CPT103 Coursework Report

# Must Read

Your report and database design must be your own work. You should not copy any code from others or let anyone develop this PMMS. Plagiarism and collusion lead to a zero mark for this coursework.

All tables must be in 3NF and the ER diagram should not contain any M: N relationships. Please strictly follow the structure of this template. Remember to double-check grammar and wording errors so that the report could be understood easily. Failing to do so will result in mark deductions. Any language other than English will be ignored when marking the report.

Your ID:

Name:

Email Address:

# Database Design Details

## Tables

In this section, you are required to explain all of the tables in your ER diagram. An example is given below. Please make sure you follow the template and everything is clearly explained.

------ The Beginning of the Example ------

[This is only an example. You don’t have to follow the same way I explained this table, but your explanation should be clear and detailed]

Table name: Staff

Table design general explanation:

The staff table is used to store the information of all staff members in the company. The primary key is staff\_id as it is unique for every staff member. [Add more explanations if you made some special considerations. Try to support your assumptions with proofs in real life]

Attributes:

|  |  |  |
| --- | --- | --- |
| Column Definition | Domain | Explanation |
| staff\_age int | 18+ | The age of staff members. The value should be larger than 18 because of the government law. This domain is checked by the domain constraint called XXXX. |
| address varchar(255) | Room *XX*, Building *YY*, *ZZ* Road, *WW* district | The address of the office, the data cannot be checked by the database directly. As a result, manual checking is required when entering data. |
| Staff\_name varchar(100) | All valid names are acceptable. For example, ‘Jun Qi’. | The name of staff members. |
| Branch\_no char(4) | Branch numbers start with B followed by 3 digits. For example, ‘b003’ | The branch number where this staff works in |
| Staff\_email varchar(255) | Valid email addresses  XXX@YYY.ZZZ | The correctness of email addresses cannot be checked directly by the database, so manual checking is required.  This column is not UNIQUE. It is an intentional design because the company sometimes offer shared email addresses for one office to improve communications. |
| Staff\_id int primary key | 2003001 | The staff id must be 7 numbers long and follow the format XXXYYYY, where XXX refers to the branch of that staff and YYYY is the individual unique number of that staff. |

Foreign keys:

The column Branch\_no references branch.branch\_no, this is to make sure that branch\_no are valid numbers that reflect existing branches of the company. It corresponds to the XXX relationship in the ER diagram.

------ The End of the Example ------

Table name:

Table design explanation

|  |  |  |
| --- | --- | --- |
| Column Definition | Domain | Explanation |
|  |  |  |
|  |  |  |
|  |  |  |

Foreign keys and reasons:

Table name:

Table design explanation

|  |  |  |
| --- | --- | --- |
| Column Definition | Domain | Explanation |
|  |  |  |
|  |  |  |
|  |  |  |

Foreign keys and reasons:

Table name:

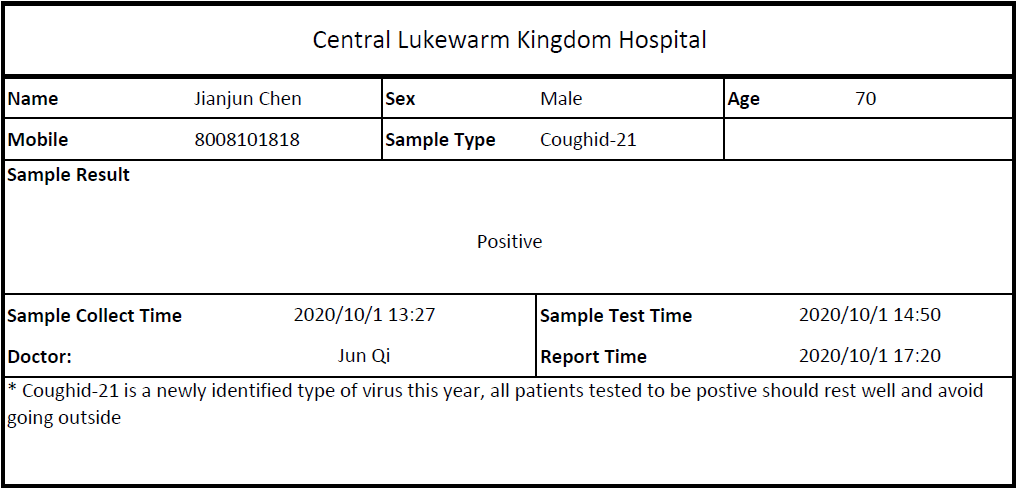
Table design explanation

|  |  |  |
| --- | --- | --- |
| Column Definition | Domain | Explanation |
|  |  |  |
|  |  |  |
|  |  |  |

Foreign keys and reasons:

[add more blocks if needed]

## Viral Test Report – Normalisation Process



Please write down the detailed normalisation process for the viral test report. Firstly, identify all attributes you can find in the viral test report as well as in the specifications (you need to add your own attributes if your database design has them). Then, at each normalisation stage, list all functional dependencies and normalise them to the higher normal form. 3NF is required for the final tables and must match your ER diagram.

**Stage 1**

**Attributes**:

**FDs (Indicate partial or transitive dependencies)**:

**Normalised tables and which normal form they are currently in**:

**Stage 2**

**Attributes**:

**FDs (Indicate partial or transitive dependencies)**:

**Normalised tables and which normal form they are currently in**:

**Stage 3 [If necessary]**

**Attributes**:

**FDs (Indicate partial or transitive dependencies)**:

**Normalised tables and which normal form they are currently in**:

# Use Cases

Remember to put all of your SQL statements into the SQL script file, including all SELECT statements and INSERT statements used to insert test data.

## Important Use Cases

This section lists some very important use cases of the PMMS. Your database design is expected to satisfy all of these use cases. Keep in mind that all use cases below should be achieved with a single SELECT statement (unless specified otherwise, sub-queries in a query is not counted as another query). Do not ignore the “explanation” or “proof” parts of this section, as they constitute the majority of your marks. If the SQL keywords/functions you learned cannot achieve these tasks, you are allowed to self-study some other keywords and use them. The example below is very simple and requires a short paragraph of explanation. But your answers should be more detailed.

------ The Beginning of the Example ------

***Use case 0***: Write a query to list all staff members in ‘B007’. In the result, list staff names.

Your SQL statement:

SELECT staff\_name FROM staff NATURAL JOIN branch where branch.branch\_no = 'B007'

Test data and explanation:

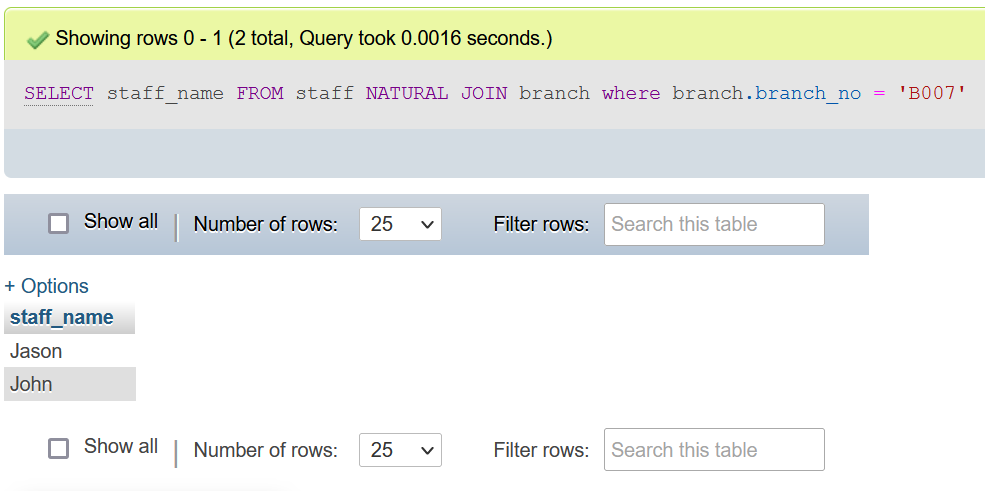
The following staff member information is added to the staff table (some attributes are hidden as they are not related to this task)

|  |  |
| --- | --- |
| Staff\_name | Branch\_no |
| ‘Jason’ | ‘B007’ |
| ‘Anna’ | ‘B002’ |
| ‘John’ | ‘B007’ |

The corresponding branch numbers ‘B007’ and ‘B002’ have already been added to the branch table.

