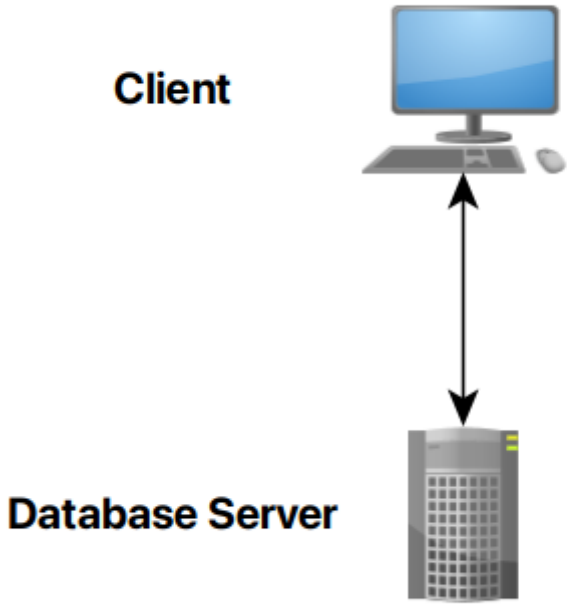
30041167—SQL Programming & Creative Writing

Task 1

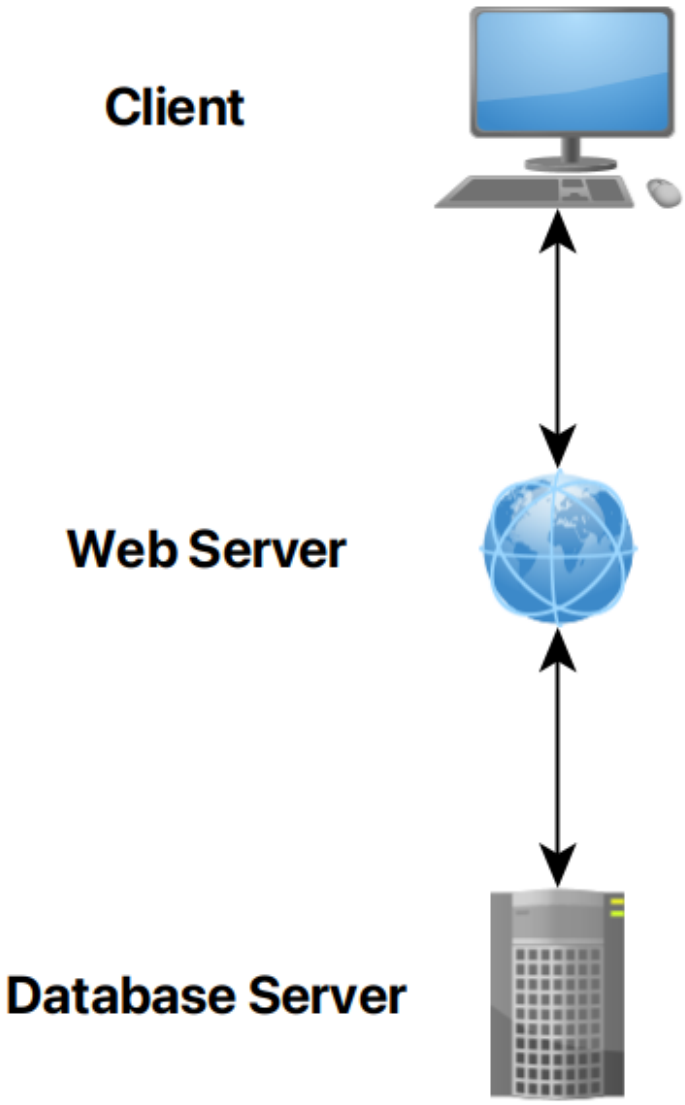
Answer the following:

- *Explain the major differences between two-tier and three-tier database architectures.*

Two-tier architectures consist of components distributed over two major systems: the client and server. The client side contains the user interfaces and application programs. The server side contains the database management systems. It handles the database's query and transaction routines—and can thus be also referred to as the query or transaction server (Elmasri & Navathe, 2016). It can be visualized as:



On the other hand, three-tier architectures consist of components distributed over the client and server sides, and with others acting as intermediaries between these two tiers. This middle tier consisting of intermediary components may contain application servers or web servers. The application servers for instance, could have programs that define business logic. Web servers, on their part, retrieve the results of queries passed on to the database servers and formats them into pages that the client/presentation layer can consume as the users demand (Elmasri & Navathe, 2016). Such an architecture can be visualized as:

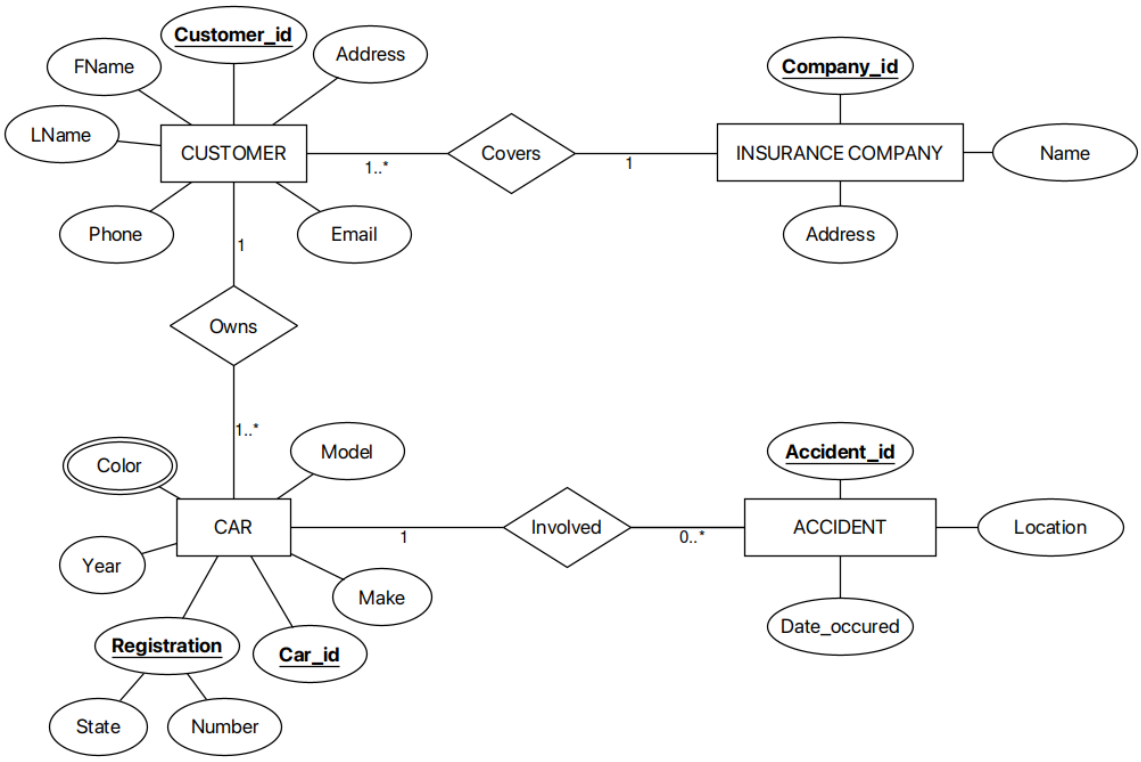


Reference

Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems*. Pearson.

Task 2

- Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



Task 3

Answer the following:

- *Discuss where databases are being used in everyday aspects of your life.*

The use of database systems has become an ubiquitous phenomenon in everyday life. Most of the computing activities that we conduct, for instance, have one form or the other of accessing a database somewhere—either locally or remotely. Take a case where one needs to transact using funds from their bank account. Or, one needs to travel and has to book a flight ticket and reserve a hotel accommodation. There are also some run-of-the-mill activities like shopping from the local store which necessitate the heavy use of database systems. Then, of course, we have online shopping which relies on multiple databases, distributed over several remote locations.

All these activities have traditionally required the passing of numeric or textual data back and forth from database systems. Yet with technological advancement, the amount and nature of data passed to and from database systems have evolved significantly. This has, in turn, steered database systems towards adapting modern technologies like the use of non-relational tables—an excellent example of which is NoSQL. Such developments have occurred hand in hand with the increasing demand for sophisticated aspects of data handling like big data. Moreover, with these features in place, services such as social media and online search have been able to offer services to their users which are more intuitive, relevant, and user friendly than ever before.

The amount of demand placed on database systems has increased drastically over the years too. Whereas in the past databases were mostly used to store textual and numeric data, nowadays they have to contend with an increased number of bytes being inserted into them and accessed at dizzying frequencies. The users' preference for consuming rich multimedia content, for instance, has meant that databases have to store vast amounts of data for one user alone. Thus, when a given service caters for millions of such users, it has to invest heavily in database systems that can handle colossal amounts of data that will then be queried incessantly during their uptimes.