

## OUTLINE OF A FIRST DATABASE COURSE

*Jeanne M. Baugh Robert Morris University, [baugh@rmu.edu](mailto:baugh@rmu.edu)*

### ABSTRACT

*Most Computer Information Systems programs require at least one database course for their majors. This paper describes a case study of a first database course in which multiple database platforms were introduced throughout the semester. The course was made more difficult in that half of the students had an extensive background in database design and the other half had no database background at all. This made the course even more challenging for the instructor. Students were first taught Microsoft Access while learning database basics including entity relationship diagrams (ERD). They then moved to MySQL for beginning SQL and finally learned Oracle and more complicated queries along with PL/SQL. Students were also expected to apply the traditional database concepts to actual database storage problems by creating an individual project using MySQL. Each database platform, Access, MySQL and Oracle presented various challenges to the students. This approach to teaching a beginning database course gave the students a great deal of SQL experience and allowed them to be more confident of their skills in that area. They were able to compare the programming of various tasks, such as triggers, using different platforms. They gained an appreciation for the similarities of the different platforms and were able to apply a concept learned on one database platform to another platform, something they will encounter in the real world after graduation.*

**Keywords:** first database course, database platform, database project, IS education

### INTRODUCTION

All students majoring in an Information Systems degree are usually required to take one database course as part of the core for the major. Such a course would emphasize database design and implementation based on a thorough analysis of requirements and information modeling. An introduction to relational database technology would be provided, highlighting the use of Structured Query Language (SQL) and report generation. An advanced course could go into more detail in the areas of advanced SQL considerations, PL/SQL, database performance and security issues, multimedia, parallel, and distributed database management systems, data warehousing, and object-oriented databases. The course described in this paper is a required introductory database course. This paper details the issues encountered by the Author in that half of the students had taken an entire semester of database design and the other half had no idea of what a database was. The Author decided to teach the course using three different platforms, MS Access, MySQL and Oracle.

Students should be given as much real world experience as possible (Seyed-Abbassi, King, Wiseman, 2007). To give students such experiences, a bridge must be established between business and academia. While teaching beginning database for many years, it has been the Author's experience that students have a relatively naive view of the complexity of business in the real world.

Additionally, giving each student an individual database project centered on a topic or organization of their choice allowed them to be more interested in the successful completion of all programming tasks for that project (Scott, Stanford, Benson, Alluri, Martin, Klotz, Ogle, Schiff, 2013).

Two different approaches to teaching a database course could be used. One focuses on the actual application software, Oracle, SQL Server etc... The other focuses on the theory of database design and implementation, with the vehicle (the database package) being secondary. Even early research has shown that the selection of the specific database software can be an essential element of a beginning database course. As far back as 2003, MySQL was selected for such a course (Denton, 2003). While the vehicle platform is of course important, a database cannot be implemented

until the design is clean. (Deperlioglu, Sarpkaya, Ergun, 2011). Alternatively, perhaps better stated, it cannot be implemented successfully if there are normalization problems.

Research has shown that student success in a database course is increased when real database examples are brought into the classroom (Dobesova, 2016). Additionally, students will “design effective modern information systems when the learning is embedded in problem-solving contexts that are relevant in the real-world” (Connolly, Begg, 2006).

Some instructors are implementing large database examples to simulate the real world. MS Access’s Northwind database has been used to show many of the various tasks found in a large database system, such as multiple tables, queries, forms, reports, Macros, VBA Class Objects, functions and modules (Dyer, Rogers, 2015).

This Author has worked as a database consultant for many years, with the most recent projects in the Orthopedic department of a large city hospital. Many of the experiences were brought into the classroom. This allowed the students to see what a database designer actually deals with in designing and implementing database projects for real users. Students were also given actual database problems from live medical databases (with data scrubbed, of course).

