Following are the list of entities along with their attributes that we need for our use-cases:

**User**

This denotes the user object who logins and updates their meal plan/workout plan and the progress.

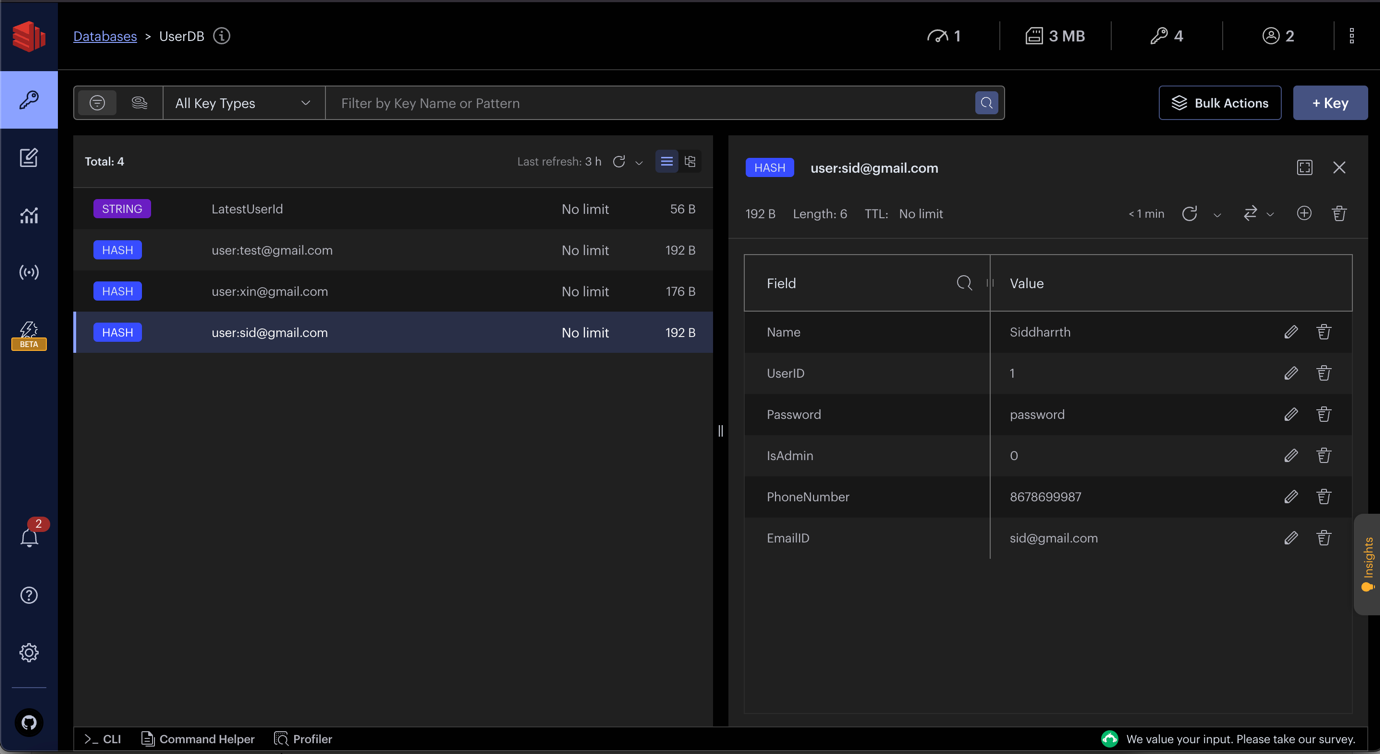
**Schema is as follows:**

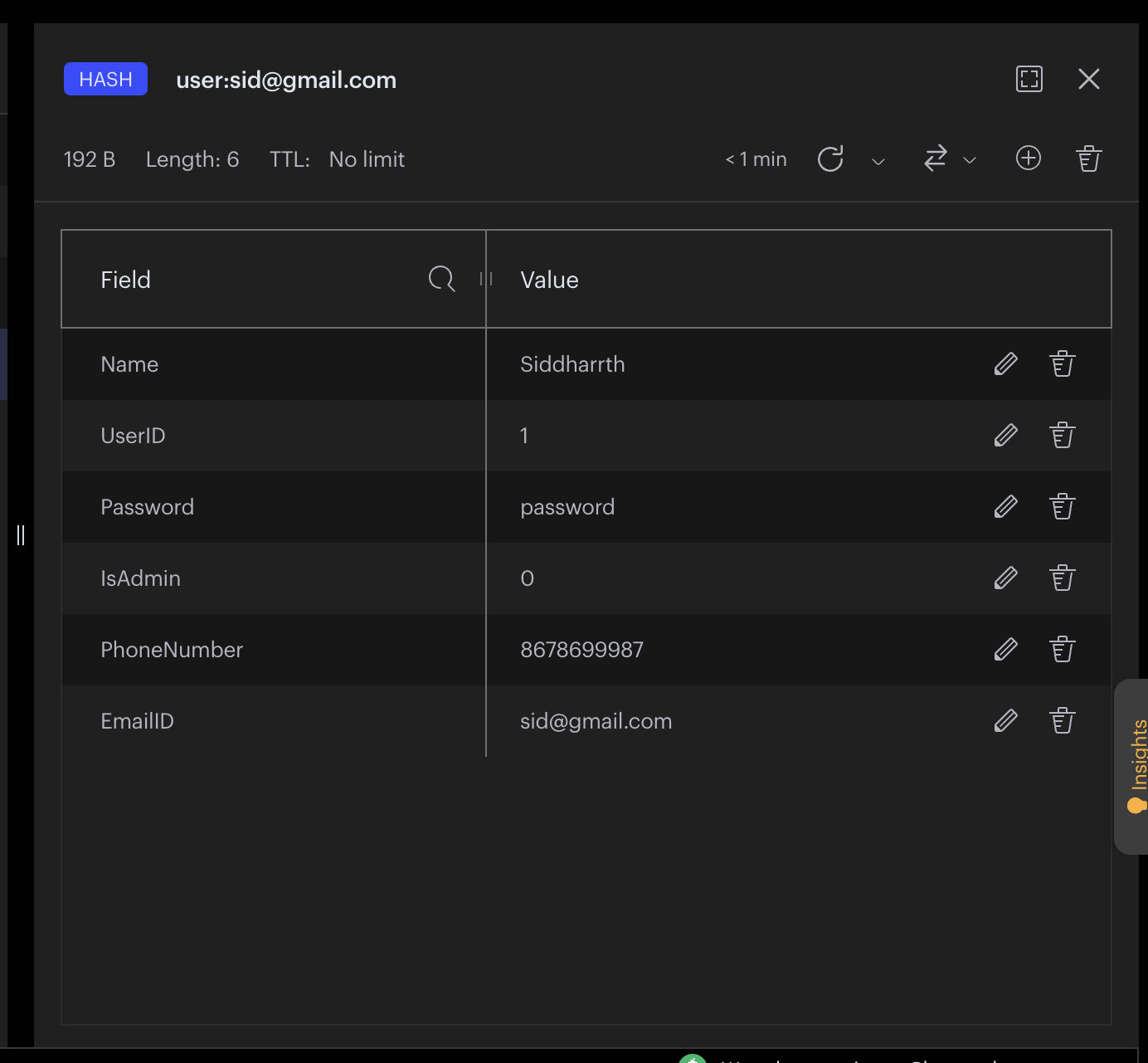
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier for the user record |
| EmailID | String | Email ID with which the user logs in. This must be unqiue |
| Password | String | Password for logging in |
| Name | String | Name of user |
| PhoneNumber | String | Phone number of user |
| IsAdmin | Int | A flag to indicate if the user is admin. Value is 1 for admin user and 0 for non-admin user. Admin users can add new workouts and meals. |

Redis is used as the Database to store User records.

The key would be format = “user:<email ID>”, and type of key is “Hash key”. EmailID and UserID are unique for a given user.

Below is the snapshot of a user record in Redis database.





