RESEARCH REPORT

RIHAM AHAMED ABDUL RAHEEM

HND COMPUTING IDM

Contents

List of figure

LITERATURE REVIEW

The literature review section of this paper contains information that is considered a supplement to understanding the research results and the conclusions I draw from the results that I will describe.

As such, the literature review of this documentation has the following goals:

- Provide background/basic information necessary for the reader of this research documentation to understand the research results.
- Give context to the description of the research documentation, so that the reader becomes more interested in the purpose of this research.
- Ensure that the reader has confidence and can rely on this research documentation for his own purposes and purposes.

SQL – THE DECLARATIVE LANGUAGE THAT RUNS DATABASES.

Sequel, pronounced "es-kue-el," is the language that machines use to communicate with databases. SQL is the standard for interacting with Relational Database Management Systems, according to the American National Standards Institute. Allowing developers to perform Create, Read, Update, and Delete (CRUD) actions on a database is one of SQL's most fundamental yet powerful features.

When it comes to Database Management Systems, there are four main languages that allow appropriate query expression and data structure within a database.

They are as follows:

1. **Data Definition Language:** Defines the structure of data and ensures that data is stored in ways patterns can be identified. DDL statements enable definition of the metadata of a database. Creating schemas, tables, indexes and constraints within a database are all done using DDL.
2. **Data Manipulation Language:** Allows accessing and manipulating data that is within a database. Whenever a machine requires accessing something stored in a database, it must do so through an implemented layer of DML.

3. **Data Control Language:** This language has a lot of similarities with DML, but the key difference is that DCL allows the control of permissions to a database, ensuring access hierarchies and also provide the functions of DML. Execution of DCL statements is heavily transactional, and also consists of rollback parameters.
4. **Transaction Control Language:** Persisting changes made to the database is done solely by this language. Each particular statement that is executed to affect the database can be visualized as a "commit" and saving or undoing these commits is what TCL does.

All four languages used in database management systems are incorporated into the Structured Query Language. As a result, it is critical in the process of using databases. The graphic below depicts how each of these four languages has been conceptualized in SQL using commands.

CHOICE OF DBMS FOR DISCUSSION PURPOSES: MICROSOFT SQL SERVER

MSSQL, one of the most widely used Relational Database Management Systems, was first launched in 1989 by Microsoft in collaboration with Sybase under the version number 1.0. During its early phases, MSSQL was very comparable to Sybase releases, however the partnership between Microsoft and Sybase ended a few years later. Microsoft, on the other hand, made sure to keep its MSSQL rights.

Since then, MSSQL has risen to incredible heights, and with several tiers of MSSQL that a firm may subscribe to, it's become the industry's most recommended and popular option. As a result, given my previous experience with it as well as the extensive exposure it has had, particularly during the 1990s, it is an ideal candidate for examination if necessary.

DATA CORRUPTION, A PROBLEM EVERY DEVELOPER FACES

SQL Server is a platform that ensures the stability of a company's operations. It is, nevertheless, susceptible to corruption, just like any other database. It is the loss or corruption of data as a result of an external factor.

As a result of data corruption, the following are the three most typical errors:

1. 5172 SQL Error

2. SQL Fatal Error 823

3. 8946 SQL Server is a database management system.

In such a circumstance, the only option is to restore a previously stored backup, which emphasizes the need of creating backups.

5172 SQL Error

The physical information of the SQL Server is saved in a primary file that contains information about the pages. The first page, dubbed a header page, contains information about the MDF file header. It contains a variety of information about the information, such as the file's dimensions, signature, and so on.

A common error encountered by users while mounting the MDF in SQL Server is error 5172. When the MDF file gets sick or corrupted, this happens. When this error occurs, the header file's information has already been mismatched, making data access problematic.

This error is most commonly caused by:

1. Bad shutdown of the server in which SQL Server is in.
2. Malicious virus attacks
3. Sudden shutdown of SQL server.
4. Broken server hardware

Recovering from this error can only be done through the expertise of a Database Administrator or a System Admin that has expertise in the data sector.

SQL Fatal Error 823

I/O operations are performed by SQL users using Windows system calls. SQL checks for any errors linked to API calls after I/O activities are completed. In SQL Server, error 823 occurs if these API calls are incompatible with the OS.

The following information is included in this error message:

• Whether the I/O request is a write or a read

• The operating system's error code and a description of the error

The error 823 message indicates that there is an issue with the underlying storage machine hardware or with a driver along the I/O request's path.

- There are contradictions in the file system.
- If the database file is corrupted.

The only way to recover from this type of error is to move to a compatible MS Windows installed machine, fitting the System Requirements for the solution to run.

STACK OVER FLOW – WHERE DEVELOPERS LEARN, SHARE AND BUILD CAREERS

Stack overflow, inspired from the commonly known computer programming error, is a question and answer website that is very famous among any developer. It is the goto source for all the needs and questions of a developer. Whether you’re just starting out or have been in the industry for some time, you will definitely come across stack overflow.

In fact, whatever issue you may have while programming, if you google for solutions the first source that will popup will be from stackoverflow. Stackoverflow is a branch of the Stack Exchange network and was created by Jeff Atwood and Joel Spolsky.

