**Task Manager**

**Database Management**

**308N-201**

**RedFoxTask**

****

Marist College

School of Computer Science and Mathematics

Submitted to:

Dr. Reza Sadeghi

9/18/2023

**Project Report of Red Fox Task**

**Team Name**

RedFoxTask

**Team Members**

1. Sam Guggino [Samantha.Guggino1@marist.edu](mailto:Sam.Guggino1@marist.edu) Team Head
2. Cody Carruthers [Cody.Carruthers1@marist.edu](mailto:Cody.Carruthers1@marist.edu) Team Member
3. Chase Evans [Chase.Evans1@marist.edu](mailto:Chase.Evans1@marist.edu) Team Member
4. Riley Penn [Riley.Penn1@marist.edu](mailto:Riley.Penn1@marist.edu) Team Member
5. Kevin Reiff [Kevin.Reiff1@marist.edu](mailto:Kevin.Reiff1@marist.edu) Team Member
6. Abigail Lee [Abigail.Lee1@marist.edu](mailto:Abigail.Lee1@marist.edu) Team Member

**Description of Team Members**

1. Sam Guggino

I am an Information Systems and Technology major with a concentration in Information Technology. I have always liked working with and on computers – seeing how things work and how the smaller pieces come together to make one larger final product. I wanted to work with my teammates because I knew one member, and I feel like we will all work well together. We chose our leader based on our schedules, and who was able to pick up the extra responsibilities.

1. Cody Carruthers

I am a Game Design major from Stratford CT. I have been a coder for a long time and used to teach game design to children when I was back at home. I wanted to work with these teammates because I knew a few of them and I think we could work together as a team. We chose the team leader based on who would have the best responsibilities and who we think could help us get the best work done and keep us on the task at hand.

1. Chase Evans:

I am a Cybersecurity + Games and Emerging Media dual major. I have had a passion for working with computers for as long as I can remember and I will always relish the opportunity to learn more. I decided to work with this team as they all seem to be thoughtful, welcoming, and have a good head on their shoulders. We chose the team leader based around our responsibilities outside of the group setting and who would be best equipped to handle the role.

1. Riley Penn

I am a Computer Science major with a concentration in Game Design and a minor in Graphic Design. I have always loved being able to create something that brings joy and purpose, and the exponential rise in technology as I have grown up brought me interest in computers. I wanted to work with these teammates because I had known three of them previously and knew with our similar interests that we would have the set of skills needed to complete this project. We chose our team leader based on responsibilities and who had the most time to commit to leading this project.

1. Kevin Reiff - I am a computer science major from Vernon NJ. I have been working as an intern for the IT Security Department at Selective Insurance since the beginning of the summer and I am still working for them throughout the semester. I wanted to work with these teammates because I knew two of them previously and we all share similar interests. We chose the team head via discussing out of class responsibilities and who was best equipped to take on the responsibility.
2. Abigail Lee

I am a computer science major with a minor in cybersecurity from Poughkeepsie, NY. Technology has always interested me and how everything works just from a few simple lines or complex code. My dad is an engineer so he was always pushing STEM on me and I loved it! I wanted to work with my teammates because everyone in this group is so intelligent and they all make a great team as they have history together.

**Table of Contents**

Project Objective …………………………………...……………….…………………………. 5

Related Work Review ……………………………...……………….…………………………. 6

Project Merits …………………………………...……………….……………………………., 7

Github Repository Address ………………………...……………….…………………………. 9

Phase 2: ER Model and EER Model …………………………………………………………. 10

Phase 3: Presentation and SQL Files …………………………………………………………. 12

References …………………………………………………………………………………….. 16

**Project Objective**

**Project Title:** Red Fox Task

The Red Fox Task is a task management system that will display a calendar for the selected day, week, month or year. It organizes the specific tasks of different users each day, and users have the ability to update these tasks at any time. The calendar should update in day, week, month and year form when a task is added. The task management system will store the data of different users in distinct SQL tables.

This task management system includes multi-level user authentication, a dynamic calendar display, personalized task management, an advanced search function, a user friendly interface, data reporting, alerts and warnings, and an easy exit function.

