**EXECUTING SQL QUERIES USING VOICE COMMANDS**

**S. BHAGYA REKHA1, K. SAI SRINIVAS2, B. AMULYA3, MD. ZUBAIR HUSSAIN4  
1** Assistant Professor, Department of Information Technology, Vignana Bharathi Institute of Technology.  
**2** Student, Department of Information Technology, Vignana Bharathi Institute of Technology.  
**3** Student, Department of Information Technology, Vignana Bharathi Institute of Technology.  
**4** Student, Department of Information Technology, Vignana Bharathi Institute of Technology.

**ABSTRACT**SQL queries are used to Create a table, remove entries, update tuples, or delete tables. Executing SQL queries might look simple, but a single error might lead to some irreversible problems. It is a long, tiring and a hectic process to remember the queries and make sure that the query is executed perfectly. If not, a wrongly executed query would lead to improper data handling and permanent loss of data. Another alternate simple way of executing a query, is to speak them out loud and clear, and let the machine deal with the query. Our project uses simple algorithms and inbuilt python functions to make this possible. And with simple knowledge about NLP procedures and working of it, the time complexity could be reduced significantly, and similar results could be observed with minimal knowledge about the subject along with high efficiency.

**INDEX TERMS**Voice commands, NLP, Machine Learning, SQL Commands, API, Querying

**INTRODUCTION**Python and Database are two of the most used technologies in everyday technical world. Our main aim of this project was to combine these two booming technologies, and use python to execute SQL queries, but instead of typing out the query, we dictate it in normal speech and that is being converted to an SQL query. Data is given to phones using keyboard but today, we are using another input medium that is the voice input. The algorithm works by extracting the key words in the query. We then generate the suitable query and execute it to produce the desired output. This application can be used by anyone who feel voice-based input is better than the conventional system, or are not familiar with the syntaxes of SQL or Python. With new developments in voice-based interfaces, like Siri from Apple and Google Assistant, queries can be handled more effectively using databases and voice-based interfaces. This proposed system has the major advantage of being completely hands-free compared to conventional query structures. The transition from imperative to declarative programming is still quite challenging for students even though declarative programming is arguably more intuitive. The purpose of this project is to assist people in their learning process who are not very familiar with executing SQL queries

**BACKGROUND AND RELATED WORK**

1. Generating Select Query In Mobile Applications

Conventional data are given to Smartphone using keyboard and by just pressing on an icon. People always use their smart phones because this keeps them connected with the outside world. This just doesn’t mean through calls, but also through social media and other connections. So, in addition to search by voice and voice shortcuts like "Navigate to", we included a voice-enabled keyboard. With the Android SDK, you can access numerous features and services to make your applications more accessible to visually disabled and hearing-impaired users. Examples include speech recognition frameworks, Text-To-Speech (TTS), etc. Android applications can leverage speech input and output. The Android SDK includes a speech package that enables speech input and speech output using speech recognition and text-to-speech services. An android database handling function transforms this generated query into a format like this:

**Cursor c = db.query**

Diagram

Description automatically generated

Fig 1 Querying using voice commands

2. Cyrus

Being revelatory, SQL stands distant better; a much better; higher; stronger; an improved a higher chance at being the programming dialect for conceptual computing following to normal dialect programming We look at the plausibility of utilizing SQL as a backend for characteristic dialect database programming. Particularly from keyword-based SQL questioning, catchphrase reliance and SQL’s table structure constraints are significantly less pronounced in our approach. We display a portable gadget voice inquiry interface, called Cyrus, to subjective social databases. Cyrus bolsters a huge sort of inquiry classes, adequate for a section level database course. Cyrus is additionally application autonomous, permits test database adjustment, and not restricted to sets of catchphrases or natural language sentence structures. Its cooperative blunder reporting is more natural, and IOS based portable stage is additionally more available compared to most modern portable and voice enabled frameworks. Although explanatory programming is seemingly more natural, study still discover the move from basic dialects to SQL colossally difficult. They frequently battle to make complex inquiries that are difficult in semantic structures, particularly those including settled sub queries or GROUP BY functions.

3. Voice Based SQL Query using Natural Language

This proposed framework has been named as Natural Language to SQL generation in extricating semantic information from social web sources. But it leads to trouble to extract the semantic substance of the query. After this, in 2016, Garima Singh proposed a strategy called an algorithm to convert common language into SQL queries for relation database from the Three level engineering of NLTSQLC which for the most part result within the downside of Computational of a few information. One of the major areas in computer science is natural language which is primarily concerned with the interaction happening between the computer and the human language. There it contains more appealing regions within the interaction between human and the computer. This incorporate spoken dialect that consolidate speech and natural language. There should be a prior knowledge about DBMS (Database Management System) within the heading of extracting information from the database. The DBMS is a program created for the reason of storing and controlling the information in a database. Here an unspecialized client finds inconvenience in extricating the information.

To find an arrangement for such an issue and ease human interaction with computers, Natural language processing strategies are worn. Natural language processing has applications in varied segments like tourism, where a traveler can get data around the popular tourist spots in a specific city, the lodgings accessible, best places adjacent and so on. Our work spotlights on extracting the proper query by giving the input within the form of voice.

**ARCHITECHTURE**

The system accepts the spoken query as its input and it is sent to speech recognition engine, the output of that phase will be the input text query which is in the mixed format. The correct input query is extracted and further sent to tokenization. Tokenization is the method of splitting the sentence into individual words and storing it in the list. Unwanted tokens are removed after storing it in the list. The tokens are mapped with the pre stored synonym database which contains the words with its synonyms. The refined text is then sent to text translator. Text translator contains clause extractor and mapper. Through which an intermediate query is being generated and tokens are mapped with the table name and attribute. The result of this phase is the SQL query. This SQL query is operated on the database and correct results is being displayed on the interface. The SQL query will be displayed on to the command prompt.