I will be using the Stackoverflow developer survey 2021 results as my secondary research quite extensively.

OBJECT ORIENTED PROGRAMMING – IMPERATIVE PROGRAMMING, AN EXAMPLE OF THE OPPOSITE OF DECLARATIVE PROGRAMMING

Object-oriented programming (OOP) is a model that revolves around objects rather than a particular set of data or a predefined logic. Programming has always been viewed in the past as a way to simply give realistic substance to logic in such a way for a particular set of or single input(s), an expected output will be reached.

The need for OOP arose due to a set of developers finding the restriction of programming just being tied to logic, and instead required the ability on "how" to define the data that will then be manipulated by logic.

Ever since this particular model was implemented, the way programming was done, changed.

The first step when it comes to OOP is what's factually known as "data modeling". In this particular step, the programmer should identify what his data objects are and what types of relationships each object has with the rest of the objects.

After data modeling has been completed, the sum of objects are "generalized" as a class of objects, which will attempt to basically give a clear understanding or a labeling of some sort that will identify what type of data each object contains, which gives a direct idea on what type of logical or otherwise steps can be taken in order to manipulate the class.

This concept called classes will allow a programmer to create custom data types when needed to satisfy the purpose of the programmer that is not already incorporated within the language itself.

Classes also ensures reusability not only from the file it is defined but also in countless other OOP projects which makes distribution of OOP codebases very easy.

Custom or technically known as an abstract data type allows a programmer to employ data structures effectively to organize and manipulate data.

These "logical steps" are referred to as "methods" in OOP. Methods are employed by classes in order to grant the programmer to effectively and logically pursue any necessary requirement. You could say that methods grant the programmer an interface to interact with the objects in the class.

Classes play a huge role in OOP; one big factor that puts OOP apart from the rest is “inheritance”. OOP allows subclasses to be created from a class. Subclasses can be thought of as children to a parent class that can “inherit” the parent’s attributes.

Inheritance allows a programmer to, through creating subclasses from a parent class analyzes the objects of a class, cuts the unnecessary wasting of time on re-inventing the wheel and efficiency during the process of coding.

The OOP model also provides security to its objects through what’s known as “encapsulation”. Encapsulation is concerned with giving access to objects only when there is a necessity in giving access which in other words means that the likelihood of the objects in a class being corrupted can be reduced effectively.

While corruption is reduced, this is a major boost to the security of the application written using OOP concepts, as it reduces the vulnerability of unauthorized access to data of a class.

PRIMARY AND SECONDARY RESEARCH DATA COLLECTION METHODOLOGIES

Sampling existing system documentation

Collecting facts from existing documentation rather than human sources is the best technique to examine the existing system. There are several types of documents that can be used to gather information from existing documents. These are some of them:

- E-mails
- Customer complaints
- Suggestion box notes
- Reports that document problems
- Problem performance reviews
- Samples of completed manual forms
- Reports and samples of completed computerized forms

Randomization and stratification are two of the most often utilized sample strategies. Randomization is the method of selecting sample data at random. The systematic technique of determining the variance in the sample data is known as stratification.

Questionnaires

Questionnaires were found to be difficult to implement directly due to the extraordinary circumstances of the country's COVID-19. Therefore Online Questionnaires were conducted with the help of Google Form

Questionnaires are also one of the most effective fact-gathering tools for gathering data from both users and workers. However, surveys' greatest strength is also their greatest weakness. A questionnaire is biased from the standpoint of the user; each user may have varying levels of expectations and knowledge to back up their understanding, making it difficult to reach a conclusion.

There are two types of questionnaires:

1. **Free-format:** Users are allowed to answer questions freely; response is not mandatory. This is ideal for situations when you require reviews, feedback, opinions, possible improvements and experiences.
2. **Fixed-format:** A predefined format of questions is put, then each user has to answer the question. Response is mandatory as the questionnaire is structured. Multiple-choice questions, rating questions and ranking questions are usually used in this format. Ideal for situations where the analyst is trying to put together the expectations, requirements, scope and the outlined structure of a solution or system.

Prototyping

Prototyping is a specific research technique for gathering the system or solution's need portion. Prototyping is a process in which a small portion of an existing system's working model is sampled, observed, interacted with, and then understood. This is also utilized in the software development cycle's design stages.

After a reliable knowledge base has been discovered and fathomed, the optimal prototyping techniques may be extracted. Before employing prototyping, a long number of information must have been compiled. This way, prototyping may be used to test the correctness of previously discovered information.

MERITS, PITFALLS AND LIMITATIONS IN ABOVE RESEARCH DATA COLLECTION METHODS THROUGH COMPARE & CONTRAST

Similar research approaches will be examined and contrasted in this section. In my opinion, there is no use in comparing or contrasting dissimilar research methodologies because the ultimate goal of each approach is distinct, or to put it another way, their goals are different. So, rather than comparing prototyping to questionnaires, a better analogy is prototyping to sampling.

