

23111-Cwk2-S-Advanced Databases

Anna McCartain

10362766

8.12.2020

NORMALIZATION

UNF

To create this unnormalized table, I transformed the data from the information source into a table, removing repeating values and identifying a key.

Quiz_id	34		
Quiz_name	SQL		
Quiz_author	Peter Parker		
Quiz_available	Yes		
Quiz_duration	60		
Student_id	44		
Student_name	Duncan Hull		
Date_Of_Attempt	22/11/2020		
Question	Which SQL statement is used to extract data from a database?	Which SQL statement is used to insert new data in a database?	With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter?"
Answer_1	SELECT	INSERT NEW	SELECT * FROM Persons WHERE FIRSTNAME <> 'Peter'
Answer_2	OPEN	INSERT INTO	SELECT [all] FROM Persons WHERE FIRSTNAME = 'Peter'.
Answer_3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer_4	GET	ADD NEW	SELECT [all] FROM Persons WHERE FIRSTNAME LIKE 'Peter'

1NF

The key concepts from 1NF are as follows; No repeating groups, the values in each column must be atomic, remove repeating groups

(quiz questions) into another relation and identify its key as well as copying primary key to form a compound key (question and quiz ID)

<u>Quiz_id</u>	34
Quiz_name	SQL
Quiz_author	Peter Parker
Quiz_available	Yes
Quiz_duration	60
Student_id	44
Student_name	Duncan Hull
Date_Of_Attempt	22/11/2020

<u>Question_ID</u>	1	2	3
<u>Quiz_ID</u>	34	34	34
Question	Which SQL statement is used to extract data from a database?	Which SQL statement is used to insert new data in a database?	With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter?"
Answer_1	SELECT	INSERT NEW	SELECT * FROM Persons WHERE FIRSTNAME <> 'Peter'
Answer_2	OPEN	INSERT INTO	SELECT [all] FROM Persons WHERE FIRSTNAME = 'Peter'.
Answer_3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer_4	GET	ADD NEW	SELECT [all] FROM Persons WHERE FIRSTNAME LIKE 'Peter'

2NF

2NF ensures that every non-key attribute of the 1NF table is fully functionally dependent on the primary key. B is fully dependent on A if it is functionally dependent on A but not any subset of it. Given the Question_ID, we can find the question without the need

for Quiz_ID. The answer is dependant on the question, not the Quiz_ID

<u>Quiz_id</u>	34
Quiz_name	SQL
Quiz_author	Peter Parker
Quiz_available	Yes
Quiz_duration	60
Student_id	44
Student_name	Duncan Hull
Date_Of_Attempt	22/11/2020

<u>Question_ID</u>	1	2	3
Question	Which SQL statement is used to extract data from a database?	Which SQL statement is used to insert new data in a database?	With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter?"
Answer_1	SELECT	INSERT NEW	SELECT * FROM Persons WHERE FIRSTNAME <> 'Peter'
Answer_2	OPEN	INSERT INTO	SELECT [all] FROM Persons WHERE FIRSTNAME = 'Peter'.
Answer_3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer_4	GET	ADD NEW	SELECT [all] FROM Persons WHERE FIRSTNAME LIKE 'Peter'

<u>Question_ID</u>	1	2	3
<u>Quiz_ID</u>	34	34	34

3NF

3NF states A relation that is in 1NF and 2NF and in which no non-key attribute is transitively dependent on a key. If transitive dependencies exist on the key remove them by placing them in a new relation along with a copy of their determinant. Inputting a student_id will return multiple quizzes, given a quiz_id we can get multiple students. The transitive dependency is the link between students and quizzes.

Quiz

<u>Quiz_id</u>	34
Quiz_name	SQL
Quiz_author	Peter Parker
Quiz_available	Yes
Quiz_duration	60

Student

<u>Student_id</u>	44
Student_name	Duncan Hull

Student takes Quiz

<u>Quiz_id</u>	34
<u>Student_id</u>	44
Date_Of_Attempt	22/11/2020

Quiz has questions

<u>Question_ID</u>	1	2	3
<u>Quiz_ID</u>	34	34	34

Questions

<u>Question_ID</u>	1	2	3
Question	Which SQL statement is used to extract data from a database?	Which SQL statement is used to insert new data in a database?	With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter?"

