Fall 2005

The Cherry Creek Sneak a Database Solution

Analissa Hilt
Regis University

Follow this and additional works at: https://epublications.regis.edu/theses
Part of the Computer Sciences Commons

Recommended Citation
Hilt, Analissa, "The Cherry Creek Sneak a Database Solution" (2005). All Regis University Theses. 387.
https://epublications.regis.edu/theses/387

This Thesis - Open Access is brought to you for free and open access by ePublications at Regis University. It has been accepted for inclusion in All Regis University Theses by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.
Use of the materials available in the Regis University Thesis Collection (“Collection”) is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the “fair use” standards of the U.S. copyright laws and regulations.
THE CHERRY CREEK SNEAK A DATABASE SOLUTION

BY: ANALISSA HILT
I. Chapter 1: Introduction
   1. Project history
   2. Project purpose
   3. Project constraints and challenges

II. Chapter 2: Project Need
   1. Cherry Creek Sneak history
   2. Technological need
   3. Student introduction to the project
   4. Project vision
   5. Project plan

III. Chapter 3: Database Management Systems
   1. Data in database management systems
   2. Conversion of data to information
   3. Establishment of database management system
   4. Historical database management
   5. Entities in a database management system
   6. Database modeling
   7. Normalization in a database

IV. Chapter 4: Systems Development Life Cycle
   1. Introduction to the systems development life cycle
   2. Planning phase
   3. Analysis phase
   4. Detailed systems design phase
   5. Implementation phase
   6. Maintenance phase
   7. Summary systems development life cycle

V. Chapter 5: Database Life Cycle
   1. Introduction to the database life cycle
   2. Initial phase
   3. Design phase
   4. Implementation phase
   5. Evolution phase
   6. Summary database life cycle

VI. Chapter 6: Capability Maturity Model
   1. Introduction to the capability maturity model
   2. Initial level
   3. Repeatable level
   4. Defined level
   5. Managed level
   6. Optimized level
   7. Summary of the capability maturity model

VII. Chapter 7: FAST Methodology
   1. Introduction to FAST
2. Involvement of the users and owners
3. Implement a problem solving approach
4. Establish phases and activities
5. Establish standards
6. Project justification
7. Allow for modification of project scope
8. Divide and conquer
9. Design for growth and change
10. Summary of the FAST methodology

VIII. Chapter 8: Project Management
  1. Total project management with SCRUM
  2. Student project management

IX. Chapter 9: Project Summary
1. Project Introduction

1.1 Project History

This project encompasses the creation of a database through application of knowledge gained in the Masters of Information Technology program at Regis University. This paper will demonstrate the knowledge gained at Regis University and the application of that knowledge during database creation. The database designed is a race registration database for an organization known as the Cherry Creek Sneak. This paper will discuss the purpose of this project, history of the project, database management systems, database design, systems and database life cycles, development methodology and management techniques as they apply to the student project.

At the time of this writing, the Cherry Creek Sneak, to be referred to as “The Sneak”, and Regis University had worked together for the previous three years to determine methods in which The Sneak could improve race awareness and increase participation while allowing Regis students the opportunity to apply the skills acquired in the various programs within the MSCIT degree program. The vision of the future for The Sneak was a fully automated on-line registration system, bar coding of race results to reduce lag time to the result posting, and reporting of racer information to various sponsors. Several technologically-based projects evolved as a result of this vision. One such project was the creation of a multi-purpose race database for The Sneak.
1.2 Project Purpose

The purpose of the database was to provide a repository of information about the race, the racers, the sponsors and other miscellaneous information that may in the future prove to provide a better understanding of the customers of The Sneak. This project was completed with the long-term goals that the information contained in the database will be utilized to reduce registration time, recruit prior year participants to participate in future events and provide a tool to intimately understand the customers of the race.

The student responsible for this project was held accountable for the design and implementation of the race information into a newly designed database. The design phase included design elements that are flexible enough that it can be utilized for future foreseen and unforeseen technological requirements. Additionally, the student’s research determined whether pre-packaged database software could be utilized or if the database should be custom designed to fit the needs of The Sneak. The implementation phase included writing all code required to generate the database and any code required to create reports or other analysis tools. Finally, during the conversion phase a handoff occurred of the database to another student for conversion of previous year’s race data and loading that data into the newly created race database.

1.3 Project Constraints and Challenges
The primary project constraint was monetary funds as there was no money budgeted for the project. This posed great challenges in acquiring a database platform, software and hardware required to create and maintain a large database. The next challenge was coordination. The main contact person with The Sneak was available only on occasion and resided outside of Colorado. These constraints made meeting to discuss the database requirements and progress a challenge. Another major challenge in creating the database was communication. The contact person with The Sneak is not a technical person. As such it was difficult to determine the requirements of the database. It was challenging to communicate what tables and fields should be contained in the database or what database structure would work best for the needs of The Sneak. As an example, an issue that was raised was that The Sneak needed to identify racers that were a member of a family. The student needed to determine whether it would be best to make the individual table a recursive table in order obtain this information or if a separate table should be created to hold family information. This technical issue was difficult to communicate to The Sneak representative to obtain their opinion as to how they would prefer to have the database built as this design may have affected other information not predicted by the student. The student determined that a separate family table would be the best option. This option was chosen to reduce the overall complexity of the database structure and provide more efficient query performance.
Another challenge faced by the student was how to define the audience or user group that would eventually be using the information system based upon their level of interest. The Sneak stakeholders or people with an interest in the information system could be considered either very small or quite wide. The small group of stakeholders would include the system owners (or The Sneak management team), the system designer, system builder, system analyst and a consultant (also known as “the student”). Additional stakeholders would include the actual race participants, sponsors, vendors or other system users. (Whitten, Bentley, Dittman 2001, p. 51) The most prominent system owner for The Sneak was Paula Tiernan. During the time of this project, Ms. Tiernan was the management lead for The Sneak project and advocate for all of The Sneak’s database requirements.

The Sneak required a database with transactional processing capability. Transaction processing systems allow the capture and processing of data. (Whitten, Bentley, Dittman 2001) In the case of The Sneak the transactional processing data is the racer registration information. The transactional processing system required internet capabilities. The ability to register online was one of the key goals of The Sneak project.

An additional goal of this project paper is to present the level of knowledge the student has obtained in the duration of the program at Regis University. This knowledge will be presented in the form of
research performed and activities performed in conjunction with the project in order to build an effective efficient database product to supply to The Sneak.

The paper is constrained by length and by topics presented. The information presented within the paper should fall within the category of academic learning as applied to the project, as too much information would jeopardize the length constraint. Knowledge gained through completion of this project should be clearly outlined throughout the paper.
2. Project Need

2.1 Project History

The Cherry Creek Sneak race began in 1982 and was founded by the Bank of Cherry Creek. The race began as a small five mile race for local residents. By 1998, the race had over 14,000 race participants, making that year the best racer turn out in the history of The Sneak. The once-small bank race was quickly growing too large for the Bank of Cherry Creek to manage. The bank determined it could no longer operate The Sneak effectively as it was beginning to negatively impact their main business. This problem had become quite obvious as bank tellers were spending more time completing various tasks for the race than they were performing operational banking activities including assisting bank customers.

Management of the race was moved to a professional firm familiar with handling special events and promotions. However, keeping the race viable and making it a worthwhile endeavor proved extremely challenging, even for the professional firm. In 1999, participation dropped to 9,000 racers. The drop in participation was attributed to the Columbine tragedy, which occurred just days before the race. Since 1999, participation in The Sneak has hovered between nine and ten thousand racers and has not shown any significant growth. The operators of The Sneak determined that in order for The Sneak to continue as a going concern, it must reach fifteen to twenty-five thousand racers by the year 2010. Increases in race
participation are key to recruiting sponsors. The sponsors are the key to the economic survival of the races. The monies paid by race participants for registration do not even fully cover the cost of the race. The true monies earned are attributed to the sponsor donations.

2.2 Technological Need

At this point, The Sneak management team realized the need for a technological advantage in recruiting racers in order to achieve the race participation goal. The participation in The Sneak did not grow significantly during the same time a similar race, the Bolder Boulder, reached a record 45,000 race participants. The Sneak management team realized that the increases realized by the Bolder Boulder were due to year-round contact with the race participants. These contacts were made through count-down emails, organized race training groups and a newsletter for the race. This type of contact was nearly impossible for The Sneak due to the lack of data integrity in the historical race database.

The management team of The Sneak did not have any technological experience; however, they realized the importance of technology and the need for technology with The Sneak. In 2001, The Sneak began to work with Regis University to determine how Regis could assist The Sneak in creating a technological vision and achieving their long-term and short-term technological goals. The student became involved in the project in the fall of 2003 as possible available projects
were posted and emailed to Regis graduate students within the MSCIT program.

2.3 Student Introduction to the Project

The initial meeting of the student and The Sneak management occurred in December of 2003. It became apparent very quickly that The Sneak was a virtual company with virtually no technology. The “database” in which the race participant information was housed was an Excel spreadsheet. The spreadsheet was not version controlled. The Sneak did not maintain a central shared directory for the race file, instead the spreadsheet was emailed back and forth among company management team members. This email system provided no data controls. As such the spreadsheet had no true data integrity.

There was no standard platform for technology as each member of the management team selected their personal computers with no standard configurations being utilized. The company was virtual in the sense that the management team did not maintain any office space and its members resided in various locations across the United States. The main contact from The Sneak management team lived in Kansas City and traveled to Colorado on a bi-weekly basis.

2.4 Project Vision

The initial project discussion involved a vision of the development and design of a database including data conversion and a web interface. The Sneak had hopes of implementing and rolling out of the new database
system by January 2004 for the beginning of racer registration for the April 2004 Cherry Creek Sneak. Unfortunately, this time frame was far too short for the level of effort required to get the database up and running. The goal was then modified to create a fully functional database with web capabilities in time for 2005 racer registration which is scheduled to begin in December 2004. This modification allowed ample time to create a solid database platform for The Sneak.

2.5 Project Plan

The student worked with a faculty representative of Regis University, Trisha Litz, and Paula Tiernan from The Sneak. Overall project management of the end to end technology solution for The Sneak was to be handled by Ms. Litz. The student was responsible for managing the design and development of a racer database on behalf of The Sneak.

The total project was managed by a philosophy known as scrum and the student project was managed by a philosophy know as waterfall or cascade. Scrum is a new development management technology (at the time of this writing) that is based on the idea that shorter smaller meetings would be more effective for technological based projects.

A detailed review of the management philosophies utilized will be detailed later in this paper. As the need for the database project has been outlined and the history of The Sneak has been reviewed, the remaining sections of this paper will review basic database principles and how those principles were applied to The Sneak. It is important to
understand the state of The Sneak prior to creation of the database to fully understand all of the future advantages that creation of the database will provide The Sneak. The next chapter will provide a review of database management systems and how they work in conjunction with databases to provide the most efficient, accurate information to interested parties.
3. Database Management Systems

3.1 Data in Database Management Systems

The Sneak needed a good data management system in order to fully understand the data related to The Sneak and turn that data into information. Data management systems allows for the generation of data, storage of data and retrieval of stored data. (Rob, Coronel, 2002, p.7)

The Database Systems Design, Implementation and Management text had an excellent perspective on data:

- Data constitutes the building blocks of information.
- Information is produced by processing data.
- Information is used to reveal the meaning of data.
- Good, relevant, and timely information is the key to good decision making.
- Good decision making is the key to organizational survival in a global environment.

