

**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) (JDBC). 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 in 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 Java file (`PostgresRepositoryProvider.java`) containing the Java 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.

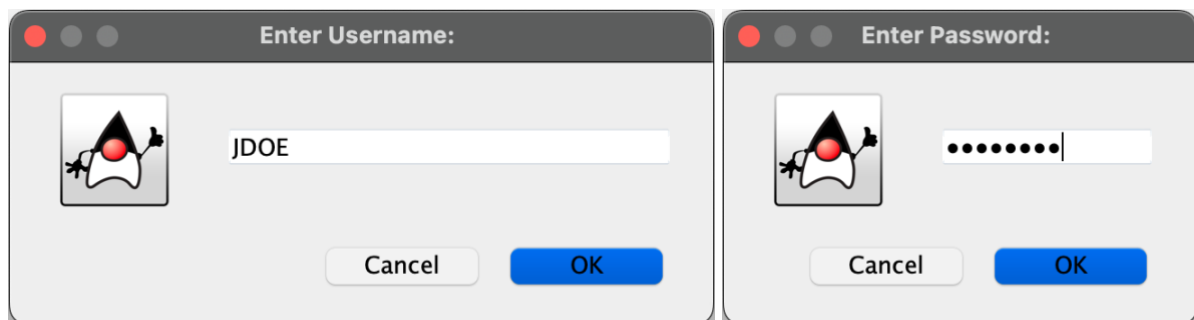
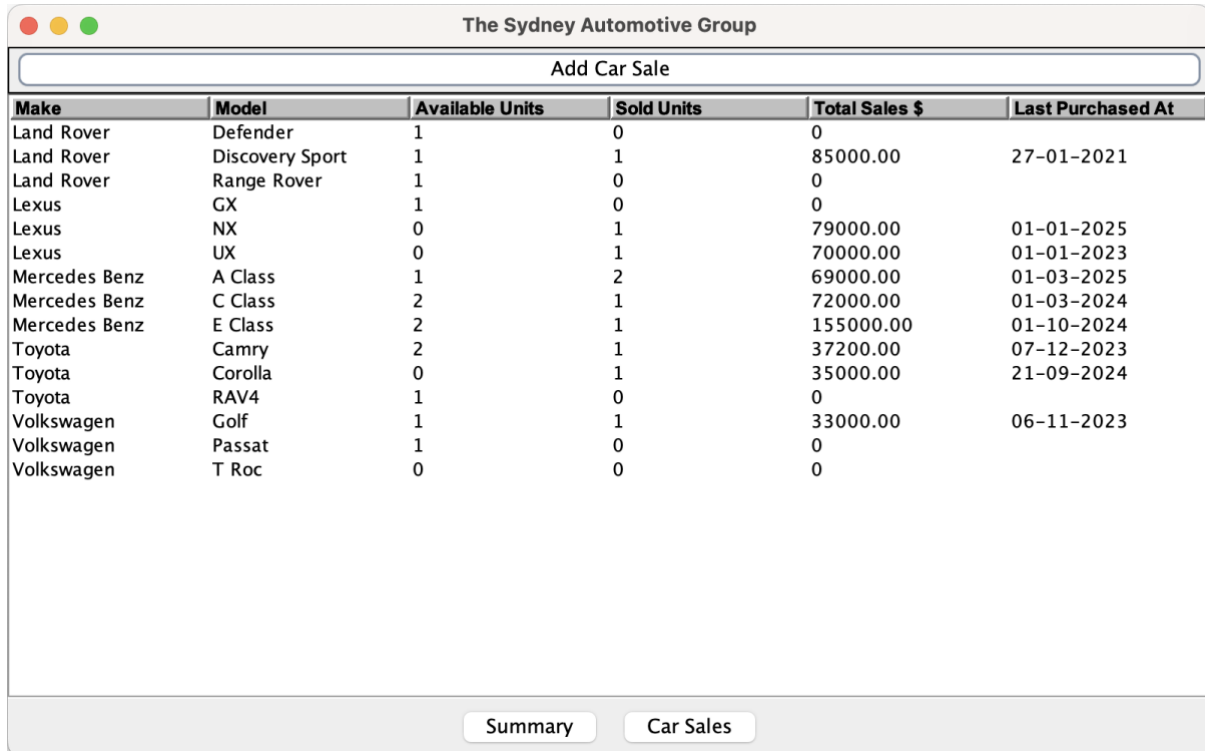


Figure 1: Login

### Viewing CarSales Summary

Once a user has 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 display in the Australian date convention (Date-Month-Year).



