**Grandfield College Scenario Module 6 Assignment**

The management is afraid of a software audit. The chief systems manager just came from a meeting where he heard that a school had just been fined $25,000 for illegally installed software.

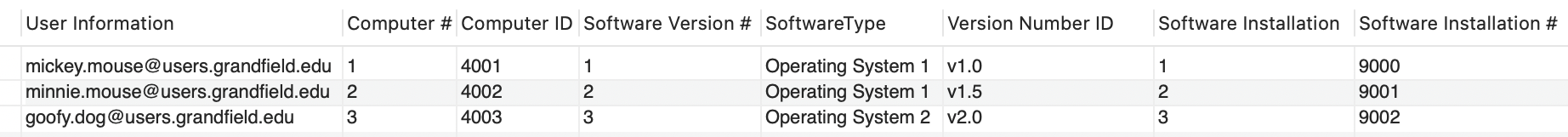
The current tracking system probably couldn’t hold up to an audit. It is crucial that this new database be up and running soon. You assure the management that it will be done as soon as is possible, but you want to make sure that it really does what it is supposed to do. If you implement before it is ready, it might make matters worse rather than better.

1. Review your diagram for the database, making sure that the design is complete and normalized.

2. Create the database in SQL Server.

**-- Create the Grandfield College Database**

CREATE DATABASE Grandfield**\_**College**\_**Database;



3. Create the tables in the new database, selecting appropriate data types for the columns, setting a primary key for each table, and setting allow nulls as appropriate.

**-- Create an 'Administration' table**

CREATE TABLE Administration (

Administration\_Key INT PRIMARY KEY,

Administration\_ID INT NOT NULL,

Contact\_Information VARCHAR(255) NOT NULL );

**-- Create a 'Faculty Staff' table**

CREATE TABLE Faculty\_Staff (

Faculty\_Staff\_Key INT PRIMARY KEY,

Faculty\_Staff\_ID INT NOT NULL,

Contact\_Information VARCHAR(255) NOT NULL );

**-- Create 'IT Department' table**

CREATE TABLE IT\_Department (

IT\_Department\_Key INT PRIMARY KEY,

IT\_Department\_ID INT NOT NULL,

Contact\_Information VARCHAR(255) NOT NULL );

**-- Create a 'Departments' table**

CREATE TABLE Departments (

Departments\_Key INT PRIMARY KEY,

Allowance\_Restriction VARCHAR(255) NOT NULL );

**-- Create a 'Computers' table**

CREATE TABLE Computers (

Computers\_Key INT PRIMARY KEY,

Computers\_ID INT NOT NULL );

**-- Create a 'Computer Users' table**

CREATE TABLE Computer\_Users (

Computer\_Users\_Key INT PRIMARY KEY,

Computers\_Key INT NOT NULL,

User\_Information VARCHAR(255) NOT NULL,

FOREIGN KEY (Computers\_Key)

REFERENCES Computers(Computers\_Key) );

**-- Create a 'Software' table**

CREATE TABLE Software (

Software\_Key INT PRIMARY KEY,

Software\_Type VARCHAR(255) NOT NULL );

**-- Create a 'Software Version' table**

CREATE TABLE Software\_Version (

Software\_Version\_Key INT PRIMARY KEY,

Software\_Key INT NOT NULL,

Version\_Number\_ID VARCHAR(50) NOT NULL,

FOREIGN KEY (Software\_Key)

REFERENCES Software(Software\_Key) );

**-- Create a 'Software Installations' table**

CREATE TABLE Software\_Installations (

Software\_Installation\_Key INT PRIMARY KEY,

Software\_Version\_Key INT NOT NULL,

Computers\_Key INT NOT NULL,

Software\_Installation\_Number INT NOT NULL,

FOREIGN KEY (Software\_Version\_Key)

REFERENCES Software\_Version(Software\_Version\_Key),

FOREIGN KEY (Computers\_Key)

REFERENCES Computers(Computers\_Key) );

4. Add some sample data to each table.

**-- Insert sample data into 'Administration' table**

INSERT INTO Administration (Administration\_Key, Administration\_ID, Contact\_Information)

VALUES (1, 1001, 'admin1@administration.grandfield.edu'),

(2, 1002, 'admin2@administration.grandfield.edu');

**-- Insert sample data into 'Faculty Staff' table**

INSERT INTO Faculty\_Staff (Faculty\_Staff\_Key, Faculty\_Staff\_ID, Contact\_Information)

VALUES (1, 2001, 'staff1@faculty.grandfield.edu'),

(2, 2002, 'staff2@faculty.grandfield.edu'),

(3, 2003, 'staff3@faculty.grandfield.edu'),

(4, 2004, 'staff4@faculty.grandfield.edu');

**-- Insert sample data into IT\_Department table**

INSERT INTO IT\_Department (IT\_Department\_Key, IT\_Department\_ID, Contact\_Information)

