2. Data Structures in Redis

To reimplement the “GhostKitchen” project using Redis, we created the following data structures by parsing the respective JSON files:

DS	**Key**	**Value examples**[1]	**Based on Json file**
set	brands	[1, 2, 3, 4, 5, 6]	brands

hash	brand:${brand_id}	{
		“brand_name”: ${brand_name},
		“brand_id”: ${brand_id}
		}
			brands

string	mealCount	Last meal number	
set	brand:${brand_id}:meals	[15, 19, 20]	brands

hash	brand:${brand_id}:meal:${meal_id}	{
		“brand_id”: ${brand_id}
		“brand_name”: ${brand_name},
		“meal_id”: ${meal_id},
		“meal_name”: ${meal_name},
		“meal_desc”: ${meal_desc},
		“calories”: ${calories},
		“price”: ${price}
		}
			brands

set	locations	[1, 2, 3, 4, 5]	locations

hash	location:${location_id}	{
		“id”: ${location_id},
		“address”: ${address},
		“state”: ${state},
		“phone_number”: ${phone_number}
		}
			locations

set	pickup_types	[1, 2, 3, 4]	pickups

hash	pickup_type:${pickup_type_id}	{
		“id”: ${pickup_type_id},
		“type”: ${pickup_type}
		}
			pickups

string	orderCount	Last order number	orders

[1] ${} means that appropriate value will be inserted

set	orders:current_orders	[384, 385, 386 …] only those where pickup_time is null[2]	orders

hash	orders:current_order:${order_number}	{
		"id": ${orderCount},
		“customer_id”: ${customer_id},
		"location_id", ${location_id},
		“location_address”: ${location_address},
		“location_state”: ${location_state},
		“location_phone”: ${location_phone},
		"meal_id", ${meal_id},
		"meal_name", ${meal_name},
		"meal_desc", ${meal_desc},
		"meal_price", ${meal_price},
		"order_time": ${order_time},
		“pickup_id”: ${pickup_id},
		“pickup_type”: ${pickup_type},
		“pickup_time”: ${pickup_time},
		“order_quantity”: ${order_quantity}
		}
			orders

set	orders:customer:${customer_id}:current_orders	[501, 502, 503] only those where pickup_time is null	orders

hash	orders:customer:${customer_id}:current_order:${order_id}	{
		"id": ${orderCount},
		“customer_id”: ${customer_id},
		"location_id", ${location_id},
		“location_address”: ${location_address},
		“location_state”: ${location_state},
		“location_phone”: ${location_phone},
		"meal_id", ${meal_id},
		"meal_name", ${meal_name},
		"meal_desc", ${meal_desc},
		"meal_price", ${meal_price},
		"order_time": ${order_time},
		“pickup_id”: ${pickup_id},
		“pickup_type”: ${pickup_type},
		“pickup_time”: ${pickup_time},
		“order_quantity”: ${order_quantity}
		}
			orders

set	customers	[1, 2, 3, 4 …]	customers

[2] Because you can’t filter in Redis, the filter on pickup_time was done in Javascript (when running prepopulate script).

hash	customer:${customer_id}	{
		"id": ${orderCount},
		“customer_id”: ${customer_id},
		“first_name”: ${first_name},
		“last_name”: ${last_name},
		“address”: ${address},
		“phone_number”:
		${phone_number},
		“age”: ${age}
		}
			customers