Assuming that any database cannot be successfully implemented without a good design, the Author faced an unusual problem with the database course described in this paper (Wynn, 2018). Approximately half of the students had a prior semester of a database design course (with no implementation), while the other half had no database class at all. The students were Computer Science majors and Computer Information Systems majors. The Computer Science students were not required to take a database design course. The Author was told this course was to be a database implementation course. How can a database project be assigned, when half of the students had no idea what an entity, attribute, relationship, cardinality, query, or any other basic database concept is? How is it even possible to start teaching database implementation with no background in database design? These were the questions the Author pondered! A plan for this class was quickly devised, where a crash course in database design was given to the students before beginning the database implementation. To the students who had already done design work, this was a good review of ER diagramming.

### **COURSE PLAN – MS ACCESS**

This course had an enrollment of 34 and as stated, about half had no idea what a database was. This first section of the course emphasizing design and Microsoft Access skills encompassed the first five to six weeks of the course. MS Access is quite user friendly while also being very powerful, supporting multiple tables and views. But, before even Access could be used, half of the students had to be brought up to speed as to what a database design actually was.

Communication skills of the database designer are extremely important. (Baugh, Davis, Kovacs, Scarpino, Wood, 2009). The students were provided with many examples of interactions with users (Hansen, Jorgen, 2012). Students were taught to “listen” to the Author, who simulated the users. Students not only need to hone their technical skills, but they need to develop a communication expertise that will allow them the best possible chance at drawing out of the user what their data needs may be. Often the users will not know what to tell the database designer about their data needs. Sometimes they will even try to tell the designer what actual fields should be in a table. They will speak in terms of the outcomes of the database, for example the desired reports. Students were taught to listen to those required reports and “back design” the database from them. They were also taught to ask questions (as many as possible) to help understand how all of the data fits together. For example, if they were designing a surgical database, and they were told that the Doctor wants to store the type of anesthesia used in the surgery, the student should then ask if there could be more than one anesthesia administered during the surgery. The students were provided with many examples of user issues and some “mistakes” the Author has had to deal with in designing and implementing databases in the medical field. The “anesthesia” issue is one mistake this Author made on an actual database. By not accounting for multiple anesthesia, the students saw that a mistake could cause a major database table change. They were able to appreciate how difficult it was to deal with this mistake after six months of data had been entered. Seeing how actual design errors cause very big problems was a valuable lesson for the students (Oswal, 2014). By knowing what the user expects from the database, the design can be adjusted accordingly before data has ever been entered.

Although database textbooks contain design assignments and projects, this Author has written many design problems that the students worked through. The kind of text that is written in these design assignments has come from the Author's many years of dealing with users and helping them define what their needs really are. The Author acted as the user in these scenarios prompted the students to ask as many questions as possible to help gain an understanding what the design should be. The students then created the entity relationship diagram (ERD) for each problem. A few condensed examples of these design problems are:

- a) You are to design a database for the Giant Eagle Grocery store. Giant Eagle is able to track the buying habits of its customers by having them use their advantage card when shopping. You will need to store all information concerning each purchase that is made by the customer – even those customers who do not have an advantage card. The manager would like to know buying habits along with product sales information.
- b) You have a friend who owns a restaurant and he is having trouble keeping track of banquet information. Recently 25 people showed up to celebrate an office Christmas party and the manager had no idea they were coming. There was no information stored on what dinners they wanted etc... No one in the restaurant was willing to take credit for the mistake.
- c) You are the coach for a youth soccer team. Your players are children between the ages of 6 and 8. You want to keep track of goals scored, who scored it and who the opponent was. You must also know who played in the games, because the players will get a trophy if they attend all 10 games. You must also keep track of who is allowed to pick up each child after practice.

A number of tricks/or shortcuts have been developed by the Author to help the student to create the entity relationship diagrams. For example, the students were taught to listen for any plural words in the problem statements. (a patient has **surgeries**, a student is enrolled in **courses**, an insurance agent has **clients**) When a plural exists, that is an indication that the entities should not be stored together.