Answer_1	SELECT	INSERT NEW	SELECT * FROM Persons WHERE FIRSTNAME <> 'Peter'
Answer_2	OPEN	INSERT INTO	SELECT [all] FROM Persons WHERE FIRSTNAME = 'Peter'.
Answer_3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer_4	GET	ADD NEW	SELECT [all] FROM Persons WHERE FIRSTNAME LIKE 'Peter'

Relational Schema

In this section, I have taken the 3NF form of the tables above and transferred into my relational schema. Student_takes_quiz and Quiz_has_questions are the 2 tables with a foreign key as we need to link which student takes which quiz and which quiz has what questions. I added some additional properties that should be present in the database; score, correct answer.

Quiz (quiz_id, quiz_name, quiz_author, quiz_available, quiz_duration):
 pk [quiz_id]

Student (student_id, student_name):
 pk [student_id]

Student_Takes_Quiz (quiz_id, student_id, date_of_attempt, score):
 pk [quiz_id]
 fk [student_id → student.student_id]

Quiz_Has_Questions (question_id, quiz_id):
 pk [quiz_id]
 fk [question_id → questions.question_id]

Questions (question_id, question, answer_1, answer_2, answer_3, answer_4, correct_answer):
 pk [question_id]

Tables

In this section, I have implemented the relational schema above by creating a database of tables. I was mindful of the data types for my fields and any referential actions for foreign keys to update or delete data from related tables.

Command Line

```
$ sudo mysql -u root -p;
mysql> CREATE DATABASE db;
Query OK, 1 row affected (0.00 sec)
mysql> USE db;
Database changed
mysql> SOURCE db_source;
Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)
```

Source:

```
CREATE TABLE Quiz(
quiz_id SMALLINT NOT NULL,
quiz_name VARCHAR(100) NOT NULL,
quiz_author VARCHAR(100) NOT NULL,
quiz_available BOOLEAN NOT NULL,
quiz_duration SMALLINT NOT NULL,
PRIMARY KEY (quiz_id)
);

CREATE TABLE Student (
student_id INT NOT NULL,
Student_name VARCHAR(100) NOT NULL,
PRIMARY KEY (student_id)
);

CREATE TABLE Student_Takes_Quiz (
quiz_id SMALLINT NOT NULL,
s_id INT NOT NULL,
```

```

date_of_attempt DATE,
Score SMALLINT NOT NULL,
PRIMARY KEY (quiz_id),
FOREIGN KEY (s_id)
    REFERENCES Student(student_id)
);

CREATE TABLE Questions (
question_id SMALLINT NOT NULL,
question VARCHAR(300) NOT NULL,
answer_1 VARCHAR(200) NOT NULL,
answer_2 VARCHAR(200) NOT NULL,
answer_3 VARCHAR(200),
answer_4 VARCHAR(200),
Correct_answer VARCHAR(200) NOT NULL,
PRIMARY KEY (question_id)
);

CREATE TABLE Quiz_Has_Questions (
q_id SMALLINT NOT NULL,
quiz_id SMALLINT NOT NULL,
PRIMARY KEY (quiz_id),
FOREIGN KEY (q_id)
    REFERENCES Questions(question_id)
);

```

The Application

This part was very challenging for me to even get my database connection through the vm. I would like to have got farther but due to time constraints I had to stop.

Firstly I made a page to ensure my connection to the database was working; I then created a table in my database for user's details (I would later link this to the students table). This table contained a BOOLEAN value to classify if user was a member of staff.


```
<!DOCTYPE html>
<html>
<head>
|   <title>Create Database</title>
</head>
<body>
```

```
<?php
```

```
    $host = "localhost";
    $un = "root";
    $pw = "root";
    $db = "db";
```

```
    $conn = mysqli_connect($host, $un, $pw, $db);
```

```
    if(!$conn)
    {
|       die("could not connect to MySQL" . mysqli_connect_error($conn));
    }
    else
    {
|       echo("connected to database server");
    }
```

```
    $sql = "CREATE TABLE user (
        userID int auto_increment primary key,
        userForename varchar(100) not null,
        userSurname varchar(100) not null,
        isStaff BOOLEAN not null,
        userEmail varchar(100) not null unique,
        userPassword varchar(255) not null
    )";
```

```
    if(mysqli_query($conn, $sql))
    {
|       echo ("Table created");
    }
```

```
    else
    {
|       echo(mysqli_error($sql));
    }
```

```
?>
```

```
</body>
</html>
```

I then transferred my connection over to a separate php document for clarity.