The multi-level user authentication gives access to the system to both users and admins. Users are able to manage their own task pages, while admins have access to security pages as well as the pages of regular users. The calendar display gives users options to choose their preference when it comes to listing their tasks, whether that is in an annual, monthly, weekly, or daily view. The personalization of the task manager continues, as users can edit each part of the tasks they create, all the way down to the color it displays as. The task management system includes an advanced search function, where users can search within their tasks using a variety of parameters. All of these features are presented in a user friendly interface that is easy to maneuver and is aesthetically pleasing to the eye. For easier functionality, an alert system is integrated that will alert users if there are overlapping dates or tasks. Users have the ability to easily exit the task management system with a simple logout option.

**Related Work Review:**

Trello - A good task manager that is used for business. It uses all-around good software, where you can keep track of tasks done, upcoming tasks, tasks to do, and tasks you are in the middle of completing[3]. However, Trello does not have a way to directly message people if something needs to be done. This would help a lot in our application.

Jira - Another good task manager which uses a timeline to create tasks that you have to do. It shows the start and end date on a type of calendar that goes throughout months[2]. However, it is a bit too confusing. You can make teams, but the tasks are confusing and awkward to look at. I think they should make it simpler and more user friendly to make people want it.

Todoist - Another great task managing software which is used by huge companies such as Disney, Microsoft, Amazon, and Netflix. This offers an easy-to-manage and visually pleasing to do list of everything you need to complete. It shows personal tasks for yourself, team tasks if you have a team company you are working with, and different workspaces to list your groceries, fitness routines, appointments, and personal goals. You can also join teams to create joint tasks[1]. Just like with Trello, what would make this better is a messaging software within it to contact your team at any time.

