INFO20003 Semester 1, 2021

Assignment 2: SQL
Due: 6:00pm Friday 30 April 2021
Submission: Via LMS https://canvas.lms.unimelb.edu.au/

Weighting: 10% of your total assessment

Phonemon Case Study

You are producing an augmented reality phone game which allows players to move around Melbourne, capturing virtual creatures called Phonemon (short for Phone Monster). The Phonemon are superimposed onto the physical world by locating them at points on the map and visualizing them via the app. (This fictitious game is based on Nintendo's popular app *Pokemon Go*.)

We record about each player the date and time when they joined the game, their unique id, username and which team they chose to join (if they are part of the team). We regularly update their current location and level. Players begin on level 1 and level up according to their achievements. There are three teams to choose from: each has a colour, a (fictitious) leader, and a Phonemon mascot.

Each species has a title, description and can be of 1 or maximally 2 different "types".

Individual Phonemons spawn regularly. We keep track of each one, recording when and where it spawns. A Phonemon starts as "wild" (the player column is null), and if it is captured, the player id is recorded. Each Phonemon has the species if belongs to and a "power" score (which may change as the game proceeds).

Our game allows in-app purchases of a range of items whose properties we record. Some items have extra properties that are recorded in a separate entity. We record the details of each purchase, including the item and quantity bought, and date and time of the purchase. Items can be of type 'F' (if they are food) or 'M' (if they are medicine), or none if they don't belong to either of the two groups. Items have a title and price. Both food and medicine are worth some points stored in tables Food and Medicine respectively.

All locations in Phonemon are expressed as a pair of decimal numbers representing latitude and longitude. Calculating the distance between two points P1 and P2 requires a complex formula which you can read about at https://en.wikipedia.org/wiki/Haversine_formula. For the purposes of this assignment, you can use the following simplified formula based on the Euclidean distance, which works well enough in Melbourne:

distance in km = *sqrt((P1.latitude – P2.latitude)^2 + (P1.longitude – P2.longitude)^2) * 100*

