

## COURSEWORK 1 - DATABASE

Module : **COMP1204 - Data Management**  
 Weighting : 30% of the overall module mark  
 Submit to : **Blackboard submission link named "Coursework 1 Submission".**  
 If there are difficulties, please submit to us by email according to the deadline  
 Deadline : March 18<sup>th</sup>, 2023, end of the day Malaysia Time  
 Feedback : 1-2 weeks after the deadline  
 Material : MySQL database  
 Prepared by Dr. Kristo Radion Purba ([kr.purba@soton.ac.uk](mailto:kr.purba@soton.ac.uk))

### General Notes

- Keep your code clean, complete, and easy to understand:
  - o No unnecessary codes
  - o Achieve the basic requirements stated in the question file, but feel free to add creativities to your code
- You can show your distinctive skills by experimenting with new and different things!
- You have to submit **ONE Word document report** and **ONE final SQL dump** which is produced using mysqldump command.
- For each question
  1. **Produce the MySQL queries**
  2. **Put the queries and briefly explain them** in your report
- You are expected to produce a report with a minimum of **6 (SIX) pages**.

### Rules

- Late submissions will be penalised at 10% per working day
- No work can be accepted after feedback has been given
- You should expect to spend up to **20 hours** on this assignment
- Please note the University regulations regarding academic integrity
- The submitted coursework must be the student's **ORIGINAL WORK**.
- Any kind of plagiarism, such as copying an existing work from the internet, or copying works from/to other students, will be penalized

This coursework aligns with the following module objectives:

- A1. The role of database systems in information management
- A2. The concept of data modelling
- A3. The relational model of data
- B2. Apply entity-relationship modelling
- B3. Normalise data
- D4. Use SQL to create, update and query a database

### Smartwatch (Health Tracker) App - Features

You and your team are making a new smartwatch device, and therefore your team need to build an application to connect it to a smartphone. You are asked to **develop a database for a smartwatch (health tracker) application**. First, choose ONE of the following applications for your reference:

- Samsung Health (<https://play.google.com/store/apps/details?id=com.sec.android.app.shealth>)
- Apple Watch app (<https://apps.apple.com/us/app/watch/id1069511734>)
- Google Fit (<https://play.google.com/store/apps/details?id=com.google.android.apps.fitness>)
- Zepp (<https://play.google.com/store/apps/details?id=com.huami.watch.hmwatchmanager>)
- Garmin Connect  
(<https://play.google.com/store/apps/details?id=com.garmin.android.apps.connectmobile>)
- You can choose other similar apps as long as it is quite popular; don't choose crappy ones.

As a new company, it is much easier to replicate it from other successful apps, compared to developing it from scratch. From the reference that you have chosen, **reverse engineer** (rebuilding it by just looking at it from the user's perspective) the **database design**.

You plan this database to be **deployed on the server**, not on the user's phone. Thus, the database must support multi-users.

It will have the following **MINIMUM** features (You need to add tables or columns as necessary):

- User** data. The app will work with multiple users. Users can log in and register. Typically, the users need to enter their phone number, email, and other biographical data.
- User History**. This is to register the user's general activity log, such as the last login date, what menus they access, how long they stay in the menu, etc. This is useful for analytical purposes.
- Heart Rate**. The smartwatches in the market typically record heart rates every certain interval. Users can also trigger the measurement manually.
- Exercise Type**. The app will have a huge selection of exercises, such as swimming (indoor and outdoor), running, treadmill, etc.
- Exercise**. Users can select the exercise that they are currently taking. During the exercise, the watch will record the heart rate, distance and pace (for swimming, running, etc.).
- User Settings**. This table records the user's settings, such as measurement unit, preferred watch face, purchased watch face, etc.
- Blood Oxygen Level**. This table records the history of the user's oxygen saturation level.
- Step Count**. This records the step count in each particular time window. For example, from 2022-02-01 19:35:00 to 2022-02-01 19:40:00, the user's step count is 120.
- Alarms**. This records the alarms set by the user.
- Calorie Counter**. This records the calories burnt in each particular time window (similar to step count).
- Sleep Quality**. This records the sleep quality in each particular time window (similar to step count). Typically, sleep can also be categorized as light sleep, deep sleep, or REM sleep.
- You can add more tables. Refer to the following links for ideas:
  - <https://drkumo.com/10-essential-health-features-of-smartwatches/>
  - <https://medium.com/@TechForAging/7-interesting-features-of-a-smart-watch-2eea36b9c55c>

### A. Database Creation (ER Diagram, Normalization, CREATE TABLE) - 30% Mark

Your task for the database creation:

- Physical ER Diagram:** Consider database normalizations since the beginning. Design the normalized ER Diagram straight away; no need to design the raw, unnormalized, database. Don't forget to include the cardinalities (one-to-one, one-to-many, optional, mandatory). Since this is a physical ERD, there will be no many-to-many connections between two tables; rather, you should make a **bridge table** if there is a many-to-many connection.
- Database Normalization Explanation:** Explain how you do the normalization, and put your proper justifications.
- Create Table Script:** Write your CREATE TABLE queries in the report, including the appropriate constraints (primary keys, uniques, foreign keys), and data types. Do not overuse data types such as VARCHAR(4000).
- Note that, typically, **derivative attributes** shouldn't be a part of the tables; they are attributes that can be derived by calculation from other data. For example, rather than storing km/h speed (for running), store the distance (km) and the duration (hours).

