

Rocky’s Ruff Retreat Database Design Document

POWERSHELL RANGERS | 23 April 2019 | ISM 4212 | Dr. Hyman

**Motto:**

It’s Database Time!

**Members:**

Brennan Harrison (Lead)

Huong Le

Brian Pueschel

Mike Sangrene

**Version History**

|  |  |
| --- | --- |
| **Version** | **Description** |
| 1.0 | First released draft |
| 2.0 | Summary of changes:   1. Rocky’s Ruff Retreat Relational Schema 2. Rocky’s Ruff Retreat Data Dictionary Header 3. Rocky’s Ruff Retreat Data Dictionary |
| 3.0 | Summary of changes:   1. Updated Rocky’s Ruff Retreat Data Dictionary    1. Converted all Primary Key and Foreign Key data types to int.    2. Dropped check constraint for all Phone Number columns.    3. Converted all Phone Number columns data types to varchar (20).    4. Converted all Zip column data types to varchar (10).    5. Renamed Occupation attribute to Type    6. Converted Type column data type to char (1).    7. Modified all Phone Number columns to allow nulls.    8. Modified Number Of Bowls Fed column to allow for nulls.    9. Modified Number Of Baths Given column to allow for nulls.    10. Modified Number Of Walks Taken column to allow for nulls. 2. Table Views 3. SPROCS |
| 4.0 | Summary of changes:   1. Table View Reports 2. User-Acceptance Tests |

**Purpose**

PowerShell Rangers prepared this document to design a data model listing the major entities, attributes, and relations that make up Rocky’s Ruff Retreat.

**Narrative**

Rocky’s Ruff Retreat is a privately owned doggie daycare located a few miles east of the University of South Florida. As a growing business they begin to realize that using paper documents to track their operations is inefficient. This understanding has compelled them to create a database to support their expanding business.

Customers can drop off one or more dogs at the daycare. A customer is tracked by their unique customer identifier, first name, last name, phone number, e-mail, and address. Each dog is tracked by its unique dog identifier, name, drop off date, pick up date, and breed. A dog is assigned to a cage. A cage can only hold one dog at a time. Each cage is tracked by a cage identification number and aisle number. Some cages that are being cleaned will not be assigned to a dog.

Dogs can be assigned to more than one employee. An employee is tracked by their identification number, first name, last name, address, phone number, and type. Some employees may or may not be assigned to any dogs, depending on their type. An employee is either a receptionist or caregiver. For each employee, the daycare wants to track the number of bowls fed, number of baths given, and number of walks taken for each dog.

**Requirements (Actors and Roles)**

CUSTOMER- A customer drops off dogs.

DOG- A dog, belonging to a customer, is taken care of by employees. A dog is assigned to a cage.

CAGE- A cage may or may not be holding a dog.

EMPLOYEE- An employee may or may not be taking care of a dog.