Each comparison/contrast will begin with a rationale for why I chose to compare or contrast them in the first place. To demonstrate that they have similar ultimate objectives or purposes and so are comparable and contrastable.

Prototyping vs Sampling existing documentation

Both research techniques use the same core, they sample different things to understand.

1. Sampling is a research technique that can be used in the earlier stages of analysis, prototyping must be used in the later stages.
2. Sampling is usually used to gather facts, prototyping is used to gauge the accuracy of facts that have already been found.
3. The facts found through sampling may be outdated, but prototyping provides accurate upto date facts based on a presently existing system.
4. Prototyping requires a certain level of knowledge and experience to perfrom, but sampling does not, it just needs the ability to read.
5. Prototyping needs a working modal to exist which is rare than finding system documentation for sampling. So the likelihood that you will use sampling is higher than prototyping.

Questionnaires vs Interviews

Both techniques rely on an outside audience, the facts that you obtain through these techniques may or may not be biased.

1. Questionnaires do not require the analyst's presence, while interviewing requires not only your presence but an interviewee as well.
2. Good communication is a mandatory requirement of interviewing; questionnaires does not have this requirement.
3. Interviewing may take long periods of time on large audiences, questionnaires can be collected within a couple hours if you wished to.
4. Interviews are ideal in situations where feelings, experiences and what not are a strong requirement to understand the problems. Questionnaires fail in times like this, because feelings can't be expressed accurately through words.
5. Questionnaires for large audiences require a budget enough to design the questionnaire, test it and then print it. Interviewing has a low budget requirement.

Justification for the choice of data collection methods

For the primary research, I have chosen to use a 5 part **online questionnaire** and the reasons as to why I have chosen a online questionnaire are:

- Allows to directly gauge personal opinions and experiences in an indirect manner, which allows the participant to psychologically feel safer and less nervous opposed to something like an interview or observation.
- Ensures that information that is gathered are more scoped to the objectives of the research.
- The least cost, ethically proper and time saving method to get personal opinions are a questionnaire.
- There is no interference by the researcher on the participants as they can take as much time to think and give a personally satisfying answer to the questions, hence improving the quality of the information gathered.

For the secondary research, I have chosen to analyze the research results of the **Stackoverflow Developer Survey of 2021**, and the reasons are:

- Stackoverflow is a medium for developers, and my audience are developers, the research was not only conducted to identify trends in the industry but also sensitive things like gender, age ranges and what not which I do not store. As such I am not responsible for the security of the data.
- The research results of the developer surveys are publicly online for use and analyzing as long as the proper credit is given. Several other industries and developers have utilized these research results for their personal projects or researches.
- Both quantitative and qualitative analysis can be conducted as the results have been released as bar charts with percentage values.
- Over 190,000 developers have participated in both of these developer surveys in total and as such the conclusions I take from them must be quite valid.

PRIMARY RESEARCH – ONLINE QUESTIONNAIRE

preview of the online questionnaire

1. **Do you consider yourself a beginner or an expert in the programming industry?**

   (3+ years of experience would make you an expert by default) – In order to classify my responses to 2 categories so that the opinions of actually experienced developers don't fluctuate with the responses of the beginner developers. If there were some fluctuations the information will be not be normally distributed to the 2 sections.

2. **Do you think that SQL is difficult to learn? Why/Why not?**

   To understand the participant's opinion on my research problem, whether if it is actually difficult or not difficult to learn SQL as a whole and what they personally think is so difficult about it.

3. **Have you had sufficient experience with SQL?**

   (Knowing basics counts as sufficient) – So as to filter out the opinions of those that have not had sufficient experience with SQL.

4. **What SQL statement(s) have you had the most difficulty with?**

   To directly identify what component(s) it is in SQL that makes it difficult to learn.

5. **How would you rate your experience with SQL in production applications from 1 to 10?**

   Gathering quantitative information makes it very easy for me to come a step closer to answering my research problem and grasp my objectives.

Outcome and Conclusions

After the questionnaire had been out on the internet for more than a week, 54 people had completed it, and out of those 54, roughly 39 had the sort of replies I was hoping for. The remaining 15 replies were either spam or those that I opted to screen out based on the answer to the questionnaire's third question. As a result, the results will only take into account the replies of 39 developers.

Outcome #1 – 76% (29) of the respondents were beginners and 24% (11) were experts.

This means that I have nearly a 1 to 3 ratio of experts to beginners answering my questionnaire. As such I have sufficient participants in order to draw mid-range valid conclusions from their answers.

Outcome #2 – 81% (23/29) of the beginners said SQL is difficult and 15% (2/11) of the experts said that SQL is difficult.

This means that my research problem is valid to a point, and that there is in fact beginners struggling with SQL, however though as it seems it's become obvious that the experts don't seem to have as much trouble. But 11 is quite a small number, if this number were higher I am sure that there'd be a much higher percentage among experts.

Outcome #3 – 72% (39/54) had basic knowledge of SQL

This question was more of a way to filter out responses and increase the quality of the information than a way of gathering information. It allowed me to filter out the responses I didn't want, much easily.

