	1002	29-APR-16	175.85

7 rows selected.

```
SQL> select* from CUSTOMER;
```

CUST_NUM	CUST_LNAME	CUST_FNAME	CUST_BALANCE
1000	Smith	Jeanne	1050.11
1001	Ortega	Juan	1066.32
1002	Rauthor	Peter	176.65

```
SQL> 
```

## 19 Data Analysis and Visualization using Excel

Use Excel PIVOT TABLE/CHART feature to do some basic Business Intelligence

Outcome	Weight
Fundamental concepts of database	♦♦◊◊◊

I reflected on how data can shape the business and applied the principles to the organization for which I have worked for about 2 months.

Outcome	Weight
Reflection	♦♦♦♦♦

I reflected on how data can shape the business and applied the principles to the organization for which I have worked for about 2 months.

Outcome	Weight
Research and critical review	♦♦♦♦♦

I reflected on how data can shape the business and applied the principles to the organization for which I have worked for about 2 months.

Date	Author	Comment
2024/09/20 17:01	Issac Han	Ready to Mark
2024/09/22 16:07	Aditya Tarigopula	Good Work on recommendations.
2024/09/22 16:07	Aditya Tarigopula	Complete

DEAKIN UNIVERSITY

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Data Analysis and Visualization using Excel

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/20 17:01

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦◊◊◊
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

I reflected on how data can shape the business and applied the principles to the organization for which I have worked for about 2 months.