VALUES (1, 3001, 'it1@itdept.grandfield.edu'),

(2, 3002, 'it2@itdept.grandfield.edu');

**-- Insert sample data into Departments table**

INSERT INTO Departments (Departments\_Key, Allowance\_Restriction)

VALUES (1, 'Allowed Access'),

(2, 'Restricted Access');

**-- Insert sample data into Software table**

INSERT INTO Software (Software\_Key, Software\_Type)

VALUES (1, 'Operating System 1'),

(2, 'Operating System 2');

**-- Insert sample data into Software\_Version table**

INSERT INTO Software\_Version (Software\_Version\_Key, Software\_Key, Version\_Number\_ID)

VALUES (1, 1, 'v1.0'),

(2, 1, 'v1.5'),

(3, 2, 'v2.0');

**-- Insert sample data into Computers table**

INSERT INTO Computers (Computers\_Key, Computers\_ID)

VALUES (1, 4001),

(2, 4002),

(3, 4003);

**-- Insert sample data into Computer\_Users table**

INSERT INTO Computer\_Users (Computer\_Users\_Key, Computers\_Key, User\_Information)

VALUES (1, 1, 'mickey.mouse@users.grandfield.edu'),

(2, 2, 'minnie.mouse@users.grandfield.edu'),

(3, 3, 'goofy.dog@users.grandfield.edu');

**-- Insert sample data into Software\_Installations table**

INSERT INTO Software\_Installations (Software\_Installation\_Key, Software\_Version\_Key, Computers\_Key, Software\_Installation\_Number)

VALUES (1, 1, 1, 9000),

(2, 2, 2, 9001),

(3, 3, 3, 9002);

**-- Display data with complete information for Software\_Installations**

SELECT

Computer\_Users.User\_Information AS 'User Information',

Computers.Computers\_Key AS 'Computer #',

Computers.Computers\_ID AS 'Computer ID',

Software\_Version.Software\_Version\_Key AS 'Software Version #',

Software.Software\_Type AS SoftwareType,

Software\_Version.Version\_Number\_ID AS 'Version Number ID',

Software\_Installations.Software\_Installation\_Key AS 'Software Installation',

Software\_Installations.Software\_Installation\_Number AS 'Software Installation #'

FROM

Software\_Installations

JOIN

Computers ON Software\_Installations.Computers\_Key = Computers.Computers\_Key

JOIN

Computer\_Users ON Computers.Computers\_Key = Computer\_Users.Computers\_Key

JOIN

Software\_Version ON Software\_Installations.Software\_Version\_Key = Software\_Version.Software\_Version\_Key

JOIN

Software ON Software\_Version.Software\_Key = Software.Software\_Key;

4. Create a database diagram and create the relationships among tables.

A diagram of a college application

Description automatically generated with medium confidence

5. Documentation: Make a data dictionary that lists each table, all the columns for that table, and the data types for each column.

|  |  |
| --- | --- |
| Administration | |
|  | |
| Administration Key | (INT, PRIMARY KEY) |
| Administration ID | (INT, NOT NULL) |
| Contact Information | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| Faculty & Staff | |
|  | |
| Faculty & Staff Key | (INT, PRIMARY KEY) |
| Faculty & Staff ID | (INT, NOT NULL) |
| Contact Information | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| Departments | |
|  | |
| Departments Key | (INT, PRIMARY KEY) |
| Allowance / Restriction | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| IT Department | |
|  | |
| IT Department Key | (INT, PRIMARY KEY) |
| IT Department ID | (INT, NOT NULL) |
| Contact Information | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| Computer | |
|  | |
| Computer Key | (INT, PRIMARY KEY) |
| Computer ID | (INT, NOT NULL) |

|  |  |
| --- | --- |
| Computer User | |
|  | |
| Computer User Key | (INT, PRIMARY KEY) |
| Computer Key | (INT, NOT NULL, FOREIGN KEY - referencing Computers(Computer Key)) |
| Computer User Information | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| Software | |
|  | |
| Software Key | (INT, PRIMARY KEY) |
| Software Type | (VARCHAR(255), NOT NULL) |

|  |  |
| --- | --- |
| Software Version | |
|  | |
| Software Version Key | (INT, PRIMARY KEY) |
| Software Version ID | (INT, NOT NULL, FOREIGN KEY - referencing Software(Software Key)) |
| Version Number / Information | (VARCHAR(50), NOT NULL) |

|  |  |
| --- | --- |
| Software Installation | |
|  | |
| Software Installation Key | (INT, PRIMARY KEY) |
| Computer Key | (INT, NOT NULL, FOREIGN KEY referencing Software Version(Software Version Key)) |
| Software Version Key | (INT, NOT NULL, FOREIGN KEY referencing Computers(Computer Key)) |
| Software Installation Number | (INT, NOT NULL) |