

**COMP9120 Database Management Systems**

## **Assignment 2: Database Application Development**

**Group assignment (16%)****Introduction**

The objectives of this assignment are to gain practical experience in interacting with a relational database management system using an Application Programming Interface (API) (Python DB-API). This assignment additionally provides an opportunity to use more advanced features of a database such as functions.

This is a group assignment for teams of 3 members. It is assumed that you will continue with your Assignment 1 group, or that you have already requested a group change. Group formations are now final.

Please also keep an eye on your email and any announcements that may be made on Ed.

**Submission Details**

The final submission of your database application is due at **11:59pm on Sunday 18th May (Week 11)**. You should submit the items for submission (detailed below) via Canvas.

***Items for submission***

Please submit your solution to Assignment 2 in the 'Assignment' section of the unit's Canvas site by the deadline, including **EXACTLY THREE** files:

- An assignment coversheet as a PDF document (.pdf file suffix) which is available for download from [this link](#) on Canvas.
- A SQL file (**SAGschema.sql**) containing the SQL statements necessary to generate the database schema and sample data. This must contain the **supplied** original schema and insert SQL statements, **and your additional stored procedures (functions)**.

## COMP9120 Assignment 2

- A Python file (`database.py`) containing the Python code you have written to access the database.

Your code should be your own, except for the scaffold code provided. You are strictly prohibited from using any generative AI tools, such as Microsoft Copilot or ChatGPT to complete any part of this assignment. Failure to comply with this policy may result in academic penalties as outlined in the academic integrity guidelines.

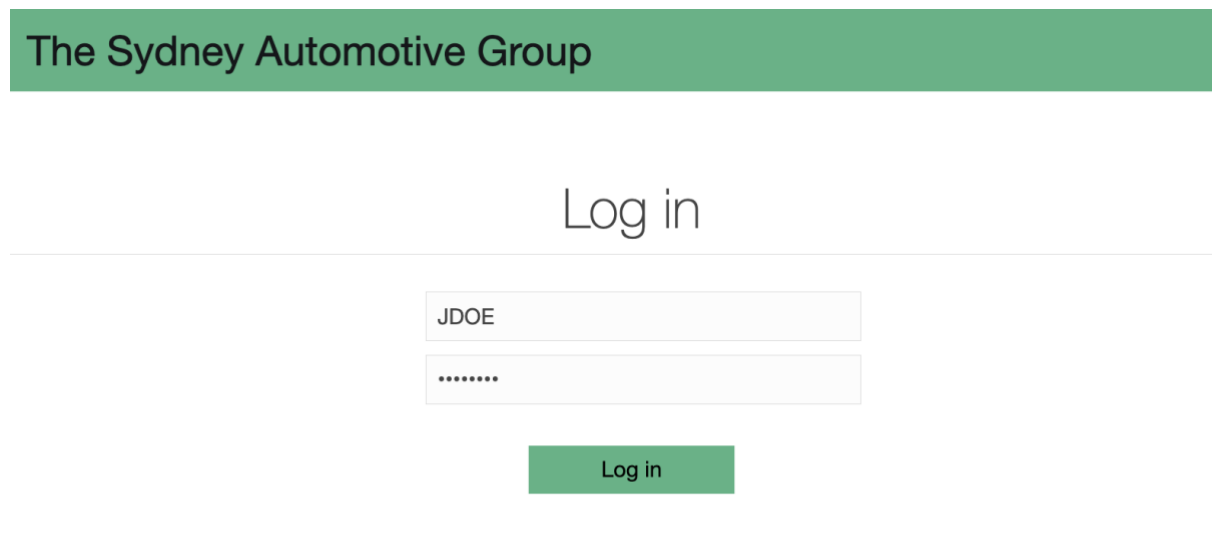
## Task 1: The Sydney Automotive Group (SAG) System

In this assignment, you will be working with a modified version of the SAG database as described in Assignment 1. This assignment builds upon and partially use your work designing the SAG database. More specifically, your task is to implement an interface, referred to as SAG system, through which a user interacts to access and manipulate the SAG database. Your main task in this assignment is to handle requests for reads and writes to the database using your User Interface (UI). We first describe the main features that the SAG system must include from a UI perspective, and then discuss where your database code needs to be implemented.

