

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **XYZ Software Solutions**  23/12/2018 | |  | | |
|  |  |  | | |  | |
|  |  | Trinity Database  Documentation | | |  | |
|  |  | | | | |  |
|  | **Introduction**  Trinity Music School based in Colombo currently practices disorganized and out of date, data processing methods in order to meet the demands in the growth of the school and flexibility, the school has request XYZ Software Solutions to create a relational database system to completely replace their school based record management system.  This document will contain the user and technical documentation. Topics discussed in the user documentation will be given more clarity in the technical documentation.  Author: K.P.I. Shenesh Perera  IDM | | | | |  |
|  |  | |  | | | |

**Contents**

[**User Documentation** 3](#_Toc1654104)

[Basic Guidance 3](#_Toc1654105)

[Types of users 4](#_Toc1654106)

[Registration Process 4](#_Toc1654107)

[What is valid data? 5](#_Toc1654108)

[The underlying processes behind the database solution 6](#_Toc1654109)

[Consumer Process 6](#_Toc1654110)

[Admin process 7](#_Toc1654111)

[What is Data Corruption? 7](#_Toc1654112)

[Making backups 7](#_Toc1654113)

[Restoring backups 8](#_Toc1654114)

[Video demonstrating how to backup and restore databases 8](#_Toc1654115)

[8](#_Toc1654116)

[Emergency situations 9](#_Toc1654117)

[**Technical Documentation** 11](#_Toc1654118)

[Data Dictionary 11](#_Toc1654119)

[Backup schedule 12](#_Toc1654120)

[Data Corruption In-Depth 13](#_Toc1654121)

[ SQL Error 5172 13](#_Toc1654122)

[ SQL Fatal 823 Error 13](#_Toc1654123)

[ 8946 SQL Server 14](#_Toc1654124)

[System Requirements 14](#_Toc1654125)

[Further improvements for the database solution 14](#_Toc1654126)

# **User Documentation**

This part of the documentation will describe most processes that exist within the system in comprehensible terms without giving into technical jargon. All common users that have a lack in the knowledge of IT, or have had no exposure to application software should read this section of the documentation before using the solution.

## Basic Guidance

In order to extract the best of this database solution, one must have a clear understanding of how this system works in order to troubleshoot if the need arises. Since it is not possible to cover every error that may occur while using this system, it is highly recommended to refer to [Microsoft SQL Server's Documentation](https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017) for further assistance in case one comes across a difficulty during usage. In order to assist breaking down the technical terms used in MS-SQL server’s docs, use Google.

## Types of users

Users have been categorized to 3 groups, depending on the role they have to perform in order to extract the best of the system. Namely, Admins, Consumers and Supporters.

Admin users have read/write permissions, can change passwords of consumers and other admins, can create consumer accounts, can backup & restore the database and can completely erase all data within the database. Admin users are suggested to not frequently operate the system. Admins can have consumer accounts to participate in read/write actions instead. All IT staff members are admins.

Consumer users have read/write permissions for all forms across the system. These users are the ones that will actively and frequently operate the system by interacting with students, suppliers and all other external audiences. All front-office staff members are consumers.

Support users do not interact with the system at all. They exist in order to ensure that the system functions well and to serve the purpose of reporting management level or the IT staff when an error of sufficient magnitude has occurred. As such these users will not be visualized in most diagrams, as they play an indirect role in the functioning of the system.

## Registration Process

During the registration process, students, teachers or suppliers will be entered to the trinity database student table through a consumer user.

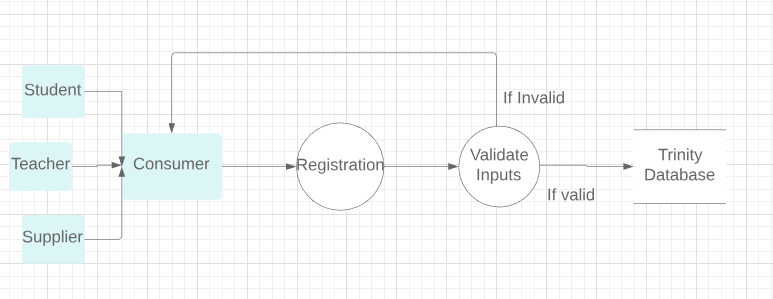


Figure 1.0, Shenesh Perera, 29/12/2018

During the registration, a consumer will receive data from a student, teacher or consumer and begin entering the received information to an entry form which is symbolized as “Registration” in the diagram to initialize the registration process, then only if the data that has been provided are accurate, the data will be entered to the trinity database, if it is invalid the database will not accept the data.

## What is valid data?

For each particular unit of data that is entered to the database, there exists something known as a “data type”. A data type is the classification of the data, for example a dog is a mammal, a human is a mammal similarly an Identification Number (ID) will be of the Integer data type, an integer is a whole number with no decimal point in it. Refer to the data dictionary in the technical documentation to know what data type a particular data is.

