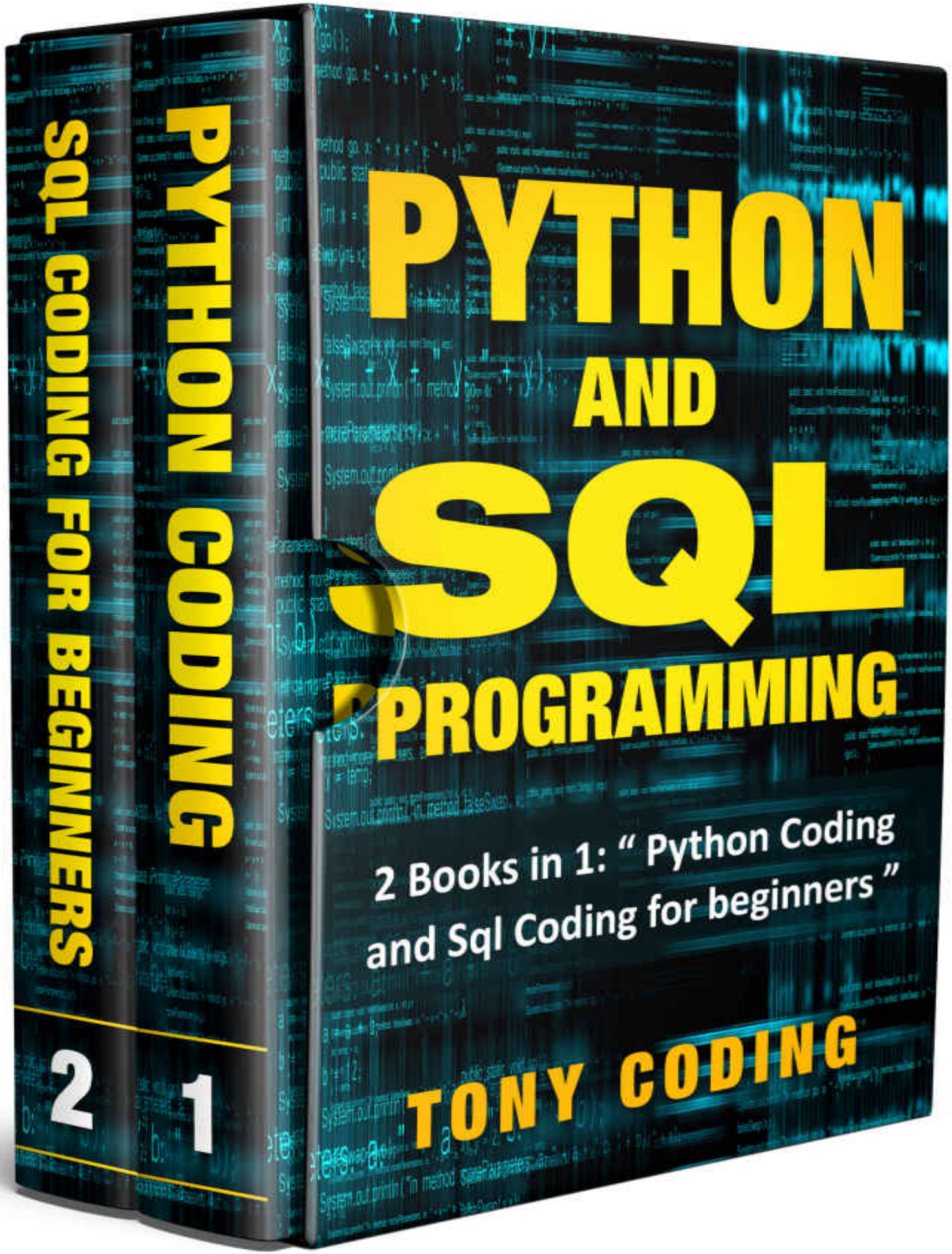

PYTHON CODING
1
SQL CODING FOR BEGINNERS
2
PYTHON
AND
SQL
PROGRAMMING
2 Books in 1: " Python Coding and Sql Coding for beginners "
TONY CODING

PYTHON AND SQL PROGRAMMING

2 Books in 1: “ Python Coding and Sql Coding for beginners “

[Tony Coding]

Legal & Disclaimer

The information contained in this book and its contents is not designed to replace or take the place of any form of medical or professional advice; and is not meant to replace the need for independent medical, financial, legal or other professional advice or services, as may be required. The content and information in this book has been provided for educational and entertainment purposes only.

The content and information contained in this book has been compiled from sources deemed reliable, and it is accurate to the best of the Author's knowledge, information and belief. However, the Author cannot guarantee its accuracy and validity and cannot be held liable for any errors and/or omissions. Further, changes are periodically made to this book as and when needed. Where appropriate and/or necessary, you must consult a professional (including but not limited to your doctor, attorney, financial advisor or such other professional advisor) before using any of the suggested remedies, techniques, or information in this book.

Upon using the contents and information contained in this book, you agree to hold harmless the Author from and against any damages, costs, and expenses, including any legal fees potentially resulting from the application of any of the information provided by this book. This disclaimer applies to

any loss, damages or injury caused by the use and application, whether directly or indirectly, of any advice or information presented, whether for breach of contract, tort, negligence, personal injury, criminal intent, or under any other cause of action.

You agree to accept all risks of using the information presented inside this book.

You agree that by continuing to read this book, where appropriate and/or necessary, you shall consult a professional (including but not limited to your doctor, attorney, or financial advisor or such other advisor as needed) before using any of the suggested remedies, techniques, or information in this book.

Table of Contents

Python Coding

An introduction to neural networks and a brief overview of the processes you need to know when programming computers and coding with python

[Tony Coding]

Legal & Disclaimer

The information contained in this book and its contents is not designed to replace or take the place of any form of medical or professional advice; and is not meant to replace the need for independent medical, financial, legal or other professional advice or services, as may be required. The content and information in this book has been provided for educational and entertainment purposes only.

The content and information contained in this book has been compiled from sources deemed reliable, and it is accurate to the best of the Author's knowledge, information and belief. However, the Author cannot guarantee its accuracy and validity and cannot be held liable for any errors and/or omissions. Further, changes are periodically made to this book as and when needed. Where appropriate and/or necessary, you must consult a professional (including but not limited to your doctor, attorney, financial advisor or such other professional advisor) before using any of the suggested remedies, techniques, or information in this book.

Upon using the contents and information contained in this book, you agree to hold harmless the Author from and against any damages, costs, and expenses, including any legal fees potentially resulting from the application of any of the information provided by this book. This disclaimer applies to any loss, damages or injury caused by the use and application, whether directly or indirectly, of any advice or information presented, whether for

breach of contract, tort, negligence, personal injury, criminal intent, or under any other cause of action.

You agree to accept all risks of using the information presented inside this book.

You agree that by continuing to read this book, where appropriate and/or necessary, you shall consult a professional (including but not limited to your doctor, attorney, or financial advisor or such other advisor as needed) before using any of the suggested remedies, techniques, or information in this book.

Introduction

Machine learning is permeating numerous aspects of our everyday lives, right from optimizing Netflix recommendations to Google searches. Machine learning has contributed to improving different facets of building mechanics in smart building space and the experiences of the occupant. You do not have to have a Ph.D. to understand the different facets and functions of machine learning. This section covers some facts about machine learning that are very basic and important to know.

Bifurcation of Machine Learning

Supervised and unsupervised machine learning are two techniques that programmers and scientists use to help machines learn. Smart buildings incorporate both types. Here is a simple example of how these types of machine learning look like: Let us assume that you want to teach a computer to recognize an ant. When you use a supervised approach, you will tell the computer that an ant is an insect that could either be small or big. You will also need to tell the computer that the ant could either be red or black. When you use an unsupervised approach, you will need to show the computer different animal groups and then tell the computer what an ant looks like and then show the computer another set of pictures and ask the computer to identify the ant until the computer learns the features specific to an ant.

Smart building spaces use both supervised and unsupervised machine learning techniques. The applications in these smart buildings allow the users to provide feedback to the building to improve the efficiency of the building.

Machines are not fully automatic

Machine learning helps computers automate, anticipate and evolve but that does not mean that they can take over the world. Machine learning uses

algorithms that human beings develop. Therefore, machine learning still needs human beings since they will need to set parameters and train the machine with different training datasets.

Machine learning helps a computer discover patterns that are not possible for human beings to see. The computer will then make an adjustment to the system. However, it is not good to identify and understand why those patterns exist. For instance, smart buildings human beings created smart buildings to ensure that the people inside the building help to improve the living conditions of the people. However, one cannot expect that a machine will learn to become more productive. A human must set up the definitions and rules that the building will need to follow.

Anyone can use machine learning

Writing a machine learning algorithm is very different from learning how to use that algorithm. After all, you do not need to learn how to program when you use an app on your phone. The best platforms always create an abstract of the program to present the users with an interface, which need minimal training to use. If you do know the basic concepts of machine learning, you are ready to go! Data scientists must edit or change the algorithms.

Machine learning has come of this age and is growing quickly. Buildings are using machine learning in different ways to make the existing infrastructure efficient and help to enhance the experience of the occupants residing in the building. Right from an energy usage standpoint, buildings are always learning and analyzing the needs of the occupants.

How does this affect us going forward? This advance in machine learning goes to say that most things will happen without the need for us to ask. Machine learning engineering could go beyond managing lighting and temperature. Machine learning implies that there will be some future state of

multiple layers and levels of automation adjusting based on the current activity.

Data Transformation is where the work lies

When you read through the different techniques of machine learning, you will probably assume that machine learning is mostly about selecting the right algorithm and tuning that algorithm to function accurately. The reality is prosaic – a large chunk of your time goes into cleansing the data and then transforming that data into raw features that will define the relationship between your data.

Revolution of Machine Learning has begun

During the 1980s there was a rapid development and advancement in computing power and computers. This gave rise to enormous amount of fear and excitement around artificial intelligence, computers and machine learning which could help the world solves a variety of ailments – right from household drudgery to diseases. As artificial intelligence and machine learning developed as formal fields of study, turning these ideas and hopes into reality was more difficult to achieve and artificial intelligence retreated into the world of theory and fantasy. However, in the last decade, the advances in data storage and computing have changed the game again. Machines are now able to work on tasks that once were difficult for them to learn.

Chapter 1:An introduction to python machine learning

Machine learning is both the application and the science of algorithms that can make some sense out of data. It is an incredibly exciting part of the computer sciences arena and, like it or not, it's here to stay. The world today is full of data and using algorithms that have the capacity to learn. This means that this data can be used to create knowledge. In the last few years, there have been quite a few open-source libraries developed, some of them incredibly powerful, and that means we are probably at the peak time to start truly understanding machine learning, to start learning how to use these algorithms to find data patterns and predict future events.

Do you remember when you got your first computer? For most people, the device was so foreign to them they couldn't even understand what they were supposed to do with it. No doubt, for many people, they still wanted one even if they had no idea what its true purpose was. Even today, there are numerous people who have found computers nothing more than a great device for playing games, binge-watching their favorite TV shows, or streaming their favorite music.

But you can do so many amazing things if you know how to tap into the true potential of these wonderful devices. Once a person knows what to do with

modern day machines, things begin to change in very big ways. We can easily take a task and go beyond the basics. When that happens, computers become far more than a glorified calculator that can decipher calculations and numbers in a fraction of a second. To get to that point there are a few things that you must understand.

Machines now do not need to have every detail of their functions automatically programmed. They can be programmed to learn a number of tasks and make the necessary adjustments to perform the functions that will allow them to work more efficiently.

Frankly, there are certain computer functions that many assume to be advanced technology but are merely things that can be done very quickly. For example, at the heart of every computer is a very complex calculator. When the computer performs an action we think is fascinating, it is merely the machine performing a number of mathematical equations to produce the results we desire.

You might want to stream your favorite movie to your computer. You click a few buttons and in a matter of seconds, scenes begin to play out in front of your eyes. Really, this function is nothing more than the computer running a bunch of basic math problems in the background, taking the sums and reconstructing them into a video on your screen.

This may seem like science fiction but the possibility is all too real, thanks to the creation of neural networks. In its simplest of terms, neural networks are a series of mathematical formulas called algorithms that identify relationships in a group of data. The network can accomplish this by mimicking the human brain and how it works.

These are complicated networks that are capable of adapting to constantly changing data so that it can achieve the best results possible without having

to redesign the criteria needed to get the optimum output.

To put it more simply, neural networks are the means of injecting flexibility into a computer system so that it processes data in a way that is similar to how the human brain works. Of course, computers are still going to be made in the same way as other machines but with each improvement, they are getting closer and closer to thinking machines rather than devices that are strictly following a static set of instructions.

Before we can fully understand neural networks, we have to get a firm grasp on what we mean when we talk about a machine that can learn. We are not talking about giving machines textbooks, homework, and exams so they can learn in the same way a student does. That would be ridiculous, but it helps to see just how a computer can mimic the human brain. So, let's look at how the human brain works first and make a comparison.

When a human being absorbs new information, they usually gain the information from something they're not familiar with. It could come in the form of a question or a statement of something new, or it could come as an experience with no verbal connection whatsoever. The information is picked up through the body's five senses and transmitted directly to the brain. The brain then reshuffles a number of its neural pathways (we call this thinking) so it can process the information and then when all the details related to the information is compared and analyzed in the brain, an answer or a conclusion is drawn and instructions are sent out to the rest of the body.

Since computers don't really think, they have to accomplish the same goal but in a different way. Information is inputted into the computer's programming, it is then processed, calculated, and analyzed based on a

number of preset algorithms, and then a conclusion, prediction, or answer is drawn and it comes out as output.

Let’s look at an example. Let’s say you want to figure out the answer to the problem 9 - 8. This is a basic math question that will require you to ‘think’ in order to get the right answer. While we will do this very quickly, we need to understand what is happening in our brain so we can see the similarity with computers.

When we receive information, our senses automatically send all the data relating to it to the brain. The brain is made up of billions of neurons that are all interconnected, creating miles upon miles of pathways where information can travel. What’s really neat about our brain is that these pathways are constantly shifting based on the data that is being transmitted. When new information is received, they will shift to create new pathways to transmit it to where it needs to go in the brain. Throughout this process, this shifting will continue until a solution is decided upon. Then instructions are sent throughout the body’s central nervous system to different parts of the body instructing them on the proper way to respond to the information received. The brain accomplishes all of this in fractions of a second.

In a neural network, the same thing happens. While these networks cannot perfectly mimic the inner workings of the brain, the process is very similar. The information is taken in and the neural network does all the work of consuming data, processing it, and coming up with a workable solution. These networks allow the computer to ‘learn’ by using algorithms.

Turning Data into Knowledge

We live in a modern world filled with technology. Every day, we hear that the planet is being stripped of its resources but there is one resource that

continues to grow, a resource that we have plenty of – data, both structured and unstructured. In the last 50 years of the 20th century, machine learning began to evolve. It was an AI subfield that used algorithms that could learn; algorithms that could derive knowledge from the data and use this knowledge to make predictions for the future. Where once a human being would have been needed to manually make the rules and build the models by analyzing vast amounts of data, machine learning does it much quicker and more efficiently. Machine learning allows us to capture the data, turn it into knowledge, and make improvements to how predictive modeling performed; the result of this is decisions driven by data.

So, you can see how important machine learning is in terms of computer sciences research, but do you realize just how often it features in everyday life? Think about the strong spam filters on your email, voice recognition on your computers and mobile devices, convenient text, decent search engines on the web, computerized Chess games and, in the future, safe self-drive cars. All of this is thanks to the advances made in machine learning.

Algorithms

No doubt, you've heard the term before. It is often associated with all sorts of technical mechanics but in recent years algorithms are being used in the development of automatic learning, the field that is leading us to advancements in artificial and computational intelligence. This is a method of analyzing data in a way that makes it possible for machines to analyze and process data. With this type of data, computers can work out and perform a number of tasks it could not originally do. They can understand different concepts, make choices, and predict possibilities for the future.

To do this, the algorithms have to be flexible enough to adapt and make adjustments when new data is presented. They are therefore able to give the

needed solution without having to create a specific code to solve a problem. Instead of programming a rigid code into the system, the relevant data becomes part of the algorithm which in turn, allows the machine to create its own reasoning based on the data provided.

How does this work?

This might sound a little confusing but we'll try to break this down into certain examples you can relate to. One of the 'learning' functions of machines is the ability to classify information. To do this, the input data can be a mix of all types of information. The algorithm needs to identify the different elements of the data and then group them into several different categories based on characteristics of similarities, differences, and other factors.

These characteristics can be any number of things ranging from identifying handwriting samples to the types of documents received. If this were code, the machine could only do one single function but because it is an algorithm which can be altered to fit a wide variety of things, the computer can receive this data and classify all sorts of groups that fit within the specific parameters of the circumstances.

This is how machines can change their functions to adapt to the situation at hand. Your email account can analyze all the emails you received, based on a pattern that you have followed, and it divides them into different groups. It can identify which emails are important and you should see right away, those that are spam and junk mail, and even sort out those that may pose a risk to your computer because it carries a virus or malware.

With these types of algorithms, machines can now learn by observing your habits and patterns and adjust their behavior accordingly. So, the very secret

to a successful and effective neural pathway depends a great deal on the algorithms your system uses.

Three Types of Machine Learning

Without algorithms, machines cannot learn. So, over the years many different ones have been developed. In this part of the chapter, we will look in detail at the three machine learning types. They are:

- Supervised learning
- Unsupervised learning
- Reinforcement Learning

We'll describe the basic differences between the three and explain the practicalities of applying the principles.

Supervised Learning – Predicting the Future

A supervised algorithm requires a detailed input of related data over a period of time. Once all the information is available to the computer, it is used to classify any new data relating to it. The computer then does a series of calculations, comparisons, and analysis before it makes a decision.

This type of algorithm requires an extensive amount of information to be programmed into the system so that the computer can make the right decision. That way, when it needs to solve a problem, it will attempt to determine which mathematical function it needs to use in order to find the correct solution. With the right series of algorithms already programmed into the system, the machine can sift through all types of data in order to find the solution to a wide variety of problems in the related category.

Supervised algorithms are referred that way because they require human input to ensure that the computer has the right data to process the information it receives.

Supervised learning has one basic goal – to learn a specified model from "labeled" data for training purposes so that predictions can be made about future data or unseen data. It is called "supervised" because the labels, or the output signals that we want, are known already.

Think about your email, about the spam filter in place. We could use supervised machine learning to train a model. The data would be labeled – emails already marked correctly as spam, those marked correctly as not spam – and the model would be trained to recognize to which group new emails belong. When you have supervised tasks such as this, using class labels, we call it a "classification task".

Predicting Class Labels Using Classification

So, classification is a subcategory with a goal of predicting under which class label new instances would be placed. This is based on previous data. The labels are discrete with values in no order that can be understood to be memberships of the instances. The example of spam filters is representative of a binary type of classification task – the algorithm will learn the rules so that it can work out the difference between potential classes – spam and not spam.

Although this is a binary classification task, the class labels are not required to be binary in nature. The learning algorithm will determine a predictive model that can then assign any of the labels from the dataset to a new instance that has no label. One example of this type of multiclass classification is the recognition of handwritten characters. We could have a

dataset that has the letters of the alphabet written in several different handwriting styles. If the user were to use an input device to give a new character, the predictive model could predict which letter of the alphabet it was with accuracy. However, the system could not recognize any digit from 0 to 9 because they would not be in the dataset used for training purposes.

Let's say that we have 30 pieces of sample data – 15 of them are labeled as a positive class with a plus sign (+) and 15 are labeled as a negative class with a minus sign (-). The dataset is called a two-dimensional dataset, meaning that each of the samples has two possible values, x1 and x2. Supervised learning algorithms could be used to learn the rule that a dashed line is used to represent the decision boundary and this rule is used to separate the classes and put any new data into one of the categories depending on the x1 and x2 values.

Continuous Outcome Prediction Using Regression

We know now that classification tasks are used for assigning unordered, categorical labels to instances. There is another type of supervised learning called regression analysis, and this is used for the prediction of continuous outcomes. With regression analysis, we have several explanatory or predictor variables, along with a variable for a continuous response. We can predict outcomes by finding relationships between the variables.

Let me give an example. We have a class of students and we want to try to predict their SAT scores. If we can find a link between how long each student studied and their scores, we could then use that data to learn models that can use the length of time studied to predict the score achieved by future students.

Reinforcement Learning – Solving Interactive Problems

Reinforcement Learning is another area of machine learning and the goal here is to develop an agent or system that can improve its performance. This improvement is based on how the agent interacts with its environment. Information that details the current environmental state tends to include a reward-signal, so this kind of machine learning can be considered a part of supervised learning. However, the feedback in Reinforcement Learning is not the right value or truth label; rather, it is an indication of how the reward function related to the action. By interacting with its environment, the agent learns several actions to maximize the reward. By using Reinforcement Learning, the agent will go through an approach of either deliberate planning or trial and error to get the reward.

Reinforcement learning is commonly used in video games where the computer must navigate and adjust its movements in order to win the game. A reward system is used so the computer knows and understands when it should make the right move, but there are also negative consequences whenever they make errors. This type of algorithm works best in situations where the computer has an obstacle that it must overcome like a rival in a game, or it could also be a self-driving car that needs to reach its destination. The entire focus of the computer is to accomplish certain tasks while navigating the unpredictable environment around it. With each mistake, the computer will readjust its moves in order to reduce the number of errors so it can achieve the desired result.

A Chess engine is a great example of this type of machine learning. The agent will decide on the moves it will make, dependent on the environment (the board) and the reward is defined as winning or losing when the game is over.

Reinforcement Learning has a number of subtypes but, in general, the agent will carry out several environmental interactions in an attempt to get the maximum reward. Each of the states is associated with a negative or a positive reward – the reward is defined by achieving a specific goal, in this case, losing or winning the game of Chess. For example, in a game of Chess, the outcome of a move is a state of that environment.

Let's go a bit deeper into this. Take the locations of each square on the Chessboard. Visiting them could be given a positive reward association – you take an opponent's piece or put his king or queen under threat, for example. Other locations have a negative reward association – losing one of your pieces by having your own king or queen put under threat in the next turn. Not every game turn will end with a piece being removed from the board so where does Reinforcement Learning come into this? Quite simply, it is about learning the series of steps that are needed by ensuring the maximum reward based on feedback – immediate and delayed.

This really is as far as we can go on Reinforcement Learning. We can only give you a basic overview because applications for this type of learning are way beyond the scope of this guide which is focusing on clustering, classification, and regression analysis.

Unsupervised Learning – Discovering Hidden Structures

With supervised learning, we already know the answer we want when the model is being trained. With Reinforcement Learning, we provide a reward measured by specific actions performed by the agent. With unsupervised learning, we are going one step further by using unlabeled data – data with unknown structure. By using these techniques, we can explore the data

structure to get meaningful information without needing to be guided by a reward or by a variable with a known outcome.

An unsupervised algorithm implies that the computer does not have all the information to make a decision. Maybe it has some of the data needed but one or two factors may be missing. This is kind of like the algebra problems you encountered in school. You may have two factors in the problem but you must solve the third on your own. A + b = c. If you know A but you have no idea what b is then you need to plug the information into an equation to solve the problem.

With unsupervised learning, this can be an extremely complex type of problem to solve. For this type of problem, you'll need an algorithm that recognizes various elements of a problem and can incorporate that into the equation. Another type of algorithm will look for any inconsistencies in the data and try to solve the problem by analyzing those.

Unsupervised algorithms clearly are much more complex than the supervised algorithms. While they may start with some data to solve a problem, they do not have all the information so they must be equipped with the tools to find those missing elements without having a human to provide all the pieces of the puzzle for them.

Use Clustering to Find Subgroups

Clustering is a technique of exploratory data analysis which lets us organize a great amount of data into subgroups or clusters. We do not have any upfront knowledge about the group memberships. Each of the clusters identified from the analysis will define a specific group of objects that are similar to a certain degree but are not quite so similar to objects that are in the other clusters. This is why you often hear clustering being termed as

“unsupervised classification”. Clustering is absolutely one of the best techniques we have for providing a structure to a set of information and determining meaningful relationships from the information. For example, think of marketers. They can use clustering to determine specific customer groups by their interests so that they can devise a targeted marketing program.

Dimensionality Reduction

This is another subfield of the unsupervised machine learning area. More often than not, high-dimensionality data is used. This means that each observation has several measurements. This presents a bit of a challenge where storage space is limited and affects the performance of the learning algorithms. Dimensionality reduction, in terms of unsupervised machine learning, is used quite often in feature preprocessing, to remove the noise from the data. This noise can cause degradation in the predictive performance of some algorithms. It is also commonly used to compress data so it fits into a smaller subspace while keeping the relevant information intact.

Occasionally, we can also use dimensionality reduction for visualizing data. For example, we could project a high-dimensionality feature set onto 1, 2, or 3D feature spaces – this would allow us to see that feature set through 2D or 3D histograms or scatterplots.

Semi-supervised learning

Semi-supervised learning is a blend of both supervised and reinforcement learning. The computer is given an incomplete set of data from which to work. Some of the data include specific examples of previous decisions

made with the available data while other data is missing completely. These algorithms work on solving a specific problem or performing very specific functions that will help them achieve their goals.

Of course, these are not the only algorithms that can be used in a computer program to help the machine learning. But, the general idea is the same. The algorithm must fit with the problem the computer needs to solve.

Machine Learning Systems – Building a Roadmap

Up to this point, we have looked at the basic machine learning concepts and the different machine learning types. Now we need to look at the rest of a machine learning system, the important parts that go with the algorithm. There are three important subsections to consider.

Preprocessing

Preprocessing is the act of getting our data into shape. Raw data is very rarely in the right format or shape required for the learning algorithm to work properly.

The images of the flowers are the raw data from which we will be looking for meaningful features. Those features could be color, intensity, hue, height, length of the flower, width of the flower, and so on. A lot of machine learning algorithms will also require the chosen features to be on identical scales for the best performance and this is achieved by transforming features in a range or by using standard distribution with “zero mean and unit variance”.

Some of the features selected may be correlated and that means, to a certain extent, they are redundant. In these cases, we can use the dimensionality reduction to compress those features down to a lower subspace. By doing

this, we don’t need as much storage space and the algorithm is significantly faster. In some cases, reduction can also make the predictive model performance more efficient, especially if the dataset has got multiple instances of noise (features that are not relevant) – you will often see this as having a low ratio of signal to noise.

To see if the algorithm is performing as it should on the training dataset while reacting positively to new data, we will need to divide the dataset randomly into two sets – training and test. The training dataset trains the model and optimizes performance, while the test set is used last to evaluate the finished model.

Training and Choosing Predictive Models

You will see throughout this guide that many of the algorithms are designed to solve specific problems. There is one important thing to remember – learning is not free. This concept can easily be related to another popular saying by Abraham Maslow in 1966 – “I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail”. Let’s put this into context. Every one of the classification algorithms has got biases built into it. No one classification model is superior to any other provided we make no assumptions regarding the task. In practical terms, it is vital that several algorithms are compared so that the best model can be selected and trained. However, before we can do this, we need to decide which metric we are using to measure the performance. One of the most common metrics is classification accuracy and this is defined as a set proportion of instances that have been classified correctly.

But, and this is a legitimate question, how will we know which of the models will perform the best on the final test and on real-world data if the test set is

used ONLY for the final evaluation and NOT on the model selection tests? To address this issue, we can use multiple techniques for cross-validation – the training dataset is split into two subsets: training and validation. This is done so that the model's generalization performance can be estimated. Lastly, although many software libraries offer learning algorithms, we should not take it for granted that the default parameters of these algorithms will fit the specific problem we are trying to solve. Instead, we will use techniques called hyperparameter optimization to fine-tune our model's performance and we'll see how this is done later. Basically, the hyperparameters can be thought of as parameters that cannot be learned from the dataset but instead, they represent dials that we can turn to improve the performance of the model.

Model Evaluation and Prediction of Unseen Data Instances

Once we have our model, the test dataset can be used to estimate how the model performs on unseen data. In this way, we can gain an estimation of the generalization error percentage. If we are happy with the way the model performs, we can use it for predicting future data. Be aware of one thing – it is vital that the parameters for the procedures we talked about, like dimensionality reduction and feature scaling, are obtained ONLY from the training dataset and are then applied to the test dataset to transform it. They are also applied to new samples of data. If we don't adhere to this, the test data may produce an overly-optimistic performance measurement.

Problems Of Machine Learning

Machine Learning is still evolving and still being used every day. As it is used more, the more we are going to be able to figure out where the

problems are and what needs to be done to fix them. Some of these problems are yet to be solved, but that does not mean that there will not be a solution later in the future.

One problem is that natural language processing and understanding language is still a problem when it comes to Machine Learning, even with the deepest networks. Because of the many different languages and dialects, it is hard for machines to decipher what is being said. But that does not stop them from trying. Programs such as Alexa, Siri and Cortana are constantly being updated so that they can better serve their users.

Images can be classified with Machine Learning, but it cannot figure out what is going on in the image. By understanding what is going on in an image, it cannot be further classified. But we are going to have the option to figure out where the influences in the image lie and how we would be able to recreate the work in the same style. This also leads to solving semantic segmentation. Machines should be taught how to identify what is going on in each pixel so that we can better understand what is going on in the image.

Deep reinforcement needs to be stabilized as well. If something is not broken, why attempt to fix it? Sometimes the old ways work the best, and there is no reason to be messing with the model that already works. With deep reinforcement being stabilized, this would be possible. This is then going to allow deep reinforcement to tackle harder problems that have yet to be touched by Machine Learning such as Montezuma's revenge.

On top of that, with stabilized reinforcement, learning the ability to control robotics would open up exponentially, and the possibilities would be endless. There is no telling how far robotics would go because they would be able to figure out the best way to receive their reward; in other words, they would be given the option to act like humans but have an incentive as to what they should do to get that reward.

GANs are generative adversarial networks that are going to work with a specific class of artificial intelligence. It is a set of algorithms that will be used with unsupervised learning which will then be implemented on a system of two neural networks working against each other in a zero-sum game framework. While they are great for those who use Machine Learning for games, they are highly stabilized and make gaming harder because they are known to frequently crash.

Training deep nets is an issue as well because it has been proven by the shattering of gradients paper that those that use Machine Learning do not understand yet how to train their nets properly. That also leads to the fact that no one quite understands what deep nets do in general. There have been those that have written papers on what deep nets are and why they are required, but then there have been those that have written papers on how deep nets are not needed, and they tend to make life more complicated. So, why are deep nets there? And do we actually need them?

Lastly, there has to be a way to get people to stop worrying so much about things such as Skynet and the economic impact. First of all, Skynet is a fictional company in the movie Terminator and is not going to happen. Before robots are able to make their own decisions if ever, there will no doubt be a failsafe in place that is going to ensure that the robots do not rise up against humans and try to kill them. All Skynet is, is a good plot for a movie franchise that millions of people love. Secondly, what is the economic impact of Machine Learning? With Machine Learning, companies are going to have the ability to figure out where they are going to make more money and that means that more money is going to be put into the economy!

Besides certain things in Machine Learning not being stabilized, Machine Learning does not have that many problems. It is hard to remember that Machine Learning is still in its infancy and is still being trained as

technology evolves, so we have to be patient and keep in mind that there are going to be more problems before they are to be fixed.

Problems with Machine Learning That Have Not Been Solved

Machine Learning is evolving as technology evolves, so it is highly likely that there are always going to be issues that will have to be dealt with before an answer is found. There are going to be ways around these issues, and even if there are not, there are going to be ways for you to analyze and get your data results despite these issues. Below, are a few issues that have yet to be solved by Machine Learning.

One, the variable event space has not been solved yet to solve this, we are going to need to be able to create a classification system that is going to respond in an efficient and meaningful way even if the event space is varying from the training data.

Another one is that there is no context understanding when it comes to the processing languages. Context analysis is a big deal for Machine Learning because it leaves a large hole open for things to be misunderstood. One possible solution is that the open AI team should come up with a solution that allows the agents to create their own rules instead of making it to where they need to find patterns.

Facial identification is quickly becoming a popular way for people to unlock their device because it is quick and a semi-secure way to keep your device from being hacked. However, it is by no means perfect because people with similar facial features will be able to get into each other’s phone. Not only that but if a female wears makeup and does not for the facial recognition picture, then she is not going to be able to get into her phone because she altered her appearance.

Automated learning is a big part of Machine Learning, and if a machine cannot automatically learn what it is supposed to, then it is not going to be useful to the company that is using it. Learning has to come from a variety of resources, and when a graph is being made based on the connected sense that is missing, then the automated learning is going to hit a snag and is going to have gaps in what it is able to do. For example, IBM's Watson has been doing an excellent job, but it still requires a system that is more geared toward automated general intelligence.

In some Machine Learning, the machine is going to need to make its own decisions based on its reasoning over a deviating test space; but this is not going to include the rule-based engines. To make the reasoning that is required, Machine Learning will need to have space for reasoning or debatable agents that will help with its reasoning.

Have you ever talked to Siri and she got confused if you are trying to talk to her or if you are talking to someone else? This is an issue that Machine Learning has with its efficient repose generation. In other words, Siri cannot create a contextual response when you are talking to her unless it is already programmed in or if she can look it up on the internet.

The three-way human rule is an issue with Machine Learning as well. Machines cannot always comprehend the knowledge that humanity has a three-way rule that takes an image, gives it a type, and then gives it a description. If Machine Learning could comprehend the three-way human rule, then classifying art, music, and others would be an easier task.

Another issue that Machine Learning has shown is its memory networks. Many people believe that working with technology means that you do not need a lot of space. However, this is far from true because to store all of the data that is collected, there needs to be a lot of space available.

Machines are not humans, so they are not going to be able to come up with the thoughts that a human is able to come up with. But machines can be taught as long as we do not push them past their limits, but we do not limit them to where all of the work still needs to be done by a human. Technology is going to have its hold-ups, but eventually, they can be solved. One reason they may not be solved yet is that the technology needed to solve it does not exist yet.

Problems That Can Be Solved

Machine Learning is helping out a lot of companies as well as individuals in their daily lives. In this section, you will see some points that were mentioned above as problems that have not been solved by Machine Learning just yet. Even though there are issues with that particular piece of Machine Learning, that does not mean that it does not work. Sometimes, these issues can be patched or fixed at a later date while the software is put out for users to use and discover what issues there may be with the software that the developers did not catch.

Machine Learning helps with spam detection. When you get into your inbox, there are always emails that automatically gets routed to your spam box because of keywords that are detected in other emails. Yet some spam is allowed through because these keywords have been found in your inbox. It is going to filter those keywords to your inbox on the slight chance that you are going to want the emails even though they are spam. Spam detection still has a long way to go, but it is getting better as more rules are made.