### ***Logging In***

The user is defined as any authorized salesperson at the SAG.

The first form presented to a user when starting the SAG system, is the **Login**, as shown in Figure 1. This feature requires the salesperson enters their username and password to be validated prior to being successfully logged in to the system. Security features such as password encryption/hashing is out of scope for this assignment. The username should be case insensitive. Once logged in, the user is directed to the Car Sales Summary page, where they can view a summary of car sales data for all makes and models.



The Sydney Automotive Group

Log in

JDOE

.....

Log in

*Figure 1: Login*

### ***Viewing CarSales Summary***

Once a user is logged in, they are directed to the Car Sales Summary page, as shown in Figure 2 below. The list of car sales must be ordered by make name and model name in ascending order. The summary includes the following information:

- *Make*: The name of the car's brand (e.g. Toyota, Ford, BMW, etc)
- *Model*: The name of a specific type of a car produces by a manufacturer. (e.g. Rav4, NX, etc)

- *Available Units*: The number of units of that specific make and model that are still available for sale.
- *Sold Units*: The number of units of that make and model that have been sold to customers.
- *Total Sales (\$)* : The total revenue generated from the sold units of that make and model.
- *Last Purchased At*: The date when the last car of that make and model was sold. It needs to be displayed in the Australian date convention (Date-Month-Year).

The Sydney Automotive Group			Summary	Add Car Sale	Logout
Welcome back, John Doe					
Summary					
Search			Find		
Make	Model	Available Units	Sold Units	Total Sales \$	Last Purchased At
Land Rover	Defender	1	0	0	
Land Rover	Discovery Sport	1	1	85000.00	27-01-2021
Land Rover	Range Rover	1	0	0	
Lexus	GX	1	0	0	
Lexus	NX	0	1	79000.00	01-01-2025
Lexus	UX	0	1	70000.00	01-01-2023
Mercedes Benz	A Class	1	2	69000.00	01-03-2025
Mercedes Benz	C Class	2	1	72000.00	01-03-2024
Mercedes Benz	E Class	2	1	155000.00	01-10-2024
Toyota	Camry	2	1	37200.00	07-12-2023
Toyota	Corolla	0	1	35000.00	21-09-2024
Toyota	RAV4	1	0	0	
Volkswagen	Golf	1	1	33000.00	06-11-2023
Volkswagen	Passat	1	0	0	
Volkswagen	T Roc	0	0	0	

*Figure 2: Viewing the Car Sales Summary page*

The user can click the Summary button at the navigation to return to the Summary page at any time.

### Finding CarSales

When a user selects a row in the Car Sales Summary Table, they can view detailed records of each sale associated with that summary, as shown in Figure 3 and Figure 4. Each car sale record includes the sale ID, make name, model name, built year, odometer reading, list price, and sale status (represented as a boolean in the IsSold field). For sold cars, additional details such as the sale date, buyer's full name, and salesperson's full name are also displayed.

The Sydney Automotive Group

SummaryAdd Car SaleLogout

Car Sales

Search

Find

ID	Make	Model	Year	Odometer	Price	IsSold	Sale Date	Buyer	Salesperson
23	Volkswagen	Golf	2023	53849	43000.00	False			
6	Volkswagen	Golf	2023	22000	33000.00	True	06-11-2023	Laura Roberts	John Doe

*Figure 3: User Clicks Volkswagen Golf Row*

Users can search the car sales data by entering a keyword in the search field next to the Find button and clicking Find. The search retrieves and displays car sale records containing the keyword in any of the following fields: make name, model name, buyer's full name, or salesperson's full name as shown in Figure 4. The search is **case insensitive**. Searching with a blank or empty keyword field will show all the logged in user's associated car sale records.

Any search results returned must be ordered such that available cars are listed at the top, followed by sold cars ordered by sale date in ascending order. Sold cars are then sorted by make name and model name in ascending order.

Requirements:

- The search results must exclude any sold car where sale dates are older than 3 years (from today's date).
- Sale date should be displayed in the Australian date convention (Date-Month-Year).