September 20, 2024

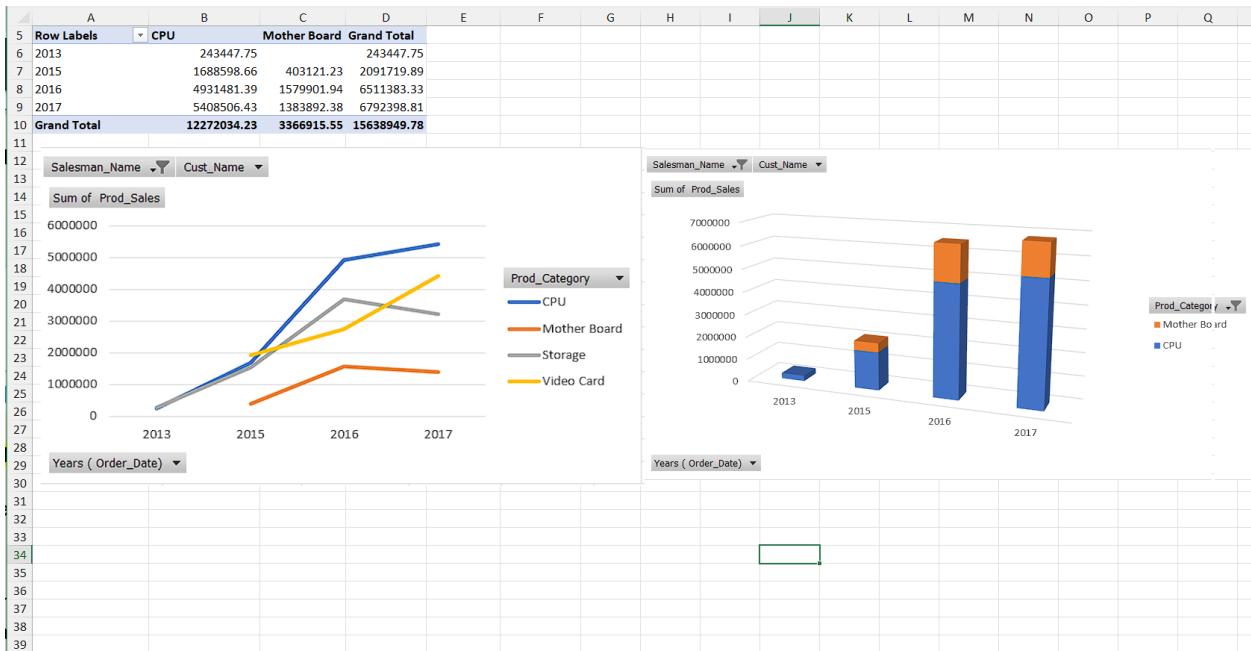


Name: Isaac Han

Class: SIT 772

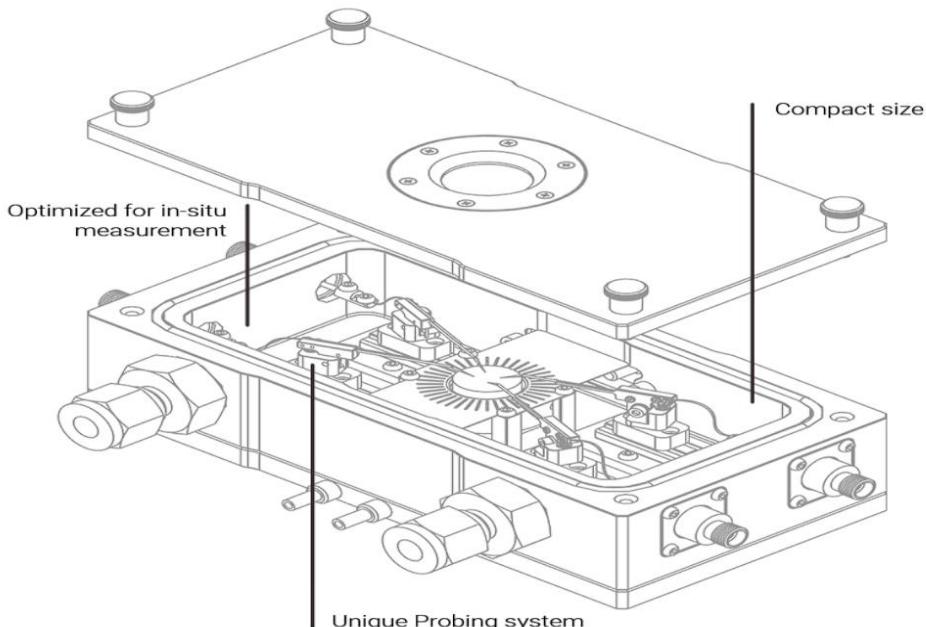
Project: 9.1P Data Analysis and Visualization Using Excel

Task1 and Task2:



### Task3:

For 2 months, I have been doing an internship at Nextron, which is a hardware company selling miniatured versions of probing stations for budget friendly and easier usage for small projects <https://www.microprobebsystem.com/index.html>. The role of probing stations is to test, to analyze failures, and to meet the quality control requirements of electrical circuits or dielectric materials. Due to its verry small size, the station is primarily used for testing new materials developed from Ph.D. students all around the world, rather than in an industrial setting.