These concepts could be applied to The Sneak. The data would be the raw facts associated with The Sneak. These raw facts would include racer first name, racer last name, race participant finish time and event date. The processing of data is the recording of the racer registration information and sponsor information as it relates to The Sneak. Race registration provides information such as race participant address information, participant age and sex. This data also includes data regarding the event itself, such as location, weather, start time and
volunteer information. Finding the meaning of the data is the purpose of information. This meaning could show, for example, why participation fell in 2004. Was it due to poor race day weather and the snow received earlier in the week?

3.2 Conversion of Data to Information

After the system is fully implemented, the data gathered will provide information as to who won the race and individual ranking within the race. The resulting information being received on a timely basis is key to The Sneak, as the racers expect race time results as quickly as possible. The individual racer times provide less value the longer the individual racer waits for the result information. The Sneak can also utilize information obtained to determine race decisions, such as how to properly target racers based upon demographics obtained from the previous year’s race.

The total information obtained was:

- Individual racer details
- Sponsor details
- Registration information
- Race results times
- Race vendor information
- Race team information
- Lost and found information

This information is composed merely of pieces of data put together in a meaningful way. Information provides meaning to data. This is the
main concept behind a database. A database is defined as a shared computer system that maintains a collection of raw data and “metadata” which is data about data. (Rob, Coronel, 2002, p. 7) The goal of this project was to create a true database for The Sneak, allowing both race management and racers to assemble pieces of data information through forms, database queries, and reports (all of which will be discussed later).

3.3 Establishment of a Database Management System

The next step once a database was created was to establish a database management system. Database management systems (DBMS) allow the sharing of data within a database among multiple applications or users through programs that control and manage access to the data stored in the database. (Rob, Coronel, 2002, p. 7). A solid database management system would allow The Sneak to share data across personal computers and across the country without jeopardizing the accuracy of the data, unlike the historical Excel-based system. Additional benefits of a database management system for The Sneak include the ability to effectively and efficiently retrieve data and change the data into usable information.

The database management system would provide for the creation of ad hoc queries to retrieve data. These are data inquiries created on the fly. An example of an ad hoc query would be “How many women between the ages twenty and thirty ran in the 2004 Cherry Creek Sneak five mile run?”. Such accessibility to near real-time data on an ad hoc basis would
allow The Sneak management team to make faster, better informed decisions regarding The Sneak and its operations.

3.4 Historical Database Management

Management of The Sneak refers to the previous years race information as being stored in a “database”. However, the true label for the data storage system utilized would be a “file system”. A file system is simply a data repository. There is no logical order of data and file systems almost always contain redundant data and/or duplicate data. In modern database design practice, redundant data is considered poor database design. (Rob, Coronel, 2002, p. 16-17) Redundant data occurs when the same data about the same entity is held in different storage locations. (Rob, Coronel, 2002, p. 16-17) Data redundancy leads to problems updating data and assuring data accuracy.

3.5 Entities in a Database Management System

An entity is the “what” of data stored. An entity is the person, place or thing that the data is about. (Rob, Coronel, 2002, p.119) As an example, the race participant is an entity; the data would be the race participant’s address information. When the address information is stored multiple times and requires an update in multiple locations the data is redundant.

3.6 Database Modeling

In order to retrieve accurate information utilizing a database management system a database must be generated. The first step to
generation of the database is database modeling. Database modeling allows for a review of business processes and how those processes can best be organized into a database structure. The modeling process generally begins with a high level entity relationship model or entity relationship diagram (ERD). The creation of the entity model reduces the complexity of the data and aids in database design. (Rob, Coronel, 2002, p. 119) The model provides a graphical depiction of the entities and the relationships among entities. The primary entities found in The Sneak were the race participant, the sponsor and the race event. Other entities were noted such as the neighbors to The Sneak race and lost and found items. The neighbor entity maintained in the database is considered a secondary separate entity.

An entity relationship diagram contains an entity, or the person, place or thing which is referred to as a singular element. (Rob, Coronel, 2002, p.119) The Sneak database refers to the race participant as an individual, the sponsor is referred to as a sponsor and the race itself is referred to as an event. Event was utilized to allow flexibility for this database to house information about other events in addition to The Sneak race. In the entity relationship model these entities are shown as rectangles. The rows in a database table are known as entity instances. (Rob, Coronel, 2002, p.119)

The second segment of an entity relationship diagram is the attributes. An attribute is a characteristic of the entity. (Rob, Coronel,
2002, p.119-124) The individual or racer in The Sneak database may have the following attributes:

- First name
- Last name
- Date of birth
- Social security number
- Address
- ...(other fields)

These attributes are found in the table as a column of the tables held in the database.

The final segment of the entity relationship diagram is the relationship. This is the relationship among data in the database or how the entities are associated. The relationship among data is represented by a diamond shape within the entity relationship diagram, when utilizing a Chen diagram. (Rob, Coronel, 2002, p. 37) Peter Chen created the Chen model for entity relationship diagrams. Dr. Chen has been a professor with Louisiana State University in the Computer Science department since 1983. (Chen) The entity-relationship model paper by Dr. Chen was created in 1977 and described a new methodology for data modeling. The paper described relationships among entities. The relationships could be described as one-to-one, one-to-many or many-to-many relationship. These relationships were described through cardinality symbols. (Essential Strategies)
The entity relationship model can be created through verbal means as well as graphical.

An example of written model would be:

*The individual participant runs in the five mile run event.*

In a Chen model this would be shown as:

![Diagram of entity relationship model]

See appendix A for entity relationship diagrams for The Sneak utilizing the Crow’s foot methodology.

The entity relationships have been identified and the next step to designing the database is obtaining the relevant data and beginning to normalize that data into a table structure.

3.7 *Normalization in a Database*

E.F. Codd (1923-2003) is considered the inventor of the modern day database due to his paper regarding database normalization. Dr. Codd’s normalization model is based on abstract and complex mathematical theory. (Krieger, 2003). Normalization is key to the entity relationship model as it allows the assignment of attributes to entities. The key advantage to normalization is the reduction of data redundancy and thus removal of data anomalies. Many factors must be considered when normalizing a database. A greater level of normalization increases the number of table joins, thus slowing data retrieval speed. A database that
is not at a high enough normalization level will have significant data redundancy, which will risk data integrity. Most modern day databases are at a normalization level somewhere between two and three. (Rob, Coronel, 2002, p. 188-191)

Database normalization is segmented into five normal forms according to the Boyce-Codd definition of normalization. First normal form requires that all key attributes are defined, there are no repeating groups and all attributes are dependent upon the primary key. (Rob, Coronel, 2002, p. 179-182) The denormalized data from The Sneak would contain data similar to the following prior to the creation of the normalized database:

<table>
<thead>
<tr>
<th>Racer Name</th>
<th>Race</th>
<th>Bib #</th>
<th>Age</th>
<th>Time Finish</th>
<th>Race Year</th>
<th>Address</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>5K run</td>
<td>999</td>
<td>28</td>
<td>35:01</td>
<td>2003</td>
<td>100 Main Street</td>
<td>M</td>
</tr>
</tbody>
</table>

The first step to normalization is to generate a primary key identifier. The primary key may have been set as the bib number however; this number may change from year-to-year. Therefore, a new identifier has been created as the primary key -- the Racer ID.

<table>
<thead>
<tr>
<th>Racer ID</th>
<th>Racer Name</th>
<th>Race</th>
<th>Bib #</th>
<th>Age</th>
<th>DOB</th>
<th>Time Finish</th>
<th>Race Year</th>
<th>Address</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Smith</td>
<td>5K run</td>
<td>999</td>
<td>28</td>
<td>1/10/73</td>
<td>35:01</td>
<td>2003</td>
<td>100 Main Street</td>
<td>M</td>
</tr>
</tbody>
</table>

There are no repeating groups in the table and all attributes are dependent on the primary key. Thus, the data was successfully converted
to first normal form according to the Boyce-Codd definition. (Rob, Coronel, 2002, p. 179-182)

The next step was to move the database to second normal form. Second normal form is achieved when the table is in first normal form and there are no partial dependencies. (Rob, Coronel, 2002, p. 182-183) In order for second normal form to be achieved, the single table above was dissected into three tables: the race participant table, the race table and a registration table. This dissection process removed the partial dependencies.

Race Participant Table:

<table>
<thead>
<tr>
<th>Racer ID</th>
<th>Racer Name</th>
<th>Bib #</th>
<th>Age</th>
<th>DOB</th>
<th>Time Finish</th>
<th>Address</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Smith</td>
<td>999</td>
<td>28</td>
<td>1/10/73</td>
<td>35:01</td>
<td>100 Main Street</td>
<td>M</td>
</tr>
</tbody>
</table>

Race Table

<table>
<thead>
<tr>
<th>Race ID</th>
<th>Race</th>
<th>Race Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5K run</td>
<td>2003</td>
</tr>
</tbody>
</table>

Registration Table

<table>
<thead>
<tr>
<th>Reg. id</th>
<th>Race ID</th>
<th>Racer ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

The Sneak tables are now segmented by entity. A table has been created for the race participant entity, a table for the race entity and a table to assist in the relationship between them. Also note that primary keys were created with the new database tables required.
The final normal form utilized for The Sneak database is third normal form. Databases are generally maintained somewhere between second and third normal form. A cost benefit analysis is required to determine if the database efficiency is increased through greater normalization.

In third normal form, the database must meet all requirements for second normal form and contain no transitive dependencies. A transitive dependency occurs when an attribute is dependent upon a non-key attribute. (Rob, Coronel, 2002, p. 183-185) In the case of The Sneak, database age was dependent on date of birth and the race date. The transitive dependency was corrected in The Sneak database by removing age and simply calculating the participant age based on date of birth and race date when required. Dr. Codd describes more extensive normalization rules; however, in practice these are rarely used and are not considered for this project. Greater normalization of the database has been determined to not improve database performance significantly.

This chapter has demonstrated how a database can be utilized to convert raw data into usable information. This information can be utilized to better understand effects on data and how to utilized data as information to improve operations as a whole. This process began with the database design including creation of database models and normalization of data into a more organized structure. The modeling technique utilized for The Sneak project was the entity relationship model.
This model was used to segment entities into database tables which were then normalized using the Boyce-Codd methodology. The next chapter will review the Systems Development Life Cycle and how The Sneak project came to maturity.
4. Systems Development Life Cycle

4.1 Introduction to the Systems Development Life Cycle

During their development, information systems go through a life cycle. This life cycle is referred to as the Systems Development Life Cycle (SDLC). This is the ten-thousand degree view of the information system. The key components of the life cycle are: planning, analysis, design, implementation and maintenance. (Rob, Coronel, 2002, p. 322-325) As of this writing, the overall information systems plan for The Sneak (of which this project was only one part) is still in the early stages of the System Development Life Cycle. In contrast, various projects in The Sneak’s technological plan are at different stages of this life cycle. The inception of the project documented in this paper brought The Sneak to the planning stage. The completion of the project led The Sneak to the end of the design phase. This chapter will walk through the steps of the System Development Life Cycle and the life cycle stages for The Sneak.

4.2 Planning Phase

The Planning phase consisted of a review of the objectives of the organization. (Rob, Coronel, 2002, p. 322-323) Before this project began, the operators of The Sneak had already answered main questions in regards to SDLC planning. It was determined that The Sneak could no longer operate with its legacy system. Modification of the legacy system was not an option as it would not provide the flexibility required. As such, it was determined that the legacy system required total replacement. The
planning phase requires a determination of feasibility. The feasibility analysis for The Sneak included hardware and software review and cost benefit analysis. The Sneak would require new hardware; however, as a web-based project, it could utilize a system shared with other Regis projects. The database software required is freeware that is relatively simple to implement. The overall project cost was determined to be minimal as the development time is donated through Regis University. The hardware required is a low-cost shared system and the software utilized is freeware.