The Sydney Automotive Group

Summary

Add Car Sale

Logout

Car Sales

Search

Find

ID	Make	Model	Year	Odometer	Price	IsSold	Sale Date	Buyer	Salesperson
5	Land Rover	Defender	2018	115000	97000.00	False			
16	Land Rover	Discovery Sport	2022	53000	89900.00	False			
14	Land Rover	Range Rover	2017	60000	128000.00	False			
11	Lexus	UX	2023	23000	70000.00	True	01-01-2023	Eva Taylor	Bob Brown
6	Volkswagen	Golf	2023	22000	33000.00	True	06-11-2023	Laura Roberts	John Doe
3	Toyota	Camry	2021	98200	37200.00	True	07-12-2023	Charlie Davis	Bob Brown
4	Toyota	Corolla	2022	65000	35000.00	True	21-09-2024	Grace Thomas	John Doe
7	Lexus	NX	2020	67000	79000.00	True	01-01-2025	Jane Smith	Bob Brown

*Figure 4: Finding Car Sales Records with Search Keyword 'ro'*

### **Adding a CarSale**

Users may also add a new car for sale by clicking on the Add Car Sale tab in the title bar. In the New Car page, they must enter car details including make name, model name, built year, odometer reading, and price, and then click the Add Car Sale button, as shown in Figure 5. A new car is available for sale by default, with no associated buyer, salesperson, or sale date.

- *Make*: A valid make name from the database. The input is case-insensitive.
- *Model*: A valid model name from the database. This input is case-insensitive.
- *Built Year*: The year the car was built.
- *Odometer*: The car's odometer reading. This must be a positive value.
- *Price*: The car's price. This must be a positive value.

The Sydney Automotive Group

Summary Add Car Sale Logout

## New Car

Make: LAND rover

Model: RangE Rover

Built Year: 2022

Odometer: 12

Price: 12345.12

Add Car Sale

*Figure 5: Adding a New Car Sale*

### **Updating a CarSale**

Users can also update a car sale record by modifying the data in the car sale details screen as shown in Figure 6, by clicking on Update Car Sale button. A car sale cannot be recorded with a future sale date (from today's date). You can access this update page by clicking on a car sale record from the car sales page.

- *Buyer*: The user must enter a valid customer ID for the buyer. The input is case-insensitive.
- *Salesperson*: The user must enter the Salesperson's username who handled the sale. The input is case-insensitive.
- *Sale Date*: the date of sale must be entered in the Australian date convention (Date-Month-Year).

The car will be **sold only if all three fields are valid**.

The Sydney Automotive Group

Summary Add Car Sale Logout

## Update Car Sale

ID: 9

Make: Toyota

Model: RAV4

Buyer: C001

Salesperson: JDoe

Sales Date: 01/01/2025

Update Car Sale

*Figure 6 - Updating a Car Sale Record*

### Database Interaction Code

The files that are needed for the Python version of assignment are as follows:

1. **SAGschema.sql**: a file which contains SQL statements you need to run to create and initialise the SAG database, before starting the application.

[https://canvas.sydney.edu.au/files/42344022/download?download\\_frd=1](https://canvas.sydney.edu.au/files/42344022/download?download_frd=1)

2. **Assignment2\_PythonSkeleton.zip**: a zip file encapsulating the Python project for the SAG system

[https://canvas.sydney.edu.au/files/42465790/download?download\\_frd=1](https://canvas.sydney.edu.au/files/42465790/download?download_frd=1)

To inspect the SAG system code, you need to unzip the ZIP archive first, which will create a folder that includes the name **Assignment2\_PythonSkeleton**. If you experience any difficulties installing and exploring the project, ask your tutor or lecturer for assistance.

The skeleton code uses a number of Python modules to implement a simple browser-based GUI for the SAG system. The main modules are the Flask framework for the GUI and the psycopg2 module for the PostgreSQL database access. Similar to tutorial 7, **you will need to install the Psycopg2 module and the Flask module.**

```
pip install psycopg2-binary
pip install flask
```



The skeleton code follows the structure described below:

- The main program starts in the `main.py` file. You need to use the correct username/password details as specified in tutorial 7 and then implement the missing database access functions – including any necessary SQL code statements required – in the data layer `database.py`.
- The presentation layer is done via a simple HTML interface that can be accessed from a web browser. The corresponding page templates are located in the `templates/` subdirectory, their CSS style file in `static/css`.
- The transition between the different GUI pages and the initialisation of the Flask framework is done in the `routes.py` file. It currently just invokes the pages, but there is no further business logic implemented yet.