**WorkoutPlan**

This denotes the object for the workout plan which is created by the user, and this object is used to identify the list of workouts under that workout plan.

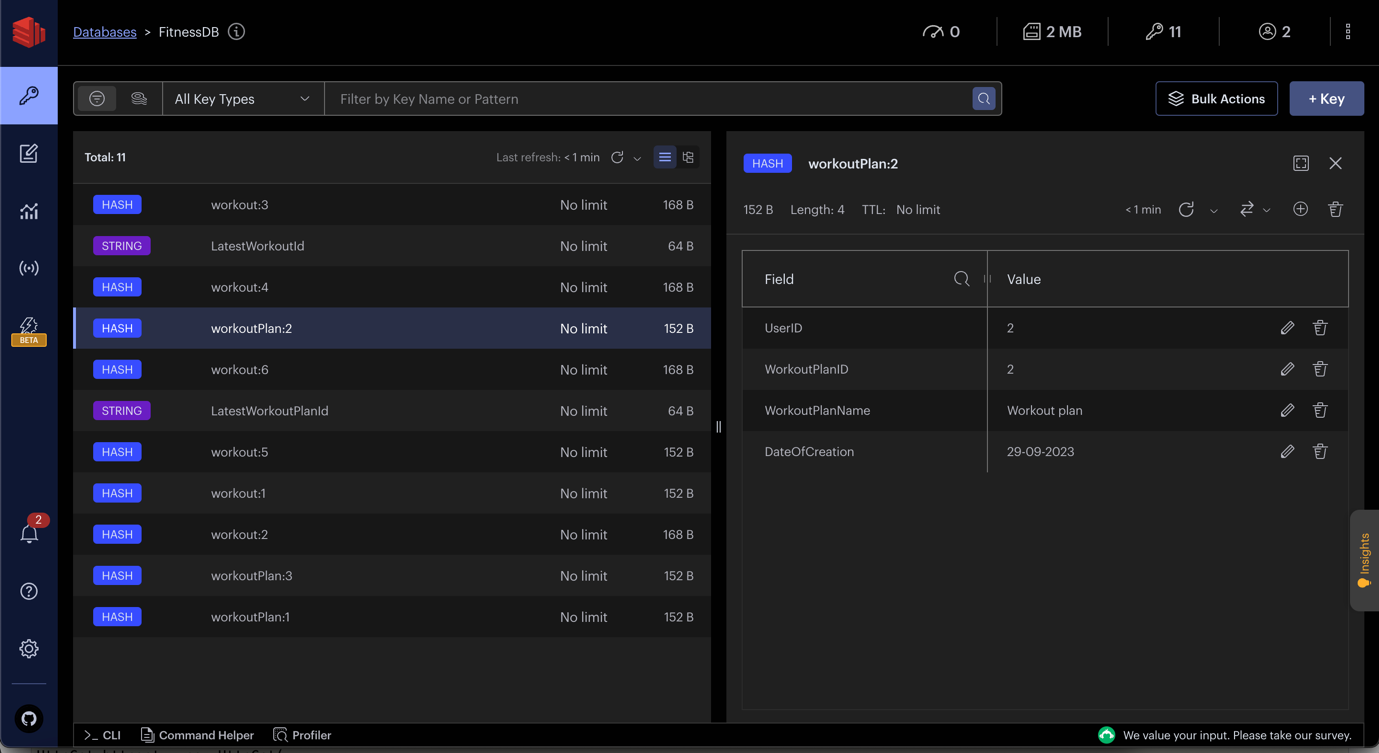
**Schema is as follows:**

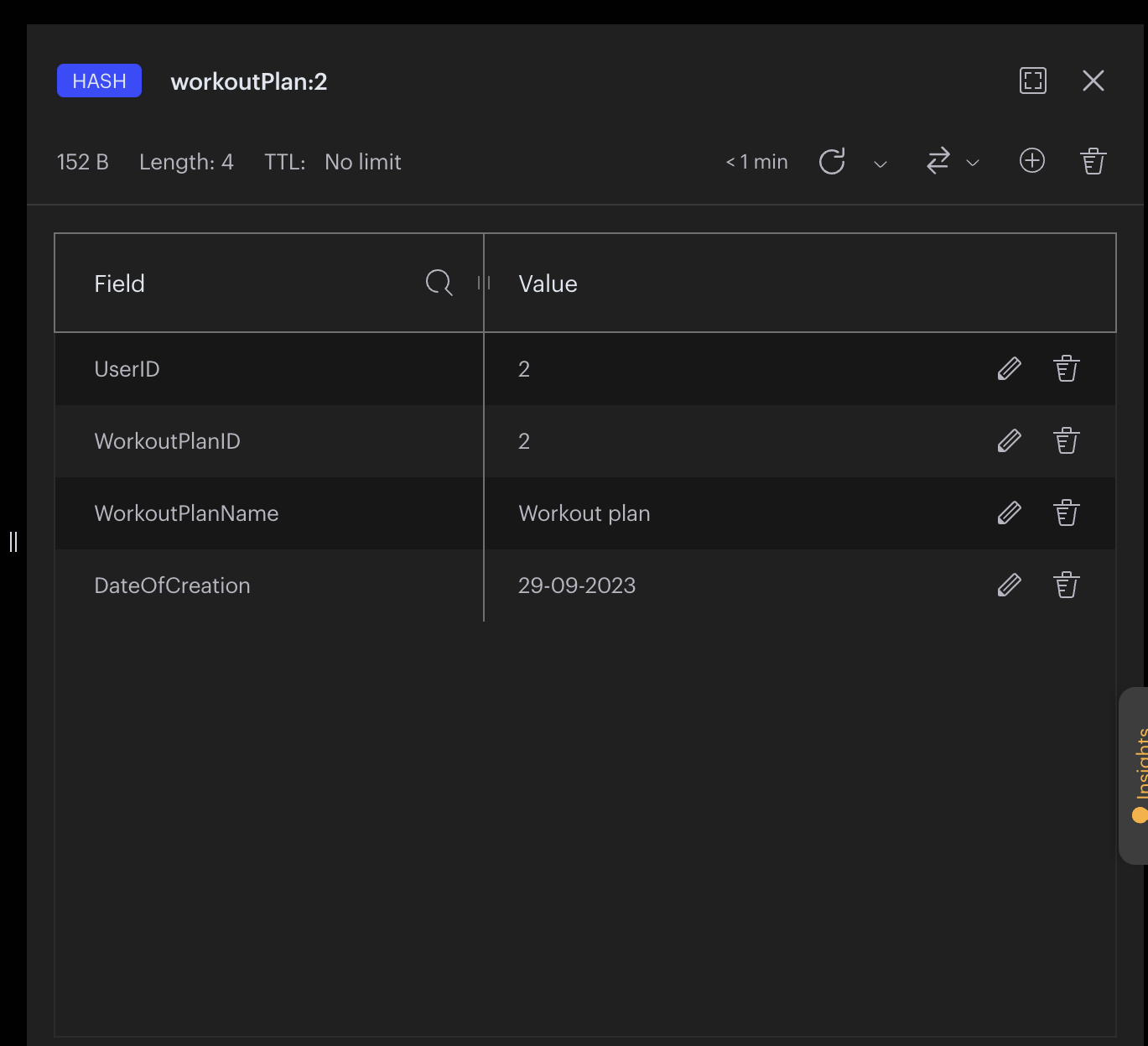
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique ID of the user who created the workout plan |
| WorkoutPlanID | Int | Unique identifier for the workout plan |
| WorkoutPlanName | String | Name of the workout plan |
| DateOfCreation | String | Date of creation of the workout plan. |

Redis is used as the Database to store WorkoutPlan records.

The key would be format = “workoutPlan:<userID>”, and type of key is “Hash key”. WorkoutPlanID must be unique for each workout plan.

Below is the snapshot of a WorkoutPlan record in Redis database.