**Entities Identified**

CUSTOMER DOG CAGE EMPLOYEE

**Entities with Nested Attributes**

EMPLOYEE

EmployeeID

First Name

Last Name

Street

City

State

Zip

Phone Number

Type

CAGE

CageID

Aisle Number

DOG

DogID

Dog Name

Drop Off Date

Pick Up Date

Breed

CUSTOMER

CustomerID

First Name

Last Name

Street

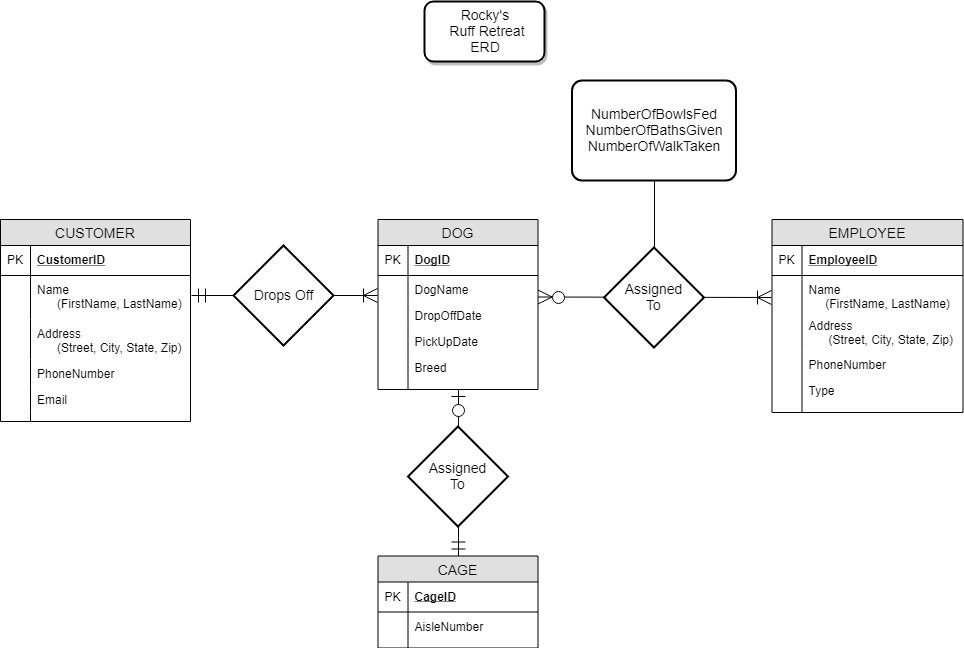
City

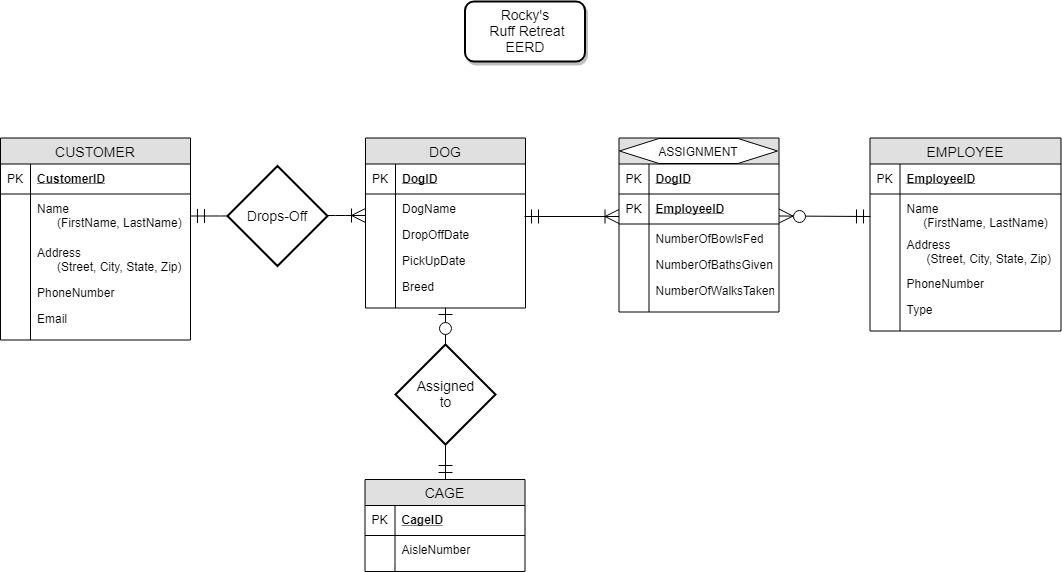
State

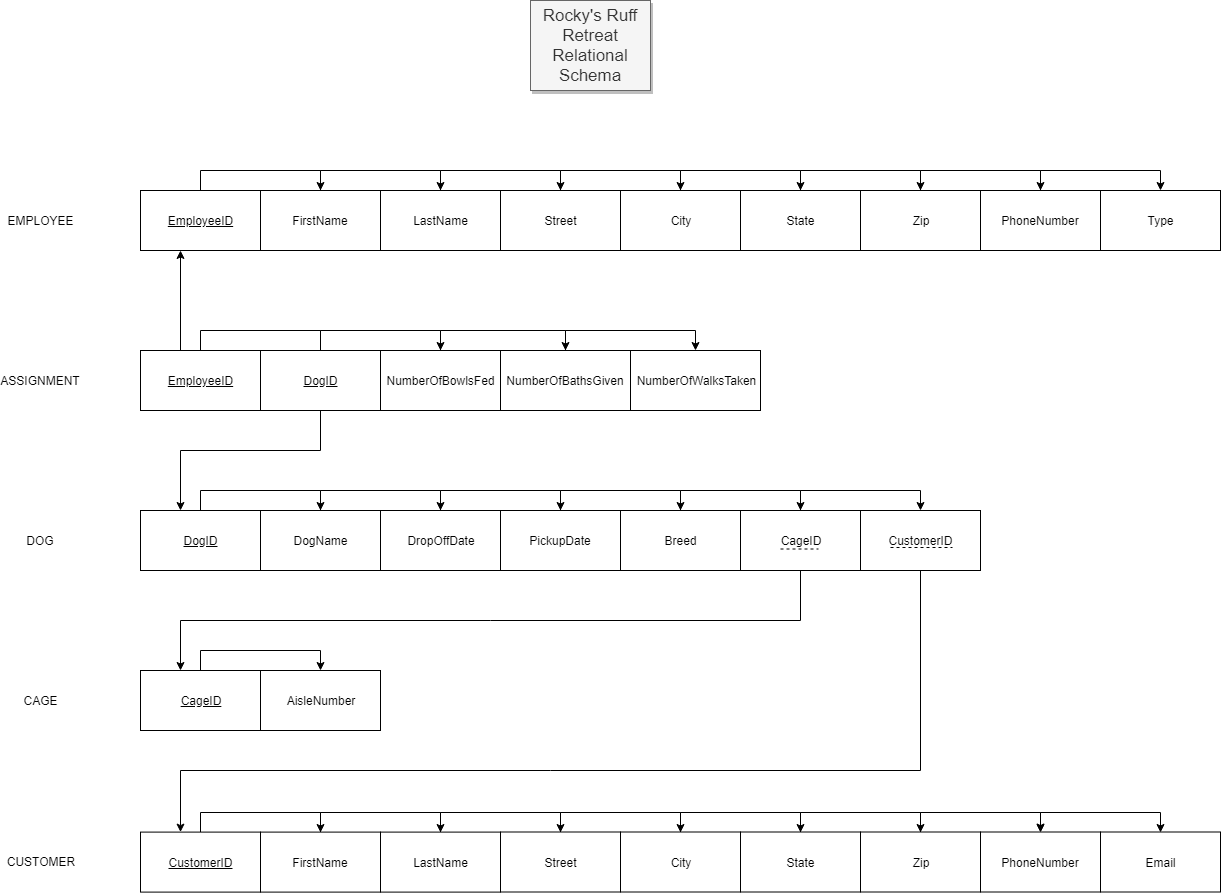
Zip

Phone Number

Email







**Data Dictionary Summary Header**

**Rocky’s Ruff Retreat Database Tables and Attributes**

Employee (EmployeeID, FirstName, LastName, Street, City, State, Zip PhoneNumber,

Type)

Assignment (EmployeeID, DogID, NumberOfBowlsFed, NumberOfBathsGiven,

NumberOfWalksTaken)

Dog (DogID, DogName, DropOffDate, PickUpDate, Breed, CageID, CustomerID)

Cage (CageID, AisleNumber)

Customer (CustomerID, FirstName, LastName, Street, City, State, Zip PhoneNumber, Email)

ROCKY’S RUFF RETREAT – DATA DICTIONARY

**(Microsoft SQL Server Notation)**

Table: **Employee**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **PK**; Unique sequential employee ID number | int |  | Y |  |  |  |  | Y |
| FirstName | First name of the employee | varchar | 15 |  |  |  |  |  |  |
| LastName | Last name of the employee | varchar | 20 |  |  |  |  |  |  |
| Street | Street of the employee | varchar | 20 |  |  |  |  |  |  |
| City | City of the employee | varchar | 15 |  |  |  |  |  |  |