The Data Model

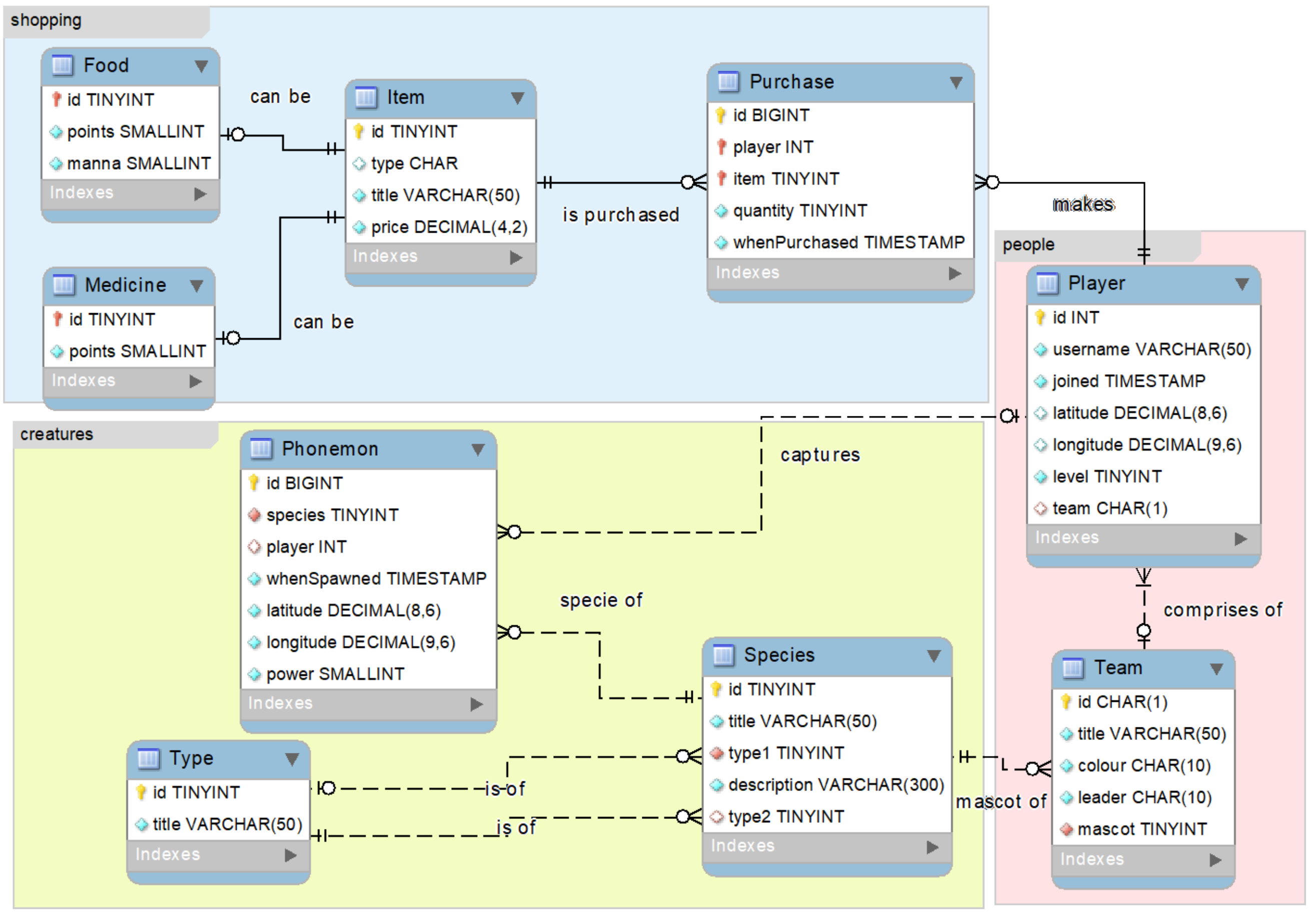


Figure 1: The ER Model for Phonemon Database

Assignment 2 Setup

A dataset is provided against which you can test your solutions to the assignment. To set up the dataset, download the file **Phonemon_2021.sql** from the Assignment on Canvas and run it in Workbench. This script creates the database tables and populates them with data.

The script is designed to run against your account on the Engineering IT server (info20003db.eng.unimelb.edu.au). If you want to install the schema on your own MySQL Server installation, uncomment the three lines at the beginning of the script that create the schema on your local server.

***Note: Do NOT disable full_group_by mode when completing this assignment**. This mode is the default, and is turned on in all default installs of MySQL workbench. You can check whether it is turned on using the command "SELECT @@sql_mode;". It should return a string containing "full_group_by".*
When testing, our test server WILL have this mode turned on, and if your query fails due to this, you will lose marks. You can run the command:

```
SET sql_mode=(SELECT CONCAT(@@sql_mode,',ONLY_FULL_GROUP_BY'));
```

… to ensure that this mode is configured properly.

The SQL Tasks

In this section are listed 10 questions for you to answer. Write one (single) SQL statement per question. Subqueries and nesting are allowed within a single SQL statement. However, you may be penalized for writing *overly* complicated SQL statements. <u>DO NOT USE VIEWS (or 'WITH' statements/common table expressions) to answer questions.</u>