Outcome #4 – Generally among the respondents, it seemed apparent that JOINs and nested queries were the factors that made SQL more difficult. However though, most of those respondents also agreed that they're very powerful features and as such the difficulty is a rich compensation.

Essentially means that there is in fact a few components in SQL that make SQL difficult. 2 of the identified statements have been shown to be JOINs and Nested Queries.

Outcome #5 – 67% of the ratings were above 7

This means despite the difficulty concerns of SQL there is, that the difficulty is a calculated risk. The power and flexibility of declarative programming certainly cuts down the difficulty, which is probably why most of the ratings were above 7.

SECONDARY RESEARCH – STACKOVERFLOW DEVELOPMENT SURVEY RESULTS 2021 OUTCOMES & CONCLUSIONS

All credits goto the respectful and wonderful community of Stackoverflow, and every piece of information that I've utilized in this research are credited towards Stackoverflow. I claim no ownership or rights over the information, I will only comment on the survey results and grasp conclusions.

Outcome #1: SQL – The #3 most popular language developer site in the world

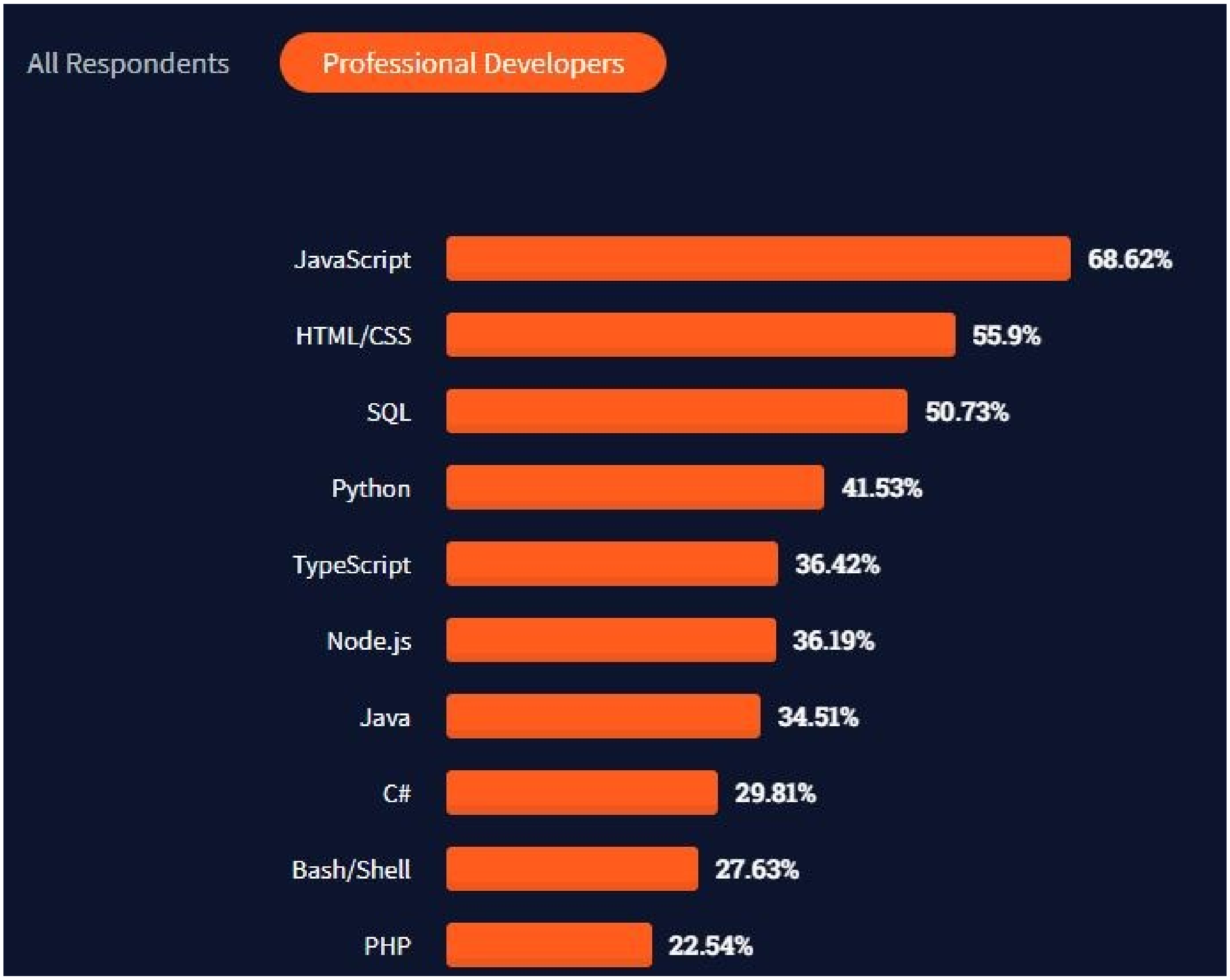


Figure 1: Most popular programming, scripting and markup language

As you can see, 50.73% of all respondents voted SQL as the third most popular language in the world for developers. This meets our second research objective "If SQL is demanded in the workplace", the result #2 will back up this claim. SQL has triumphed over imperative languages like Java, which most developers don't have much trouble mastering the basics. Essentially, this means that SQL's place in this world cannot and will not be replaced by other technology.