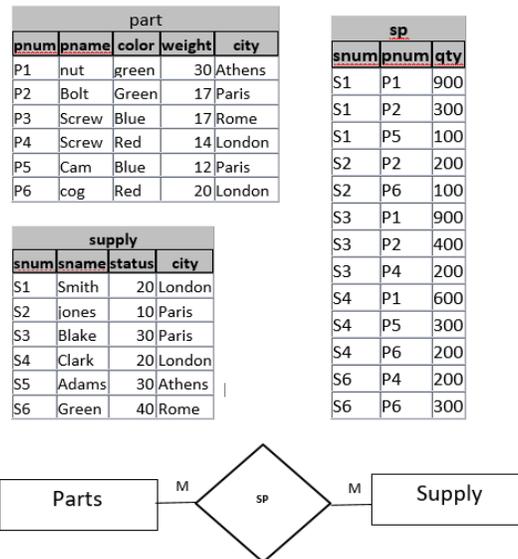
Another trick the Author presented concerning the E-R design is a trick to define the cardinality or relationship type. Is the relationship one-to-one, one-to-many, or many-to-many? Defining the type of relationship is often very difficult for the first time database student (Seyed-Abbassi, King, Wiseman, 2007). The students were taught to start with one of the entities in the relationship and ask if each of those entities can have multiple connections on the other side. If they say Yes, then an "M" is placed on the other side. Then they go to the other side of the relationship and ask the same question in the opposite direction. Using the ERD in Figure 1, of the appendix, the method is applied as such: Start with the Patient entity set and ask, "Can each patient have multiple surgeries"? If the answer is YES, then an "M" is placed next to the surgery entity set. Then go to the surgery set side and ask, "Can a surgery be related to more than one patient"? In this case, the answer is NO, so a "1" is placed next to the Patient entity set. This method will work, no matter which entity set you may start with. This is not the notation database books use for cardinalities, but in learning to create the designs, students do well with this method. Later in the semester the standard notations were discussed, for example, (1:N) or (M:N) or (1:1). Students were also given examples of multiple relationships. One example of this is the special case when one of the related entities participates as 1 and the others as N.

Once students had successfully created somewhat simple entity relationship diagrams, they used Microsoft Access to implement the designs. Access was used to help them to understand tables, fields, relationships and simple queries. Through the query by example screen in Access, students were also able to quickly see how entities (tables) were related.

The students were given some instruction in normalization, but because of time constraints, this was limited. Normalization is essential and a topic that should have a great deal of time invested in it. This issue will be addressed in the conclusion of this paper.

By using MS Access, this author saw that the understanding of basic database concepts was accelerated quite quickly after just a few weeks. Students who had never been given any instruction in database design were able to understand, at least in a basic way, the entity relationship diagram. None of the students had seen queries in the past and therefore the entire class was given a crash course in what a query is. Students were given query assignments using a number of databases were created by the Author, which were comprised of 3 to 5 tables. Students were able to perform queries using the query by example screen in Access. The Author then presented queries that are more complicated to the

students in an effort to point out that even a very simple database can provide a challenge to the programmer in answering a query of the database. For example, a very simple database, as seen in many database books, is the Parts-Supplier database( wikipedia.org). The design (schema) of the database is defined in figure 1.



**Figure 1.** Parts-Supplier database

In this schema, each part can be supplied by multiple suppliers and each supplier can supply multiple parts. At this point in the course, students were easily able to create the query:

1. List all parts supplied by supplier S1

However, students saw that even a simple database such as this, can yield challenging queries:

2. List all suppliers who do not supply the same parts as supplier Smith

The students became aware quite quickly that they had a long way to go in learning to successfully query a database.

### COURSE PLAN – MySQL

Once students had a general understanding of database design and basic SQL statements, they moved on to learning MySQL. This occurred at about week six. MySQL was chosen for a few reasons. These students would eventually be taking a senior project course where they would be using MySQL. Therefore, this would give them a head start for learning the application. Secondly, the software is free for download and the most popular database software available today. (See Figure 2 in the appendix for the survey of the most popular databases according to a 2018 StackOverflow survey)

<https://www.eversql.com/most-popular-databases-in-2018-according-to-stackoverflow-survey/>

Finally, the downloadable software is fully functional and thus allowed the student to create MySQL databases that simulate a multi-user environment, even though the software was installed on their personal computer.