**Project Merits**

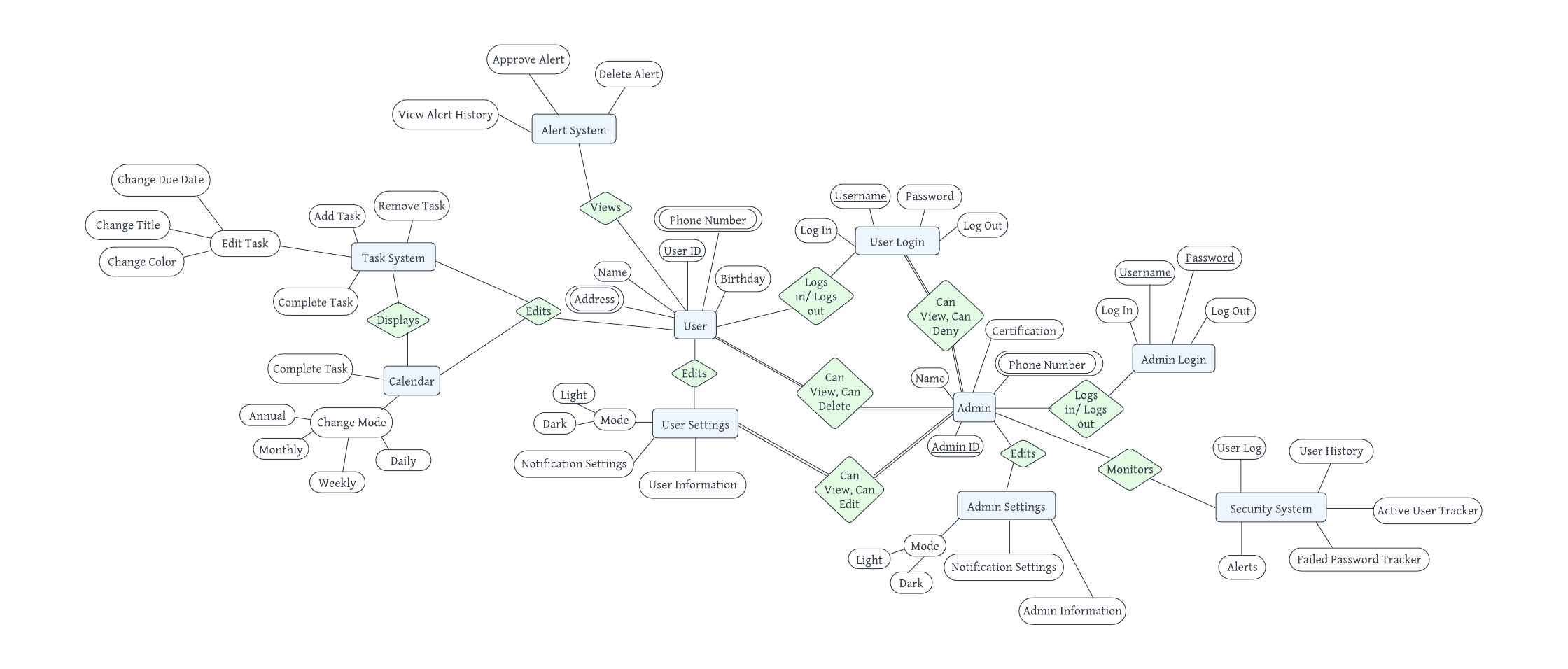
1. Core Capabilities:
   1. **Multi-Level User Authentication:** Both admin and standard user roles are available in our task managing system, each with unique rights. This improves security and manageability by ensuring that the appropriate users have access to the appropriate functionality. Unlike Trello and Todoist, our task managing system includes a dedicated admin role for enhanced security and user management.
   2. **Dynamic Calendar Display:** Users may choose between weekly, monthly, and annual views, giving them the flexibility to plan and arrange work as needed. While Jira provides a timeline, our task managing system offers a more straightforward calendar display that is easier to interpret.
   3. **Personalized Task Management:** Tasks particular to each account can be added, modified, or removed by the user. Each assignment has necessary details including the title, start and end times, as well as a description. Unlike Jira, our task managing system aims for simplicity and user-friendliness in task management.
   4. **Advanced Search Functionality:** Users have the option of searching for tasks using several parameters, including duration, title, and time. This makes finding certain activities simple and quick. This search functionality exceeds the capabilities of Trello, which lacks an advanced search feature based on multiple parameters.
   5. **User-Friendly Interface:** Our task managing system will have an initial welcome page as well as a structured menu of all functions on each page, making it easy to use and intuitive. Unlike Jira, which can be confusing to new users, our task manager system focuses on user experience..
   6. **Data Reporting:** To assist users in their responsibilities in a structured fashion, the system can create tabular reports that include well-organized calendars. Compared to Todoist, this function is more comprehensive.
   7. **Alerts and Warnings:** Our system will notify users anytime they try to input contact information that already exists in the database in order to prevent repetition. This feature is not common in the other task management systems mentioned.
   8. **Graceful Exit:** The user experience is improved via a special exit function that thanks the user for using the program.
2. Why Choose Our Product?
   1. **Highly Customizable:** Our task managing system may be customized to meet individual needs thanks to the option to switch between calendar views and alter task information.
   2. **Secure User Management:** Our task managing system prioritizes data security and will have separate SQL databases for various user categories as well as strict password requirements.
   3. **Efficiency:** Users can better manage their time and responsibilities with the help of the sophisticated search capabilities and tabular reports that our system will have.
   4. **User-Centric Design**: Even individuals with rudimentary technological knowledge can utilize the system with ease because of its user-friendly interface.
   5. **Data integrity:** The built-in alter system that we will implement into our task managing system will prevent duplicate entries, maintaining the reliability of the stored data.
   6. **Comprehensive Solution:** Our task managing system will provide a one-stop shop for all task management requirements, covering task creation, reporting and even user administration (for admins).

**Github Repository Address**

GitHub Repository: <https://github.com/SamGuggino/CMPT308_TaskManager_RedFoxTask>