4.3 Analysis Phase

The analysis phase of the system development life cycle required an examination of organizational needs and user requirements. (Rob, Coronel, 2002, p. 323-324) The Sneak required a system that would allow for virtual access to racer information that was accurate, timely and organized in an efficient manner. Accuracy of the race data is difficult to achieve due to the redundant data maintained in the Excel spreadsheet. Race information year to year was just added to the previous year Excel spreadsheet. As such, the same racer information could appear in the spreadsheet as many as twenty-two times or once for each race year.

Accurate information was difficult to obtain as racers may register their race information via mail, store entry, on the race day or on-line. As an example, the racer manually completes a form at a store location. The race participant information is later entered into the database by a third
party vendor. The registration information may be incorrectly input into the Excel spreadsheet. Previously, the key component of the race has been the posting of race results. These race results have taken up to almost two weeks to post in prior years. The information was critical as many participants use this information to qualify for other larger races. The information is needed as quickly as possible to meet other race registration deadlines. As The Sneak participants tend to be competitive race runners the final race times and standings are considered critical information. If The Sneak is not able to provide accurate and timely race result information the participants will not return in the next race year. Thus, this information must be accurate and timely in order to have repeat race participation. The analysis phase also included the development of data flow diagrams and entity relationship diagramming. (Rob, Coronel, 2002, p. 323-324) The Sneak entity-relationship diagram can be found at Appendix A. The purpose of the diagrams is to compare the theoretical model to the processes. (Rob, Coronel, 2002, p. 119)

4.4 Detailed Systems Design Phase

The next phase is the detailed systems design phase. This phase of the system development life cycle is the phase in which technical specifications are generated. Additionally, a go/no-go decision on the technological change is made during this phase. (Rob, Coronel, 2002, p. 324) The completion of the detailed system design phase should include approval from the sponsoring management team for the project.
4.5 Implementation Phase

Implementation was the final phase that The Sneak database achieved during this project. The implementation only included the creation of the database and not full implementation of the database. Generally implementation will include: installation of hardware and software along with design implementation. (Rob, Coronel, 2002, p. 324-325) For the purposes of this project only database design implementation will be achieved.

4.6 Maintenance Phase

The final phase of the system development life cycle is maintenance. During this phase the system is fully operational and changes to the system are requested and prioritized. (Rob, Coronel, 2002, p. 325) The changes may be in one of three categories: corrective, adaptive or perfective. The corrective maintenance items are those that are required due to system errors or bugs encountered. Adaptive maintenance items are changes made to the system to meet the needs of a changing business environment. The final maintenance type perfective is changes made to enhance the operating capabilities of the current system. (Rob, Coronel, 2002, p. 325) The maintenance phase of The Sneak database will occur in future projects with The Sneak.

4.7 Summary of the Systems Development Life Cycle

Systems Development Life Cycle stages include: planning, analysis, design, implementation and maintenance. Through the planning
phase the objective of The Sneak to obtain a technological solution for race registration was unveiled. The analysis phase uncovered many pitfalls of the historical Excel spreadsheet method of race participant tracking and the inaccurate ineffective data that was the result of this spreadsheet system. The design phase included design of the database and future system enhancements. The implementation phase only included implementation of the database for the purpose of this project as future projects will include additional system implementations. The final phase, maintenance, will not be achieved in this project but will be achieved in other future related Regis projects. As technological systems have a life cycle, so do databases themselves. The next chapter will discuss the life cycle of a database and how that life cycle relates to The Sneak.
5. Database Life Cycle

5.1 Introduction to the Database Life Cycle

The Database Life Cycle is the life cycle that a database experiences. This life cycle contains stages that are similar to that of a total technological system; however they are more specifically refined to database needs. The Sneak database itself has experienced and will continue to experience a database life cycle. This life cycle will continue for The Sneak in the form of future projects. The life cycle for a database contains six phases. These phases include: initial study, database design, implementation and loading, testing and evaluation, operation and maintenance and evolution. (Rob, Coronel, 2002, p. 325-350)

5.2 Initial Phase

The goal of the initial database study is to: analyze the company situation, define problems and constraints, define objectives and define boundaries and scope. The initial database study includes not only examination of the current system but why the current system is not meeting the needs of the organization. Key to the initial database design is communication with the future end-users of the database. This is crucial when creating a database to fit the needs of the users in order for the database to be effective for the organization. (Rob, Coronel, 2002, p. 325-330) During the initial study of The Sneak it was determined that the legacy system was no more than a series of Excel spreadsheets. This did not meet the organizational needs as The Sneak could not compare
participant information from one year to another. Additionally, it was difficult to gather information about the race participants. The data within the Excel sheets was inconsistently created and generally lacked accuracy. Participant registration from one year to the next could be listed as J. Doe, Jane D., Jayne Doe etc. This is mainly due to the inconsistent means of populating the Excel sheet and lack of historical database information to retrieve prior participant information. The management team of The Sneak was desperate to have a race database in which participant information could be analyzed and utilized to increase racer participation.

4.3 Design Phase

The next phase of the database life cycle is database design. There are four main segments to the database design phase including: creation of the conceptual database design, software selection, creation of the logical design and creation of the physical design.

Conceptual design incorporates an analysis of database requirements, entity relationship modeling, verification of the data model and distributed database design. (Rob, Coronel, 2002, p. 332-342)

During the data analysis and requirements gathering it was important for the student to answer the following questions:

- What are the information needs?
- Who are the information users?
- What are the information sources?
• And what data elements are required to provide the required information?. (Rob, Coronel, 2002, p. 333)

What are the information needs?

The Sneak needs various pieces of information to be shared with the sponsors of The Sneak. The information should show that its participants patronize the sponsors. The sponsor will benefit from having their respective organization represented at The Sneak.

Who are the information users?

The information gathered by The Sneak will be used by event sponsors, event vendors, race participants and The Sneak organization.

What are the information sources?

The primary information source for The Sneak is the racer registration information. Secondary race participant information may be provided by the vendors and sponsors of The Sneak. The information constitution of The Sneak is primarily the registration.

What data elements are required to provide the required information?

All primary race participant information is maintained in the registration process. Previously, the registration process would only capture very basic information regarding the participant. It is hoped that implementation of a database will allow greater information gathering about the race participants. Information could include household income, other activities, health information and additional mailing list options.
A primary methodology for conceptual design review is end-user data viewing. The student was able to create this end-user view as a 2004 Cherry Creek Sneak race participant. The student originally attempted to enter participant registration information via the local store registration process. However, it was determined that The Sneak web site listed a store location that was not participating in the registration process. This was an obvious data accuracy issue that was quickly exposed and could have been prevented with utilization of a database. The database could have a list of store locations and a report would be generated to verify the locations with participating stores.

As a secondary resort, the student registered for the race via the website on-line. The on-line registration process linked the student to a secondary web-hosting registration company. Although The Sneak intended this site migration to be seamless, it was quite obvious to the user. Some potential participants may be concerned with the use of information provided and opt to not register for the race on-line. The on-line registration system required basic race participant information and did not provide the student possible useful information such as number of participants in the same event, prior year race result information, or event details such as parking and directions to the event. One key piece of information to provide could be the current number of people registered in each wave. This would be helpful for elite racers to move to a less popular wave to increase start time speed. The student determined that
the current information provided was not useful to the race participant as an end-user.

Throughout the database design phase the student attempted to obtain a sample or copy of the current database information. The Sneak however, was not able to provide the information requested. The Excel spreadsheet was being reviewed by a mailing company whose purpose was to scrub the participant information, remove false or invalid address information and consolidate duplicate race participant information. The mailing house company commented as to how unlike most true databases the Excel sheet was designed. The mailing house was able to remove duplicates and validate address information via a specialized software program. The database sample information was requested on several occasions throughout the project and was not received. As such, it was obvious to the student that a sponsor or vendor of The Sneak would have an impossible time obtaining useful information regarding Sneak participants that would add value to the commitment made to The Sneak.

Conceptual design continued with the creation of entity relationship modeling and normalization of data regarding The Sneak. Entity relationship modeling requires the following steps: identification and analysis of business rules, identification of main entities, definition of relationship among entities, definition of attributes of the entities and their respective primary keys, normalization of entity information, completion of an initial entity relationship diagram, verification of the entity model by the
end users and modification of the entity model based on end-user feedback received. (Rob, Coronel, 2002, p. 335-342)

The simple form of business rules for The Sneak would be:

- A racer registers for an event.
- A racer runs in an event.
- Racers are granted awards.
- Sponsors sponsor an event.
- Sponsors sponsor an award.
- Vendors vend at an event.

The determination of these business rules assisted in the creation of the entity relationship model. Through these business rules it was determined that the key entities are the racer, the event, an award, a sponsor and vendors. These entities have various relationships among each other and some entities may not have a relationship. An example of the relationship would be the racer running in the event. The racer is a participant in the event.

In addition to racers The Sneak requested the ability to identify other non-runner participants such as VIPs, media personnel and volunteers at the event. A determination was required as to how to best handle these additional types of participants. Should each of these types have their own table or is there another method to handle the additional types? In this case, the student determined the best method to handle the additional participant types was to generate a participant type table
and identify each individual participant type. This method was chosen due to the fact that an individual could possibly be a racer and a VIP. It was determined this would be the most effective way to handle an individual wearing multiple hats without creating a large amount of redundant data in the database. If a table were created to maintain VIP information and a separate race participant table were created, the same person potentially would have two records stored in separate tables. This method would create redundant data.

The primary key is a unique identifier for an entity. (Rob, Coronel, 2002, p. 120-121) Primary keys are critical to a database as they are an integral method for joining tables together and they allow for database normalization. Each entity within the database should have a primary key associated with the entity records. In addition to the various entities containing a primary key, the student determined that it would allow for the best tuning ability if each table had its own primary key. The primary keys allow for indexing of database tables in the future if required. The student identified the primary key for the entities previously mentioned to be event_id, sponsor_id, individual_contributor_id and award_id. Throughout the database design, the student identified the primary key as tablename_id. This allows for easy identification of the primary key. Easy identification of the primary was important to the project as the next step projects will rely heavily on the primary key information.
Normalization of entity data included breaking the individual contributor table, sponsor table and event table into two tables. This allowed the student additional flexibility as the person will likely not change in the future but address, telephone may change. As such, detail tables were created for each main entity. This breakdown of table information will allow for future adjustments and growth.

Entity models for The Sneak can be located at Appendix A. The initial entity relationship diagram appeared quite different than the final version of the entity model. Various revisions were made as discussions with The Sneak management led to entity modeling modifications.

There are many database management system (DBMS) software solutions available. As such, a decision is required to determine which DBMS software will provide the best solution for the organization. This decision making process included consideration of the following factors: cost, DBMS features and tools, underlying database model, portability of the database to various platforms and hardware requirements to operate the DBMS software. (Rob, Coronel, 2002, p. 342)

The main consideration for The Sneak was cost. As no budget was made for any additional hardware or software was made to complete the database project cost was a limiting factor.

In addition to cost, the next most prudent factor for The Sneak was portability. Portability was an important factor as The Sneak was considering changing web hosting companies, searching specifically for
one with an improved database engine. Historically, the “database” was fully portable as an emailed Excel spreadsheet, parts of which were transferred to non-normalized tables in Microsoft Access on The Sneak's original web host. The database needed to be as flexible as possible. Database flexibility is important as it allows for growth. DBMS were not considered a significant factor as The Sneak will not impress significant system demands on the database. It is mainly utilized as a reporting tool. Database demands would only be significant from March through May. Increased demand is attributed to registration occurring and racers viewing results. As such, a system of shared database, where database CPU can be transferred to The Sneak database, was needed.

