< DATE 17/03/17>

Proof Reading Website

Deliverable 2

Week 8

Group 18

Padraig Punch (8487103)

Tresor Twendimbadi (16136616)

Alam Syed Hasnain (15523108)

Stephen Hughes (1544444)

Contents

1. Introduction..2

2. Database Tables...3

3. Entity Relationship Diagram…………………………………………………………………………………………………8

Introduction

This report details the database schema as created by Group 18 for the CS4056 Web Infrastructures project. The document consists of a single section, containing;

- **Database Table Synopsis** – a simple listing of the attributes and datatypes for every table in the database. In some cases this is followed by a short note explaining the logic behind the grouping of certain attributes in the same table.
- **Table SQL Creation Declarations** – the SQL statements used to create each table on the group's MySQL database, including information on Foreign Key constraints, use of auto-increment, and default attribute values.
- **Entity Relationship diagram –** An entity relationship diagram of the tables.
- **Link to Git Repository file** - https://github.com/CS4014group18/web filename is web7.sql The sql file was generated using MySQL Workbench constraint names were deleted to remove problems with duplicate names.

Database Tables

This section displays all database tables, attributes and datatypes, with the SQL create table statements included to show foreign keys, default values and auto-increments.

User

ID	FirstName	LastName	Email	Password	Reputation
VARCHAR(8)	VARCHAR(30)	VARCHAR(30)	VARCHAR(50)	VARCHAR(50)	INT

```
CREATE TABLE IF NOT EXISTS `group18`.`User` (
  `ID` VARCHAR(8) NOT NULL,
  `FirstName` VARCHAR(30) NOT NULL,
  `LastName` VARCHAR(30) NOT NULL,
  `Email` VARCHAR(50) NOT NULL,
  `Password` VARCHAR(50) NOT NULL,
  `Reputation` INT ZEROFILL NULL,
  PRIMARY KEY (`ID`),
  UNIQUE INDEX `Email_UNIQUE` (`Email` ASC));
```

NOTE Email is declared as unique and Reputation is initialised to zero.

Major

ID	Major
VARCHAR(8)	VARCHAR(50)

```
CREATE TABLE IF NOT EXISTS `group18`.`Major` (
  `ID` VARCHAR(8) NOT NULL,
  `Major` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`ID`),
  CONSTRAINT
    FOREIGN KEY (`ID`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

Banned

ID	Moderator	Date	Reason
VARCHAR(8)	VARCHAR(8)	DATETIME	VARCHAR(50)

```
CREATE TABLE IF NOT EXISTS `group18`.`Banned` (
  `ID` VARCHAR(8) NOT NULL,
  `Moderator` VARCHAR(8) NOT NULL,
  `Date` DATETIME NOT NULL,
  `Reason` VARCHAR(50) NULL,
  PRIMARY KEY (`ID`),
  INDEX `moderator_idx` (`Moderator` ASC),
  CONSTRAINT
    FOREIGN KEY (`ID`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT
    FOREIGN KEY (`Moderator`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

Task

idTaskNo	UserCreated	Title	Type	Description	Pages	Words	Format	Sample	DeadlineClaiming	DeadlineSubmission
INT	VARCHAR(8)	VARCHAR(100)	VARCHAR(50)	VARCHAR(200)	INT	INT	VARCHAR(10)	TEXT	DATETIME	DATETIME

```
CREATE TABLE IF NOT EXISTS `group18`.`Task` (
  `idTaskNo` INT NOT NULL,
  `UserCreated` VARCHAR(8) NOT NULL,
  `Title` VARCHAR(100) NOT NULL,
  `Type` VARCHAR(50) NOT NULL,
  `Description` VARCHAR(200) NOT NULL,
  `Pages` INT NOT NULL,
  `Words` INT NOT NULL,
  `Format` VARCHAR(10) NOT NULL,
  `Sample` TEXT NULL,
  `DeadlineClaiming` DATETIME NOT NULL,
  `DeadlineSubmission` DATETIME NOT NULL,
  PRIMARY KEY (`idTaskNo`),
  INDEX `UserCreated_idx` (`UserCreated` ASC),
  CONSTRAINT
    FOREIGN KEY (`UserCreated`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

Claimed

idClaimed	ID	TaskNo	date
INT	VARCHAR(8)	INT	DATETIME