The author created a number of handouts during the initial use of MySQL. Even though, by this point, all students had received some instruction in database design and query creation, many still had a long way to go in their comprehension. Among those handouts created were:

- Installing MySQL on your Personal Computer
- Create a new database (schema)
- Create a new table in the currently selected database and Insert a record into a table

- Enter multiple data into a table via workbench
- Export database from MySQL(structure and table)
- Import a database into MySQL
- Using workbench versus the command line

To get a jump-start on working with databases using MySQL, a few of the simple databases that were used in the MS Access section of the course were imported into MySQL. The real instruction into query design was then begun.

As those who teach database know, there is different syntax used to write queries depending on the platform one is using. For example, using the Parts-Supplier database described earlier, a SQL statement to add the primary key in MySQL is:

```
ALTER TABLE Supply  
ADD PRIMARY KEY (snum);
```

That same task to be executed in Oracle is:

```
ALTER TABLE Supply  
ADD CONSTRAINT PK_supply  
PRIMARY KEY (snum);
```

A great of SQL help can be found at the online site <https://www.w3schools.com/sql/>

This site contains examples of the syntax of SQL statements for various database platforms.

For a number of weeks the students used MySQL to learn both Data Manipulation Language (DML) and Data Definition Language (DDL) statements. Although MySQL does have the workbench, a unified visual tool for database developers, for the most part, students were required to write the proper SQL statement for the specific task assigned and run them from the command line.

An example of an early MySQL assignment is included in figure 3 of the Appendix.

The Author was beginning to see that the gap between students who had a previous database design class and those who had not was beginning to close. Although the design skills of the former students for the most part was better, the Author could not see an appreciable difference in their SQL skills. Exam scores supported this observation.

The students were gaining experience with MySQL and writing various SQL statements such as CREATE, UPDATE, ALTER, INSERT, DELETE , JOIN etc...

They were next assigned the first part of an individual database project in which they were to model a collection of data of their choice. They were to model their data by creating the proper entity relationship diagram along with a detailed explanation of the functionality of their database project. As stated earlier in this paper, approximately half of the students had only had about three weeks of formal database design work. Thus, the corrections to many of the student's ERDs was significant. It did require one on one help from the Author for a number of them. Interestingly, some of the students who had the design course also required help with their designs.

### **COURSE PLAN – ORACLE**

The final phase of the course was for the students to study the Oracle platform. Oracle was introduced at about week nine of this fourteen week course. It must be noted that Oracle was required for this course at this particular institution where the course was taught. Because the students had already studied the SQL Language using Access and MySQL, the transition to Oracle SQL was not exceeding difficult. The biggest issue for the students when beginning Oracle, was understanding how to connect to the server, how to log into Oracle and how to transfer files to and from their personal computers. Once again, the Author created various handouts to help students work with these processes. Among the handouts were:

- VPN instructions
- Setting up the Oracle connection
- Simple Linux commands
- Putty instructions

The Oracle SQL commands did not present as much a problem as expected. At this point, the students had already been exposed to the differences between MySQL SQL commands and Oracle SQL commands. Therefore, they were the commands, using the proper Oracle syntax.

This part of the course also involved studying PL/SQL, Procedural Language extensions to SQL, or Procedural Language/ Structured Query Language. SQL executes one single query at a time while PL/SQL executes an entire block of code. All students in this course had studied programming prior to this database class, so teaching tasks such as conditional statements and loops using PL/SQL was not as difficult as it could have been if the students had not been as prepared. A sample PL/SQL assignment working with 5 tables is contained in figure 4 in the appendix. Again, their biggest problem seemed to be transferring the results of their PL/SQL code to their PC. Students also were taught how to code triggers in both MySQL and Oracle.