The DMBS selected by the student was MySQL. The main consideration for choosing MySQL was the fact that it is available as freeware.

Additionally, MySQL provides a multi-threaded SQL server for use on multiple CPUs within a client server system. MySQL is a multi-user platform which is an important factor for The Sneak as they may have multiple users of the system at a given time. The database can be secured in MySQL through the use of the secure password system. MySQL is open source, meaning that it is technology that is open for use and any one could modify it. It is written in a combination of C and C++ both of which are very common programming languages. MySQL will have the ability to handle the growth of The Sneak database as some
MySQL systems are known to be as large as sixty thousand tables with over five billion rows of data. MySQL is supported by many different backend systems, administrative tools, programming interfaces and client libraries. It can be accessed via TCP/IP, Unix or NT platforms. Application Program Interfaces (APIs) can be written in C, C++, Java, Eiffel, PHP, Perl, Python or Tcl and interface with MySQL. MySQL is ANSI SQL99 compliant and can be used with a general public license (Widenius, Axmark 2002, p. 4-11) The flexibility of MySQL is key to The Sneak as the web hosting company and hardware platform will likely change in the future.

The next challenge was to translate the conceptual design determined in the entity relationship diagram to a design model for the DBMS. This is the logical design phase. This phase of design included design of tables, indexes, views, transactions, access authority and other database components. (Rob, Coronel, 2002, p. 342-344) The Sneak's table design can be reviewed in appendix B. The final design for views and transactions will be generated in the next phase of database generation for The Sneak and will be customized for the on-line tool that has not yet been designed. The index generation will be completed in future Sneak projects. It has not yet been determined where indexing would provide the most benefit to the database.

The Sneak operations database is considered to be fairly simple and few objects and procedures are required by the database. As such, a
centralized database design is the most appropriate design for The Sneak. Centralized databases are most successful when database demands are simple and a single operator can manage the entire database. (Rob, Coronel, 2002, p. 351) The simplicity of the database needs make the centralized database a perfect fit for The Sneak.

5.4 Implementation Phase

The next step for The Sneak database was implementation. During implementation data conversion will occur, system procedures are defined, performance is measured, security is measured and backup and recovery systems are refined. (Rob, Coronel, 2002, p. 345-349)

The implementation of data conversion can occur through manual entry or through an API (Automatic Program Interface). An API is a method of copying the data from one database into another database. As the data within the tables for The Sneak is integrated via foreign keys and primary keys it is crucial that the data is loaded in the correct manner. All primary key data must be loaded before foreign key data can be loaded into the database. (Rob, Coronel, 2002, p. 345-349) The order is essential as foreign key tables are dependent on the existence of the corresponding primary key data.

The Sneak will also require procedures for how the database will be managed, accessed and maintained. These procedures are known as system procedures. Strict standard procedures protect the integrity of the
data as it will avoid poorly executed database changes leading to future data integrity problems. (Rob, Coronel, 2002, p. 345-349)

Performance of The Sneak database is critical as customers of The Sneak will be accessing the database. One performance requirement is the ability of the users of the database to retrieve information from the database in a reasonable amount of time considering the cost of retrieval. A database can run extremely fast if the database management team is willing to pay the price for the fast retrieval speed. Data retrieval speed is typically measured by the number of transactions processed by the database per second. This will be one of the many future considerations of The Sneak. (Rob, Coronel, 2002, p. 347-348) The Sneak does not have historical information as to transaction processing or retrieval request speed. The needs of the database will be analyzed as it is utilized.

The Sneak database is currently password controlled for administrative use only. In the future the database access will need to be expanded to not only The Sneak management team but additionally to users of the website including racers, sponsors and vendors. At that time security will become a critical factor to control the data integrity. The data must be assured safe storage. More widely defined access controls must be implemented to allow users access to the specific data that they will require without compromising data that they should not be allowed access. (Rob, Coronel, 2002, p. 348) As an example, race participants
should be allowed to query results, enter data via registration only but not allowed to retrieve or update sponsor information.

In the event that securities are breached on the database and the database is corrupted, it is important that the pre-corruption database can be retrieved. This is an example of a backup and recovery system. A backup system can protect the database from hardware failure, database failure or corruption. A backup plan must be implemented in the next phase of The Sneak’s database. This backup plan should contain a frequency of backups, possibly on a sliding scale to account for the greater need for back ups as the race event date is closer. (Rob, Coronel, 2002, p. 348)

5.5 Evolution Phase

The final phase for database life cycle is evolution. Evolution will not be experienced in conjunction with this project as the database is not mature enough to evolve. Evolution will be encountered as the usage and needs of the database change in the future. An example of a possible evolution of The Sneak database is the creation of additional tables due to new processes such as on-line payment capability. This would require the creation of tables to maintain security for credit card payment and credit card payment transactional tables.

5.6 Summary Database Life Cycle

The database life cycle for The Sneak today and into the future has been shown to include: initial study, database design, implementation and
loading, testing and evaluation, operation and maintenance and evolution. (Rob, Coronel, 2002, p. 322-323) The initial phase highlighted the fact that a new database was required in order to convert The Sneak data into useful information. In the design phase, a conceptual design was generated, software selections were made, physical design was completed and logical design was generated. The implementation phase for The Sneak would only reach partial completion as the database would be implemented but the historical data would not be converted and loaded into the database. The operation and maintenance phase of design will occur throughout the next year as the first race year is experienced utilizing the new database system. The final phase of evolution will be experienced by The Sneak in future years as the database requirements and needs are changed as business needs change.
6. Capability Maturity Model

6.1 Introduction to the Capability Maturity Model

As the technological systems of The Sneak mature, it is important to recognize the systems technological maturity level. This can be accomplished through review of the capability maturity model would find the pre-project state of The Sneak as initial. The capability maturity model (CMM) has five levels. (Whitten, Bentley, Dittman 2001, p. 76)

6.2 Initial Level

The first level of the CMM is labeled initial. The initial level can be described as chaos or anarchy. The Sneak is described in this manner due to the lack of controls surrounding the Excel spreadsheet known as “the database”. There were no version controls in place and it was updated on an ad hoc basis. (Whitten, Bentley, Dittman 2001, p. 76-79) Even the data within the Excel spreadsheet appears to be obtained in disorganized and random manner. The Sneak entered the project at this first level of CMM known as initial. A goal of the project is to assist The Sneak in moving down the CMM path towards the managed level the, fourth level of CMM (Whitten, Bentley, Dittman 2001, p. 76-79)

6.3 Repeatable Level

The next step towards the goal was the second level of the CMM, which is labeled repeatable. This stage involves refining and defining project management processes and practices. In order to meet the requirement of repeatable, the management process must be defined to
the point of reusability. The Sneak achieved level two through the
database creation process. This is achieved as the design allows for
repeatable processing from one year to the next. The goal of stage two is
to create a foundation for project management. This basic foundation was
achieved once the database was designed and implemented. As the
database matures, additional features will be incorporated into the
database. The additional features will then need to advance to the level of
repeatability. This process can be assisted along through the use of
development standards and process documentation. The creation of
standards for development allows the system to reach the CMM level of
repeatability at a faster rate. This is due to the fact that the developer is
not beginning each development process at the earliest stage of the initial
level. (Whitten, Bentley, Dittman 2001, p. 76-77)

6.4 Defined Level

Level three of the CMM is defined. This translates to a stable,
repeatable development process either through the use of system
development tools or processes. The key to the third level is that the
process has stabilized. The Sneak will achieve the third stage of CMM in
later projects. These projects are still in design phase as such
development tools have not yet been defined. In order to achieve the
defined level, the technological system must be consistently stable over
time. The Sneak is not expected to reach this level for at least a year and
likely longer. The extended time period is due to the constant
development of new technological projects in association with The Sneak.
As the technological environment will likely be changing in the near future,
it will be quite difficult to achieve a stabilized technological environment.
(Whitten, Bentley, Dittman 2001, p. 76-77)

6.5 Managed level

Level four, the ultimate goal of The Sneak projects, is managed.
This stage takes project management to the next level by integration of
measurable goals in quality, productivity or cost. This goal will be
achieved by The Sneak once the database is complete, data conversion is
completed, and the web-front-end interface is fully implemented. At this
point the goals can be truly measured. One goal that The Sneak holds in
highest priority is the ability to provide race time results within one day of
race completion. This goal would one of the first measured. Additional
goals of The Sneak that would be measured is the ability to provide
reports regarding race participant demographics and attendance. A more
technological goal would be the retrieval time lag when queries are made
against the database for race results. The time lag must be of a
reasonable level or the person querying the data will become frustrated
and not return to the website and possibly the race. (Whitten, Bentley,
Dittman 2001, p. 76-77)

6.6 Optimized Level

The final level is optimized. In the optimized level, the organization
has developed a standard change management process and the only
adjustments made are due to lessons learned and elimination of inefficiencies in the development process. Level five is not the goal of the project as the maturity required will need to develop over time. The time required to obtain the optimized level may be many years as the technological environment is changing so dramatically. (Whitten, Bentley, Dittman 2001, p. 76-77)

6.7 Summary of the Capability Maturity Model

The capability maturity model provides a basic framework to determine the maturity level of an organization’s technology stack. The CMM contains five levels: initial, repeatable, defined, managed and optimized. (Whitten, Bentley, Dittman 2001, p. 76-79) The Sneak began this project in the earliest stages of initial. The goal of the project is to bring The Sneak ahead to a defined level with the actual database. The technology stack as a whole for The Sneak is expected to achieve the repeatable level of the CMM. The Sneak is only expected to achieve this level to due the large number of technological projects currently being generated for The Sneak. Once The Sneak has completed more of the technological projects currently in development it is expected they will move to the managed level of the Capability Maturity Model.
7. FAST Methodology

7.1 Introduction to FAST

Systems development requires a methodology or a development process. These methodologies are commercially available; however, due to budgetary constraint issues, The Sneak utilized a generic methodology known as FAST: “Framework for the Application of Systems Techniques”. FAST comprises eight main principles. Each of these principles will be reviewed and applied to The Sneak. (Whitten, Bentley, Dittman 2001, p. 78)

7.2 Involvement of the Users and System Owners

The first principle of the FAST methodology calls for involvement of the owners and users of the system. This principle is important as typically people are resistant to change. This is especially the case when the people feel an ownership or developmental stake in the system being changed or removed without their acceptance and or approval. The goal of this principle is to create open communication with the owners and users to avoid misunderstandings or miscommunications regarding the purpose of the change. This was not a major issue for The Sneak as the management team initiated many of the changes and end-users are generally one-time temporary employees or volunteers. (Whitten, Bentley, Dittman 2001, p. 79)

7.3 Implement a Problem Solving Approach
The second principle of the FAST methodology implements a problem solving approach to technology. The problem solving approach assures that the goal of the system change is at the forefront of the development process. Problem solving involves understanding the problem for The Sneak, which is the ability to track important racer information including: address, race times and personal details about the racer. Problem solving additionally defines solution requirements, identifies possible solutions, design, implementation of solutions and observation of the impact of a given solution. This project contained a migration to the point of design and partial implementation of the database solution. (Whitten, Bentley, Dittman 2001, p. 79-80)

