

COMP23111 Database Systems Coursework 2

Tamoor Shahid

December 2020

Contents

1	Part A: Normalisation	2
1.1	UNF	2
1.2	1NF	2
1.3	2NF	3
1.4	3NF	4
2	Part B: Relational Schema	5
3	Part C: Implementation	5
4	Part D: The Application	6
4.1	Registration and Logging in	6
4.2	Welcome to my application	6
4.3	Take a quiz	7
4.4	Create and Edit Your Own Quiz	8
5	Part E: Stored Procedures and Triggers	10
5.1	MySQL used to create the stored procedure.	10
5.2	MySQL used to create the trigger.	11

1 Part A: Normalisation

In order to do the coursework, first we have to normalise the data given to us. So I decided to use a spreadsheet to layout tables in order to put the data into UNF.

1.1 UNF

It was quite a simple process, I copied all the data from the report and put it in a spreadsheet. I also identified that the quiz ID is the key in the data.

Quiz ID	34		
Quiz Name	SQL		
Quiz Author	Peter Parker		
Quiz Available	Yes		
Quiz Duration	60 minutes		
Student ID	44		
Student Name	Duncan Hull		
Date of Attempt	22/11/2020		
Questions	Which SQL statements 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 Person WHERE FirstName = 'Peter'
Answer 3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer 4	GET	ADD NEW	SELECT [all] FROM Person WHERE FirstName LIKE 'Peter'

1.2 1NF

Now that UNF has been completed, I need to convert my data into 1NF. In order to do that, I need to check for atomicity and repeating values. There aren't any atomicity issues in the table besides the quiz duration. In order to fix that, we'll change the title so it refers to all quizzes in minutes so the field can display just the integer 60. The picture below doesn't show the change until 2NF as I forgot to change it until I reached the next stage. In terms of repeating values, the quiz data needs to be moved into a new table as it's repeating and doesn't fit in the table. I called the new table **Quiz Questions** to separate it from the original table. The new table also has the foreign key quiz ID and is combined with the question attribute in order to make a compound key so every question in the quiz has a unique identifier. Quiz ID also relates the first table to the new table too.

Quiz ID	34		
Quiz Name	SQL		
Quiz Author	Peter Parker		
Quiz Available	Yes		
Quiz Duration	60 minutes		
Student ID	44		
Student Name	Duncan Hull		
Date of Attempt	22/11/2020		
Quiz Questions			
Quiz ID	34	34	34
Questions	Which SQL statements 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 Person WHERE FirstName = 'Peter'
Answer 3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer 4	GET	ADD NEW	SELECT [all] FROM Person WHERE FirstName LIKE 'Peter'

1.3 2NF

For 2NF, I need to make sure that the non-key attributes are fully dependent on the keys of their respective table. In the table 'Quiz Questions', the answers for each question are fully dependent on the question being asked **and** the quiz itself as multiple quizzes could have the same question but a different set of answers. However, the original table doesn't have attributes that are fully dependent on 'Quiz ID'. Quiz Name, Quiz Author, Quiz Available and Quiz duration are all fully dependent on the quiz ID. However, the Student ID, Student Name and Date of Attempt isn't fully dependent. Student Name and Date of Attempt is reliant on both the quiz they took and the student that took that quiz. And Student ID doesn't have anything to do with Quiz ID. So they will be moved into separate tables. Two new tables will be created, one for the student's details and one for the date of attempt. As the date of attempt is about both the student and the quiz, it'll be the perfect place to include the score of the quiz too. The student information table will have the student ID as the primary key. The attempt table relies on both student ID and quiz ID so it will require a compound key of both foreign keys. In terms of relations, the Quiz ID links to both the attempt table and the quiz questions table. The student ID links to the attempt table too. Through the attempt table, the quiz ID connects to the student ID and vice versa.