**Phase 2: ER Model and EER Model:**

**ER Model:**

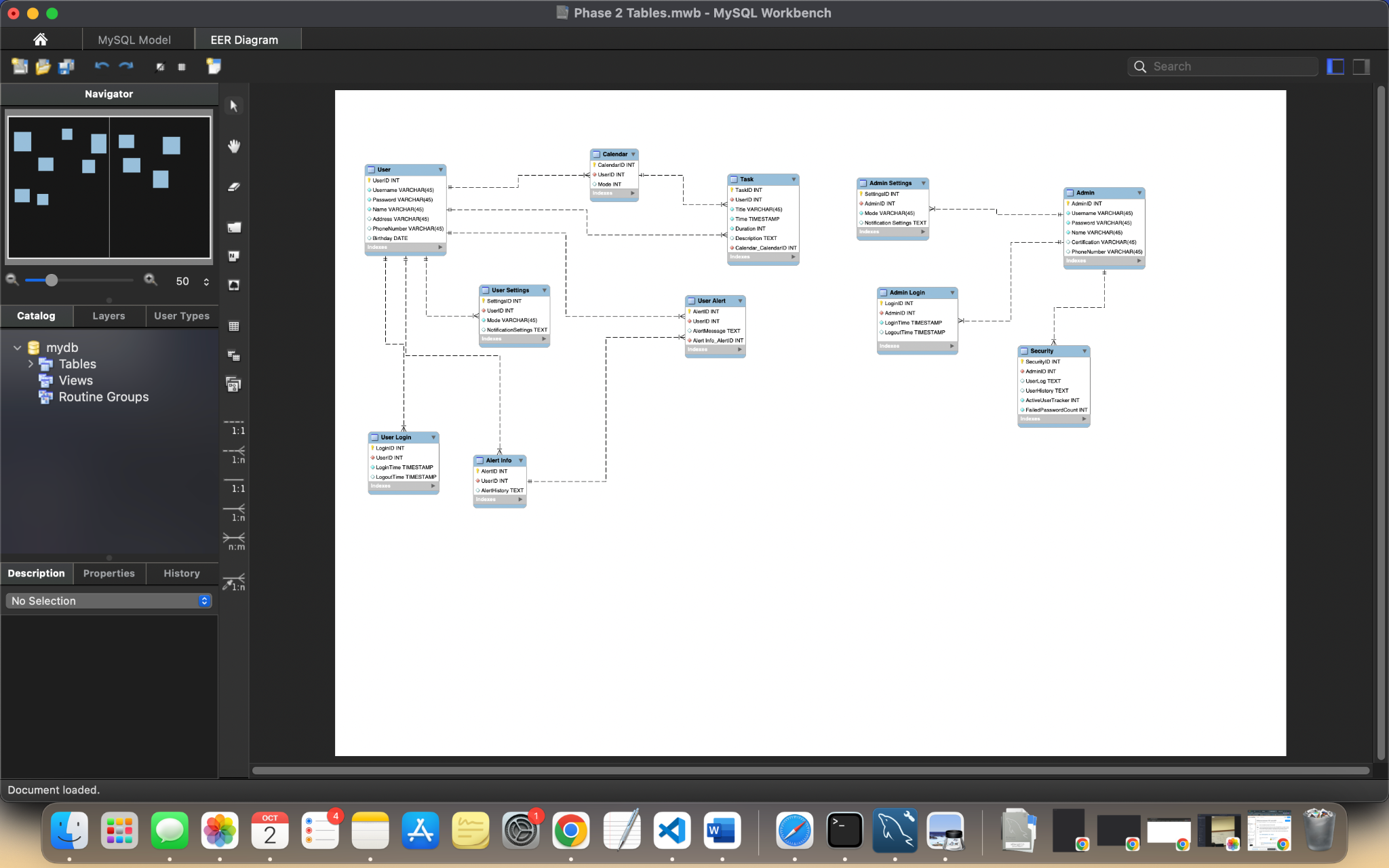


In our Entity Relationship Model, each entity represents what the user and admin can do with RedFoxTask. When we created this, we wanted an easy to read diagram that clearly states the required data and where it will be stored. The diagram has a many-to-many cardinality as users and admin can work on multiple tasks at once.

As a user, you have all of the basic information such as user ID, name, birthday, address, and phone number. They can also update in user settings the notifications, user information and view the display in either light or dark mode. The user can also view the alert system which can approve, delete or pull up alert history. The task system allows you to complete, remove, and add tasks as well as edit the style of each one. The task system also can display a calendar view where users can view a daily, weekly, monthly or annual calendar as they check off their to-do list.