## The underlying processes behind the database solution

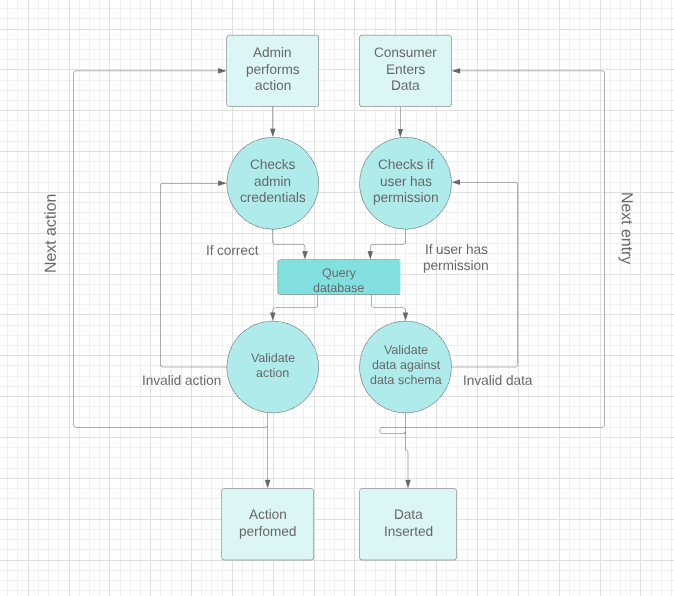


Figure 1.1, Shenesh Perera 29/12/2018

### Consumer Process

A consumer only has read/write permissions to the database. If consumer user attempts to do something that is not reading or writing to the database, the database will not allow that consumer to enter that data to the database. Then after permission checking is done, the database is “queried” or in simpler terms the database is questioned to see if the datatypes of the entered data matches with the table’s data schema. A data schema is what holds the datatypes of each data in a table. If they match, then the data is inserted and the next entry can be performed, if they don’t then the question or “query” is rejected.

### Admin process

An admin is capable of more than reading or writing. They are capable of deleting data, making backups, restoring them etc. so in order to do this solution must see if the admin’s username and password are correct, and if they are then query the database then see if that action can be performed on the database. If that action can be performed then it will be performed, and then the next action can be performed or if it can’t be performed then the action is rejected.

## What is Data Corruption?

SQL Server is a platform used for business stability. However, just like any other database, it's prone to corruption. It is the damaging or destruction of data due to some sort of external factor.

There are 3 most common errors that occur as a result of data corruption:

1. SQL Error 5172
2. SQL Fatal 823 Error
3. 8946 SQL Server

The only solution in such a situation will be to restore a backup that has been previously stored, as such making backups is important. Refer to [Data Corruption In-Depth](#_Data_Corruption_In-Depth) for further details about these errors and how to fix them.

## Making backups

Following are the steps that must be taken in order to make a full backup of the database trinityHS.

* In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
* Expand Databases, right-click trinityHS, point to Tasks, and then click Back Up.
* On the General page in the Destination section select Disk from the Back up to: drop-down list.
* Click Remove until all existing backup files have been removed.
* Click Add and the Select Backup Destination dialog box will open.
* Navigate to the location you want to store the backup in, then add .bak to the file name in the text box.
* Click OK.
* Click OK.

Refer to [Backup Schedule](#_Backup_schedule) to know when and how often you should make backups.

## Restoring backups

Following are the steps that must be taken in order to restore a full backup of the database trinityHS.

* In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
* Right-click Databases and select Restore Database.
* On the General page, select Device under the Source section.
* Click the browse button to open the Select backup devices dialog box. Click Add and navigate to your backup. Click OK after you have selected your disk backup file.
* Click OK to return to the General page.
* Click Options in the Select a page pane.
* Under the Restore options section, check Overwrite the existing database (WITH REPLACE).
* Click OK

## Video demonstrating how to backup and restore databases

## 

Click [here](https://youtu.be/1Dwni-aLeII) if the video doesn’t work.

## Emergency situations

If for some reason the database has been lost, there are no backups or restoring procedures don’t work, then follow these steps to run the following set of queries:

* After opening SQL Server Management Studio, navigate to the 2nd menu-bar and click New Query.
* Copy-paste the below stated set of queries in to it.
* Highlight the 1st line and click Execute on the 3rd menu bar.
* Then highlight the rest and click execute.

After following this process, a fresh and empty database called trinityHS will be created with all the tables. Simply the database solution will be recreated.