You can run the code by running “`python main.py`”. This starts a local web server and prints out some debug messages in the terminal; the GUI can then be accessed with any web browser on the same computer via the local URL <http://127.0.0.1:5001/> (If that doesn't work you can also try <http://0.0.0.0:5002/>). Please note that, to terminate the application, you will need to stop the local web server which is running in the background.

## Task 2: Functions Implementation

### Core Functionality

In this assignment, you are provided with a Python skeleton project that must serve as the starting point for your assignment. Your task is to provide a complete implementation for the file `database.py`, as well as make any modifications necessary to the database schema (i.e., `SAGschema.sql`).

Specifically, you need to modify and complete these five functions:

1. `checkLogin` (for login)
2. `getCarSalesSummary` (for viewing car sales summary)
3. `findCarSales` (for finding car sales)
4. `addCarSale` (for adding a car sale)
5. `updateCarSale` (for updating a car sale)

Note that, for each function, the corresponding action and outcome should be **implemented by issuing SQL queries** to the database management system. If you directly output the result set, pre-process, manipulate and/or make changes to the input or output datasets using Python code or additional modules (libraries) i.e. without issuing SQL queries, you are considered as cheating, and you will get penalised heavily and most likely get zero point for the assignment.

**No additional Python modules or libraries should be imported.**

## Marking

This assignment is worth **16%** of your final grade for the unit of study. Your group's submission will be marked according to the attached rubric.

### Group member participation

**If members of your group do not contribute sufficiently, you should alert the unit coordinator as soon as possible.** The course instructor has the discretion to scale the group's mark for each member as follows:

Percentage of contribution	Proportion of final grade received
< 5% contribution	0%
5 - 10% contribution	20%
11 - 15% contribution	40%
16 - 20% contribution	50%
21 - 24% contribution	60%
25 - 28% contribution	80%
29 - 30% contribution	90%
> 30% contribution	100%

Note: The above table assumes that each group will have 3 members, so, on average, around 33% contribution is expected from each member of the group. In special case, if a group has less than 3 members then the contribution percentage will be adjusted accordingly. You must justify your contribution percentage by providing a detailed explanation of your individual contribution on the assignment coversheet mentioned before. You must also regularly record and maintain a diary of your group meetings and discussions on Canvas. Furthermore, we may run random face-to-face interviews to understand and justify your contribution, if needed.

## Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of **16** points.

	Part Marks (0 – 1.5 pts)	Full Marks (2 – 2.5 pts)
<b>Login</b>	Can correctly login the user 'jdoe' and validate the username and password.	All valid users can be logged in successfully, and unsuccessful user logins should be rejected.
<b>View CarSales Summary</b>	Correctly display the car sales summary (see Figure 3).	Correctly display car sales summary in the correct order.

<b>Find CarSales</b>	Correctly lists car sale records for the keyword "ro" (see Figure 5)	Correctly list car sale records for all possible keywords in the correct order.
<b>Add CarSale</b>	Can correctly add a car sale record to the database.	Can correctly add all valid car sale records to the database. Car sale record entered with invalid details should be rejected.
<b>Update CarSale</b>	Can correctly update the status of a car sale record.	Can correctly update details of all car sale records, ensuring the updated details are valid.
<b>Stored Procedure (Function)</b>	A couple of stored procedures (functions) are correctly created in the submitted SQL file.	A couple of stored procedures (functions) are correctly created in the submitted SQL file, and correctly called in two of the five specified functions.

	<b>No Marks (0 pt)</b>	<b>Full Marks (1 pt)</b>
<b>Record Keeping of Group Discussions</b>	One or more issues reported or found with group member contribution, or with maintaining records of group meetings and discussions regularly on Canvas.	No issue reported or found with group member contribution. All group members participate and regularly maintain a diary of group meetings and discussions on Canvas.