7.4 Establish Phases and Activities

The establishment of phases and activities in the third principle allowed the student to tailor design phases to the needs of The Sneak. The preliminary investigation phase brought the student to The Sneak website to learn about the race and events surrounding the event. This familiarity with the event assisted to understand possible data requirements. The problem analysis phase was uncovered through various conversations with The Sneak management team. This phase could also be characterized as discovery since various problems that The Sneak was experiencing were discovered during this phase. Requirements analysis went hand in hand with problem analysis. As each problem was uncovered the database project requirements were
reviewed. The design phase, construction and implementation were mainly handled by the student solely with input from The Sneak management and the program proctor on an as needed basis. (Whitten, Bentley, Dittman 2001, p. 80)

### 7.5 Establish Standards

The fourth principle of FAST is the establishment of standards. The standards identified by the student were the software standard for design and implementation and documentation of the design. No hard and fast standards were developed as the project composed of solely database design and no coding development was involved in the project. It is recommended by the student that all future code contain a header of the most recent update information including the purpose of the code. Additionally, all code sections should have a brief comment detailing the purpose of the code. The software utilized for the design of the database was Visio as it provided the most flexible design capabilities. The software utilized for implementation of the database was MySQL. MySQL is further detailed in chapter five. (Whitten, Bentley, Dittman 2001, p. 80-82)

### 7.6 Project Justification

The fifth principle is the justification of the systems project as a capital investment instead of an operational expense. This is a key principle for organizations under significant project budget constraints. In many corporations today information technology projects will not be
approved if they are not able to show a return on investment in the project. The Sneak project offers possible returns in lower mailing house costs as a result of more efficient data. The mailing house will no longer be paid to clean the data each year at the completion of the event. Additionally, monies for postage will not be wasted on old or invalid addresses. The other savings or earnings to The Sneak are more discrete in nature. The Sneak will have the ability to attract more sponsors. The Sneak will have the ability to provide information to the sponsors about who participated in The Sneak. This allows the sponsors the ability to properly market themselves at each event. These are investment returns as a result of the project; however they are more intangible in nature. (Whitten, Bentley, Dittman 2001, p. 82-83)

7.7 Allow for Modification of Project Scope

Principle six was often used in The Sneak project: Allowing the ability to modify or cancel project scope. (Whitten, Bentley, Dittman 2001, p. 83) The project scope for The Sneak was initially quite large. The originally scoped project included creation of the database, implementation of the database, data conversion, reporting design and online web form creation. This initial project was changed to only include database design, implementation and data conversion and the final project removed the data conversion portion as it was reassigned to another Regis student. Oddly, this project had the opposite issue that most projects encounter. Typically information systems experience scope
creep or taking on more than the original project required. The total needs of The Sneak and overall scope is increasing, however individual student demands have declined as more students have been brought into the project.

7.8 Divide and Conquer

Principle seven, divide and conquer, was important to understand the components required for The Sneak database. The ideology behind the principle is that division into subsystems allows for greater understanding of the project as a whole. (Whitten, Bentley, Dittman 2001, p. 83-84) The Sneak was subdivided by entity perspective. The initial database requirements analyzed were those for the racers and race registration. The next step was to review the needs for the race event itself. This included the type of information that would possibly be gathered in conjunction with a race event. The last main entity reviewed was the sponsor and the information that could be gathered about the sponsor that would prove beneficial for reporting purposes. Additional database information was also reviewed including: lost and found, medical aid, vendor and neighbor information. These segments of the database were not considered a primary entity. However, The Sneak requested the ability to capture such information. Since this type of information was not directly related to registration, it was handled as its own separate entity.

7.9 Design for Growth and Change
The most difficult principle to address was the ability to design a system that had the ability to grow and change. The student considered this principle the most important of all, as The Sneak management team only had a vague understanding of database capabilities and the direction of the database and future usage. The student understood that once The Sneak realized the true power of a normalized database, they would want to use the database to its fullest capacity. Poor database design without the freedom of growth would restrict the database from being used to its full potential. Statements made by The Sneak management in casual passing would require careful analysis as to whether they should be included as part of the database. An example of such statements is that The Sneak thought it would be interesting to track the clicks made on The Sneak website. A determination was required whether or not to include a table that would handle the click data. It was determined that such a table was not required at this time and could be added at a later stage if required, due to the fact that the web hosting company was already tracking this information. Other casual statements were reviewed and incorporated into the database such as the desire to track people that were treated by the medical response team. The main goal of the student was to build as much flexibility into the database as possible. A way this was made possible was through the addition of thirty attribute fields to each table. These attribute fields could be used in the future for whatever means necessary. (Whitten, Bentley, Dittman 2001, p. 84)
7.10 Summary of the FAST Methodology

The FAST methodology worked well for The Sneak. The small number of database users at The Sneak assisted this methodology to work very smoothly. In review, the first principle of the methodology was to obtain involvement of the owners and users of the system. This principle was likely the easiest to achieve as the system stakeholders requested the project occur and the managers of The Sneak were very involved. The second principle of utilizing a problem solving approach was of high importance for The Sneak. Many of the systems requests were presented as problems of the legacy system. This allowed for the most comprehensive view of the issues at hand. Principle three, the establishment of phases and activities contained one main activity and that activity was requirements analysis. Principle four, establishment of standards included a future recommendation regarding the standards. Justification of the project, principle five, was relatively easy as it was very apparent to The Sneak management that they were not obtaining the information they needed and a database was the solution to obtain that information. The sixth principle ability to modify or cancel project scope occurred throughout The Sneak project. This principle was likely over used, as the initial student project moved from creating an entire technological solution for The Sneak to implementation of database design. Principle seven, divide and conquer, was quite useful to the student as the database was divided into primary entity components
during design to fully understand each entity and then how the entities were related. The principles associated with FAST proved very effective during the database design for The Sneak. (Whitten, Bentley, Dittman 2001, p. 79-84)
8. Project Management

8.1 Total Project Management with Scrum

The Sneak project as a whole for all students was managed utilizing a project management technique known as scrum. Scrum is a relatively new (late 1990s) software development management philosophy. Scrum was developed by Ken Schwaber and Mike Beedle due to a need for a more effective technical project management methodology. The general premise behind scrum is that there should be one project leader, many smaller meetings and smaller goals. (Schwaber, Beedle 2001, p. 7-12)

Scrum based projects have a central project manager or scrum master. It is the duty of the scrum master to control the backlog, scrum meetings and sprints. The Sneak scrum leader was the project’s proctor or Regis faculty representative Ms. Litz. Throughout the project, the scrum leader recorded various backlog issues surfaced by The Sneak management team. Backlog items are requests by The Sneak for any technological solution of any kind. The scrum master coordinated what items would be included with each sprint of the project and who on the project would be responsible for the sprint. The project sprints are the mini-goals set by the scrum master. Finally, the scrum master organized all scrum meeting sessions and kept those sessions on track to effectively address project issues at hand. (Schwaber, Beedle 2001, p. 7-12)
The task list of a scrum project is the product backlog. The product backlog is a list of features or functions that the system users would like to have implemented. (Schwaber, Beedle 2001, p. 7-12) The product backlog for The Sneak project included everything from the database itself to a bar coding system for racer timing results. The backlog is designed to be a list of any and all customer requests regardless of feasibility. All requests will make the initial product backlog list, though not all items on the list are guaranteed to be implemented. The backlog is constantly prioritized by the product owner or person with the vested interest in the project. In the case of this project, this was The Sneak management team. On a periodic basis, the product backlog was reviewed with the product owner to assure that the project prioritization was still appropriately being applied, as needs may change over time. The overall needs of The Sneak continually showed a focused need for a database and a method to quickly determine race results. All other backlog items were tabled for later sprints of the project.

A sprint is a release of system functionality results. The first sprint released for The Sneak was a database and online entry system for volunteers. The next sprint was the design of the database. This sprint contained many smaller sprints, such as the definition of the entity-relationship diagram, table design and review of design. The true practice for a scrum project calls for many small releases of the system being designed. (Schwaber, Beedle 2001, p. 7-12) Due to the nature of The
Sneak project, it was not feasible to do many small releases as there are only a handful of team members responsible for large segments of the project. The ideal situation to utilize scrum is in a large system development environment, however, it can be applied to smaller developer groups.

The key to the scrum philosophy is the daily scrum. This is a regular (intended to be daily) meeting of all project leads and interested parties to discuss the current status of the project, next project steps and project accomplishments made. (Schwaber, Beedle 2001, p. 7-12) The Sneak project maintained scrum meetings, however, due to the nature of the project these meetings were monthly scums. As all participants were working towards the project goal at different rates and on a part time basis, there was not enough change in the project to require daily or even weekly scrum meetings. The monthly meetings were considered ample. The meetings allowed each project lead to discuss their progress within the last month and their respective proposed direction. This also allowed the project leads to review their respective understanding of The Sneak’s needs to assure that the system being developed was in-line with the expectations of The Sneak. It was the opinion of the student that two meetings per month may be more effective for the sprints. One of the meetings could include The Sneak management team. The second meeting would be utilized by the Regis faculty and student project owner to review the total thesis project progress and status.
The scrum project management philosophy allowed the appropriate level of flexibility required for The Sneak project. The backlog maintained the changing needs of The Sneak. The scrum master was able to effectively coordinate the respective student projects. The place where The Sneak project fell down was during the “daily scrums”. The meetings to be coordinated via teleconference on a month basis were often missed and as such the total project fell behind. Had the daily scrum schedule been more consistent the project possibly could have moved along more quickly? Overall, the scrum methodology was a good approach to manage this project and will likely be utilized in the future.

8.2 Student Project Management

The student project was managed on a linear timetable basis. The project was managed in a structural manner with set goals and achievement dates. These dates and goals changed as the project scope changed based on the needs of The Sneak. The timetable for the student project including completed Regis coursework can be reviewed in Appendix C.

The initial meeting with The Sneak indicated that the project would be completed in January including implementation in preparation for the 2004 race. Quickly it became evident that a solid database could not be generated in such a short timeframe. As such, it was decided with cooperation from The Sneak that the database solution would be completed in May of 2004, after completion of the current year race.
In December, the project was originally to implement a technological plan for The Sneak that included: a database, on-line registration system, hosting company selection, bar-coding race results and data conversion. By January it was determined that this was too large of a task for a single student. Two new students were added to The Sneak project team. One student was tasked with generating an overall current and future technology plan and the second student was given the web registration portion of the project. The bar coding of race results was tabled until the other portions of the project were completed. Now the scope for the student was to create a database and convert the Excel spreadsheet information into the database by May. This was a more attainable goal.

The Sneak met with the team from Regis on a monthly basis. It was the goal of the student to have reviewed the entity relationship diagrams and update them with feedback provided at each of the meetings. The student was to complete the software selection process by April in order to be ready for data conversion for the May goal. In order to get a hands-on perspective of the race, the student participated in the 2004 Cherry Creek Sneak. The participation allowed the student to review the effectiveness of data collection and determine if there would be any additional information that may be useful in the future.

In May, The Sneak and the Regis team met again to discuss the database and the project progression. In the May meeting a new student
joined the project. This student was tasked with the conversion of historical race data into the new database. As such, the student’s role changed again. Now, the student would be responsible for database design and implementation. Conversion of data no longer applied to the student project. The student would hand off the database to the latest team member for data conversion purposes.

The database was near completion by the end of May. Some minor changes were made as a result of the June meeting, including the addition of update date and by information on each table. This information was considered quite important by The Sneak management team. The final database was handed to the conversion student in mid July. This was very timely, as the previous race conversion information was finally ready for the conversion process. A gnat chart timetable of the student’s project and master’s completion can be viewed in appendix C.

