SCRUMMY 1.0 DATABASE DOCUMENTATION

1. Adminstrate Table

Description:

This table manages administrative user data in the system. It stores administrators' personal and authentication details, including their names, usernames, passwords, and email addresses.

Columns:

	Name	Data type	Description
	idAdmin	INT	Unique identifier for an Admin.
	FirstName	VARCHAR(45)	First name of the Admin.
	LastName	VARCHAR(45)	Last name of the Admin.
	Username	VARCHAR(255)	Username for the Admin.
	Password	VARCHAR(45)	Password for the Admin.
	Email	VARCHAR(45)	Email address of the Admin.

Relationships:

Primary table for administrative users. Does not have direct relationships mentioned.

2. Project_has_User Table

Description:

Acts as a junction table between `Project` and `User` tables. It is used to associate users with projects, indicating which users participate in or access specific projects.

Columns:

	Name	Data type	Description
	User_idUser	INT	Foreign key referencing User.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

Links `User` and `Project` tables to associate users with projects they are part of.

3. User Table

Description:

Stores information about the users in the system. This includes basic personal information, contact details, and authentication data (username and password). It also links each user to their role in the system, indicating their permissions and level of access.

Columns:

	Name	Data type	Description
🔑	idUser	INT	Unique identifier for a User.
	FirstName	VARCHAR(45)	First name of the User.
	LastName	VARCHAR(45)	Last name of the User.
	Username	VARCHAR(255)	Username for the User.
	Password	VARCHAR(64)	Password for the User.
	Email	VARCHAR(45)	Email address of the User.
	Role_idRole	INT	Foreign key referencing Role.(Nullable)

Relationships:

Assigned to a `Role` (linked to `Role` table).

Participates in various `Projects` (through `Project_has_User` table).

4. ConfluenceGroup Table

Description:

Manages different groups within the system, likely for collaboration or organizational purposes. Each group is associated with a project, suggesting a structure for managing project-specific teams or working groups.

Columns:

	Name	Data type	Description
🔑	idGroup	INT	Unique identifier for a Confluence Group.
	GroupName	VARCHAR(45)	Name of the Group.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

Groups are associated with specific `Projects`, likely for project management or access control purposes.

5. Page Table

Description:

Holds data for individual pages, possibly within a project documentation or wiki system. This includes the content of the pages and their association with specific confluence groups and projects.

Columns:

	Name	Data type	Description
🔑	idPage	INT	Unique identifier for a Page.
	PageName	VARCHAR(45)	Name of the Page.
	Content	MEDIUMTEXT	Content of the Page. (Nullable)
	ConfluenceGroup_idGroup	INT	Foreign key referencing ConfluenceGroup.
	ConfluenceGroup_Project_idProject	INT	Foreign key referencing Project within ConfluenceGroup.

Relationships:

Pages are linked to specific `ConfluenceGroups`, suggesting a documentation or wiki-like structure within projects.

6. ProductBacklog Table

Description:

Represents the product backlog in a Scrum framework. It stores the list of all tasks, features, and requirements that need to be addressed in a project. Each entry is linked to a specific project.

Columns:

	Name	Data type	Description
🔑	idProductBacklog	INT	Unique identifier for a Product Backlog.
	Name	VARCHAR(45)	Name of the Backlog.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

Each `Project` has a `ProductBacklog`, a key component in Scrum for managing the list of work to be done.

7. Project Table

Description:

Contains details of projects managed within the system. This includes basic information like the project name and description, as well as a link to the responsible administrator.

Columns:

	Name	Data type	Description
🔑	idProject	INT	Unique identifier for a Project.
	ProjectName	VARCHAR(45)	Name of the Project.
	Description	VARCHAR(255)	Description of the Project. (Nullable)
	Adminstrate_idAdmin	INT	Foreign key referencing Admin.

Relationships:

Projects are possibly overseen by an `Admin`. They contain `ProductBacklogs` and are linked to `Users` via `Project_has_User`.

8. Sprint Table

Description:

Manages information about sprints, which are short, time-boxed periods when a specific set of work has to be completed. This table includes details about the sprint duration, title, and its associated project.