| State | State of the employee | char | 2 |  |  |  | LIKE ‘[A-Z][A-Z]’ |  |  |
| Zip | Zip code of the employee | varchar | 10 |  |  |  |  |  |  |
| PhoneNumber | Phone number of the employee | varchar | 20 |  |  |  |  | Y |  |
| Type | Type of employee | char | 1 |  |  |  | ([Type] = (‘C’)  OR  [Type] = (‘R’)) |  |  |

Table: **Assignment**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **CPK**; **FK** to Employee table | int |  |  |  |  |  |  | Y |
| DogID | **CPK**; **FK** to Dog table | int |  |  |  |  |  |  | Y |
| NumberOfBowlsFed | Number of bowls fed to a dog from employee | tinyint |  |  |  |  |  | Y |  |
| NumberOfBathsGiven | Number of baths given to a dog from employee | tinyint |  |  |  |  |  | Y |  |
| NumberOfWalksTaken | Number of walks taken for a dog from employee | tinyint |  |  |  |  |  | Y |  |

Table: **Dog**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| DogID | **PK**; unique sequential dog ID number | int |  | Y |  |  |  |  | Y |
| DogName | Name of the dog | varchar | 15 |  |  |  |  |  |  |
| DropOffDate | Date the dog was dropped off | date |  |  |  |  |  |  |  |
| PickUpDate | Date the dog was picked up | date |  |  |  |  |  |  |  |
| Breed | Breed of the dog | varchar | 30 |  |  |  |  |  |  |
| CageID | **FK**; cage the dog was held in | int |  |  |  |  |  | Y |  |
| CustomerID | **FK**; customer that owns the dog | int |  |  |  |  |  | Y |  |