```
CREATE TABLE IF NOT EXISTS `group18`.`Claimed` (
  `idClaimed` INT NOT NULL,
  `ID` VARCHAR(8) NOT NULL,
  `TaskNo` INT NOT NULL,
  `date` DATETIME NOT NULL,
  PRIMARY KEY (`idClaimed`),
  INDEX `ID_idx` (`ID` ASC),
  INDEX `TaskNo_idx` (`TaskNo` ASC),
  CONSTRAINT
    FOREIGN KEY (`ID`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT
    FOREIGN KEY (`TaskNo`)
    REFERENCES `group18`.`Task` (`idTaskNo`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

Tags

idTags	Description
INT	VARCHAR(30)

```
CREATE TABLE IF NOT EXISTS `group18`.`Tags` (
  `idTags` INT NOT NULL,
  `Description` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`idTags`));
```

UserTags

idUserTags	ID	Tag
INT	VARCHAR(8)	INT

```
CREATE TABLE IF NOT EXISTS `group18`.`UserTags` (
  `idUserTags` INT NOT NULL,
  `ID` VARCHAR(8) NOT NULL,
  `Tag` INT NOT NULL,
  PRIMARY KEY (`idUserTags`),
  INDEX `ID_idx` (`ID` ASC),
  INDEX `idTags_idx` (`Tag` ASC),
  CONSTRAINT
    FOREIGN KEY (`ID`)
    REFERENCES `group18`.`User` (`ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT
    FOREIGN KEY (`Tag`)
    REFERENCES `group18`.`Tags` (`idTags`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

TaskTags

idTaskTags	TaskNo	Tag
INT	INT	INT

```
CREATE TABLE IF NOT EXISTS `group18`.`TaskTags` (
  `idTaskTags` INT NOT NULL,
  `TaskNo` INT NOT NULL,
  `Tag` INT NOT NULL,
  PRIMARY KEY (`idTaskTags`),
  INDEX `taskNo_idx` (`TaskNo` ASC),
  INDEX `idTags_idx` (`Tag` ASC),
  CONSTRAINT
    FOREIGN KEY (`Tag`)
    REFERENCES `group18`.`Tags` (`idTags`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT
    FOREIGN KEY (`TaskNo`)
    REFERENCES `group18`.`Task` (`idTaskNo`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

StatusName

idStatusName	Status
INT	VARCHAR(13)