1. How many species have a description which contains the word "this"? Your query should return results of the form: (speciesCount). **(1 mark)**
2. Player 'Cook' is about to battle player 'Hughes'. For both players, show the player's username and the total summed power of all the Phonemons they own. Your query should return results of the form: (username, totalPhonemonPower). **(1 mark)**
3. How many players does each team have? List the team names with their player counts, in descending order. Return results as: (title, numberOfPlayers). **(1 mark)**
4. Which species have a type of "grass"? Return results as: (idSpecies, title) **(2 marks)**
5. List the players who never purchased any food item. Your query should return results of the form: (idPlayer, username). **(2 marks)**
6. Each player is at a particular level in the game. What is the total amount that has been spent on purchases by all players of a given level? Your query should return results of the form: (level, totalAmountSpentByAllPlayersAtLevel) in the descending order of the amount spent. **(2 marks)**
7. Which item was purchased the most? In case of a tie, find all such items. Your query should return results of the form: (item, title, numTimesPurchased) **(2 marks)**
8. Find the number of (distinct) food items available, and any players who have purchased all types of food items at least once. Your query should return results of the form: (playerID, username, numberDistinctFoodItemsPurchased). **(3 marks)**
9. We'll refer to the Euclidean distance, rounded to 2 decimal places, between the *closest* two Phonemon as 'X'. We wish to count the number of Phonemon PAIRS which are at distance (to 2 decimals places) X from each other. Return as (numberOfPhonemonPairs, distanceX). *HINT1: use the ROUND() function. HINT2: Ensure you are not double counting the pairs: eg phonemon1 and phonemon2 are distance 0.11 from each other. If this is the minimum distance and no other phonemon are also at distance 0.11 from another phonemon, then the return value should be (1, 0.11), since there is a single PAIR which are at distance 0.11 from each other.* **(3 marks)**
10. Some players are really into a certain type of Phonemon... List the usernames of players that have captured at least one of every species of a given type, and the title of that type. Return a separate row for each type that a player has caught all species of. Your query should return results of the form: (username, title). *An example: if there are 3 Phonemon species with a type of 'bug', and player 'Greg' has caught at least 1 Phonemon of each of these species, then Greg will appear in the list as (Greg, bug). If additionally, Greg had caught one of every species with type 'fairy', then a second row of the output would be (Greg, fairy).* **(3 marks)**

SQL Response Formatting

To help us mark your assignment queries as quickly/accurately as possible, please ensure that:

- Your query returns the projected attributes in the same order as given in the question, and do not include additional columns. E.g., if the question asks for (userId, name), please write **`"SELECT userId, name …"`** instead of **`"SELECT name, userId, phone …"`** *(you can name the columns using '**AS'** however you'd like, only the order matters)*
- Please do NOT use "databaseName.tableName" format. E.g., please write **`"SELECT userId FROM users…"`** instead of **`"SELECT userId FROM coltonc.users …"`**
- Ensure that you are using single quotes(') for strings (e.g. **`…WHERE name = 'bob'…`**) and double quotes (") only for table names (e.g. **`SELECT name FROM "some table name with spaces"…`**)
- Ensure that you match the capitalisation of table/attribute names: some OS's are case insensitive but others are case sensitive. E.g. for an attribute "name" in table "user", please write **`"SELECT name, FROM user …"`** instead of **`"SELECT Name FROM User…"`**

Submission Instructions

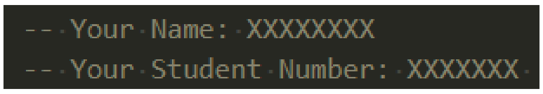
Your submission will be in the form of an SQL script. There is a template file on the LMS, into which you will paste your solutions and fill in your student details (more information below).

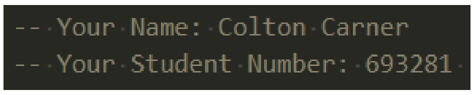
This .sql file should be submitted on Canvas by **6pm** on the due date of **Friday 30 April**. Name your submission as 987654.sql, where 987654 corresponds to YOUR student id.