Outcome #2: SQL – The #4 most popular language developer site in the world

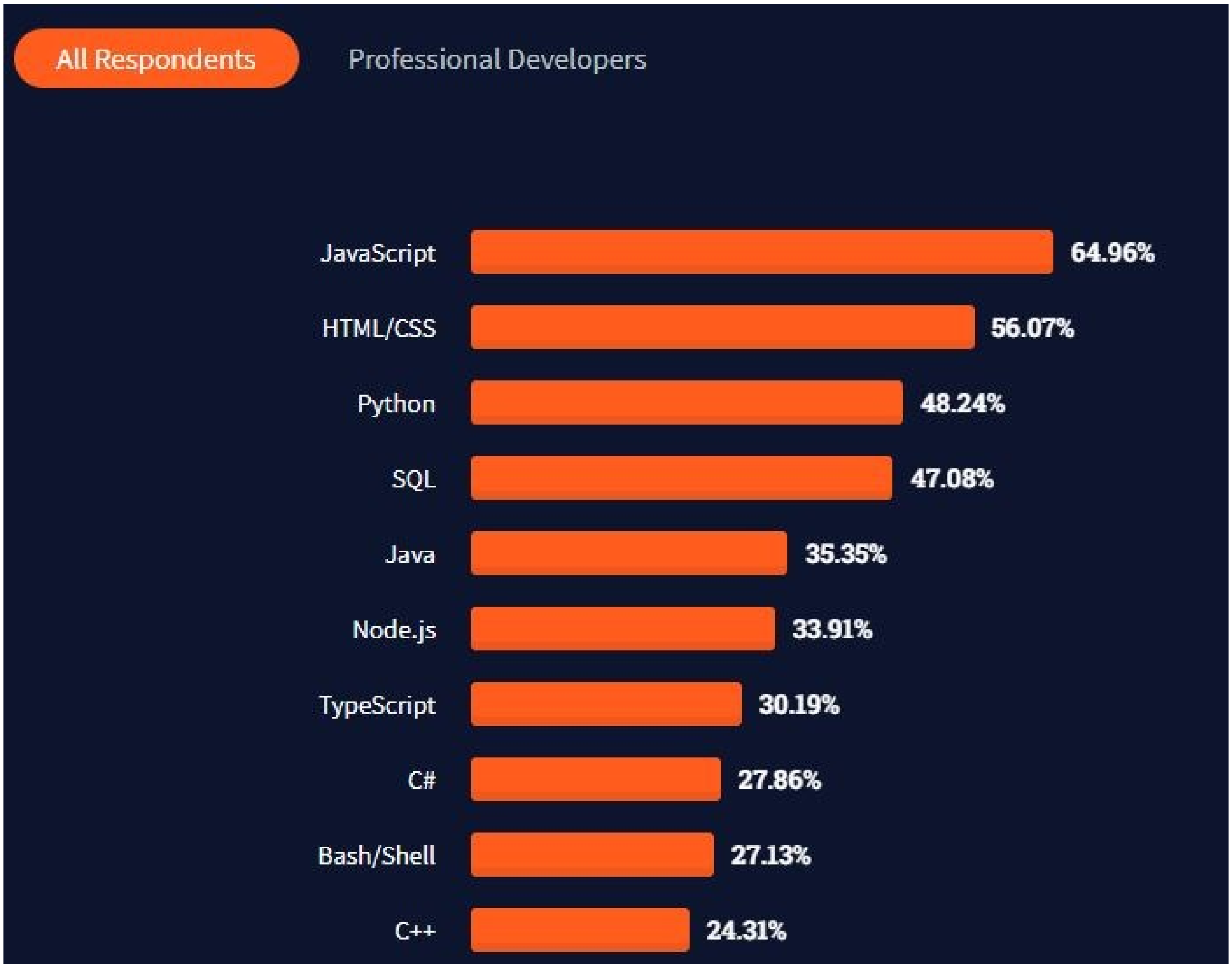


Figure 2: Most popular programming, scripting and markup language All respondets

As you can see, 47.08% of all respondents voted SQL as the fourth most popular language in the world.

Outcome #3 – 61.83% votes SQL as #13 most loved language while 38.17% votes SQL as dreaded language