Quiz ID	34	Student ID	44
Quiz Name	SQL	Student Name	Duncan Hull
Quiz Author	Peter Parker		
Quiz Available	Yes	Student ID	44
Quiz Duration (mins)	60	Quiz ID	34
		Date of Attempt	22/11/2020
		Score	3
Quiz Questions			
Quiz ID	34	34	34
Questions	Which SQL statements 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 Person WHERE FirstName = 'Peter'
Answer 3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer 4	GET	ADD NEW	SELECT [all] FROM Person WHERE FirstName LIKE 'Peter'

1.4 3NF

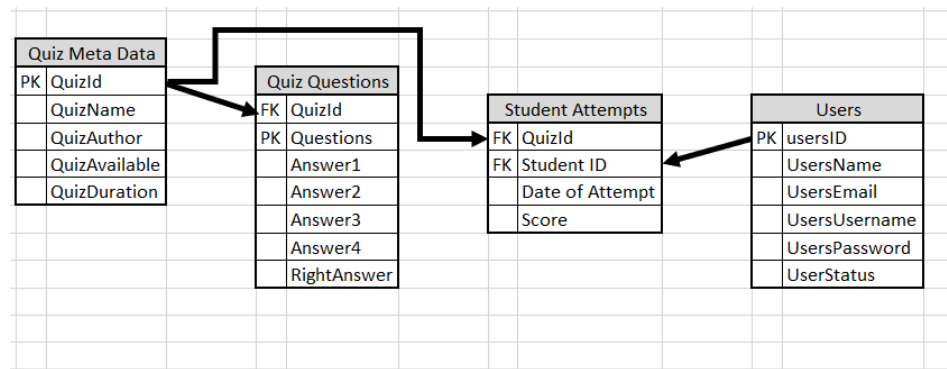
The final step is 3NF. In this step, we have to compare non-key attributes to each other in order to see if the attributes relied on any other attributes. I touched on this when comparing the attributes above to the primary key as they would have been separated here if they weren't separated earlier for not being related to the primary key. There aren't any non-key attributes that are dependent on any other non-key attribute so this step is completed.

Quiz ID	34	Student ID	44
Quiz Name	SQL	Student Name	Duncan Hull
Quiz Author	Peter Parker		
Quiz Available	Yes	Student ID	44
Quiz Duration (mins)	60	Quiz ID	34
		Date of Attempt	22/11/2020
		Score	3
Quiz Questions			
Quiz ID	34	34	34
Questions	Which SQL statements 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 Person WHERE FirstName = 'Peter'
Answer 3	EXTRACT	ADD RECORD	SELECT * FROM Persons WHERE FIRSTNAME = 'Peter'
Answer 4	GET	ADD NEW	SELECT [all] FROM Person WHERE FirstName LIKE 'Peter'

2 Part B: Relational Schema

Now that I've done the normalisation, I can create a database schema. As I've already mentioned how the tables are linked together, it isn't too difficult to create the relational schema.

Though I did have to add new fields to make sure the database would operate in the next two parts. The users table was required lots of details including log in details, whether they are a staff or student and several other details such as name to decide the authors of quizzes and to make the application feel more personal and welcoming. So the studentId table was replaced with this table. The 'Quiz Questions' table required a field that determined what the right answer was so I added that to that table.



3 Part C: Implementation

Now that my SQL planning has been completed, it's time to implement it. Installing XAMPP to create my database, I started. First I created a database using PHPmyAdmin. Afterwards, I started to create tables by using

```
CREATE TABLE 'quizmetadata' (  
    quizId int AUTO_INCREMENT NOT NULL UNIQUE PRIMARY KEY,  
    quizName varchar(255),  
    quizAuthor varchar(255),  
    quizAvailable BIT,  
    quizDuration int);
```

I did the same for the rest of the tables, resulting in four tables.

```
CREATE TABLE quizquestions (  
    quizId int,  
    questions varchar(500) NOT NULL,  
    answer1 text Not Null,
```

```

answer2 text Not Null,
answer3 text Not Null,
answer4 text Not Null,
rightAnswer INT NOT NULL,
PRIMARY KEY (questions, quizId),
FOREIGN KEY(quizId) REFERENCES quizmetadata(quizId))

```

Afterwards, I started to insert the sample data into the SQL database so I had something to look at and experiment with when creating the application instead of an empty database.

```

INSERT INTO 'quizquestions'(
    'quizId', 'questions', 'answer1', 'answer2', 'answer3', 'answer4')
VALUES (34,'Which SQL statement is used to extract from a
database?', 'SELECT', 'OPEN', 'EXTRACT', 'GET')

```

4 Part D: The Application

I made a small website using PHP and MySQL to use the database. Due to lack of time, I wasn't able to apply much CSS to make the application more appealing but that would be something I'd do if I were to continue the project.