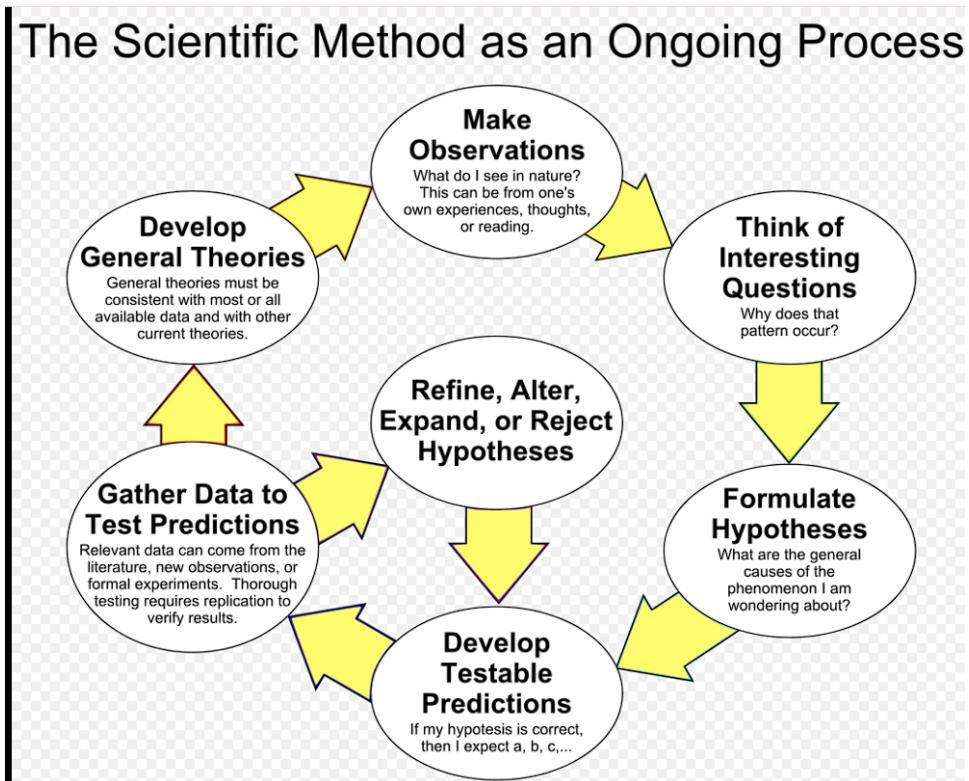


The company is a small and a startup company; even though I sense there can be great data pipeline optimizations, most of them do not actively collect data. Data is primarily collected by hardware testers for the durability tests and marketing agents who primarily deal with the Google ads. What I heard is that they are interested in finding the right keywords that describe the products for the Google ads. Ads are competitive, so allocating the right amount of money for the right keywords is important, rather than allocating a thin amount of money across the keywords.

Rather than how they use the data, I see several ways that data pipelining and visualization can give the competitive advantage to the revenue and productivity. The first one is to manage data using database software with data integrity rather than with Microsoft excel. I saw data integrity problem several times. Another is to implement database models for productivity improvement. In the most basic sense, many workers are looking for some information with an incomplete key and file directories; one example is to find the software source code using the version number to A/S, where the version number is a hand-written number that multiple source codes correspond to. I find the Github's hash code to be a collision safe primary key easy to check out the source code handed to the customer.

Additionally, product tests, quality, and functionality are dependent on the developer as there are no testers; developers test their own creations. This situation creates customer level product feedback where a large portion of the work is allocated to fixing the customer's complaints, one of the solutions is of course separating the workflow, but another reasonable product feedback process is to use the Toyota Way. Even though the Toyota Way is a high-level principle, a portion of continuous development is with statistical analysis of testing the current development with data.

From Wikipedia:



At least in software, there is a module called logging, which can summarize what has happened by far. It is often the case that customers report problems with the product, and the company can only reply with generic solutions from experience. Even though the generic speculations and advice from physics Ph.D.'s are professional, they do not directly correspond to the specific problems that buyers may have. On the other hand, logger object can provide necessary information to specify what has gone wrong. By writing down faulty products and normal products, one can obtain a database. By data exploration with BI, it would be possible to explore data, hypothesize product defects and verify the predictions with data. Potentially, changing a production line with another one to improve the product quality.

## The Netflix Recommendation System (Singular Value Decomposition of Matrix)

The products that company sells are of small quantities and are very expensive compared to regular electronic devices like computers and air conditioners. It is a research device that targets very specific individuals all around the world. Since the sales data is also small and very defined, one particular interest may be how to target additional individuals. In terms of machine learning language, it is desired to find a cluster of individuals who are interested in products, from a lack of data. This technique is called SVD decomposition or the Netflix Recommendation System. This is a particularly effective marketing as the marketing cost is small because of targeting small amounts of people and the profit of successful marketing is large due to a large margin associated with each product's sales.

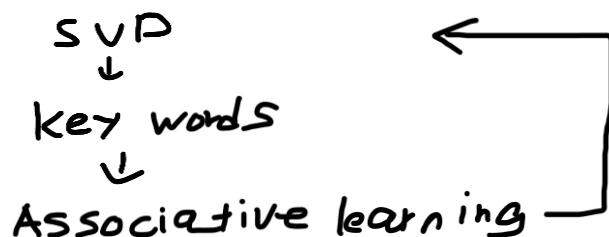
Just like movies, everyone evaluates very small amounts of movies out of all possible movies. Obviously, Netflix wanted to sell movies to individuals who didn't watch the movie and evaluated them. One of the most successful recommendation algorithms they developed was the Netflix Recommendation System, which has to do with decomposing matrix. Below is the summary diagram of what I have thought about a marketing strategy with data instead of manually parameter tuning with the google ads.

$$Utility = \sum (N_{\text{effective\_exposure}}) * \mu_{\text{profit}} - \mu_{\text{cost}} * N_{\text{Exposure}}$$

$$N_{\text{effectiv\_exposure}} = \sum (G_i * \mu_i)$$

$$\text{Cost} = \sum C(i, g_i)$$

We want to maximize Utility, which is a profit from the marketing. Effective exposure determines the portion of people who are exposed to the ads and purchase the products, whereas the exposure determines the number of people who are exposed to the ads. Effective exposure is determined by how many portions of people we want to expose the ads to and the mean positive response from it. Finally, the cost is the sum of money needed to expose the ads to  $g_i$  percents of people in each group.