**WorkoutPlanRecord**

This denotes the object for each workout in the workout plan (which is created by the user), and this object denotes the details of that workout which includes the no of units (per day) set by the user and calories burnt.

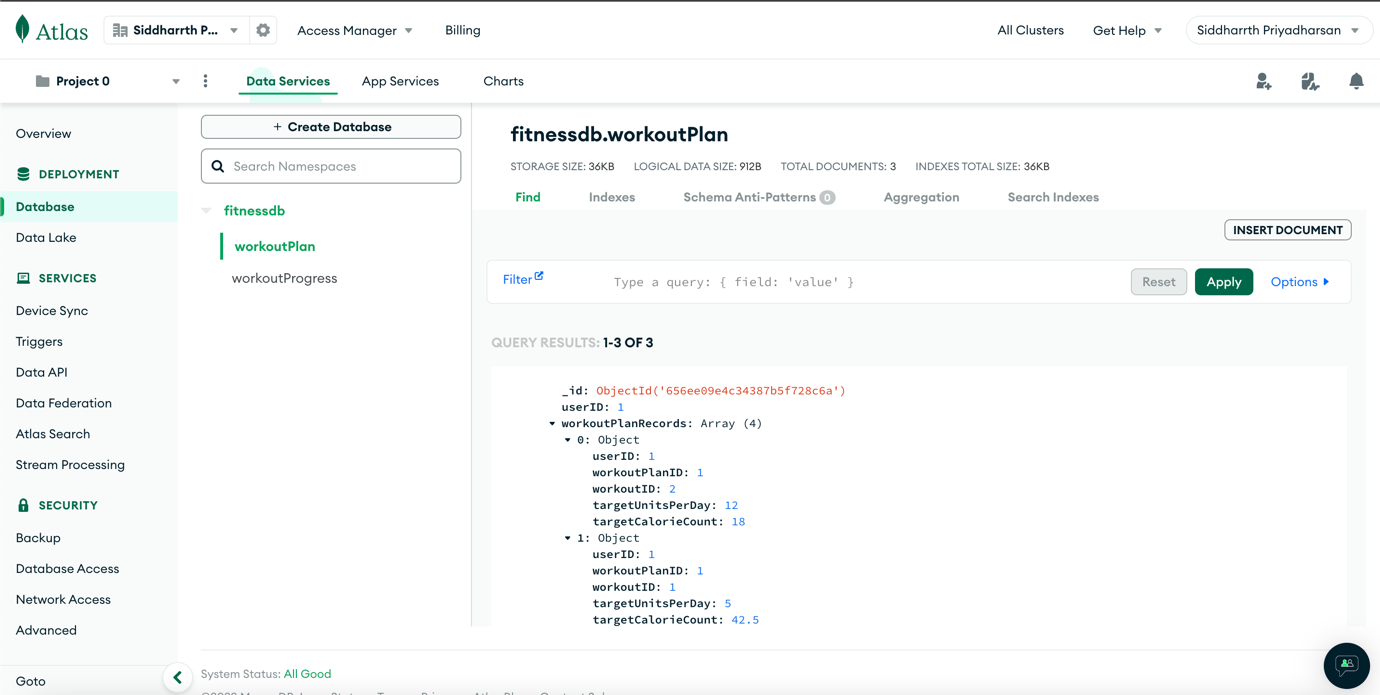
**Schema is as follows:**

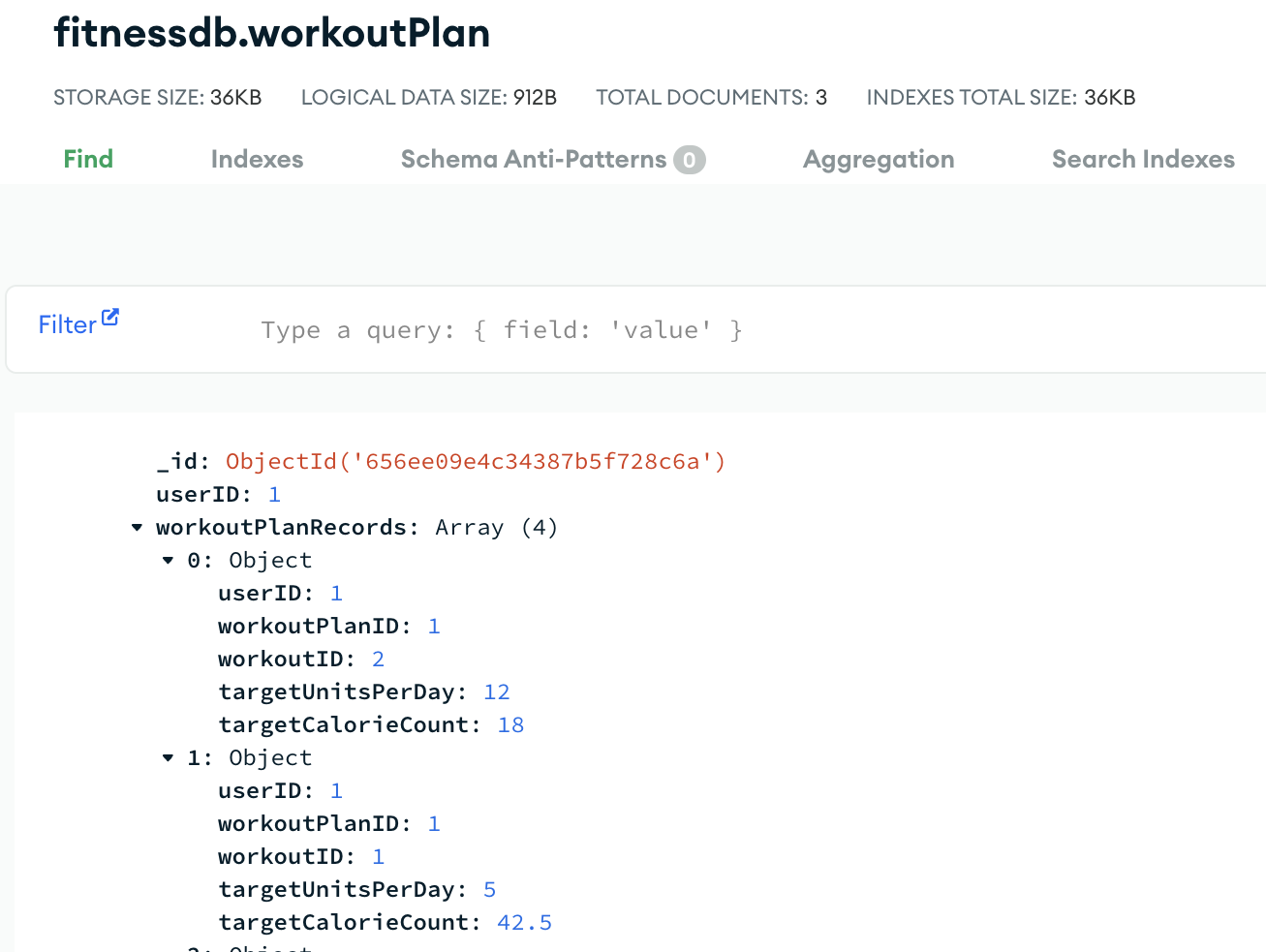
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier for the user record |
| WorkoutPlanID | Int | ID of the associated workout plan |
| WorkoutID | Int | Unique identifier for the Workout record |
| TargetUnitsPerDay | Int | Target number of units of workout for a day |
| TargetCalorieCount | Int | Number of calories burnt for the given target units for a day |

{WorkoutPlanID, WorkoutID} is a tuple that will uniquely identify a WorkoutPlanRecord record.

MongoDB is used to store WorkoutPlanRecord entity. And we filter the documents in the MongoDB based on the UserID. All the WorkoutPlanRecord records of a given user is stored in a single document in MongoDB.

Below is the snapshot of WorkoutPlanRecord in MongoDB – fitnessdb.workoutPlan:





**Workout**

This denotes the object for workout, and this entity comprises of the details of that workout such as name, calories burnt (in numbers) per unit.