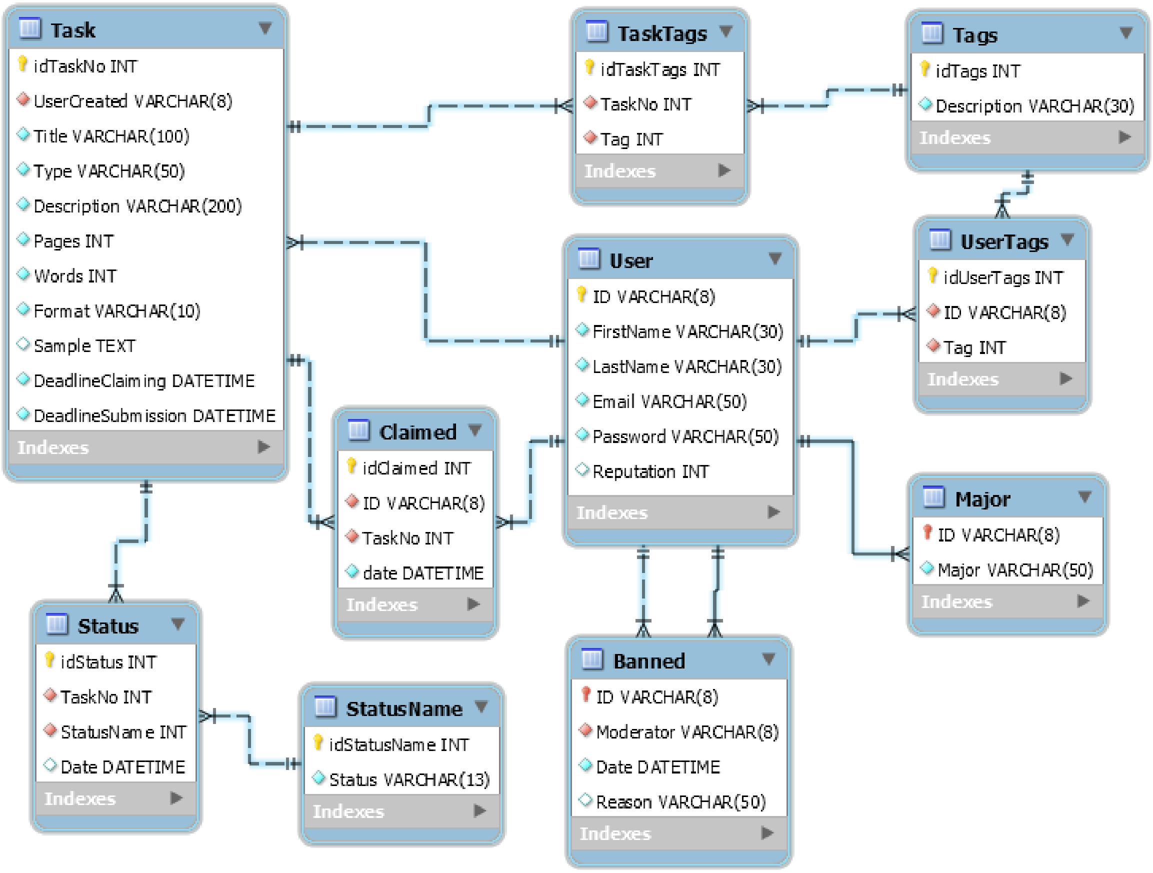
```
CREATE TABLE IF NOT EXISTS `group18`.`StatusName` (
  `idStatusName` INT NOT NULL,
  `Status` VARCHAR(13) NOT NULL,
  PRIMARY KEY (`idStatusName`));
```

Status

idStatus	TaskNo	StatusName	Date
INT	INT	INT	DATETIME

```
CREATE TABLE IF NOT EXISTS `group18`.`Status` (
  `idStatus` INT NOT NULL,
  `TaskNo` INT NOT NULL,
  `StatusName` INT NOT NULL,
  `Date` DATETIME NULL,
  PRIMARY KEY (`idStatus`),
  INDEX `idStatus_idx` (`StatusName` ASC),
  INDEX `TaskNo_idx` (`TaskNo` ASC),
  CONSTRAINT
    FOREIGN KEY (`StatusName`)
    REFERENCES `group18`.`StatusName` (`idStatusName`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT
    FOREIGN KEY (`TaskNo`)
    REFERENCES `group18`.`Task` (`idTaskNo`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

Entity Relationship Diagram


Task
idTaskNo INT
UserCreated VARCHAR(8)
Title VARCHAR(100)
Type VARCHAR(50)
Description VARCHAR(200)
Pages INT
Words INT
Format VARCHAR(10)
Sample TEXT
DeadlineClaiming DATETIME
DeadlineSubmission DATETIME
Indexes
TaskTags
idTaskTags INT
TaskNo INT
Tag INT
Indexes
Tags
idTags INT
Description VARCHAR(30)
Indexes
UserTags
idUserTags INT
ID VARCHAR(8)
Tag INT
Indexes
User
ID VARCHAR(8)
FirstName VARCHAR(30)
LastName VARCHAR(30)
Email VARCHAR(50)
Password VARCHAR(50)
Reputation INT
Indexes
Claimed
idClaimed INT
ID VARCHAR(8)
TaskNo INT
date DATETIME
Indexes
Major
ID VARCHAR(8)
Major VARCHAR(50)
Indexes
Status
idStatus INT
TaskNo INT
StatusName INT
Date DATETIME
Indexes
StatusName
idStatusName INT
Status VARCHAR(13)
Indexes
Banned
ID VARCHAR(8)
Moderator VARCHAR(8)
Date DATETIME
Reason VARCHAR(50)
Indexes