The role of BI is to aggressively model the general functions that I have used. From the continuous development diagram posted, to develop a general theory (i.e.  $\mu_i = 0.3$  for example), it requires developing testable predictions, gather data to test predictions, and refine, alter, expand, or reject hypothesis through data. To make predictions, it is often helpful to rely on data visualization. Data visualization can also make it simple to filter out non-senses from testing hypotheses and make more refined predictions if combined with data processing techniques like PCA. Just like a stock market, the general theories are often time dependent. It means that developing the general theory periodically is often necessary.

In summary, storing data with modern databases offers a greater benefit than managing with Microsoft excel. The BI and data visualization can not only be used in business decisions and marketing, the BI is also helpful in developing the general theories surrounding the business, enabling more advanced modeling and production.

## 20 Interaction with a database via an User Interface

Desgin an user interface form to populate and display required results from a database and prepare a video demonstration

Outcome	Weight
Fundamental concepts of database	♦♦♦◊◊

It was an assignment requiring GUI and server interaction through Php. There was a configuration error where directly calling the GUI caused PHP not to load; it was solved by calling the GUI through localhost.

Outcome	Weight
Structured Query Language (SQL)	♦♦◊◊◊

I mostly used SQL languages I created in mini-projects; hardly new SQL commands were made besides a few modifications.

Outcome	Weight
Reflection	♦♦♦♦♦

It was an assignment requiring GUI and server interaction through Php. There was a configuration error where directly calling the GUI caused PHP not to load; it was solved by calling the GUI through localhost.

Outcome	Weight
Research and critical review	♦♦♦♦♦

I further developed the idea by proposing an automatic trigger that sends an email to an employee the moment a row is inserted into the inquiry table. I felt this idea is quite powerful automatizing sales and timely response to the customer with products. Additionally, even call centers might be able to apply such techniques.

Date	Author	Comment
2024/09/24 22:14	Issac Han	Ready to Mark
2024/09/24 22:14	Issac Han	<a href="https://youtu.be/Qbqm__-xbcH8">https://youtu.be/Qbqm__-xbcH8</a>
2024/09/24 22:21	Issac Han	This is the final assignment, and it was a pleasure to have you! Thanks.
2024/09/29 18:50	Aditya Tarigopula	good work
2024/09/29 18:50	Aditya Tarigopula	Complete

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Interaction with a database via an User Interface

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/24 22:14

*Tutor:*  
Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Structured Query Language (SQL)	♦♦◊◊◊
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

It was an assignment requiring GUI and server interaction through PHP. There was a configuration error where directly calling the GUI caused PHP not to load; it was solved by calling the GUI through localhost.