Users > annamccartain > Desktop > untitled folder > 🐘 conn.php

```
1
2      <?php
3      $host = "localhost";
4      $un = "root";
5      $pw = "root";
6      $db = "INSERT NAME ";
7
8      $conn = mysqli_connect($host, $un, $pw, $db);
9      ?>
10
```

The login page consisted of a simple if else depending on if the email was empty; and 2 functions.

```
<?php

    require("conn.php");

    if(!empty($_POST['userEmail']))
    {
        authenticateUser();
    }
    else
    {
        getUserDetails();
    }
```

This function creates a form on the html the user can fill out to login.

```
function getUserDetails()
{
    echo("
        <form method='POST' >
        <table>
            <tr><td>Email</td><td><input type='text' name='userEmail'></td></tr>
            <tr><td>Password</td><td><input type='text' name='userPassword'></td></tr>
        </table>
        <input type='submit' name='Login'>
        </form>
    ")
}
```

To authenticate the user; we had to first make the connection variable global so it could be accessed from this. I then made variables \$em and \$pw to hold the text values from the form. The SQL query fetches the users input password from the database under this email address under this connection. Then the password

entered is compared with that of the database; reverifying every time user retries to enter.

```
function authenticateUser()
{
    global $conn;

    $em = $_POST['userEmail'];
    $pw = $_POST['userPassword'];

    $pw = password_hash($pw, PASSWORD_DEFAULT);

    $sql = "SELECT userPassword FROM user WHERE userEmail = '$em'";

    if($result = mysqli_query($conn, $sql))
    {
        echo("SQL OK");
    }
    else
    {
        echo("something went wrong" . mysqli_error($conn));
    }

    while($row = mysqli_fetch_array($result))
    {
        if(password_verify($pw, $row['userPassword']))
        {
            echo("password is good");
        }
        else
        {
            echo("password incorred");
        }
    }
}

?>

</body>
</html>
```

The register page is similar to the login; just with adding in the additional details to create a user; I will post some screenshots below of my implementation.

```

<?php

require("conn.php");

if(!empty($_POST['userEmail']))
{
    addUserDetails();
}
else
{
    getUserDetails();
}

function getUserDetails()
{
    echo("
        <form method='POST' >
        <table>
            <tr><td>Forename</td><td><input type='text' name='userForname'></td></tr>
            <tr><td>Surname</td><td><input type='text' name='userSurname'></td></tr>
            <tr><td>isStaff</td><td><input type='TRUE/FALSE' name='isStaff'></td></tr>
            <tr><td>Email</td><td><input type='text' name='userEmail'></td></tr>
            <tr><td>Password</td><td><input type='text' name='userPassword'></td></tr>
        </table>
        <input type='submit' name='Register'>
    </form>
    ")
}

```

```

function addUserDetails()
{
    global $conn;

    $fn = $_POST['userForename'];
    $sn = $_POST['userSurname'];
    $is = $_POST['isStaff'];
    $em = $_POST['userEmail'];
    $pw = $_POST['userPassword'];

    $pw = password_hash($pw, PASSWORD_DEFAULT);

    $sql = "INSERT INTO user(
        userForename,
        userSurname,
        isStaff,
        userEmail,
        userPassword
    ) VALUES(
        '$fn', '$sn', '$is', '$em', '$pw'
    )";

    if(mysqli_query($conn, $sql))
    {
        echo("User has registered");
    }
    else
    {
        echo("something went wrong" . mysqli_error($conn));
    }
}

?>

</body>
</html>