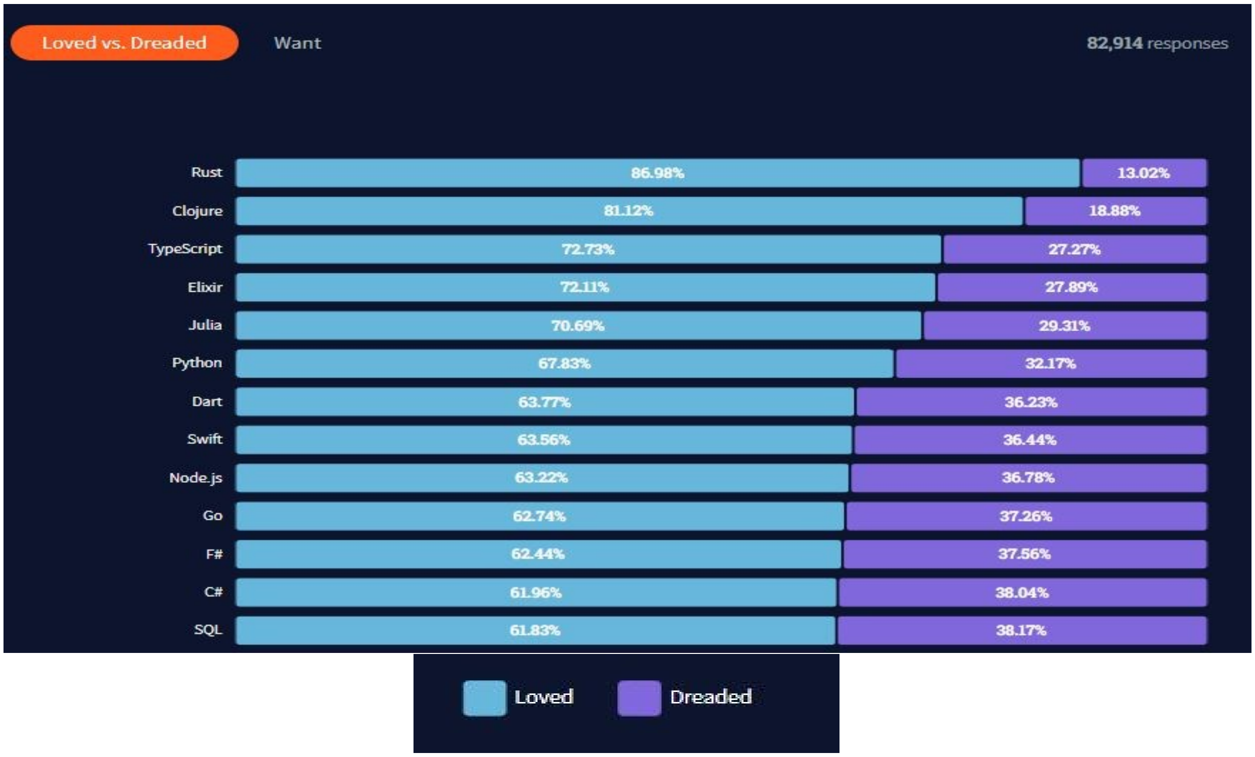


Figure 3: Most love dreaded languages

Isn't this result a bit of a contradiction? Generally speaking, more individuals enjoy the language than find it repulsive, yet it is feared by a significant number of people. What gives you the impression that this is the case? Simple, learning and mastering it is harder. This result should not be interpreted as implying that SQL is not difficult to learn just because more people like it; the reason for this is that SQL was voted the #15 most feared language by 38.17% of those polled.

If so what is the conclusion that can be gained from this? Two conclusions can be taken from this outcome:

- There are people that love the language this is for certain, as such this proves that the language is very powerful and satisfies the needs of the developers, **this essentially proves that the language has significant influence in the job industry.**
- There are also people that dread the language, this means that the language is actually difficult despite it being lovable. **As such this proves that there must be pitfalls in the process of learning or working with the language.**

CONCLUSION

The research ended on the 30th of September 2021, 4 days after the actual planned date of finishing the research. Over the course of the month, the primary and secondary research methods were all utilized in such a way to find the answer to the research question and hopefully satisfy the research objectives. As such the following are the main conclusions that could be made from all the outcomes of this entire research:

- SQL is a very powerful and popular declarative language in the world, mainly because it lacks conditional logic. It is more of a language that says what to do than how to do it, as such for contexts and purposes like data science, machine learning etc. it is irreplaceable.
- SQL cannot and will not be replaced by any other declarative language for database programming, the secondary research proved this with very large numbers that there is no competitor much less that can provide the functionalities of SQL.
- SQL is in fact a difficult language to learn, notably from the primary research outcomes saying that Nested queries and JOINs are culprits.
- However, the difficulty of the language is compensable due to it being extremely powerful. The 2 culprits nested queries and JOINs are very powerful features in SQL. As such it is reasonable to say that they are quite difficult to utilize, satisfying the 1st research objective.
- Both the fact that SQL is demanded and is a difficult language means that it is safe to assume that there must be quite a few pitfalls in the learning or mastering process of SQL. As such it is reasonable to say that the 3rd research objective is satisfied.

The conclusion to the research question **"Is SQL difficult to learn"** is **"Yes, it is difficult to learn SQL but the power, features, job demand and popularity of the language overwhelms the difficulty".**

REFLECTION ON RESEARCH METHODOLOGIES AND POSSIBLE FUTURE IMPROVEMENTS

I would just like to say that the research was very successful and I learned quite a bit about task/project management, the different levels people are at, and the subtleties in the development industry that most developers don't know about. This was my first attempt at researching something computer based and there were so many problems in the middle of it all, but every step I took got progressively closer to reaching the research goals. Finally, I answered the research question and solved the research problem. Any member of the target audience who wants to read my research documentation will be able to learn a lot.