**Schema is as follows:**

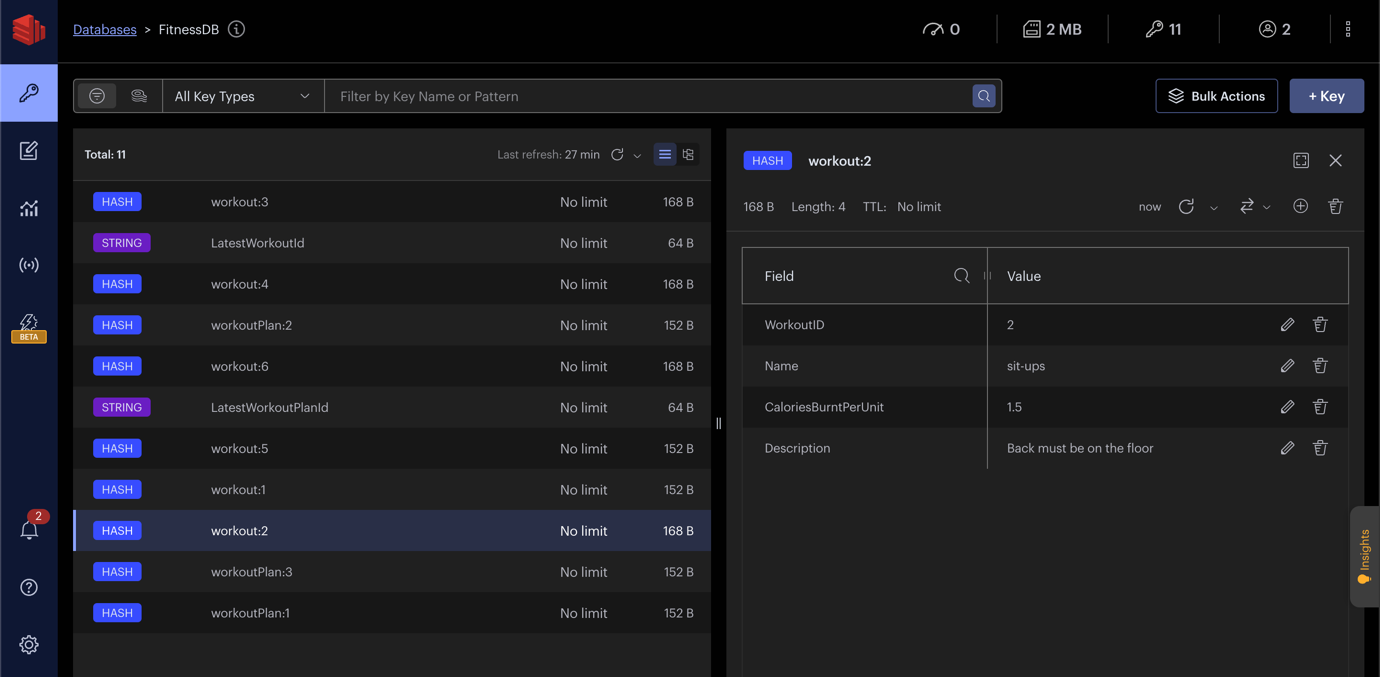
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| Name | String | Name of the workout |
| WorkoutID | Int | Unique identifier for the workout |
| CaloriesBurntPerUnit | Int | No of calories burnt per unit of the workout |
| Description | String | Description about the workout |

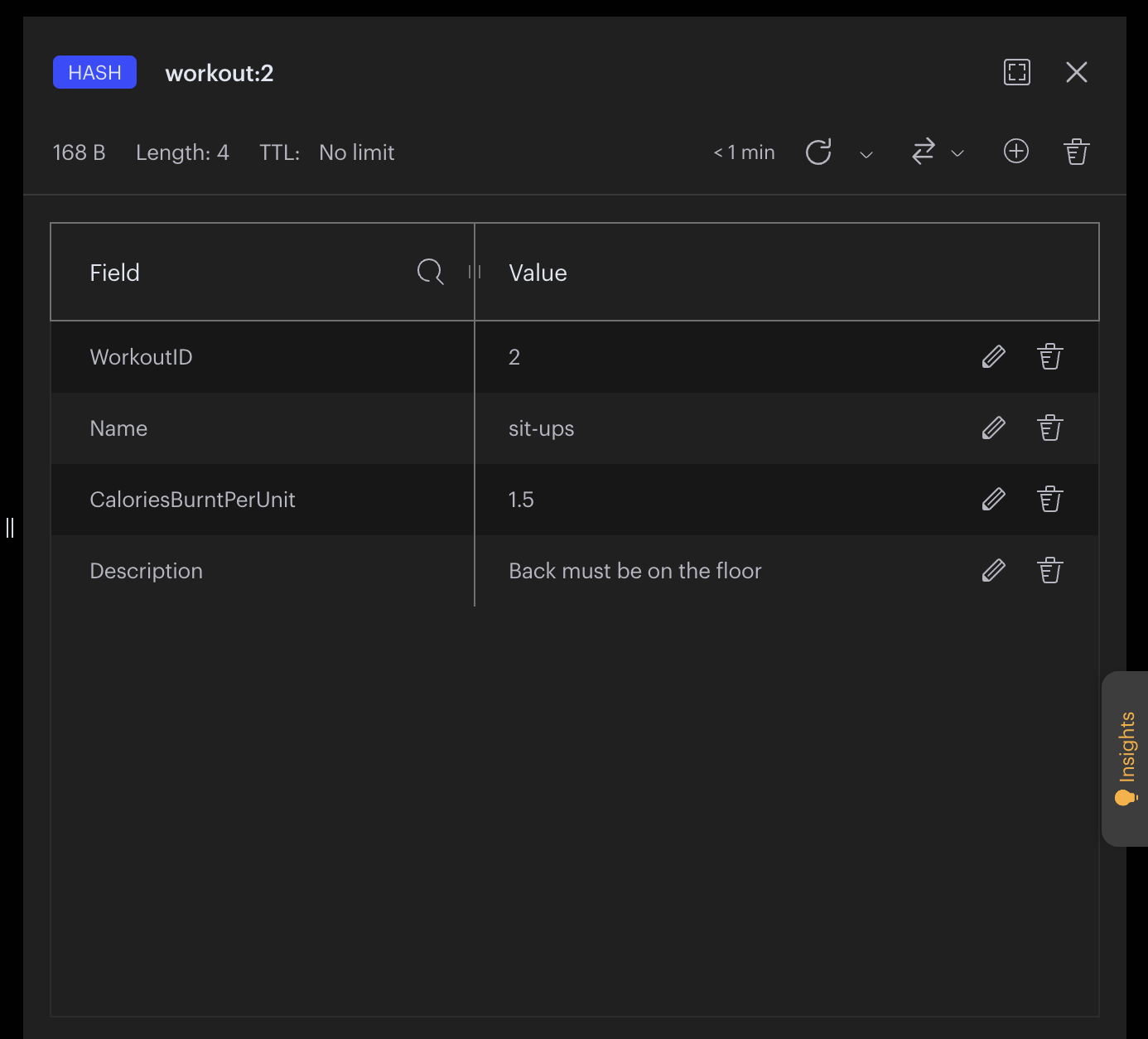
WorkoutID is the primary key of the table.

Redis is used as the Database to store Workout records.

The key would be format = “workout:<workoutID>” and type of key is “Hash key”. workoutID is unique for a given workout record.

Below is the snapshot of a workout record in Redis database.





**UserWorkoutRecord**

This denotes the object for each workout done by the user, and this entity comprises of the details such as the date of workout, no of units done on that day.