### **FINAL STUDENT PROJECT**

As stated earlier in the paper, the students each created a database of their own choice using MySQL. A number of students actually created a database for a real user. The requirements of the project were:

- Create a database that was to contain at least 6 tables
- Create a detailed explanation of the outcomes of the database project
- Create a detailed ERD for the database
- Create all required DDL statements to create the database according to the ERD
- Create all required DML statements and the query results as defined in the required queries. (The Author defined required queries for each student according to what the student had specified as the outcomes of their database)

The students created a portfolio documenting all requirements. They also turned in the self-contained database file exported from MySQL.

### **CONCLUSION**

This Author has been teaching beginning database for a number of years, but this is the first time the Author taught it using three different platforms. The course at times was difficult from the side of the instructor. Nevertheless, the students reported that they were very excited to learn the various technologies. They also reported that creating their own individual database project increased their interest in the course. They felt that creating a project that they selected, gave them a better understanding of what is done in the real world. Some said they felt they would be more marketable in their eventual job search.

Student feedback was in the form of student surveys. Most reported that they enjoyed learning the three platforms. Of the students who had never had a database design course, about half stated that they felt they were playing catch up during the first part of the course. The final student database project was very popular with the students. They felt that it enabled them to relate the technology to a real problem.

The final grade was an indication of the success of the students in this database course. 75% of the students received an A for the course. Interestingly, those A grades were evenly distributed to both sets of students, those with prior database skills and those without. This was a surprising result for the Author. The remainder of the students were evenly distributed with B and C grades with 3 students failing the course.

Although the course ended quite well, some observations must be noted.

- A database implementation course with half the students having no database design work and the other half having an entire semester of database design is not recommended. It is not fair to either set of students. It

also puts the instructor in a difficult position. Which set of students does one teach to? In the Author's view, it is impossible to teach how to create a database without knowing how to design one.

- Because this course was a database implementation course, it seemed as though there was not enough time to adequately cover normalization, especially for those students who had not previously had a design course. This issue should be addressed in the design of this implementation course. Again, the design of the database is the core of the database.
- Having students each create a different database project was difficult. The Author actually ended up trouble shooting 35 different database designs (Baugh, 2015).
- An instructor, using three different database platforms must be “up” on various issues or errors that may arise in each platform. It is understood that working with the various SQL versions can be challenging.

Finally, although this Author was initially concerned about teaching this course, the experience ended up being very rewarding. Watching students who possessed no database knowledge grow to understanding database application in three different database platforms was gratifying as an instructor. The students became very confident, not only in their database skills, but in their ability to relate what was learned on one platform to another. Technologies change far too quickly in the computer field and as a computer instructor, what we hope to do is teach the students “how to learn”. Using this course structure helped the Author accomplish that goal.

#### REFERENCES

- Baugh, J., Davis, G., Kovacs, P., Scarpino, J., Wood, D. (2009) EMPLOYERS AND EDUCATORS WANT INFORMATION SYSTEMS GRADUATES TO BE ABLE TO COMMUNICATE, *Issues in Information Systems*, Vol X No 1. pp 198-207, 2009 (Also appears in The Proceedings of IACIS)
- Baugh, J. M. (2015). First Database Course--Keeping It All Organized. *Information Systems Education Journal*, 13(3), 24–33. Retrieved from <http://reddog.rmu.edu:2060/login.aspx?direct=true&db=eric&AN=EJ1137144&site=ehost-live&scope=site>
- Connolly, T. M., & Begg, C. E. (2006). A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education*, 17(1), 43-53. Retrieved from <https://reddog.rmu.edu/login?url=https://reddog.rmu.edu:3479/docview/200067319?accountid=28365>
- Denton, J. W., & A, G. P. (2003). Selection and use of MySQL in a database management course. *Journal of Information Systems Education*, 14(4), 401-407. Retrieved from <https://reddog.rmu.edu/login?url=https://reddog.rmu.edu:3479/docview/200134857?accountid=28365>
- Deperlioglu, O., Sarpkaya, Y., & Ergun, E. (2011). Development of a Relational Database for Learning Management Systems. *Turkish Online Journal Of Educational Technology - TOJET*, 10(4), 107-120.
- Dobesova, Z. (2016). Teaching database systems using a practical example. *Earth Science Informatics*, 9(2), 215-224. doi:<http://reddog.rmu.edu:2081/10.1007/s12145-015-0241-3>
- Hansen, Jorgen, (2012) Practical Elements in Danish Engineering Programs, Including the European Project Semester, *Industry and Higher Education*, v26 n4 p328-335 Aug 2012. 8 pp.
- Oswal, S. K. (2014). Access to digital library databases in higher education: Design problems and infrastructural gaps. *Work*, 48(3), 307–317. <https://reddog.rmu.edu:3345/10.3233/WOR-131791>  
[https://en.wikipedia.org/wiki/Suppliers\\_and\\_Parts\\_database](https://en.wikipedia.org/wiki/Suppliers_and_Parts_database)
- Seyed-Abbassi, B., King, R., & Wiseman, E. (2007). The Development of a Teaching Strategy for Implementing a Real-World Business Project into Database Courses. *Journal Of Information Systems Education*, 18(3), 337-343.
- Scott Stanford, M., Benson, L. C., Alluri, P., Martin, W. D., Klotz, L. E., Ogle, J. H., ... Schiff, S. (2013). Evaluating Student and Faculty Outcomes for a Real-World Capstone Project with Sustainability Considerations. *Journal of Professional Issues in Engineering Education & Practice*, 139(2), 123–133. [https://reddog.rmu.edu:3345/10.1061/\(ASCE\)EI.1943-5541.0000141](https://reddog.rmu.edu:3345/10.1061/(ASCE)EI.1943-5541.0000141)