Columns:

	Name	Data type	Description
🔑	idSprint	INT	Unique identifier for a Sprint.
	StartDate	DATETIME	Start date of the Sprint.
	EndDate	DATETIME	End date of the Sprint.
	Title	VARCHAR(255)	Title of the Sprint.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

Each `Sprint` is part of a `Project`, representing a time-boxed effort to complete a set of tasks.

9. SprintBacklog Table

Description:

Stores items that are part of a sprint backlog. Each item is a task or a feature that needs to be completed during the sprint. This table links back to the sprints and the overarching project.

Columns:

	Name	Data type	Description
🔑	idSprintBacklog	INT	Unique identifier for a Sprint Backlog.
	Priority	VARCHAR(45)	Priority of the backlog item. (Nullable)
	Status	VARCHAR(45)	Status of the backlog item.
	Sprint_idSprint	INT	Foreign key referencing Sprint.
	Sprint_Project_idProject	INT	Foreign key referencing Project within a Sprint.

Relationships:

Contains items to be completed in a specific `Sprint`, linked to both the `Sprint` and its parent `Project`.

10. TaskSB Table

Description:

Represents individual tasks within a sprint backlog. It includes details about the task such as title, description, status, and priority. Each task is assigned to a specific sprint backlog item.

Columns:

	Name	Data type	Description
🔑	idTask	INT	Unique identifier for a Task.
	Title	VARCHAR(45)	Title of the Task.
	Description	VARCHAR(255)	Description of the Task. (Nullable)
	Status	VARCHAR(45)	Status of the Task.
	Priority	INT	Priority of the Task. (Nullable)
	Assignee	INT	User ID of the person assigned to the Task. (Nullable)
	SprintBacklog_idSprintBacklog	INT	Foreign key referencing SprintBacklog.
	SprintBacklog_Sprint_idSprint	INT	Foreign key referencing Sprint within SprintBacklog.
	SprintBacklog_Sprint_Project_idProject	INT	Foreign key referencing Project within Sprint.

Relationships:

- **SprintBacklog:** Each task in the `TaskSB` table would be associated with an entry in the `Sprint Backlog` table, indicating which sprint backlog the task belongs to.
- **Assignee/User:** There may be a relationship to a `User` table, indicating which user (if any) is assigned to complete the task.
- **SubtaskSB:** If the system includes user Subtasks as separate entities, there might be relationships to these tables, showing how tasks are broken down into smaller parts.

11. User_Role_Project Table

Description:

This table to be a complex junction table linking users, their roles, and projects. It's likely used to manage and track the specific roles users hold within different projects.

Columns:

	Name	Data type	Description
	User_idUser	INT	Foreign key referencing User.
	Role_idRole	INT	Foreign key referencing Role.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

This table to link `Users`, their `Roles`, and the `Projects` they are associated with, possibly to manage access and responsibilities within projects.

12. SubtaskSB Table

Description:

Manages subtasks under specific tasks in a sprint backlog. This table is used for breaking down tasks into smaller, more manageable units of work, each with its own status, priority, and assignee.

Columns:

	Name	Data type	Description
🔑	idSubtask	INT	Unique identifier for a Subtask.
	Title	VARCHAR(45)	Title of the Subtask.
	Description	VARCHAR(255)	Description of the Subtask. (Nullable)
	Status	VARCHAR(45)	Status of the Subtask.
	Assignee	VARCHAR(255)	User ID of the person assigned to the Subtask. (Nullable)
	Priority	INT	Priority of the Subtask. (Nullable)
	TaskSB_idTask	INT	Foreign key referencing TaskSB.
	TaskSB_SprintBacklog_idSprintBacklog	INT	Foreign key referencing SprintBacklog within a Task.
	TaskSB_SprintBacklog_Sprint_idSprint	INT	Foreign key referencing Sprint within SprintBacklog.
	TaskSB_SprintBacklog_Sprint_Project_idProject	INT	Foreign key referencing Project within a Sprint.

Relationships:

Represents smaller tasks (`Subtasks`) that are part of a larger `TaskSB`, which is itself part of a `SprintBacklog`.