Assessment Criteria:

- This question aligns with objectives A2, A3, B2, B3
- ER Diagram: 30% Mark
- Normalization: 30% Mark
- Create table: 20% Mark
- Having creative justifications and/or improvements: 20% Mark

### B. Data Insertion (INSERT) - 10% Mark

After your data has been created, you need to:

- Input at least 10 data in each table.** **Do not** input random data; insert sensible ones.
- Provide the **insertion queries** on your report.

Assessment Criteria:

- This question aligns with objectives D4
- SQL query quality: 100% Mark

### C. Analytics (VIEW, SELECT) - 40% Mark

Reporting or data analytics is an important part of a system, especially for managerial decisions. For example, a manager will ask some analytical questions like "How is the app usage statistics each month, grouped by month?". Your task:

- Create a minimum of 30 (THIRTY) pairs of "**Analytical Questions and Queries**". An example of one expected answer could be:  
**Question:** How is the app usage on each month, grouped by month?  
**Query:** SELECT MONTH(entry\_date), AVG(quit\_time - login\_time) FROM  
 user\_history uh INNER JOIN other\_table t ON t.id\_field=uh.id\_field  
 GROUP BY MONTH(entry\_date)  
 ORDER BY some\_columns
- The query will typically involve **JOIN, GROUP BY, and ORDER BY**. The queries are to be justified by complexity.

Assessment Criteria:

- This question aligns with objectives A1, A2, D4
- SQL query quality: 30% Mark
- Analytical question and SQL query complexity: 70% Mark

#### D. Scalability Scenario 1 - Merging - 10% Mark

**Note: You must only proceed to this step once you have done the previous tasks.**

At this point, you have successfully implemented the database design for the smartwatch (health tracker) application, congratulations! Database **scalability** is an important consideration when designing a database. It is the ability to handle changing demands by adding/removing resources.

Your application and smartwatch products are now very successful, with a lot of users and counting. Your company wants to offer **Brand X** (another popular smartwatch brand) for acquisition, and company X agrees. In simple terms, your company buys **Brand X**, and now your database and **Brand X**'s database **will be merged**. Assuming your database and X's database designs are almost similar, the following will happen:

- a. From the user's perspective, they will feel a seamless transition.
- b. Your boss wants you to create a flag/marker from which a certain user is registered; from **Brand X**, from our own brand, or from new users after the acquisition.

Your task:

- a. **Explain in detail in your report**, the steps that you need to do to accommodate the needs described above; it can be what columns to add to what table, what tables to add, etc.
- b. **Write the QUERIES** necessary to accommodate the changes; it can be the queries to add columns, create new tables, etc.

Assessment Criteria:

- This question aligns with objectives A1, A2, A3, B3, D4
- SQL query quality: 50% Mark
- Having creative solutions and justifications: 50% Mark

#### E. Scalability Scenario 2 - Split - 10% Mark

**Note: You must only proceed to this step once you have done the previous tasks, including Scalability Scenario 1.**

Three months after you have done the acquisition (merging) process with **Brand X**, your company decides to make some changes. The changes are:

- a. There are some conflicts among the top management in the company. Your company decides to split the application into two, **App A** and **App B**. Note that this split process often happens for various reasons, such as Huawei and Honor, Douyin and TikTok, etc.
- b. Even though the smartwatch app (and the whole smartwatch business) is split into two, due to budget constraints, the management decided to **keep only one database instance for now**.
- c. Existing users will be split randomly and (almost) equally into **App A** and **App B**. You should write a query to randomly assign the users to **App A** or **App B**.
- d. The database must have a flag/marker, the user uses the database from **App A** or **App B**.

Your task:

- a. **Explain in detail in your report**, the steps that you need to do to accommodate the needs described above; it can be what columns to add to what table, what tables to add, etc.
- b. **Write the QUERIES** necessary to accommodate the changes; it can be the queries to add columns, create new tables, etc.

Assessment Criteria:

- This question aligns with objectives A1, A2, A3, B3, D4
- SQL query quality: 50% Mark
- Having creative solutions and justifications: 50% Mark

----- END OF FILE -----