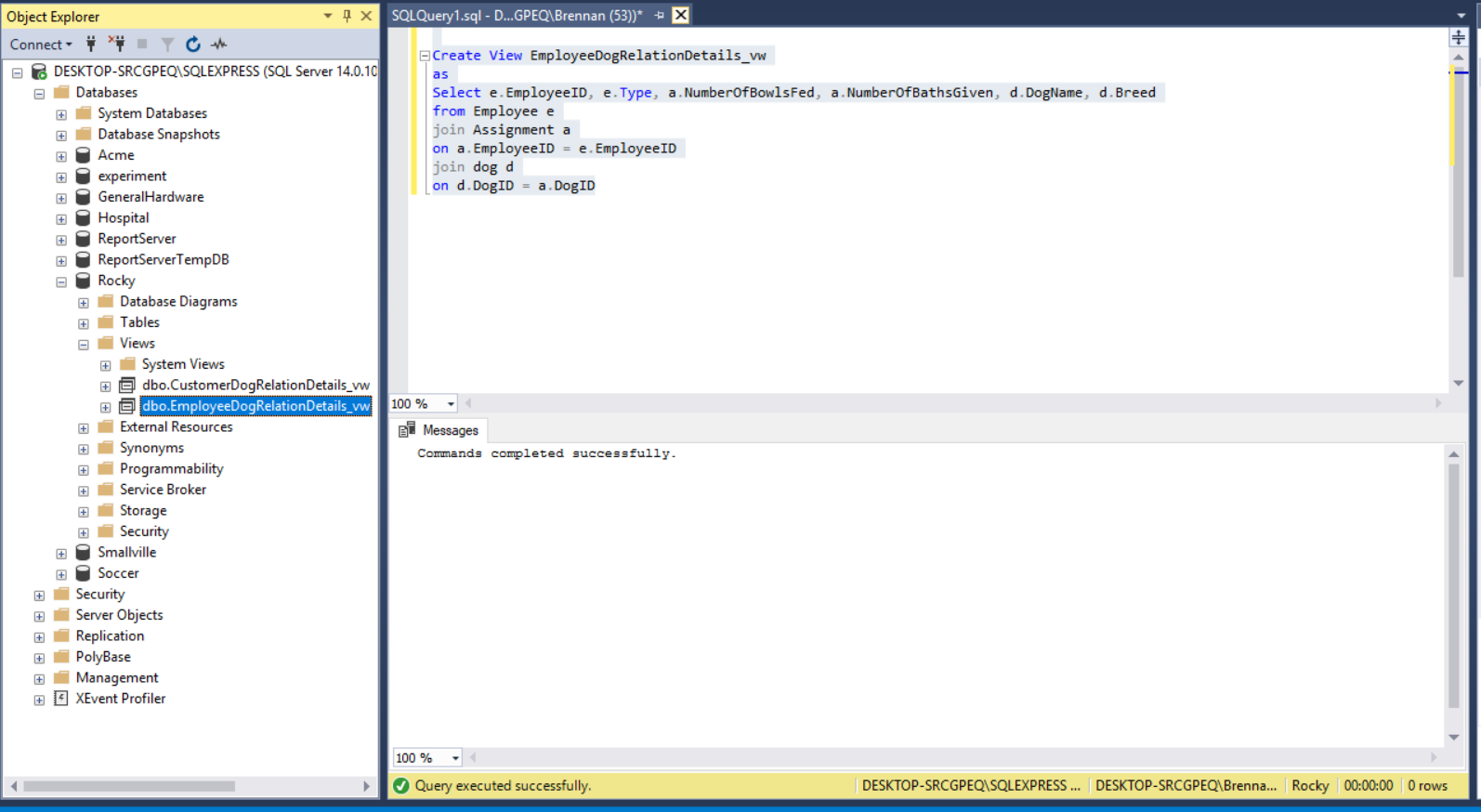
Table: **Cage**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| CageID | **PK**; unique sequential cage ID number | int |  | Y |  |  |  |  | Y |
| AisleNumber | Aisle that the cage is located | tinyint |  |  |  |  |  |  |  |

