Table of Contents

Project Introduction

Can you design, implement, and start using your very own database, all in one course term? You can, even if you have never touched a database in your life! The theoretical concepts and applied skills you learn throughout the course prepare you well for this task, and have been carefully crafted and sequenced for rapid learning. Further, the teaching team stands ready to help fill any gaps, so you are not dependent on the course materials alone. You will undoubtedly put in many hours of hard work on this project, but you will leave the course with your own database, and more importantly, the skills to build other ones.

The start-to-finish experience you gain from this project will help prepare you for database work in the industry or academia. If your boss asks you to develop a database, you'll know how and have the experience to back up your confidence. If you decide to pursue database research, you'll have a solid foundation. Take this project to heart. It will pay off!

The remainder of this document provides you important details on getting started with your project. For this first iteration, you will decide what kind of data you want to capture in your database and provide an overview of your project. Read through this document carefully and ask the teaching team if something is unclear. You will be spending a lot of time designing, implementing, and testing a database for this project.

What Is A Database Used For?

Before you create your own database, we need to address an obvious question: what is a database used for? Most are aware that organizations use databases to store data long term, but may not be aware that databases also provide the critical features of data security and efficient access. Databases ensure that only authorized people can make use of the data, typically with extensive and customizable data security configurations. Databases also ensure that just the right data items can be retrieved quickly (typically in less than a second) from amongst a potentially vast amount of storage. This sets databases apart from file-based data storage.

The next important question is, who, or what, accesses the database in an organization? Broadly, a database is accessed either by I.T. professionals or by applications. I.T. professionals, such as database administrators, database developers, data analysts, and the like, access the database to program it, administer it, or to more directly work with the data it contains through use of business intelligence tools. Such people have more intimate access to the database and usually have a higher level of trust within the organization. Everyone else gets access to the data indirectly through an application. The application directly interacts with the database, and the end users use the application. Whether you realize it or not, you regularly use applications that have a database backend, be they web-based or otherwise. For example, if you use Amazon.com, Ebay.com, and Student Link at Boston University, , you are using a web application that has a database backend. You directly interact with the web application, and the application interacts with the database.

Why does this matter to you? When you're designing your database, you need to keep in mind who or what will use the database and how it will be used. Will it be for personal use? For a club? For your place of employment? For a mobile application? For an existing real-world business (such as Amazon or Ebay)? Only you can answer the who, what, and how for your database. A database is not designed in a vacuum; rather, it is designed intentionally to support who or what will use it, and to support how it's expected to be used.

What You Will Be Creating

Next, you need an idea of what you will be creating for this project, both in this and later iterations. An outline is given below. Unless you have worked with relational databases previously, you will likely not understand all of the terms used at this point. Do not worry. You will learn the terms as you need to develop the items they relate to. Keep in mind that the iterations build on each other, so as you progress through the iterations, you will first revise what you developed in prior iterations, then add additional material. A similar outline is provided in every iteration to help you keep track of what you have completed and will be completing.

Current Iteration	Iteration 1	*Project Direction Overview* – You provide an overview that describes who the database will be for, how you envision it will be used, the focus of what data it stores, and most importantly, why you are interested in it. *Use Cases and Fields* – You provide use cases that enumerate steps of how the database will be typically used, also identify significant database fields needed to support the use case. *Summary and Reflection* – You concisely summarize your project and the work you have completed thus far, and additionally record your questions, concerns, and observations, so that you and your facilitator or instructor are aware of them and can communicate about them.
Future Iterations	Iteration 2	*Structural Database Rules* – You define structural database rules which formally specify the entities, relationships, and constraints for your database design. *Conceptual Entity Relationship Diagram (ERD)* – You create an initial ERD, the universally accepted method of modeling and visualizing database designs, to visualize the entities and relationships defined by the structural database rules.
	Iteration 3	*Conceptual Extended Entity Relationship Diagram (EERD)* – You add specialization-generalization into your conceptual ERD and structural database rules. *Initial DBMS Physical ERD* – You create an initial DBMS physical ERD, which is tied to a specific relational database vendor and version, with SQL-based constraints and datatypes.
	Iteration 4	*Full DBMS Physical ERD* – You define the attributes for your database design and add them to your DBMS Physical ERD. *Normalization* – You normalize your DBMS physical ERD to reduce or eliminate data redundancy. *Database Structure* – You create your tables, sequences, and constraints in SQL.
	Iteration 5	*Reusable, Transaction-Oriented Store Procedures* – You create and execute reusable stored procedures that complete the steps of transactions necessary to add data to your database. *Questions and Queries* – You define questions useful to the organization or application that will use your database, then write queries to address the questions. *Index Placement and Creation* – To speed up performance, you identify columns needing indexes for your database, then create them in SQL.
	Iteration 6	*History Table* – You create a history table to track changes to values, and develop a trigger to maintain it. *Data Visualizations* – You tell effective data stories with data visualizations.