CREATE DATABASE trinityHS

USE trinityHS

CREATE TABLE class (

c\_id int PRIMARY KEY,

c\_start datetime,

c\_end datetime,

c\_type varchar(50)

)

CREATE TABLE supplier (

supp\_id int IDENTITY(100, 20) PRIMARY KEY,

supp\_address varchar(80),

supp\_name varchar(100),

supp\_contact\_no varchar(20)

)

CREATE TABLE instrument (

i\_id int PRIMARY KEY,

supp\_id int FOREIGN KEY REFERENCES supplier(supp\_id),

i\_name varchar(100),

i\_quantity int,

i\_rental money

)

CREATE TABLE student (

s\_id int PRIMARY KEY,

i\_id int FOREIGN KEY REFERENCES instrument(i\_id),

s\_name varchar(100),

s\_type varchar(50),

s\_regist\_date datetime,

s\_dob datetime,

s\_contact\_no varchar(20)

)

CREATE TABLE teacher (

t\_id int PRIMARY KEY,

t\_name varchar(100),

t\_contact\_no varchar(20)

)

CREATE TABLE attendance (

a\_index int IDENTITY(10, 5) PRIMARY KEY,

c\_id int FOREIGN KEY REFERENCES class(c\_id),

s\_id int FOREIGN KEY REFERENCES student(s\_id),

attend\_time datetime NOT NULL,

daily\_cost money,

)

CREATE TABLE class\_subjects\_linked (

cs\_index int IDENTITY(10, 5) PRIMARY KEY,

c\_id int FOREIGN KEY REFERENCES class(c\_id),

t\_id int FOREIGN KEY REFERENCES teacher(t\_id),

c\_subject varchar(50)

)

# **Technical Documentation**

## Data Dictionary

The data dictionary consists of all the required details, including data types, examples and where each field exists in the database. During troubleshooting the data dictionary will prove to be very useful as such please refer to the following thoroughly before using the database solution.

Data Dictionary Key

**Field Name**: The name of the column within a table.

**Table name**: The name of the table.

**Data Type**: The type of the data listed on the data schema of the table.

**Data format**: The general syntax of a field name.

**Field Size**: The maximum character limit for a field name.

**Example**: A possible data that satisfies the data schema and other constraints.

*The “\*” symbols indicate the primary keys while “\*\*” indicate foreign keys*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Table Name | Data Type | Data Format | Field Size | Example |
| c\_id(\*) | Class | int |  |  | 1111 |
| c\_start | Class | datetime | YYYY-MM-DD 00:00:00 AM/PM |  | 20181018 08:00:00 AM |
| c\_end | Class | datetime | YYYY-MM-DD 00:00:00 AM/PM |  | 20181018 08:00:00 AM |
| c\_type | Class | varchar(50) | FULL\_TIME, PART\_TIME, PART\_TIME\_EVENING | 50 | FULL\_TIME |
| supp\_id(\*) | Supplier | int |  |  | 100 |
| supp\_address | Supplier | varchar(80) |  | 80 | St Joseph's St, Negombo |
| supp\_name | Supplier | varchar(100) |  | 100 | Yamaha Dealer |
| supp\_contact\_no | Supplier | varchar(20) |  | 20 | 031892834 |
| i\_id(\*) | Instrument | int |  |  | 103 |
| supp\_id(\*\*) | Instrument | int |  |  | 100 |
| i\_name | Instrument | varchar(100) |  | 100 | Violin |
| i\_quantity | Instrument | int |  |  | 56 |
| i\_rental | Instrument | money | VALUE.00 |  | 20398.00 |
| s\_id(\*) | Student | int |  |  | 1 |
| i\_id(\*\*) | Student | int |  |  | 103 |
| s\_name | Student | varchar(100) |  | 100 | Saline Man |
| s\_type | Student | varchar(50) | FULL\_TIME, PART\_TIME, PART\_TIME\_EVENING | 50 |  |
| s\_regist\_date | Student | datetime | YYYY-MM-DD 00:00:00 AM/PM |  | 20181018 08:00:00 AM |
| s\_dob | Student | datetime | YYYY-MM-DD 00:00:00 AM/PM |  | 20181018 08:00:00 AM |
| s\_contact\_no | Student | varchar(20) |  | 20 | FULL\_TIME |
| t\_id(\*) | Teacher | int |  |  | 10001 |
| t\_name | Teacher | varchar(100) |  | 100 | Dumbledore McField |
| t\_contact\_no | Teacher | varchar(20) |  | 20 | 079 123467 |
| a\_index(\*) | Attendance | int |  |  | 15 |
| c\_id(\*\*) | Attendance | int |  |  | 1111 |
| s\_id(\*\*) | Attendance | int |  |  | 1 |
| attend\_time | Attendance | datetime | YYYY-MM-DD 00:00:00 AM/PM |  | 20181018 08:00:00 AM |
| daily\_cost | Attendance | money | VALUE.00 |  | 2000.00 |
| cs\_index(\*) | class\_subjects\_linked | int |  |  | 15 |
| c\_id(\*\*) | class\_subjects\_linked | int |  |  | 1111 |
| t\_id(\*\*) | class\_subjects\_linked | int |  |  | 10001 |
| c\_subject | class\_subjects\_linked | varchar(50) |  | 50 | Cleaning Instruments |

