Exercise 13-4 Add a database to the Tip Calculator app

In this exercise, you'll create a database that you can use with the Tip Calculator app.

Add a database class

1. Start Android Studio and open the project named ch13_ex4_TipCalculator.
2. Review the code. Note that it includes a Tip class that you can use to store the data for a tip.
1. Add a database class that creates a table with these column names and data types:

   ```
   _id             INTEGER
   bill_date       INTEGER
   bill_amount     REAL
   tip_percent     REAL
   ```

   When the database class creates the database, it should also insert two rows of test data. (You can use 0 for the bill date values, but make up some bill amount and tip percent values such as 40.60 and .15.)
2. Add a public getTips method that returns an ArrayList<Tip> object that contains all columns and rows from the database table.
3. Switch to the activity class. Then, add code to its onResume method that calls the getTips method and loops through all saved tips. For each saved tip, this code should use LogCat logging to send the bill date milliseconds, the bill amount, and the tip percent to the LogCat view.
4. Run the app. At this point, it should create the database, add two rows of test data, and print the data for each row to the LogCat view.

Save tip calculations

5. Switch to the database class and add a method that saves tip calculations.
6. In the layout for the activity, add a Save button below the other widgets.
7. In the class for the activity, add code that saves the current bill amount and tip percent to the database when the user clicks the Save button. This code should also clear the bill amount from the user interface.
8. Run the app and test the Save button. To do that, you can execute the onResume method by navigating away from the app and navigating back to it. Then, you can check the LogCat view to make sure the data is saved when you click on the Save button.

Display the date and time of the last saved tip

9. Switch to the database class and add a method that returns a Tip object for the last tip that was saved.

10. Add code to the onResume method that displays the date and time of the last saved tip in the LogCat view. This date and time should be in this format:

    ```
    Sep 4 2016 15:54:00
    ```

    You can use a method of the Tip class to return this format.

11. Execute the onResume method again. Then, check the LogCat view to make sure the data is displayed correctly.

Set default tip percent to average tip percent

12. Switch to the database class and add a method that gets the average tip percent. To do that, you can provide an array of strings for the columns parameter like this:

    ```
    String columns[] = {"AVG(" + TIP_PERCENT + ")"};
    ```

    (This assumes the TIP_PERCENT constant contains the name of the column that stores the tip percent.)

13. Add code to the onResume method that displays the average tip percent in the LogCat view.

14. Execute the onResume method again by navigating away from the app and navigating back to it. Then, check the LogCat view to make sure the average tip percent is displayed correctly.

15. Modify the code for the Save button so it sets the tip percent to the average tip percent. That way, after you click the Save button, it should set the tip percent to the average tip percent.