**COIT11237 Database Design & Implementation**

**Assessment details for ALL students**

# QProperty database

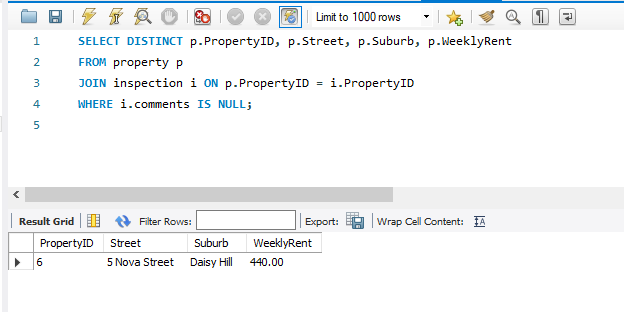
**QProperty** database is used to maintain the data related to property management by a property management agency. The main contents that the database includes the data of managing agents, the listed properties, prospective customers, inspections, repair jobs related to the managed properties, and the contracted tradesmen who perform repair jobs or provide services. The interested customers can inspect the listed properties many times. The database also records the related details about the property repair jobs and services as common scenarios in our daily life. For the SQL query exercise purpose, we only are interested in the above contents which do not include the leasing contract etc. The **QProperty** database contains the following tables:

* **customer** table that has the list of clients’ details.
* **agent** table that has the list of the property agents’ details.
* **property** table that contains the property details.
* **inspection** table that contains the details that customers have inspected properties.
* **repairjob** table that contains the relevant details of the property repair works that were carried out by tradesmen.
* **tradesman** table that contains the details of tradesmen.

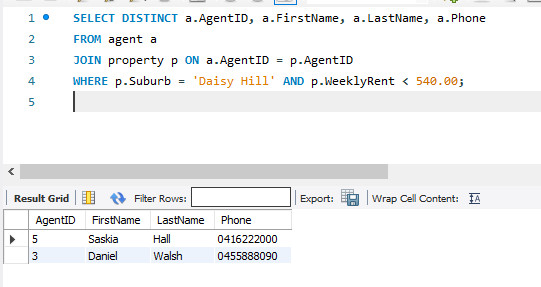
# Part A – SQL Queries

**Questions**

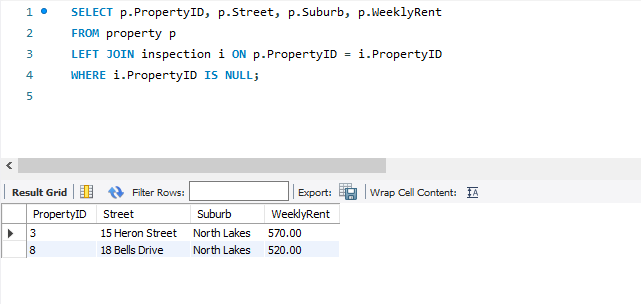
1. **For inspections, show details of properties that have been inspected but did not have comments.**



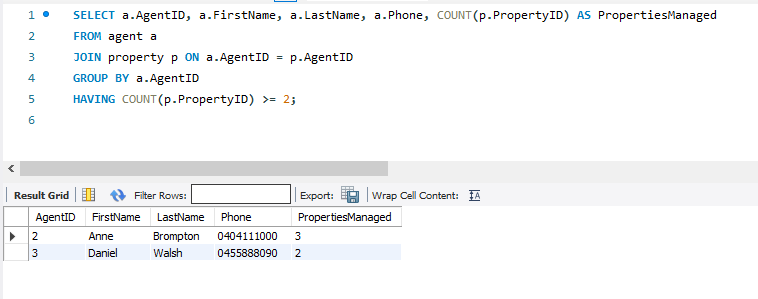
1. **Show the details of agent(s) who managed the properties in the suburb – *Daisy Hill* with the weekly rent less than $540.**



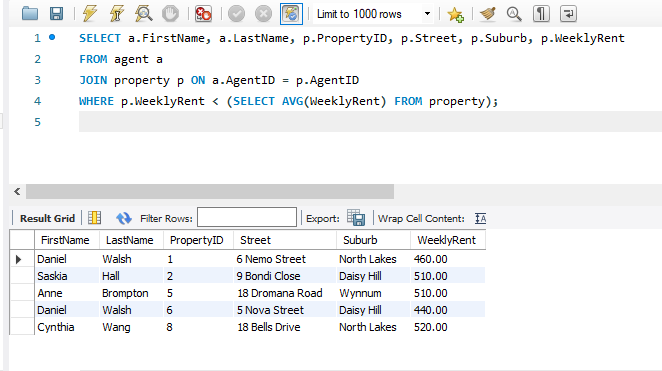
1. **Show properties that have not been inspected.**



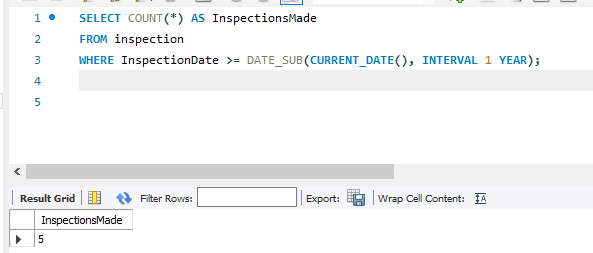
1. **Which agent(s) manages two or more properties? Include the agent’s ID, name, phone, and the number of properties managed.**



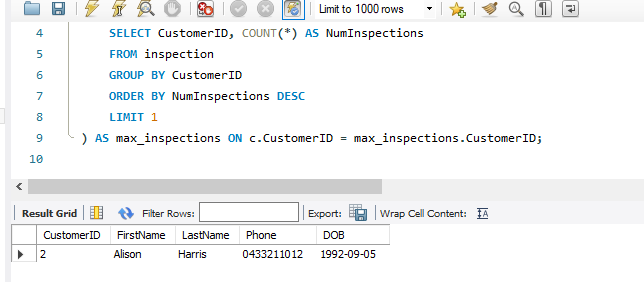
1. **Show the names of agents and their managed properties for which the weekly rents are lower than the average weekly rent recorded in the system.**



1. **In the past year, how many inspections have been made?**

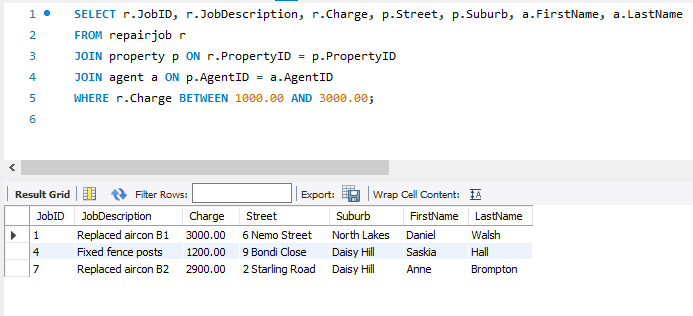


1. **Find out which customer(s) who has/have the highest number of inspections. Display their ID, name, phone, and date of birth.**

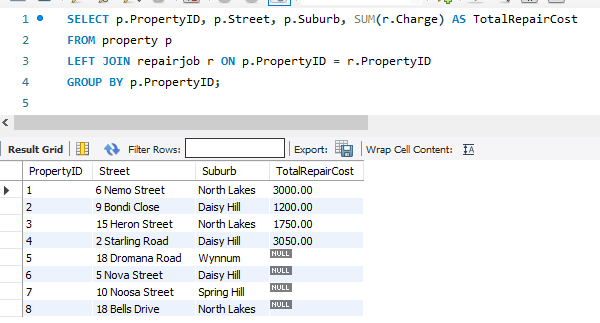


1. **List the details of those repair jobs that have been completed with the charge between $1000 and**

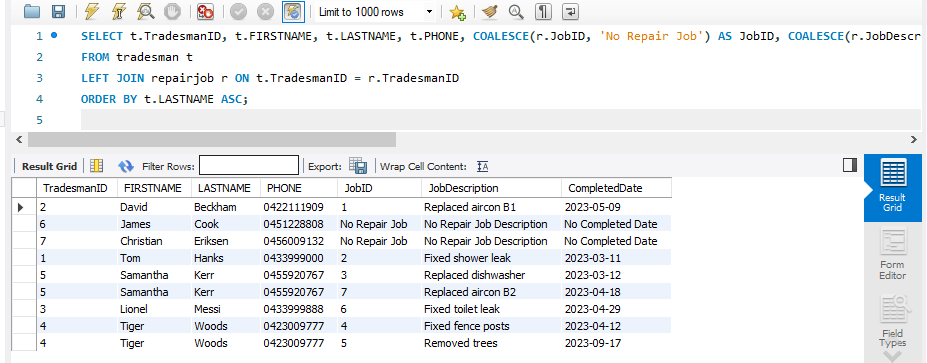
**$3000. These details should include the job number, job description, charge, property address, and corresponding managing agent name. Hint: understand how the BETWEEN operator works.**



1. **List the properties that have repair jobs and what the total cost is for each property.**



1. **Show the ID, name, and phone number of tradesperson along with the repair jobs which have been completed. Also include the tradesperson who have not done any repair jobs. Display the results in ascending order of the tradesperson’s surname.**



# Part B – Screenshots and Query Analysis

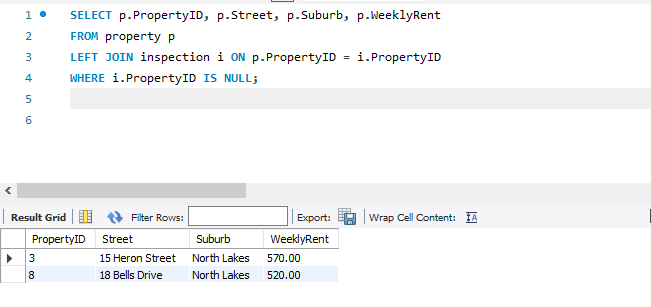
# Questions

1. Explain the methodology of the SQL Query that you used to answer the question 3 in Part A. Explain another methodology that can be used to produce the same output. Write both SQL queries as part of your answer.

A LEFT JOIN between the property table and the inspection table can be utilized for this. We can include all properties from the property table by left joining these tables, Nevertheless of whether they have corresponding records in the inspection table or not. After that we filter the results and only include rows where there is no match in the inspection table that is where inspection.PropertId is NULL. This verifies that properties are retrieved that have not been inspected.

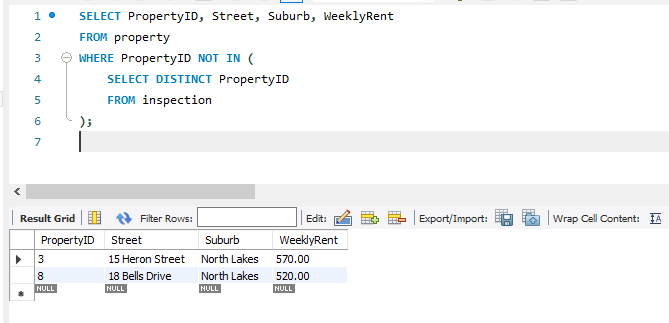
Alternative methodology 1:

Using LEFT JOIN and IS NULL:



Alternative methodology 2:

Using subquery with NOT IN operator:



1. Modification of Data: Write a SQL statement to update the Property ID=1 weekly rent with an increase of $10. (You can name this SQL statement script as **B3.sql**)