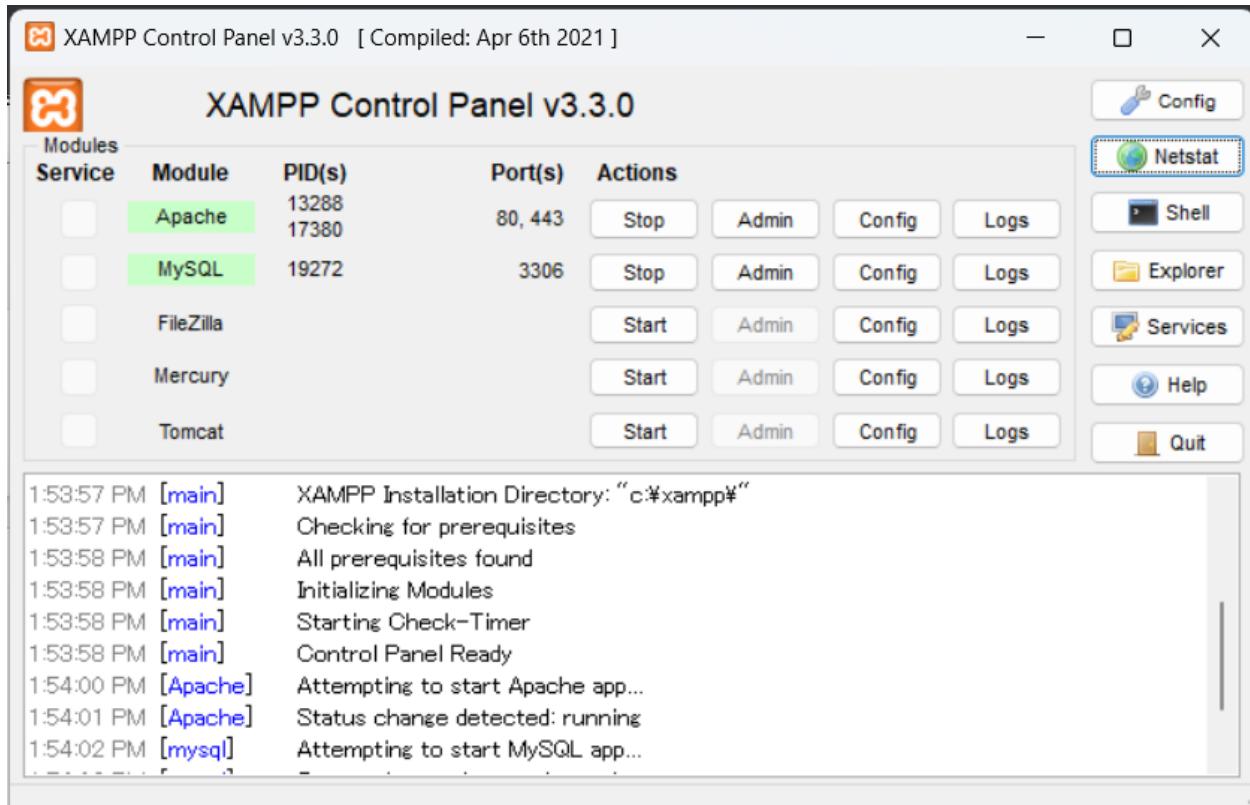
September 24, 2024



Name: Isaac Han

Unit: SIT 772

1. Downloaded and launched Apache and MySQL. Using resource manager, disconnected previous MySQL server.



2. run <http://localhost/phpmyadmin/index.php?route=/server/sql>

Went to SQL and ran the commands from 7.2 to create table.

The screenshot shows the phpMyAdmin interface for a database named 'finalassignment'. The current table is 'billofmaterial'. The 'Structure' tab is selected, displaying the following table definition:

```

CREATE TABLE `billofmaterial` (
  `SPECIFICATION_ID` char(5) NOT NULL,
  `COMPONENT_NAME` varchar(30) DEFAULT NULL,
  `QUANTITY` int(11) DEFAULT 0
)

```

**Table structure**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	SPECIFICATION_ID	char(5)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	COMPONENT_NAME	varchar(30)	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	QUANTITY	int(11)			Yes	0			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

**Indexes**

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a> <a href="#">Rename</a> <a href="#">Drop</a>	PRIMARY	BTREE	Yes	No	SPECIFICATION_ID	0	A	No	

**Add** 1 column(s) after QUANTITY [Go](#)

**Create an index on** 1 columns [Go](#)

3. (to XAMPP root folder) Create the form interface; HTML script given by the ChatGPT

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Product</title>
</head>
<body>
    <h2>Product Entry Form</h2>
    <form action="insert_product.php" method="POST">
        <label for="product_id">Product ID:</label>
        <input type="text" id="product_id" name="product_id" required><br><br>

        <label for="product_status">Product Status:</label>
        <select id="product_status" name="product_status" required>
            <option value="FIN">Finished</option>
            <option value="CON">Construction</option>
            <option value="DEV">Development</option>
            <option value="CAN">Cancelled</option>
        </select><br><br>

        <label for="software_hash">Software GitHub Hash:</label>
        <input type="text" id="software_hash" name="software_hash" required><br><br>

        <button type="submit">Add Product</button>
    </form>
</body>
</html>

```

4. (to XAMPP root file) put PHP file. Created by ChatGPT and modified errors.

The screenshot shows a code editor window with the title "Explorer (Ctrl+Shift+E)". The file path is "C: > xampp > htdocs > insert\_product.php". The code itself is a PHP script for inserting data into a MySQL database. It includes connection setup, form data validation, and an SQL query to insert the data into the "PRODUCT" table. The code is numbered from 1 to 45.

