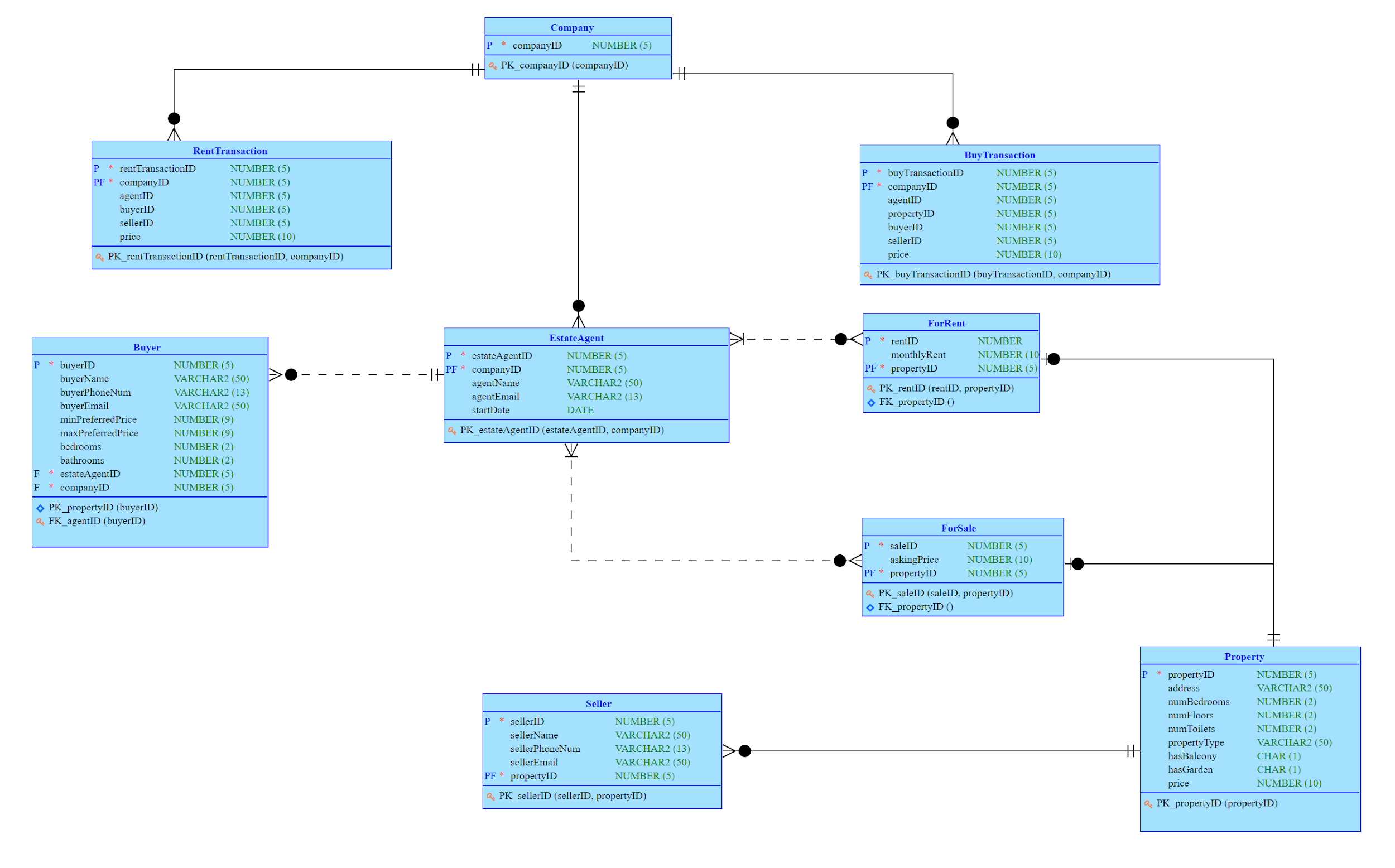
Database CA Case Study - Real Estate

**Michael Lenghel C16434974**

**Eamonn Keogh C16757629**

**Povilas Kubilius C16370803**

# ERD Diagram of Real Estate Company



For the real estate agency we decided to separate to include all sold properties in buy transactions and all properties put up for rent by the company in rent transactions. At the moment there is only one company (or branch) but this allows for expansions for multiple companies with different transactions.

The real estate agents are directly connected to the buyer and act as a link to the property and seller. Property links to a “ForSale” and “ForRent” table to designate whether or not a property is fo sale or for rent. This was made as different tables with a one-to-one link in order to show if the attribute designating it’s cost was one time ordeal, or monthly payments, as otherwise it could be confusing. It also makes queries clearer when shifting through the tables as you always are aware if a particular property is for sale or for rent, without needing to always check an attribute.

The buyer themselves is held in the “estateAgent” table and the seller is held in the “property“ table.

A company has many rent transactions as well as buy transactions and real estate agents and to model this we made a one-to-many relation to the company and these tables. An estate agent has many buyers, but a buyer can have only one estate agent from that company. An estate agent has many properties (which can be rented or selling) and likewise, a property can have many or one sellers.

In designing our PL/SQL, we created a way to dynamically query the database based on user input. Based on the specific role of the user, they needed to provide information in order to provide specific functionality.

For example, a buyer role could query the database as a way to search for properties to buy. The buyer would input the maximum price they are willing to buy the property for, and number of preferred bedrooms, and the PL/SQL would query the database of properties and return properties that match the buyer's requirements, displaying the available properties and some details about the property.

For the role of the seller in PL/SQL, new properties as well as sellers can be inserted into the relational database and all the buyers that can afford the seller's property is displayed to the seller. Error checking included ensuring that the budget entered was not negative, as well as handling a number of predefined exceptions, such as invalid data and incorrect data.

For the role of the estate agent PL/SQL, properties that have been rented and are related to the estate agent that was reponsible for this transaction should then be removed from the market “ForRent” and inserted into the “RentTransaction” to keep the state of the database consistent. Meaning that, only the available properties for rent after this transaction should be listed on the market i.e. the “ForRent” table. Error checking was done to ensure that if there were no properties available for rent, that no such data existed (based on the agent input), any others found that didnt match threw an exception and rolled back to the previous state of the database. This leaves the database in a consistent state before the transaction was fully processed by the esate agent.

In designing our queries we tried to show the flexibility within our ERD design through using a number of different techniques, such as minus, difference, anti-joins, correlated sub-joins etc. within as many different roles within the table as possible. On top of that it provides useful information we can use, and insights on the state of the company as a whole.