As an admin, you have your name, admin ID, phone number, and certification as your basic information. Just like the user, admins can change their notifications, information, and display mode in admin settings as well as having the same login page. The user has a partial participation as they have no control over what the admin can do because they have total participation. This means that the admin differs from the user as they can view, edit, delete and deny the user, user settings, and logins. The admin can also monitor the security system, which provides a user log, user history, user tracker, failed password tracker, and alerts.

**EER Model:**



With the Enhanced Entity Relationship Model, you can see that the user and the admin have one to many relationships connecting to the tasks and settings in their system. The primary keys are the IDs in each of the tables while the foreign keys are the user IDs or alert information in each table. The user has a relationship with all of the entities such as user settings, alert and user alert information, calendar, tasks, and user login while the admin has a similar layout where they have control over admin settings, admin login, and security.

**Phase 3: Presentation and SQL Files:**

**Presentation:**

Slide 2: Default Values

* A DEFAULT clause indicates a default value for a column.
* It can be a literal constant or an expression.
* Expression values are put in parentheses.
* For TIMESTAMP and DATETIME columns, CURRENT\_TIMESTAMP can be used without parentheses.
* SERIAL DEFAULT VALUE is used for an integer column, acting as an alias for NOT NULL AUTO\_INCREMENT UNIQUE.
* BLOB, TEXT, GEOMETRY, and JSON data types can only be assigned default values if they are written as expressions.

Slide 3: Rules for Expression Default Values

* Allowed constructs: Literals, built-in functions, and operators.
* Disallowed constructs: Subqueries, parameters, variables, stored functions, and loadable functions.
* An expression default cannot use a column that has an AUTO\_INCREMENT attribute.
* An expression default value for one column can refer to other table columns, but references must be to columns that appear earlier in the table definition.
* This rule also applies to ALTER TABLE to reorder columns.

Slide 4: Altering Tables

* For CREATE TABLE ... LIKE and CREATE TABLE ... SELECT, the destination table uses default values from the original table.
* If an expression default value has a nondeterministic function, it cannot be used for statement-based replication.
* This includes INSERT and UPDATE.
* If binary logging is disabled, the statement will execute.
* If binary logging is enabled and binlog\_format is set to STATEMENT, the statement will execute but a warning will be sent to the error log.
* If binlog\_format is set to MIXED or ROW, the statement will execute.

Slide 5: Inserting Rows

* To insert a new row, the default value of a column with an expression value can be inserted by removing the column name or setting it to DEFAULT.
* The use of DEFAULT(col\_name) to specify the default value for a named column can only be used for columns that have a literal default value.
* Not all storage engines allow expression values.
* ER\_UNSUPPORTED\_ACTION\_ON\_DEFAULT\_VAL\_GENERATED will occur.

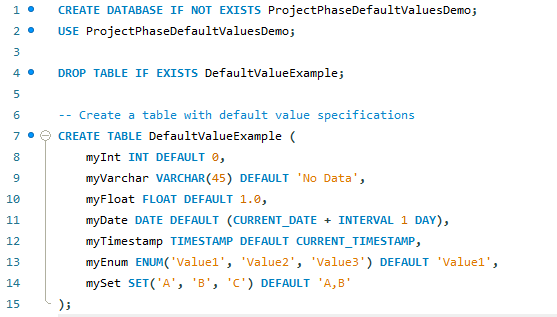
Slide 6: Implicit Default Handling

* If there is no explicit DEFAULT value, MySQL will:
* Define the column with DEFAULT NULL clause if the column can take NULL as a value.
* Define the column with no explicit DEFAULT clause if null cannot be used.
* For data entry into a NOT NULL column:
* In strict SQL mode, an error will occur.
* In non-strict mode, MySQL sets the column to the implicit default value for the column data type.