Have you ever wondered how the credit card companies are able to know when someone has used your card? It is because of the transaction record that they keep! Every month, the credit card company knows your usual stores and about how much is going to be spent. If they see any activity outside of your usual area or outside of how much you usually spend, they

are going to say that the card was stolen and report it as such. This can cause problems later on if you decide to go on a trip or if you spend over what you normally spend.

It is not perfect, but Machine Learning is learning how to recognize zip codes written on envelopes so that they are able to sort the envelopes out into geographical locations and that way the mail is sorted faster and is delivered faster. There are still some issues with this as machines are not always able to read a person's handwriting since everyone writes in a different manner.

As mentioned, Siri or Cortana is able to understand what you are saying. But, what if other machines were also capable of doing this? Thanks to Machine Learning, this is now becoming a reality. The biggest issue is that everyone has different dialects and speaks a different language. Not only that, but new words are being added to the dictionary almost daily. But, even with there being issues in machine’s understanding of what is being said to them, this does not stop them from trying to fulfill the request that is made.

When you upload a picture on Facebook, there is an algorithm that looks at the facial features of those in the picture and asks you if you would like to tag that person. This allows the person to see the picture where they were tagged and allows them to decide if they want it on their profile or not. The same algorithm is used when a user decides to use the facial recognition software on their phone or computer to unlock their device.

Everyone knows when they log into websites such as Amazon, there is a list of products that are recommended to you because of your past purchases or because of things that you have looked at. There is a model that is built-in to the code of the website that allows it to figure out ways to target you specifically as an individual so that you are given a more personalized shopping experience. In doing this, Amazon and other websites that use the

same algorithm are hoping that they are going to be able to keep you coming back to spend money with them. This algorithm can also be seen in Facebook because it allows you to log into other websites so that it can provide you with ads that you may want to see instead of generic ads that are not going to catch your attention.

The medical field has to advance with technology, or else it falls behind. Because of Machine Learning, the medical field is able to take the symptoms that a patient is feeling. This allows them to plug the symptoms into their computer and get a list of diagnosis. This is not going to be perfect, but it is going to point the healthcare provider in the right direction so that they are able to figure out what is wrong with the patient.

When it comes to trading stock, it is vital to know when you are going to trade or hold on to your stock so can make more money. Machine Learning has come up with an algorithm that looks at the financial information and analyzes it so that stockbrokers are able to know when they should trade and when they should let their stocks go. This is helping many companies save money, so they are not buying useless stocks or so that they are not holding onto a stock that is about to lose all of its value.

Data entry by humans is going to have issues such as duplication and inaccurate data. But, with Machine Learning, algorithms work with predictive modeling algorithms to improve these issues. They learn how to perform time-intensive documentation and other data entry tasks. This opens up more free time for those with the proper knowledge to be able to work on the more intensive problem-solving tasks because they do not need to deal with the menial data entry jobs that machines are able to do.

Machine Learning is used by most of the companies that are in business today. It may not be used in a way that is automatically visible to the general public or even their employees, but it is being used even in the simple way

of analyzing data. This aids them in making better decisions in gaining money rather than losing money.

Where And How Machine Learning Is Used

Machine learning is already in widespread use in many different industries. It's used for anti-lock braking, autopilot in airplanes, search engines, product recommendations, maps, and driving directions, spam screeners, language translation, personal digital assistants, weather maps, and more. Any organization that has a lot of data and is looking for better ways to understand and use that data can benefit from this technology.

Several companies have used machine learning to build some of the most lucrative businesses on a getting-to-know-you-better platform. Google, Facebook, Apple, LinkedIn, and Twitter are all using machine learning to make sure you have access to relevant data. Some companies are even pushing the envelope of privacy by analyzing your search history or personal or professional connections. By learning more about you they can personalize their service for you. That's why your Facebook newsfeed is completely unique to you. It's also why two people googling the exact same search term may get completely different results.

Any time you're on a website and you see something like "recommended for you," you're probably benefiting from machine learning. Amazon may examine your previous purchases and then use machine learning to recommend other products you might like. Netflix tries to figure out your taste in movies to suggest other shows you might enjoy. YouTube tracks what you watch and recommends related videos. Each of these online services uses machine learning to try to turn you into a long-term customer or user.

Other organizations are using machine learning for automatic translation. YouTube might use it to transcribe video and generate captions. Some sites use a very similar machine learning technology to translate captions into different languages. Through natural language processing (NLP), powered by machine learning, some services translate captions into a synthesized voice that sounds similar to a human being.

With machine learning, computers tap the power of artificial intelligence to process massive data sets and extract valuable information from that data. They can find connections in the data that humans could never detect on their own and may never think to look for. Sometimes we can understand how these intelligent computers made the connection and other times we cannot.

One of the most interesting aspects of machine learning is that it doesn't replicate human learning; it's a completely different way to find connections, make decisions and gain greater understanding. So if you're planning to use machine learning in your organization, you have to start thinking about how machines learn, so you can start to collect data that will help the machine learn to perform the task you need it to perform.

Machine learning didn't start to really gather steam until companies had large datasets, and that's not a coincidence; data is the fuel that drives machine learning. That's why some of the first companies that benefited from machine learning were the ones that had access to massive datasets. Data plays a crucial role in what the machine learns, how well it learns, and how fast it learns. Usually, the more data you have, and the better that data is, the sooner the machine will start delivering useful insights.

So before you start your machine learning project you should think about your data. Is it high quality? Do you have enough for the computer to learn something new? Is the dataset broad enough to accurately represent whatever

you're asking the machine to examine? The broader the view, the more likely it is to find something interesting. You don't want your program looking through a keyhole. If you plan to use this technology, think about some strategies for collecting diverse, high-quality datasets.

Chapter 2: The process of neural networks

We can talk about neural network architecture now that you know more about deep learning and its applications. Neural networks are very important for machines. A properly programmed neural network will help the machine think like a person and process information in that way. These networks are made out of layers. These layers can process many kinds of information. Each layer is made out of neurons that all work together to solve problems. Since neural networks help machines think like humans, it makes sense that machines with neural networks are capable of learning. Pattern recognition and data classification are some of the things that these machines are used for.

When neurons change and adjust, the network can learn, similar to human beings.

A person who does not dabble with AI, somebody that you may talk to on the street, could be shocked to learn that they have encountered a lot of artificial intelligence and other kinds of machine learning. Millions of dollars are spent by some of the most popular companies in order to research that

will improve their business. Some of these companies are Apple, Google, Facebook, Amazon, and IBM.

Our day to day lives might be impacted by this research already, even though you might not know it. With internet searches, for example, you will be shown options on websites and searches that match the keywords you typed in. Machine learning is important for this as it is the main thing that allows your browser to filter through the millions of possible recommendations.

The same principle is used in Netflix recommendations or spam filters that help filter through your emails. In the medical industry, this is used for the classification of medication and has an important role in the Human Genome Project where it goes through the countless combinations of patterns in DNA that can tell you something about your family history, risk factors, or health prospects.

These systems are applicable in most industries as they are highly adaptable and sophisticated. This is made possible through algorithms that guide the computer through many learning processes. With the correct algorithms, you can teach a system to detect abnormal behaviors that can happen within a pattern. This helps the system learn how to predict possible outcomes that may occur in a wide variety of situations.

An artificial neural network is something that contains algorithms of different kinds, data receptors and a plethora of other elements. Ever since they were introduced during the 1950s, artificial neural networks have been a remedy when it comes to the future of science. They are patterned similarly to the human brain. They allow the machine to learn during the training phase of programming. This knowledge is then used as a foundation for solutions that would be applied to problems in the future.

Historical Background

Neural networks have been a thing since before computers. The problem was that people were not proficient enough for utilizing them. This is why the recent advances made in the field of neural networks are so important. Any developments in computer technologies help the research of neural networks. A craze started and people started being enthusiastic over the field. However, most attempts were fruitless, as no advancements that could help to improve the efficiency and accuracy of our machines. The enthusiasm started to decrease. However, some of the researchers remained steadfast and continued their studies. They worked hard in order to develop what we have now, a technology model that is accepted by most people in the industry.

The first artificial neural network was made by Warren McColloch and Walter Pitts in the year 1943. This was called the McCulloch-Pitts neurons. The network was not used to perform or automate complex tasks as the duo did not have the technology needed for the further development of the network.

What Are They and How Do They Work?

Terms like artificial intelligence, machine learning, and deep learning are all terms that refer to processes that are happening in and to the neural network. People might be skeptical when it comes to the way machines learn. However, we assure you that it really means that they are trained like human minds are.

A computational distributed model is made up of simple parallel processors with a plethora of tiny connections is a good way to think about these networks. The human brain is made from many, many neurons that are interconnected via synapses which allow them to make analysis and computations in our cerebral cortex. Learning is achieved with the change in the connections in our brains allow us to acquire new skills and learn new skills so that we can solve difficult problems.

A good way to think of these networks is to think of many simple parallel processors integrated with hundreds (or thousands) of tiny connections that make up a computational distributed model. In the human brain, there are millions of neurons all interconnected by synapses that allow them to make computations and analysis in the cerebral cortex. As these connections are made, learning is achieved allowing the person to acquire new skills so they can accomplish complex problems.

Hundreds of homogenous processing units that are interconnected through links are elements of a neural network. The unique configurations of connections and simplicity are what make this design truly beautiful. Data goes into the network through an input layer and goes to the output layer. In the meantime, the data is processed through the many layers in between until the problem is computed and a final solution is carried out.

Only a few units that transfer information were the gist of the structure of the simple neural networks in their earlier days. Today, however, a network could be made up of millions of different units that are intertwined and work together in order to emulate the process of learning. More modern networks are able to solve very difficult and complex problems in many ways.

Why use Neural Networks?

Large volumes of data are dedicated to making improvements to the industry by the industry itself. There are many variables in these datasets that can make it difficult for humans to find patterns in that appear in the datasets themselves. Via neural networks, we can recognize these patterns more easily. Without them, computers could find it to be a difficult task to identify trends in the dataset. By taking the time to train a neural network, an engineer can feed the network huge datasets in order to turn it into an expert in a selected area. With this trained network, the engineer can predict the output for any possible input. This also allows the engineer to be able to

answer some questions about the future. Neural networks have many advantages and here are some of them:

- A neural network can adapt to new tasks and learn how to do new things because of it using supervised machine learning
- The network can be used to report back any information that was fed to it during the learning stage
- Machines with neural network architecture work far faster and provide more accurate results. This is because of the ability to compute data in parallel.
- Neural networks can be fixed fairly easily. While performance will be affected if the network is damaged, the network will still remember some of the properties.

The McCulloch-Pitts Neuron - What is It?

There are a lot of similarities between the human brain and artificial neural networks. This makes sense due to the fact that these networks were made to emulate how the human brain This much is easy to understand. The following are some of the similarities between them:

- Millions of artificial neurons make them up and each one of them can compute problems and solve them
- Each neuron has many different connections
- They are both non-linear and parallel
- They can learn through the change in the connection between units
- They adapt to new knowledge when they find an error instead of penalizing themselves

- Based on the data that they never came across before, they can produce outputs

These all describe how the neural network, as a whole, works. While the two are very similar on the surface, we should look into how the smallest of the units work. The McCulloch-Pitts neuron is the smallest part of the network.

In the brain, the basic unit of processing is a neuron. In the artificial neural network, this unit is the main part of processing any type of information and calculations of any kind. It is made from the following three elements:

- The weight and synaptic efficacy of connections, as well as the connections themselves
- An agent that summarizes and processes input signals and outputs linear combinations
- A system that limits the neuron's output

The 1940s was the first time that the MCP neuron was introduced. This neuron got its name from the logician Walter Pitts and the neuroscientist Warren S. McCulloch. They tried to understand what happens inside of the human brain when we try to produce and understand complex patterns and replicate them through the connections between the many basic cells.

The design of old systems was very simple. The input was quantified from zero to one and the output was limited to the same domain. Every input was either inhibitory or excitatory. This simplicity made the design limited the learning capabilities of the machine.

While sometimes simplicity can be great, it also has its downfalls. It costs the computational ability to comprehend complex topics.

The MCP neuron was supposed to sum up all of the inputs. The neuron takes all of the positives and negatives that appear and compiles them, adding plus

one if an input is positive and taking one away if the input is negative.

Neural Networks versus Conventional Computers

Neural networks and computers do not apply the same kind of solution to every problem. The latter usually use algorithms in order to find solutions. Conventional computers also have another method of solving a problem. If you taught it the steps that it should follow it should do good by them in order to find a solution. What this tells us is that humans and computers can solve similar problems. Where computers shine is solving problems that people can’t.

As we have said numerous times, the human brain and neural networks work in a similar manner: through a network of interconnected neurons. These networks usually work in parallel for the best efficiency. An engineer can teach a network to complete a task by giving it an example of how it should approach the solution. This means that selecting the data you feed to the system is very important. If you are not careful when selecting the data you might make it harder for the system to learn the process. The networks are unpredictable as, due to the data they are fed, it might learn to solve problems that the engineer didn’t foresee.

Computers apply some cognitive approaches when solving problems, as well. If the engineer gives the computer proper instructions the computer can act on them and solve problems. Using a high-level programming language, it takes only a few steps for the engineer to provide the instructions to the computer. The computer, later, takes these instructions and translates it into a language that the computer can understand. This process allows the engineer to predict how the computer will go about solving certain problems. If the computer reports a problem while processing, the engineer can conclude that there should be an error in the software or hardware.

Conventional computers work in tandem with neural networks. Arithmetic calculations are a kind of task that is best solved by conventional algorithmic computers. On the other hand, other, more complex tasks, can be solved by neural networks most efficiently. Most tasks are best solved by combining the two approaches so that the machine can work at peak efficiency,

Types of Neural Networks

Fully connected neural network

A network layer is the most basic type of neural network architecture. It is composed out of three layers of neurons which are interconnected. The input layer is the first layer. It is connected to a hidden layer of neurons which then proceed to the output layer. The input layer is where the engineer gives data to the system. The nodes that connect the hidden layer and the input layer dictate how the input layer views the data it is given. What kind of data outputted depends on the weights and connections between the secret layer and output layer.

This kind of architecture is simple, but still interesting because the hidden layers can represent data in many ways. The weights mentioned before are nodes that connect layers. They also determine when these layers need to be activated. An engineer can modify these weights and change the way the layers interact. He can do this to ensure the way the hidden layer shows data is conducted the correct way.

It is fairly easy to tell a multilayer and single-layer architecture apart. When it comes to single-layer architecture, neurons are connected at the nodes. This means that the processing power of the network is maximized due to all of the layer being interconnected. When it comes to multilayer architecture there are more layers to the system. Here neurons are not interconnected, but the layers are.

Perceptrons

The term perception was coined in 1960 by Frank Rosenblatt. This happened during a time where neural network architecture developed greatly. Perceptions represent a kind of McCulloch and Pitts model which is assigned a pre-processing and fixed weight. This makes the system easier to use when it comes to recognizing the patterns, resembling the function in human beings. This network can be used in other processes too.

Feed-forward Networks

There are several names for this kind of network. Bottom-up or top-down are both alternative names for feed-forward networks. This means that signals and data will flow in only one direction, from the input point to the output point. This network does not have a feedback loop. This means that output gotten in one instance will not affect output that comes in another layer. This is one of the simplest networks to make as input is directly associated with output. This network is often used in recognizing patterns.

Convolutional neural networks

Fully connected neural networks are what convolutional neural networks bears the most similarity to. Many layers made from many neurons are what makes up this system. Each neuron is assigned with a weight. This is due to the training data that was used during the teaching phase. When input is given to a neuron they will make a dot product, which is followed by a non-linearity. You might wonder what the difference between a fully connected neural network and a convolutional network is. A CNN views each input in the original dataset to be an image which an engineer can encode properties of the system into. This reduces the number of instances in the original dataset and it makes it easier for the network to use the forward function. These types of neural networks are used by most deep learning programs.

Feedback networks

Feedback networks are specific due to the movement of the signals. They flow in both directions and induce a loop into the network. These networks are extremely difficult to make, but are very powerful. The network will function in a state of absolute equilibrium until you create a change. This state of constant change will continue until the system equalizes again. If an engineer feeds a new dataset to the system, it will try to find a new point of equilibrium. This is the reason why feedback networks are recurrent and interactive. Below we will discuss recurrent neural networks.

Recurrent neural networks

Recurrent neural networks are special because the information always loops. The network will consider the input and asses what it has learned, once it decides. RNN usually has short-term memory. In combination with Long Short-term memory, it gains long-term memory. We will further discuss this bellow.

It is difficult to explain RNN without using an example. Let’s say that the network you are using is a regular feed-forward neural network. Let’s say that you input the word “brain”. The system will separate the word into characters and go over them one by one, always forgetting the previous characters. This means that this model cannot be used to predict which character is next unless it has already been gone over. Due to its internal memory, an RNN remembers previous characters and predicts the next ones. While producing outputs, it copies them and puts them back into the network itself. This means that RNNs will produce, on top of the present information, five immediate past information. Due to all of this, the following inputs are all a part of an RNN:

- The Present Data

- The Recent Data

As always, the dataset selection is very important. The dataset used to teach the model will affect how the system uses sequences while predicting upcoming characters in a text. This places RNN in front of most algorithms when it comes to the functions they can perform. Unlike a feed-forward network, RNN applies weight to both the previous and current input data and shifts the weights that have already been assigned, while a feed-forward network assigns weights to neurons in each layer in order to produce and output.

Generative adversarial network

A GAN, also known as a generative adversarial network, is made up of two networks that have been pitted against each other. Due to this, we call it an adversarial network. GAN can be used by most engineers in a system, because it can learn to mimic any distribution or dataset. GAN can be used to build something that is unique to you in many domains. It can simultaneously process pictures, prose, speech, etc. You are bound to be impressed by the output of these networks.

A generator and a discriminator are parts of this network. The discriminator evaluates instances made by the generator, while the generator creates the data instances themselves. The discriminator's job is to identify if a new instance is from a premade training dataset or not.

Training the Neural network

Training your neural network is one of the most important parts of making a neural network. There are a few methods to do this, however, only one method has the most positive results. Error backpropagation, also known as the error propagation algorithm, symmetrically adjusts the weights on

connections and neurons. This means that if the system makes a mistake it will learn from it and come closer to the correct solution every time.

This kind of training has two stages: stage 1, also known as the forward propagation, and stage 2, also known as back propagation.

In stage 1, a calculation of all of the activated neurons in all of the layers is performed. During this stage, there is no change to synaptic connection weights. What this means is that the default values will be used in the first iteration of the system. During phase 2, however, we are given an actual answer from the network and we can compare the output to the expected output in order to determine the error rate.

The error rate is then taken into account and it is returned to one of the synaptic connections. Modifying the weights then decreases the difference between the expected value and the value we got. This process happens on and on until the error margin is decreased to the point where it can’t be decreased anymore.

- Forward Propagation: in forward propagation, the first input is the initial data that is then transferred to the hidden layers where it is processed until an output is produced. The activation functions, the depth of the data, and the width of the data all depend on the basic architecture of the network. Depth tells us how many hidden layers exist within the system. The width tells us the number of neurons in each layer and the activation functions instruct the system on exactly what to do.

- Backward Propagation: It allows for the weight of the connections to be adjusted via a supervised learning algorithm. This is done in order to reduce the difference between the solution we got and the expected solution.

Neural networks are a very interesting field of study and keeps getting more and more intricate, now using machine learning. It has a huge amount of

potential to aid the creation of future developments in computer science.

- They are adept at solving problems whose solutions require a certain degree of error
- They can use experience from solving previous problems and use it to solve problems it encounters for the first time.
- Their implementation is a piece of cake as the definitions for duplication, neurons, and creating connections are easy to understand
- It completes operations fairly quickly as every neuron operates on only the value it received as input
- Stable outputs directly relate to the input values
- Before producing a result, they can take all of the inputs into accounts

Neural networks still have a few drawbacks, even with all of those advantages. Some of them are:

- It has a certain similarity to black boxes. You can determine what happened, but there is no way to determine why a certain result was produced
- The memory cannot be described or localized in the network itself
- They can only be used by computers with compatible hardware as they have unique computer needs
- Producing proper calculations can be very time consuming, however, as the training techniques are extensive and can take a while to execute.
- The only method of solving problems is algorithms and you have to give them the correct one for the problem
- The accuracy of the output values can vary

- A large number of examples is needed for a good learning process that can produce solutions to be made

Neural networks are completely capable of independent decision making based on the number of inputs and variables. Because of this, they can create an unlimited number of recurring iterations to solve problems without human interference. When we see these networks in action, you'll find a numeric vector that represents the various types of input data. These vectors could be anything from pixels, audio and/or video signals, or just plain words. These vectors can be adjusted via a series of functions producing an output result.

At first glance, you might think that there is not that much to say about neural networks. However, when you look into it a bit more, you start to see the intricacies behind it. The same system can be used to handle the most basic of problems and the most complex alike. The only thing that changes is the number of weights that are placed on each value.

We have already pointed out that these algorithms are an integral part of machine learning. They are used to sift through all sorts of data, pull out any information that could be useful to reach the targeted goal and bring you to the closest possible solution to a problem. All of this is done without having to write a specific code for the computer to actually solve the problem because of something called a 'neural network'.

But what exactly is a neural network? Let's go back and take another look at the human brain so we can get a better understanding of this new technology.

The brain holds billions of tiny little neurons that are poised to receive data and process it. These neurons are all interconnected through a complex web

with each neuron holding a certain amount of information. These neurons send signals to each other as they process the data they receive.

In a computer, a neural network is created artificially. The architecture has been around for decades but the technology has advanced enough just recently for it to be implemented into any usable and functional form.

In an artificial neural network (ANN) these neurons are mimicked by thousands (sometimes millions) of tiny little processing units, all linked together. Each one of these artificial neurons has a purpose, which is determined by the configuration or the topology of the network.

There are several layers of these neurons and each layer has its own specific purpose. There is the input layer, where all the data flows into the network, the output layer where the solution is produced, and there could be numerous hidden layers where much of the processing work is done.

Training and Selection of a Predictive Model

As we will see in the next sections, various machine learning algorithms have been developed with the aim of solving different problems. An important element that can be drawn from David Wolpert's well-known No Free Lunch Theorems is that there is no "free" learning. For example, each classification algorithm has its inherent flaws, and no classification model can claim absolute superiority if we have no information on the task at hand. In practice, it is therefore essential to compare at least a certain group of different algorithms, so as to train them and then select the model that offers the best performance. But before we can compare different models, we need to decide which metrics to use to measure performance. A commonly used metric is the accuracy of the classification, which is defined as the proportion between the instances correctly classified.

A legitimate question arises: how can we understand which model performs better on the final test dataset and on the real data if we do not use this test dataset for choosing the model, but we keep it for the final evaluation of the model itself? In order to solve the problem inherent in this question, various cross-validation techniques can be used, in which the training dataset is further subdivided into subsets of validation training, in order to estimate the performance of generalization of the model. Finally, we cannot even expect that the standard parameters of the various learning algorithms provided by software libraries are optimal for our specific problem. Therefore, we will frequently use hyper-parameter optimization techniques, which will help us optimize the performance of the model. Intuitively, we can consider these hyper-parameters as parameters that are not learned from the data but represent the "knobs" of the model, on which we can intervene to improve their performance, as we will see with greater clarity in the next sections when we will put them to work on examples effectively.

Evaluation of Models and Forecasting Of Data Instances Never Seen Before

It is important to note that the parameters of the procedures we have just discussed (reduction of the scale and size of the features) can only be obtained from the training dataset and that these same parameters are then reapplied to transform the test dataset and also each new sample some data. Otherwise, the performance measured on the test data could be overly optimistic.

Chapter 3:Learn coding with python

When Guido van Rossum developed the first Python language compiler in the late 1980s, little did he know that the language will be more famous than popular languages in machine learning and Artificial Intelligence. The fact is —in the last couple of years; Python language has emerged as a solution for most machine learning problems.

Python language is beginner-friendly, yet very powerful. It is no wonder that Python language is finding its applications in some of the most popular systems such as Google, Pinterest, Mozilla, Survey Monkey, Slideshare, YouTube, and Reddit as a core developer language. Also, Python's syntax is extremely simple to understand and follow if you're a beginner or advanced programmer.

If you're an advanced developer of C, C++, Java or Perl, you'll find programming in Python to be extremely simple. If you're an experienced developer, you can accomplish great things with Python. Besides developing games, data analysis, and machine learning, Python language can also be used to code general AI systems and development of GUIs.

This chapter explores how you can engineer ML systems in Python language. Let's get started.

Getting started with Python

Obviously, to kick start developing machine learning systems in Python, you need to install it on your computer and set the programming environment. If you're a novice Python programmer, learning basics of Python installation and setting up the environment will go a long way in promoting your bottom line.

The installation process

The process of downloading and installing the Python language interpreter is pretty simple. If you're using the latest Linux distribution—whether it's Ubuntu, Fedora or Mint—then you'll find the most recent version of Python already installed. All you have to do is to update your system. If you're using a Debian-based Linux distribution, follow these steps to update your system:

· Launch the Terminal app and type the following command at the command prompt:

su apt-get update

· Type your root password and press the Enter key

· Wait for the update process to be completed.

If you're using a Redhat-based Linux distribution such as Fedora, follow these steps to update your system:

· Launch the Terminal app and type the following command at the command prompt:

su apt-get update

· Type your root password and press the Enter key

· Wait for the update process to be completed.

On the other hand, if you're using other OS's apart from Linux, you have to download and install Python yourself. Also, if you're using an older version of Linux that has no Python, then you have to install it manually.

Follow the steps outlined below to install Python on Linux distributions:

· Launch the Terminal app (Ensure that you're connected to the internet)

· Type "su" at the command prompt and press the enter key

· Type your root password and hit the enter key

· If you're using Debian-based Linux distribution such as Ubuntu, then type: "apt-get install python" at the command prompt and press the enter key

· On the other hand, if you're using the Red Hat/ RHEL / CentOS Linux distributions such as Fedora, then type: "yum install python" at the command prompt and hit the Enter key.

· Wait for the installation to complete.

· Update the system by typing: "su apt-get update" if you're using Debian-based Linux distributions or "su yum update" if you're a Redhat/ RHEL / CentOS Linux distribution user.

What about Windows OS?

Before you download and install Python, decide on the version of Python language that you would want to install. As a rule of thumb, always go for the latest version. As of writing this book, the latest version was 3.6.2. Here are steps that can help you install Python on Windows OS:

· Go to www.python.org and download the current version of Python. Select the appropriate version depending on the nature of your OS (32 bit, or 64 bit).

· Open the Python file that you've just downloaded

· Click on the "Accept the default settings" from the on-screen instructions and wait for the installation process to complete.

If you're a Mac OS X or Sierra user, then you'll find Python 2.7 already portedto the OS. Therefore, you don't have to install or configure anything if you want to begin using Python. However, if you wish to install the latest version of Python, you need to use the Homebrew. Here are steps that can help install Python on your Mac OS:

· Open your Terminal or any of your favorite OSX terminal emulator

· Type the following command at the command prompt:“/user/bin/ruby-e"$(curl-fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)".

· Now proceed to install Python language interpreter by typing the following command at the command prompt: “brew install python.”

· Wait for the installation process to complete.

Now that you’ve installed Python, what next?

It’s now time to begin developing your ML systems. But not that fast! You should decide on what text editor you’ll use. You can opt to select your best editors to help you code and execute your program. Some of the most popular Text Editors are Emacs, Geany, Komodo Edit and Sublime Text.

But since we all know the pitfalls of the Text Editors—such as running the code manually from the Python Shell—I won’t advise you to use them. Instead, use the Python IDLE (Integrated Development Environment). I have been using it ever since without problems. However, you can choose an IDE that suits you.

Python IDLE has the following features:

· Syntax highlighting

· Auto-completion of code statements

· Smart indentation

· Integrated debugger with the stepping, persistent breakpoints, and call stack visibility features.

Launching Python

To get started, you have to understand how to launch the Python app. You can launch Python from the Terminal or use the desktop environment for starting the IDLE app. Simply launch the Terminal and type: “idle” at the command prompt. Now that you’ve launched the Python, it’s now time to begin coding.

Let’s now create our first program in Python. Follow these steps to write your first Python program:

· Open the Python IDLE.

· Write the Python language statements (instructions) in the IDLE window.

· Run the program

That’s it! Simple. Isn’t it?

Now, here’s a quick way to see the programming process in action...Proceed and copy/paste the following code into your Python IDLE window.

print (“Hello World! This is my first Machine Learning program”)

Run the program. What do you see as the output?

Well, the phrase “Hello World! This is my first Machine Learning program” appears.

Congratulations! You’ve just written your first Python code. I know you’re now excited to begin coding ML systems. Don’t worry so much about the meaning of statements. If you are a machine learning novice, mastering some Python programming concepts will help you understand how to design ML applications.

Next up, let’s dive in together and get to the basics of Python programming.

An Overview of Python

Now that you have executed your first Python program, what else do you need to know? Well, it's now time to understand the vital components of any Python code including its structure. All Python programs have the following structure:

```
import sys

def main ():

main ()

{

Program statements

}
```

As you can see from this program structure, all Python codes should always start with the keyword "import." Now, what are we importing? Python language is object-oriented. Therefore, it has components of all the object-oriented programming languages. One such property is inheritance or in simple terms; code reuse. The ability to inherit features of codes in Python allows programmers to reuse pieces of codes that had been written elsewhere.

Technically speaking, the import statement tells the Python interpreter to declare classes that have already been used in other Python packages without referring to their full package names. For instance, the statement: "import sys" informs the interpreter to include all the system libraries such as print whenever the Python program is starting.

What does the statement "def main ( ):" mean?

Whenever a Python program is loaded and executed, the computer's memory —the Random Access Memory—contains the objects with function definitions. The function definitions provide the programmers with the

capabilities of instructing the control unit to place the function object into the appropriate section of the computer's memory. In other words, it's like instructing the control unit to check the main memory and initialize the program that needs to be executed.

The function objects in the memory can be specified using names. That's where the statement "def main ( ):" comes in. It simply tells the control unit to start executing the Python code statements that are placed immediately after the statement "def main ( ):"

For example, the Python code below creates a function object and assigns it the name "main":

```
def main ():
  if len (sys.argv) == 10:
    name = sys.argv [2]
  else:
   name = "Introduction to Machine Learning."
  print ("Hello"), name
```

In the above code, the Python interpreter will run all the function statements in the Python file by placing the set of functions objects in the memory and linking each of them with the namespace. This will happen when the program is initialized with the import statement.

But more fundamentally, "What are the different elements of Python code?" Well, all Python programs have the following components:

· Documenting the program. Any statement in the program (except the first) that starts with "#" is treated as a command line or comment line and

will be ignored during execution. This will allow you to comment on sections of the code for proper documentation.

· Keywords. The keywords are instructions that the interpreter recognizes and understands. For instance, the word “print” in the earlier program is a keyword. In Python, there are two main types of keywords: the functions and the control keywords. Functions are simple verbs such as print that tell the interpreter what to do while the control keywords control the flow of execution.

· Modules. Python program is shipped with a large list of modules that increase its functionality. The modules will help you to organize your code in a manner that’s easy to debug and control the code.

· Program statements. The program statements are sentences or instructions that tell the control unit to perform a given operation. Unlike most programming languages, the Python statements don’t need a semicolon at the end.

· Whitespace. The white spaces are a collective name that given to tabs, the spaces, and newlines/carriage returns. The Python language is strict on where the white space should be placed in the code.

· Escape sequences. The Escape sequences are special characters that are used for output. For instance: the sequence “\n” in the program tells Python to output on a new line.

Python variables

There’s so much that goes on in the main memory of the computer whenever you run a program. Understanding the concept of variables and data types will help you to write efficient programs.

A program is simply a sequence of instructions (statements) that directs your computer to perform a particular task. For instance, the previous program printed the phrase “Hello World! This is my first program” on the screen when it was executed. But before you could see the output on the screen some data had to be kept in computer’s memory.

The use of data applies to all programming languages—Python included—therefore, understanding the mechanisms of data management in the computer’s memory is the first step towards developing robust, efficient and powerful applications.

A variable can be conceived as a temporary storage location in the computer’s main memory and specifically the Random Access Memory. This temporary storage location is what will hold the data that you would like to use in the program. In other words, the variable location of memory that holds data whenever your program is executing. So, whenever you define a variable, you’ll actually be reserving a temporary storage location in the computer’s memory.

All the variables that you define must have names and an equivalent data type— a sort of classification of the variable that specifies the type of value the variable should hold. The data types help to specify what sort of mathematical, relational or even logical operations that you can apply to the variable without causing an error. Ideally, when you assign variables to data types, you can begin to store numbers, characters, and even constants in the computer’s main memory.

Since Python language is an oriented programming language, it is not “statically typed.” This means that the interpreter regards every variable as an object. Therefore, you have to declare the variables before using them in your program. So, how can you declare variables in Python?

Names or identifiers usually declare Python variables. Just like any other programming languages that you have so far learned, the conventions for naming the variables must strictly be adhered to. Below are some naming conventions that you should follow when declaring variables:

· All variable names should always begin with a letter (A to Z and a to z) or an underscore. For instance, “age” is a valid variable name while “–age” isn’t a valid variable name.

· Any variable name you declare cannot start with a number. For instance, 9age is not a valid variable name.

· Special symbols shouldn’t be used when declaring variable names. For instance, @name isn’t allowed as a variable name.

· The maximum number of characters to use for your variable name shouldn’t exceed 255.

To reserve a temporary memory location in the name of a variable, you don’t have to use the explicit declaration like other programming languages. If you’ve had experience in other programming languages such as Pascal or C, I am sure you know that declaring a variable explicitly before assigning any value is a must.

In Python, the declaration of variables usually occurs automatically the moment you assign a value to it. For instance, the statement;

age=10

Automatically reserves a temporary storage location in memory space called “age” and assigns 10 to it.

It is also possible to assign a single value to several variables simultaneously. For instance, the statement below reserves temporary memory spaces for 2 variables namely: age and count and assigns the value 30:

age, count=30

Python language has different categories of data types that are used to define the storage methods and mathematical operations. Below are examples data types in Python language:

- Numbers
- String
- List
- Tuple
- Dictionary

a) Numbers

The Number data types stores numeric values. The number objects will automatically be initialized whenever you assign a specific value to the variable. For instance, the code illustrated below creates 2 variable objects (age and count) and assigns them the values 10 and 30 respectively:

age = 10

count= 30

If you want to delete reference to the Number object, you'll use the word "del" followed by the variable name that you wish to delete. Consider the code below that deletes two variables: age and count that have already been declared and used."

del age, count

Python language supports four different categories of number types. These are:

- int. when used in a declaration it refers to signed integers. These include those whole numbers that range from 8 bits to 32 bits.