4.1 Registration and Logging in

The application allowed users to do several things. They could log in or register an account.

- [Registration](#)
- [Log in](#)

Log In

Username:

Password:

- [Registration](#)
- [Log in](#)

Registration

Name:

Username:

Password:

Confirm Password:

Email:

☐ Staff
☐ Student

4.2 Welcome to my application

After logging in, you'll be greeted with a welcome page. The navigation menu at the top will have changed too, letting you see your profile, take quizzes, view

previous quizzes you have attempted and if you're a staff member then be able to edit and create quizzes. In the examples below, the user Tamamo is a student so she only is able to take quizzes and attempt quizzes. In the other example, Duncan is a member of staff so in addition to what a student can do, he can also create and edit quizzes.

- [Profile](#)
- [Take Quizzes](#)
- [Quiz Attempts](#)
- [Log Out](#)

The Front Page!

Hello there Tamamo!

Your email address is tamamo@fgo.co.jp and you are a student.

- [Profile](#)
- [Edit Quizzes](#)
- [Take Quizzes](#)
- [Quiz Attempts](#)
- [Log Out](#)

The Front Page!

Hello there Duncan Hull!

Your email address is DuncanHull@manchester.ac.uk and you are a staff.

The profile page is nothing special, just displays a few details about yourself.

- [Profile](#)
- [Edit Quizzes](#)
- [Take Quizzes](#)
- [Quiz Attempts](#)
- [Log Out](#)

Duncan Hull

Your user ID is: 4.

Your username is KittenDestroyer.

Your email address is DuncanHull@manchester.ac.uk

You are a staff.

4.3 Take a quiz

If one wants to take a quiz, they can click on the take quizzes link and be brought to a list of quizzes that can be taken. It'll display information about each quiz and a button so you can take the respective quiz. It won't display any non-available quizzes either, just the quizzes that are available.

List of Available Quizzes

Quiz ID	Quiz Name	Quiz Author	Quiz Duration	Take Quiz
56	Select the letter	Bethany	40 minutes	56
57	Summer	Mother Nature	25000 minutes	57
67	Laxy	Duncan Hull	23 minutes	67
69	Yah	Duncan Hull	56 minutes	69

When one starts taking a quiz, they'll be given a couple of radio options that

will keep track of the answer that the user has given. It'll go through each question in the database before finishing.

Which SQL statement is used to extract from a database?

- ☐ SELECT
- ☐ OPEN
- ☐ EXTRACT
- ☐ GET

Next Question

Eventually they will be brought to the page that shows their quiz attempts, including the quiz they just took. This page is also accessible through the navigation above.

Quiz Attempts

Quiz ID	Quiz Name	Quiz Author	Quiz Duration	Date Attempted	Last Score
56	Select the letter	Bethany	40 minutes	2020-12-10	1
67	Laxy	Duncan Hull	23 minutes	2020-12-10	3
69	Yah	Duncan Hull	56 minutes	2020-12-11	4

4.4 Create and Edit Your Own Quiz

Finally the last section the website is the edit quizzes section. Clicking on edit quizzes will bring you to a list of quizzes that the user has made, not showing any quizzes that the user hasn't made themselves. There's an option to edit quizzes if you want to. Or you can go lower and view two forms, whether to add a new quiz or to update the details of an existing quiz. To update a quiz, you must give the quiz's ID so the database knows which quiz should be updated. The checkbox next to 'Quiz Available?' can be left alone if you don't want the quiz to be available or can be ticked if you want the quiz to be available. I decided to make the quiz available column to consist of 1s and 0s but in hindsight, I could have chosen something even more user friendly. However, most people would understand what it means so it's not something that requires urgent change.

Edit Quizzes

Edit Existing Quizzes

Quiz ID	Quiz Name	Quiz Author	Quiz Duration	Quiz Available	Update Quiz
34	SQL	Duncan Hull	60 minutes	1	<input type="button" value="Edit"/>
66	Test for updates	Duncan Hull	76 minutes	0	<input type="button" value="Edit"/>
67	Laxy	Duncan Hull	23 minutes	1	<input type="button" value="Edit"/>
69	Yah	Duncan Hull	56 minutes	1	<input type="button" value="Edit"/>

Create a Quiz

Quiz Name

Quiz Duration

Update a Quiz

Quiz ID

Quiz Name

Quiz Duration

Quiz Available? ☐

Clicking on one of the edit quiz buttons will bring you to a page that will list details about each question and let you make changes to it. For example, the SQL quiz will display the title of its name in the top corner of the screen and have a table of its answers and which is the right one.