13. UserStoryPB Table

Description:

Contains user stories in the product backlog. User stories are short, simple descriptions of a feature told from the perspective of the user. This table links each story to the product backlog and the assigned sprint.

Columns:

	Name	Data type	Description
🔑	idUserStoryPB	INT	Unique identifier for a User Story in the Product Backlog.
	Title	VARCHAR(45)	Title of the User Story.
	Description	VARCHAR(45)	Description of the User Story. (Nullable)
	Status	VARCHAR(45)	Status of the User Story.
	Priority	INT	Priority of the User Story. (Nullable)
	Assignee	INT	User ID of the person assigned to the User Story. (Nullable)
	ProductBacklog_idProductBacklog	INT	Foreign key referencing ProductBacklog.
	ProductBacklog_Project_idProject	INT	Foreign key referencing Project within ProductBacklog.
	AssignedSprint	VARCHAR(45)	Sprint to which the User Story is assigned. (Nullable)

Relationships:

`User Stories` in the `ProductBacklog` represent features or requirements of the project. They are assigned to `Sprints` for implementation.

14. TaskPB Table

Description:

Similar to `TaskSB`, but for tasks in the product backlog. It includes task details like title, description, status, and which sprint the task is assigned to.

Columns:

	Name	Data type	Description
🔑	idTaskPB	INT	Unique identifier for a Task in the Product Backlog.
	Title	VARCHAR(45)	Title of the Task.
	Description	VARCHAR(255)	Description of the Task. (Nullable)
	Status	VARCHAR(45)	Status of the Task.
	Assignee	VARCHAR(40)	User ID of the person assigned to the Task. (Nullable)
	Priority	INT	Priority of the Task. (Nullable)
	ProductBacklog_idProductBacklog	INT	Foreign key referencing ProductBacklog.
	ProductBacklog_Project_idProject	INT	Foreign key referencing Project within ProductBacklog.
	AssignedSprint	VARCHAR(45)	Sprint to which the Task is assigned. (Nullable)

Relationships:

`Tasks` in the `ProductBacklog` are specific work items related to the project's features or requirements. They are planned for implementation in specific `Sprints`.

15. UserStorySB Table

Description:

Manages user stories within the sprint backlog. These are tasks or features to be completed during a specific sprint, linked to the larger sprint backlog and the overall project.

Columns:

	Name	Data type	Description
🔑	idUserStorySB	INT	Unique identifier for a User Story in the Sprint Backlog.
	Title	VARCHAR(45)	Title of the User Story.
	Description	VARCHAR(45)	Description of the User Story. (Nullable)
	Status	VARCHAR(45)	Status of the User Story.
	Assignee	INT	User ID of the person assigned to the User Story. (Nullable)
	Priority	INT	Priority of the User Story. (Nullable)
	SprintBacklog_idSprintBacklog	INT	Foreign key referencing SprintBacklog.
	SprintBacklog_Sprint_idSprint	INT	Foreign key referencing Sprint within SprintBacklog.
	SprintBacklog_Sprint_Project_idProject	INT	Foreign key referencing Project within Sprint.

Relationships:

Represents `User Stories` within a `SprintBacklog`, detailing the features or requirements to be addressed during the sprint.

16. SubtaskPB Table

Description:

Handles subtasks within tasks of the product backlog. This table allows for granular tracking and management of smaller task components in the broader scope of the project's product backlog.

Columns:

	Name	Data type	Description
	idSubtaskPB	INT	Unique identifier for a Subtask in the Product Backlog.
	Title	VARCHAR(45)	Title of the Subtask.
	Description	VARCHAR(255)	Description of the Subtask. (Nullable)
	Status	VARCHAR(45)	Status of the Subtask.
	Assignee	INT	User ID of the person assigned to the Subtask. (Nullable)
	Priority	INT	Priority of the Subtask. (Nullable)
	TaskPB_idTaskPB	INT	Foreign key referencing TaskPB.
	TaskPB_ProductBacklog_idProductBacklog	INT	Foreign key referencing ProductBacklog within TaskPB.
	TaskPB_ProductBacklog_Project_idProject	INT	Foreign key referencing Project within ProductBacklog.