· long. These are long integers. They can be represented either in octal and hexadecimal numbering notation.

· float. These are floating real point values. They may range from 8 bits to 64 bits long.

· complex. These are complex numbers.

b) Strings

Strings are stored as consecutive sets of characters in the computer's memory locations. Python language allows you to use either pair of single or double quotes when defining the strings. Other subsets of string variable types can be specified using the slice operator ([ ] and the [:]) with the indexes that range from 0 at the beginning of the string. The plus (+) operator performs string concatenation (joining of two or more strings) while the asterisk (*) operator performs string repetition.

File Handling in Python

A file is simply a named location on disk that has stored related information. Files are used to store data on a permanent basis on a non-volatile storage device such as a hard disk. You are aware that the main memory—and specifically the RAM—can't be used to store data on a permanent basis. Therefore, files must always be stored on a permanent storage device.

When a file is created on a secondary storage device, the following file information is automatically generated by the respective OS:

· Name of the file

· Its location

· File size

· File attributes such as read/write read only and archive

You can use this mode if you are dealing with non-text data such as images.

The File Object Attributes

Once you've opened a file, the file object is created automatically with the following information that is related to the file.

The Python code below illustrates an example of how you can work with file object attributes:

```
myfile = open("Readme.txt", "wb")

print "The name of the file is: ", myfile.name

print "Is the file closed or not? : ", myfile.closed

print "The opening mode of the file is: ", myfile.mode

print "The Softspace flag of the file is: ," myfile.softspace
```

The above code will produce the following output:

```
The name of the file is: Readme.txt

Is the file closed or not? : False

The opening mode  of the file is: wb

The Softspace flag of the file is:
```

The close() Method

The close() method of a file object is used to flush any unwritten information and close the file object. When the file has been closed, no more writing can take place. You should note that Python automatically closes the file when the reference object of the file is reassigned to another file. That is why it is a good programming practice to use the close() method to close a file.

Below is the syntax for closing a file:

```
fileObject.close()
```

Below is an example code of how you can close a file in Python:

```
myfile = open("Readme.txt", "wb")

print "The name of the file is: ", Readme.name

myfile.close()
```

Reading and Writing Files

The file object gives a set of access methods that make the process of reading and writing to files much easier. In particular, the two methods that you'll need are read() and write() methods.

The write method

Here's the syntax for the write() method:

```
fileObject.write(string)
```

Below is an example of a Python code that uses the write() method:

```
myfile = open("Readme.txt", "wb")

myfile.write( "I have learned several aspects of Python Programming in this book.\n And the book is great for Advanced Programmers! \n")

myfile.close()
```

The above method creates a Readme.txt file and writes the given content in that file. When you finally open the file, here will be its contents:

I have learned several aspects of Python Programming in this book. And the book is great for Advanced Programmers!

The read method

The read () method is used to read a string from a file that you have opened. It is crucial to note that Python strings can also have binary data, apart from the text data. Below is its syntax:

fileObject.read([count])

In the above example, the passed parameter is the number of bytes that are supposed to be read from the opened file. The read method begins reading from the start of the file, and if the count is missing, then it will attempt to read as much information as possible until the end of the file.

Here is an example:

```
myfile = open("Readme.txt", "r+")

mystr = myfile.read(10);

print "Read string is: ", mystr

# Check current position

position = myfile.tell();

print "Current file position : ", position

# Reposition the pointer at the start of the string once again

position = myfile.seek(0, 0);

mystr = myfile.read(10);

print "Again read Mystring is: ", mystr

# Close the opened file

myfile.close()
```

Looking at the above code, you'll realize that you have to create a loop for you to read the entire string. One easy way to read the text file and parse each line the statement readlines on with the file object. The Python's readlines reads every data in the in the text file. Here's an example:

```
myfile = open("Readme.txt", "r")

lines = myfile.readlines()
```

f.close()a

Now the above code reads the entire file

What about reading the file line by line?

You can use the while statement to help you read file line by line. Below is an example:

```
myfile = open("Readme.txt")
line = myfile.readline()
while line:
    print line
    line = myfile.readline()
myfile.close()
```

Up to this point, we've only described Python language in terms of structure, some elementary data structures, and file handling. I believe this is necessary to get you started with engineering machine learning system. However, what we've covered isn't exhaustive. Truth be told—there's so much to learn in Python. For you to master machine learning, you've to dig deeper into Python programming.

Have you been using the classification as a type of machine learning? Probably yes, even if you did not know about it. Example: The email system has the ability to automatically detect spam. This means that the system will analyze all incoming messages and mark them as spam or non-spam.

Often, you, as an end user, have the option to manually tag messages as spam, to improve the ability to detect spam. This is a form of machine learning where the system takes the examples of two types of messages:

spam and so-called ham (the typical term for "non-spam email") and using these cases automatically classify incoming mails fetched.

What is a classification? Using the examples from the same domain of the problem belonging to the different classes of the model train or the "generate rules" which can be applied to (previously unknown) examples.

Dataset Iris is a classic collection of data from the 1930s; This is one of the first examples of modern statistical classifications. These measurements allow us to identify the different types of flower.

Today, the species are identified through DNA, but in the 30s the role of DNA in genetics had not yet been recorded. Four characteristics were selected for each plant sepal length (length of cup slip) sepal width (width of cup slip) petal length, and petal width. There are three classes that identify the plant: Iris setosa, Iris versicolor, and Iris virginica.

Formulation of the problem

This dataset has four characteristics. In addition, each plant species was recorded, as well as the value of class characteristics. The problem we want to solve is: Given these examples, can we anticipate a new type of flower in the field based on measurements?

This is the problem of classification or supervised learning, where based on the selected data, we can "generate rules" that can later be applied to other cases. Examples for readers who do not study botany are: filtering unwanted email, intrusion detection in computer systems and networks, detection of fraud with credit cards, etc.

Data Visualization will present a kind of triangle, circle type, and virginica type of mark x.

The model has already discussed a simple model that achieves 94% accuracy on the entire data set. The data we use to define what would be the threshold was then used to estimate the model.

What I really want to do is to assess the ability of generalization model. In other words, we should measure the performance of the algorithm in cases where classified information, which is not trained, is used.

Transmitting device stringent evaluation and use the "delayed" (Casually, Held-out) data is one way to do this.

However, the accuracy of the test data is lower! While this may surprise an inexperienced person who is engaged in machine learning, it’s expected to be lower by veterans. Generally, the accuracy of testing is lower than the accuracy of training. Using the previous examples you should be able to plot a graph of this data. The graph will show the boundary decisions.

Consider what would happen if the decision to limit some of the cases near the border were not there during the training? It is easy to conclude that the boundaries move slightly to the right or left.

NOTES: In this case, the difference between the accuracy of the measured data for training and testing is not great. When using a complex model, it is possible to get 100% accuracy in training and very low accuracy testing! In other words, the accuracy of the training set is a too optimistic assessment of how good your algorithm is. Experiments always measured and reported the accuracy of testing and accuracy on the set of examples that are not being used for the training!

A possible problem with the hold-out validation is that we are only using half of the data used for training. However, if you use too much data for training, assessment error testing is done on a very small number of

examples. Ideally, we would use all the data for the training and all the data for testing, but it was impossible.

A good approximation of the impossible ideals is a method called cross-validation. The simplest form of cross-validation is Leave-one-out cross-validation.

When using cross-checking, each example was tested on a model trained without taking into account that data. Therefore, cross-validation is a reliable estimate of the possibilities of generalization model. The main problem with the previous method of validation is a need for training of a large number (the number grows to the size of the set).

Instead, let’s look at the so-called v-fold validation. If, for example, using 5-fold cross-validation, the data is divided into five parts, of which in each iteration 4 parts are used for training and one for testing.

The number of parts in the initial set of components depends on the size of the event, the time required for the training model, and so on. When generating fold data, it is very important to be balanced.

You may also start to notice that there are many different companies, from startups to more established firms, that are working with machine learning because they love what it is able to do to help their business grow. There are so many options when it comes to working with machine learning, but some of the ones that you may use the most often are going to include:

- Statistical research: machine learning is a big part of IT now. You will find that machine learning will help you to go through a lot of complexity when looking through large data patterns. Some of the options that will use statistical research include search engines, credit cards, and filtering spam messages.

- Big data analysis: many companies need to be able to get through a lot of data in a short amount of time. They use this data to recognize how their customers spend money and even to make decisions and predictions about the future. This used to take a long time to have someone sit through and look at the data, but now machine learning can do the process faster and much more efficiently. Options like election campaigns, medical fields, and retail stores have used machine learning for this purpose.
- Finances: some finance companies have also used machine learning. Stock trading online has seen a rise in the use of machine learning to help make efficient and safe decisions and so much more.

As we have mentioned above, these are just three of the ways that you are able to apply the principles of machine learning in order to get the results that you want to aid in your business or even to help you create a brand new program that works the way that you want. As technology begins to progress, even more, you will find that new applications and ideas for how this should work are going to grow as well.

Data modeling is an important aspect of Data Science. It is one of the most rewarding processes that receive the most attention among learners of Data Science. However, things aren't the same as they might look because there is so much to it rather than applying a function to a given class of package.

The biggest part of Data Science is assessing a model to make sure that it is strong and reliable. In addition, Data Science modeling is highly associated with building information feature set. It involves different processes which make sure that the data at hand is harnessed in the best way.

Robust Data Model

Robust data models are important in creating the production. First, they must have better performance depending on different metrics. Usually, a single

metric can mislead the way a model performs because there are many aspects in the classification problems.

Sensitivity analysis describes another important aspect of Data Science modeling. This is something that is important for testing a model to make sure it is strong. Sensitivity refers to a condition which the output of a model is meant to change considerably if the input changes slightly. This is very undesirable because it must be checked since the robust model is stable.

Lastly, interpretability is an essential aspect even though it is not always possible. This is usually related to how easy one can interpret the results of a model. But most modern models resemble black boxes. This makes it hard for one to interpret them. Besides that, it is better to go for an interpretable model because you might need to defend the output from others.

How Featurization Is Achieved

For a model to work best, it must require information that has a rich set of features. The latter is developed in different ways. Whichever the case, cleaning the data is a must. This calls for fixing issues with the data points, filling missing values where it is possible and in some situations removing noisy elements.

Before the variables are used in a model, you must perform normalization on them. This is achieved using a linear transformation on making sure that the variable values rotate around a given range. Usually, normalization is enough for one to turn variables into features once they are cleaned.

Binning is another process which facilitates featurization. It involves building nominal variables which can further be broken down into different binary features applied in a data model.

Lastly, some reduction methods are important in building a feature set. This involves building a linear combination of features that display the same

information in fewer dimensions.

Important Considerations

Besides the basic attributes of Data Science modeling, there are other important things that a Data Scientist must know to create something valuable. Things such as in-depth testing using specialized sampling, sensitivity analysis, and different aspects of the model performance to improve a given performance aspect belong to Data Science modeling.

To help you understand just how deep the field of deep learning goes and just how much it has changed everyone’s lives already, I will dedicate this section to showing you specific examples of deep learning and how it is used in its myriad of applications.

Keep in mind, this is not meant to advertise any kind of product or service, but to show you that deep learning is far more common than many people think and that it is not a field pertaining to the higher levels of each industry, but one that belongs to all of us to some extent.

So, without further ado, let’s dive in:

Image Curation on Yelp

Although Yelp may not be as popular as it used to be, it still has a very important role to play in how people experience the new places in their living areas (or the different locations they visit as tourists, for example).

At first, Yelp may seem like anything but a tech company - but they are using actual machine learning to make sure their users come back to the site because it provides them with actual, helpful information.

More specifically, Yelp has developed a machine learning system capable of classifying, categorizing, and labeling images submitted by users - but more importantly, this system helps Yelp do this in a genuinely efficient way. This

is extremely important for the company, given the huge amounts of image data they receive every day.

Pinterest Content

Who knew searching for wedding ideas on Pinterest is fueled by machine learning?

Pinterest's main purpose is that of curating existing content - so it makes all the sense in the world that they have invested in machine learning to make this process faster and more accurate for their users.

The system developed by Pinterest is capable of moderating spam and helping users find content that is more relevant to their own interests, their styles, and their searches.

Facebook's Chatbots

By this point, it is more than likely that you have stumbled upon at least one chatbot in Facebook Messenger.

These apparently simplistic chatbots are, in fact, a form of primordial artificial intelligence. Sure, Skynet is not typing from the other end of the communication box, but even so, chatbots are a fascinating sub-field of artificial intelligence - one that is developing quite steadily.

Facebook Messenger allows any developer to create and submit their own chatbots. This is incredibly helpful for a variety of companies that emphasize their customer service and retention, because these chatbots can be used for this precise purpose. Sometimes, Messenger chatbots are so well-built that you may not even realize that you are talking to a, "robot."

Aside from chatbots, Facebook invests a lot in developing AI tools capable of reading images to visually impaired people, tools capable of filtering out spam and bad content, and so on.

In some ways, a company that might not seem to have a lot to do with technological innovation is pushing the boundaries of one of the most exciting fields of the tech world: artificial intelligence.

Google’s Dreamy Machines

Google is one of the companies constantly investing in artificial intelligence (often, with amazing results). Not only have they developed translation systems based on machine learning, but pretty much every area of their activity is somewhat related to artificial intelligence too.

Don’t be fooled - Google has its hands in much more than search engines. In recent years, they have invested a lot in a very wide range of industries, including medical devices, anti-aging tech, and, of course, neural networks.

The DeepMind network is, by far, one of the most impressive neural network research projects ran by Google. This network has been dubbed as the “machine that dreams” when images recreated by it were released to the public, opening everyone’s eyes to how artificial intelligence, “perceives” the world.

Baidu Voice Search

Since China is the leading country in artificial intelligence research, it only makes sense that their leading search company, Baidu, is heavily invested in the realm of artificial intelligence too.

One of the most notable examples here is their voice search system which is already capable of mimicking human speech in a way that makes it undistinguishable form, well, **actual** human speech.

IBM’s Watson

We couldn’t have missed Watson from this list, mostly because this is one of the first impressively successful artificial intelligence endeavors in history.

Most people know IBM's Watson from its participation in **Jeopardy!**, but the supercomputer built by the super tech giant IBM can do **much** more than just compete in televised shows.

In fact, Watson has proved to be very useful to hospitals, helping them propose better treatment in some cancer cases. Given the paramount importance of this type of activity in medicine, it can be said that Watson helps to save actual lives - which is a truly great example of how AI can serve mankind.

Salesforce's Smart CRM

Salesforce is one of the leading tech companies, specifically in the field of sales and marketing, where the tool helps businesses maximize their sales potential and close more deals with their customers.

Salesforce is based on a machine learning tool that can predict leads and assigns scores for each of them. For sales people and marketing pros, this is a true gold mine because it makes the entire sale process smoother, more fluent, and, overall, more efficient.

Where Do You Come From, Where Do You Go, Deep Learning?

Clearly, deep learning advances are quite fascinating. Many take them for granted simply because the speed at which they have developed in recent years means that every year brings a new tool to the market - a tool to use in medicine, healthcare, business, commerce, and more.

The future of deep learning cannot be predicted with certainty - if we had an ultra-powerful AI, it might be able to make an accurate prediction of what will happen next. Even so, **human brains** figure that the following will happen over the next few years:

Better Learning

The more they learn, the more powerful machines become. We have a long way to go before we see the first full AI that is capable of mimicking thought processes and emotions - but the more AI is learning, the faster it will continue to grow.

As I was saying earlier in this book, it is a snowballing effect - so the more the “machine learning ball” is rolling, the larger it will become, and the more strength it will have.

Better Cyber Attack Protection

While humans might be able to beat codes created by humans, it might be a little more difficult for hackers to break in when an AI is protecting the realms of data held by a company. Soon enough, artificial intelligence will be capable of better monitoring, prevention, and responses when it comes to database breaches, DDoS attacks, and other cyberthreats.

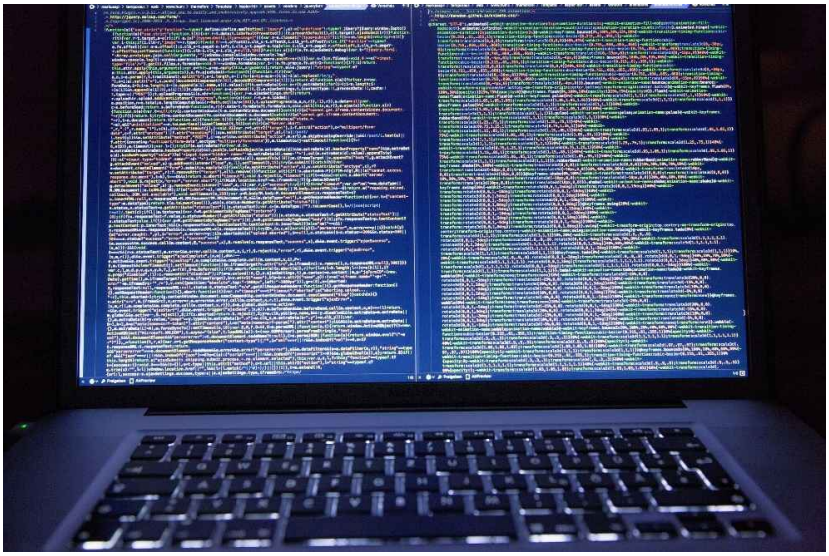
Better Generative Models

Generative models aim to mimic human beings as much as they can, in very specific areas. The Baidu example in the previous section is a very good indicator here. Over the next few years, we will start to see a lot more of these very convincing generative models, to the point where we will not be able to make a clear distinction between humans and machines (at least in some respects).

Better Training for Machines

Machine learning training is fairly new, given the rapid ascension of this industry in the past couple of decades. The more we train our machines, however, the better we will become at it - and this means that the machines themselves will be able to make better, more accurate decisions.

Chapter 4: The programmatic way



There are many models of Machine Learning. These theoretical describe the heuristics used to accomplish the ideal, allowing the machine to learn on their own. Below is a list and description of some of the most popular.

Decision Tree

Just about everyone has used the decision tree technique. Either formally or informally, we decide on a single course of action from many possibilities based on previous experience. The possibilities look like branches and we take one of them and reject the others.

The decision tree model gets its name from the shape created when its decision processes are drawn out graphically. A decision tree offers a great deal of flexibility in terms of what input values it can receive. As well, a tree's outputs can take the form of a category, binary, or numerical. Another strength of decision trees is how the degree of influence of different input variables can be determined by the level of decision node in which they are considered.

A weakness of decision trees is the fact that every decision boundary is a forced binary split. There is no nuance. Each decision is either yes or no, one

or zero. As well, the decision criteria can consider only a single variable at a time. There cannot be a combination of more than one input variable.

Decision trees cannot be updated incrementally. That is to say, once a tree has been trained on a training set, it must be thrown out and a new one created to tackle new training data.

Ensemble Methods address many tree limitations. In essence, the ensemble method uses more than one tree to increase output accuracy. There are two main ensemble methods — bagging and boosting.

The bagging ensemble method (known as Bootstrap Aggregation) is mean to reduce decision tree variance. The training data is broken up randomly into subsets and each subset is used to train a decision tree. The results from all trees are averaged, providing a more robust predictive accuracy than any single tree on its own.

The boosting ensemble method resembles a multi-stage rocket. The main booster of a rocket supplies the vehicle with a large amount of inertia. When its fuel is spent, it detaches and the second stage combines its acceleration to the inertia already imparted to the rocket and so on. For decision trees, the first tree operates on the training data and produces its outputs. The next tree uses the earlier tree's output as its input. When the input is in error the weighting it is given, makes it more likely the next tree will identify and at least partially mitigate this error. The end result of the run is a strong learner emerging from a series of weaker learners.

Linear Regression

The premise of linear regression methods rests on the assumption that the output (numeric value) may be expressed as a combination of the input variable set (also numeric). A simple example might look like this:

x = a1y1, a2y2, a3y3…

Where x is the output, a1...an are the weights accorded to each input, and y1...yn are the inputs.

The strength of a linear regression model lies in the fact it can produce well in terms of scores and performance. It is also capable of incremental learning.

A weakness of the linear regression model is the fact that it assumes linear input features, which might not be the case. Inputs must be tested mathematically for linearity.

K-Means Clustering Algorithm

K-Means clustering algorithms can be used to group results that talk about similar concepts. So, the algorithm will group all results that discuss jaguar as an animal into one cluster, discussions of Jaguar as a car into another cluster, and discussions of Jaguar as an operating system into a third. And so on.

Neural Network

We have covered neural networks in detail above. The strengths of neural networks are their ability to learn non-linear relationships between inputs and outputs.

Bayesian Network

Bayesian networks produce probabilistic relationships between outputs and inputs. This type of network requires all data to be binary. The strengths of the Bayesian network include high scalability and support for incremental learning. We discussed Bayesian models in more detail earlier in the book. In particular, this Machine Learning method is particularly good at classification tasks such as detecting if an email is or is not spam.

Python programming language does not allow special characters such as @, $, /, and % within identifiers. Python is a case sensitive programming language. Therefore, identifiers such as ‘Python’ and ‘python’ are two different identifiers in Python programming language.

Below are the naming conventions for identifiers in Python.

- Class name in Python always begins with an uppercase letter and all other Python identifiers starts with a lowercase letter.
- A Python identifier is private when such identifier begins with a single leading underscore.
- A Python identifier is strongly private when such identifier begins with two leading underscores.
- A Python identifier is a language-defined special name when such identifier ends with two trailing underscores.

Python Reserve Words

Reserve words in any programming language are special commands that compiler or interpreters understands, and these reserve words cannot be used as a constant or variable, or any other identifier names in that programming language.

Python has the following reserve words, and all such keywords contain lowercase letters only.

Python Keywords

Representing a Statement as Multi-Line

Statements in the Python language ends with a new line. If the statement is required to be continued into the next line, then the line continuation character (\) is used in Python language. This line continuation character (\) denotes that the statement line should continue as shown in the below screenshot. In the below example, we have three variables result1, result2 and result3 and the final output is copied to the variable named result. Instead of writing the equation statement in a single line (result=result1+result2+result3), here, we have used line continuation character (\) so that, it could be written in three lines but represents a single statement in Python language.

Also, a Python statement which is defined within braces (), {} and [] does not require the line continuation character (\) when written as a multi-line statement. This kind of Python statements are still interpreted as a single statement without the use of the line continuation character (\).

Quotation in Python

The Python language permits the use of single ('), double (") and triple ('" or """) codes to represent a string literal, making sure that the same type of quote begins and ends that string. In the below example, single, double and triple codes are used to represent a string in a word, sentence or paragraph. When we print this variable, they print the string irrespective of single, double and triple codes used for representing string literal in Python language.

Comments in Python

Any comment in the Python language is represented by a hash sign (#) provided it is not used inside a string literal between codes (single, double or triple). All characters after the hash sign (#) and up to the end of the physical

line are the part of comment and Python interpreter ignores this statement while interpreting the whole program. In the below example, the interpreter will just print the string present inside the print command and will ignore the parts mentioned after a sign before and after as comments.

Using Blank Lines

A blank line in the Python language is either a line with white spaces or a line with comments (i.e. statement starting with a hash sign (#)). The Python interpreter while interpreting a blank line, ignores it and no machine readable code will be generated. A multiline statement in Python is terminated after entering an empty physical line.

Waiting for the User

Using the Python programming language, we can set up the prompt which can accept a user's input. The following line of the program will display a prompt, which says "Press any key to exit", and waits for the user input or action.

Multiple Statements on a Single Line

The Python language allows to write multiple statements on a single line if they are separated by a semicolon (;) as demonstrated in the example below.

Command Line Arguments

On UNIX OS, which has Python interpreter installed, we can take help and see all the lists of the functions. These are the basic ones. The below screenshot demonstrates the help command on the UNIX system and all the functions or short codes used.

First, we need to define a robot. They are machines, often programmable by a computer, that are able to carry out a series of complex actions without intervention. A robot can have its controls systems embedded or be controlled by an external device. In popular literature and film, many robots are designed to look like people, but in fact, they are usually designed to perform a task and that requirement determines how they appear.

Robots have been with us for almost as long as computers. George Devol invented the first digitally operated and programmable one in 1954, called Unimate. In 1961, General Motors bought it to use for lifting hot metal die castings. And like computers, robots have changed our society. Their strength, agility, and ability to continue to perfectly execute the same repetitive tasks have proved an enormous benefit. And while they did cause some serious disruption to the manufacturing industries, putting many people out of work, their ascension in our societies has provided far more employment opportunities than they have taken.

Robots in current use can be broken down into several categories:

Industrial Robots/Service Robots

These robots are probably familiar. You have likely seen them on the television or streaming video of automated factories. They usually consist of an “arm” with one or more joints, which ends with a gripper or manipulating device. They first took hold in automobile factories. They are fixed in one location and are unable to move about. Industrial robots will be found in manufacturing and industrial locations. Service robots are basically the same in design as industrial robots but are found outside of manufacturing concerns.

Educational Robots

These are robots used as teacher aids or for educational purposes on their own. As early as the 1980s, robots were introduced in classrooms with the turtles, which were used in classrooms where students could train them using the Lego programming language. There are also robot kits available for purchase like the Lego Mindstorm.

Modular Robots

Modular robots are consisted of several independent units that work together. They can be identical or have one or more variation in design. Modular robots are able to attach together to form shapes that allow them to perform tasks. The programming of modular robotic systems is of course, more complex than a single robot, but ongoing research in many universities and corporate settings is proving that this design approach is superior to single large robots for many types of applications. When combined with Swarm Intelligence (see below), modular robots are proving quite adept at creative problem-solving.

Collaborative Robots

Collaborative robots are designed to work with human beings. They are mostly industrial robots that include safety features to ensure they do not harm anyone as they go about their assigned tasks. An excellent example of this kind of collaborative robot is Baxter. Introduced in 2012, Baxter is an industrial robot designed to be programmed to accomplish simple tasks but is able to sense when it comes into contact with a human being and stops moving.

Of course, all the examples above do not require any artificial intelligence. When robots are coupled with machine learning, researchers use the term “Robotic Learning”. This field has a contemporary impact in at least four important areas:

Vision

Machine Learning has allowed robots to visually sense their environment and to make sense of what they are seeing. New items can be understood and classified without the need to program into the robot ahead of time what it is looking at.

Grasping

Coupled with vision, Machine Learning allows robots to manipulate items in their environment that they have never seen before. In the past, in an industrial factory, each time a robot was expected to interact with a different-shaped object, it would have to be programmed to know how to manipulate this new object before it could be put to work. With Machine Learning, the robot comes equipped with the ability to navigate new item shapes and sizes automatically.

Motion Control

With the aid of Machine Learning, robots are able to move about their environments and avoid obstacles in order to continue their assigned tasks.

Data

Robots are now able to understand patterns in data, both physical and logistical, and act accordingly on those patterns.

Example of Industrial Robots and Machine Learning

One example of the benefit of applying Machine Learning to robots is of an industrial robot which receives boxes of frozen food along a conveyor. Because it is frozen, these boxes often have frost, sometimes quite a lot of frost. This actually changes the shape of the box randomly. Thus, a traditionally-trained robot with very little tolerance for these shape changes would fail to grasp the boxes correctly. With Machine Learning algorithms,

the robot is now able to adapt to different shapes, random as they are and in real time, and successfully grasp the boxes.

Another industrial example includes a factory with over 90,000 different parts. It would not be possible to teach a robot how to manipulate these many items. With Machine Learning, the robot is able to be fed images of new parts it will be dealing with and it can determine its own method to manipulate them.

In 2019, there will be an estimated 2.6 million robots in service on the planet. That's up a million from 2015. As more and more robots are combined with Machine Learning algorithms, this number is sure to explode.

There are a lot of reasons you may want to Python. Or, you may just be looking for a place to start with the wide, wide world of programming. Regardless of your specific purpose and goal, I can guarantee you that the Python programming language will provide you with all of the tools that you need to do exactly what it is that you want to do.

If you're browsing the introductory pages of this book on a preview or something of the like, on the fence as to whether or not you want to learn Python, well, I can tell you that you probably should.

See how much easier it is? And this is just with rudimentary programming concepts. When you get into heavier programming concepts like variables, functions, and other things of that nature, Python actually goes out of the way to make them really simple and functional to handle. We'll get more into this in the other chapters, of course.

On top of this, Python, as I've already said, is a language which has endless opportunity for usage and growth. There are very few explicitly **bad** situations for Python. This makes it a great language to learn initially for the simple fact that you'll be able to get a ton of mileage out of it. There will be

very few times where, as a beginning programmer, you'll come up with a concept you'd like to carry out that you **won't** be able to conquer perfectly well with Python.

So enough of all that, how do we get started with Python? Well, you're going to need a few things first. Of course, you're going to need a computer, preferably with an active internet connection, but there are going to be more things that you need.

Let's start our programming adventure right here. I'm going to teach you everything that you need to get going with Python, so just follow along.

The first thing that you're going to need to grab is Python itself. You can get it by going to the Python website at http://python.org. You'll be able to download and install it for your respective operating system. Note that this isn't entirely relevant if you're working on Linux or macOS. Most versions of these operating systems will generally have a version of Python on it. You may, though, need to downgrade. If so, you're going to need to search for instructions which are relevant to your specific operating system because the instructions can vary depending upon which operating system you're using.

If you don't know whether Python is pre-installed or not, you can figure it out by going to either the Terminal (in macOS or Linux systems) or elevated Powershell (in Windows) and running the command "python". If your instance of Terminal or Powershell says something along the lines of "Command not recognized", then congratulations, you need to install Python. If it recognizes the command and says "Python 3.x.x" in the program initiation text, you need to downgrade.

Anyhow, on the Python site, you need to be certain that you're getting Python version 2.7.12, and **not** Python version 3. This is for one simple reason: Python 2 and 3 are a bit different in the way that they handle certain

core functions. Not glaringly so, but one of the reasons that someone learns to program is so that they can speak directly to the computer. Learning Python 2 will give you many more opportunities to do this. This is for the simple reason that it's been **around** longer. There's much more code written for Python 2 than Python 3. You'll be able to work with a lot more code if you learn to work with Python 3 first. Now, granted, Python 3 is of course bleeding edge. But when you're learning to program, bleeding edge is not always best. It's preferable here that you learn how to read and write and deal with more code than you would otherwise. Basically, you're trying to cover as much ground with as little as possible when you're starting out, and Python 2 accommodates you for that goal perfectly. The few things you'll have to learn or relearn for Python 3 once you're more experienced with Python or programming in general will be non-factors compared to trying to understand deprecated code samples from Python 2 which aren't relevant to the code that you're trying to write.

With that out of the way, once you have Python 2.7.13 (or the latest version of Python 2 available) installed, you're going to need to get a text editor. It doesn't matter exactly what text editor you use, there are several good different candidates. The one I specifically recommend to use is called Atom, and you can get it at http://atom.io. There are a few reasons that I recommend this one. Firstly, it's free. What's better than high quality free software? Secondly, it's super simple to use and jam packed with features right out of the box, yet without being a sort of bloated and ugly software. Thirdly, it's incredibly extensible: there's nigh endless support for it through various different extensions and things of the sort online. There's so much that you can do with Atom. The endless possibilities of it and Python complement each other perfectly. And last but not least, you can get it for every major operating system.

Once you have Atom and Python installed, let's get right down to business on your first program. I'd recommend firstly that you create a new folder on your computer in an easily accessible place like the desktop or high up on your C drive. Name it something along the lines of Python. I don't recommend adding spaces in the name because it will slow down the process of navigating to it by just a little bit. Anyway, after you do that, you're going to open Atom. Once it's open, you're going to right click on the sidebar and click "Add Project Folder". Then you navigate to the project folder on your computer and select it. Then, you double click the folder in the sidebar in order to make it the active folder, and you right click on the sidebar and select "New File". Then after that, you're going to type hello.py.

In the file, you're going to type the following:

```
print "hello world!\n"
```

Then go ahead and save. Open up your Terminal or Powershell in order to run this. Navigate to the file you just made. If you don't know how to navigate in the command line, it's a worthwhile skill for any programmer in the making. If you don't know how to get around in the command line, then you need to. You can find a lot of useful and simple guides on Google that can teach you the basics in no time. I'm sorry to keep delegating things to Google searches, but it's really just that these topics go far beyond the scope of this book and, in the interest of staying focused and on topic, I think it's pertinent that I stay relevant to the topics at hand.

Anyway, you're going to navigate to the file and run it with the following command:

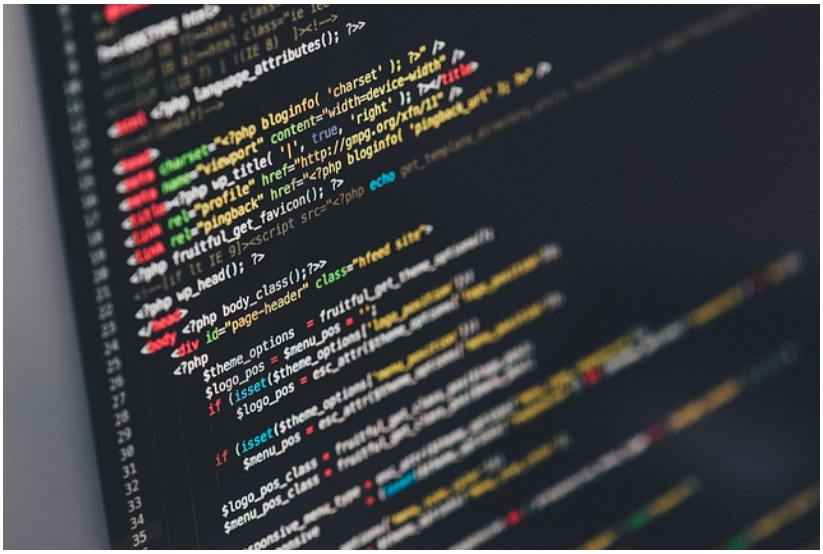
```
python hello.py
```

If everything goes according to plan, then you should get an output just like this:

hello world!