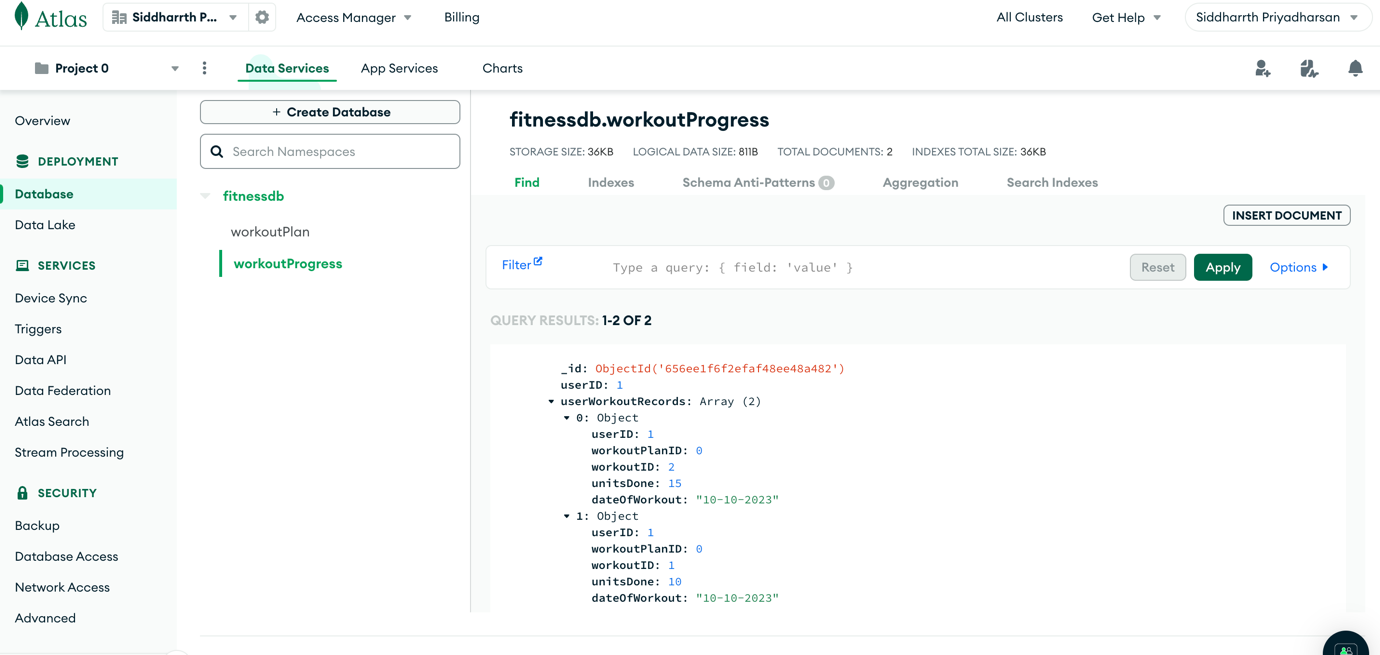
**Schema is as follows:**

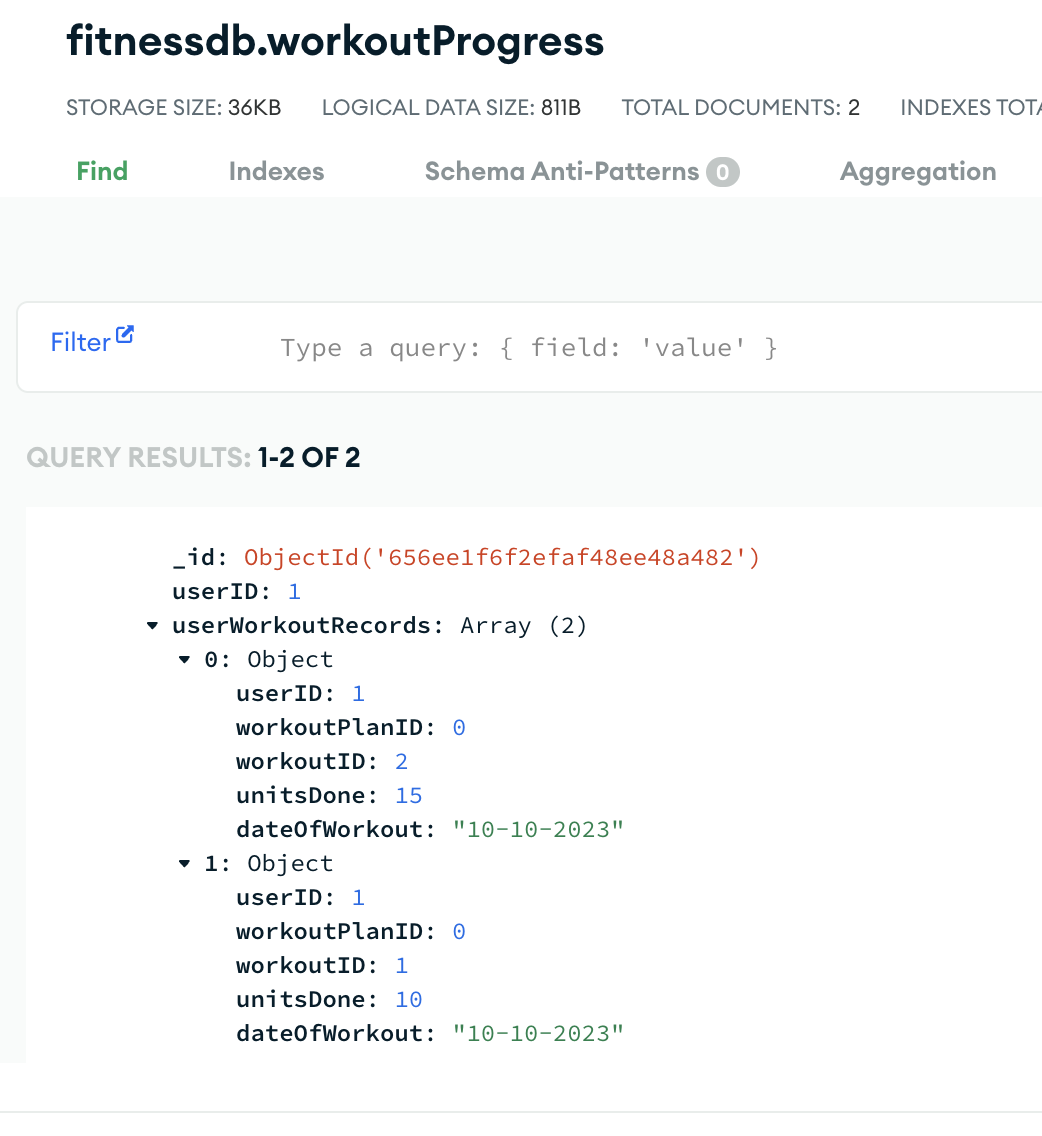
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier for the user |
| WorkoutPlanID | Int | ID of the associated workout plan |
| WorkoutID | Int | ID of the associated workout |
| DateofWorkout | String | Date of the workout performed. |

{WorkoutPlanID, WorkoutID, DateofWorkout} is a tuple that will uniquely identify a UserWorkoutRecord record.

MongoDB is used to store UserWorkoutRecord entity. And we filter the documents in the MongoDB based on the UserID. All the UserWorkoutRecord records of a given user is stored in a single document in MongoDB.

Below is the snapshot of UserWorkoutRecord in MongoDB – fitnessdb.workoutProgress:





**MealPlan**

This denotes the meal plan object which is created by the user, and this object is used to identify the list of food items under that meal plan.