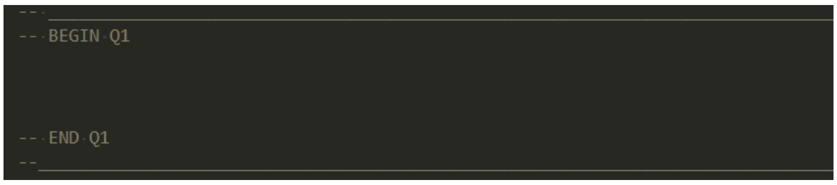
Filling in the template file:

The template file on the LMS has spaces for you to fill in your student details and your answers to the questions. There is also an example prefilled script also available on the LMS. Below are screenshots from those two documents explaining the steps you need to take to submit your solutions:

Step	**Example**

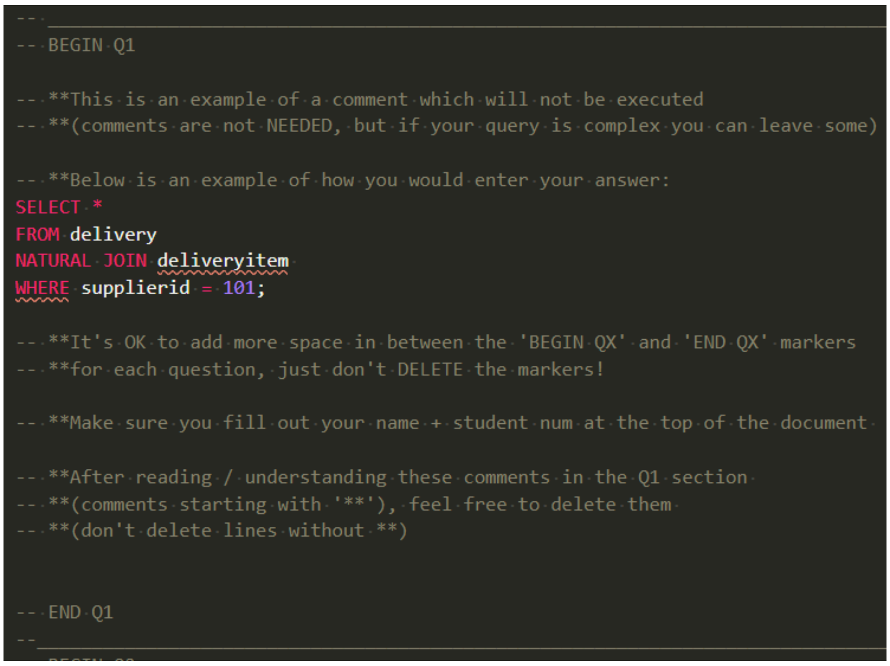
1. At the top of the template, you'll need to replace "XXXXXXXX" with your student number and name	Template
	
	`-- Your Name: XXXXXXXX`
	`-- Your Student Number: XXXXXXX`

	Example Filled in
	
	`-- Your Name: Colton Carner`
	`-- Your Student Number: 693281`

2. For each question 1-10, place your SQL solution in between the "BEGIN QX" and "END QX" markers. **Ensure each query is**	Template
	
	`--`
	`-- BEGIN Q1`
	
	`-- END Q1`
	`--`

terminated with a semicolon “;”

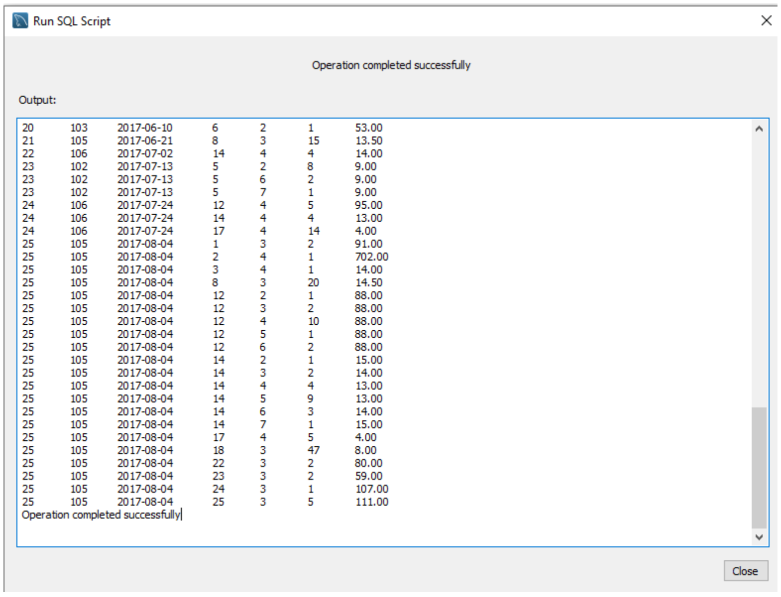
Example Filled in



```
--
-- BEGIN Q1

-- **This is an example of a comment which will not be executed
-- **(comments are not NEEDED, but if your query is complex you can leave some)

-- **Below is an example of how you would enter your answer:
SELECT *
FROM delivery
NATURAL JOIN deliveryitem
WHERE supplierid = 101;

-- **It's OK to add more space in between the 'BEGIN QX' and 'END QX' markers
-- **for each question, just don't DELETE the markers!

-- **Make sure you fill out your name + student num at the top of the document

-- **After reading / understanding these comments in the Q1 section
-- **(comments starting with '**'), feel free to delete them
-- **(don't delete lines without **)


-- END Q1
--
```