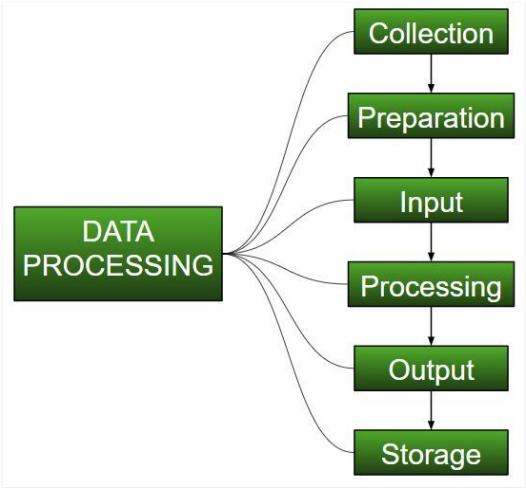
If that’s the case, then congratulations! You just wrote your very first Python program. You’ve taken the first crucial steps to being a fully able programmer. It’s all uphill from here.

Chapter 5: Organize data using effective pre-processing techniques



Data processing is the act of changing the nature of data into a form that is more useful and desirable. In other words, it is making data more meaningful and informative. By applying machine learning algorithms, statistical knowledge, and mathematical modeling, one can automate this whole process. The output of this whole process can be in any form like tables, graphs, charts, images, and much more, based on the activity done and the requirements of the machine.

This might appear simple, but for big organizations and companies like Facebook, Twitter, UNESCO, and health sector organizations, this whole process has to be carried out in a structured way. The diagram below shows some of the steps that are followed:



Let’s look in detail at each step:

Collection The most important step when getting started with Machine Learning is to ensure that the data available is of great quality. You can collect data from genuine sources such as Kaggle, data.gov.in, and UCI dataset repository. For example, when students are getting ready to take a competitive exam, they always find the best resources to use to ensure they attain good results. Similarly, accurate and high-quality data will simplify the learning process of the model. This means that during the time of testing, the model would output the best results.

A great amount of time, capital, and resources are involved in data collection. This means that organizations and researchers have to select the correct type of data which they want to implement or research.

For instance, to work on the Facial Expression Recognition requires a lot of images that have different human expressions. A good data will make sure that the results of the model are correct and genuine.

Preparation

The data collected can be in raw form. Raw data cannot be directly fed into a machine. Instead, something has to be done on the data first. The preparation stage involves gathering data from a wide array of sources, analyzing the datasets, and then building a new data set for additional processing and exploration. Preparation can be done manually or automatically and the data should be prepared in numerical form to improve the rate of learning of the model.

Input

Sometimes, data already prepared can be in the form which the machine cannot read, in this case, it has to be converted into readable form. For conversion to take place, it is important for specific algorithm to be present.

To execute this task, intensive computation and accuracy is required. For example, you can collect data through sources like MNIST, audio files, twitter comments, and video clips.

Processing

In this stage, ML techniques and algorithms are required to execute instructions generated over a large volume of data with accuracy and better computation.

Output

In this phase, results get procured by the machine in a sensible way such that the user can decide to reference it. Output can appear in the form of videos, graphs, and reports.

Storage

This is the final stage where the generated output, data model, and any other important information are saved for future use.

Data Processing in Python

Let’s learn something in python libraries before looking at how you can use Python to process and analyze data. The first thing is to be familiar with some important libraries. You need to know how you can import them into the environment. There are different ways to do this in Python.

You can type:

From math import *

In the first way, you define an alias m to library math. Then you can use different functions from the math library by making a reference using an alias m. factorial ().

In the second method, you import the whole namespace in math. You can choose to directly apply factorial () without inferring to math.

Note:

Google recommends the first method of importing libraries because it will help you tell the origin of the functions.

The list below shows libraries that you’ll need to know where the functions originate from.

NumPy: This stands for Numerical Python. The most advanced feature of NumPy is an n-dimensional array. This library has a standard linear algebra function, advanced random number capability, and tools for integration with other low-level programming languages.

SciPy: It is the shorthand for Scientific Python. SciPy is designed on NumPy. It is among the most important library for different high-level

science and engineering modules such as Linear Algebra, Sparse matrices, and Fourier transform.

scikit-learn: This is designed for machine learning. It was created on matplotlib, NumPy, and SciPy. This specific library has a lot of efficient tools for machine learning and statistical modeling. That includes regression, classification, clustering, and dimensionality community.

StatsModels: This library is designed for statistical modeling. Statsmodels refers to a Python module which permits users to explore data, approximate statistical models, and implement statistical tests.

Other libraries

- Requests used to access the web.
- Blaze used to support the functionality of NumPy and Pandas.
- Bokeh used to create dashboards, interactive plots, and data applications on the current web browsers.
- Seaborn is used in statistical data visualization.
- Regular expressions that are useful for discovering patterns in a text data
- NetWorx and Igraph applied to graph data manipulations.

Now that you are familiar with Python fundamentals and crucial libraries, let's now jump into problem-solving through Python.

An exploratory analysis in Python with Pandas

If you didn't know, Pandas is an important data analysis library in Python. This library has been key at improving the application of Python in the data science community. Our example uses Pandas to read a data set from an

analytics Vidhya competition, run exploratory analysis, and create a first categorization algorithm to solve this problem.

Before you can load the data, it is important to know the two major data structures in Pandas. That is Series and DataFrames.

Series and DataFrames

You can think of series as a 1-dimensional labeled array. These labels help you to understand individual elements of this series via labels.

A data frame resembles an Excel workbook, and contains column names which refer to columns as well as rows that can be accessed by row numbers. The most important difference is that column names and row numbers are referred to as column and row index.

Series and data frames create a major data model for Pandas in Python. At first, the datasets have to be read from data frames and different operations can easily be subjected to these columns.

Import libraries and data set

This chapter will use the following python libraries:

- **NumPy**
- **Matplotlib**
- **Pandas**

Once you have imported the library, you can move on and read the dataset using a function read_csv(). Below is how the code will look till this point.

```
import pandas as pd
import numpy as np
import matplotlib as plt
%matplotlib inline
#Reading the dataset in a dataframe using Pandas
df = pd.read_csv("/home/kunal/Downloads/Loan_Prediction/train.csv")
```

Notice that the dataset is stored in

Once you read the dataset, you can decide to check a few top rows by using the **function head().**

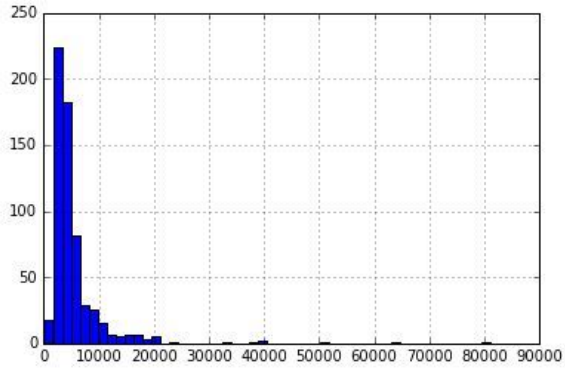
Next, you can check at the summary of numerical fields by using the **describe () function.**

Distribution analysis

Since you are familiar with basic features of data, this is the time to look at the distribution of different variables. Let's begin with numeric variables- ApplicantIncome and LoanAmount.

First, type the commands below to plot the histogram of ApplicantIncome.

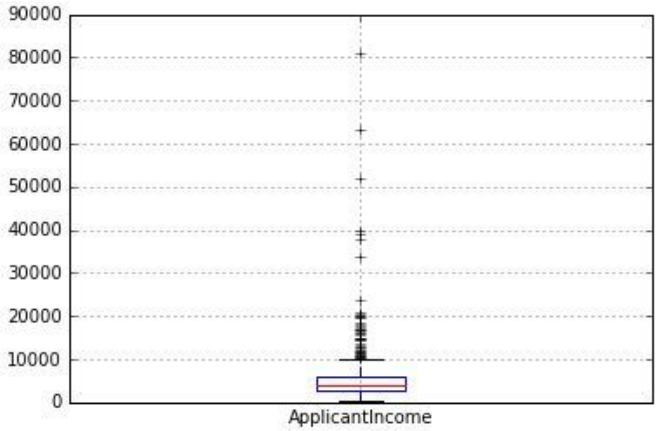
```
df['ApplicantIncome'].hist(bins=50)
```



Notice that there are a few extreme values. This is why 50 bins are needed to represent the distribution clearly.

The next thing to focus on is the box plot. The box plot for fare is plotted by:

```
df.boxplot(column='ApplicantIncome')
```



This is just a tip of an iceberg when it comes data processing in Python.

Let’s look at:

Techniques for Preprocessing Data in Python

Here are the best techniques for Data Preprocessing in Python.

1. Rescaling Data

When you work with data that has different scales, you need to rescale the properties to have the same scale. The properties are rescaled between the range 0 to 1 and refer to it as normalization. To achieve this, the

MinMaxScaler class from scikit-learn is used.

```
>>> import pandas, scipy, numpy
>>> from sklearn.preprocessing import MinMaxScaler
>>> df=pandas.read_csv( 'http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv ',sep=';')
>>> array=df.values
>>> #Separating data into input and output components
>>> x=array[:,0:8]
>>> y=array[:,8]
>>> scaler=MinMaxScaler(feature_range=(0,1))
>>> rescaledX=scaler.fit_transform(x)
>>> numpy.set_printoptions(precision=3) #Setting precision for the output
>>> rescaledX[0:5,:]
```

For example:

```
>>> rescaledX[0:5,:]
array([[0.248, 0.397, 0.   , 0.068, 0.107, 0.141, 0.099, 0.568],
       [0.283, 0.521, 0.   , 0.116, 0.144, 0.338, 0.216, 0.494],
       [0.283, 0.438, 0.04 , 0.096, 0.134, 0.197, 0.17 , 0.509],
       [0.584, 0.11 , 0.56 , 0.068, 0.105, 0.225, 0.191, 0.582],
       [0.248, 0.397, 0.   , 0.068, 0.107, 0.141, 0.099, 0.568]])
```

After rescaling, you get the values between 0 and 1. By rescaling data, it confirms the use of neural networks, optimization algorithms as well as those which have distance measures such as the k-nearest neighbors.

2. Normalizing Data

In the following task, you rescale every observation to a specific length of 1. For this case, you use the Normalizer class. Here is an example:

```
>>> from sklearn.preprocessing import Normalizer
>>> scaler=Normalizer().fit(x)
>>> normalizedX=scaler.transform(x)
>>> normalizedX[0:5,:]
```

```
>>> normalizedX[0:5,:]
array([[2.024e-01, 1.914e-02, 0.000e+00, 5.196e-02, 2.079e-03, 3.008e-01,
        9.299e-01, 2.729e-02],
       [1.083e-01, 1.222e-02, 0.000e+00, 3.611e-02, 1.361e-03, 3.472e-01,
        9.306e-01, 1.385e-02],
       [1.377e-01, 1.342e-02, 7.061e-04, 4.060e-02, 1.624e-03, 2.648e-01,
        9.533e-01, 1.760e-02],
       [1.767e-01, 4.416e-03, 8.833e-03, 2.997e-02, 1.183e-03, 2.681e-01,
        9.464e-01, 1.574e-02],
       [2.024e-01, 1.914e-02, 0.000e+00, 5.196e-02, 2.079e-03, 3.008e-01,
        9.299e-01, 2.729e-02]])
```

3. Binarizing Data

If you use the binary threshold, it is possible to change the data and make the value above it to be 1 while those that are equal to or fall below it, 0. For this task, you use the Binarized class.

```
>>> from sklearn.preprocessing import Binarizer
>>> binarizer=Binarizer(threshold=0.0).fit(x)
>>> binaryX=binarizer.transform(x)
>>> binaryX[0:5,:]
```

```
>>> binaryX[0:5,:]
array([[1., 1., 0., 1., 1., 1., 1., 1.],
       [1., 1., 0., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 0., 1., 1., 1., 1., 1.]])
```

As you can see, the python code will label 0 over all values equal to or less than 0, and label 1 over the rest.

4. Mean Removal

This is where you remove mean from each property to center it on zero.

5. One Hot Encoding

When you deal with a few and scattered numerical values, you might need to store them before you can carry out the One Hot Encoding. For the k-distinct values, you can change the feature into a k-dimensional vector that has a single value of 1 and 0 for the remaining values.

```
>>> from sklearn.preprocessing import OneHotEncoder
>>> encoder=OneHotEncoder()
>>> encoder.fit([[0,1,6,2],
[1,5,3,5],
[2,4,2,7],
[1,0,4,2]
])
```

6. Label Encoding

Sometimes labels can be words or numbers. If you want to label the training data, you need to use words to increase its readability. Label encoding changes word labels into numbers to allow algorithms operate on them. Here's an example:

```
>>> from sklearn.preprocessing import LabelEncoder
>>> label_encoder=LabelEncoder()
>>> input_classes=['Havells','Philips','Syska','Eveready','Lloyd']
>>> label_encoder.fit(input_classes)
```

Machine learning Projects

You can read volumes of theory, but that can never help you build confidence in the subject when compared to hands-on practice. Most people

believe that they become masters when they read textbooks and articles. But, when they try to apply the theory, they notice that it is harder than it looks.

When you work on projects, you improve your skills and have the chance to explore interesting topics. You can also add these projects to your portfolio, which will make it easier for you to land a job.

Data Analysis

The data analysis project is one that most people begin with since it helps to hone some statistical skills. By working on this project, you can build your practical intuition around machine learning. The goal of this exercise is to help you use different models and apply those models to a variety of datasets. There are three reasons why you should begin with this project.

• You will learn to identify which model will fit your problem. For example, some datasets may have missing information. When you work on this project, you will know which model you should use for such datasets. You can always dig through texts and articles to learn which model is better, but it is better to see it in action.

• You will develop the art of creating prototypes of models. It is often difficult to identify if a model will work best for the problem without trying it.

• You must keep the process flow or workflow of the machine in mind when you build a model. You will master this technique when you work on this project. For example, you can practice:

How to import data

Which tools should you use to clean data

How to split the data into training sets

Pre-processing of data

What transformations must you make to the data

Since you use different models, you can focus on the development of the skills mentioned above. For instructions, you can look at the Python (sklearn) and R (caret) documentation pages. You must also practice classification, clustering and regression algorithms.

Social Media

Social media is synonymous with big data because of the volumes of content that users generate. You can mine this data and keep tabs on trends, public sentiments and opinions. Facebook, YouTube, WhatsApp, Instagram and many other platforms have data that you can use to achieve your objective.

Social media data is important for branding, marketing and for a business since every generation spends more time on social media when compared to its preceding generation. Twitter is one of the platforms you should start with when you begin to practice machine learning. You have interesting data sets and metadata that can open an endless path for analysis.

Fantasy Leagues or Sports Betting

If you have read the book Moneyball, you know how the Oakland A's changed the face of baseball through their analytical scouting. Based on the analysis, the team built a highly competitive squad and bought the players for almost a third of the price when compared to the Yankees players. If you have not read the book yet, you must grab a copy and read it now.

There are large volumes of data in the sports world that you can play with. You can use data for games, teams, players and scores to analyze patterns. These data are available online. For instance, you can try the following projects and see where they lead you.

You can bet on specific box scores in different games based on the data available on the Internet right before a game begins. You can analyze the data and identify where you should place your bets to increase your chances of winning.

You can scout for talent in schools and colleges using the statistics for different players. This analysis can help college scouts identify the players they want on their team.

You can become a manager for a fantasy team and use your analysis to win games and leagues.

The sports domain is especially useful since it helps a beginner practice exploratory analysis and data visualization. These skills will help you identify what types of data you should include in your analysis.

Stock Market

An aspiring data scientist will always base his projects on the stock market since that is like Disneyland for them. The stock market is an interesting area to work on since you have different types of data available. You can find data on fundamentals, prices, stocks, economic scenarios, macroeconomic and microeconomic indicators, volatilities and many more.

The data available is granular which makes it easier for people to obtain historical data, also called time series data, for every company. Using this data, you can build creative trading strategies. Since financial markets have small cycles, you can validate any prediction you make immediately.

You can analyze the data and assess the movement of prices for different stocks for a period of 6 months. You should use the quarterly reports provided by the company to make this prediction.

You can use a time series models and recurrent neural network models to forecast the correlation between the stock prices. You can also assess the volatility of the prices.

It is simple to build a financial trading model to improve your skills in machine learning. However, it is difficult to profit from these models here. Experts recommend that you do not use real money to trade until you have perfected the model.

Writing Machine Learning Algorithms

It is important for a beginner to learn how to write machine learning algorithms from scratch for two reasons:

There is no better way to understand the mechanics of the algorithms since you must think about the process flow of the algorithm thereby helping you master algorithms.

You can translate statistical and mathematical instructions into working code, which allows you to apply algorithms in your academic research.

It is best to start off with simpler algorithms since you must make hundreds of decisions. Once you are comfortable with simple algorithms, you can extend the functionality of those algorithms to complex algorithms like regression and clustering.

Your packages may be slow and may not be as fancy as the algorithms in existing packages. It took the developers years to build these packages.

Chapter 6:Getting grip to a deeper textual and social media data

Machine learning technological approach is radically different from the way companies traditionally exploit data. Instead of starting with business logic and applying the data, machine learning techniques allow data to create logic. One of the key benefits of this approach is the removal of commercial assumptions and prejudices that may lead managers to customize a strategy that may not be the best.

THE IMPACT OF MACHINE LEARNING ON APPLICATIONS

We boldly stated that with machine learning, you start with the data and let the data take you to logic. How does a company achieve the goal? As with all the development and deployment of complex applications, you need a planning process to understand the business issue to be solved and to collect the appropriate data sources.

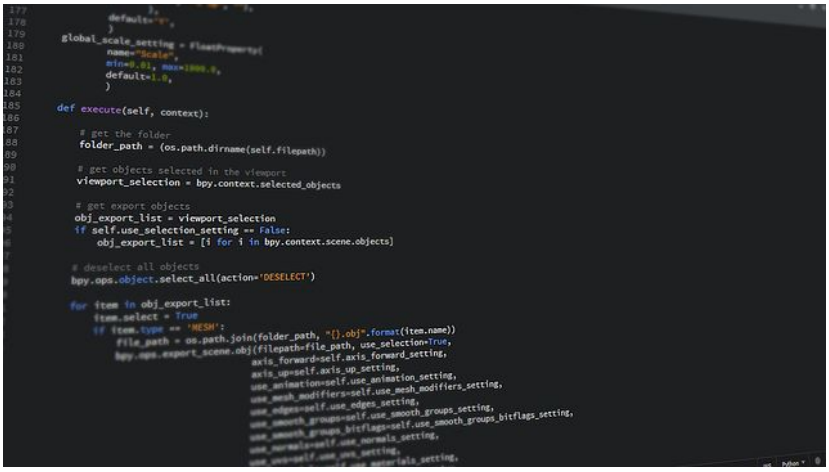
The formulas used are pretty basic, most of them are not different from what you learned in high school. The difference is that computers can do them much faster than we can with our human minds. Even if we are the best at figuring out these types of math problems, humans can never calculate them as quickly as a computer can. Our minds aren’t built that way. While it may

take us a minute or so to calculate a few numbers in our head, computers have the capability of figuring out millions of equations in a fraction of the time.

But that does not mean that computers are an improved version of the human mind. There are things that the human mind can do equally fast that computers have yet to figure out. Actions like pattern recognition, unstructured problem solving and functioning in 3-dimensional space. Yes, more advanced computers can perform these types of tasks and solve problems, but they have to be given structure and parameters in areas where rules must be followed. Human minds are flexible enough to adapt their thinking to varying circumstances in a vast array of areas. Yes, computers can operate cars and come to quick conclusions, but they have difficulty multitasking and maneuvering in areas without bumping into things or encountering an error.

However, with each passing year, the latest computers are capable of doing more and more things that they could never do in the past. It seems to be just a matter of time before they will one day be capable of doing everything that humans can do but much faster and far more efficiently.

Chapter 7: Optimize your machine learning systems and algorithms



You will find that with machine learning, it is important to recognize that there will be a relationship that will form between this process and the probability theory. Machine learning can be a broad field, and this means that it can intersect with some other fields. The fields that it interacts with will depend on the specific project you will work with. Probability and statistics often merge with machine learning so understanding how these three can work together can be important for your project.

There are a few different ways that statistics and the probability theory will be really important to the whole learning process that goes on with machine learning. First, you have to be able to pick out the right algorithm, and there are quite a few different ones that you can pick from as you will see later on as we progress through this book. The algorithm that you end up picking out needs to have a good balance of things like accuracy, training time, complexity, and a number of parameters. And as you work more with machine learning, you will notice that each project will need a different combination of these factors.

Using the probability theory and statistics, you can better pick out the right parameters for the program, the validation strategies, and make sure that you pick out the right algorithm for your needs. They can be helpful as well for

letting you know what level of uncertainty is present inside of your choice so you can guess how much you can trust what is going on.

The probability theory and statistics will help you out quite a bit when it comes to working in machine learning and can help you to understand what is going on with the projects you are working on.

Looking at random variables

Now, the first topic we need to look at when it comes to statistics is random variables. With probability theory, these random variables will be expressed with the “X” symbol, and it is the variable that has all its possible variables come out as numerical outcomes that will come up during one of your random experiments. With random variables, there will be either continuous or discrete options. This means that sometimes your random variables will be functions that will map outcomes to the real value inside their space. We will look at a few examples of this one to help it make sense later on.

We will start out with an example of a random variable by throwing a die. The random variable that we will look at will be represented by X, and it will rely on the outcome that you will get once the die is thrown. The choices of X that would come naturally here will go through to map out the outcome denoted as 1 to the value of i.

What this means is that if X equals 1, you would map the event of throwing a one on your die to being the value of i. You would be able to map this out with any number that is on the die, and it is even possible to take it to the next step and pick out some mappings that are a bit strange. For example, you could map out Y to make it the outcome of 0. This can be a hard process to do, and we aren’t going to spend much time on it, but it can help you to see how it works. When we are ready to write out his one, we would have

the probability, which is shown as P of outcome 1 of random variable X. it would look like the following:

PX(i) or (x=i)

Distribution

Now we need to look at what the probability distribution is like with this process. What we mean here is that we will look at see what the probability of each outcome will be for the random variable. Or, to make it simple, we will see how likely it is that we will get a specific number, like a six or a three, when we throw the die.

To get started with this, we will need to look at an example. We will let the X, or the random variable, be our outcome that we get once the diet is thrown. We will also start with the assumption that the die is not loaded so that all six sides will have the same probability of showing up each time that you throw the diet. The probability distribution for throwing your die and getting a specific number includes:

PX(1) = PX(2) = … = PX(6) = 1/6

In this example, it matches up to the what we did with the random variables, it does have a different type of meaning. Your probability distribution is more about the spectrum of events that can happen, while our random variable example is all about which variables are there. With the probability theory, the P(X) part will note that we are working with our probability distribution of the random variable X.

While looking through these examples, you can notice that your distribution will sometimes include two or more variables at the same time. When this happens, we will call it a joint distribution. Your probability will now be determined by each of the variables if there are more than one, that is now involved.

To see how this process will work, let's say that the X is random and that it is defined by what outcome you get when you throw the die, and the Y will be a random variable that will tell you what results that you get when you flip a coin. We will assign a 1 to this coin toss if we get heads at the end, and a 0 will show up if you get tails. This makes it easier when we figure out what the probability distribution is for both of these variables.

Independence

Another variable that you can work with when doing machine learning is to figure out how much independence the problem has. When you are doing random variables, you will find that they will end up being independent of what the other random variables are as long as the variable distribution doesn't change when a new variable is introduced to the equation.

You can make some assumptions about your data in machine learning to help make things easier when you already know about the independence. An example of this is the training sample of "j and i" will be independent of any underlying space when the label of sample "i" is unaffected by the features sample "j". No matter what one of the variables turns out, the other one is not going to be affected by that.

Think back to the example of the die and the coin flip. It doesn't matter what number shows up on the die. The coin will have its own result. And the same can be said the other way around as well. The X random variable is always going to be independent of the Y variable. It doesn't matter the value of Y, but the following code needs to be true for it:

$P(X) = P(X|Y)$.

In the case above, the values that come up for X and for Y variables are dropped because, at this point, the values of these variables are not going to matter that much. But with the statement above, it is true for any type of

value that you provide to your X or Y, so it isn't going to matter what values are placed in this equation.

The Building Blocks Needed for Machine Learning

There are some algorithms that you will want to learn how to use to do well when you work on machine learning. But before we get to those, it is important to learn a few of the basic building blocks of machine learning. Doing this will really help you when you are ready to work with the machine learning algorithms.

These algorithms are great because they help you to do a lot of amazing things in machine learning, and they are the main reason why you would want to use machine learning.

The learning framework

Let's say that you decide that it is time to go on vacation to a new island. The natives that you meet on this island are really interested in eating papaya, but you have very limited experience with this kind of food. But you decide that it is good to give it a try and head on down to the marketplace, hoping to figure out which papaya is the best and will taste good to you.

Now, you have a few options as to how you would figure out which papaya is the best for you. You could start by asking some people at the marketplace which papayas are the best. But since everyone will have their own opinion about it, you will end up with lots of answers. You can also use some of your past experiences to do it.

At some point or another, you have worked with fresh fruit. You could use this to help you to make a good choice. You may look at the color of the papaya and the softness to help you make a decision. As you look through the papaya, you will notice that there are a ton of colors, from dark browns

to reds, and even different degrees of softness, so it is confusing to know what will work the best.

Learner's input

The first section of the framework that you need to look at is called the learner's input. To do this, you need to find a domain set and then focus on it. This domain can be an arbitrary set that is found in the objects, which in this framework is known as the points, that you need to get labeled. So, going back to the exercise about the papaya, you would have the domain set be any of the papayas that you are checking out. Then the domain points would be able to use the vectors of features, which in this case includes the softness and color of the fruit.

Once you have determined what domain points and domain sets you want to use, you can then go through and create the label set that you will use. In this exercise, the label set will hold onto the predictions that you will make about the papayas. You can look at each papaya and then make a prediction on how it tastes and whether it is the best one for you.

The label set that you get with this exercise will have two elements. The X will be any of the papayas that you think will taste bad. And then the Y will be the ones that you feel taste the best.

From here, you can work on what is known as the training data. This training data will be a set which can hold the sequence pairs that you will use when testing the accuracy of your predictions. So, with the exercise of the papayas, the training data will be the papayas that you decide to purchase. You will then take these home and taste them to see what tastes the best. This can help you to make better decisions later on when you purchase papayas. If you find that you really like a specific softness or color, ensure that you purchase that kind the next time.

Learner's output

Now that you have your input in, you will want to work on the output. The output is basically going to be the creation of a rule of prediction. It often goes by the name of predictor, classifier, or hypothesis, which you will then use to take the domain points and label them. With the papaya example, this rule will be the standard, which you get to set to where you want, and which will be used to help you figure out whether a papaya that you purchase will taste good or not, even before you eat it in the future.

When you first start, you are basically making guesses because you have no idea which papaya will be good or not. You can use some of your experiences from the past to help if you want, but since you haven't had a papaya before, it is hard to know what will taste good or not. But as you try out papayas and get some more experience with them, you will find that your future predictions will get much better.

Data generalization model

Once you have done the learner's input and output, you will need to take a look at what is known as a data generalization model. This model is nice because it can help you to create your own data for training based on the probability distribution of the domain sets that you used with the papayas. With this example, the model will be the method that you will use to decide what papayas you want to grab at the market to test them out at home.

In the beginning, you may not have any idea of what the distribution is. The data generalization model is designed to help you out, even if you don't know which ones to pick out from the beginning.

Measure of success

Before you can take time to work with the model above, you must make sure that you have some sort of method in place that can help you figure out

whether you are successful or not in the project. There are a ton of options that you can choose with the papayas, but there must be some indicator along the way that will help you make the best predictions about whether you will see that success that you want or not.

Since the goal of this experiment is to help you figure out the fruits that will taste the best, so you are set in the future to get the ones that you like, you can use the error of the predictor simply by making sure that you pick out a range of different types of papayas when you are at the market. With all of this variety, it is easier to taste them at home and figure out which ones you like the most. Make sure to write down your observations as you eat each one. This will help you when you go back to the market because then you are more likely to pick out the fruits that you like the best.

PAC learning strategies

While we spent some time talking about how to set up a hypothesis and a training data set to get started with learning strategies in the above section, we still haven't spent time learning about PAC learning, There are two parameters that need to be found in this kind of learning including the accuracy parameter and the output classifier.

First, we need to look at the accuracy parameter. This is the parameter that will be used to determine how often the output classifier that you set up at the start will be able to make correct predictions. These predictions need to be set up to be based on information that you provide. You can also work with what is known as a confidence parameter. This parameter will measure how likely it is that your predictor can reach a certain level of accuracy. Accuracy can be important based on the type of project that you are working on so you should definitely look into this kind of learning if your project needs to maintain a high level of accuracy.

There are several ways that PAC can come in handy when you are doing a project. You may want to use it when you do training data to help see how accurate the model you have is. you may want to bring it into the learning when you feel that some uncertainties will come up and you want to ensure that your computer can handle them. Keep in mind that any time you work with the PAC learning model; it is always going to generate a few random training sets that you should watch out for.

Generalization models in machine learning

In machine learning, when you are considering what the idea of generalization is about, you are basically seeing that there are two components present and you will need to use both of them before you can get through all the data. The components that need to be present include the reliability assumption and revisiting the true error rate.

Any time that you can work with this, and you can meet the reliability assumption, you will be able to expect that the algorithm that you use in machine learning to get the results is pretty reliable for helping you know the distribution. But, there are also times when the assumption that you make here is not going to be very practical. This means that the standards that you picked out may have been unrealistic and that you went with the wrong algorithm to get all the work done.

In addition, the type of algorithm that you try to pick out for machine learning doesn’t guarantee that you come up with a hypothesis that is something you like. Unlike using the Bayes predictor, which is an algorithm we will talk about more, later on, these algorithms are not set up to find which type of error rate is the best for you either.

In machine learning, there will be times when you need to make assumptions and use the experience that you have, either in that area or a similar area, to

get things done. In some cases, you may even need to do some experimenting to figure out what you want to do. But machine learning can help to get this done.

These are the basic building blocks that you will need to understand and use when you are doing machine learning. These are so important because they can help you see how the programs that run on machine learning are working.

Conclusion

Thank you for making it to the end of this book. The impact of Machine Learning on our world is already ubiquitous. Our cars, our phones, our houses, and so much more are already being controlled and maintained through rudimentary Machine Learning systems. But in the future, Machine Learning will radically change the world. Some of those changes are easy to predict. In the next decade or two, people will no longer drive cars, instead, automated cars will drive people. But in many other ways, the effect of Machine Learning on our world is difficult to predict. Will Machine Learning algorithms replace so many jobs, from trucking to accounting, to many other disciplines, that there won't be much work left for people? In 100 years, will there be work for anyone at all? We don't know the answer to questions like this because there is so far no limit to what Machine Learning can accomplish, given time and data and the will to use it to achieve a particular task.

The future is not necessarily frightening. If there is no work in the future, it won't mean that things aren't getting done. Food will still be grown, picked, transported to market, and displayed in stores. It's just that people won't have to do any of that labor. As a matter of fact, stores won't be necessary either, since the food we order can be delivered directly to our homes. What will the world be like if human beings have almost unlimited leisure time? Is this a possible future?

There is no telling where this technology will take us in the future. Right now it is one of the most talked about topics in the field of IT. This is primarily because of its amazing potential in so many areas.

If technology continues to improve at such a rapid rate, there is a good chance that in the not too distant future, machines themselves will be

programming other machines. At that point, the best question to ask is not what machines will be doing in the future but what will we?

This book provides information on what machine learning is and types of machine learning. It also gives you information on the subjects that laid the foundation for machine learning. You will gather information on different algorithms used in machine learning. As a beginner, it is always good to practice some algorithms used in machine learning to enhance your understanding. There are some projects in the book that you can complete over the weekend or extend them if you want to. It is important to practice as much as you can to improve your knowledge on machine learning. It is difficult to remember the many words that are used in machine learning.

SQL CODING FOR BEGINNERS

AN ESSENTIAL TOOL FOR DEVELOPERS USING STATEMENTS FOR CONTROLLING AND MODIFYING TABLES, AND AN INTERMEDIATE-LEVEL GUIDE FOR LEARNING SQL PROGRAMMING STEP BY STEP

[Tony Coding]

Legal & Disclaimer

The information contained in this book and its contents is not designed to replace or take the place of any form of medical or professional advice; and is not meant to replace the need for independent medical, financial, legal or other professional advice or services, as may be required. The content and information in this book has been provided for educational and entertainment purposes only.

The content and information contained in this book has been compiled from sources deemed reliable, and it is accurate to the best of the Author's knowledge, information and belief. However, the Author cannot guarantee its accuracy and validity and cannot be held liable for any errors and/or omissions. Further, changes are periodically made to this book as and when needed. Where appropriate and/or necessary, you must consult a professional (including but not limited to your doctor, attorney, financial advisor or such other professional advisor) before using any of the suggested remedies, techniques, or information in this book.

Upon using the contents and information contained in this book, you agree to hold harmless the Author from and against any damages, costs, and expenses, including any legal fees potentially resulting from the application of any of the information provided by this book. This disclaimer applies to any loss, damages or injury caused by the use and application, whether directly or indirectly, of any advice or information presented, whether for

breach of contract, tort, negligence, personal injury, criminal intent, or under any other cause of action.

You agree to accept all risks of using the information presented inside this book.

You agree that by continuing to read this book, where appropriate and/or necessary, you shall consult a professional (including but not limited to your doctor, attorney, or financial advisor or such other advisor as needed) before using any of the suggested remedies, techniques, or information in this book.

Table of Contents

Introduction

It is best to start at the beginning. SQL is a programming language that stands for ‘Structured Query Language,’ and it is a simple language to learn considering it will allow interaction to occur between the different databases that are in the same system. This database system first came out in the 70s, but when IBM came out with its own prototype of this programming language, then it really started to see a growth in popularity and the business world started to take notice.

The version of SQL that was originally used by IBM, known back then as ORACLE, was so successful that the team behind it eventually left IBM and became its own company. ORACLE, thanks to how it can work with SQL, is still one of the leaders in programming languages and it is always changing so that it can keep up with everything that is needed in the programming and database management world.

The SQL is a set of instructions that you can use to interact with your relational database. While there are a lot of languages that you can use to do this, SQL is the only language that most databases can understand. Whenever you are ready to interact with one of these databases, the software can go in and translate the commands that you are given, whether you are giving them in form entries or mouse clicks. These will be translated into SQL statements that the database will already be able to interpret.

There are three main components, and we will discuss these in more depth in the next chapter. But these main ones will be ‘Data Control Language,’ ‘Data Definition Language,’ and ‘Data Manipulation Language.’