**Schema is as follows:**

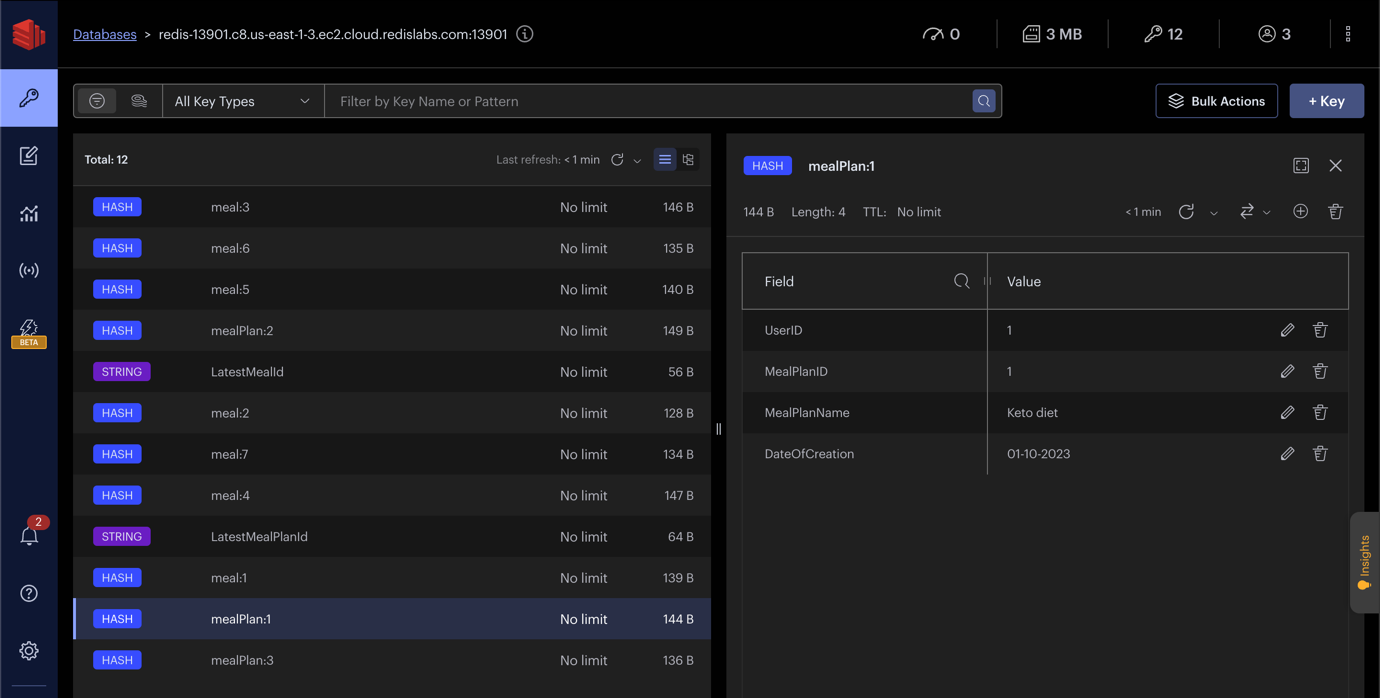
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier of the user record who created the meal plan |
| MealPlanID | Int | Unique identifier for the meal plan |
| MealPlanName | String | Name of the meal plan |
| DateOfCreation | String | Date of creation of the meal plan. |

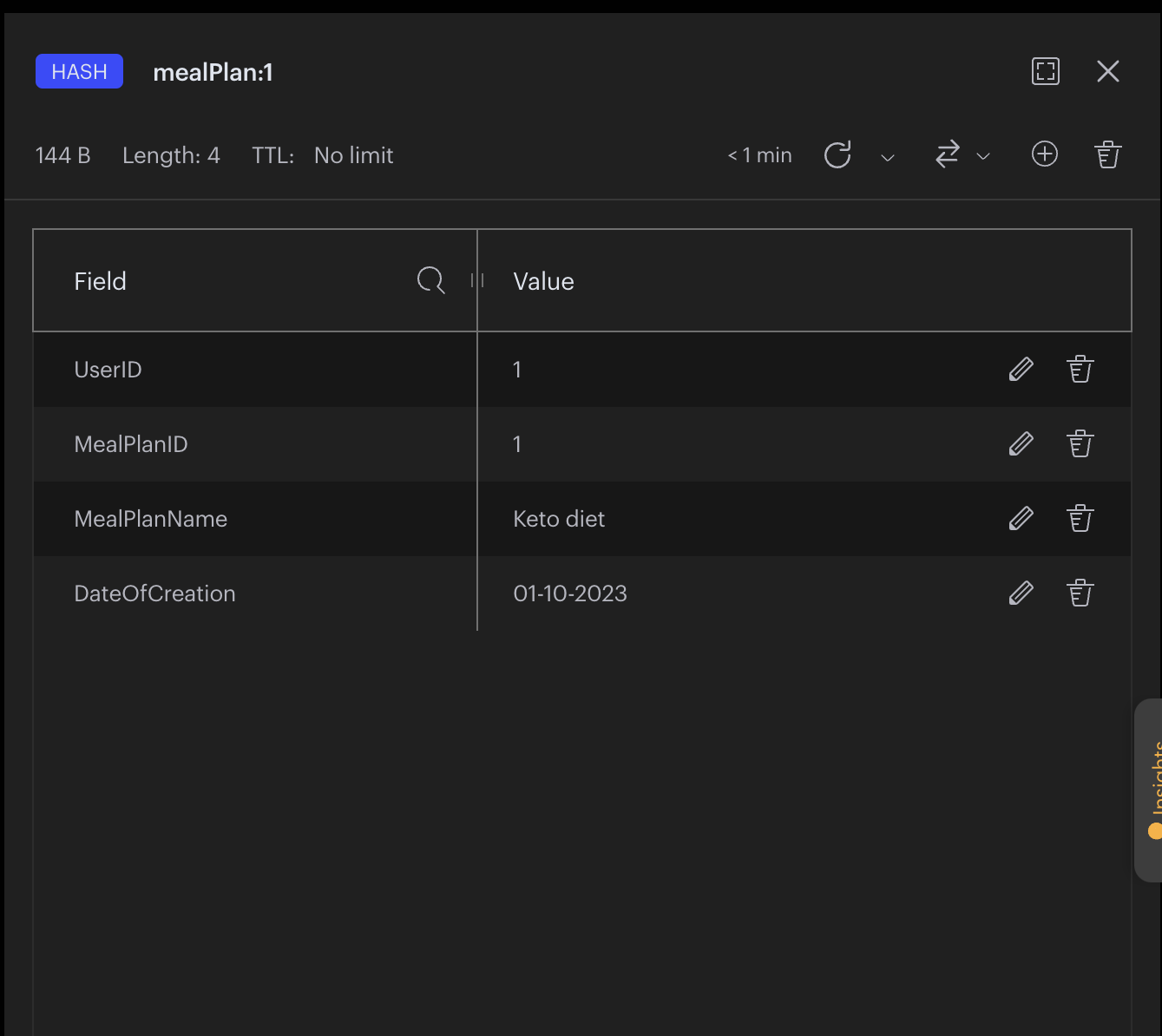
MealPlanID is the unique identifier for MealPlan entity.

Redis is used as the Database to store MealPlan records.

The key would be format = “mealPlan:<userID>” and type of key is “Hash key”. MealPlanID is unique for a given MealPlan record.

Below is the snapshot of a MealPlan record in Redis database:





**MealPlanRecord**

This denotes the object for each food item in the meal plan (which is created by the user), and this object denotes the details of that food item (meal) which includes the quantity (in grams) set by the user, calories.