The student project was managed on a very linear basis. The project management was a traditional approach with date deadlines. The student did adapt often to the changing requirements of the project and was flexible to the needs of The Sneak at all times. This management of the student project was quite different from the scrum management of the total Sneak project. The scrum management is designed for groups not individual projects. Scrum was very effective for the project needs especially the sprints and sprint meetings. These could only be improved by holding such meetings on a bi-monthly basis. This would be especially
useful as it was rare that all team members were able to attend the sprints.
9. Project Summary

This paper has brought together a discussion of the creation of a race database for The Cherry Creek Sneak. The discussion included an introduction to the paper and its purpose, history of the project, the purpose of database management systems, the systems development life cycle, database life cycles, capability maturity model, FAST and project management.

The goal of the paper was to demonstrate knowledge gained through studies at Regis University and apply that knowledge to the master’s thesis project. The project goal was to design a race database for The Cherry Creek Sneak. Throughout the paper the knowledge of databases and database management was applied specifically to data from The Sneak.

The history of The Sneak and the use of Excel spreadsheets was provided to demonstrate how greatly The Sneak would benefit from a database implementation. The Sneak is making major changes in their technology stack through database implementation. These changes will hopefully assist in the development and growth of The Sneak.

The implementation of a database management system for The Sneak will assist in the conversion of raw race data to information. The conversion of data to information through a normalization process will allow for ad hoc querying and a new technology platform.
The systems development life cycle assisted in the transformation process through planning, analysis, design, implementation and maintenance of the technology platform.

Through the database life cycle, The Sneak moved from an initial state of anarchy to implementation of a true database. The future of The Sneak holds final testing of the database, evaluation of the functionality, operations and an evolution as The Sneak needs grow.

A capability maturity model showed that the simple implementation of a database moved The Sneak from a state of anarchy to a defined managed technology stack that in the future will be optimized to be as effective and efficient as possible.

The FAST methodology was utilized to define the pitfalls of the previous system and focus the database implementation to correct previous system problems.

A dual project management approach involving scrum and a traditional project management methodology worked in harmony to create an achievable project timeline involving many project team members.

The key to The Sneak project was the demonstration that databases can significantly improve operations of an organization when properly designed and implemented. This was achieved for The Sneak through teamwork and knowledge obtained through studies at Regis University.
Bibliography


Tiernan, Paula. Personal interview. May 2004


Appendix A – Entity Relationship Diagrams
Appendix B – Database Tables
CREATE TABLE Lost(
    Lost_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    Lost_Item VARCHAR(50),
    Lost_Description VARCHAR(250),
    Lost_FirstName VARCHAR(50),
    Lost_LastName VARCHAR(50),
    Lost_ContactNum NUMERIC(9,0),
    Updated_by NUMERIC(10,0),
    Updated_Date Date,
    PRIMARY KEY (Lost_ID));

CREATE TABLE Found(
    Found_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    Found_Item VARCHAR(50),
    Found_Description VARCHAR(250),
    Found_FirstName VARCHAR(50),
    Found_LastName VARCHAR(50),
    Found_ContactNum NUMERIC(9,0),
    Updated_by NUMERIC(10,0),
    Updated_Date Date;
    PRIMARY KEY (Found_ID));

CREATE TABLE Individual_Contributor(
    Individual_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    Individual_FName VARCHAR(50) NOT NULL,
    Individual_LName VARCHAR(50) NOT NULL,
    Individual_DOB DATE NOT NULL,
    IND_Gender VARCHAR(1) NOT NULL,
    IND_Start_Date DATE,
    IND_END_Date DATE,
    IND_Research VARCHAR(1),
    IND_Old_Database VARCHAR(1),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
CREATE TABLE Individual_Detail(

  Detail_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  Detail_IND_ID SMALLINT UNSIGNED NOT NULL REFERENCES Individual_Contributor,
  Address_Line1 VARCHAR(250),
  Address_Line2 VARCHAR(250),
  City VARCHAR(50),
  State VARCHAR(2),
  ZIP NUMERIC(5,0),
  Phone1 VARCHAR(9),
  Phone2 VARCHAR(9),
  Phone3 VARCHAR(9),
  Emergency_Name VARCHAR(100),
  Emergency_Number VARCHAR(9),
  Shirt_Size VARCHAR(3),
  SSN VARCHAR(9),
  Contact_Preference VARCHAR(25),
  Email1 VARCHAR(100),
  Email2 VARCHAR(100),
  Start_date Date,
  End_date Date,
  Company VARCHAR(100),
  School VARCHAR(100),

  PRIMARY KEY (Individual_ID));
Publicy_sent DATE,
Mailing_list VARCHAR(1),
Email_List VARCHAR(1),
Waiver_Signed DATE,
History_Code VARCHAR(50),
Ski_Board VARCHAR(1),
Climb VARCHAR(1),
Raquet_Sport VARCHAR(1),
Football VARCHAR(1),
Basketball VARCHAR(1),
Baseball VARCHAR(1),
Hockey VARCHAR(1),
Soccer VARCHAR(1),
Swim VARCHAR(1),
Bike VARCHAR(1),
Other_Sport VARCHAR(250),
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPdated_Date DATE,
PRIMARY KEY (Detail_ID));
--Create table EVENT for the Race Database.
CREATE TABLE EVENT(
    EVENT_ID          SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    EVENT_NAME        VARCHAR(250) NOT NULL,
    EVENT_DESCRIPTION VARCHAR(250),
    EVENT_MANAGER     VARCHAR(100),
    EVENT_DATE        DATE,
    Attribute1        VARCHAR(250),
    Attribute2        VARCHAR(250),
    Attribute3        VARCHAR(250),
    Attribute4        VARCHAR(250),
    Attribute5        VARCHAR(250),
    Attribute6        VARCHAR(250),
    Attribute7        VARCHAR(250),
    Attribute8        VARCHAR(250),
    Attribute9        VARCHAR(250),
    Attribute10       VARCHAR(250),
    Attribute11       VARCHAR(250),
    Attribute12       VARCHAR(250),
    Attribute13       VARCHAR(250),
    Attribute14       VARCHAR(250),
    Attribute15       VARCHAR(250),
    Attribute16       VARCHAR(250),
    Attribute17       VARCHAR(250),
    Attribute18       VARCHAR(250),
    Attribute19       VARCHAR(250),
    Attribute20       VARCHAR(250),
    Attribute21       VARCHAR(250),
    Attribute22       VARCHAR(250),
    Attribute23       VARCHAR(250),
    Attribute24       VARCHAR(250),
    Attribute25       VARCHAR(250),
    Attribute26       VARCHAR(250),
    Attribute27       VARCHAR(250),
    Attribute28       VARCHAR(250),
    Attribute29       VARCHAR(250),
    Attribute30       VARCHAR(250),
    Updated_by        NUMERIC(10,0),
    Updated_Date      Date,
    PRIMARY KEY (EVENT_ID));

--Create table EVENT_Detail for the Race Database as a foreign key table to EVENT.
CREATE TABLE EVENT_DETAIL(
    EVENT_DETAIL_ID          SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    EVENT_ID                 SMALLINT UNSIGNED NOT NULL REFERENCES EVENT,
    EVENT_DETAIL_NAME        VARCHAR(100),
    DISTANCE                 VARCHAR(50),
CREATE TABLE SPONSOR

TYPE VARCHAR(50),
CATEGORY VARCHAR(50),
RACE_MAP_LINK VARCHAR(250),
DETAIL_DATE DATE,
TEMP NUMERIC(4,1),
HUMIDITY NUMERIC(4,2),
WIND NUMERIC(4,0),
SUN VARCHAR(50),
EVENT_CITY VARCHAR(100),
EVENT_STATE VARCHAR(2),
WAVE VARCHAR(25),
START_TIME VARCHAR(10),
END_TIME VARCHAR(10),
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPDATED_DATE Date,
PRIMARY KEY
(EVENT_DETAIL_ID))

--Create table SPONSOR table for the Race Database.
CREATE TABLE SPONSOR(
CREATE TABLE SPONSOR_DETAIL(SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
SPONSOR_DETAIL_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
SPONSOR_ID REFERENCES SPONSOR,
CONTACT_FIRSTNAME VARCHAR(100),
CONTACT_LASTNAME VARCHAR(100),
CONTACT_TITLE VARCHAR(100),
CONTACT_NUMBER1 NUMERIC(9,0),
CONTACT_NUMBER2 NUMERIC(9,0),
CONTACT_NUMBER3 NUMERIC(9,0),
SPONSOR_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
SPONSOR_NAME VARCHAR(100),
ACTIVE VARCHAR(1),
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
Update_Date Date,
PRIMARY KEY (SPONSOR_ID));

--Create table SPONSOR_DETAIL table for the Race Database as a reference table to SPONSOR.
ADDRESS_LINE1 VARCHAR(100),
ADDRESS_LINE2 VARCHAR(100),
CITY VARCHAR(50),
STATE VARCHAR(2),
ZIP NUMERIC(5,0),
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPDATED_DATE Date,
PRIMARY KEY (SPONSOR_DETIAL_ID));

--Create table NEIGHBOR table for the Race Database.
CREATE TABLE NEIGHBOR(
    NEIGHBOR_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    BUSINESS_NAME VARCHAR(100),
    FIRSTNAME VARCHAR(50),
    LASTNAME VARCHAR(50),
    ADDRESS_LINE1 VARCHAR(100),
    ADDRESS_LINE2 VARCHAR(100),
    CITY VARCHAR(50),
STATE                  VARCHAR(2),
ZIP                    NUMERIC(5,0),
PHONE1                 NUMERIC(9,0),
PHONE2                 NUMERIC(9,0),
PHONE3                 NUMERIC(9,0),
CONTACT_PREFERENCE    VARCHAR(50),
EMAIL1                 VARCHAR(100),
EMAIL2                 VARCHAR(100),
FIRSTNAME2             VARCHAR(50),
LASTNAME2              VARCHAR(50),
GENDER                 VARCHAR(1),
GENDER2                VARCHAR(1),
START_DATE             DATE,
END_DATE               DATE,
Attribute1             VARCHAR(250),
Attribute2             VARCHAR(250),
Attribute3             VARCHAR(250),
Attribute4             VARCHAR(250),
Attribute5             VARCHAR(250),
Attribute6             VARCHAR(250),
Attribute7             VARCHAR(250),
Attribute8             VARCHAR(250),
Attribute9             VARCHAR(250),
Attribute10            VARCHAR(250),
Attribute11            VARCHAR(250),
Attribute12            VARCHAR(250),
Attribute13            VARCHAR(250),
Attribute14            VARCHAR(250),
Attribute15            VARCHAR(250),
Attribute16            VARCHAR(250),
Attribute17            VARCHAR(250),
Attribute18            VARCHAR(250),
Attribute19            VARCHAR(250),
Attribute20            VARCHAR(250),
Attribute21            VARCHAR(250),
Attribute22            VARCHAR(250),
Attribute23            VARCHAR(250),
Attribute24            VARCHAR(250),
Attribute25            VARCHAR(250),
Attribute26            VARCHAR(250),
Attribute27            VARCHAR(250),
Attribute28            VARCHAR(250),
Attribute29            VARCHAR(250),
Attribute30            VARCHAR(250),
Updated_by             NUMERIC(10,0),
UPdated_Date           Date,

PRIMARY KEY (NEIGHBOR_ID);
--Create table VENDOR table for the Race Database.
CREATE TABLE VENDOR(
    VENDOR_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    VENDOR_NAME VARCHAR(100),
    VENDOR_DESCRIPTION VARCHAR(250),
    ACTIVE VARCHAR(1),
    RESEARCH_REQUIRED VARCHAR(1),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    Updated_by NUMERIC(10,0),
    Updated_Date Date,
    PRIMARY KEY (VENDOR_ID));