To illustrate how this works, think about a simple online catalog that allows you to search. The search page will often contain a form that will just have a

text box. You can enter the name of the item that you would like to search using the form and then you would simply need to click on the search button. As soon as you click on the search button, the web server will go through and search through the database to find anything related to that search term. It will bring those back to create a new web page that will go along with your specific request.

At first, this will seem really complicated, and you may be worried about how much work it will be to get it set up. But when you start to work on a few codes, you will find that it is not actually that hard to work with. Often, just reading out the SQL statement will help you to figure out what the command will do. Take a look at the code below:

```
DELETE
FROM students
WHERE graduation_year = 2014
```

Read that code out loud, and you can probably guess what will happen when you decide to execute your code. Most of the codes that are presented in the SQL will work like this, so you do not need to be an expert programmer to make the SQL language work for you. Considering this is a language that allows the authorized users to look through a database and find the information that they want, it makes sense that the SQL language is really easy to use.

SQL may sound like it is a complicated language for you to learn, but it is actually really easy to figure out. We will spend some time looking at all the different things that you can do when you decide to use the SQL language to help you control the database that you use in your company. It is made to be easy and to make things easier when you are working with the database, or

when you are trying to make it easy on your user to find information and SQL certainly delivers on this.

Chapter 1: Basics of SQL

If you are interested in learning a new coding language, there are a lot of different options that you can choose from, and it really depends on what you are looking for and what you want to do with them. Some of these languages are good for helping you to create a good website. Some are better for beginners while others are good for those who are more advanced. Some are good for creating a smartphone application or for working on your own game to share with others.

Traditionally, many companies would choose to work with the 'Database Management System,' or the DBMS to help them to keep organized and to keep track of their customers and their products. This was the first option that was on the market for this kind of organization, and it does work well. But over the years there have been some newer methods that have changed the way that companies can sort and hold their information. Even when it comes to the most basic management system for data that you can choose, you will see that there is a ton more power and security than you would have found in the past.

Big companies will be responsible for holding onto a lot of data, and some of this data will include personal information about their customers like address, names, and credit card information. Because of the more complex sort of information that these businesses need to store, a new 'Relational

Database Management System' has been created to help keep this information safe in a way that the DBMS has not been able to.

Now, as a business owner, there are some different options that you can pick from when you want to get a good database management system. Most business owners like to go with SQL because it is one of the best options out there. The SQL language is easy to use, was designed to work well with businesses, and it will give you all the tools that you need to make sure that your information is safe. Let's take some more time to look at this SQL and learn how to make it work for your business.

How this works with your database

If you decide that SQL is the language that you will work on for managing your database, you can take a look at the database. You will notice that when you look at this, you are basically just looking at groups of information. Some people will consider these to be organizational mechanisms that will be used to store information that you, as the user, can look at later on, and it can do this as effectively as possible. There are a ton of things that SQL can help you with when it comes to managing your database, and you will see some great results.

There are times when you are working on a project with your company, and you may be working with some kind of database that is very similar to SQL, and you may not even realize that you are doing this. For example, one database that you commonly use is the phone book. This will contain a ton of information about people in your area including their name, what business they are in, their address, and their phone numbers. And all this information is found in one place so you won't have to search all over to find it.

This is kind of how the SQL database works as well. It will do this by looking through the information that you have available through your

company database. It will sort through that information so that you are better able to find what you need the most without making a mess or wasting time.

Relational databases

First, we need to take a look at the relational databases. This database is the one that you will want to use when you want to work with databases that are aggregated into logical units or other types of tables, and then these tables have the ability to be interconnected inside of your database in a way that will make sense depending on what you are looking for at the time. These databases can also be good to use if you want to take in some complex information, and then get the program to break it down into some smaller pieces so that you can manage it a little bit better.

The relational databases are good ones to work with because they allow you to grab on to all the information that you have stored for your business, and then manipulate it in a way that makes it easier to use. You can take that complex information and then break it up into a way that you and others are more likely to understand. While you might be confused by all the information and how to break it all up, the system would be able to go through this and sort it the way that you need in no time. You are also able to get some more security so that if you place personal information about the customer into that database, you can keep it away from others, in other words, it will be kept completely safe from people who would want to steal it.

Client and server technology

In the past, if you were working with a computer for your business, you were most likely using a mainframe computer. What this means is that the machines were able to hold onto a large system, and this system would be good at storing all the information that you need and for processing options.

Now, these systems were able to work, and they got the job done for a very long time. If your company uses these and this is what you are most comfortable with using, it does get the work done. But there are some options on the market that will do a better job. These options can be found in the client-server system.

These systems will use some different processes to help you to get the results that are needed. With this one, the main computer that you are using, which would be called the 'server,' will be accessible to any user who is on the network. Now, these users must have the right credentials to do this, which helps to keep the system safe and secure. But if the user has the right information and is on your network, they can reach the information without a lot of trouble and barely any effort. The user can get the server from other servers or from their desktop computer, and the user will then be known as the 'client' so that the client and server are easily able to interact through this database.

How to work with databases that are online

There are a lot of business owners who will find that the client and server technology is the one that works for them. This system is great for many companies, but there are some things that you will need to add or take away at times because of how technology has been changing lately. There are some companies that like the idea that their database will do better with the internet so that they can work on this database anywhere they are located, whether they are at home or at the office. There are even times when a customer will have an account with the company, and they will need to be able to access the database online as well. For example, if you have an account with Amazon, you are a part of their database, and you can gain access to certain parts through this.

As the trend continues for companies to move online, it is more common to see that databases are moving online as well and that you must have a website and a good web browser so that the customer can come in and check them out. You can always add in usernames and passwords to make it more secure and to ensure that only the right user can gain access to their information. This is a great idea to help protect personal and payment information of your customers. Most companies will require that their users pick out security credentials to get on the account, but they will offer the account for free.

Of course, this is a system that is pretty easy to work with, but there will be a number of things going on behind the scenes to make sure that the program will work properly. The customer can simply go onto the system and check the information with ease, but there will be a lot of work for the server to do to make sure that the information is showing up on the screen in the right way, and to ensure that the user will have a good experience and actually see their own account information on the screen.

For example, you may be able to see that the web browser that you are using uses SQL or a program that is similar to it, to figure out the user that your data is hoping to see.

Why is SQL so great?

Now that we have spent some time talking about the various types of database management systems that you can work with, it is time to discuss why you would want to choose SQL over some of the other options that are out there. You not only have the option of working with other databases but also with other coding languages, and there are benefits to choosing each one. So, why would you want to work with SQL in particular? Some of the great benefits that you can get from using SQL as your database management system includes:

Incredibly fast

If you would like to pick out a management system that can sort through the information quickly and will get the results back in no time, then SQL is one of the best programs to use for this. Just give it a try, and you will be surprised at how much information you can get back, and how quickly it will come back to you. In fact, out of all the options, this is the most efficient one that you can go with.

Well defined standards

The database that comes with SQL is one that has been working well for a long time. In addition, it has been able to develop some good standards that ensure the database is strong and works the way that you want. Some of the other databases that you may want to work with will miss out on these standards, and this can be frustrating when you use them.

- You do not need a lot of coding

If you are looking into the SQL database, you do not need to be an expert in coding to get the work done. We will take a look at a few codes that can help, but even a beginner will get these down and do well when working in SQL.

- Keeps your stuff organized

When it comes to running your business, it is important that you can keep your information safe and secure as well as organized. And while there are a ton of great databases that you can go with, none will work as well as the SQL language at getting this all done.

- Object-oriented DBMS

The database of SQL relies on the DBMS system that we talked about earlier because this will make it easier to find the information that you are

searching for, to store the right items, and do so much more within the database.

These are just a few of the benefits that you can get when you choose to work with the SQL program. While some people do struggle with this interface in the beginning, but overall there are a ton of good features to work on with SQL, and you will really enjoy how fast and easy it is to work with this language and its database.

You may believe that SQL is an incomplete programming language. If you want to use SQL in an application, you must combine SQL with another procedural language like FORTRAN, Pascal, C, Visual Basic, C++, COBOL, or Java. SQL has some strengths and weaknesses because of how the language is structured. A procedural language that is structured differently will have different strengths and weaknesses. When you combine the two languages, you can overcome the weaknesses of both SQL and the procedural language.

You can build a powerful application when you combine SQL and a procedural language. This application will have a wide range of capabilities. We use an asterisk to indicate that we want to include all the columns in the table. If this table has many columns, you can save a lot of time by typing an asterisk. Do not use an asterisk when you are writing a program in a procedural language. Once you have written the application, you may want to add or delete a column from the table when it is no longer necessary. When you do this, you change the meaning of the asterisk. If you use the asterisk in the application, it may retrieve columns which it thinks it is getting.

This change will not affect the existing program until you need to recompile it to make some change or fix a bug. The effect of the asterisk wildcard will then expand to current columns. The application could stop working if it

cannot identify the bug during the debugging process. Therefore, when you build an application, refer to the column names explicitly in the application and avoid using the asterisk.

Since the replacement of paper files stored in a physical file cabinet, relational databases have given way to new ground. Relational database management systems, or RDBMS for short, are used anywhere information is stored or retrieved, like a login account for a website or articles on a blog. Speaking of which, this also gave a new platform to and helped leverage websites like Wikipedia, Facebook, Amazon, and eBay. Wikipedia, for instance, contains articles, links, and images, all of which are stored in a database behind-the-scene. Facebook holds much of the same type of information, and Amazon holds product information and payment methods, and even handles payment transactions.

With that in mind, banks also use databases for payment transactions and to manage the funds within someone's bank account. Other industries, like retail, use databases to store product information, inventory, sales transactions, price, and so much more. Medical offices use databases to store patient information, prescription medication, appointments, and other information.

To expand further, using the medical office for instance, a database gives permission for numerous users to connect to it at once and interact with its information. Since it uses a network to manage connections, virtually anyone with access to the database can access it from just about anywhere in the world.

These types of databases have also given way to new jobs and have even expanded the tasks and responsibilities of current jobs. Those who are in finance, for instance, now have the ability to run reports on financial data; those in sales can run reports for sales forecasts, and so much more!

In practical situations, databases are often used by multiple users at the same time. A database that can support many users at once has a high level of concurrency. In some situations, concurrency can lead to loss of data or the reading of non-existent data. SQL manages these situations by using transactions to control atomicity, consistency, isolation, and durability. These elements comprise the properties of transactions. A transaction is a sequence of T-SQL statements that combine logically and complete an operation that would otherwise introduce inconsistency to a database. Atomicity is a property that acts as a container for transaction statements. If the statement is successful, then the total transaction completes. If any part of a transaction is unable to process fully, then the entire operation fails, and all partial changes roll back to a prior state. Transactions take place once a row, or a page-wide lock is in place. Locking prevents modification of data from other users taking effect on the locked object. It is akin to reserving a spot within the database to make changes. If another user attempts to change data under lock, their process will fail, and an alert communicates that the object in question is barred and unavailable for modification. Transforming data using transactions allows a database to move from one consistent state to a new consistent state. It's critical to understand that transactions can modify more than one database at a time. Changing data in a primary key or foreign key field without simultaneously updating the other location, creates inconsistent data that SQL does not accept. Transactions are a big part of changing related data from multiple table sources all at once. Transactional transformation reinforces isolation, a property that prevents concurrent transactions from interfering with each other. If two simultaneous transactions take place at the same time, only one of them will be successful. Transactions are invisible until they are complete. Whichever transaction completes first will be accepted. The new information displays upon completion of the failed transaction, and at that point, the user must decide if

the updated information still requires modification. If there happened to be a power outage and the stability of the system fails, data durability would ensure that the effects of incomplete transactions rollback. If one transaction completes and another concurrent transaction fails to finish, the completed transaction is retained. Rollbacks are accomplished by the database engine using the transaction log to identify the previous state of data and match the data to an earlier point in time.

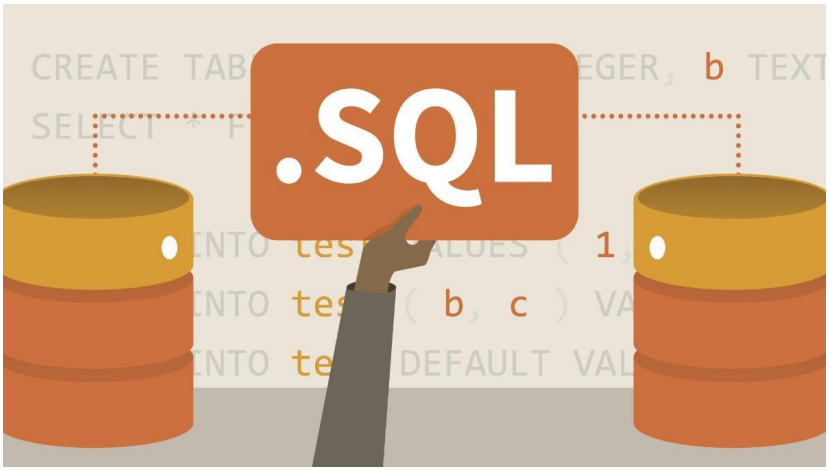
There are a few variations of a database lock, and various properties of locks as well. Lock properties include mode, granularity, and duration. The easiest to define is duration, which specifies a time interval where the lock is applied. Lock modes define different types of locking, and these modes are determined based on the type of resource being locked. A shared lock allows the data reads while the row or page lock is in effect. Exclusive locks are for performing data manipulation (DML), and they provide exclusive use of a row or page for the execution of data modification. Exclusive locks do not take place concurrently, as data is being actively modified; the page is then inaccessible to all other users regardless of permissions. Update locks are placed on a single object and allow for the data reads while the update lock is in place. They also allow the database engine to determine if an exclusive lock is necessary once a transaction that modifies an object is committed. This is only true if no other locks are active on the object in question at the time of the update lock. The update lock is the best of both worlds, allowing reading of data and DML transactions to take place at the same time until the actual update is committed to the row or table. These lock types describe page-level locking, but there are other types beyond the scope of this text. The final property of a lock, the granularity, specifies to what degree a resource is unavailable. Rows are the smallest object available for locking, leaving the rest of the database available for manipulations. Pages, indexes,

tables, extents, or the entire database are candidates for locking. An extent is a physical allocation of data, and the database engine will employ this lock if a table or index grows and more disk space is needed. Problems can arise from locks, such as lock escalation or deadlock, and we highly encourage readers to pursue a deeper understanding of how these function.

It is useful to mention that Oracle developed an extension for SQL that allows for procedural instruction using SQL syntax. This is called PL/SQL, and as we discussed at the beginning of the book, SQL on its own is unable to provide procedural instruction because it is a non-procedural language. The extension changes this and expands the capabilities of SQL. PL/SQL code is used to create and modify advanced SQL concepts such as functions, stored procedures, and triggers. Triggers allow SQL to perform specific operations when conditional instructions are defined. They are an advanced functionality of SQL, and often work in conjunction with logging or alerts to notify principals or administrators when errors occur. SQL lacks control structures, the for looping, branching, and decision making, which are available in programming languages such as Java. The Oracle corporation developed PL/SQL to meet the needs of their database product, which includes similar functionality to other database management systems, but is not limited to non-procedural operations. Previously, user-defined functions were mentioned but not defined. T-SQL does not adequately cover the creation of user-defined functions, but using programming, it is possible to create functions that fit neatly within the same scope as system-defined functions. A user-defined function (UDF) is a programming construct that accepts parameters, performs tasks capable of making use of system defined parameters, and returns results successfully. UDFs are tricky because Microsoft SQL allows for stored procedures that often can accomplish the same task as a user-defined

function. Stored procedures are a batch of SQL statements that are executed in multiple ways and contain centralized data access logic. Both of these features are important when working with SQL in production environments.

Chapter 2: Installing SQL developer



While almost all of the SQL queries presented here are general, it will eventually be easy for you to adjust to whatever type of SQL the server may use.

Before you can perform any SQL task on your computer, you first have to download SQL software.

Many options are available. You can utilize the free MySQL databases software. Hence, we will be focusing on how to download this application (“Chapter 5 Installing MySQL on Microsoft Windows”).

What Is MySQL?

MySQL is a tool (database server) that uses SQL syntax to manage databases. It is a Relational Database Management System (RDBMS) that you can use to facilitate the manipulation of your databases.

In the case where you are managing a website using MySQL, ascertain that the host of your website supports MySQL, too.

Here’s how you can install MySQL on Microsoft Windows. We will be demonstrating with Windows because it is the most common operating system used on computers.

How to Install MySQL on Microsoft Windows on Your Computer

Step 1 – Visit the MySQL Website

Go to https://dev.mysql.com/downloads/installer/ and browse through the applications to select MySQL. Make sure that you obtain the MySQL from its genuine website to prevent downloading viruses, which can be harmful to your computer.

Step 2 – Select the Download Option

Next, click on the Download option. This will bring you to the MySQL Community Server, and to the MySQL Community Edition. Click Download.

Step 3 – Choose Your Windows Processor Version

Choose your Windows processor version by reading through the details given on the page. Choose from the Other Downloads label. You can choose either the 32-bit or 64-bit.

Click the Download button for the Windows (x86, 32-bit), ZIP Archive or the Windows (x86, 64-bit), ZIP Archive, whichever is applicable to your computer.

Step 4 – Register on the Site

Before you can download your selected version, you will need to register by answering the sign in form for an Oracle account.

There is no need to reply to the questions that are optional. You can also click on the No Thanks button.

There is another option of just downloading the server without signing up, but you will not get to enjoy some freebies like the ability to download some white papers and technical information, faster access to MySQL downloads, and other services.

Step 5 – Sign in to Your MySQL Account

After registering, you can now sign in to your new account. A new page will appear where you can select your area through the displayed images of flags. Afterwards, you can click the download button and save it in your computer.

This could take several minutes.

Step 6 – Name the Downloaded File

After downloading the MySQL file, you can name it and save it on your Desktop or your C: drive, whichever you prefer.

Step 8 – Browse Your MySQL packages

The MySQL Enterprise Server page will appear, giving you some information about what your MySQL package contains.

There are packages offered for a small fee, but since we're just interested in the community server, just click Next until you reach the Finish button.

Step 9 – Uncheck the Box Register the MySQL Server Now

On set-up completion by the Wizard, a box that asks you to configure and register your MySQL server will appear. Uncheck the Register the MySQL Server Now box, and check the small box that says Configure the MySQL Server Now.

Then, click Finish.

Step 10 – Click Next on the Configuration Wizard Box

A new box will appear, and you just have to click Next.

Step 11 – Select the Type of Configuration

A box that asks you to select your configuration type will appear. Tick the small circle for the Detailed Configuration. Click the Next button.

Step 12 – Select the Server Type

You will encounter three choices: The Developer Machine, the Server Machine, and the Dedicated MySQL Server Machine.

Select the Server Machine because it has medium memory usage, which is ideal for someone who is interested in learning more about MySQL.

The Developer Machine uses minimal memory and may not allow you to maximize the usage of your MySQL.

On the other hand, the Dedicated MySQL Server Machine is for people who work as database programmers, or who are full-time MySQL users. It uses all of the available memory in your computer, so we don't recommend it here.

Step 13 – Select the Database Usage

For database usage, there are three choices: Multifunctional Database, Transactional Database Only, and Non-Transactional Database Only. Choose the Multifunctional Database because you are using this for more general purposes.

The Transactional and Non-Transactional are used for more specific purposes.

Click Next at the bottom of the display box.

Step 14 – Choose the Drive Where the InnoDB Datafile Will Be Located

You may want to consider avoiding the default settings. Instead, you can select which drive from your computer you want to store your InnoDB datafile on. Choose the drive you prefer and then click Next.

Step 15 – Set the Number of Concurrent Connections to the Server

This will indicate the number of users that will be connecting simultaneously to your server. The choices are: Discussion Support (DSS)/OLAP, Online Transaction Processing (OLTP), and Manual Setting.

You should select DSS/OLAP since there is no need for a high number of concurrent connections. OLTP is necessary for highly loaded servers, while the Manual Setting can be a hassle to set more than once.

After setting this, click Next.

Step 16 – Set the Networking Options

Enable TCP/IP Networking by checking off the small box before it. Below it, add your port number and then check the small box labeled Enable Strict Mode to set the server’s SQL mode.

Click Next.

Step 17 – Select Your Default Character Set

Select the Standard Character Set since it is the default for English, and works well with other related Western European languages.

The other two choices, Manual Default Character Set and Best Support for Multilingualism, are best for those who speak a language other than English.

Tick the small circle before the Standard Character Set and click Next.

Step 18 – Set the Windows Options

Tick the two choices displayed, which are Install as Windows Server, and Include Bin Directory in Windows Path. These will allow you to work with your MySQL from your command line.

Selecting Install as Windows Server will automatically display the Service Name. The small box below the Service Name must be checked, too.

Click Next.

Step 19 – Set the Security Options

Set your password. The box will indicate where you can type it.

Click Next.

Step 20 – Execute Your Configurations

All you have to do is click Execute and your computer will configure itself based on your specifications.

Once configuration is complete and all the boxes are checked, click Finish.

Step 21 – Set Verification Process

Go to the Start Menu, type cmd, and hit Enter. This will take you to the command panel.

Type the following and hit Enter:

mysql -u root -p

Notice that there is a space between MySQL and the dash symbol, between u and root, and between root and the dash symbol.

The command panel will ask for your password. Type your password and hit Enter.

A MySQL prompt will appear. You can type any SQL command to display the databases. Remember to add the semicolon at the end of your SQL statement. Close your command panel for now.

Using your MySQL can motivate you to learn more about other related applications, such as PHP and similar products.

However, it is important that you learn the basics of SQL first.

Chapter 3: Working with data, tables and columns

Your tables are used to store the data or information in your database. They are composed of rows and columns as discussed in chapter 1. Specific names are assigned to the tables to identify them properly and to facilitate their manipulation. The rows of the tables contain the information for the columns.

Knowing how to create tables is important for a beginner, who wants to learn SQL.

The following are the simple steps:

Step #1 – Enter the keywords CREATE TABLE

These keywords will express your intention and direct what action you have in mind.

Example: CREATE TABLE

Step #2 – Enter the table name

Right after your CREATE TABLE keywords, add the table name. The table name should be specific and unique to allow easy and quick access later on.

Example: CREATE TABLE “table_name”

The name of your table must not be easy to guess by anyone. You can do this by including your initials and your birthdate. If your name is Henry Sheldon, and your birthdate is October 20, 1964, you can add that information to the name of your table.

Let’s say you want your table to be about the traffic sources on your website, you can name the table “traffic_hs2064”

Take note that all SQL statements must end with a semicolon (;). All the data variables must be enclosed with quotation marks (“ “), as well.

Example: CREATE TABLE traffic_hs2064

Step #3 – Add an open parenthesis in the next line

The parenthesis will indicate the introduction of the columns you want to create.

Example: CREATE TABLE “table_name”

(

Let’s apply this step to our specific example.

Example: CREATE TABLE traffic_hs2064

(

In some instances, the parentheses are not used.

Step #4 – Add the first column name

This should be related to the data or information you want to collect for your table. Always separate your column definitions with a comma.

Example: CREATE TABLE "table_name"
("column_name" "data type",

In our example, the focus of the table is on the traffic sources of your website. Hence, you can name the first column "country".

Example: CREATE TABLE traffic_hs2064
(country

Step #4 – Add more columns based on your data

You can add more columns if you need more data about your table. It's up to you. So, if you want to add four more columns, this is how your SQL statement would appear.

Example: CREATE TABLE "table_name"
("column_name1" "data type",
"column_name2" "data type",
"column_name3" "data type",
"column_name4" "data type");

Add the closing parenthesis and the semi-colon after the SQL statement.

Let's say you have decided to add for column 2 the keyword used in searching for your website, for column 3, the number of minutes that the

visitor had spent on your website, and for column 4, the particular post that the person visited. This is how your SQL statement would appear.

Take note:

- The name of the table or column must start with a letter, then it can be followed by a number, an underscore, or another letter. It's preferable that the number of characters does not exceed 30.
- You can also use a VARCHAR (variable-length character) data type to help create the column.

- **Common data types are:**
  - **date** – date specified or value
  - **number (size) – you should specify the maximum number of column digits inside the open and close parentheses**
  - **char (size) – you should specify the size of the fixed length inside the open and close parentheses.**
  - **varchar (size) – you should specify the maximum size inside the open and close parentheses. This is for variable lengths of the entries.**
  - **Number (size, d) – This is similar to number (size), except that 'd' represents the maximum number of digits.**

    Hence if you want your column to show 10.21, your date type would be:

    number (2,2)

    Example: CREATE TABLE traffic_hs2064

(country varchar (40),

keywords varchar (30),

time number (3),

post varchar (40));

Step #5 – Add CONSTRAINTS, if any

CONSTRAINTS are rules that are applied for a particular column. You can add CONSTRAINTS, if you wish. The most common CONSTRAINTS are:

- **"NOT NULL"** – this indicates that the columns should not contain blanks
- **"UNIQUE"** – this indicates that all entries added must be unique and not similar to any item on that particular column.

In summary, creating a table using a SQL statement will start with the CREATE TABLE, then the "table name", then an open parenthesis, then the "column names", the "data type", (add a comma after every column), then add any "CONSTRAINTS".

Add the closing parenthesis and the semicolon at the end of your SQL statement.

Chapter 4: Data types in SQL

There are various data types that you should be familiar with. This is because they make use of SQL language that are significant in understanding SQL more.

There are six SQL data types

1. Date and Time Data

As the name implies, this type of data deals with date and time.

The date and time data type are of following types:

1. DATE: The format for DATE is YYYY-MM-DD. Where, DD is the date of the Month, MM is for month and YYYY is for the year.
2. DATETIME: This would display date and time in the format YYYY-MM-DD HH:MM:SS. Where YYYY is for 4 digits of the year, MM is for 2 digits of the month and DD is for two digits for the date of the month. HH is for hours, MM is for minutes and SS is for seconds.
3. TIMESTAMP: Time stamp displays date and time in DATETIME format but without the hyphens in between. So, the date and time is displayed as YYYYMMDDHHMMSS.
4. TIME: save the time in HH:MM:SS format.

Examples are: datetime (FROM Feb 1, 1816 TO July 2, 8796), smalldatetime (FROM Feb 1, 2012 TO Mar 2085, date (Jun 1, 2016) and time (3:20 AM.).

2. Exact Numeric Data

There are 7 forms of numeric data types listed below. Make a list and keep it handy! You will be referring to this list often while programming:

3. Integer (INT) : covers the range of all positive and negative integers**. It can take up to 11 digits.**
4. Very Small Integer (TINYINT) : very small integer that can be positive or negative, and within the range from -128 to 127. It can take up to 4 digits.
5. Small Integer (SMALLINT): used for small positive or negative integers within the range of -32768 to 32767. It can take up to 5 digits.
6. Medium Sized Integer (MEDIUMINT): Positive or negative integers up to 9 digits.
7. Large Integer (BIGINT): Positive or negative integers up to 20 digits.
8. Floating-point number (FLOAT(N,D)): This category is used to define decimal numbers. You will have to manually define the length of the number N and the length of the decimal D. Unless specified otherwise, the default value is N =10, D = 2. (10 being the length of the number before decimal, and 2 being the length of numbers after the decimal point.
9. Double (DOUBLE(M,D)): A floating point number with the default value of N =16, D = 4.

3. Binary Data

Binary data have different types, as well. These are: Binary (fixed), varbinary (variable length binary) varbinary (max) (variable length binary) and image.

They are classified according to the length of their bytes, with Binary having the shortest and the fixed value.

4. Approximate Numeric Data

These have two types, the float and the real. The float has a value FROM - 1.79E +308 TO 1.79E +308, while the real data has a value FROM -3.40E +38 TO 3.40E +38

5. Unicode Character Strings Data

There are four types of Unicode Character Strings Data namely; ntext, nchar, nvarchar, and nvarchar (max). They are classified according to their character lengths.

For ntext, it has a maximum character length of 1,073,741,823, which is variable.

For nchar, it has a unicode maximum fixed length of 4,000 characters.

For nvarchar (max), it has a unicode variable maximum length of 231 characters.

For nvarchar, it has a variable maximum length of 4,000 unicode characters.

6. Character Strings Data

A string is a data type which is used to represent text. All characters in string data type are stored as text. It can also contain text and numbers. The string data types are as follows:

1. CHAR(L): This data type is used to define string characters with the fixed length of 1 to 255 characters. It is not necessary to define the length and the default value is 1.
2. VARCHAR(L): This data type is used to define string data of variable length between 1 to 255 characters. You must define the length of varchar data type, with length = L.
3. BLOB: used to store large amount of binary data such as images or any other file. It can have maximum length of 65535 characters and is not case sensitive. There is no need to specify length.
4. TEXT: is like BLOB, except that TEXT is case sensitive.
5. TINYBLOB: is same as BLOB with maximum length of 255 characters.
6. TINYTEXT: is same as TEXT with maximum length of 255 characters.
7. MEDIUMBLOB: same as BLOB with maximum length of 16777215 characters.
8. MEDIUMTEXT: same as TEXT with maximum length of 16777215 characters.
9. MEDIUMBLOB: same as BLOB with maximum length of 16777215 characters.
10. MEDIUMTEXT: same as TEXT with maximum length of 16777215 characters.
11. LONGBLOB: same as BLOB with maximum length of 4294967295 characters.

12. LONGTEXT: same as TEXT with maximum length of 4294967295 characters.
13. ENUM: Enumeration is for lists, used to create a list of string objects.

While entering records in database, there can be fields for which there may be no values, and there can be fields for which it is mandatory to have values. All people may not have a middle name for example, so in that case we can allow “middle_name” to contain blank values whereas “first_name” and “last_name” should be mandatory. When it is not mandatory to have a value in a field, it is defined as NULL. When it is mandatory to have a field value, it is defined as NOT NULL.

The character Strings Data have almost similar types as the Unicode Character Strings Data, only, some have different maximum values and they are non-unicode characters, as well.

For text, it has a maximum variable length of 2,147,483,647 non-unicode characters.

For char, it has a non-unicode maximum fixed length of 8,000 characters.

For varchar (max), it has a non-unicode variable maximum length of 231 characters.

For varchar, it has a variable maximum length of 8,000 non-unicode characters.

Miscellaneous Data

Aside from the 6 major types of data, miscellaneous data are also stored as tables, SQL variants, cursors, XML files, unique identifiers, cursors and/or timestamps.

You can refer to this chapter when you want to know about the maximum values of the data you are preparing.

It helps to have some information on these values.

Chapter 5: Ensuring data integrity

SQL database must do more than just store data. It must ensure that the data it stores is correct. If the integrity of the data is compromised, the data might be inaccurate or inconsistent, bringing into question the reliability of the database itself. In order to ensure the integrity of the data, SQL provides a number of integrity constraints, rules that are applied to base tables to constrain the values that can be placed into those tables. You can apply constraints to individual columns, to individual tables, or to multiple tables. In this chapter, I discuss each type of constraint and explain how you can apply them to your SQL database.

Understand Integrity Constraints

SQL integrity constraints, which are usually referred to simply as constraints, can be divided into three categories:

- **Table-related constraints**: A type of constraint that is defined within a table definition. The constraint can be defined as part of the column definition or as an element in the table definition. Constraints defined at the table level can apply to one or more columns.
- **Assertions**: A type of constraint that is defined within an assertion definition (separate from the table definition). An assertion can be related to

one or more tables.

- **Domain constraints**: A type of constraint that is defined within a domain definition (separate from the table definition). A domain constraint is associated with any column that is defined within the specific domain.

Of these three categories of constraints, table-related constraints are the most common and include the greatest number of constraint options. Table-related constraints can be divided into two subcategories: table constraints and column constraints. The constraints in both these subcategories are defined in the table definition. A column constraint is included with the column definition, and a table constraint is included as a table element, similar to the way columns are defined as table elements. (Chapter 3 discusses table elements and column definitions.) Both column constraints and table constraints support a number of different types of constraints. This is not the case for assertions and domain constraints, which are limited to only one type of constraint. Figure 4-1 provides an overview of the types of constraints that can be created.

At the top of the illustration, you can see the three categories of constraints. Beneath the Table-Related Constraints category are the Column Constraints subcategory and the Table Constraints subcategory, each of which contains specific types of constraints. For example, table constraints can include unique (UNIQUE constraints and PRIMARY KEY constraints), referential (FOREIGN KEY constraints), and CHECK constraints, while column constraints can include the NOT NULL constraint as well as unique, referential, and CHECK constraints. However, domains and assertions support only CHECK constraints.

NOTE: In some places, the SQL:2006 standard uses the term “table constraint” to refer to both types of table-related constraints. I use the term “table-related” to avoid confusion.

Use NOT NULL Constraints

Null signifies that a value is undefined or not known. This is not the same as a zero, a blank, an empty string, or a default value. Instead, it indicates that a data value is absent. You can think of a null value as being a flag. (A flag is a character, number, or bit that indicates a certain fact about a column. The flag serves as a marker that designates a particular condition or existence of something.) In the case of null, if no value is provided for a column, the flag is set, indicating that the value is unknown, or null. Every column has a nullability characteristic that indicates whether the column will accept null values. By default, all columns accept null values.

NOTE: Some RDBMSs allow you to change the default nullability of any new column you create. In addition, some systems support a NULL constraint, which you can use to designate that a column will accept null values.

The NOT NULL constraint will work just as a column constraint. It is not supported for table constraints, assertions, or domain constraints. Implementing a NOT NULL constraint is a very straightforward process. Simply use the following syntax when creating a column definition:

< column name> { <data type> | <domain> } NOT NULL

```
CREATE TABLE COMPACT_DISC_ARTISTS
(  ARTIST_ID       INT         NOT NULL,
  ARTIST_NAME     VARCHAR(60) NOT NULL,
  PLACE_OF_BIRTH  VARCHAR(60) );
```

Notice that the PLACE_OF_BIRTH column does not include a NOT NULL constraint. As a result, if a value isn't supplied for this column (when a row is inserted), a null value will be inserted. (The null flag will be set.) Figure 4-2 shows how the table might look if rows were inserted that contained no value for the PLACE_OF_BIRTH column.

As you can see, the ARTIST_ID and ARTIST_NAME columns do not—, and cannot, —contain null values. The PLACE_OF_BIRTH column, on the other hand, contains two null values.

Add UNIQUE Constraints

The UNIQUE constraint allows you to require that a column or set of columns contains unique values, meaning values that are different from all other rows in the same table. For example, take a look at Figure 4-3, which shows the CD_INVENTORY table. The table contains three columns: ARTIST_NAME, CD_NAME, and COPYRIGHT.