3. Test that your script is valid SQL by running it from MySQL Workbench (file > “Run SQL Script”). Make sure to select the default schema as the Phonemon schema you imported when prompted.

Running the example filled in template provided, using the schema we’ve used in labs (dep-store) results in ‘success’. (You would run yours on the schema for the A2 assignment)



Operation completed successfully

Output:

```
20    103    2017-06-10    6     2    1     53.00
21    105    2017-06-21    8     3    15    13.50
22    106    2017-07-02    14    4    4     14.00
23    102    2017-07-13    5     2    8     9.00
23    102    2017-07-13    5     6    2     9.00
23    102    2017-07-13    5     7    1     9.00
24    106    2017-07-24    12    4    5     95.00
24    106    2017-07-24    14    4    4     13.00
24    106    2017-07-24    17    4    14    4.00
25    105    2017-08-04    1     3    2     91.00
25    105    2017-08-04    2     4    1     702.00
25    105    2017-08-04    3     4    1     14.00
25    105    2017-08-04    8     3    20    14.50
25    105    2017-08-04    12    2    1     88.00
25    105    2017-08-04    12    3    2     88.00
25    105    2017-08-04    12    4    10    88.00
25    105    2017-08-04    12    5    1     88.00
25    105    2017-08-04    12    6    2     88.00
25    105    2017-08-04    14    2    1     15.00
25    105    2017-08-04    14    3    2     14.00
25    105    2017-08-04    14    4    4     13.00
25    105    2017-08-04    14    5    9     13.00
25    105    2017-08-04    14    6    3     14.00
25    105    2017-08-04    14    7    1     15.00
25    105    2017-08-04    17    4    5     4.00
25    105    2017-08-04    18    3    47    8.00
25    105    2017-08-04    22    3    2     80.00
25    105    2017-08-04    23    3    2     59.00
25    105    2017-08-04    24    3    1     107.00
25    105    2017-08-04    25    3    5     111.00
Operation completed successfully
```

Requesting a Submission Deadline Extension

If you need an extension due to a valid (medical) reason, you will need to provide evidence to support your request by *6pm, Thursday 29 April*. Medical certificates need to be at least two days in length.

To request an extension:

- Email Farah Zaib Khan (farah.khan@unimelb.edu.au) from your university email address, supplying your student ID, the extension request and supporting evidence.
- If your submission deadline extension is granted you will receive an email reply granting the new submission date. Do not lose this email!

Reminder: INFO20003 Hurdle Requirements

To pass INFO20003, you must pass two hurdles:

- **Hurdle 1:** Obtain at least 50% (15/30) or higher for the three assignments (each worth 10%)
- **Hurdle 2:** Obtain at least 50% (35/70) or higher for the combination of quizzes and end of semester exam

Therefore, it is our recommendation to students that you attempt every assignment and every question in the exam.

GOOD LUCK!