Project Direction Overview

What kind of data should I capture? How should I choose a project direction? Where do I start? You may have many questions like this, but the answer is simple. *Choose something you're interested in*. The exciting part is, you can create a database about whatever you'd like! You will be working with this data a lot during the term, and we want you to enjoy it.

Once you've selected something of interest, you need to explain your choice so your facilitator or instructor will know what you're working toward. Create an overview in your design document which describes who the database will be for, what kind of data it will contain, and how you envision it will be used, and most importantly, why you are interested in it.

Since a database is not used in isolation, your project overview will likely have some references to non-database-related areas such as application functionality or organizational processes. Such references are just fine and expected, as long as they help you and your facilitator or instructor understand your database. Just keep in mind that you will be focused on designing and implementing the database component in this course.

To help guide you, I select my own project in an area of my interest, and give you examples in this and all remaining term project iterations. Mine is only one project, so please do not let my choices limit your creativity or project direction. My project is a database for a mobile application, but you may find that your project is something drastically different than mine, which is expected. Further, you may create a project that is much better than mine. Think of these examples as something that gives you a basic idea of how to proceed, nothing more. First I provide an overview of my project.

TrackMyBuys Project Direction Overview

I would like to develop a mobile app which tracks all purchases someone makes, named "TrackMyBuys". When someone uses this app, they will have their full purchase history at their disposal across all stores. Typically, when a person purchases something online, they can only see their purchase history with that same vendor. If they purchase from 20 different stores, they need to visit 20 different websites to see all of their purchase history. Further, there is no history they can access for when they purchase in a brick-and-mortar store. People like to know what they purchase and when for many reasons, including returns, buying or recommending a similar product possibly many years after the original purchase, and even tracking down warranties when something breaks. TrackMyBuys will store purchases made across all stores, making it the one stop for any purchase history.

Here is are some brief examples of how someone would use the application. Aria makes many online purchases. A couple weeks ago, she purchased a throwable camera from Amazon, and a virtual reality headset from EBay. Using a browser extension, TrackMyBuys automatically recorded each purchase. Aria also visited a nearby Old Navy store and purchased a sweater, which she then manually recorded in the app. Today, her friend Brianna told her she really likes the throwable camera as well as the sweater, and wants to buy them herself. Aria quickly looks up her purchases in TrackMyBuys on her smartphone, then forwards the information onto Brianna. If Aria ever needs to lookup her other purchases in the future, TrackMyBuys will be at her disposal, even as many years pass by.

While obviously there will be a significant programming component for TrackMyBuys, for this course I am focusing on the database component. The database will store the names and locations for the stores purchased from. It will record the details of each purchase such as the date and price of the purchase. Since TrackMyBuys will be used by many people, the database will need to associate the purchases with different accounts. Since I plan to charge for premium features, there will need to be some kind of billing information. I'm sure there are details I'm not thinking of this week, but these will become apparent in future weeks.

Notice that in the overview, I did not attempt to provide every detail, but rather provided a high-level summary. The summary was complete enough that it's clear what the application and database will do, but left the details to be itemized in other sections of the design document.

Use Cases and Fields

For this component of your design document, you define use cases that enumerate steps of how the database will be typically used. In software engineering, a use case is a list of steps defining interactions between a person and the system. Uses cases are commonly used to help formally describe an application. This effort will help you and your facilitator understand how a person or application interacts with your database. The focus should be on how your database will be typically used, not on edge cases or insignificant areas.

For each use case, you also identify significant database fields needed to support the use case. This helps you determine how the use case affects the database specifically, as opposed to how it affects the general system or application. Give each field a reasonable name, explain what kind of data it stores, and why such a field is needed to support the use case. Do not worry about discerning every last field; this will be detailed in other sections of the design document. Here you are identifying the significant fields that the database will clearly need to store to support the use cases.

Define *at least five* use cases for this iteration. You may ultimately add more to your design, because you may envision additional use cases as further define your database in later iterations; however, at least five use cases is a good place to start, to define the core uses of your database.