Relationships:

Represents smaller tasks (`Subtasks`) within a `TaskPB` in the `ProductBacklog`, further breaking down the work required for project features.

17. Calendar

Description:

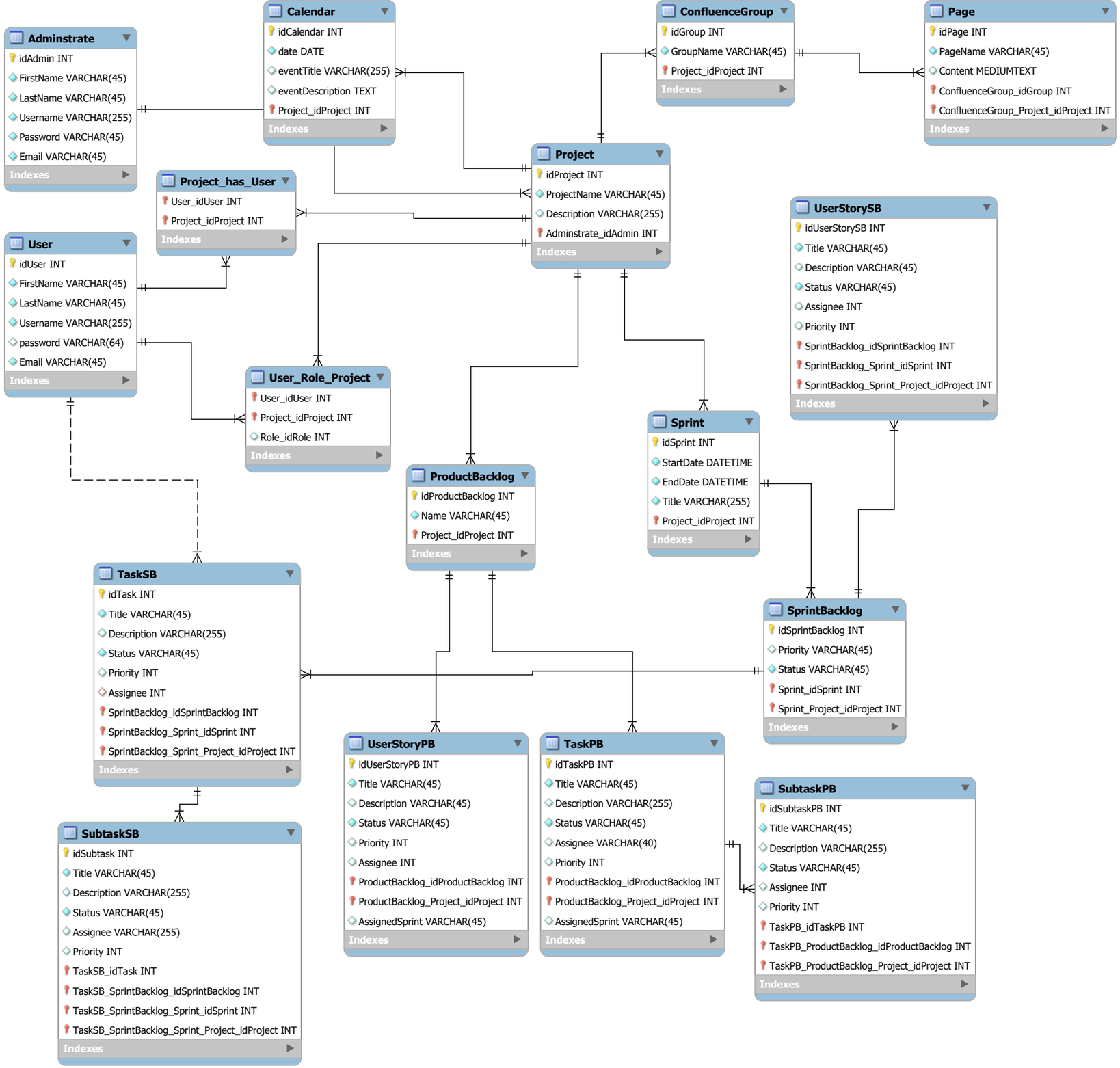
This table designed to store calendar events related to projects, with each event having a title, description, and associated date.