Using questionnaires and the results of stackoverflow development surveys was quite difficult, but it was worth it. Especially using the research results, I had already compressed the information of many respondents, which allowed me to perform an effective quantitative and qualitative analysis of the research data. The questionnaires were useful for the main research method, but improvements are possible. I would say that if I were to re-examine the same topic, I would combine questionnaires with live interviews to get a more information-rich set of results and conclusions.

Alternative research methodologies for improving

Some alternative research methodologies I could have taken instead would be interviews and use sampling methods of some type. Interviewing the students who had difficulties in the institute where I got the inspiration to start this project would have added to the total number of participants for this research. The sampling methods could've somehow allowed me to find more indirect connotations between the research objectives as opposed to those that I have taken from the secondary and primary research. Basically, these 2 research methods opposed to or with the 2 research methodologies that I have already utilized might have bought more context and richer information to use as the basis of this research.

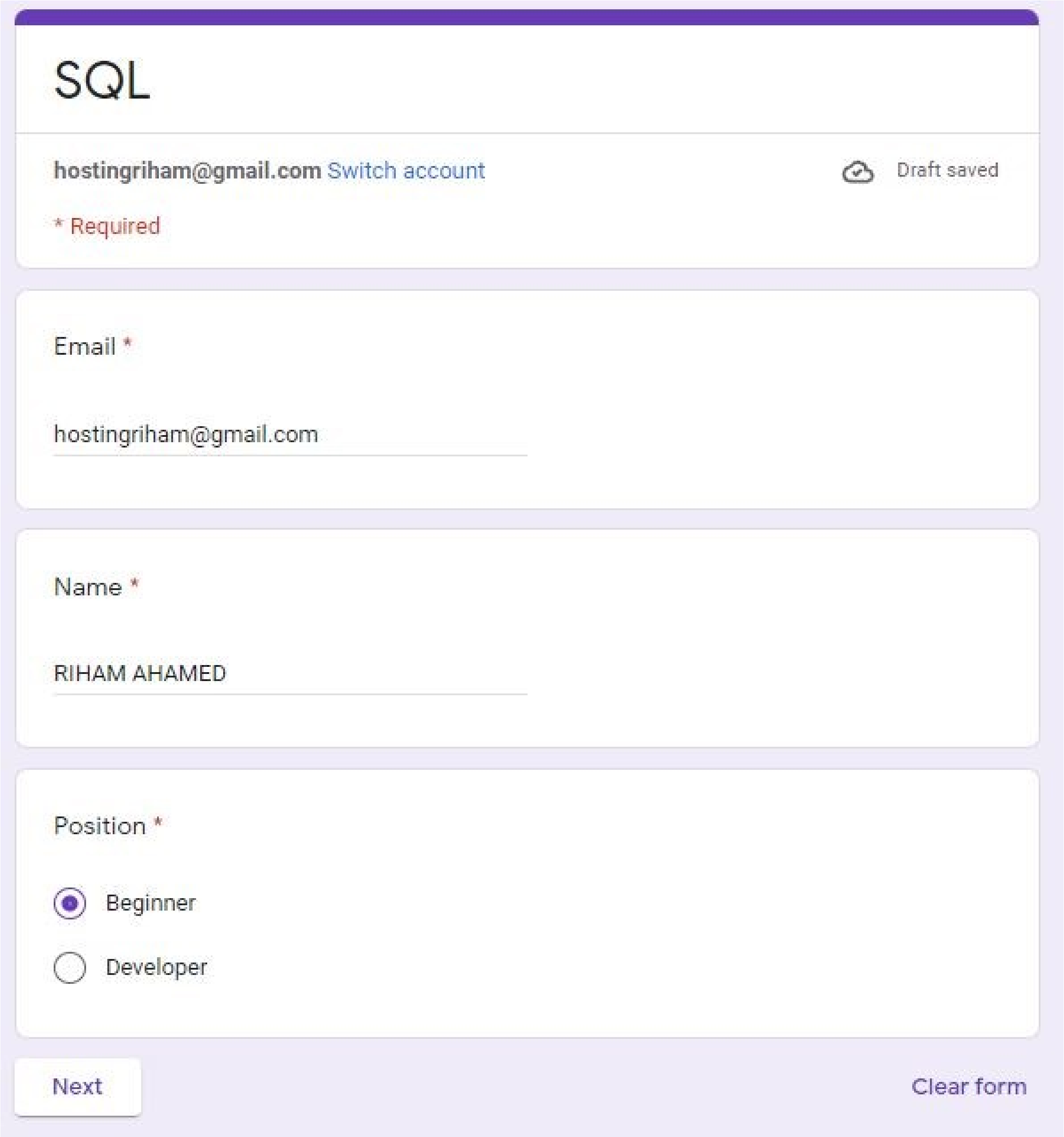
Lessons learnt

This research lasted a month and I learnt a lot of things and improved on a lot of things that I already possess, some of those lessons are:

- SQL has a very high job demand around the world, I did not ever think that a declarative language that was used alongside some imperative language would have a such a high direct demand.
- More beginners seem to struggle with SQL than I originally thought, in fact there are a lot more misconceptions like SQL is not needed in fullstack web applications or that NoSQL is superior to SQL because it provides a more OOP or a functional approach to database programming.
- Pacing yourself during a research is very important, I tried to tackle multiple things at the same time which resulted in a delay in my original plan.
- Thinking about costs, ethics and security is very important in a research that includes living participants.
- Structuring your research skeleton is crucial as this allows you to understand what has to be researched and how it has to be researched.