```
C: > xampp > htdocs > insert_product.php
1  <?php
2  // Database connection details
3  $servername = "localhost";
4  $username = "root"; // Default XAMPP username
5  $password = ""; // Default XAMPP password (leave blank if default)
6  $dbname = "finalassignment"; // Replace with your database name
7
8  // Create connection
9  $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // Check connection
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 // Get the form data
17 $product_id = $_POST["product_id"];
18 $product_status = $_POST["product_status"];
19 $software_hash = $_POST["software_hash"];
20
21 // Validate the inputs
22 if (empty($product_id) || empty($product_status) || empty($software_hash)) {
23     echo "All fields are required.";
24 } elseif (!preg_match("/^A-Za-z0-9-[7]/", $product_id)) {
25     echo "Product ID must be 7 alphanumeric characters.";
26 } elseif (!in_array($product_status, ["FIN", "CON", "DEV", "CAN"])) {
27     echo "Invalid product status.";
28 } elseif (!preg_match("/^A-Za-z0-9-[10]/", $software_hash)) {
29     echo "GitHub hash must be 10 alphanumeric characters.";
30 } else {
31     // SQL query to insert data into the PRODUCT table
32     $sql = "INSERT INTO PRODUCT (PRODUCT_ID, PRODUCT_STATUS, SOFTWARE_GITHUB_HASHFUN)
33             VALUES ('$product_id', '$product_status', '$software_hash')";
34
35     if ($conn->query($sql) === TRUE) {
36         echo "New product added successfully.";
37     } else {
38         echo "Error: " . $sql . "<br>" . $conn->error;
39     }
40 }
41
42 // Close the connection
43 $conn->close();
44 ?>
45
```

## 5. Checking

### a. Product Entry Success

The screenshot shows the phpMyAdmin interface for a database named 'finalassignment'. The left sidebar shows the database structure with tables like 'finalassignment', 'customer', 'invoice', and 'product'. The 'product' table is selected, and its structure is displayed on the right. A new row has been added to the 'product' table, containing the following data:

PRODUCT_ID	PRODUCT_STATUS	SOFTWARE_GITHUB_HASHFUN
MPS-CHU	FIN	a123456789

Below the table, there are various operations like Print, Copy to clipboard, Export, Display chart, and Create view. A message at the top indicates that 1 row was affected by the last query.

New product added successfully.

### b. Failed to add a new product:

## Product Entry Form

Product ID:

Product Status:

Software GitHub Hash:

The screenshot shows a web browser window with the URL `localhost/insert_product.php`. The page displays an error message: "Product ID must be 7 alphanumeric characters." Below the message is a table with a single row of data:

	PRODUCT_ID	PRODUCT_STATUS	SOFTWARE_GITHUB_HASH
<input type="checkbox"/>	MPS-CHU	FIN	a123456789

Below the table, there is a message: "Product did not change its entry".

c.

## 21 Learning Summary Report

Prepare a learning summary report reflect on your learning in the unit against the ULOs

Outcome	Weight
Fundamental concepts of database	◆◆◆◆◆

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

Outcome	Weight
Relational Database Modelling	◆◆◆◆◆

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

Outcome	Weight
Structured Query Language (SQL)	◆◆◆◆◆

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

Outcome	Weight
Reflection	◆◆◆◆◆

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

Outcome	Weight
Research and critical review	◆◆◆◆◆

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

Date	Author	Comment
2024/09/28 00:15	Issac Han	Ready to Mark
2024/09/29 18:30	Aditya Tarigopula	good work
2024/09/29 18:30	Aditya Tarigopula	Complete

DATABASE FUNDAMENTALS

ONTRACK SUBMISSION

---

## Learning Summary Report

---

*Submitted By:*

Issac HAN  
s224509577  
2024/09/28 00:15

*Tutor:*

Aditya TARIGOPULA

Outcome	Weight
Fundamental concepts of database	♦♦♦♦♦
Relational Database Modelling	♦♦♦♦♦
Structured Query Language (SQL)	♦♦♦♦♦
Reflection	♦♦♦♦♦
Research and critical review	♦♦♦♦♦

This is the summary of all the assignment that I have done, and I believe I have achieved all the learning outcomes of the class.

September 28, 2024





# SIT772

## Database Fundamentals

Learning Summary Report

Isaac Han  
STUDENT ID: s224509577

**Self-Assessment Details**

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment				✓

[In the actual final report, my works will be attached here, right?]

Checklist	Included
Learning Summary Report	
All tasks required for the target grade completed	✓
Evidence of any additional task(s) or activities completed	✓

**Declaration**

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: Isaac Han

## Portfolio Overview

This portfolio includes work that demonstrates that I have achieved all unit learning outcomes for SIT772 Database Fundamentals to a High Distinction level.