You might decide that you want the values in the CD_NAME column to be unique so that no two CD names can be alike. If you applied a UNIQUE constraint to the column, you would not be able to insert a row that contained a CD_NAME value that already existed in the table. Now suppose that you realize that making the CD_NAME values unique is not a good idea because it is possible for more than one CD to share the same name. You decide to take another approach and use a UNIQUE constraint on the ARTIST_NAME and CD_NAME columns. That way, no ARTIST_NAME/CD_NAME pair can be repeated. You can repeat an ARTIST_NAME value or a CD_NAME value, but you cannot repeat the exact same combination of the two. For example, the table already contains a row with an ARTIST_ NAME value of Joni Mitchell and a CD_NAME

value of Blue. If a UNIQUE constraint had been applied to these two columns, you could not add another row that contained both of these values.

NOTE: I should point out that the tables used for illustration of concepts in this chapter are not necessarily good designs. For example, names of people and things are seldom good choices to uniquely identify rows of data because they are rather long (compared to numbers), tend to change, and are prone to problems with duplicate values. However, these tables were chosen because they illustrate the concepts well.

Now that you have a basic understanding of how UNIQUE constraints are applied, let's take a look at the syntax that you use to create them. Remember, I said that you can create a UNIQUE constraint that is either a column constraint or a table constraint. To create a column constraint, add it as part of the column definition, as shown in the following syntax:

< column name> { <data type> | <domain> } UNIQUE

If you want to add a unique constraint as a table constraint, you must add it as a table element in the table definition, as shown in the following syntax:

[CONSTRAINT <constraint name>]

UNIQUE (<column name> [{, <column name> } . . .])

As you can see, applying a UNIQUE constraint as a column constraint is a little simpler than applying it as a table constraint. However, if you apply the constraint at the column level, you can apply it to only one column. Regardless of whether you use column constraints or table constraints, you can define as many UNIQUE constraints as necessary in a single table definition.

Now let’s return to the table in Figure 4-3 and use it to create code examples for applying UNIQUE constraints.

The ARTIST_NAME column and CD_NAME column must now contain unique combinations of values in order for a row to be added to the CD_INVENTORY table.

Until now, I have told you that a UNIQUE constraint prevents duplicate values from being entered into a column or columns defined with that constraint. However, there is one exception to this—the null value. A UNIQUE constraint permits multiple null values in a column. As with other columns, null values are permitted by default. You can, however, override the default by using the NOT NULL constraint in conjunction with the UNIQUE constraint. For example, you can add NOT NULL to the CD_NAME column definition:

```
CREATE TABLE CD_INVENTORY
( ARTIST_NAME  VARCHAR(40),
  CD_NAME      VARCHAR(60)  NOT NULL  UNIQUE,
  COPYRIGHT    INT );
```

You can also add NOT NULL to a column definition that’s referenced by a table constraint:

```
CREATE TABLE CD_INVENTORY
( ARTIST_NAME  VARCHAR(40),
  CD_NAME      VARCHAR(60)  NOT NULL,
  COPYRIGHT    INT,
  CONSTRAINT UN_ARTIST_CD UNIQUE (CD_NAME) );
```

In each case, both the NOT NULL constraint and the UNIQUE constraint are applied to the CD_NAME column, which means the CD_NAME values

must be unique and without null values.

Add PRIMARY KEY Constraints

As I mentioned in the “Add UNIQUE Constraints” section, a PRIMARY KEY constraint, like the UNIQUE constraint, is a type of SQL unique constraint. Both types of constraints permit only unique values in the specified columns, both types can be applied to one or more columns, and both types can be defined as either column constraints or table constraints.

The reason for these restrictions is the role that a primary key (unique identifier) plays in a table. As you might recall from Chapter 1, each row in a table must be unique. This is important because SQL cannot differentiate between two rows that are completely identical, so you cannot update or delete one duplicate row without doing the same to the other. The primary key for a table is chosen by the database designer from available candidate keys. A candidate key is a set of one or more columns that uniquely identify each row.

KEY constraint is very similar to that of defining a UNIQUE constraint. If you want to add a PRIMARY KEY constraint to a column definition, use the following syntax:

< column name> { <data type> | <domain> } PRIMARY KEY

If you want to add a PRIMARY KEY constraint as a table constraint, you must add it as a table element in the table definition, as shown in the following syntax:

[CONSTRAINT <constraint name>]

PRIMARY KEY (<column name> [{, <column name> } . . .])

As with the UNIQUE constraint, you can use a column constraint to define a primary key if you’re including only one column in the definition.

This method creates a primary key on the ARTIST_ID and ARTIST_NAME columns, so that the combined values of both columns must be unique, although duplicates can exist within the individual column. An experienced database designer will quickly point out to you that this is a superkey, which means that it has more columns in it than the minimum needed to form a primary key. And that is true—ARTIST_ID by itself is unique and we really don't need to add ARTIST_NAME to it in order to form a primary key, and we want to be sure that duplicate values of ARTIST_ID are not entered into the table, which means we should have a primary key with only ARTIST_ID in it. It was only done here to illustrate that a primary key can contain multiple columns and to define one that way, a table constraint must be used.

Ask the Expert

Q: Can the columns in a table belong to both a UNIQUE constraint and a PRIMARY KEY constraint?

A: Yes, as long as they're not the exact same columns. For example, suppose you have a table that includes three columns: ARTIST_ID, ARTIST_NAME, and PLACE_OF_BIRTH. You can define a PRIMARY KEY constraint that includes the ARTIST_ID and ARTIST_ NAME columns, which would ensure unique value pairs in those two columns, but values within the individual columns could still be duplicated. However, you can then define a UNIQUE constraint that includes only the ARTIST_NAME column to ensure that those values are unique as well. (This certainly isn't the best design, but it illustrates my point.) You can also create a UNIQUE constraint that includes the ARTIST_NAME and PLACE_ OF_BIRTH columns to ensure unique value pairs in those two columns. The only thing you can't do is create a UNIQUE constraint that includes the exact same columns as an existing PRIMARY KEY constraint and vice versa.

Q: You state that a column that is included in a PRIMARY KEY constraint will not accept null values. What happens if that column is configured with a NOT NULL constraint as well?

A: Nothing different happens. The table is still created in the same way. A column definition that includes PRIMARY KEY is saying the same thing as a column definition that includes NOT NULL PRIMARY KEY. In fact, prior to SQL-92, the NOT NULL keywords were required on all columns included in a PRIMARY KEY constraint.

The same was true for UNIQUE constraints. It wasn't until SQL-92 that null values were permitted in columns included in a UNIQUE constraint, which clearly set them apart from PRIMARY KEY constraints. Also, be wary of variations across vendor implementations. For example, Oracle will automatically add NOT NULL constraints to columns included in a PRIMARY KEY constraint, while SQL Server (or at least some versions of it) will display an error if you attempt to create a PRIMARY KEY constraint using columns that have not been specified with a NOT NULL constraint.

Add FOREIGN KEY Constraints

Up to this point, the types of constraints that I've discussed have had to do primarily with ensuring the integrity of data within a table. The NOT NULL constraint prevents the use of null values within a column, and the UNIQUE and PRIMARY KEY constraints ensure the uniqueness of values within a column or set of columns. However, the FOREIGN KEY constraint is different in that it is concerned with how data in one table relates to data in another table, which is why it is known as a referential constraint—it references another table. (Actually, there is an exception called a recursive relationship where the foreign key refers to another row in the same table, but I'm going to ignore this special case for now in order to focus on the basics.)

You might recall from Chapter 1 that tables in a relational database are linked together in a meaningful way in order to ensure the integrity of the data. This association between tables forms a relationship that provides referential integrity between tables. Referential integrity prevents the manipulation of data in one table from adversely affecting data in another table.

The CD_TITLE_ID column in the CD_TITLES table is configured with a PRIMARY KEY constraint, as is the PUBLISHER_ID column in the CD_PUBLISHERS table. Both these columns are shaded in the illustration.

Notice that the CD_TITLES table contains a column named PUBLISHER_ID. This column includes values from the PUBLISHER_ID column of the CD_PUBLISHERS table. In fact, the PUBLISHER_ID values in the CD_TITLES table should include only values that come from the PUBLISHER_ID column in the CD_PUBLISHERS table. You should not be able to insert a row into CD_TITLES if the PUBLISHER_ID value is not listed in the CD_PUBLISHERS table. At the same time, if you alter or delete a PUBLISHER_ID value in the CD_PUBLISHERS table, you should be able to predict the outcome of your action if those same values exist in the CD_TITLES table. Under no circumstances would you want to delete a publisher and leave PUBLISHER_ID values in the CD_TITLES table that reference a publisher that no longer exists. These results can be achieved by using a FOREIGN KEY constraint. A FOREIGN KEY constraint enforces referential integrity between two tables by ensuring that no action is taken on either table that adversely affects the data protected by the constraint.

In the tables shown in Figure 4-5, the FOREIGN KEY constraint must be configured on the PUBLISHER_ID column of the CD_TITLES table. The

FOREIGN KEY constraint restricts the values in that column to the values of a candidate key (usually the primary key) in the related table. Only valid data values are permitted in the FOREIGN KEY column or columns.

NOTE: The table that contains the foreign key is the referencing table. The table that is being referenced by the foreign key is the referenced table. Likewise, the column or columns that make up the foreign key in the referencing table are referred to as the referencing columns. The columns being referenced by the foreign key are the referenced columns.

When creating a FOREIGN KEY constraint, you must follow several guidelines:

- The referenced columns must be defined with either a UNIQUE or PRIMARY KEY constraint on them. As you might guess, the primary key is most commonly used for the referenced columns.
- A FOREIGN KEY constraint can be created as a table constraint or column constraint. If you create the foreign key as a column constraint, you can include only one column. If you create the foreign key as a table constraint, you can include one or more columns.
- The foreign key in the referencing table must include the same number of columns that are being referenced, and the referencing columns must each be configured with the same data types as their referenced counterparts. However, the referencing columns do not have to have the same names as the referenced columns.
- If you don’t specify the referenced columns when you define a FOREIGN KEY constraint, then the columns defined in the primary key of the referenced table are used as the referenced columns.

These guidelines will become clearer as I explain how to implement a foreign key.

First, let's take a look at the basic syntax used to create that constraint. If you want to add a FOREIGN KEY constraint as a column constraint, you must add the constraint to a column definition, as shown in the following syntax:

< column name> { <data type> | <domain> } [NOT NULL]

REFERENCES <referenced table> [(<referenced columns>)]

[MATCH { FULL | PARTIAL | SIMPLE }]

[<referential triggered action>]

If you want to add a FOREIGN KEY constraint as a table constraint, you must add it as a table element in the table definition, as shown in the following syntax:

[CONSTRAINT <constraint name>]

FOREIGN KEY (<referencing column > [{, <referencing column> } . . .])

REFERENCES <referenced table> [(<referenced columns>)]

[MATCH { FULL | PARTIAL | SIMPLE }]

[<referential triggered action>]

```
CREATE TABLE CD_TITLES
( CD_TITLE_ID  INT,
  CD_TITLE     VARCHAR(60),
  PUBLISHER_ID INT REFERENCES CD_PUBLISHERS );
```

This statement defines a FOREIGN KEY constraint on the PUBLISHER_ID column. Notice that, in order to add a column constraint, all you have to do

is add the REFERENCES keyword and the name of the referenced table. Also notice that the foreign key contains the same number of columns as the primary key in the referenced table, and the referenced and referencing columns are the same data type. Remember, if you're not referencing the primary key in the referenced table, you must also include the name of the column or columns—for example, REFERENCES CD_PUBLISHERS (PUBLISHER_ID).

NOTE: Before you can create a foreign key on a table, the referenced table must already exist and a UNIQUE or PRIMARY KEY constraint must be defined for that table.

In the next example, I create a foreign key that is a table constraint. Unlike the previous example, I include the name of the referenced column in this constraint definition, even though it isn't necessary:

```
CREATE TABLE CD_TITLES
( CD_TITLE_ID   INT,
  CD_TITLE      VARCHAR(60),
  PUBLISHER_ID  INT,
  CONSTRAINT FK_PUBLISHER_ID FOREIGN KEY (PUBLISHER_ID)
    REFERENCES CD_PUBLISHERS (PUBLISHER_ID) );
```

The last two lines of code are the constraint definition. The name of the constraint, FK_PUBLISHER_ID, follows the CONSTRAINT keyword. Constraint names aren't necessary because the DBMS will assign a system-generated name if one is not supplied. However, it's a good practice to supply your own because constraint names often appear in error messages when SQL statements attempt to violate constraints, and names you supply will be easier to recognize than ones the DBMS supplied for you. Following

the constraint name, the FOREIGN KEY keywords indicate the type of constraint, which is followed by the referencing column name, PUBLISHER_ID. This is the name of the column on which the constraint is being placed. If there were multiple column names, they would be separated by commas. The name of the referencing column is then followed by the REFERENCES keyword, which is followed by the name of the referenced table, CD_PUBLISHERS. The name of the referenced column follows the name of the referenced table.

That's all there is to it. Once the constraint is defined, you would not be able to place values in the PUBLISHER_ID column of the CD_TITLES table unless those values already existed in the primary key of the CD_PUBLISHERS table. You should note, however, that the values in the foreign key do not have to be unique, as they must be in the CD_PUBLISHERS primary key. Values in the foreign key can be repeated any number of times, unless the column is limited by a unique constraint.

```
CREATE TABLE ARTISTS_MUSIC_TYPES
( ARTIST_NAME  VARCHAR(60),
  DOB          DATE,
  TYPE_ID      INT,
   CONSTRAINT FK_CD_ARTISTS FOREIGN KEY ( ARTIST_NAME, DOB )
              REFERENCES PERFORMING_ARTISTS (ARTIST_NAME, ARTIST_DOB) );
```

In this statement, there are two referencing columns (ARTIST_NAME and DOB) and two referenced columns (ARTIST_NAME, ARTIST_DOB).The ARTIST_NAME columns in the two tables have the same data type, and the DOB column has the same data type as the ARTIST_DOB column. As you can see, one of the referencing columns (DOB) has a different name than its referenced counterpart (ARTIST_DOB).

Ask the Expert

Q: In Figure 4-6 and in the preceding examples, you created a FOREIGN KEY constraint on the ARTIST_NAME and DOB columns in the ARTISTS_MUSIC_ TYPES table. What would the primary key be for this table?

A: Remember that a primary key must uniquely identify each row in a table. However, because pairs of values in the ARTIST_NAME and DOB columns can be repeated (which means that they can be repeated in the individual columns as well), those two columns cannot be used by themselves as a primary key for this table. On the other hand, the TYPE_ID column can have repeating values as well, so that column cannot be used by itself.

In addition, you probably wouldn’t want to combine the TYPE_ID column with one of the other two columns because it is conceivable that you would have repeating rows (for example, two artists with the same name performing the same types of music, such as the two blues musicians named Sonny Boy Williamson, or two artists with the same date of birth performing the same type of music). As a result, your best solution (aside from adding another column to the table) is to roll all three columns into the primary key. Together, the three columns would uniquely identify each row because it is highly unlikely that anyone would share the same name, date of birth, and type of music (although anything is possible, which is why, ultimately,

adding another column that is guaranteed to be unique is the very best way to go).

The MATCH Clause

Now that you have an understanding of how to define a basic FOREIGN KEY constraint, let's look at another line of the FOREIGN KEY syntax:

[MATCH { FULL | PARTIAL | SIMPLE }]

You can tell from the brackets that this is an optional clause. And in fact, very few vendor products currently support this clause (it's not supported by SQL Server 2005, Oracle 11g, or MySQL 5.0, for example), so you won't see it used much at all. However, it is described in the SQL Standard, which means we can expect more vendor product support in the future. Its purpose is to allow you to decide how to treat null values in the foreign key columns, with regard to permitting values to be inserted into the referencing columns. If the columns do not permit null values, then the MATCH clause does not apply. You have three options that you can use in the MATCH clause:

- If MATCH FULL is specified, all referencing columns must have a null value or none of these columns can have a null value.
- If MATCH PARTIAL is specified, one or more referencing columns can have null values as long as the remaining referencing columns have values that equal their corresponding referenced columns.
- If MATCH SIMPLE is specified and one or more referencing columns have null values, then the remaining referencing columns can have values that are not contained in the corresponding referenced columns. The SIMPLE option is implied if the MATCH clause is not included in the FOREIGN KEY constraint definition.

NOTE: You probably wouldn't want to permit null values in your referencing columns in the ARTISTS_MUSIC_TYPES table, particularly for the ARTIST_NAME column. And if either of these columns were used in the primary key, you would not be able to permit null values. However, in order to demonstrate how the MATCH options work, let's assume that null values are permitted.

If you decide to use the MATCH clause, you simply add it to the end of your FOREIGN KEY constraint definition, as shown in the following SQL statement (assuming your implementation of SQL supports it):

```
CREATE TABLE ARTISTS_MUSIC_TYPES
( ARTIST_NAME  VARCHAR(60),
  DOB          DATE,
  TYPE_ID      INT,
   CONSTRAINT FK_CD_ARTISTS FOREIGN KEY ( ARTIST_NAME, DOB )
     REFERENCES PERFORMING_ARTISTS MATCH FULL );
```

To insert data into the referencing columns (ARTIST_NAME and DOB), both values have to be null or they must be valid data values from the referenced columns in the PERFORMING_ ARTISTS table.

The <referential triggered action> Clause

The final clause in the FOREIGN KEY constraint syntax is the optional <referential triggered action> clause. The clause allows you to define what types of actions should be taken when attempting to update or delete data from the referenced columns—if that attempt would cause a violation of the data in the referencing columns. For example, suppose you try to delete data

from a table's primary key. If that primary key is referenced by a foreign key and if the data to be deleted is stored in the foreign key, then deleting the data from the primary key would cause a violation of the FOREIGN KEY constraint. Data in referencing columns must always be included in the referenced columns.

The point to remember about the <referential triggered action> clause is that you are including in the definition of the referencing table (through the foreign key) an action that should be taken as a result of something being done to the referenced table.

NOTE: The::= symbol (two consecutive colons plus an equals sign) is used in the SQL:2006 standard to separate a placeholder in the angle brackets from its definition. In the preceding syntax, the <referential action> placeholder is defined. The placeholder is used in the code preceding the definition. You would then take the definition (the five keywords) and use them in place of the <referential action> placeholder as it is used in the ON UPDATE and ON DELETE clauses.

As you can see from the syntax, you can define an ON UPDATE clause, an ON DELETE clause, or both, and you can define them in any order. For each of these clauses you can choose one of five referential actions:

- If CASCADE is used and data is updated or deleted in the referenced columns, the data in the referencing columns is updated or deleted.
- If SET NULL is used and data is updated or deleted in the referenced columns, the values in the corresponding referencing columns are set to null. Null values have to be supported in the referencing columns for this option to work.

- If SET DEFAULT is used and data is updated or deleted in the referenced columns, the values in the corresponding referencing columns are set to their default values. Default values must be assigned to the referencing columns for this option to work.
- If RESTRICT is used and you try to update or delete data in your referenced columns that would cause a foreign key violation, you are prevented from performing that action. Data in the referencing columns can never violate the FOREIGN KEY constraint, not even temporarily.
- If NO ACTION is used and you try to update or delete data in your referenced columns that would cause a foreign key violation, you are prevented from performing that action. However, data violations can occur temporarily under certain conditions during the execution of an SQL statement, but the data in the foreign key is never violated in its final state (at the end of that execution). The NO ACTION option is the default used for both updates and deletes, if no referential triggered action is specified.

If you decide to use the <referential triggered action> clause, you simply add it to the end of your FOREIGN KEY constraint definition, as shown in the following SQL statement:

```
CREATE TABLE ARTISTS_MUSIC_TYPES
( ARTIST_NAME  VARCHAR(60),
  DOB          DATE,
  TYPE_ID      INT,
   CONSTRAINT FK_CD_ARTISTS FOREIGN KEY ( ARTIST_NAME, DOB )
      REFERENCES PERFORMING_ARTISTS ON UPDATE CASCADE ON DELETE CASCADE );
```

If you update data in or delete data from the referenced columns in PERFORMING_ ARTISTS, those changes will be made to the referencing columns in the ARTISTS_MUSIC_ TYPES table.

The data model incorporates a few more elements than you have seen before. It identifies tables, columns within those tables, data types for those columns, constraints, and relationships between tables. You should already be familiar with how tables, columns, and data types are represented, so let's take a look at constraints and relationships:

- The columns included in the primary key are in the top section of the table, and the other columns lie in the bottom section. For example, in the COMPACT_DISCS table, the COMPACT_DISC_ID column is the primary key. In some cases, as in the COMPACT_ DISC_TYPES table, all columns are included in the primary key.
- Each foreign key is represented by an [FK].
- Defaults, UNIQUE constraints, and NOT NULL constraints are identified with each applicable column.
- Relationships, as defined by foreign keys, are represented by lines that connect the foreign key in one table to the candidate key (usually the primary key) in another table.

You'll find this data model useful not only for this exercise, but for other Try This exercises in the book, all of which will continue to build upon or use the INVENTORY database.

NOTE: Data models come in many varieties. The model I use here is specific to the needs of the book. You'll find in the real world that the models will differ from what you see here. For example, relationships between tables might be represented differently, and column definition information might not be quite as extensive.

Step by Step

1. Open the client application for your RDBMS and connect to the INVENTORY database.

2. You first need to drop the four tables (COMPACT_DISCS, COMPACT_DISC_TYPES, MUSIC_TYPES, and CD_LABELS) that you already created. Enter and execute the following SQL statements:

```
DROP TABLE COMPACT_DISCS      CASCADE;

DROP TABLE COMPACT_DISC_TYPES CASCADE;

DROP TABLE MUSIC_TYPES        CASCADE;

DROP TABLE CD_LABELS          CASCADE;
```

NOTE: If you created either the ARTISTS table or the ARTIST_CDS table when trying out examples or experimenting with CREATE TABLE statements, be sure to drop those as well.

NOTE: The CASCADE option is not supported by SQL Server, and in Oracle must be written as CASCADE CONSTRAINTS.

Now you can begin to recreate these tables and create new ones. You should create the tables in the order outlined in this exercise because the tables referenced in foreign keys will have to exist—with primary keys created—before you can create the foreign keys. Be sure to refer to the data model in Figure 4-7 for details about each table that you create.

3. The first table that you're going to create is the MUSIC_TYPES table. It contains two columns: TYPE_ID and TYPE_NAME. You'll configure the

TYPE_ID column as the primary key, and you'll configure a UNIQUE constraint and NOT NULL constraint on the TYPE_NAME column. Enter and execute the following SQL statement:

```
CREATE TABLE MUSIC_TYPES
( TYPE_ID    INT,
  TYPE_NAME  VARCHAR(20)  NOT NULL,
  CONSTRAINT UN_TYPE_NAME UNIQUE (TYPE_NAME),
  CONSTRAINT PK_MUSIC_TYPES PRIMARY KEY (TYPE_ID) );
```

4. The next table that you'll create is the CD_LABELS table. The table includes the LABEL_ ID column, which will be defined as the primary key, and the COMPANY_NAME column, which will be defined with a default and the NOT NULL constraint. Enter and execute the following SQL statement:

```
CREATE TABLE CD_LABELS
( LABEL_ID      INT,
   COMPANY_NAME  VARCHAR(60)  DEFAULT 'Independent'  NOT NULL,
  CONSTRAINT PK_CD_LABELS PRIMARY KEY (LABEL_ID) );
```

5. Now that you've created the CD_LABELS table, you can create the COMPACT_ DISCS table. The COMPACT_DISCS table contains a foreign key that references the CD_LABELS table. This is why you created CD_LABELS first. Enter and execute the following SQL statement:

```
CREATE TABLE COMPACT_DISCS
( COMPACT_DISC_ID  INT,
  CD_TITLE         VARCHAR(60)  NOT NULL,
```

LABEL_ID INT NOT NULL,

CONSTRAINT PK_COMPACT_DISCS PRIMARY KEY (COMPACT_DISC_ID),

CONSTRAINT FK_LABEL_ID FOREIGN KEY (LABEL_ID) REFERENCES CD_LABELS);

6. The next table, COMPACT_DISC_TYPES, includes two foreign keys, along with its primary key. The foreign keys reference the COMPACT_DISCS table and the MUSIC_TYPES table, both of which you've already created. Enter and execute the following SQL statement:

```
CREATE TABLE COMPACT_DISC_TYPES
( COMPACT_DISC_ID INT,
  MUSIC_TYPE_ID   INT,
  CONSTRAINT PK_COMPACT_DISC_TYPES
    PRIMARY KEY ( COMPACT_DISC_ID, MUSIC_TYPE_ID),
  CONSTRAINT FK_COMPACT_DISC_ID_01
        FOREIGN KEY (COMPACT_DISC_ID) REFERENCES COMPACT_DISCS,
  CONSTRAINT FK_MUSIC_TYPE_ID
    FOREIGN KEY (MUSIC_TYPE_ID) REFERENCES MUSIC_TYPES );
```

7. Now you can create the ARTISTS table. Enter and execute the following SQL statement:

8. The last table you'll create (at least for now) is the ARTIST_CDS table.

9. Close the client application.

Try This Summary

Your database now has six tables, each one configured with the necessary defaults and constraints. In this Try This exercise, we followed a specific order for creating the tables in order to more easily implement the foreign keys. However, you could have created the tables in any order, without their foreign keys—unless the referenced table was already created—and then added in the foreign keys later, but this would have added extra steps. In fact, had you wanted to, you could have altered the tables that had existed prior to this exercise (rather than dropping them and then recreating them), as long as you created primary keys (or UNIQUE constraints) on the referenced tables before creating foreign keys on the referencing tables. Regardless of the approach you take, the end result should be that your database now has the necessary tables to begin moving on to other components of SQL.

Define CHECK Constraints

Earlier in the chapter, in the "Understand Integrity Constraints" section, I discussed the various constraint categories and the types of constraints they support. (Refer back to Figure 4-1 for an overview of these categories.) One type of constraint—, the CHECK constraint, —can be defined as table constraints, column constraints, domain constraints, or within assertions. A CHECK constraint allows you to specify what values can be included in a column. You can define a range of values (for example, between 10 and 100), a list of values (for example, blues, jazz, pop, country), or a number of other conditions that restrict exactly what values are permitted in a column.

CHECK constraints are the most flexible of all the constraints and are often the most complicated. Despite this, the basic syntax used for a CHECK constraint is relatively simple. To create a column CHECK constraint, use the following syntax in a column definition:

< column name> { <data type> | <domain> } CHECK (<search condition>)

To create a table CHECK constraint, use the following syntax in a table definition:

[CONSTRAINT <constraint name>] CHECK (<search condition>)

I'll be discussing domain constraints and assertions later in this section.

As you can see by the syntax, a CHECK constraint is relatively straightforward. However, the values used for the <search condition> clause can be very extensive and, consequently, quite complex. The main concept is that the <search condition> is tested (one could say "checked") for any SQL statement that attempts to modify the data in a column covered by the CHECK constraint, and if it evaluates to TRUE, the SQL statement is allowed to complete; if it evaluates to FALSE, the SQL statement fails and an error message is displayed. The best way for you to learn about the clause is by looking at examples. However, most <search condition> components are based on the use of predicates in order to create the search condition. A predicate is an expression that operates on values. For example, a predicate can be used to compare values (for instance, COLUMN_1 > 10). The greater- than symbol (>) is a comparison predicate, sometimes referred to as a comparison operator. In this case, the predicate verifies that any value inserted into COLUMN_1 is greater than 10.

Many <search condition> components also rely on the use of subqueries. A subquery is an expression that is used as a component within another expression. Subqueries are used when an expression must access or calculate multiple layers of data, such as having to search a second table to provide data for the first table.

Both predicates and subqueries are complicated enough subjects to be beyond the scope of a discussion about CHECK constraints, and indeed each subject is treated separately in its own chapter. Despite the fact that both topics are discussed later in the book, I want to provide you with at least a few examples of CHECK constraints to give you a feel for how they're implemented in an SQL environment.

The first example we'll look at is a CHECK constraint that defines the minimum and maximum values that can be inserted into a column. The following table definition in this example creates three columns and one CHECK constraint (as a table constraint) that restricts the values of one of the columns to a range of numbers between 0 and 30:

```
CREATE TABLE CD_TITLES
( COMPACT_DISC_ID  INT,
  CD_TITLE         VARCHAR(60)  NOT NULL,
  IN_STOCK         INT          NOT NULL,
    CONSTRAINT CK_IN_STOCK CHECK ( IN_STOCK > 0 AND
IN_STOCK < 30 ) );
```

If you were to try to enter a value into the IN_STOCK column other than 1 through 29, you would receive an error. You can achieve the same results by defining a column constraint:

```
CREATE TABLE CD_TITLES
( COMPACT_DISC_ID  INT,
  CD_TITLE         VARCHAR(60)  NOT NULL,
  IN_STOCK         INT          NOT NULL
```

```
    CHECK ( IN_STOCK > 0 AND IN_STOCK < 30 ) );
```

Let’s take a closer look at the <search condition> clause in these statements, which in this case is (IN_STOCK > 0 AND IN_STOCK < 30). The clause first tells us that any value entered into the IN_STOCK column must be greater than 0 (IN_STOCK > 0). The AND keyword tells us that the conditions defined on either side of AND must be applied. Finally, the clause tells us that the value must be less than 30 (IN_STOCK < 30). Because the AND keyword is used, the value must be greater than 0 and less than 30.

Another way that a CHECK constraint can be used is to explicitly list the values that can be entered into the column. This is a handy option if you have a limited number of values and they’re not likely to change (or will change infrequently). The following SQL statement creates a table that includes a CHECK constraint that defines in which decade the music belongs:

```
CREATE TABLE CD_TITLES
( COMPACT_DISC_ID  INT,
  CD_TITLE         VARCHAR(60)  NOT NULL,
  ERA              CHAR(5),
  CONSTRAINT CK_ERA CHECK ( ERA IN ( '1940s', '1950s',
    '1960s', '1970s', '1980s', '1990s', '2000s' ) ) );
```

The value entered into the ERA column must be one of the seven decades represented by the search condition. If you tried to enter a value other than a null value or one of these seven, you would receive an error. Notice that the IN operator is used to designate that the ERA column values must be one of the set of values enclosed by parentheses following the keyword IN.

If the number of parentheses starts to confuse you, you can separate your code into lines that follow the embedding of those parentheses. For example, the preceding statement can be written as follows:

```
CREATE TABLE CD_TITLES
(
  COMPACT_DISC_ID  INT,
  CD_TITLE        VARCHAR(60)  NOT NULL,
  ERA             CHAR(5),
  CONSTRAINT CK_ERA CHECK
  (
    ERA IN
    (
      '1940s', '1950s', '1960s', '1970s', '1980s', '1990s', '2000s'
    )
  )
) ;
```

Each set of parentheses and its content is indented to a level that corresponds to the level of embedding for that particular clause, just like an outline. Using this method tells you exactly which clauses are enclosed in which set of parentheses, and the statement is executed just the same as if you hadn’t separated out the lines. The downside is that it takes up a lot of room (which is why I don’t use this method in this book), although it might be a helpful tool for you for those statements that are a little more complicated.

Now let's look at one other example of a CHECK constraint. This example is similar to the first one we looked at, only this one is concerned with values between certain numbers:

```
CREATE TABLE CD_TITLES
( COMPACT_DISC_ID  INT,
  CD_TITLE        VARCHAR(60)  NOT NULL,
  IN_STOCK        INT          NOT NULL,
  CONSTRAINT CK_IN_STOCK CHECK
   ( ( IN_STOCK BETWEEN 0 AND 30 ) OR
     ( IN_STOCK BETWEEN 49 AND 60 ) ) ) ;
```

In this statement, you use the BETWEEN operator to specify a range which includes the endpoints. Because you are creating two different ranges, you enclose each range specification in parentheses: (IN_STOCK BETWEEN 0 AND 30) and (IN_STOCK BETWEEN 49 AND 60).

These two range specifications are then connected by an OR keyword, which indicates that either one or the other condition must be met. As a result, any value entered into the IN_STOCK column must be from 0 through 30 or from 49 through 60.

Defining Assertions

An assertion is merely a type of CHECK constraint that can be applied to multiple tables. For this reason, an assertion must be created separately from a table definition. Unfortunately, most vendor products, including Oracle 11g, SQL Server 2005, and MySQL 5.0, don't yet support assertions. To create an assertion, use the following syntax:

```
CREATE ASSERTION <constraint name> CHECK <search conditions>
```

Creating an assertion is very similar to creating a table CHECK constraint. After the CHECK keyword, you must provide the necessary search condition(s). Now let's take a look at an example. Suppose the CD_TITLES table includes a column for the number of compact discs in stock. You want the total for that table to always be less than the maximum inventory you want to carry.

```
CREATE ASSERTION LIMIT_IN_STOCK CHECK
  ( ( SELECT SUM (IN_STOCK) FROM CD_TITLES ) < 5000 );
```

In this statement, I am using a subquery, (SELECT SUM (IN_STOCK) FROM CD_ TITLES), and comparing it to 5000. The subquery begins with the SELECT keyword, which is used to query data from a table. The SUM function adds the values in the IN_STOCK column, and the FROM keyword specifies which table the column is in. The results of this subquery are then compared (using the less than comparison operator) to 5000. If you try to add a value to the IN_STOCK column that would cause the total to exceed 5000, you will receive an error.

Creating Domains and Domain Constraints

The last type of CHECK constraint is the kind that you insert into a domain definition. For the most part, the constraint definition is similar to what you've seen before, except that you do not tie the constraint to a specific column or table. In fact, domain constraints use the VALUE keyword when referring to the value within a column defined with that particular domain. Let's look at the syntax for creating a domain:

```
CREATE DOMAIN <domain name> [ AS ] <data type>
[  DEFAULT <default value>  ]
[  CONSTRAINT <constraint name> ] CHECK ( <search condition>  )
```

You should already be familiar with most of the elements in this syntax. I discuss data types and default clauses in Chapter 3, and the constraint definition is similar to what you've seen so far in this chapter.

In the following example, I create a domain that's based on the INT data type and that requires all values to be between 0 and 30:

```
CREATE DOMAIN STOCK_AMOUNT AS INT
  CONSTRAINT CK_STOCK_AMOUNT CHECK (VALUE BETWEEN 0 AND 30 );
```

The only really new item here (other than the CREATE DOMAIN clause) is the keyword VALUE, which, as I said, refers to the value of the column defined with the STOCK_ AMOUNT domain. As a result, if you try to insert a value (into one of those columns) that is not between 0 and 30, you will receive an error.