```

The final page I made was a 'create Quiz' page. It involved 2 separate forms; one for creating the quiz, and one for creating a question. The question form can be used multiple times; each time it is submitted; the question will be linked to quiz via 'quiz has questions' table. I did not have time to implement my feature involving isStaff boolean; it would involve only users with this marked as TRUE to execute the sql statements. Additionally with UPDATE and DELETE of the quiz questions and tables would have been implemented in a similar layout to the CREATE, using the 'SELECT FROM' format from the login page.

```
<?php
```

```
require("conn.php");
```

```
if(!empty($_POST['quiz_id']))
```

```
{
```

```
    addQuizDetails();
```

```
}
```

```
else
```

```
{
```

```
    getQuizDetails();
```

```
}
```

```
function getQuizDetails()
```

```
{
```

```
    echo("
```

```
        <form method="POST" >
```

```
        <table>
```

```
            <tr><td>quiz_id</td><td><input type="text" name="quiz_id"></td></tr>
```

```
            <tr><td>quiz_name</td><td><input type="text" name="quiz_name"></td></tr>
```

```
            <tr><td>quiz_author</td><td><input type="text" name="quiz_author"></td></tr>
```

```
            <tr><td>quiz_available</td><td><input type="TRUE/FALSE" name="quiz_available"></td></tr>
```

```
            <tr><td>quiz_duration</td><td><input type="in mins" name="quiz_duration"></td></tr>
```

```
        </table>
```

```
        <input type="submit" name="Register">
```

```
    </form>
```

```
")
```

```
    echo("
```

```
        <form method="POST" >
```

```
        <table>
```

```
            <tr><td>question_id</td><td><input type="text" name="question_id"></td></tr>
```

```
            <tr><td>question</td><td><input type="text" name="question"></td></tr>
```

```
            <tr><td>answer_1</td><td><input type="text" name="answer_1"></td></tr>
```

```
            <tr><td>answer_2</td><td><input type="text" name="answer_2"></td></tr>
```

```
            <tr><td>answer_3</td><td><input type="text" name="answer_3"></td></tr>
```

```
            <tr><td>answer_4</td><td><input type="text" name="answer_4"></td></tr>
```

```
            <tr><td>correct_answer</td><td><input type="text" name="correct_answer"></td></tr>
```

```
        </table>
```

```
        <input type="submit" name="Register">
```

```
    </form>
```

```
")
```

```
}
```

```

function addQuizDetails()
{
    global $conn;

    $qi = $_POST['quiz_id'];
    $qn = $_POST['quiz_name'];
    $qa = $_POST['quiz_author'];
    $qav = $_POST['quiz_available'];
    $qd = $_POST['quiz_duration'];

    $qei = $_POST['question_id'];
    $qen = $_POST['question'];
    $a1 = $_POST['answer_1'];
    $a1 = $_POST['a1'];
    $a2 = $_POST['a2'];
    $a3 = $_POST['a3'];
    $a4 = $_POST['a4'];
    $ca = $_POST['ca'];

    $sql = "INSERT INTO Quiz(
        quiz_id,
        quiz_name,
        quiz_author,
        quiz_available,
        quiz_duration,
    ) VALUES(
        '$qi', '$qn', '$qa', '$qav', '$qd'
    )";

    if(mysqli_query($conn, $sql))
    {
        echo("quiz is added");
    }
    else
    {
        echo("something went wrong" . mysqli_error($conn));
    }
}

```

```

$sql1 = "INSERT INTO Questions (
    question_id
    question
    answer_1
    answer_2
    answer_3
    answer_4
    correct_answer)
VALUES (
    'qei', 'qen', 'a1', 'a2', 'a3', 'a4', 'ca')";
if(mysqli_query($conn, $sql1))
{
    echo("question added");
}
else
{
    echo("error:" . mysqli_error($conn));
}

$sql2 = "INSERT INTO Quiz_Has_Questionss (
    q_id
    quiz_id)
VALUES ('qi', 'qei')";
}

if(mysqli_query($conn, $sql2))
{
    echo("question connected to quiz");
}
else
{
    echo("error:" . mysqli_error($conn));
}

```

Stored Procedures and Triggers

I Created a stored procedure that displays the student names and their scores for the quizzes where they achieved less than 40%:

```

DELIMITER %%
CREATE PROCEDURE failedQuiz
BEGIN
    SELECT s_id, Score FROM Student_Takes_Quiz
    WHERE Score < 40;

```



```
END %%  
DELIMITER ;
```

I Created a trigger that will log the staff id, the quiz id and the current date and time, when a staff user deletes a quiz. I created a separate table to log these values.

```
CREATE TABLE DeletedField (  
  staffID INT NOT NULL,  
  quizID INT NOT NULL,  
  date_time DATETIME NOT NULL  
);  
  
CREATE TRIGGER logDelete  
BEFORE DELETE on Quiz FOR EACH ROW  
BEGIN  
  INSERT INTO DeletedField  
    SET quiz_id = OLD.quiz_id,  
        staff_id = OLD.staff_id,  
        date_time = CURRENT_TIMESTAMP();  
END
```