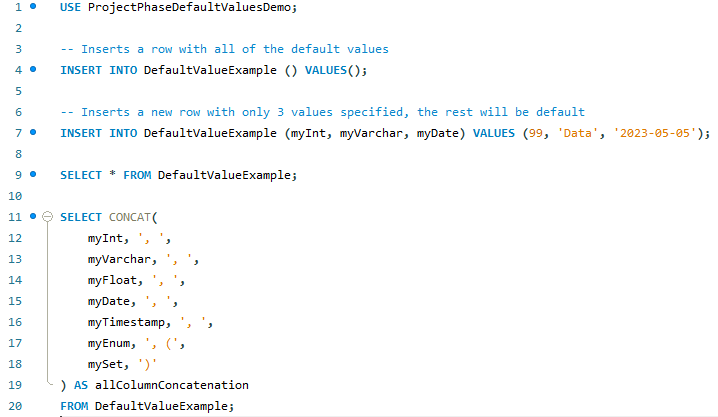
Slide 7: Implicit Defaults

* SHOW CREATE TABLE statement reveals columns with an explicit DEFAULT clause.
* Numeric types default to 0.
* Integer or floating-point types with AUTO\_INCREMENT attribute default to the next value.
* Date and time types (except TIMESTAMP) default to the correct "zero" value.
* TIMESTAMP's default depends on explicit\_defaults\_for\_timestamp system variable.
* If enabled, TIMESTAMP defaults to the current date and time.
* Otherwise, it defaults to the current date and time if it's the first TIMESTAMP column.
* Default value for non-ENUM string types is the empty string.
* ENUM defaults to the first enumeration value.

Slide 8: Demo 1



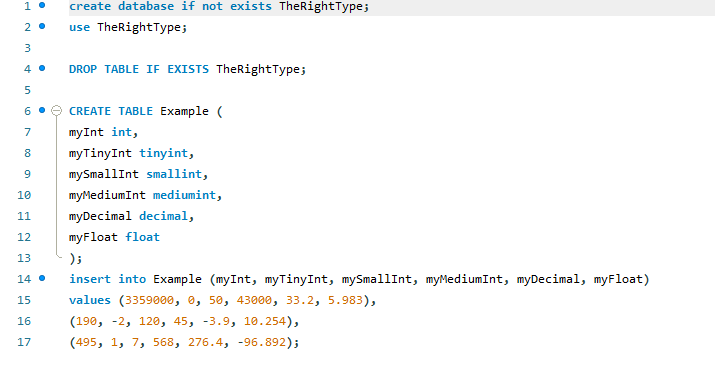
Slide 9: Demo 2



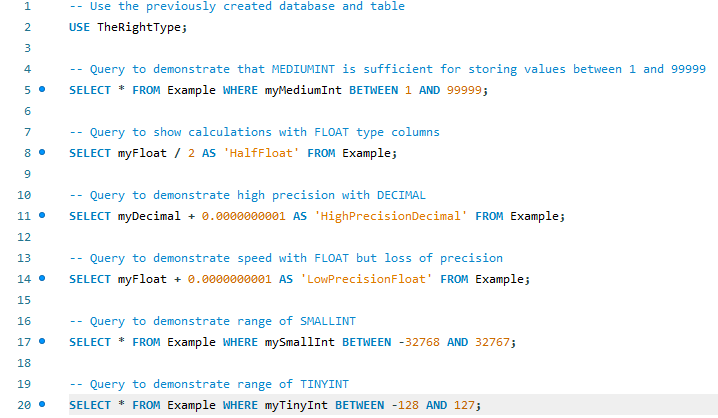
Slide 10: Choosing the Right Type for a Column

* Optimal storage: Prefer the most precise data type in most cases.
* Example: Use MEDIUMINT UNSIGNED for values 1-99999 in an integer column.
* DECIMAL column calculations: Have a precision of 65 decimal digits for basic operations (+, -, \*, and /).
* Speed over accuracy: Consider using the DOUBLE type when precision isn't critical.
* High precision option: Convert to a fixed-point type stored in a BIGINT for precision.
* Perform calculations with 64-bit integers and convert results back to floating-point values as needed.

Slide 11: Demo 3



Slide 12: Demo 4



**References:**

1. “A To-Do List to Organize Your Work & Life.” *Todoist*, todoist.com/. Accessed 30 Sep. 2023.
2. Atlassian. “Jira: Issue & Project Tracking Software.” *Atlassian*,   
    www.atlassian.com/software/jira. Accessed 30 Sep. 2023.
3. “Trello Brings All Your Tasks, Teammates, and Tools Together.” *Trello*, trello.com/home.   
    Accessed 30 Sep. 2023.