--Create table VENDOR_DETAIL table for the Race Database as a reference table to SPONSOR.
CREATE TABLE VENDOR_DETAIL(
    VENDOR_DETAIL_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    VENDOR_ID SMALLINT UNSIGNED NOT NULL REFERENCES VENDOR,
    CONTACT_FIRSTNAME VARCHAR(100),
    CONTACT_LASTNAME VARCHAR(100),
CONTACT_TITLE VARCHAR(100),
CONTACT_NUMBER1 NUMERIC(9,0),
CONTACT_NUMBER2 NUMERIC(9,0),
CONTACT_NUMBER3 NUMERIC(9,0),
ADDRESS_LINE1 VARCHAR(100),
ADDRESS_LINE2 VARCHAR(100),
CITY VARCHAR(50),
STATE VARCHAR(2),
ZIP NUMERIC(5,0),
PRODUCT_DESCRIPTION VARCHAR(250),
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPdated_Date Date,
PRIMARY KEY (VENDOR_DETIAL_ID));

--Create table FIRST_AID table for the Race Database.
CREATE TABLE FIRST_AID(
FIRSTAID_ID SMALLINT UNSIGNED NOT NULL,
AUTO_INCREMENT,
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRSTNAME</td>
<td>VARCHAR(50),</td>
</tr>
<tr>
<td>LASTNAME</td>
<td>VARCHAR(50),</td>
</tr>
<tr>
<td>ADDRESS_LINE1</td>
<td>VARCHAR(100),</td>
</tr>
<tr>
<td>ADDRESS_LINE2</td>
<td>VARCHAR(100),</td>
</tr>
<tr>
<td>GENDER</td>
<td>VARCHAR(1),</td>
</tr>
<tr>
<td>CITY</td>
<td>VARCHAR(50),</td>
</tr>
<tr>
<td>STATE</td>
<td>VARCHAR(2),</td>
</tr>
<tr>
<td>ZIP</td>
<td>NUMERIC(5,0),</td>
</tr>
<tr>
<td>PHONE1</td>
<td>NUMERIC(9,0),</td>
</tr>
<tr>
<td>PHONE2</td>
<td>NUMERIC(9,0),</td>
</tr>
<tr>
<td>PHONE3</td>
<td>NUMERIC(9,0),</td>
</tr>
<tr>
<td>WAIVER_SIGNED</td>
<td>VARCHAR(1),</td>
</tr>
<tr>
<td>CONTACT_PREFERENCE</td>
<td>VARCHAR(50),</td>
</tr>
<tr>
<td>EMAIL1</td>
<td>VARCHAR(100),</td>
</tr>
<tr>
<td>EMAIL2</td>
<td>VARCHAR(100),</td>
</tr>
<tr>
<td>START_DATE</td>
<td>DATE,</td>
</tr>
<tr>
<td>END_DATE</td>
<td>DATE,</td>
</tr>
<tr>
<td>TREATMENT_TYPE</td>
<td>VARCHAR(100),</td>
</tr>
<tr>
<td>TREATMENT_DESCR</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>STATEMENT</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>PERSON_TYPE</td>
<td>VARCHAR(50),</td>
</tr>
<tr>
<td>RACE_NUMBER</td>
<td>NUMERIC(10,0),</td>
</tr>
<tr>
<td>Attribute1</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute2</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute3</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute4</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute5</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute6</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute7</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute8</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute9</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute10</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute11</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute12</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute13</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute14</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute15</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute16</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute17</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute18</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute19</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute20</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute21</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute22</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute23</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute24</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute25</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute26</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute27</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute28</td>
<td>VARCHAR(250),</td>
</tr>
</tbody>
</table>
CREATE TABLE PRICE(
    PRICE_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    PRICE_AMT NUMERIC(10,2),
    PRICE_DESCRIPTION VARCHAR(250),
    GROUP_ID NUMERIC(10,0),
    TEAM_ID NUMERIC(10,0),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    Updated_by NUMERIC(10,0),
    UPdated_Date Date,
    PRIMARY KEY (PRICE_ID));

--Create table RESULTS table for the Race Database as a reference table to SPONSOR.
CREATE TABLE RESULTS(
    RESULT_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    EVENT_DETAIL_ID SMALLINT UNSIGNED NOT NULL REFERENCES EVENT_DETAIL,
    INDIVIDUAL_DETAIL_ID SMALLINT UNSIGNED NOT NULL REFERENCES INDIVIDUAL_DETAIL,
    START_TIME DATE,
    END_TIME DATE,
    FINISH_PHOTO_LINK VARCHAR(250),
    EFFECTIVE_START_DATE DATE,
    EFFECTIVE_END_DATE DATE,
    BAR_CODE_NUMBER NUMERIC(100,0),
    BIB_NUMBER NUMERIC(50,0),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    Updated_by NUMERIC(10,0),
    UPDATED_DATE Date,
    PRIMARY KEY (RESULT_ID));

--Create table AWARD table for the Race Database as a reference table to SPONSOR.
CREATE TABLE AWARD(
    AWARD_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    SPONSOR_ID SMALLINT UNSIGNED REFERENCES SPONSOR,
    EVENT_ID SMALLINT UNSIGNED REFERENCES EVENT,
    EVENT_DETAIL_ID SMALLINT UNSIGNED REFERENCES EVENT_DETAIL,
    AWARD_DESCRIPTION VARCHAR(250),
    AWARD_TYPE VARCHAR(250),
    EFFECTIVE_START_DATE DATE,
    EFFECTIVE_END_DATE DATE,
    BAR_CODE_NUMBER NUMERIC(50,0),
    BIB_NUMBER NUMERIC(50,0),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    Updated_by NUMERIC(10,0),
    UPDATED_DATE Date,
    PRIMARY KEY (AWARD_ID));

--Create table AWARDRESULT table for the Race Database as a reference table to
SPONSOR.

CREATE TABLE AWARD_RESULT(
    AWARD_RESULT_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    RESULT_ID SMALLINT UNSIGNED REFERENCES RESULT,  
    AWARD_ID SMALLINT UNSIGNED REFERENCES AWARD,  
    EFFECTIVE_START_DATE DATE,  
    EFFECTIVE_END_DATE DATE,  
    Attribute1 VARCHAR(250),  
    Attribute2 VARCHAR(250),  
    Attribute3 VARCHAR(250),  
    Attribute4 VARCHAR(250),  
    Attribute5 VARCHAR(250),  
    Attribute6 VARCHAR(250),  
    Attribute7 VARCHAR(250),  
    Attribute8 VARCHAR(250),  
    Attribute9 VARCHAR(250),  
    Attribute10 VARCHAR(250),  
    Attribute11 VARCHAR(250),  
    Attribute12 VARCHAR(250),  
    Attribute13 VARCHAR(250),  
    Attribute14 VARCHAR(250),  
    Attribute15 VARCHAR(250),  
    Attribute16 VARCHAR(250),  
    Attribute17 VARCHAR(250),  
    Attribute18 VARCHAR(250),  
    Attribute19 VARCHAR(250),  
    Attribute20 VARCHAR(250),  
    Attribute21 VARCHAR(250),  
    Attribute22 VARCHAR(250),  
    Attribute23 VARCHAR(250),  
    Attribute24 VARCHAR(250),  
    Attribute25 VARCHAR(250),  
    Attribute26 VARCHAR(250),  
    Attribute27 VARCHAR(250),  
    Attribute28 VARCHAR(250),  
    Attribute29 VARCHAR(250),  
    Attribute30 VARCHAR(250),  
    Updated_by NUMERIC(10,0),  
    UPdated_Date Date,  
    PRIMARY KEY (AWARD_RESULT_ID));

/*Create table REGISTRATION table for the Race Database as a reference table to SPONSOR.*

CREATE TABLE REGISTRATION(
    REGISTRATION_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    EVENT_ID SMALLINT UNSIGNED REFERENCES
EVENT_DETAIL_ID
INDIVIDUAL_DETAIL_ID
TEAM_ID
GROUP_ID
REGISTRATION_METHOD_ID
PARTICIPANT_TYPE_ID
WAIVER_SIGNED
REGISTRATION_DATE
BIB_NUMBER
BAR_CODE_NUMBER
EFFECTIVE_START_DATE
EFFECTIVE_END_DATE
Attribute1
Attribute2
Attribute3
Attribute4
Attribute5
Attribute6
Attribute7
Attribute8
Attribute9
Attribute10
Attribute11
Attribute12
Attribute13
Attribute14
Attribute15
Attribute16
Attribute17
Attribute18
Attribute19
Attribute20
Attribute21
Attribute22
Attribute23
Attribute24
Attribute25
Attribute26
Attribute27
Attribute28
Attribute29
Attribute30
Updated_by
UPdated_Date

EVENT,
SMALLINT UNSIGNED REFERENCES
EVENT_DETAIL,
SMALLINT UNSIGNED NOT NULL REFERENCES INDIVIDUAL_DETAIL,
SMALLINT UNSIGNED REFERENCES TEAM,
SMALLINT UNSIGNED REFERENCES RACE_GROUP,
SMALLINT UNSIGNED REFERENCES REGISTRATION_METHOD,
SMALLINT UNSIGNED REFERENCES PARTICIPANT_TYPE,
VARCHAR(1),
DATE,
NUMERIC(10,0),
NUMERIC(100,0),
DATE,
DATE,
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
VARCHAR(250),
NUMERIC(10,0),
Date,
--Create table REGISTRATION_METHOD table for the Race Database.
CREATE TABLE REGISTRATION_METHOD(
    REGISTRATION_METHOD_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    METHOD_DESCRIPTION VARCHAR(250),
    STORE_NAME VARCHAR(250),
    STORE_LOCATION VARCHAR(250),
    EVENT_DAY VARCHAR(1),
    ONLINE VARCHAR(1),
    MAIL VARCHAR(1),
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    PRIMARY KEY (REGISTRATION_METHOD_ID));

--Create table TEAM table for the Race Database.
CREATE TABLE TEAM(

CREATE TABLE TEAM(
    TEAM_ID SMALLINT UNSIGNED NOT NULL NOT NULL,
    TEAM_NAME VARCHAR(100),
    TEAM_DESCRIPTION VARCHAR(250),
    ORG_AFFILIATION VARCHAR(250),
    ORG_TYPE VARCHAR(250),
    KEY_MEMBER1 VARCHAR(100),
    KEY_MEMBER2 VARCHAR(100),
    KEY_MEMBER3 VARCHAR(100),
    EFFECTIVE_START DATE,
    EFFECTIVE_END DATE,
    Attribute1 VARCHAR(250),
    Attribute2 VARCHAR(250),
    Attribute3 VARCHAR(250),
    Attribute4 VARCHAR(250),
    Attribute5 VARCHAR(250),
    Attribute6 VARCHAR(250),
    Attribute7 VARCHAR(250),
    Attribute8 VARCHAR(250),
    Attribute9 VARCHAR(250),
    Attribute10 VARCHAR(250),
    Attribute11 VARCHAR(250),
    Attribute12 VARCHAR(250),
    Attribute13 VARCHAR(250),
    Attribute14 VARCHAR(250),
    Attribute15 VARCHAR(250),
    Attribute16 VARCHAR(250),
    Attribute17 VARCHAR(250),
    Attribute18 VARCHAR(250),
    Attribute19 VARCHAR(250),
    Attribute20 VARCHAR(250),
    Attribute21 VARCHAR(250),
    Attribute22 VARCHAR(250),
    Attribute23 VARCHAR(250),
    Attribute24 VARCHAR(250),
    Attribute25 VARCHAR(250),
    Attribute26 VARCHAR(250),
    Attribute27 VARCHAR(250),
    Attribute28 VARCHAR(250),
    Attribute29 VARCHAR(250),
    Attribute30 VARCHAR(250),
    Updated_by NUMERIC(10,0),
    UPDATED_DATE Date,
    PRIMARY KEY (TEAM_ID));