## Backup schedule

Admins are charged with the responsibility to back up the database trinityHS following the steps mentioned in the user documentation obeying the following conditions respectively:

1. By default, a backup of the database must be performed every once in a week, and then stored in an external device.
2. Backup and compress once in every year, then perform a flush on the database on the 1st of January 00:00 every year.
3. Backup immediately if:

* The data schema of any of the tables are edited in any way.
* A table is dropped.
* A table is added.
* Data cleaning procedures have been performed.
* Further data normalization has been performed.
* A significantly sensitive or valued set of data has been inserted to one of the tables.

## Data Corruption In-Depth

### SQL Error 5172

The SQL Server saves its physical database in a primary file that has data as per the pages. The first page holds the information of the MDF file header, which is called a header page.

It consists of various information about the database such as the size of the file, signature, etc. During the process of attaching the MDF in SQL Server, a frequent error encountered by users is error 5172.

This generally occurs when the MDF file becomes unhealthy or damaged. Once this error occurs, the information of the header file has already been mismatched, making it difficult to access the data.

This error usually occurs due to:

1. Bad shutdown of the server system.
2. Attacks from a malicious virus.
3. Inappropriate shutdown of SQL Server.
4. Malfunctioning of server hardware.

Recovering from this error can only be done through the expertise of a Database Administrator or a System Admin that has expertise in the data sector. If professionals capable of this resolving this error cannot be found, restore a backup on a clean install of Microsoft SQL server as documented [here](http://www.mustbegeek.com/install-sql-server-2012-windows-server-2012-r2/).

### SQL Fatal 823 Error

SQL users use Windows APIs for the execution of I/O operations. Upon the completion of I/O operations, SQL verifies for any error associated with the API calls. If these API calls are incompatible with the Operating System, error 823 occurs in SQL Server.

This error message consists of the following information:

* Whether the I/O operation is a write or read request
* The offset within the file where the I/O operation was tried
* The file against which the I/O operation was executed
* The error code of the Operating System and error description

The error 823 message signifies that there is an issue with the underlying storage machine hardware or a driver that is in the path of I/O request.

Users may face this error when:

* There are contradictions in the file system.
* If the database file is corrupted.

The only way to recover from this type of error is to move to a compatible MS Windows installed machine, fitting the [System Requirements](#_System_Requirements) for the solution to run.

### 8946 SQL Server

The main reason behind error 8946 is corruption in the SQL database. Once the error arises, it makes it tough to access the files.

Moreover, it can happen when there is not a valid page header to a specific assigned page. There are various reasons for the error. Some of these are power failure, virus infection, sudden shutdown of the machine, and hardware failure.

Once the headers are corrupted or damaged, the entire data is damaged, possibly resulting in data loss. In order to avoid this, it is important to fix this issue as soon as possible. In order to fix this, contact a database administrator or developer.

## System Requirements

The database solution has been developed and run in a system matching the following specifications and will also definitely run in a system similar to the following specs:

**Processor**: Intel® Core™ i3-4000M CPU @ 2.4 GHz

**RAM**: 6.00 GB

**System type**: 64-bit

**Operating System**: Windows 8.1

**Microsoft SQL Server**: MSSQL v11.0 RTM

## Further improvements for the database solution

If at any point in time Trinity Music School would perform any improvements on the existing installation, show the following list to the company that have taken up the contract:

1. Deploy the database to a centralized cloud database server, implement SSH in order to connect to the server. You may use PuTTy if personal implementation is beyond the scope.
2. Enforce encrypted bit streams between users and host systems using a standard encryption algorithm like American Encryption Standard, on the 256-bit variant in GCM mode.
3. Provide instructions to Trinity Music School to perform encrypted backups or produce a system that will automate the backup process but make encrypted backups instead.
4. Move away from a SQL RDBMS like MSSQL and use a NoSQL installation like MongoDB as the data schema and scenario are yet to immature for SQL and therefore reduces flexibility during data storage.
5. Use auto-scheduling and jobs in SSMS to automate backups and perform data cleaning upon authorization.
6. Introduce an MDF viewer to trinityHS so that they can view their tables without running SQL commands.
7. If the database has been deployed to a centralized cloud, then proceed to develop a RESTful API to interact with the database over a load balancing layer like NGINX and an encryption monitor.
8. Introduce Trinity Music School to SAP Crystal Reports in order to visualize data better for easier management decisions.