Try This 4-2 Adding a CHECK Constraint

In this Try This exercise, which is relatively short, you will be using the ALTER TABLE statement to modify the COMPACT_DISCS table. You will be adding a column to the table and then defining a CHECK constraint that restricts the values that can be entered into the column. The additional column and constraint will have no impact on other tables in the INVENTORY database or on the relationship between tables. You can download the Try_This_04.txtfile, which contains the SQL statements used in this exercise.

Step by Step

1. Open the client application for your RDBMS and connect to the INVENTORY database.

2. You're going to modify the COMPACT_DISCS table by adding the IN_STOCK column.

Enter and execute the following SQL statement:

```
ALTER TABLE COMPACT_DISCS
  ADD COLUMN IN_STOCK  INT  NOT NULL;
```

NOTE: For Oracle and SQL Server, omit the keyword COLUMN

3. Now that the column exists, you can add a CHECK constraint to the table definition. You could have entered the constraint as a column constraint, but adding it separately as a table constraint allows you to do each step separately so you can see the results of your actions.

The CHECK constraint limits the values that can be entered into the IN_STOCK column. Each value must be greater than 0, but less than 50. Enter and execute the following SQL statement:

```
ALTER TABLE COMPACT_DISCS
  ADD CONSTRAINT CK_IN_STOCK CHECK ( IN_STOCK > 0 AND IN_STOCK < 50 );
```

4. Close the client application.

Chapter 6. Learn database creation and administration

In order to create or add data, you first need to be able to create a table in the database. In order to create a new table, ‘CREATE TABLE’ is the statement to be entered into the database.

You should first place the words ‘CREATE TABLE’. Then, a table name should be entered. Open parenthesis should be followed by the keywords. The column name and data type should be followed by closed parenthesis where additional parameters are defined. Statements in SQL should all end with an “;” just how all English sentences end with a period.

There are a few rules that should be followed when using SQL:

- All the names of tables and columns should start with a letter.
- After the column names are properly started, numbers, letters or even underscores can follow in the rest of the column name.
- A total of 30 characters is the maximum length.
- You can’t use keywords such as ‘select’, ‘create’, or ‘insert’ as this will confuse the database.

For instance, let’s say you wanted to base your table on quotes found in books. Your table will consist of the four types of data: text, character, book, and year. This will organize the text recalled in a book, the character that in a book or that says text, the book that they can be found in and the year that

the book was published. An example below is how you would create the proper table:

CREATE TABLE books_quotes

(‘Q_TEXT’ varchar (200),

‘Q_CHARACTER’ varchar (20),

‘Q_BOOK’ varchar (20),

‘Q_YEAR’ number (4));

The result of this command will create an empty table that contains columns.

- ‘Q_TEXT’ can accept a string as long as 200 characters
- ‘Q_CHARACTER’ can accept a string as long as 20 characters
- ‘Q_BOOK’ can accept a 20 character long string
- ‘Q_YEAR’ can accept a listing of a year with four numbers

The next step will be to fill out the book quotes and data into the table. There are a lot of graphic interface tools for managing the tables and data used in a database. An SQL script is simply a collection of commands that are able to be executed in a sequential manner. This method can be quite useful when you have a lot of data to fill into a table. In order to insert or add a row into your database, the command is ‘INSERT’. Here is the format in order to insert data:

INSERT INTO table_name

(column_1, column_2, … column_n)

VALUES (value_1, value_2, … value_n);

When you need to insert a row of data into a table, the keyword ‘INSERT’ should be followed by the keyword ‘INTO’ and then the table name should be entered. The parenthesis should contain the column names and be separated by commas but this is an optional step but is a good type of practice to keep in SQL. This practice will help the columns be clearly defined and that the right data is being entered into the right columns. After you have done this, you will then need to define what data will be inserted. The keyword ‘VALUES’ while a list of values follows and will be enclosed by parenthesis. Strings shouldn’t be enclosed in single quotes. Numbers should also not be enclosed. The SQL script should look like this:

INSERT INTO Book_quotes

(Q_TEXT, Q_CHARACTER, Q_BOOK,

VALUES (‘quotes placed here,

(‘more quotes placed here’, ‘and another quote’,

As a beginner in SQL, you must know how to create DATABASES. Databases are simply systematic collections of data that can be stored in your computer, where they can be retrieved easily for future use.

The system is called DBMS (Database Management System), and is used extensively by all institutions that need to store large volumes of information or data.

Examples of data are: Name, Age, Address, Country, ID number and all vital information that have to be organized and stored.

The retrieval of these databases is possible through database software or programs such as, SQL, MySQL, ORACLE and other similar apps.

Creating databases is simple with this SQL statement:

Example: CREATE DATABASE“database_name”;

If you want to create a“MyStudents” database, you can state the SQL query this way:

Example: CREATE DATABASE MyStudents;

If you want to create a“My_Sales”database, you can state your SQL this way:

Example: CREATE DATABASE My_Sales;

The names of your databases must be unique within the RDBMS (Relational Database Management System). After creating your database, you can now create tables for your databases.

You can double check if your database exists by this SQL query:

Example: SHOW DATABASES;

This SQL statement will display all the databases that you have created.

It is important to note that your ability to retrieve or fetch the data that you have stored is one vital consideration.

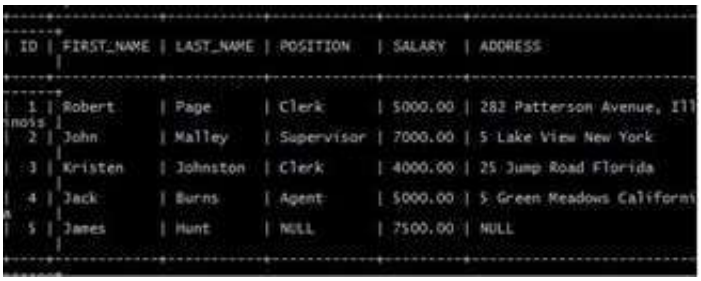
Therefore, you have to choose the most applicable and most appropriate SQL server or software that you can optimize and synchronize with the computer you are using.

Chapter 7: Modify and control tables

Changing a Table's Name

The ALTER TABLE command can be used with the RENAME function to change a table's name.

To demonstrate the use of this statement, use the EMPLOYEES table with the following records:



ID	FIRST_NAME	LAST_NAME	POSITION	SALARY	ADDRESS
1	Robert	Page	Clerk	5000.00	282 Patterson Avenue, Illinois
2	John	Malley	Supervisor	7000.00	5 Lake View New York
3	Kristen	Johnston	Clerk	4000.00	25 Jump Road Florida
4	Jack	Burns	Agent	5000.00	5 Green Meadows California
5	James	Hunt	NULL	7500.00	NULL

To change the EMPLOYEES table name to INVESTORS, use the following statement:[60]

ALTER TABLE EMPLOYEES RENAME INVESTORS;

Your table is now called INVESTORS.

Modifying Column Attributes

A column's attributes refer to the properties and behaviors of data entered in a column. Normally, you set the column attributes when you create the table. However, you may still change one or more attributes using the ALTER TABLE command.

You may modify the following:

- Column name
- Column Data type assigned to a column
- The scale, length, or precision of a column
- Use or non-use of NULL values in a column

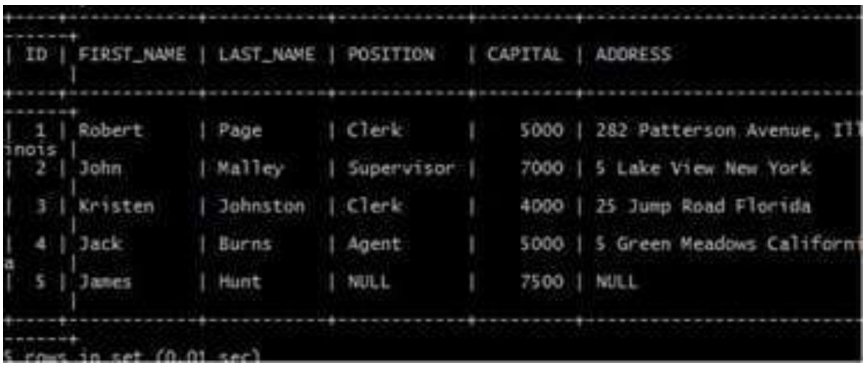
Renaming Columns

You may want to modify a column's name to reflect the data it contains. For instance, since you renamed the EMPLOYEES database to INVESTORS, the SALARY column will no longer be appropriate. You can change the column name to something like CAPITAL. Likewise, you may want to change its data type from DECIMAL to an INTEGER TYPE with a maximum of ten digits.

To do so, enter the following statement:

ALTER TABLE INVESTORS CHANGE SALARY CAPITAL INT(10);

[61]The result is the following:



Deleting a Column

At this point, the Position column is no longer applicable. You can drop the column using the following statement:

ALTER TABLE INVESTORS

DROP COLUMN Position;

Here's the updated INVESTORS table: [62]

Adding a New Column

Since you're now working on a different set of data, you may decide to add another column to make the data on the INVESTORS table more relevant. You can add a column that will store the number of stocks owned by each investor. You may name the new column STOCKS. This column will accept integers up to 9 digits.

ID	FIRST_NAME	LAST_NAME	CAPITAL	ADDRESS	STOCK
1	Robert	Page	5000.00	282 Patterson Avenue, Illinois	NUL
2	John	Malley	7000.00	5 Lake View New York	NUL
3	Kristen	Johnston	4000.00	25 Jump Road Florida	NUL
4	Jack	Burns	5000.00	5 Green Meadows California	NUL
5	James	Hunt	7500.00	NULL	NUL

You can use the following statement to add the STOCKS column:

```
ALTER TABLE INVESTORS ADD STOCKS INT(9);
```

The following is the updated INVESTORS table: [63]

Modifying an Existing Column without Changing its Name

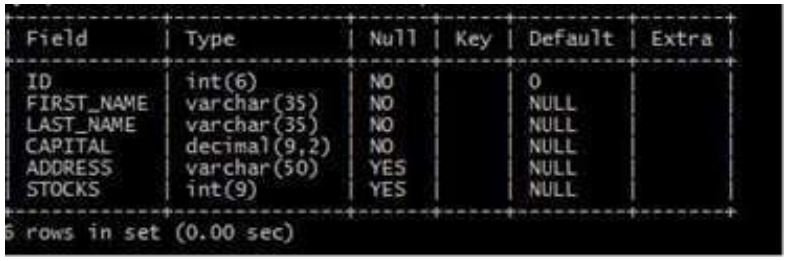
You may also combine the ALTER TABLE command with the MODIFY keyword to change the data type and specifications of a table. To demonstrate this, you can use the following statement to modify the data type of the column CAPITAL from an INT type to a DECIMAL type with up to 9 digits and two decimal numbers.

ALTER TABLE INVESTORS MODIFY CAPITAL DECIM AL(9.2) NOT NULL;

By this time, you may be curious to see the column names and attributes of the INVESTORS table. You can use the 'SHOW COLUMNS' statement to display the table's structure. Enter the following statement:

SHOW COLUMNS FROM INVESTORS;

Here's a screenshot of the result: [64]



Field	Type	Null	Key	Default	Extra
ID	int(6)	NO		0	
FIRST_NAME	varchar(35)	NO		NULL	
LAST_NAME	varchar(35)	NO		NULL	
CAPITAL	decimal(9,2)	NO		NULL	
ADDRESS	varchar(50)	YES		NULL	
STOCKS	int(9)	YES		NULL	

6 rows in set (0.00 sec)

Rules to Remember when Using ALTER TABLE

- Adding Columns to a Database Table

When adding a new column, bear in mind that you can't add a column with a NOT NULL attribute to a table with existing data. You will generally specify a column to be NOT NULL to indicate that it will hold a value. Adding a NOT NULL column will contradict the constraint if the existing data don't have values for a new column.

- Modifying Fields/Columns

1. You can easily modify the data type of a column
2. .You can increase the number of digits that numeric data types hold but you will only be able to decrease it if the largest number of digits stored by a table is equal to or lower than the desired number of digits.

3. You can increase or decrease the decimal places of numeric data types as long as they don't exceed the maximum allowable decimal places.

If not handled properly, deleting and modifying tables can result to loss of valuable information. So, be extremely careful when you're executing the ALTER TABLE and DROP TABLE statements.

Deleting Tables

Dropping a table will also remove its data, associated index, triggers, constraints, and permission data. You should be careful when using this statement.

Here's the syntax:

DROP TABLE table_name;

For example, if you want to delete the INVESTORS TABLE from the xyzcompany database, you may use

the following statement:

DROP TABLE INVESTORS;

The DROP TABLE command effectively removed the INVESTORS table from the current database.

If you try to access the INVESTORS table with the following command:

SELECT* FROM INVESTORS;

SQL will return an error, like this:

Combining and joining tables

You can combine data from several tables if a common field exists between them. To do this, use the JOIN statement.

SQL supports several types of JOIN operations:

INNER JOIN

The INNER JOIN, or simply JOIN, is the most commonly used type of JOIN. It displays the rows when the tables to be joined have a matching field.

Here's the syntax: [65]

SELECT colum n_name(s)

FROM table1

INNER JOIN table2

ON table1.colum n_name=table2.colum n_name;

In this variation, the JOIN clause is used instead of INNER JOIN.

SELECT colum n_names(s)

FROM table1

JOIN table2

ON table1.colum n_name=table2.colum n_name;

LEFT JOIN

The LEFT JOIN operation returns all left table rows with the matching right table rows. If no match is found, the right side returns NULL.

[66]Here's the syntax for LEFT JOIN:

SELECT colum n_name(s)

FROM table1

LEFT JOIN table2

ON table1.colum n_name=table2.colum n_name;

In some database systems, the keyword LEFT OUTER JOIN is used instead of LEFT JOIN. Here's the syntax for this variation:[67]

SELECT colum n_name(s)

FROM table1

LEFT OUTER JOIN table2

ON table2.colum n_name=table2.colum n_name;

RIGHT JOIN

This JOIN operation returns all right table rows with the matching left table rows.

The following is the syntax for this operation: [68]

SELECT colum n_name(s)

FROM table1

RIGHT JOIN table2

ON table2.colum n_name=table2.colum n_name;

In some database systems, the RIGHT OUTER JOIN is used instead of LEFT JOIN. Here's the syntax for this variation:

SELECT colum n_name(s)

FROM table1

RIGHT OUTER JOIN table2

ON table2.colum n_name=table2.colum n_name;

http://www.sql-join.com/sql-join-types/

FULL OUTER JOIN

This JOIN operation displays all rows when at least one table meets the condition. It combines the results from both RIGHT and LEFT join operations.

Here's the syntax:

SELECT colum n_name(s)

FROM table1

FULL OUTER JOIN table2

ON table2.colum n_name=table2.colum n_name;

To demonstrate the JOIN operation in SQL, use the Branch_Sales and Branch_Location tales:

Branch_Sales Table

Branch	Product_ID	Sales
New York	101	7500.00
Los Angeles	102	6450.00
Chicago	101	1560.00
Philadelphia	101	1980.00
Denver	102	3500.00
Seattle	101	2500.00
Detroit	102	1450.00

Location Table

Region	Branch
East	New York City
East	Chicago
East	Philadelphia
East	Detroit
West	Los Angeles
West	Denver
West	Seattle

The objective is to fetch the sales by region. The Location table contains the data on regions and branches while the Branch_Sales table holds the sales data for each branch. To find the sales per region, you need to combine the data from the Location and Branch_Sales tables. Notice that these tables have a common field, the Branch, which is the field that links the two tables.

The following statement will demonstrate how you can link these two tables by using table aliases:

```
SELECT A1.Region Region, SUM(A2.Sales) Sales
FROM Location A1, Branch_Sales A2
WHERE A1.Branch = A2.Branch
GROUP BY A1.Region;
```

This would be the result: [69]

In the first two lines, the statement tells SQL to select the fields ‘Region’ from the Location table and the total of the ‘Sales’ field from the Branch_Sales table. The statement uses table aliases. The ‘Region’ field was aliased as Region while the sum of the SALES field was aliased as SALES.

Table aliasing is the practice of using a temporary name for a table or a table column. Using aliases helps make statements more readable and concise. For example, if you opt not to use a table alias for the first line, you would have used the following statement to achieve the same result:

SELECT Location.Region Region,

SUM(Branch_Sales.Sales) SALES

Alternatively, you can specify a join between two tables by using the JOIN and ON keywords. For instance, using these keywords, the query would be:

SELECT A1.Region REGION, SUM(A2.Sales) SALES

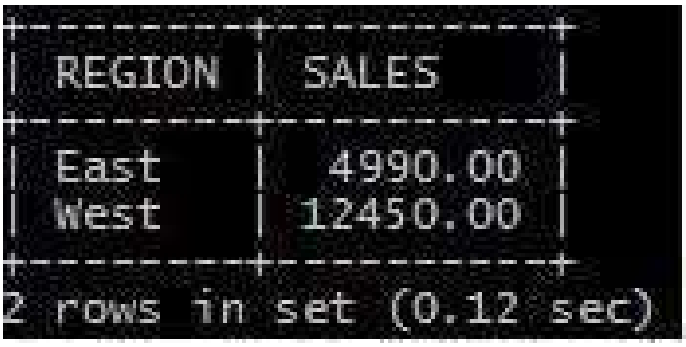
FROM Location A1

JOIN Branch_Sales A2

ON A1.Branch = A2.Branch

GROUP BY A1.Region;

The query would produce an identical result: [70]



```
+--------+----------+
| REGION | SALES    |
+--------+----------+
| East   |  4990.00 |
| West   | 12450.00 |
+--------+----------+
2 rows in set (0.12 sec)
```

Using Inner Join

An inner join displays rows when there is one or more matches on two tables. To demonstrate this, use the following tables:

Branch_Sales table

Branch	Product_ID	Sales
New York	101	7500.00
Philadelphia	101	1980.00
Denver	102	3500.00
Seattle	101	2500.00
Detroit	102	1450.00

Location_table

Region	Branch
East	New York
East	Chicago
East	Philadelphia
East	Detroit
West	Los Angeles
West	Denver

West	Seattle

You can achieve this by using the INNER JOIN statement.

You can enter the following:

SELECT A1.Branch BRANCH, SUM(A2.Sales) SALES

FROM Location A1

INNER JOIN Branch_Sales A2

ON A1.Branch = A2.Branch

GROUP BY A1.Branch;

[71]This would be the result:

```
+---------------+----------+
| BRANCH        | SALES    |
+---------------+----------+
| Denver        | 3500.00  |
| Detroit       | 1450.00  |
| New York      | 7500.00  |
| Philadelphia  | 1980.00  |
| Seattle       | 2500.00  |
+---------------+----------+
5 rows in set (0.00 sec)
```

Take note that by using the INNER JOIN, only the branches with records in the Branch_Sales report were included in the results even though you are actually applying the SELECT statement on the Location table. The ‘Chicago’ and ‘Los Angeles’ branches were excluded because there are no records for these branches in the Branch_Sales table.

Using Outer Join

In the previous example, you have used the Inner Join to combine tables with common rows. The OUTER JOIN command is used for this purpose.

The example for the OUTER JOIN will use the same tables used for INNER JOIN: the Branch_Sales table and Location_table.

This time, you want a list of sales figures for all stores. A regular join would have excluded Chicago and Los Angeles because these branches were not part of the Branch_Sales table. Therefore, you want to do an OUTER JOIN.

The statement is the following:

SELECT A1.Branch, SUM(A2.Sales) SALES

FROM Location A1, Branch_Sales A2

WHERE A1.Branch = A2.Branch (+)

GROUP BY A1.Branch;

Please note that the Outer Join syntax is database-dependent. The above statement uses the Oracle syntax.

Here's the result: [72]

Branch	Sales
Chicago	NULL
Denver	3500.00
Detroit	1450.00
Los Angeles	NULL
New York	7500.00
Philadelphia	1980.00
Seattle	2500.00

When combining tables, be aware that some JOIN syntax have different results across database systems. To maximize this powerful database feature, it is important to read the RDBMS documentation.

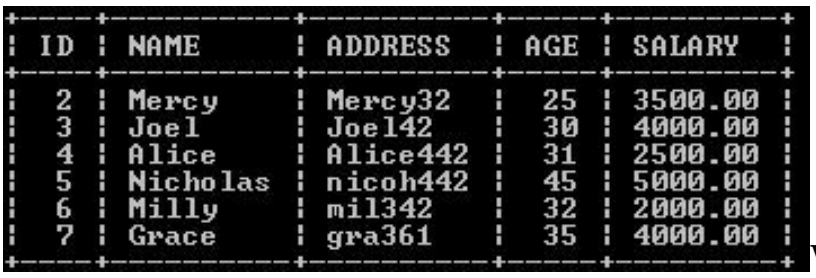
LIMIT, TOP and ROWNUM Clauses

The TOP command helps us retrieve only the TOP number of records from the table. However, you must note that not all databases support the TOP command. Some will support the LIMIT while others will support the ROWNUM clause.

The following is the syntax to use the TOP command on the SELECT statement:

SELECT TOP number|percent columnName(s)

FROM tableName



ID	NAME	ADDRESS	AGE	SALARY
2	Mercy	Mercy32	25	3500.00
3	Joel	Joel42	30	4000.00
4	Alice	Alice442	31	2500.00
5	Nicholas	nicoh442	45	5000.00
6	Milly	mil342	32	2000.00
7	Grace	gra361	35	4000.00

WHERE [condition]

We want to use the EMPLOYEES table to demonstrate how to use this clause. The table has the following data: [73]

The following query will help us fetch the first 2 rows from the table:

SELECT TOP 2 * FROM EMPLOYEES;

Note that the command given above will only work in SQL Server. If you are using MySQL Server, use the LIMIT clause as shown below:

SELECT * FROM EMPLOYEES

LIMIT 2;[74]

```
mysql> SELECT * FROM EMPLOYEES
    -> LIMIT 2;
+----+-------+---------+-----+---------+
| ID | NAME  | ADDRESS | AGE | SALARY  |
+----+-------+---------+-----+---------+
|  2 | Mercy | Mercy32 |  25 | 3500.00 |
|  3 | Joel  | Joel42  |  30 | 4000.00 |
+----+-------+---------+-----+---------+
2 rows in set (0.37 sec)

mysql>
```

Only the first two records of the table are returned.

If you are using an Oracle Server, use the ROWNUM with SELECT clause, as shown below:

SELECT * FROM EMPLOYEES

WHERE ROWNUM <= 2;

ORDER BY Clause

This clause helps us sort our data, either in ascending or descending order. The sorting can be done while relying on one or more columns. In most databases, the results are sorted in an ascending order by default.

The ORDER BY clause uses the syntax below:

SELECT columns_list

FROM tableName

[WHERE condition]

In the ORDER BY clause, you may use one or more columns. However, you must ensure that the column you choose to sort the data is in the column list. Again, we will use the EMPLOYEES table with the data shown below: [75]

```
+----+----------+----------+-----+---------+
| ID | NAME     | ADDRESS  | AGE | SALARY  |
+----+----------+----------+-----+---------+
|  2 | Mercy    | Mercy32  |  25 | 3500.00 |
|  3 | Joel     | Joel42   |  30 | 4000.00 |
|  4 | Alice    | Alice442 |  31 | 2500.00 |
|  5 | Nicholas | nicoh442 |  45 | 5000.00 |
|  6 | Milly    | mil342   |  32 | 2000.00 |
|  7 | Grace    | gra361   |  35 | 4000.00 |
+----+----------+----------+-----+---------+
```

Now, we need to use the NAME and SALARY columns to sort the data in ascending order. The following command will help us achieve this:

```
SELECT * FROM EMPLOYEES
  ORDER BY NAME, SALARY;
```

The query will return the following result: [76]

```
mysql> SELECT * FROM EMPLOYEES
    ->    ORDER BY NAME, SALARY;
+----+----------+-----------+-----+---------+
| ID | NAME     | ADDRESS   | AGE | SALARY  |
+----+----------+-----------+-----+---------+
|  4 | Alice    | Alice442  |  31 | 2500.00 |
|  7 | Grace    | gra361    |  35 | 4000.00 |
|  3 | Joel     | Joel42    |  30 | 4000.00 |
|  2 | Mercy    | Mercy32   |  25 | 3500.00 |
|  6 | Milly    | mil342    |  32 | 2000.00 |
|  5 | Nicholas | nicoh442  |  45 | 5000.00 |
+----+----------+-----------+-----+---------+
6 rows in set (0.32 sec)

mysql>
```

We can also use the SALARY column to sort the data in descending order:

```
SELECT * FROM EMPLOYEES
  ORDER BY SALARY DESC;
```

This is the result:

```
mysql> SELECT * FROM EMPLOYEES
    ->    ORDER BY SALARY DESC;
+----+----------+-----------+-----+---------+
| ID | NAME     | ADDRESS   | AGE | SALARY  |
+----+----------+-----------+-----+---------+
|  5 | Nicholas | nicoh442  |  45 | 5000.00 |
|  3 | Joel     | Joel42    |  30 | 4000.00 |
|  7 | Grace    | gra361    |  35 | 4000.00 |
|  2 | Mercy    | Mercy32   |  25 | 3500.00 |
|  4 | Alice    | Alice442  |  31 | 2500.00 |
|  6 | Milly    | mil342    |  32 | 2000.00 |
+----+----------+-----------+-----+---------+
6 rows in set (0.05 sec)

mysql>
```

https://www.tutorialspoint.com/sql/sql-top-clause.htm

GROUP BY Clause

This clause is used together with the SELECT statement to group data that is related together, creating groups. The GROUP BY clause should follow the WHERE clause in SELECT statements, and it should precede the ORDER BY clause.

The following is the syntax:

SELECT column_1, column_2

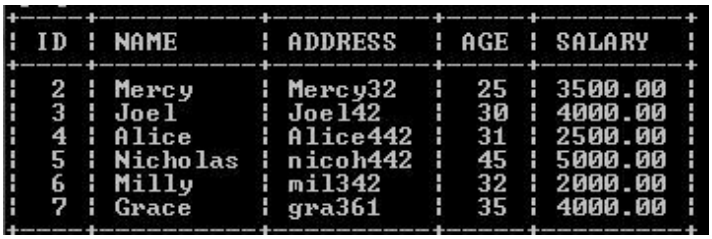
FROM tableName

WHERE [conditions]

GROUP BY column_1, column_2

ORDER BY column_1, column_2

Let’s use the EMPLOYEES table with the data given below: [77]



ID	NAME	ADDRESS	AGE	SALARY
2	Mercy	Mercy32	25	3500.00
3	Joel	Joel42	30	4000.00
4	Alice	Alice442	31	2500.00
5	Nicholas	nicoh442	45	5000.00
6	Milly	mil342	32	2000.00
7	Grace	gra361	35	4000.00

If you need to get the total SALARY of every customer, just run the following command:

SELECT NAME, SALARY, SUM(SALARY) FROM EMPLOYEES

GROUP BY NAME;

The DISTINCT Keyword

This keyword is used together with the SELECT statement to help eliminate duplicates and allow the selection of unique records.

This is because there comes a time when you have multiple duplicate records in a table and your goal is to choose only the unique ones. The DISTINCT keyword can help you achieve this. This keyword can be used with the following syntax:

SELECT DISTINCT column_1, column_2,.....column_N

FROM tableName

WHERE [condition]

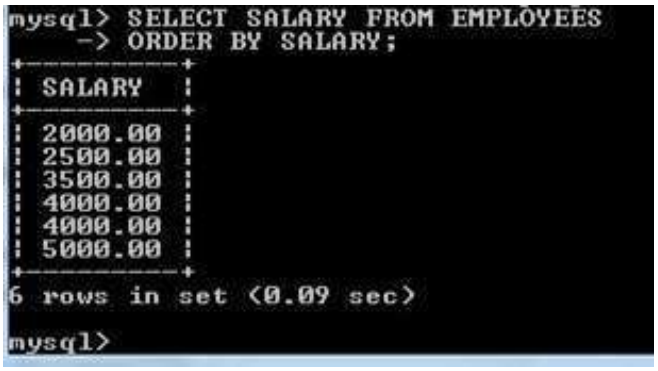
We will use the EMPLOYEES table to demonstrate how to use this keyword. The table has the following data: [78]

ID	NAME	ADDRESS	AGE	SALARY
2	Mercy	Mercy32	25	3500.00
3	Joel	Joel42	30	4000.00
4	Alice	Alice442	31	2500.00
5	Nicholas	nicoh442	45	5000.00
6	Milly	mil342	32	2000.00
7	Grace	gra361	35	4000.00

We have a duplicate entry of 4000 in the SALARY column of the above table. This can be seen after we run the following query:

SELECT DISTINCT SALARY FROM EMPLOYEES

ORDER BY SALARY; [79]



```
mysql> SELECT SALARY FROM EMPLOYEES
    -> ORDER BY SALARY;
+----------+
| SALARY   |
+----------+
| 2000.00 |
| 2500.00 |
| 3500.00 |
| 4000.00 |
| 4000.00 |
| 5000.00 |
+----------+
6 rows in set (0.09 sec)

mysql>
```

We can now combine the query with the DISTINCT keyword and see what the query returns:

[80]

```
mysql> SELECT DISTINCT SALARY FROM EMPLOYEES
    -> ORDER BY SALARY;
+----------+
| SALARY   |
+----------+
| 2000.00 |
| 2500.00 |
| 3500.00 |
| 4000.00 |
| 5000.00 |
+----------+
5 rows in set (0.07 sec)

mysql>
```

SQL Sub-queries

Until now, we have been executing single SQL queries to perform insert, select, update, and delete functions. However, there is a way to execute SQL queries within the other SQL queries. For instance, you can select the records of all students in the database with an age greater than a particular student. In this chapter, we shall demonstrate how we can execute sub-queries or queries-within-queries in SQL.

You should have tables labeled “Student” and “Department” with some records.

First, we can retrieve Stacy's age, store it in some variable, and then, using a "where" clause, compare the age in our SELECT query. The second approach is to embed the query that retrieves Stacy's age inside the query that retrieves the ages of all students. The second approach employs a sub-query technique. Have a look at Query 1 to see sub-queries in action.

Query 1

```
Select * From Student

where StudentAge >

(Select StudentAge from Student

where StudName = 'Stacy'

)
```

Notice that in Query 1, we’ve used round brackets to append a sub-query in the “where” clause. The above query will retrieve the records of all students from the “Student” table where the age of the student is greater than the age of “Stacy”. The age of “Stacy” is 20; therefore, in the output, you shall see the records of all students aged greater than 20. The output is the following:

StudI	StudName	StudentAge	StudentGender	**DepID**

D				
1	Alice	21	Male	**2**
4	Jacobs	22	Male	**5**
6	Shane	22	Male	**4**
7	Linda	24	Female	**4**
9	Wolfred	21	Male	**2**
10	Sandy	25	Female	**1**
14	Mark	23	Male	**5**
15	Fred	25	Male	**2**
16	Vic	25	Male	**NULL**
17	Nick	25	Male	**NULL**

Similarly, if you want to update the name of all the students with department name “English”, you can do so using the following sub-query:

Query 2

```
Update Student
Set StudName = StudName + ' Eng'
where Student.StudID in (
Select StudID
from Student
Join
Department
On Student.DepID = Department.DepID
where DepName = 'English'
```

)

In the above query, the student IDs of all the students in the English department have been retrieved using a JOIN statement in the sub-query. Then, using an UPDATE statement, the names of all those students have been updated by appending the string “Eng” at the end of their names. A WHERE statement has been used to match the student IDs retrieved by using a sub-query.

SQL Character Functions

SQL character functions are used to modify the appearance of retrieved data. Character functions do not modify the actual data, but rather perform certain modifications in the way data is represented. SQL character functions operate on string type data. In this chapter, we will look at some of the most commonly used SQL character functions.

Note:

- Concatenation (+)

Concatenation functions are used to concatenate two or more strings. To concatenate two strings in SQL, the ‘+’ operator is used. For example, we can join student names and student genders from the student column and display them in one column, like the following:

```
Select StudName +' '+StudentGender as NameAndGender
from Student
```

- Replace

The replace function is used to replace characters in the output string. For instance, the following query replaces “ac” with “rs” in all student names.

SQL Constraints

Constraints refer to rules that are applied on the columns of database tables. They help us impose restrictions on the kind of data that can be kept in that table. This way, we can ensure that there is reliability and accuracy of the data in the database.

Constraints can be imposed at column level or at the table level. The column constraints can only be imposed on a single column, while the table level constraints are applied to the entire table.

NOT NULL Constraint

The default setting in SQL is that a column may hold null values. If you don't want to have a column without a value, you can specify this.

Note that NULL means unknown data rather than no data. This constraint can be defined when you are creating table. Let's demonstrate this by creating a sample table:

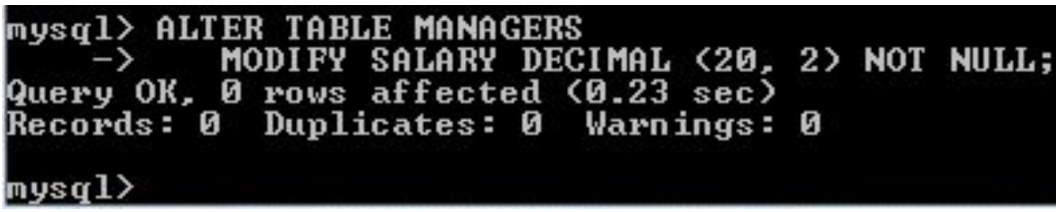
```
CREATE TABLE MANAGERS(
   ID   INT NOT NULL,
   NAME VARCHAR (15) NOT NULL,
   DEPT VARCHAR(20)  NOT NULL,
   SALARY   DECIMAL (20, 2),
   PRIMARY KEY (ID)
);
```
[81]

Above, we have created a table named MANAGERS with 4 columns, and the NOT NULL constraint has been imposed on three of these columns. This means that you must specify a value for each of these columns with the constraint; otherwise, an error will be raised.

Note that we did not impose the NOT NULL constraint to the SALARY column of our table. It is possible for us to impose the constraint on the

column even though it has already been created.

ALTER TABLE MANAGERS



MODIFY SALARY DECIMAL (20, 2) NOT NULL; [82]

Now, the column cannot accept a null value.

Default Constraint

This constraint will provide a default value to the column if a value for the column is not specified in the INSERT INTO column.

Consider the example given below:

```
CREATE TABLE SUPPLIERS(
  ID INT NOT NULL,
  NAME VARCHAR (15) NOT NULL,
  ADDRESS CHAR (20) ,
 ARREARS DECIMAL (16, 2) DEFAULT 6000.00,
  PRIMARY KEY (ID)
);
```

Suppose you had already created the table but you needed to add a default constraint on the ARREARS column. You can do it like this:

MODIFY ARREARS DECIMAL (16, 2) DEFAULT 6000.00;

This will update the table, and a default constraint will be created for the column.