Here are some use cases and fields for TrackMyBuys.

TrackMyBuys Use Cases and Fields

One important usage of the database is when a person signs up for an account and installs the application.

Account Signup/Installation Use Case

1. The person visits TrackMyBuys' website or app store and installs the application.
2. The application asks them to create an account when its first run.
3. The user enters their information and the account is created in the database.
4. The application asks them to install browser plugins so that their purchases can be automatically tracked when they make them.

From a database perspective, this use case requires storing information about accounts (from steps #2 and #3). Steps #1 and #4 apply to the user and application but not the database directly. Significant fields for an account for this application are listed in the table below.

Field	What it Stores	Why it's Needed
AccountName	This field stores a summary name associated with each account,	In case the same person has more than one account, they can select the correct one from a dropdown list.
FirstName	This is the first name of the account holder.	This is necessary for displaying the person's name on screens and addressing them when sending them emails or other communications.
LastName	This is the last name of the account holder.	This is necessary for displaying the person's name on screens and addressing them when sending them emails or other communications.
CustomerSince	This is the date the account what was created.	This is useful for determining the age of an account, which is useful for a variety of reasons, such as giving discounts to long-term customers, customer retention, etc …
AccountBalance	This is the balance owed by the customer.	This is useful to keep track of customers that owe money to use the application.

Another important usage of the database is when an online purchase is made and automatically recorded in TrackMyBuys via browser extensions.

Automatic Purchase Tracking Use Case

1. The person visits an online retailer and makes a purchase.
2. The TrackMyBuys browser plugin detects that the purchase is made, and records the relevant information in the database such as the purchase date, price, product, store, etc.

Step #2 highlights that the database will store relevant information about an online purchase. Significant fields are detailed below.

Field	What it Stores	Why it's Needed
PurchaseDate	This is the date of the online purchase.	This is useful so that the person knows on what day they made the purchase, and so that they can search for purchases by date.
PurchasePrice	This is the total cost of the purchase.	This is useful so that the person knows how much money they spent on a purchase, so they can search for purchases by price.
Product	This is the name of the product that was	This is necessary for so the person knows what product they bought.

	purchased.	
Store	This is the name of the store where the purchase took place.	This is necessary so the person knows at what store they made the purchase.
URL	This is the URL where the purchase took place online.	This is useful so the person could go back to the same online store and know what website they were visiting when they made the purchase.

Another important usage is when a person decides to lookup past purchases.

Purchase Lookup Use Case

1. The person signs into TrackMyBuys.
2. The person selects the option to lookup past purchases.
3. TrackMyBuys pulls a short list of recent purchases from the database, and also gives the user the option to search.
4. The user searches based upon a date range, which causes a database search.
5. The application pulls all purchases matching the criteria from the database.
6. The user selects the purchase they are interested in.
7. TrackMyBuys displays all recorded information about the purchase.
8. At that point, the user can close out of the purchase, or share the purchase by selecting the Share option.

The database would make use of the fields already highlighted for the second use case, Automatic Purchase Tracking Use Case. This use case helps us understand more of how the user and application would interact with the database, but does not appear to introduce additional database fields beyond the second use case.

Notice that after reading through the use cases for TrackMyBuys, you have a better idea of how the database will be used, how the application will interact with the database, and what data the database will store.

Summary and Reflection

At this point, you and your facilitator or instructor have an idea of what kind of database you are creating, which is great! To tie things together, concisely summarize your project and the work you have completed in your design document. Your summary need not be long. Express briefly the main purpose of your project and a few significant details, to tie together the ideas you have for your project.

Additionally, you may have some questions, concerns, and other observations. Take some time to write these down in your design document, so that you and your facilitator or instructor are aware of them and can communicate about them.

Here are is a summary of my TrackMyBuys project with some reflections.

TrackMyBuys Summary and Reflection

My database is for a mobile app named TrackMyBuys which records purchases made across all stores, making it the one stop for any purchase history. Typically, when a person purchases something, they can only see their purchase history with that same vendor, and TrackMyBuys seeks to provide a single interface for all purchases. The database must support a person entering, searching, and even analyzing their purchases across all stores.

As I think about the use cases and fields defined in other sections of the design document, I realized that TrackMyBuys may need to store all kinds of different data I did not originally envision. For example, when I start down the path of billing people for using the application, there might be a lot of information to store such as addresses to send bills to, payment and credit card information, balances, payment history, and the like. I am a little daunted by the possible size and complexity of the database, and may need some help focusing on the areas I can complete during the course term.