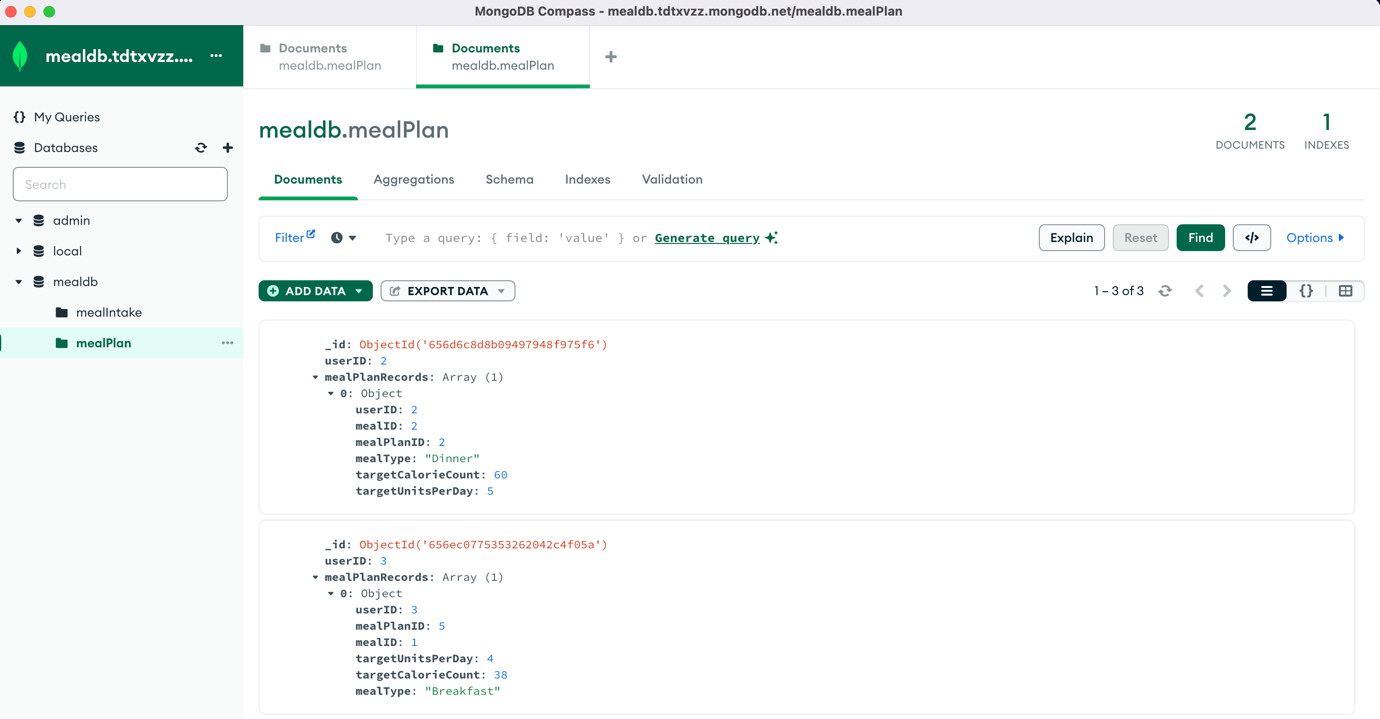
**Schema is as follows:**

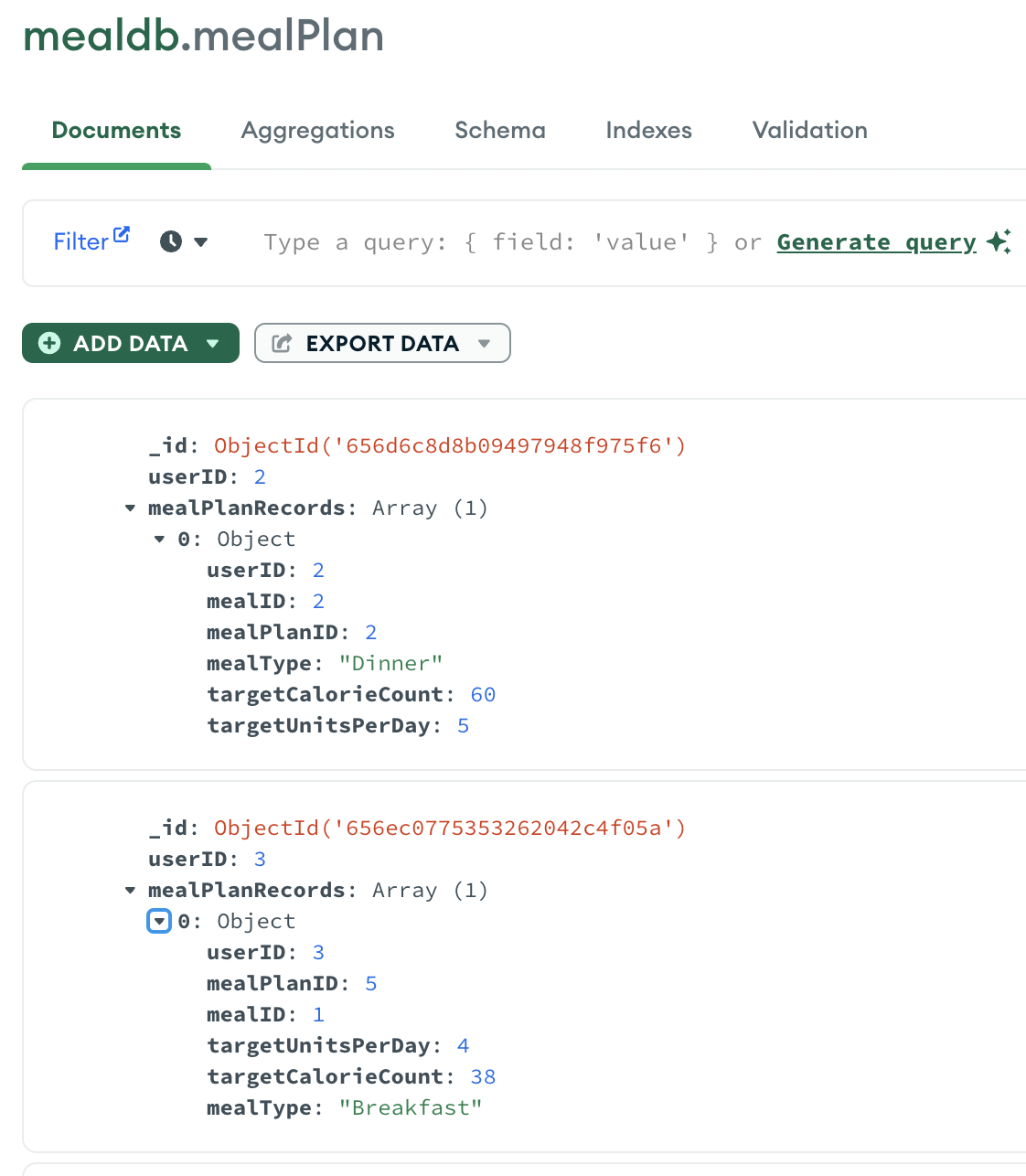
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier of the user record |
| MealPlanID | Int | ID of the associated meal plan |
| MealID | Int | Unique identifier for the meal (food item) record |
| Quantity | Int | Target number of units of that item for a day |
| CalorieCount | Int | Number of calories corresponding to the given target units for a day |
| MealType | String | Type of meal. It can be one of {“Breakfast”, “Lunch”, “Dinner”} |

{MealPlanID, MealID} is a tuple that will uniquely identify a MealPlanRecord record.

MongoDB is used to store MealPlanRecord entity. And we filter the documents in the MongoDB based on the UserID. All the MealPlanRecord records of a given user is stored in a single document in MongoDB.