This test data set contains staff members that are in ‘B007’ and not in ‘B007’. The expected result of the query should only contain staff in ‘B007’. Staff in other branches should be properly filtered out. For this test to work, all existing data in staff and branch need to be deleted first.

The result of the SELECT statement (screenshot):



Both the SQL statement and results must appear in the same screenshot!

------ The End of the Example ------

Use case 1: A person can potentially get infected if he was in the same district with someone. The government requires that, if someone is tested to be positive, all people in the same district as him in the past 48 hours (before the positive report is published) need to take viral tests. Assume that a person called Mark was tested to be positive at 19:30 on 09-Oct-2021. Mark’s telephone number is 233636. Please write a query that can get the phone numbers of all citizens who will potentially get infected because of him.

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 2: Please first clearly describe the format of GPS locations. The format must be a valid format that is used in real life. Then mimic what happens to your database when a user moves into the range of a base station and then moves out one hour later by listing all SQL statements involved in the process.

The GPS format and where did you learn it from (show the website link or the screenshot of the book):

Your SQL statement(s) for travel record insertion:

The result of the SELECT statements (screenshot):

Use case 3: The Lukewarm Kingdom wants to find out the hospitals that can do viral tests efficiently. The report generation time is calculated using (report time - sample test time). Please write a query to find out which hospital has the least average report generation time.

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 4: List the phone numbers of all citizens who did two viral tests with the time window from 2021-10-03 00:00 to 2021-10-05 00:00. The two viral tests must have a gap time of at least 24 hours (at least 24 hours apart).

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 5: List the high-risk, mid-risk and low-risk districts using one query. High-risk districts should be listed first, followed by mid-risk districts and then low-risk districts. Example:

|  |  |
| --- | --- |
| **district\_name** | **risk\_level** |
| Centre Lukewarm Hillside | high |
| Lenny town | high |
| Glow Sand district | mid |
| Raspberry town | low |
| Bunny Tail district | low |

Your SQL statement:

The result of the SELECT statement (screenshot):

Use case 6: List all positive cases found in the district called “Centre Lukewarm Hillside” on 2021-10-04. The result should include the names and phone numbers of people tested to be positive.

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 7: Calculate the increase in new positive cases in the district called “Centre Lukewarm Hillside” on 2021-10-05 compared to 2021-10-04. The result should show a single number indicating the increment. If there are fewer new positive cases than yesterday, this number should be negative.

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshots):

Use case 8: Assume that the spread rate of a virus is calculated by dividing the total number of people that were in the same district as the positive case with 48 hours (calculated in use case 1) by the total number of people among them that later confirmed to be infected in 14 days. Again, assume that a person called Mark (telephone number is 233636) was tested to be positive at 19:30 on 09-Oct-2021 and he is the only person in the country that has coughid-19. Please write a query that calculates the spread rate of the virus.

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshots):

## Extended Use Cases

Apart from the use cases proposed in the previous section, your database could also support more scenarios. Please follow the same format in the previous section and write down your own 10 use cases. You are allowed to use keywords learned outside of the lectures. Practical use cases displaying good innovations will receive higher marks.

Use case 1:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 2:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 3:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 4:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 5:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 6:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 7:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 8:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 9:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):

Use case 10:

Your SQL statement:

Your test data and why it can prove that the SELECT statement works (Important! Please explain carefully):

The result of the SELECT statement (screenshot):