This concern notwithstanding, I am excited to continue developing this project and hope to turn TrackMyBuys application into a real, working mobile application. Any feedback on making this better is appreciated.

Items to Submit

In summary, for this iteration, you create a database design document which contains the following items. Make sure to use the template provided with this iteration to ensure you are submitting all necessary items.

Component	Description
Project Direction Overview	Provide an overview that describes who the database will be for, what kind of data it will contain, how you envision it will be used, and most importantly, why you are interested in it.
Use Cases and Fields	Provide at least five use cases that enumerate steps of how the database will be typically used, and also identify significant database fields needed to support each use case.
Summary and Reflection	Create a concise summary of your project and the work you have completed thus far, and additionally record your questions, concerns, and observations, so that you and your facilitator or instructor are aware of them and can communicate about them.

Evaluation

Your iteration will be reviewed by your facilitator or instructor with the criteria outlined in the table below. Note that the grading process:

- involves the grader assigning an appropriate letter grade to each criterion.
- uses the following letter-to-number grade mapping – A+=100,A=96,A-=92,B+=88,B=85,B-=82,C+=88,C=85,C-=82,D=67,F=0.
- provides an overall grade for the submission based upon the grade and weight assigned to each criterion.
- allows the grader to apply additional deductions or adjustments as appropriate for the submission.
- applies equally to every student in the course.

5 points per day will be subtracted for late submissions. Submissions beyond 5 days late will not be accepted. Please contact your facilitator for any exceptions.

Criterion	A	B	C	D	F
Project Direction Focus and Completeness (30%)	The project direction overview completely and clearly explains who the database will be for, how it will be used, and the focus of what kind of data it stores.	The project direction overview mostly explains who the database will be for, how it will be used, and the focus of what kind of data it stores.	The project direction overview partly explains who the database will be for, how it will be used, and the focus of what kind of data it stores. Some aspects of the overview may be unclear or ambiguous.	The project direction overview minimally explains who the database will be for, how it will be used, and the focus of what kind of data it stores. Much of the overview is not clear.	The project direction overview is missing or does not explain who the database will be for, how it will be used, and the focus of what kind of data it stores.
Use Case Quality (30%)	The use cases are all focused on core uses of the database, and are completely intelligible. Each use case coherently describes one overall activity. At least 5 use cases have been defined.	The use cases are mostly focused on core uses of the database, and are intelligible. Each use case describes one overall activity. 1 use case (out of 5) may be missing.	The use cases are somewhat focused on core uses of the database, and are somewhat intelligible. 2 or 3 use cases (out of 5) may be missing.	The use cases overall are focused on less important uses of the database. They may be hard to understand. Each use case may describe many activities. 4 use cases (out of 5) may be missing.	The use cases are either missing or are entirely focused on unimportant uses of the database. They may be hard to understand. Each use case may describe many activities.

Field Quality (30%)	The fields are entirely focused on long-term storage of information. Most significant fields have been identified for every use case. The choice of fields is well justified. Fields have been defined for at least 5 use cases.	The fields are mostly focused on long-term storage of information. Some significant fields have been identified for every use case. The choice of fields is justified. Fields for 1 use case (out of 5) may be missing.	The fields are partly focused on long-term storage of information. A small percentage of significant fields have been identified for every use case. The choice of fields is partially justified. Fields for 2 or 3 use cases (out of 5) may be missing.	The fields are mostly focused on transient information that is not needed long-term. Many significant fields have not been identified. The choice of fields is minimally justified. Fields for 4 use cases (out of 5) may be missing.	The fields are missing, or may be entirely focused on transient information that is not needed long-term. Many significant fields have not been identified. The choice of fields is not justified.
Summary Quality (10%)	The summary concisely and clearly summarizes the work for Iteration 1.	The summary largely summarizes the work for Iteration 1 in a clear and concise fashion.	The summary summarizes the work for Iteration 1 in a somewhat clear and concise fashion.	Some significant work for Iteration 1 may not be summarized. The summary may be unnecessarily wordy or unclear.	The summary is missing, or does not summarize virtually any significant work for Iteration 1. The summary may be wordy and unclear.

Use the **Ask the Teaching Team Discussion Forum** if you have any questions regarding how to approach this iteration. Make sure to include your name in the filename and submit it in the *Assignments* section of the course.