Below is the snapshot of MealPlanRecord in MongoDB – mealdb.mealPlan:





**Meal**

This denotes the object for food item (meal), and this entity comprises of the details of that item such as calorie (in units)

**Schema is as follows:**

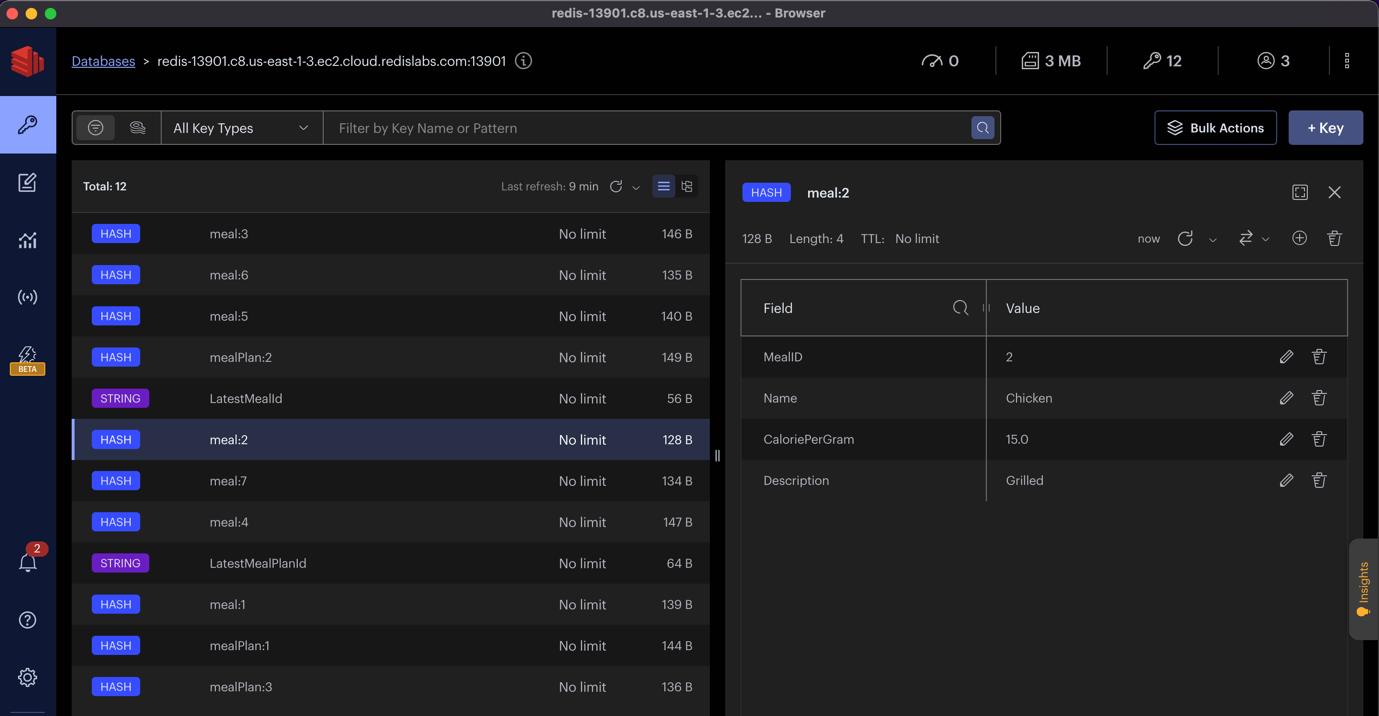
|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| Name | String | Name of the food item (meal) |
| MealID | Int | Unique identifier for the Meal (food item) |
| CaloriesPerGram | Int | No of calories in each unit of that food item (meal) |
| Description | String | Description about the food item |

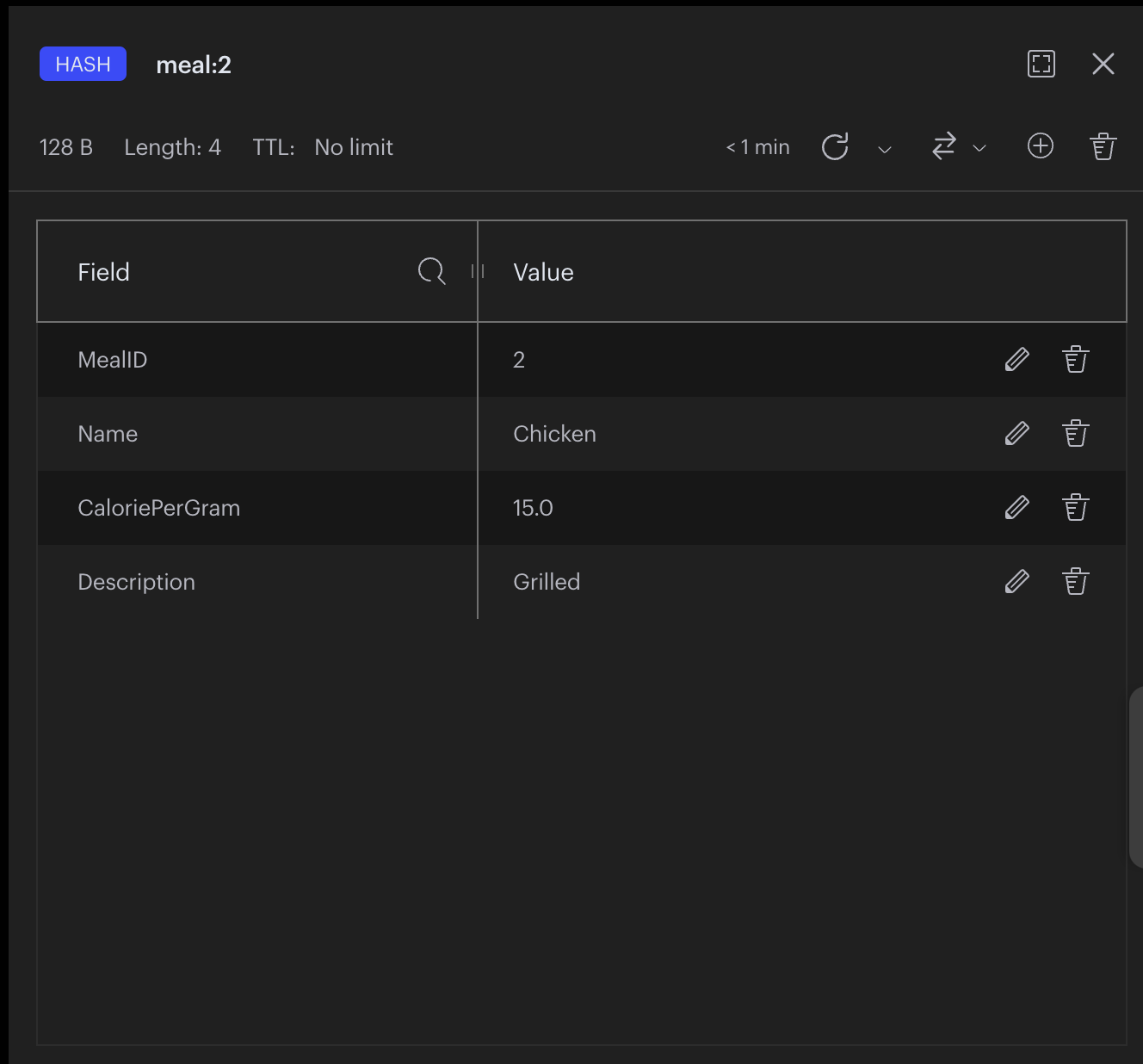
MealID is the unique identifier of Meal record.

Redis is used as the Database to store Meal records.

The key would be format = “meal:<MealID>” and type of key is “Hash key”. MealID is unique for a given Meal record.

Below is the snapshot of a Meal record in Redis database:





**UserMealIntakeRecord**

This denotes the object for food item consumed by the user for a meal, and this entity comprises of the details such as the meal type (Breakfast / Lunch / Dinner), date of intake, quantity consumed (in grams)

**Schema is as follows:**

UserID

MealPlanID

MealID

MealType

QuantityConsumed (in grams)

DateofIntake

|  |  |  |
| --- | --- | --- |
| Column | Data type | Description |
| UserID | Int | Unique identifier for the user |
| MealPlanID | Int | ID of the associated meal plan |
| MealID | Int | ID of the associated Meal (food item) |
| QuantityConsumed | Int | Number of units of that food item consumed by the user on that day |
| DateofIntake | String | Date of the meal consumed |
| MealType | String | Type of meal. It can be one of {“Breakfast”, “Lunch”, “Dinner”} |

{MealPlanID, MealID, DateOfIntake} is a tuple that will uniquely identify a UserMealIntakeRecord record in the database.

MongoDB is used to store UserMealIntakeRecord entity. And we filter the documents in the MongoDB based on the UserID. All the UserMealIntakeRecord records of a given user is stored in a single document in MongoDB.

Below is the snapshot of UserMealIntakeRecord in MongoDB – mealdb.mealIntake:

