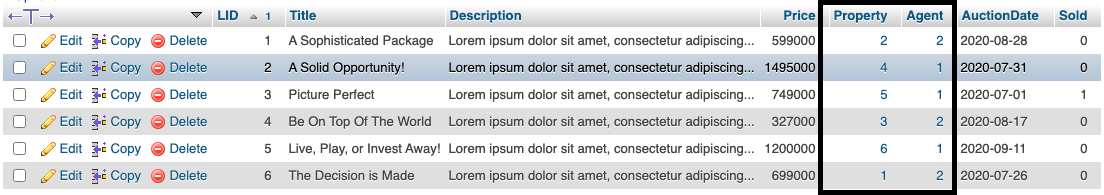
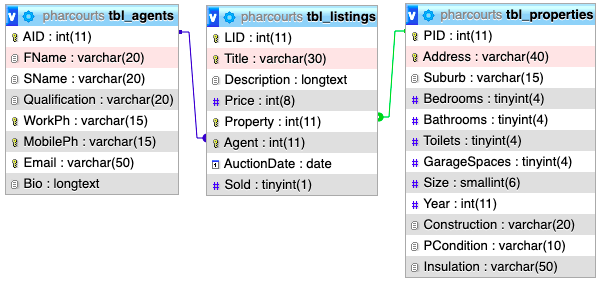
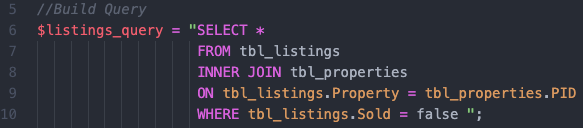
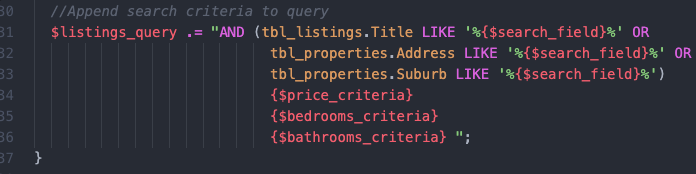
# Linking Data in Related Tables using Queries or Keys

**Database**

My Listings table has two instances of linked data. The first link is to a record in the tbl\_properties table which contains all of the information for the property the listing is selling and the second link is to a record in the tbl\_agents which contains the information for the agent selling the property. These are both One to Many relationships and the relationship allows the data to be stored separately to make it easier to manipulate but they can still be referenced all together to get all necessary information

**Search – Query (SQL)**

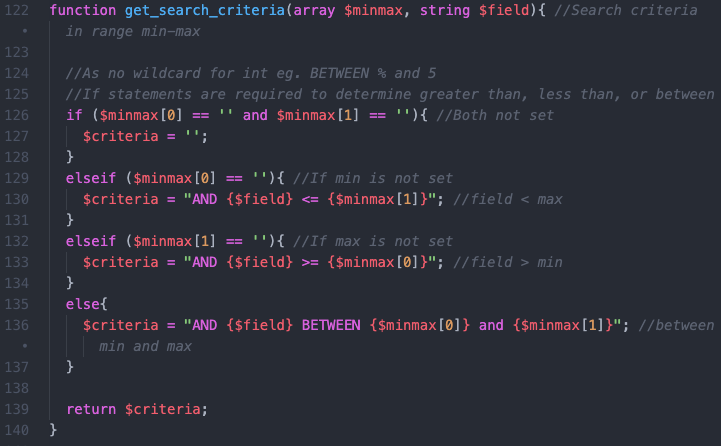
For my search query, since the search criteria allows the user to filter by fields from tbl\_listings and tbl\_properties I need to use an INNER JOIN to link the data in order to successfully filter by fields in either table.

**Listing Profile – Query (PHP)**

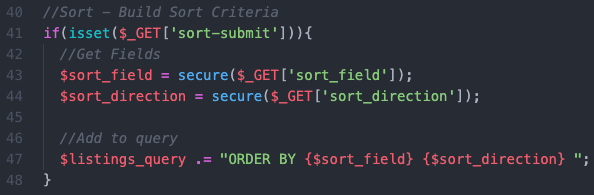
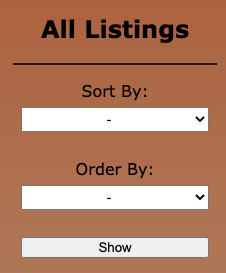
In order to display listing and property information I require another join in order to combine the two records and get the appropriate information. Since I have a reference to the associated property ID in a field in the listing I can get the ID from tbl\_listings.Property and query it from tbl\_properties and then combine the arrays from the results into one big array with all of the associated listing and property fields. This is essentially a manual/PHP version of the SQL INNER JOIN function. I query the listing then I take the property ID and query that from properties, then merge the arrays to create a ‘INNER JOIN’ or array with all of the associated information. There was originally a compatibility problem using INNER JOIN in a single query for this instance however I had to restructure because of another problem so now an INNER JOIN would be more efficient and effective in this instance, however I am leaving the PHP level inner join to show another possible method of linking data from tables.

# Writing Custom Queries to Filter and/or Sort Data

**Search Listings**

For my search listings query it initially queries all listings with associated properties that haven’t been sold. However, depending on the search criteria from the search bar and max-min field the WHERE condition expands. It checks for what has been put into the search bar from listings.Title, properties.Address and properties.Suburb. This criteria is then added to the query. If it is empty all of these fields will query ‘%%’ which means any value. Then the query can be expanded on futher with the additional min-max values. Take price for example, if MIN and MAX are ANY then no criteria is added to the query. If a MIN is set and MAX is ANY then it looks for values greater than MIN and the inverse if MAX is set and MIN is ANY. Then if they are both set it looks for values between MIN and MAX. This same operation is applied to all three additional fields (price, bedrooms, bathrooms) to build a custom query each time a search is applied.

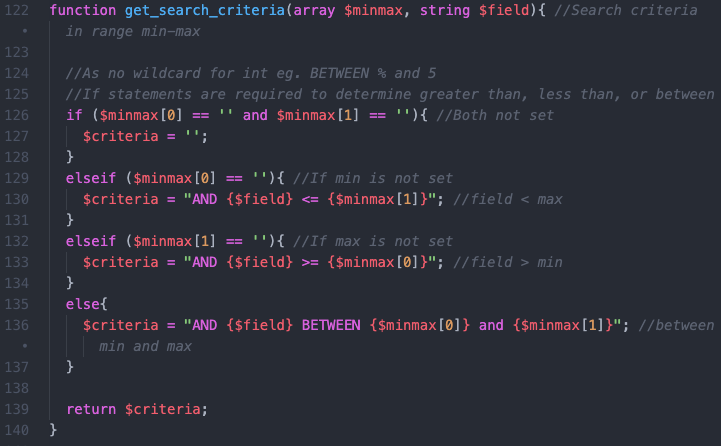
**Sort Listings**

The sort listings form sort all listings by the submitted criteria. It uses the same initial query as the search does however instead of appending a WHERE condition it appends an ORDER BY condition to the query which will get all listings and display them sorted.

# Using Logical, Mathematical, and/or Wildcard Operators

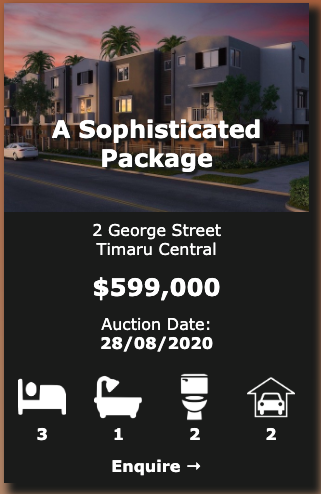
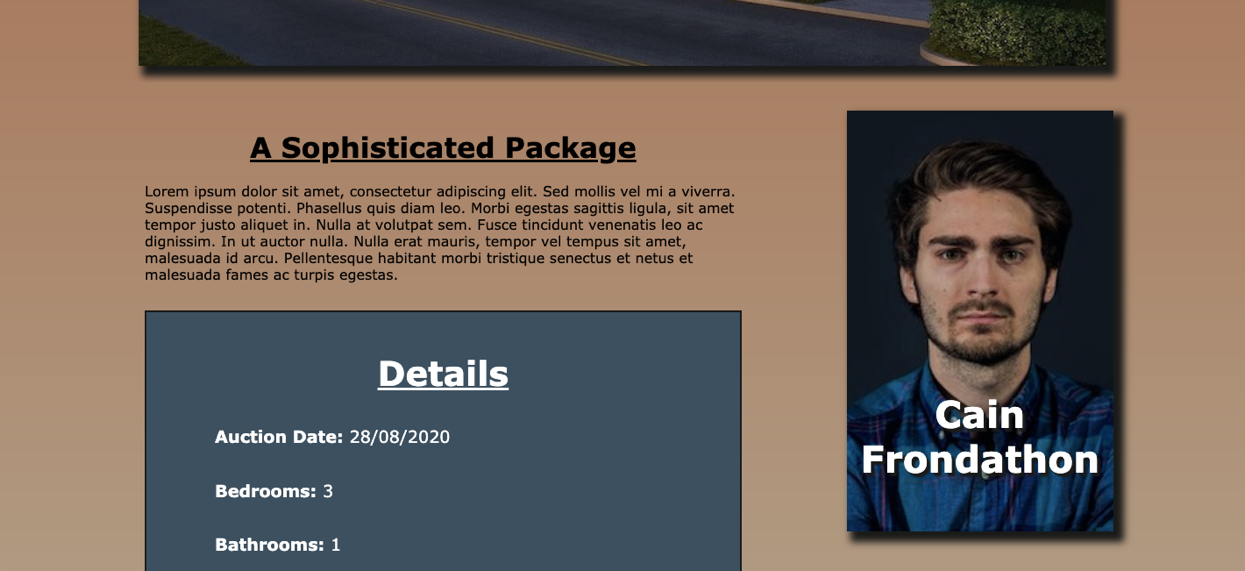
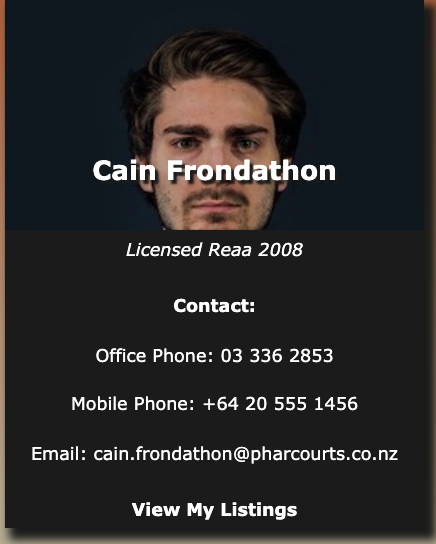
**Search Listings**

I have used operators that fit criteria in almost all of my queries that involve a WHERE condition however all of these operators are present in my search query. I have used Logical operators in my additional options if a minimum and maximum value is given then in order to get all results in that range the operator BETWEEN was used. Just above this I used the Mathematical operators >= and <= when only a minimum or maximum number but not both is defined. This allows me to get all listings with an amount below or above the maximum and minimum. And I have used Wildcard operators in my search bar criteria where I am checking if field is LIKE ‘%searched item%’ which is saying if a field contains a value with any combination of characters before, then the searched item, then any combination of characters after. This means if I search ‘central’ it will get all records where the field contains the string of letters ‘central’ either at the start, end, or somewhere in the middle.



# Customising Presentation of the Data

**Profiles**

For both listings and agents I have ‘cards’ that a small amount of their information is displayed on and then a full ‘profile’ page that displays all of the information for the listing. These profiles have custom layouts and formatting to present the data in an aesthetic and ease to read way. In addition after a user has submitted or edited a record using a form on the site it redirects to the profile for the record to display the new/updated information.

**Featured**

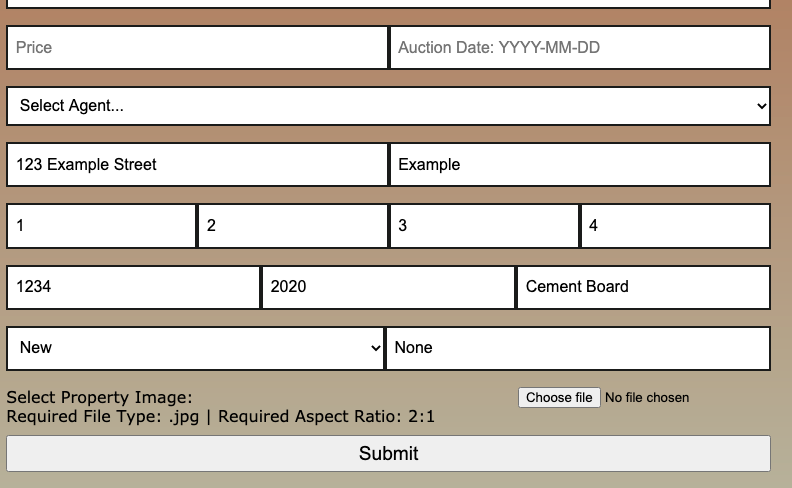
I also have a few properties which I have queried that display on the home page as ‘featured’ properties which have the property image and then the most important information, encouraging a user to click on it and navigate to the profile to learn more.

# Using Custom Forms to Add User Input to the Database

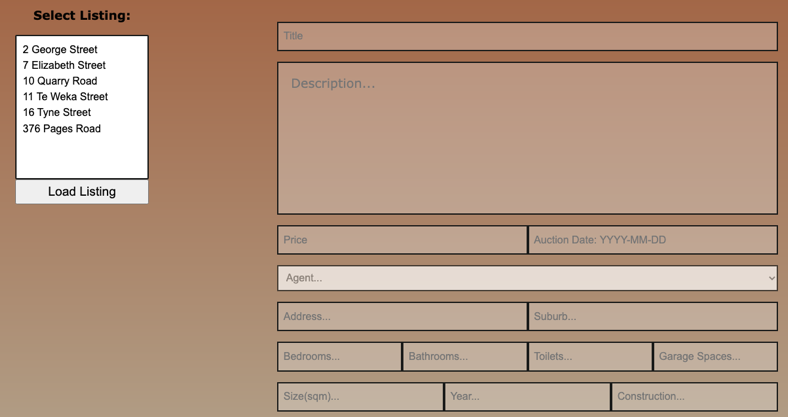
**Add/Edit Listings**

Due to my listings having an associated property and properties already existing the design for this form was quite complicated in comparison to my other ones. The basic layout of it has all of the fields for listing and properties. It also has a hidden field which contains a property ID. The reason for this is the user, using another form in the sidebar, can load in an existing property that isn’t currently assigned to a listing so they don’t have to create a brand new property for something that already exists. The problem that occurred with this however was that if an existing property is used then I still want to add a new listing but the property information needs to switch from being appended to being updated. My solution to this was to set the hidden field to the ID of the property otherwise have it be null. Then I can have checks in my form and validation so that if that is set it ‘autofills’ the values of the form fields with the property information and then when it is submitted I can check whether it is a new property or existing one and adjust the query accordingly. If an existing property is used then the image field is also no longer required as if the property exists an image for it must also exist as you cannot have one without the other. For my edit listings form the form display is identical to the add form however the sidebar contains a form with all listings a ‘load listings’ button. The whole form is disabled until a listing is loaded and then the form is ‘autofilled’ with the listing information. Then when it is submitted it simply updates the existing listing and property, or deletes it if the delete button is pressed. Due to the relationship between listing and property all associated listings must be deleted before a property is removed otherwise it will cause an error since there will be a reference to a non-existent property.

A screenshot of a cell phone

Description automatically generatedAdd Form:

Edit Form:



**Add/Edit Agents**

The add/edit agents forms work almost identically to the listings forms. The add agent is much simpler it simply provides all of the necessary fields to be filled out then adds the new agent with the user submitted data (assuming it is valid). And the edit agents form has a side bar with all of the agents and the same form initially disabled. Then when an agent is loaded into the form it ‘autofills’ their information and updates/deletes the agent information. As the agents aren’t dependant on the relationship the same way listings and properties are these were much simpler forms to create.

Add Agent:

A screenshot of a cell phone

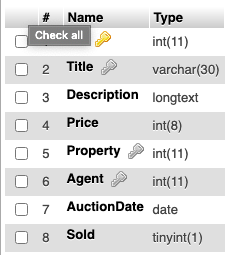
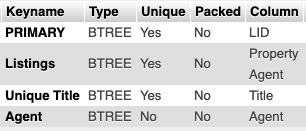
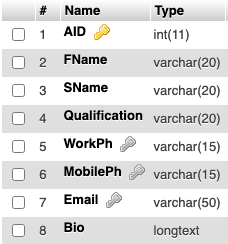
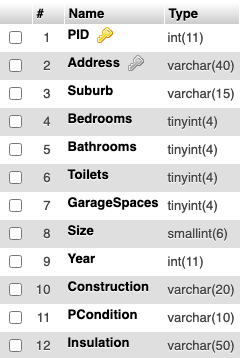
Description automatically generatedA screenshot of a cell phone

Description automatically generatedEdit Agent:

# Setting Validation Rules for Data Entry

**SQL Level**

I have set types for each of my fields so that they have to be a specific data type and length in order to be submitted. For any string values I have used VarChar datatypes and set a maximum length so data that is longer than it should be for that field is rejected and won’t submit to the database. For integers I have made use of TinyInt, SmallInt and Int to control the range of integers expected in the integer field (since there is no LEN for integers). For some fields I have also added unique keys to ensure there isn’t duplicate data. I have used this on the contact information in tbl\_agents (MobilePh, WorkPh, and Email) as well as tbl\_properties.Address and tbl\_listings.Title since there should be no duplicates of this information.

Properties Table: Listings Table: Agents Table:

**HTML Level**

At the HTML level I have applied much finer input validation through the use of input attributes and regular expressions. Almost all fields have a required attribute which means it must contain content in order for the form to be submitted so it cannot submit blank data. Some fields don’t require this such as Insulation which has a condition to default to ‘None’ if blank and the image isn’t required when updating an existing record (edit forms and add listing with existing property) as it already exists and uploading an image will replace the existing one. On a large number of my fields I have also applied regular expression patterns. Regular Expressions (regex) require the content of the field to match a set pattern in order for it to be submitted otherwise it considered the data invalid and alerts the user. Regex can be used very widely such as to check that an email is actually an email or a phone number is only numbers and long enough to be a valid phone number since they have common patterns all numbers adhere to. The inclusion of this criteria significantly restricts the invalid data a user an input into my database and helps to ensure both the consistency of my records and the validity of my data. I have used regex on the following fields:

Agent Forms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Description** | **Regex** | **Valid Example** | **Invalid Example** |
| Mobile Phone | Check whole field for valid phone number | ^[+]\*[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s\./0-9]\*$ | +64 123 1234 | My Phone Number |
| Work Phone | Check whole field for two digits space three digits space four digits | ^\d{2}[\s|-]\d{3}[\s|-]\d{4}$ | 03 123 4567 | +64 123 1234 |

Listing Forms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Description** | **Regex** | **Valid Example** | **Invalid Example** |
| Price | Only numbers | ^[1-9]\d\*$ | 2134567 | 12e456 |
| Auction Date | Must Match YYYY-MM-DD and month/day | ^\d{4}-(0[1-9]|1[0-2])-([0-3][0-9])$ | 2020-12-27 | 2020/13/40 |
| Address | Must be number followed by minimum 2 words | ^\d+(\s[A-z]+){2,} | 123 Main Street | 123 Main |
| Suburb | Must be 1-3 words | ^(\w+\s?){1,3} | Redwood | This is my suburb |
| Bedrooms, Bathrooms, Toilets, Garage Spaces | Must be number 0-99 | ^([0-9]|[1-8][0-9]|9[0-9])$ | 12 | 101 |
| Size | Must be in range 1-32767 | ^([1-9]|[1-8][0-9]|9[0-9]|[1-8][0-9]{2}|9[0-8][0-9]|99[0-9]|[1-8][0-9]{3}|9[0-8][0-9]{2}|99[0-8][0-9]|999[0-9]|[12][0-9]{4}|3[01][0-9]{3}|32[0-6][0-9]{2}|327[0-5][0-9]|3276[0-7])$ | 362 | 0 |
| Year | Must be 4 digit number | ^\d{4}$ | 2020 | 192 |