The only data language I knew was Pandas and R; without knowing much of structured query language, I dived into database languages. I have known the importance of data as basis for statistics but did not know there is a simple way of representing data relations with an entity relationship diagram, enabling intuitive and higher level of understanding of data. In conjunction to the unit learning outcome of interpreting and explaining concepts of data, information, and knowledge, the task Reflection on Data-driven systems and entity relationship diagram explained high-level reasoning can transform raw data into information or even into knowledge without making calculations, like covariance matrices. The task data analysis and visualization also reinforced this idea that knowledge from data and business intelligence can be initiated from simple visualization without statistics. I also noted that PK, FK, normalization of table, and database management can have data integrity, database rule reinforcement, automatic trigger, and handling large data faster unlike traditional file system and excel may not be capable of.

Regarding designing, implementing, evaluating, and maintaining databases through SQL, I had a 2-month internship with a small company as a software engineer, and I was able to design and apply a sample database. In the mini projects, I connected how customer inquiry can affect software, hardware, and manufacturing team. Especially with the experience that the company had hard time finding the correct software version, I put collision-free GitHub hash function as the primary key of software table. With Apache, HTML, PHP, and SQL server I was able to imitate a customer enquiry in website (HTML) automatically triggering database insertion and sending email to an employee through tasks 4.2, 7.2, 8.2, and 10.1. I also applied SQL statements in the database to make analysis; automatic price reduction if name contains “Univ,” and making a virtual view table if the product was not sent and finished.

Lastly, I have researched a potential machine learning algorithm in a probing company and applied how learning models require data. Especially, large volumes of data might not equate to a better learning algorithm. In the process of active learning and probably approximately correct learning with a box model, there are cases where additional data do not contribute to better accuracy. Multiple image filtering and enhancement techniques can process low quality data into higher quality. In the case of a special linear support vector machine algorithm called Haar Cascade for facial detection, a large volume of data is fed into the learning algorithm with labels. With external device feeding buffered streamed frames, a correct data pipelining is essential for object detection. Techniques like storing the last detected cached frame can help overcome non-continuous detection and flickering.

## Reflections

The most important thing I learned was that simple relationships may be more effective than statistical calculations. I studied statistics for machine learning and mostly focused data relations with statistics. However, with designing databases, making the entity diagram, and making BI, a more intuitive approach may be more effective. For this reason, designing a table with well-planned relationships can save a great deal of time.

### I feel I learnt these topics, concepts, and/or tools really well:

When I first learned the basics of SQL, most of them are like the table structure of R and Pandas, which made me feel comfortable with dealing with the basics of SQL.

### I found the following topics particularly challenging:

I found PSL and PL to be a bit challenging because it was the time where I learned something new and did not read the book due to the internship; I lacked time to reinforce learning with the textbook and felt those new syntax to be challenging.

### I found the following topics particularly interesting:

I found the trigger to be extremely interesting. In fact, this is one of the main topics of assignment of interaction with a database through GUI. I did not implement the trigger function, but the baseline is that when an inquiry is made through HTML, the row would be inserted through PHP and this would trigger sending emails to workers.

### I still need to work on the following areas:

From the course, the procedural query language is something that I am not comfortable using due to frequent mistakes. I feel integrating databases to procedural language like Python would substitute the benefits of PSL, at least in the range of manageable size of data. But I still need to work on it to use in the future.

### The things that helped me most were:

I found the textbook was extremely helpful when I got started with database learning. Its case studies were frequently used to answer some of the tasks questions and I was able to come up with more refined approaches.

### My progress in this unit was ...:



I tried to keep up getting the High Distinction and had a hard time getting it done due to an internship I started. I frequently spent time doing assignments after getting back from work, and I am glad that I mostly finished the work. One regret is that I dropped reading the textbook once I started work; I planned to finish reading the textbook but couldn't.

### If I did this unit again, I would do the following things differently:

I would spend more time on reading textbooks.

### Other...:

By studying this unit and working in a company at the same time, I was able to imagine several database techniques that the company might be lacking. This helped me developing the concepts of the unit further.