SQL

[Go back](#)

[Delete this Quiz](#)

Quiz Number	Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer
1	Do you like SQL?	NO	Er...	YES!	Maybe	3
2	Which SQL statement is used to extract from a database?	SELECT	OPEN	EXTRACT	GET	1
3	Which SQL statement is used to insert new data into a database?	INSERT NEW	INSERT INTO	ADD RECORD	ADD NEW	2
4	With SQL, how do you select all the records from a table named "Persons" where the value of the column "FirstName" is "Peter"?	SELECT * FROM Persons WHERE FIRSTNAME < "Peter"	SELECT [all] FROM Person WHERE FIRSTNAME = "Peter"	SELECT * FROM Persons WHERE FIRSTNAME = "Peter"	SELECT [all] FROM Person WHERE FirstName LIKE "Peter"	3

Then below it, there will be three more forms similar to the page before. But in this case, you just have to let the user know which question you wish to delete in terms of the number they appear in. So the first question would be 1, the second would be 2. There will be an option to add a new question, showing an error if all the fields aren't filled in as faulty input would ruin the database. Below that is an option to delete a question and below that is an option to update any existing question.

Add Question

Question:

Answer 1:

Answer 2:

Answer 3:

Answer 4:

Which is the right answer:

Add Question

Delete Question

Which question to delete:

Delete Question

Update Question

Which question to update:

Question:

Answer 1:

Answer 2:

Answer 3:

Answer 4:

Which is the right answer:

Update Question

If I were to continue on this project, I would use JavaScript so the quiz can be done on one page instead of having to go to different pages. And I would add a lot of CSS to make the website feel fun and nice to use.

5 Part E: Stored Procedures and Triggers

In addition to the application, I had made a stored procedure which can be used to display the names and scores of failed students (less than 40%) and I made a trigger that logs into a new table that records the user's ID, quiz's ID and the date of deletion.

5.1 MySQL used to create the stored procedure.

```
DELIMITER ^
CREATE PROCEDURE getFailures()
BEGIN
```

```

create temporary table idtbl
(
  quizId INT,
  questionTotal INT
);
# Combine all the quizId and get the total questions for each one.
INSERT INTO idtbl(quizId,questionTotal) SELECT quizId,COUNT(quizId)
FROM quizquestion GROUP BY quizId;

#Now to run through the studentAttempts table and compare it to the question
total. If they achieved less than 40%, they failed.
create temporary table quizScoretbl
(
  quizId INT,
  questionTotal INT,
  studentId INT,
  score INT
);
INSERT INTO quizScoretbl(quizId, questionTotal, studentId, score)
SELECT idtbl.quizId, idtbl.questionTotal, studentattempts.studentId,
studentattempts.score FROM idtbl INNER JOIN studentattempts ON
idtbl.quizId=studentattempts.quizId;

create temporary table quizScoreUpdatedtbl
(
  quizId INT,
  questionTotal INT,
  studentId INT,
  score INT
);
INSERT INTO quizScoreUpdatedtbl(quizId, questionTotal, studentId, score)
SELECT * FROM quizScoretbl WHERE score/questionTotal < 0.4;

SELECT users.userName, quizScoreUpdatedtbl.score FROM users INNER JOIN
quizScoreUpdatedtbl WHERE users.userId = quizScoreUpdatedtbl.studentId;

END ^
DELIMITER ;

```

5.2 MySQL used to create the trigger.

```

DELIMITER ^
CREATE TRIGGER updateQuizDeletionLog
BEFORE DELETE ON quizmetadata FOR EACH ROW
BEGIN
  DECLARE staffid int;

```

```
SELECT users.userId INTO staffid FROM users INNER JOIN quizmetadata WHERE  
users.userName = quizmetadata.quizAuthor AND quizmetadata.quizId=OLD.quizId;  
INSERT INTO quizdeletionlog(staffid, quizId, time)  
VALUES (staffid, OLD.quizId, NOW());  
  
END  
DELIMITER ;
```