ATTACHED FILE

Google form link : https://forms.gle/6xujwYh1m5KbjKL97



SQL

hostingriham@gmail.com Switch account

Draft saved

* Required

Email *

hostingriham@gmail.com

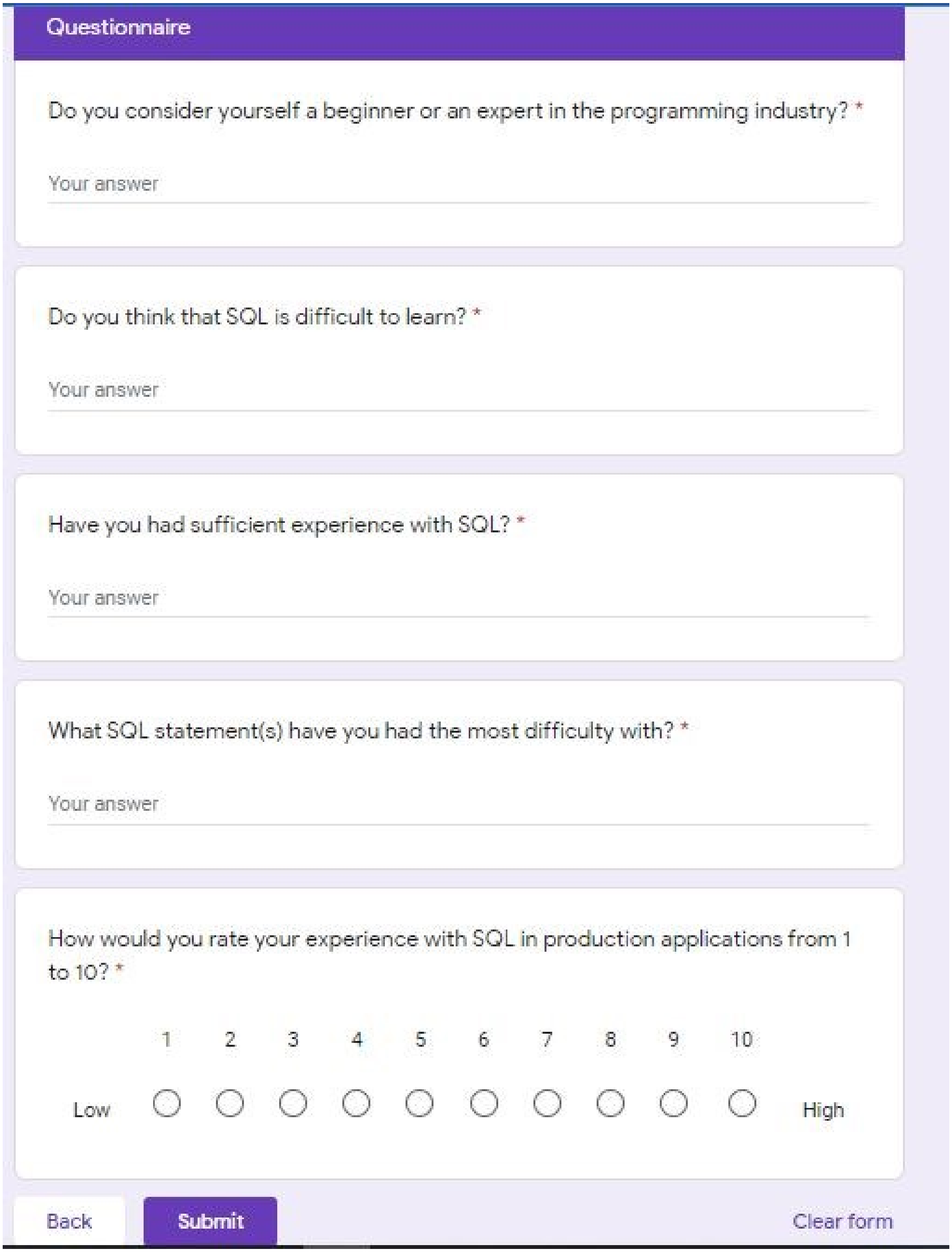
Name *

RIHAM AHAMED

Position *

● Beginner

○ Developer



Questionnaire

Do you consider yourself a beginner or an expert in the programming industry? *

Your answer

Do you think that SQL is difficult to learn? *

Your answer

Have you had sufficient experience with SQL? *

Your answer

What SQL statement(s) have you had the most difficulty with? *

Your answer

How would you rate your experience with SQL in production applications from 1 to 10? *

	1	2	3	4	5	6	7	8	9	10	
Low	○	○	○	○	○	○	○	○	○	○	High