Wynn, D., Jr. (2018). Dragon Air: A Database Design Case. *Information Systems Education Journal*, 16(2), 18–24.  
Retrieved from <http://reddog.rmu.edu:2060/login.aspx?direct=true&db=eric&AN=EJ1176558&site=ehost-live&scope=site>

APPENDIX

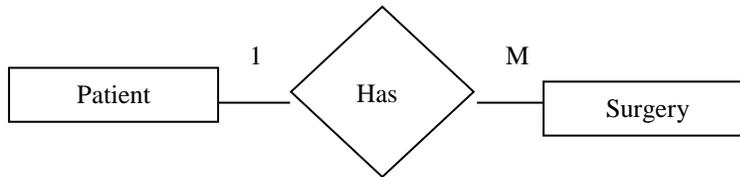


Figure 1. E-R Design (Patient has multiple surgery)

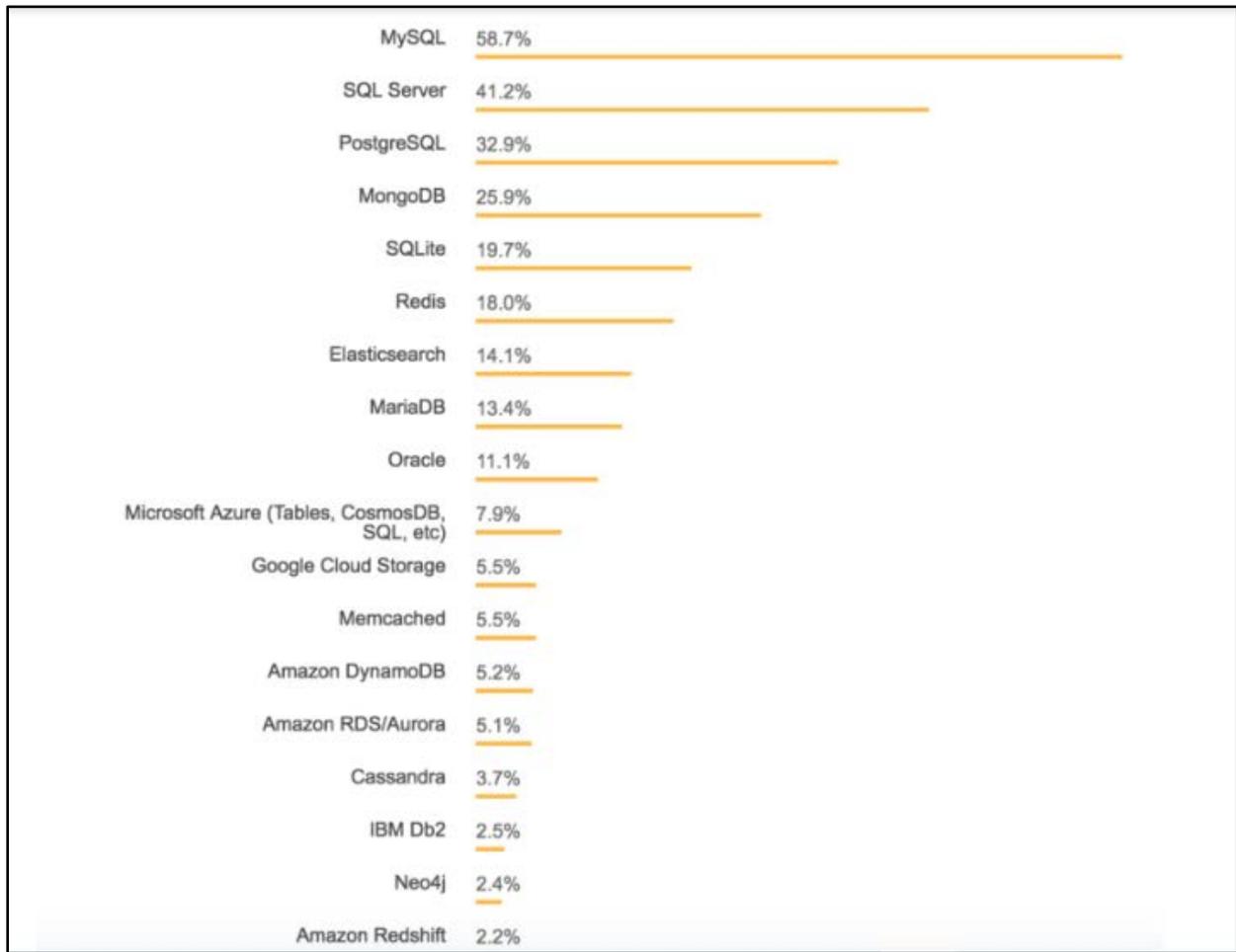


Figure 2 Popularity of Database software as defined in 2018 StackOverflow survey

SQL Assignment

1. You must use the proper SQL statements executed from the command prompt to create the following tables.

**Student:**

Stu_ID	Integer, primary key
fname	characters at most 25, not null
lname	characters at most 25, not null
age	Integer, not null

**Expense:**

Trans_num	Integer, primary key, auto increment
Stu_ID	integer, not null
Type	characters at most 255, not null
Amount	double, not null

2. Create the proper SQL statement to make the Stu\_ID in the Expense table a foreign key related to the Stu\_ID in the Student table
3. Create the proper SQL statements to add at least 2 records to the student table
4. Create the proper SQL statement to add at least 4 records to the expense table (related to the students in the student table)
5. Create the proper SQL statement to list all of the information in the student table (copy the output of the statement along with the SQL statement to your word document)
6. Create the proper SQL statement to list all of the information in the expense table (copy the output of the statement along with the SQL statement to your word document)
7. Create the proper SQL statement to list all student information along with the Trans\_num, Type and Amount from the expense table (copy the output of the statement along with the SQL statement to your word document)

**Figure 3. Sample beginning SQL Assignment**

PL/SQL Assignment

You are to use the Premiere database and write a PL/SQL program to create a customer invoice. On this invoice you will need the customer number, first name and last name, the number ordered and order date. You are also to display the sales rep for the invoice. Additionally, for all parts ordered, you will need the part number, part description, quoted price, line price and final amount due for the invoice. You are also to calculate 7% sales tax. (The line price is the quantity times the quoted price. Tax is 7% times all of the line prices for the invoice. Final amount due is the addition of all line prices and the tax)

The tables that are to be used in the assignment are:

- Customer
- Part
- Order\_line
- Sales\_rep
- Orders

**Figure 4. Sample PL/SQL Assignment**