Algorithm

Step 1. Accept the input from the user in the form of speech.

Step 2. The speech is converted into text by using speech recognition engine.

Step 3. The correct form of the statement is saved and other statements are discarded.

Step 4. Divide the input query and store it in a list, i.e. tokenizing the input query statement

Step 5. Remove the unnecessary token from the list like, the, an, etc.

Step 6. Map the tokens with the table name and attributes of the database.

Step 7. Now find the tables which will contain the pair of ((attribute which do not belong to the table in the query), (other attributes present in the table in the query)).

**ADVANTAGES**

* Separates usefulness from presentation.
* Clear partition - superior understanding.
* Changes constrained to well characterize components.
* Effective organize performance.

**MODULES**

Speech Recognition

It is the intrigue sub-field of computational linguistics that creates strategies and technologies that empowers the recognition and interpretation of spoken dialect into content by computers. It is additionally known as "Automatic Speech Recognition" (ASR), or just "Speech To Text" (STT). It incorporates knowledge and research within the phonetics, computer science, and Electrical Engineering areas. A few speech recognition frameworks require "preparing" (too called "Enrolment") where a person speaker peruses text or separated vocabulary into the framework. The framework analyses the person's particular voice and employments it to fine-tune the recognition of that person's speech, coming about in expanded accuracy. Systems that don't utilize training are called "Speaker Autonomous" frameworks. Systems that utilize preparing are called "Speaker Dependent".

Text To Speech

The method of changing over talked speech or sound into content is called Speech To Text Converter. The method is ordinarily called Speech Acknowledgment. Speech recognition is utilized to characterize the broader operation of determining content from speech which is known as Speech Understanding. We frequently relate the method of recognizing an individual from their voice that's Voice acknowledgment or Speaker recognition, so it is off-base to use this term for it. Systems for the most part utilize the articulation model. It is imperative to memorize that there's nothing like a widespread Speech Recognizer. If you need to get the finest quality of translation, you'll be able to specialize the above models.

**IMPLEMENTATION SCREENS**

Text

Description automatically generated

*Fig 2. Output when asked to display all the records in the table*

Text

Description automatically generated

*Fig 3. Output when a particular condition is mentioned*

Text

Description automatically generated

*Fig 4. Fetching only the required columns.*

**CONCLUSION**

This project describes a to be converted to an API called “Executing SQL queries using Voice Commands”, which provides the user with an option to speak their normal language which is then converted to an appropriate SQL query. Voice Inputs are ubiquitous and becoming more and more sophisticated. This has been changing the landscape of people’s daily life and has opened the doors for many interesting innovations and Ease of life applications. This project is a core building block behind applications which no longer require traditional learning and executing queries, but rather speak them out and let the system convert it for the user. It takes the microphones’ reading as inputs and predicts a possible query for it. This paper presents a comprehensive survey of the recent advances in voice recognition, and how it can be implemented to achieve various types of projects which lead to ta better and easier life. We introduce the basic concepts of how to use the voice recognition and to convert the natural language into something which can be made into another programming language.

The projects first service can be extended in future. This can be achieved by adding few more additional decisions or add not just select privilege, but also try to grant admin privileges by implementing TCL and DDL commands into the program as well. At present, we strived to achieve the DML commands that is the SELECT query.

**REFERENCES**

[1] J. Weizenbaum, “ELIZA - A computer program for the study of natural language communication between man and machine,” Communications of the ACM, vol. 9, no. 1, pp. 36-45, January 1966.

[2] T. Winograd, “Procedures as a representation for data in a computer program for understanding natural language,” MIT AI Technical Report 235, February 1971.

[3] B. B. Huang, G. Zang, and P. C.-Y. Sheu, “A natural language database interface based on probabilistic context free grammar,” IEEE International Workshop on Semantic Computing and Systems, Huangshan, China, 14-15 July 2008.

[4] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum, “Developing a natural language interface to complex data,” in ACM Transactions on Database Systems, vol. 3, no. 2, pp. 105-147, June 1978.

[5] N. T. Dang, and D. T. T. Tuyen, “Natural language question answering model applied to document retrieval system,” World Academy of Science, Engineering and Technology, vol. 51, pp. 36-39, 2009

[6] English stop words list - Github. hŠps://tinyurl.com/y86xbn9e, November 2016. Accessed: January 18, 2018.

[7] A. Ahadi, J. C. Prior, V. Behbood, and R. Lister. A quantitative study of the relative diffculty for novices of writing seven different types of SQL queries. In ACM ITiCSE, Vilnius, Lithuania, July 4-8, 2015, pages 201–206, 2015.

[8] P. Brusilovsky, S. A. Sosnovsky, M. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou. Learning SQL programming with interactive tools: From integration to personalization. TOCE, 9(4):19:1–19:15, 2010.

[9] C. Chen and M. Chen. Mobile formative assessment tool based on data mining techniques for supporting web-based learning. Computers & Education, 52(1):256– 273, 2009.

[10] B. Genc, M. Siala, B. O’Sullivan, and G. Simonin. Finding robust solutions to stable marriage. In IJCAI, Melbourne, Australia, August 19-25, 2017, pages 631–637.

[11] M Uma, V Sneha, G Sneha, J Bhuvana, B Bharathi. "Formation of SQL from Natural Language Query using NLP" , 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019