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 Summary button can be used to return to the Summary page at any time or to refresh the page after a user has added or updated a car sale record, allowing users to see the most up-to-date information.

### ***Finding Car Sales***

When a user selects a row in the Car Sales Summary Table, they can view detailed records of each sale associated with that summary by clicking the Car Sales button, 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 be displayed.

The Sydney Automotive Group

Add Car Sale

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	

Summary Car Sales

Figure 3: User Clicks Volkswagen Golf Row and Car Sales Records Button

The Sydney Automotive Group

Add Car for Sale

ro  
Find  
23 - Volkswagen Golf  
6 - Volkswagen Golf

CarSale ID: 23  
Make: Volkswagen  
Model: Golf  
BuiltYear: 2023  
Odometer: 53849  
Price: 43000.00  
IsSold: false  
Buyer:   
Salesperson:   
Sale Date:   
Update Car Sale

Summary Car Sales Records

Figure 4: Display of All Sales Records Related to Volkswagen Golf

If the user does not select a row and clicks the Car Sales button, all car sale records associated with the logged-in user are displayed.

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 5. 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 sales 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 screenshot shows a web application titled "The Sydney Automotive Group". At the top, there is a search bar with the text "Add Car Sale". Below this, on the left, is a search interface with a text input containing "ro" and a "Find" button. A dropdown list of search results is visible, showing a list of car models with their corresponding IDs. The selected item is "4 - Toyota Corolla". To the right of the search results, the details for the selected car are displayed in a form. The details include: CarSale ID: 4, Make: Toyota, Model: Corolla, Built Year: 2022, Odometer: 65000, Price: 35000.00, IsSold: true, Buyer: Grace Thomas, Salesperson: John Doe, and Sale Date: 21-09-2024. At the bottom of the details section, there is an "Update Car Sale" button. At the very bottom of the application, there are two buttons: "Summary" and "Car Sales".

ro	Find
5 - Land Rover Defender	
16 - Land Rover Discovery Sport	
14 - Land Rover Range Rover	
11 - Lexus UX	
6 - Volkswagen Golf	
3 - Toyota Camry	
4 - Toyota Corolla	
7 - Lexus NX	

CarSale ID: 4
Make: Toyota
Model: Corolla
Built Year: 2022
Odometer: 65000
Price: 35000.00
IsSold: true
Buyer: Grace Thomas
Salesperson: John Doe
Sale Date: 21-09-2024

Update Car Sale

Summary Car Sales

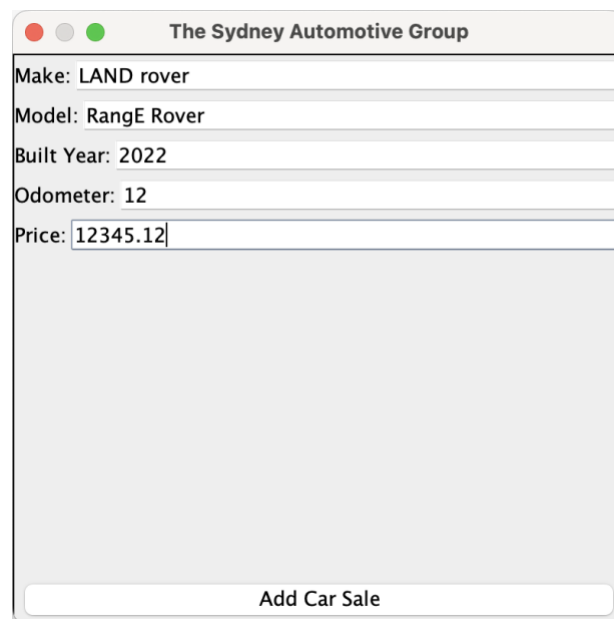
*Figure 5: Finding Car Sales Records with Search Keyword 'ro'*

Selecting a car from the list displays the details, as shown in Figure 4 and Figure 5.

### ***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 6. 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 screenshot shows a window titled "The Sydney Automotive Group". Inside, there is a form with the following fields: "Make: LAND rover", "Model: RangE Rover", "Built Year: 2022", "Odometer: 12", and "Price: 12345.12". At the bottom of the form is a button labeled "Add Car Sale".

*Figure 6: 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 7, and clicking on Update Car Sale button. A car sale cannot be recorded with a future sale date (from today's date).

- *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

Add Car Sale

Find

9 - Toyota RAV4

CarSale ID: 9  
 Make: Toyota  
 Model: RAV4  
 Built Year: 2019  
 Odometer: 92900  
 Price: 48000.00  
 IsSold: false  
 Buyer: C001  
 Salesperson: JDoe  
 Sale Date: 01-01-2025

Update Car Sale

Summary Car Sales

*Figure 7: Updating a Car Sale Record*

The 'Summary' or 'Car Sales' buttons can be used to refresh the car sale details from the database, allowing users to verify whether a car sale record has been added or updated successfully.

### Database Interaction Code

The files that are needed for the Java 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\_JavaSkeleton.zip**: a zip file encapsulating the Java project for the SAG system  
[https://canvas.sydney.edu.au/files/42465787/download?download\\_frd=1](https://canvas.sydney.edu.au/files/42465787/download?download_frd=1)

*This skeleton code is compatible with most versions of Java. If you're uncertain which version to install, we recommend using Java 8 for maximum compatibility and ease of setup.*

To look through the SAG system code, you'll need to import the Eclipse project archive file into Eclipse.

Please begin by trying to import the project into Eclipse using the steps below:

- 1) Right Click the zip file containing the assignment 2 archive > 7zip > Extract Here.
- 2) Start Eclipse and choose a new location for your workspace in a new folder.
- 3) Click on File > Import > General > Existing Projects into Workspace > Next.



- 4) Browse to the folder you extracted zip file to, and then click Select Folder.
- 5) Ensure the Assignment 2 Project is ticked in the Projects box and click Finish.

If you experience any difficulties importing the eclipse project, ask your tutor or lecturer for assistance.

Once you import the project in Eclipse, you should notice that there are 3 main packages in the solution: **Presentation**, **Business**, and **Data**. The UI (in the Presentation package) is currently coded to invoke logic in the business layer (eg: see `CarSaleProvider` class in the Business package), which in turn delegates responsibility for interacting with the database to an appropriate repository in the data layer (eg: see `PostgresRepositoryProvider` in the Data package). Notice that separating concerns in this way makes it easier to write a different `RepositoryProvider` in the data layer (eg: we could write a `MySQLRepositoryProvider`) if we wanted to change our database management system. In this assignment, you will mainly be working on writing the appropriate queries and SQL commands to fulfil the SAG system functionality described in Task 1; where these SQL commands and queries should be written in the `PostgresRepositoryProvider`. You should use the correct username/password details as specified in tutorial 7. The application's main method can be found in the **`CarSaleTrackerFrame`** class. You can run this main method to run the entire application by right clicking the **`CarSaleTrackerFrame`** file on the Project Explorer > Run As > Java Application.

## Task 2: Functions Implementation

### *Core Functionality*

In this assignment, you are provided with a Java skeleton project that must serve as the starting point for your assignment. Your task is to provide a complete implementation for the file **`PostgresRepositoryProvider.java`**, 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 Java code or additional packages (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 Java packages 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.