Query Spec and Interface Spec

Group 1

Our query language is used to query a database of cars at a car dealership. The language uses the keyword “get” to specify which Car attributes to retrieve from the datastore. When retrieving multiple attributes, the user can separate attributes with “, ” or use “get *” to retrieve all attributes. After specifying the return from the query, the user can use the “where” keyword to narrow down the results to identify specific Cars and their attributes. This language allows the user to make comparisons between a Car’s attributes and specified values by using =, >, >=, <, and <=. For example, “get model where make = “Jeep” will retrieve the model of all Cars in the datastore where the make of the car is Jeep (case sensitive). Multiple of these comparisons can be made in a single query. For example, if you want to find the make, model, and msrp of all blue cars that are currently in stock at the dealership, you can run the query “get make, model, msrp where color = “blue” and quantity >= 1” to display cars that fit these requirements.

Keywords

- get
- ,
- where
- and
- =
- >
- >=
- <
- <=
- *
- make
- model
- color
- msrp
- mpg
- horsepower
- quantity

Example Queries

get model where make = “Jeep”
get make, model where msrp <= 40000 and color = “white”
get mpg where model = “Audi” and mpg > 25
get make, model, msrp where color = “blue” and quantity >= 1

Admin Functions

- load_data(filename) - Load data from JSON file into datastore. This will use the necessary functions in the Firebase connection and authentication file.
    - Parameters:
        - filename: the name of the JSON file containing the data to load
    - Return: No return

Parser Functions

- get_input() - prompts the user to input a query, sends input to the parse function
    - Parameters: None
    - Return: return
- parse(query) - Split the query string into arguments as required by the query engine.
    - Parameters:
        - Query: the input string to be split into arguments
    - Return: A list of instructions
- query(arg_list) - queries the firebase datastore with the list of arguments generated. This will use the necessary functions in the Firebase connection and authentication file.
    - Parameters:
        - Parameters: the parameters to the query (ie. color = “white”)
    - Return:
        - The queried data in dataframe form (can be nothing)
- display_result(df, var_list)
    - Parameters:
        - df: the data frame to display
        - var_list: list containing car features to be displayed in the output
    - Return: No return

Functional Interface

- def __init__ (self, connection)
    - Returns initialized firebase connection
- def query_by_make(self, make)
    - Returns query based on make
- def get_by_uuid(self, uuid)
    - Returns query by uuid