Columns:

	Name	Data type	Description
🔑	idCalendar	INT	Unique identifier for a Calender.
	date	DATE	Start date of the event.
	eventTitle	VARCHAR(255)	Storing the title of an event.
	eventDescription	TEXT	Detailed descriptions of events.
	Project_idProject	INT	Foreign key referencing Project.

Relationships:

The presence of the "Project_idProject" column indicates a relationship with a "Project" table, signifying that each event is linked to a specific project.


Adminstrate
idAdmin INT
FirstName VARCHAR(45)
LastName VARCHAR(45)
Username VARCHAR(255)
Password VARCHAR(45)
Email VARCHAR(45)
Indexes
Calendar
idCalendar INT
date DATE
eventTitle VARCHAR(255)
eventDescription TEXT
Project_idProject INT
Indexes
ConfluenceGroup
idGroup INT
GroupName VARCHAR(45)
Project_idProject INT
Indexes
Page
idPage INT
PageName VARCHAR(45)
Content MEDIUMTEXT
ConfluenceGroup_idGroup INT
ConfluenceGroup_Project_idProject INT
Indexes
Project_has_User
User_idUser INT
Project_idProject INT
Indexes
Project
idProject INT
ProjectName VARCHAR(45)
Description VARCHAR(255)
Adminstrate_idAdmin INT
Indexes
User
idUser INT
FirstName VARCHAR(45)
LastName VARCHAR(45)
Username VARCHAR(255)
password VARCHAR(64)
Email VARCHAR(45)
Indexes
UserStorySB
idUserStorySB INT
Title VARCHAR(45)
Description VARCHAR(45)
Status VARCHAR(45)
Assignee INT
Priority INT
SprintBacklog_idSprintBacklog INT
SprintBacklog_Sprint_idSprint INT
SprintBacklog_Sprint_Project_idProject INT
Indexes
User_Role_Project
User_idUser INT
Project_idProject INT
Role_idRole INT
Indexes
Sprint
idSprint INT
StartDate DATETIME
EndDate DATETIME
Title VARCHAR(255)
Project_idProject INT
Indexes
ProductBacklog
idProductBacklog INT
Name VARCHAR(45)
Project_idProject INT
Indexes
TaskSB
idTask INT
Title VARCHAR(45)
Description VARCHAR(255)
Status VARCHAR(45)
Priority INT
Assignee INT
SprintBacklog_idSprintBacklog INT
SprintBacklog_Sprint_idSprint INT
SprintBacklog_Sprint_Project_idProject INT
Indexes
SprintBacklog
idSprintBacklog INT
Priority VARCHAR(45)
Status VARCHAR(45)
Sprint_idSprint INT
Sprint_Project_idProject INT
Indexes
UserStoryPB
idUserStoryPB INT
Title VARCHAR(45)
Description VARCHAR(45)
Status VARCHAR(45)
Priority INT
Assignee INT
ProductBacklog_idProductBacklog INT
ProductBacklog_Project_idProject INT
AssignedSprint VARCHAR(45)
Indexes
TaskPB
idTaskPB INT
Title VARCHAR(45)
Description VARCHAR(255)
Status VARCHAR(45)
Assignee VARCHAR(40)
Priority INT
ProductBacklog_idProductBacklog INT
ProductBacklog_Project_idProject INT
AssignedSprint VARCHAR(45)
Indexes
SubtaskPB
idSubtaskPB INT
Title VARCHAR(45)
Description VARCHAR(255)
Status VARCHAR(45)
Assignee INT
Priority INT
TaskPB_idTaskPB INT
TaskPB_ProductBacklog_idProductBacklog INT
TaskPB_ProductBacklog_Project_idProject INT
Indexes
SubtaskSB
idSubtask INT
Title VARCHAR(45)
Description VARCHAR(255)
Status VARCHAR(45)
Assignee VARCHAR(255)
Priority INT
TaskSB_idTask INT
TaskSB_SprintBacklog_idSprintBacklog INT
TaskSB_SprintBacklog_Sprint_idSprint INT
TaskSB_SprintBacklog_Sprint_Project_idProject INT
Indexes