--Create table GROUP table for the Race Database.
CREATE TABLE RACE_GROUP(
    GROUP_ID SMALLINT UNSIGNED NOT NULL NOT NULL,
    GROUP_NAME VARCHAR(100),
    ...);
GROUP_DESCRIPTION VARCHAR(250),
ORG_AFFILIATION VARCHAR(250),
oRG_TYPE VARCHAR(250),
KEY_MEMEBER1 VARCHAR(100),
KEY_MEMEBER2 VARCHAR(100),
KEY_MEMEBER3 VARCHAR(100),
EFFECTIVE_START DATE,
EFFECTIVE_END DATE,
Attribute 1 VARCHAR(250),
Attribute 2 VARCHAR(250),
Attribute 3 VARCHAR(250),
Attribute 4 VARCHAR(250),
Attribute 5 VARCHAR(250),
Attribute 6 VARCHAR(250),
Attribute 7 VARCHAR(250),
Attribute 8 VARCHAR(250),
Attribute 9 VARCHAR(250),
Attribute 10 VARCHAR(250),
Attribute 11 VARCHAR(250),
Attribute 12 VARCHAR(250),
Attribute 13 VARCHAR(250),
Attribute 14 VARCHAR(250),
Attribute 15 VARCHAR(250),
Attribute 16 VARCHAR(250),
Attribute 17 VARCHAR(250),
Attribute 18 VARCHAR(250),
Attribute 19 VARCHAR(250),
Attribute 20 VARCHAR(250),
Attribute 21 VARCHAR(250),
Attribute 22 VARCHAR(250),
Attribute 23 VARCHAR(250),
Attribute 24 VARCHAR(250),
Attribute 25 VARCHAR(250),
Attribute 26 VARCHAR(250),
Attribute 27 VARCHAR(250),
Attribute 28 VARCHAR(250),
Attribute 29 VARCHAR(250),
Attribute 30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPdated_Date Date,

PRIMARY KEY (GROUP_ID));

--Create table SPONSOR_REGISTRATION table for the Race Database as a reference table to SPONSOR.
CREATE TABLE
SPONSOR_REGISTRATION(
    SPONSOR_REGISTRATION_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    SPONSOR_ID SMALLINT UNSIGNED NOT NULL REFERENCES SPONSOR,
EVENT_ID           SMALLINT UNSIGNED REFERENCES
EVENT_DETAIL_ID    SMALLINT UNSIGNED REFERENCES
SPONSOR_ACTION_ID  SMALLINT UNSIGNED REFERENCES
REGISTRATION_DATE  SMALLINT UNSIGNED REFERENCES
EFFECTIVE_START_DATE  SMALLINT UNSIGNED REFERENCES
SPONSOR_ACTION,   DATE,
DATE,
DATE,
Attribute1          VARCHAR(250),
Attribute2          VARCHAR(250),
Attribute3          VARCHAR(250),
Attribute4          VARCHAR(250),
Attribute5          VARCHAR(250),
Attribute6          VARCHAR(250),
Attribute7          VARCHAR(250),
Attribute8          VARCHAR(250),
Attribute9          VARCHAR(250),
Attribute10         VARCHAR(250),
Attribute11         VARCHAR(250),
Attribute12         VARCHAR(250),
Attribute13         VARCHAR(250),
Attribute14         VARCHAR(250),
Attribute15         VARCHAR(250),
Attribute16         VARCHAR(250),
Attribute17         VARCHAR(250),
Attribute18         VARCHAR(250),
Attribute19         VARCHAR(250),
Attribute20         VARCHAR(250),
Attribute21         VARCHAR(250),
Attribute22         VARCHAR(250),
Attribute23         VARCHAR(250),
Attribute24         VARCHAR(250),
Attribute25         VARCHAR(250),
Attribute26         VARCHAR(250),
Attribute27         VARCHAR(250),
Attribute28         VARCHAR(250),
Attribute29         VARCHAR(250),
Attribute30         VARCHAR(250),
Updated_by          NUMERIC(10,0),
UPdated_Date        Date,

PRIMARY KEY
(SPONSOR_REGISTRATION_ID));

--Create table SPONSOR_ACTION table for the Race Database.
CREATE TABLE SPONSOR_ACTION(
  SPONSOR_ACTION_ID  SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  ACTION_NAME         VARCHAR(100),
DESCRIPTION VARCHAR(250),
MONETARY_VALUE NUMERIC(20,0),
CATEGORY VARCHAR(100),
EFFECTIVE_START DATE,
EFFECTIVE_END DATE,
Attribute1 VARCHAR(250),
Attribute2 VARCHAR(250),
Attribute3 VARCHAR(250),
Attribute4 VARCHAR(250),
Attribute5 VARCHAR(250),
Attribute6 VARCHAR(250),
Attribute7 VARCHAR(250),
Attribute8 VARCHAR(250),
Attribute9 VARCHAR(250),
Attribute10 VARCHAR(250),
Attribute11 VARCHAR(250),
Attribute12 VARCHAR(250),
Attribute13 VARCHAR(250),
Attribute14 VARCHAR(250),
Attribute15 VARCHAR(250),
Attribute16 VARCHAR(250),
Attribute17 VARCHAR(250),
Attribute18 VARCHAR(250),
Attribute19 VARCHAR(250),
Attribute20 VARCHAR(250),
Attribute21 VARCHAR(250),
Attribute22 VARCHAR(250),
Attribute23 VARCHAR(250),
Attribute24 VARCHAR(250),
Attribute25 VARCHAR(250),
Attribute26 VARCHAR(250),
Attribute27 VARCHAR(250),
Attribute28 VARCHAR(250),
Attribute29 VARCHAR(250),
Attribute30 VARCHAR(250),
Updated_by NUMERIC(10,0),
UPdated_Date Date,
PRIMARY KEY (SPONSOR_ACTION_ID));

--Create table Participant_Type table for the Race Database.
CREATE TABLE Participant_Type(
  Participant_Type_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  Participant_Type VARCHAR(100),
  PARTICIPANT_DESCRIPTION VARCHAR(250),
  EFFECTIVE_START DATE,
  EFFECTIVE_END DATE,
  Attribute1 VARCHAR(250),
  ...
CREATE TABLE Family(
  Family_ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  MOTHER1 NUMERIC(10,0),
  MOTHER2 NUMERIC(10,0),
  FATHER1 NUMERIC(10,0),
  FATHER2 NUMERIC(10,0),
  CHILD1 NUMERIC(10,0),
  CHILD2 NUMERIC(10,0),
  CHILD3 NUMERIC(10,0),
  CHILD4 NUMERIC(10,0),
  CHILD5 NUMERIC(10,0),
  CHILD6 NUMERIC(10,0),
  CHILD7 NUMERIC(10,0),
  PRIMARY KEY (Participant_Type_ID));
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHILD8</td>
<td>NUMERIC(10,0)</td>
</tr>
<tr>
<td>CHILD9</td>
<td>NUMERIC(10,0)</td>
</tr>
<tr>
<td>CHILD10</td>
<td>NUMERIC(10,0)</td>
</tr>
<tr>
<td>Attribute1</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute2</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute3</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute4</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute5</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute6</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute7</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute8</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute9</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute10</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute11</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute12</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute13</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute14</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute15</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute16</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute17</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute18</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute19</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute20</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute21</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute22</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute23</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute24</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute25</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute26</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute27</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute28</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute29</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Attribute30</td>
<td>VARCHAR(250),</td>
</tr>
<tr>
<td>Updated_by</td>
<td>NUMERIC(10,0),</td>
</tr>
<tr>
<td>UPdated_Date</td>
<td>Date,</td>
</tr>
<tr>
<td>PRIMARY KEY</td>
<td>(FAMILY_ID))</td>
</tr>
</tbody>
</table>
Appendix C – Project Gnatt Chart
<table>
<thead>
<tr>
<th>Stage Description</th>
<th>From</th>
<th>To</th>
<th>Jan-02</th>
<th>Feb-02</th>
<th>Mar-02</th>
<th>Apr-02</th>
<th>May-02</th>
<th>Jun-02</th>
<th>Jul-02</th>
<th>Aug-02</th>
<th>Sep-02</th>
<th>Oct-02</th>
<th>Nov-02</th>
<th>Dec-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>6030 Presentation of Technical Material</td>
<td>05/02</td>
<td>06/26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6410 Applied Systems Analysis &amp; Design</td>
<td>07/01</td>
<td>08/31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5700 Survey of Programming Logic</td>
<td>09/02</td>
<td>10/01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6300 Computer System Architecture</td>
<td>11/03</td>
<td>12/03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6900 PL/SQL Programming</td>
<td>01/04</td>
<td>02/24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6520 Application Development with Oracle Forms</td>
<td>03/05</td>
<td>04/25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6700 Database Administration</td>
<td>05/02</td>
<td>06/26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6300 Database Architecture</td>
<td>11/03</td>
<td>12/03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6250 Database Concepts</td>
<td>11/03</td>
<td>12/03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6100 Java Programming</td>
<td>01/04</td>
<td>02/24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6700 Enterprise Portal Design</td>
<td>02/05</td>
<td>03/25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6705 Professional Project Proposal</td>
<td>07/26</td>
<td>08/24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6705 Professional Project Presentation</td>
<td>07/26</td>
<td>08/24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Hand Off</td>
<td>07/26</td>
<td>08/24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Regis University
School for Professional Studies Graduate Programs
Professional Project

Certification of Authorship of Professional Project Work

Print Student's Name: ANALIZA HILT

Telephone: 970-461-5040 Email: AJ.CURLER@YAHOO.COM

Date of Submission: DECEMBER 9, 2005 Degree Program: MSCIT

Title of Submission: THE CHERRY CREEK SNEAK DATABASE SOLUTION

Advisor/Faculty Name: Edward W. Thompson

Certification of Authorship:
I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the MSCIT Degree Program.

[Signature]
Student Signature

[Date]
December 5, 2005
Regis University
School for Professional Studies Graduate Programs

Professional Project Approval Form

Advisor/Professional Project Faculty Approval Form

Student's Name: ______________________________________________________________________
Program: ___________________________ MSCIT

Professional Project Title: ______________________________________________________________________

Advisor Name: ___________________________

Project Faculty Name: ___________________________

Advisor/Faculty Declaration:
I have advised this student through the Professional Project Process and approve of the final
document as acceptable to be submitted as fulfillment of partial completion of requirements
for
the __________________________ Degree Program.

December 6, 2005
Original Signature

Chairperson Project Faculty Approval:

December 6, 2005
Original Signature
REGIS UNIVERSITY
School for Professional Studies
Graduate Programs Final Project/Thesis

Authorization to Publish Student Work on WWW

I, ____________________________, the undersigned student,
in the __________________________ Degree Program hereby authorize Regis University to publish through

a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: THE CHERRY CREEK SNAKE DATABASE SOLUTION

__________________________________________
Student Signature

10-3-05
Date

Complete if applicable:

☐ The Work contains private or proprietary information of the following parties and their attached
permission is required as well: ____________________________

☐ I do not authorize Regis University to publish my work on the WWW.

__________________________________________
Student Signature

Date