If you no longer need this default constraint to be in place, you can drop it by running the command given below:

```
ALTER TABLE SUPPLIERS
  ALTER COLUMN ARREARS DROP DEFAULT;
```

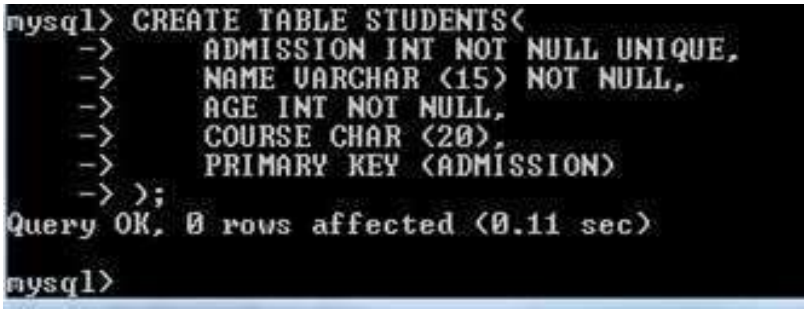
Unique Constraint

This constraint helps us avoid the possibility of having two or more records with similar values in one column. In the “employees” table, for example, we may need to prevent two or more employees from sharing the same ID.

The following SQL Query shows how we can create a table named “STUDENTS”. We will impose the UNIQUE constraint on the “ADMISSION” column:

```
CREATE TABLE STUDENTS(
   ADMISSION INT NOT NULL UNIQUE,
   NAME VARCHAR (15) NOT NULL,
   AGE INT NOT NULL,
   COURSE CHAR (20),
   PRIMARY KEY (ADMISSION)
);
```

[83]



Properly created indexes enhance efficiency in large databases. The selection of fields on which to create the index depends on the SQL queries that you

use frequently.

Perhaps you had already created the STUDENTS table without the UNIQUE constraint. The constraint can be added on the ADMISSION column by running the following command:

ALTER TABLE STUDENTS

MODIFY ADMISSION INT NOT NULL UNIQUE;

This is demonstrated below:

ALTER TABLE STUDENTS

ADD CONSTRAINT uniqueConstraint UNIQUE(ADMISSION, AGE);

The constraint has been given the name “uniqueConstraint” and assigned two columns, ADMISSION and AGE.

Anytime you need to delete the constraint, run the following command combining the ALTER and DROP commands:

ALTER TABLE STUDENTS

DROP CONSTRAINT uniqueConstraint;

The constraint will be deleted from the two columns. For MySQL users, the above command will not work. Just run the command given below:

ALTER TABLE STUDENTS

DROP INDEX uniqueConstraint; [84]

```
mysql> ALTER TABLE  STUDENTS
    ->      DROP INDEX uniqueConstraint;
Query OK, 0 rows affected (0.15 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Primary Key

The primary key constraint helps us identify every row uniquely. The column designated as the primary key must have unique values. Also, the column is not allowed to have NULL values.

Each table is allowed to only have one primary key, and this may be made up of single or multiple fields. When we use multiple fields as the primary key, they are referred to as a composite key. If a column for a table is defined as the primary key, then no two records will share same value in that field.

A primary key is a unique identifier. In the STUDENTS table, we can define the ADMISSION to be the primary key since it identifies each student uniquely. No two students should have the same ADMISSION number. Here is how the attribute can be created:

```
CREATE TABLE STUDENTS(
   ADMISSION INT NOT NULL,
   NAME VARCHAR (15) NOT NULL,
   AGE INT NOT NULL,
   PRIMARY KEY (ADMISSION)
);
```

The ADMISSION has been set as the Primary Key for the table. If we describe the table, you will find that the field is the primary key, as shown below: [85]

In the “Key” field above, the “PRI” indicates that the ADMISSION field is the primary key.

You may want to impose the primary key constraint on a table that already exists. This can be done by running the command given below:

```
ALTER TABLE STUDENTS
```

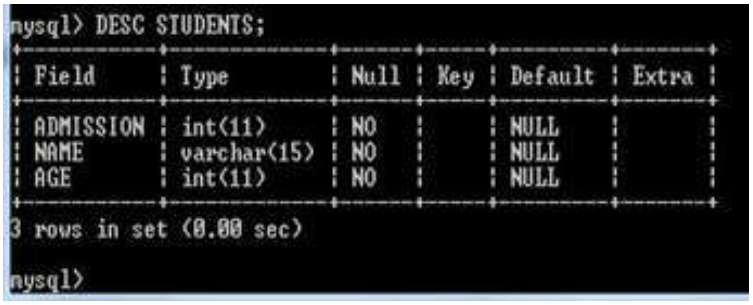
ADD CONSTRAINT PK_STUDADM PRIMARY KEY (ADMISSION, NAME);

In the above command, the primary key constraint has been given the name “PK_STUDADM” and assigned to two columns namely ADMISSION and NAME. This means that no two rows will have the same value for these columns.

The primary key constraint can be deleted from a table by executing the command given below:

ALTER TABLE STUDENTS DROP PRIMARY KEY;

After running the above command, you can describe the STUDENTS table and see whether it has any primary key: [86]



This shows that the primary key was dropped successfully.

Foreign Key

This constraint is used to link two tables together. Sometimes, it is referred to as a referencing key. The foreign key is simply a column or a set of columns that match a Primary Key in another table.

Consider the tables with the structures given below:

STUDENTS table:

CREATE TABLE STUDENTS(

ADMISSION INT NOT NULL,

NAME VARCHAR (15) NOT NULL,

AGE INT NOT NULL,

PRIMARY KEY (ADMISSION)

);

FEE table:

CREATE TABLE FEE (

ID INT NOT NULL,

DATE DATETIME,

STUDENT_ADM INT references STUDENTS (ADMISSION),

AMOUNT float,

PRIMARY KEY (ID)

);[87]

In the FEE table, the STUDENT_ADM field is referencing the ADMISSION field in the STUDENTS table. This makes the STUDENT_ADM column of FEE table a foreign key.

If we had created the FEE table without the foreign key, we could've still added it with the following command:

ALTER TABLE FEES

ADD FOREIGN KEY (STUDENT_ADM) REFERENCES STUDENTS (ADMISSION);

The foreign key will be added to the table. If you need to delete the foreign key, run the following command:

ALTER TABLE FEE

DROP FOREIGN KEY;

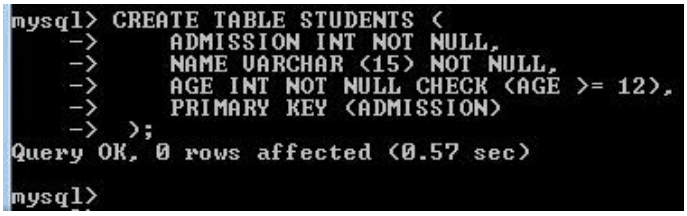
The foreign key will then be removed from the table.

CHECK Constraint

This constraint is used to create a condition that will validate the values that are entered into a record. If the condition becomes false, the record is violating the constraint, so it will not be entered into the table.

We want to create a table called STUDENTS and we don't want to have any student who is under 12 years of age. If the student doesn't meet this constraint, they will not be added to the table. The table can be created as follows:

```
CREATE TABLE STUDENTS (
    ADMISSION INT NOT NULL,
    NAME VARCHAR (15) NOT NULL,
    AGE INT NOT NULL CHECK (AGE >= 12),
    PRIMARY KEY (ADMISSION)
);
```



The table has been created successfully.

If you had already created the STUDENTS table without the constraint but then needed to implement it, run the command given below:

```
ALTER TABLE STUDENTS
    MODIFY AGE INT NOT NULL CHECK (AGE >= 12 );
```

The constraint will be added to the column AGE successfully.

It is also possible for you to assign a name to the constraint. This can be done using the below syntax:

```
ALTER TABLE STUDENTS
  ADD CONSTRAINT checkAgeConstraint CHECK(AGE >= 12);
```

https://www.tutorialspoint.com/sql/sql-index.htm

INDEX Constraint

An INDEX helps us quickly retrieve data from a database. To create an index, we can rely on a column or a group of columns in the database table. Once the index has been created, it is given a ROWID for every row before it can sort the data.

Properly created indexes enhance efficiency in large databases. The selection of fields on which to create the index depends on the SQL queries that you use frequently.

Suppose we created the following table with three columns:

```
CREATE TABLE STUDENTS (
ADMISSION INT NOT NULL,
NAME VARCHAR (15) NOT NULL,
AGE INT NOT NULL CHECK (AGE >= 12),
  PRIMARY KEY (ADMISSION)
);
```

We can then use the below syntax to implement an INDEX on one or more columns:

```
CREATE INDEX indexName
  ON tableName ( column_1, column_2.....);
```

Now, we need to implement an INDEX on the column named AGE to make it easier for us to search using a specific age. The index can be created as follows:

CREATE INDEX age_idx

ON STUDENTS (AGE); [89]

```
mysql> CREATE INDEX age_idx
    ->     ON STUDENTS (AGE);
Query OK, 1 row affected (0.20 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql>
```

If the index is no longer needed, it can be deleted by running the following command:

ALTER TABLE STUDENTS

DROP INDEX age_idx; [90]

```
mysql> ALTER TABLE STUDENTS
    ->     DROP INDEX age_idx;
Query OK, 1 row affected (0.13 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql>
```

ALTER TABLE Command

SQL provides us with the ALTER TABLE command that can be used for addition, removal and modification of table columns. The command also helps us to add and remove constraints from tables.

Suppose you had the STUDENTS table with the following data: [91]

ADMISSION	NAME	AGE
3420	NICHOLAS	10
1234	john	32
3456	mercy	23

ALTER TABLE STUDENTS ADD COURSE VARCHAR(10); [92]

```
mysql> ALTER TABLE STUDENTS ADD COURSE VARCHAR(10);
Query OK, 3 rows affected (0.15 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

When we query the contents of the table, we get the following:[93]

ADMISSION	NAME	AGE	COURSE
3420	NICHOLAS	10	NULL
1234	john	32	NULL
3456	mercy	23	NULL

This shows that the column has been added and each record has been assigned a NULL value in that column.

To change the data type for the COURSE column from VarChar to Char, execute the following command:

ALTER TABLE STUDENTS MODIFY COLUMN COURSE Char(1); [94]

```
mysql> ALTER TABLE STUDENTS MODIFY COLUMN COURSE Char(1);
Query OK, 3 rows affected (0.13 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

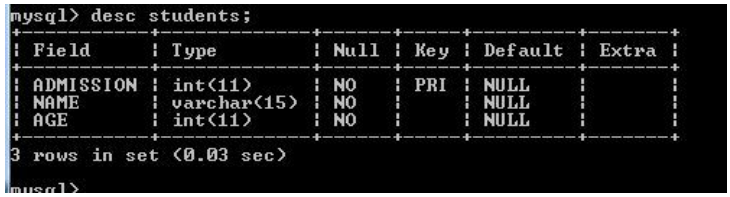
We can combine the ALTER TABLE command with the DROP TABLE command to delete a column from a table. To delete the COURSE column from the STUDENTS table, we run the following command:

ALTER TABLE STUDENTS DROP COLUMN COURSE; [95]

```
mysql> ALTER TABLE STUDENTS DROP COLUMN COURSE;
Query OK, 3 rows affected (0.18 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

We can then view the table details via the describe command to see whether the column was dropped successfully: [96]



The above figure shows that the column was dropped successfully.

Chapter 8: Security

Databases are containers that maintain all kinds of data, corporate secrets, employee data that is sensitive, lists of employees scheduled for separation, and many other types of information that needs proper security and access control. Many companies today are using Microsoft's active directory to manage users and sort them into access profiles using a group policy process. How this works in practice is that, employees of a company are assigned group permissions based on their job title, and within those group permissions, more individualized permissions are created depending on employee rank within a group. SQL does interact with Active Directory for access control, but it does not provide the internal security regarding authorization. The SQL application itself provides these services. There are four main components of database security: authentication, authorization, encryption, and access control.

Authentication pertains to validating whether a user has permission to access any resources of the system. The most common method of authentication by far is the username and password, verifying the credentials of a potential

user. Single sign-on systems also exist which use certificate authentication that the user does not interact with directly. The end user's system is prepared with information that provides authentication automatically without prompt.

Corporations go to great lengths to ensure system access is authenticated and appropriately authorized. Encryption strengthens this access control by scrambling data into indecipherable gibberish to any potential interceptors of transmitted data. Microsoft SQL Server uses RSA encryption to protect data. RSA is a data encryption algorithm that uses a layered hierarchical structure along with key management to secure data.

Authorization is the process that determines what resources within a system an authenticated user accesses. Once a client has provided acceptable credentials, the next step is to decide which entities the subject has permission to access or modify.

Lastly, SQL uses change tracking to maintain a log of all the actions of potentially unauthorized users. It is also possible to track the activities of all authorized users, but that isn't a part of the security functionality of change tracking. Power-users or super-users with elevated permissions may have access to all systems, but that does not authorize them as users of the database. Tracking changes protect a system from operations performed by users with elevated credentials.

SQL uses a security model comprised of three classes that interplay with each other: principals, securables, and permissions. Principals have permission to access specific objects. Securables refer to resources within a database that the system regulates access. Permission is the right to view, edit, or delete securables, and this access is pre-defined.

This security model belongs primarily to Microsoft SQL server, but there are equivalents within other SQL management products such as MySQL, DB2,

or 11G. The theories behind modeling access control are widespread across the IT industry and cover much more than database security. These same principals are behind nearly all enterprise applications that house sensitive data.

Security is a profound subject, and there are comprehensive books available on securing enterprise resources. The goal of this chapter is to focus on the most relevant topics after providing a brief overview of the general landscape of database security. The primary elements covered briefly above are an excellent base to expand upon the ideas of access roles and schema related to security.

Schema, in previous chapters appears, but as it relates to security, the schema defines ownership of resources and identifies principals based on user's relation to the owner. Microsoft defines schema as "A collection of database objects that are owned by a single person and form a single namespace." The single namespace refers to a limitation that does not allow two tables in the same schema to have the same name. Data consistency and referential ease are the guiding ideas behind SQL's design, and this is the reason behind the limitation. Principals can be a single user or single login as well as a group principle. Multiple users sharing one role are grouped using group policies, and all can cooperatively own a schema or many schemas. Transferring schema from one principle to another is possible and does not require renaming unless the new owner maintains a schema that would create a duplicate name. There are T-SQL statements for managing schema, but a majority of this work belongs to database administrators, not the principles of a database.

Roles are another layer of security for identifying access dependent upon title and responsibilities. There are many different kinds of roles available. SQL comes with fixed server roles and fixed database roles which provide

implicit permissions. It is also possible to create customized application roles, user-defined server roles, and user-defined database roles. The following is a list of Microsoft's fixed server roles:

Roles are very important to database security. All the other security functions are built on top of roles. Authentication determines who can access databases; authorization determines which principles have access to which schema. Encryption protects data from interception from those external to the company, as well as potential hazards internally. Sensitive data that leaks internally or is intercepted unintentionally is more dangerous than external threats. Roles maintain the minutiae of which users are allowed to perform any given operation within a database. Microsoft teaches security principals that provide a user with the least amount of access possible to perform their duties. This theory prevents users from having access to resources that they are not trained to use. The same guiding ideas are used with SQL. The purpose of roles is to limit the amount of access an employee has to a database that does not pertain to their specific responsibilities within the schema. Owners are considered “sysadmins”, but all other users of a database have specialized functions based on their training and experience. Database administrators are usually part of the IT department, but database owners are rarely within the same department as the database administrators, and so there is a level of autonomy that is necessary because administrators are responsible for maintaining the structure and consistency of dozens and even hundreds of databases. These rules and roles are all for the end goal of keeping consistent and referential data.

The users themselves are a major threat to data integrity. There are some users who should never have access to data and there are others who should only have restricted access to the data. You should identify a way to classify

the users into different categories to ensure that not every user has the access to classified or privileged information.

If you create the schema, you can specify who the owner is. If you are the owner of the schema, you can decide who you want to grant access to. If you do not grant some privileges, they are withheld by SQL. As the owner of the schema, you can also decide if you want to revoke the access that somebody has to your database. Every user must pass an authentication process before he can access the data he needs to. This process should help you identify the user. That procedure is implementation-dependent.

You can protect the following database objects using SQL:

- Views
- Columns
- Tables
- Character Sets
- Domains
- Translations
- Collations

There are different types of protection that you can use in SQL, and these include adding, seeing, deleting, modifying, using and referencing databases. You can also use different tools that are associated with protecting the queries.

You can give people access using the GRANT statement and remove the access using the REVOKE statement. When you control the use of the SELECT statement, the database tool will control who can view a specific database object like a column, view or table. When you control the use of the INSERT command, you can determine who can enter rows into a table.

When you restrict the use of the UPDATE command, you only allow some people to modify the data in the table. The same can be said about the DELETE statement.

People believe that their information is protected if you can control who can view, create, modify or delete data. It is true that your database is protected from most threats, but a hacker can still access come confidential information using some indirect methods.

A database has referential integrity if it is designed correctly. This means that the data in one table will always be consistent with the data in another table. Database developers and designers always apply constraints to tables which restrict what data can be entered in the database. If you use databases with referential integrity, users can create new tables that use the foreign key in a confidential table. This will allow them to source information from that table. This column will then serve as the link through which anybody can access confidential information.

Let us assume that you are a Wall Street stock analyst, and many people trust your assessment about which stock will give them the best returns. When you recommend a stock, people will always buy it and this increases the value of the stock. You maintain a database called FOUR_STAR that has the information and all your analyses. The top recommendations are in your newsletter, and you will restrict user access to this table. You will identify a way to ensure that only your subscribers can access this information.

You are vulnerable when anybody other than you creates a table that will use the same name for the stock field as the foreign key.

You should never grant people privileges if you know that they will misuse them. People never come with a trust certificate, but you would never lend your car to someone you do not trust. The same can be said about giving someone REFERENCE privileges to an important database.

The previous example explains why it is important that you maintain control of the REFERENCES privilege. The following reasons explain why it is important to use REFERENCES carefully:

- If another user were to specify a constraint in the HOT STOCKS by using the RESTRICT option, the DBMS will not allow you to delete a row from that table. This is because the referential constraint is being violated.
- The first person will need to drop or remove the constraints on a table if you want to use the DROP command to delete your table.

In simple words, it is never a good idea to let someone else define the constraints for your database since this will introduce a security breach. This also means that the user may sometimes get in your way.

If you want to maintain a secure system, you should restrict the access privilege that you grant to different users. You should also decide which users can access the data. Some people will need to access the data in the database to carry on with their work. If you do not give them the necessary access, they will constantly badger you and ask you to give them some information. Therefore, you should decide how you want to maintain the database security. You can use the WITH GRANT OPTION clause to manage database security. Let us consider the following examples:

```
GRANT UPDATE
ON RETAIL_PRICE_LIST
TO SALES_MANAGER WITH GRANT OPTION
```

The statement is similar to the GRANT UPDATE statement that allows the sales manager to update the retail price list. This statement also gives the manager the right to grant any update privileges to people she trusts. If you use this version of the GRANT statement, you should trust the fact that the

grantee will use the privilege wisely. You should also trust the fact that the grantee will grant the privilege to only the necessary people.

```
GRANT ALL PRIVILEGES
ON FOUR_STAR
TO BENEDICT_ARNOLD WITH GRANT OPTION;
```

You have to be careful when you use statements like the one above. If you misspell the name or give the wrong person access, you cannot guarantee the security of your database.

Chapter 9: Pivoting Data in SQL

Pivoting data is converting your data, which are presented in rows, into column presentations.

Through the use of PIVOT queries, you can manipulate the rows and columns to present variations of the table that will help you in analyzing your table. PIVOT can present a column into multiple columns. You have also the option to use UNPIVOT query. UNPIVOT does the opposite of what PIVOT does.

It is extremely useful in multidimensional reporting. You may need it in generating your numerous reports.

How can you compose your PIVOT query?

Step #1–Ascertain that your SQL can allow PIVOT queries

Is the version of your SQL server appropriate for PIVOT queries. If not, then you cannot accomplish these specific queries.

Step #2 - Determine what you want displayed in your results

Identify the column or data you want to appear in your results or output page.

Step #3–Prepare your PIVOT query, using SQL

Use your knowledge of SQL to compose or create your PIVOT query.

To understand more about PIVOT, let's use the table below as a base table.

ProductSales

ProductName	Year	Earnings
RazorBlades1	2015	12000.00
BarHandles1	2016	15000.00
RazorBlades2	2015	10000.00
BarHandles2	2016	11000.00

Let's say you want an output that will show the ProductName as the column headings. This would be your PIVOT query:

Example #1:

```
SELECT * FROM ProductSales
PIVOT (SUM(Earnings)
    FOR ProductNames IN ([RazorBlades1], [BarHandles1], [RazorBlades2], [BarHandles2]) AS PVT
```

With the PIVOT query above, your ProductSales table will now appear like this:

#ProductNamesPIVOTResults

ProductName	RazorBlades1	BarHandles1	RazorBlades2	BarHandles 2
Year	2015	2016	2015	2016
Earnings	12000.00	15000.00	10000.00	11000.00

You can also manipulate the table based on your preferences.

Example #2

If you want the Year to be the column headings, you can write your PIVOT query this way:

```
SELECT * FROM ProductSales
PIVOT (SUM(EARNINGS)
FOR Year IN ([2015], [2016]) AS PVT
```

The resulting output would be the table below:

YearPivotResult

ProductName	2015	2016
RazorBlades1	12000.00	NULL
BarHandles1	15000.00	NULL
RazorBlades2	NULL	10000.00

BarHandles2	NULL	11000.00

There are more ways for you to use your PIVOT query.

For your UNPIVOT query, it will work the opposite way your PIVOT query does. It will convert the columns into rows.

Example

Using Exercise #1, you can come up with this UNPIVOT query:

```
SELECT ProductName, Year, Earnings
FROM #ProductNamesPivotResult
UNPIVOT (Earnings FOR ProductName IN ([RazorBlades1], [BarHandles1], [RazorBlades2], [BarHandles2]) AS UPVT
```

The output would be the original table:

#ProductSalesUnpivotResult

ProductName	Year	Earnings
RazorBlades1	2015	12000.00
BarHandles1	2016	15000.00
RazorBlades2	2015	10000.00
BarHandles2	2016	11000.00

The different data types that work in sql

Now that we have taken the time to look at some of the commands that you will want to use inside of the SQL language, it is time to take a look at some

of the data types that you can use to make this SQL language work well while you are creating your new code. These data types will change based on what you would like to have done inside that database as well as the different items that you would like to offer or sell to your customers.

The various data types that are most commonly found inside of this SQL system will be the attributes that will go with the information that is on the inside and then these particular characteristics will be placed on a table that you can read through and retrieve any time that you would want to have a look.

Let’s take a look at how this works. A good example is when you require that a field that you want to work with is only able to hold only numeric values. You will find that you can use SQL to set it all up so that the user will only be able to place a number inside of that database, or at least in that particular cell of the table. So, if you have a cell on the table where you only want them to be able to place their phone number or their credit card number, you could add this on as a restriction. By assigning the right types of data to the fields that you would like to have come up inside the table, you are making sure that fewer errors come up on the side of the customer.

One thing that you should think about when you are working with your SQL system is that all the versions that you could work with will be a bit different and you may have to add in some different choices when you are picking out the data that you would like to use. You must check out the rules that come with the version of the SQL you are using to make sure that all of it will stay in order. For most cases, you can use the data points that are specific to the version that you chose to make sure that the database is working the way that you would like.

There are some basic types of data that you can use when you are working with SQL, no matter which version you are working with. Some of the data

types that you can use include:

- **Time and date values**
- **Numeric strings**
- **Character strings**

Characters that are fixed in length

Now it is time to take a look at some of the types of data that you can work with inside of SQL. The first one that you may want to use is the fixed length character. If you want to work with constant characters or even strings that will stay the same all the time, you will need to make sure that you get them saved in the system the right way. The typical data type that you will want to work with when picking out these characters is this sample:

CHARACTER(n)

Look carefully at the example above, the ‘n’ that we placed inside the parenthesis is the maximum length or the assigned length that you will allow that particular field to be. For example, you may use this for the phone number field. You would want to just keep it at ten characters so that the customer will only put a phone number and nothing else inside of it. You could also do this for some of the other parts of the code, but you will need to be careful with this. For example, if you put these restrictions on the name of the customer, they could end up with some issues if their name is longer than the characters that you allow.

Depending on the version of SQL, you may find that it will work with the ‘CHAR’ data type instead, to save information that you will need that is a fixed length. It is a good idea to work with this kind of data type when you would like to work with any information that is alphanumeric. So, if you would like to set up a part that will allow the user to place in their name of their state, but you would like to make sure that they use the abbreviation of

the state rather than the whole name, you would be able to use the character limit and set it at two so that everyone knows how it should work.

When you are working with the character of the data type, your user will not be adding in information that is longer than the character limit that you set. Let's say that you live in South Dakota, but you would like to make sure that the user can only put in the abbreviation you will need to set the character for two. If the user tries to put in more than two characters at this point, it is not going to work for them.

There are many times when you are working with your database where you would want to add a limit to the characters that you and the user can use. But when you are working with the password and the username that your user should be allowed to work with, it is best to not use a fixed character length for these. Let the user pick out what is best for this. Sometimes you may want to set a minimum length to help with security but never set a maximum so that the user has the freedom they need to create a good username and password.

Variable characters

It is also possible that you can work with variable characters rather than the fixed length characters that we talked about above. Instead of telling the user how many characters they are allowed to have, you can allow the user to pick out how long this should be. This one is a good option when you are working with fields such as the name (which can easily reach varying lengths depending on the person) as well as with the usernames or the passwords that the user would like to use to make them unique. If you have a field that you would like to use this way, the notation that you would use includes is this:

CHARACTER VARYING (n)

With the option that we used above, you will see that the ‘n’ will be the number that will identify the maximum or the assigned value. You will then be able to pick out from more than a few different kinds that you would like to put in here to work with your variable characters such as ANSI, VARCHAR, or VARCHAR2.

For this particular data type, there is no requirement that you would have to meet when it comes to filling out the spaces for your user. If you assigned a length that was for 15 characters, the user would be able to add in some less if they would like, and you would not have issues like you probably would with the fixed length characters we talked about above.

Any time that you are interested in working with some character strings that may be considered variables in your language, you should make sure that, as much as you can, you are using varying data types. The reason for this is because it will allow you to maximize the space that you can use in your database and will make it easier for the user to input their information without confusing the database in the process.

Numeric values

Now we will take a look at how these numeric values can work on your code with SQL. It is possible to work with numeric values any time that you would like when you are in the SQL system. These values are the ones that you will store right in the field using a number rather than using a letter. These values will go by some different names depending on which numeric value you would like to use in your code. However, there are some that are more common to work with, and these would include:

- **Bit varying(n)**
- **Double precision**

- **Integer**
- **Float**
- **Bit**
- **Real**
- **Decimal**

Literal strings

Another thing that you can take a look at and work with when you are in SQL is ‘literal strings.’ These will consist of a series of characters, such as a list of names and phone numbers, that will be specified by the user or the database. For the most part, you will see that these strings will hold onto a lot of data which has similar characteristics that go with it.

Any time that you are working with literal strings, you will sometimes run into trouble specifying what kind of data you would like to use. As a result, you will spend your time specifying the type of string that you would want to use to ensure that it all works. It is good to understand that when you work to make these strings, especially if the string will end up being alphanumeric, you will need to make sure that these are quotes that can go around the world. You have the choice to go with either the single or the double quotation marks on this one, you should just make sure that you use it the same on both sides of the word to help the compiler out.

Boolean values

You are also able to work with what are known as **Boolean** values. These **Boolean** values are important because they will perform a lot of different functions inside the SQL system, and that can make it easier to work on some of the searches that you want to do. When you work with Boolean values, there will be three options of what will come up. These three options are true, false, and null.

As you are working with the **Boolean** values, you will also find that using these values can be helpful when you want to compare several units of data inside of your table. You would be able to use those units and figure out if they are matching or not and then will receive the right answer based on the **Boolean** values. For example, you would be able to use this in SQL to specify the parameters of a search, and all of the conditions that come back to you will either be true, false, or null based on the things that you are trying to compare.

With these kinds of values, you should note that it will give you the results only when the answer will be true. If the answer to these values ends up being null or false, the data will not be retrieved out of the database for you or for the user. However, if you end up with a value that is true, you can notice that all the information is true and it will show up for you.

A good way to look at this is when your user is in the database and would like to do a search to find something there. If the keywords of the product match what the user is typing into the search bar, then true answers will show up while everything else will stay away.

If you have your own website and you sell things online, you will find that you will spend a lot of time working with these **Boolean** values. These values will ensure that you can get the right results to show up in your program. So, if the user goes through and types in a keyword that they would like to be able to find inside of the code, the **Boolean** value would be able to look at that and find the results that would fit in.

Don’t worry if this sounds complex, the SQL system is capable of helping you get this done in a simple manner so that your customers will find the things that they want without having to worry or fight with the website.

There are many different applications where you can use some of the **Boolean** expressions in SQL, but as a beginner, this is probably the method

that you will use the most often. You are also able to use it when you want to look through your database and find specific answers to questions, or when you are looking through the store or program and want to find specific information. When you add some of this information into tables into SQL, you can speed this process up more than ever before.

It is important to remember how these Boolean results will work. They are in charge of deciding whether something matches up with what you are searching for. If something does match up with the keywords that you place into the system, then they will show up. But if other objects or items do not match up with this, then they will be shown as false or null and will not show up.

Let's say that you are using a website to sell some of your products, for example, clothing. Someone comes onto your site and wants to look for dresses they can use for special occasions. If they type in the word 'dress' on the search bar, you want to make sure that they can see the products that they want, rather than something else. This means that dresses should show up on the screen after the search is done rather than shoes or accessories.

With the **Boolean** values, this will happen. If the user inputs 'dress' on the system, the **Boolean** value will go through each item and check to see whether they are true to this statement. If the item has the word 'dress' in it, then these will show up on your screen. If the item does not have that keyword with it, the **Boolean** value will see this as a false statement and will not list it up on the screen. As long as the **Boolean** value is working the way that it should, your user can get the items that they want, and only these items will show up on the screen.

This can be incredibly useful for anyone who is selling things online, especially if they have a large inventory to choose from and need the

program, such as SQL, to sort through a ton of data to figure it out. The **Boolean** values will do all the work for you to help prevent issues.

As you can see through this chapter, there are many data types that you can work with inside of SQL to make your searches and your program easier and simpler. Each data type will work differently so that you will be provided the right results and get more than what you need.

Chapter 10: What is data definition language

Now that we have taken some time to learn what SQL is about as well as some of the basics of the databases that business owners will use, it is time to get more in-depth about this system. This chapter will take some time to learn some of the commands that you would need to use to make sure that you get this system to do what you would like.

If this sounds scary in the beginning, do not worry about it. This is an easy language to learn, and the commands will be intuitive to work with. It is nowhere near as complicated as some of the other coding languages that you may want to work with. This chapter will help you out with these commands by splitting them up into six categories so that you understand how each of them works. The six categories of commands that you will use when working with SQL include the following:

Data definition language

This category is known as the 'DDL,' and it is one of the aspects that you need to learn about the inside of SQL. This is in charge of allowing you to generate objects into the database before you arrange them in the way that

will work the best for you. For example, this is the aspect of the SQL system that you should use any time that you would like to make some changes in your table, such as adding or taking away objects. There are some commands that you can use that can help you to see these changes. These include:

- **Create a table**
- **Alter a table**
- **Drop a table**
- **Create index**
- **Alter an index**
- **Drop view**
- **Drop index**

Data manipulation language

The next thing that you can work with inside the SQL language is the data manipulation language. This one is called ‘DML’ when working with SQL, and it is the part that you can use when you are trying to modify the objects that are inside of your database. This will make it easier for your user and yourself to have some more freedom when you are looking through the information in your database and allows you to add in something new that will help the database work better.

Data query language

When you are working with the data query language or the ‘DQL,’ you will work with what many experts believe is the most powerful aspect of what

you can do with the SQL system. If you would like to work with the DQL inside of this program, there is only one command that you will need to make this work. The command that you will need to use to make this work is 'Select.' This is an easy command to work with that will offer you a variety of ways to do things with it, including running queries while you are inside your relational database. If you are interested in looking through some results that are more detailed, you would need to make sure that you are working with this select command through DQL.

Data control language

The data control language is another part of SQL that you should learn how to use when you want to add some more power to the codes that you are writing. This will work out well when you would like to have some more control over who can get access to the database. If your company can deal with the personal information of your customers, such as their credit card information, it is a good idea to use the data control language so that you can place some limits on who can use these databases and access the information.

The DCL command will be used to help generate the objects that you need to control who to access the information in the database so that only the right people can get into the database. You can control who can get onto the database, who can distribute the information that is in the database and so much more. Some of the commands that you would be able to use inside of the DCL include:

- **Revoke**
- **Alter password**

- **Grand**
- **Create synonym**

Conclusion

The manipulation, in this case, refers to the addition of data to the created database and manipulation of the same. It also involves its deletion. This can be attributed to the fact that SQL comes with a simple syntax which anyone can grasp with much ease. You can use SQL to create a database within your database management system. Once the database has been created, you need to create tables within the database so you can use them for storage of your data. This means that tables are the actual data storage structures in a database. The table is a combination of rows and columns that can be used for data storage. This means that the tables organize your data in the form of rows and columns. SQL comes with built-in commands you can use to insert data into the tables. Once the data has been added to the table, you can still manipulate it whenever there is a need. In addition, you can add constraints to the tables so you can restrict the kind and format of data that can be stored in the database tables. The data stored in the tables can be viewed anytime there is a need. You can also delete the data and the various objects you have stored in the database.

The next step is to get started with using SQL as the programming language that you need to take care of all your database needs. There are a lot of different databases out there, but if you want your company to provide good customer service, there isn't a better database to work with than SQL. SQL is fast, has been around long enough that it has a solid reputation, and so much more.

This guidebook has spent some time going over all the things that you will need to know to get started with the SQL language. Whether you are storing some important personal information about the customer on the database or you would like to use it to set up your online store, SQL can help you get started. We will take a look at how to work with SQL how to create tables,

some of the commands that you can use, and so much more. When you are done with this guidebook, you are certain to know some of the basics of working with SQL and can understand some of the basic components that you can work with.

When you are ready to learn more about how SQL works and how you can use it in your own programming to keep information organized and to work on your database, make sure to check out this guidebook to help you out.