Table: **Customer**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| CustomerID | **PK**; unique sequential customer ID number | int |  | Y |  |  |  |  | Y |
| FirstName | First name of the customer | varchar | 15 |  |  |  |  |  |  |
| LastName | Last name of the customer | varchar | 20 |  |  |  |  |  |  |
| Street | Street customer lives on | varchar | 20 |  |  |  |  |  |  |
| City | City customer lives in | varchar | 15 |  |  |  |  |  |  |
| State | State customer lives in | char | 2 |  |  |  | LIKE ‘[A-Z][A-Z]’ |  |  |
| Zip | Zip code of the customer | varchar | 10 |  |  |  |  |  |  |
| PhoneNumber | Phone number of customer | varchar | 20 |  |  |  |  | Y |  |
| Email | Email address of the customer | varchar | 70 |  |  |  |  | Y |  |

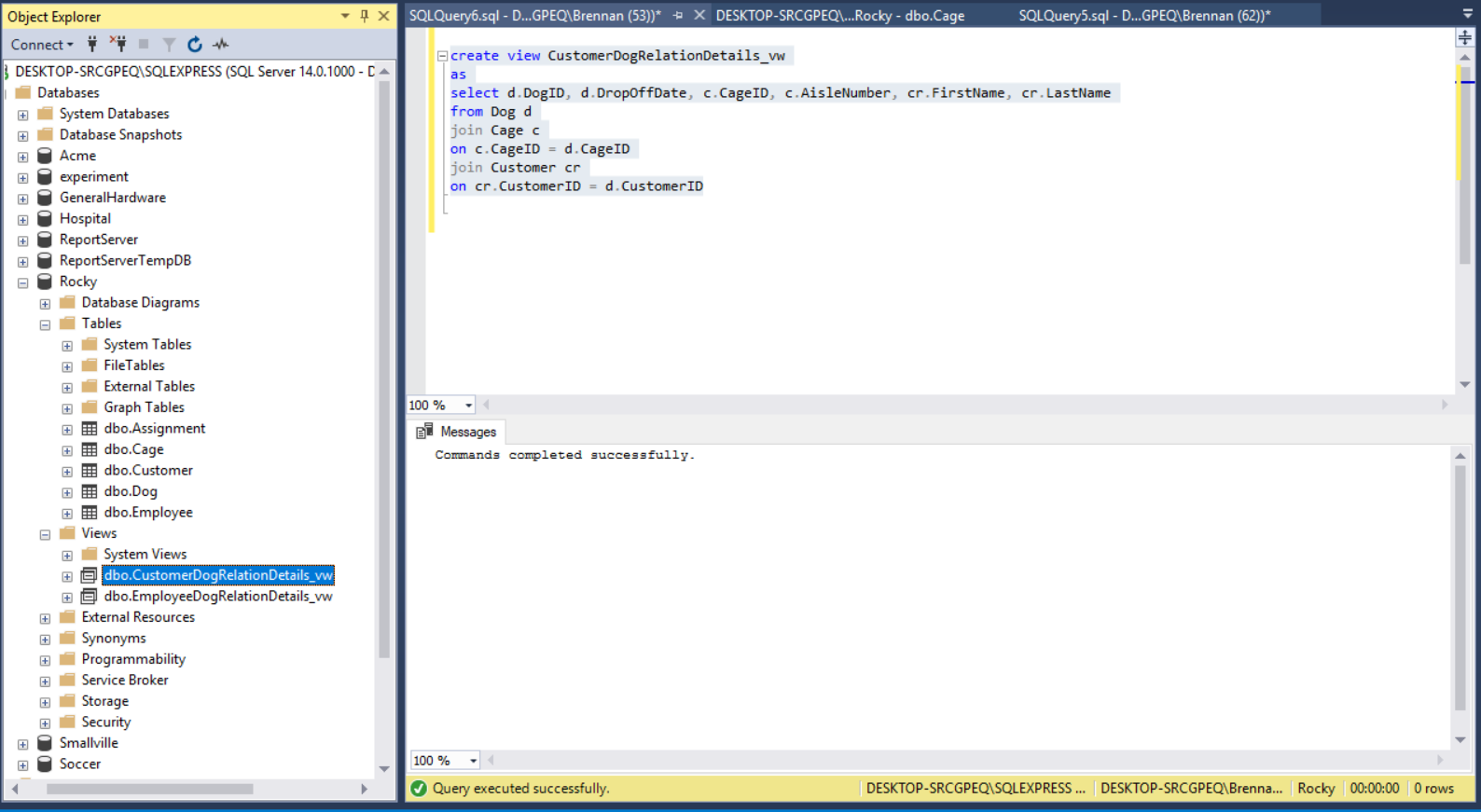
**Employee Dog Relation Details Table View**



**Employee Dog Relation Details Table View Description**

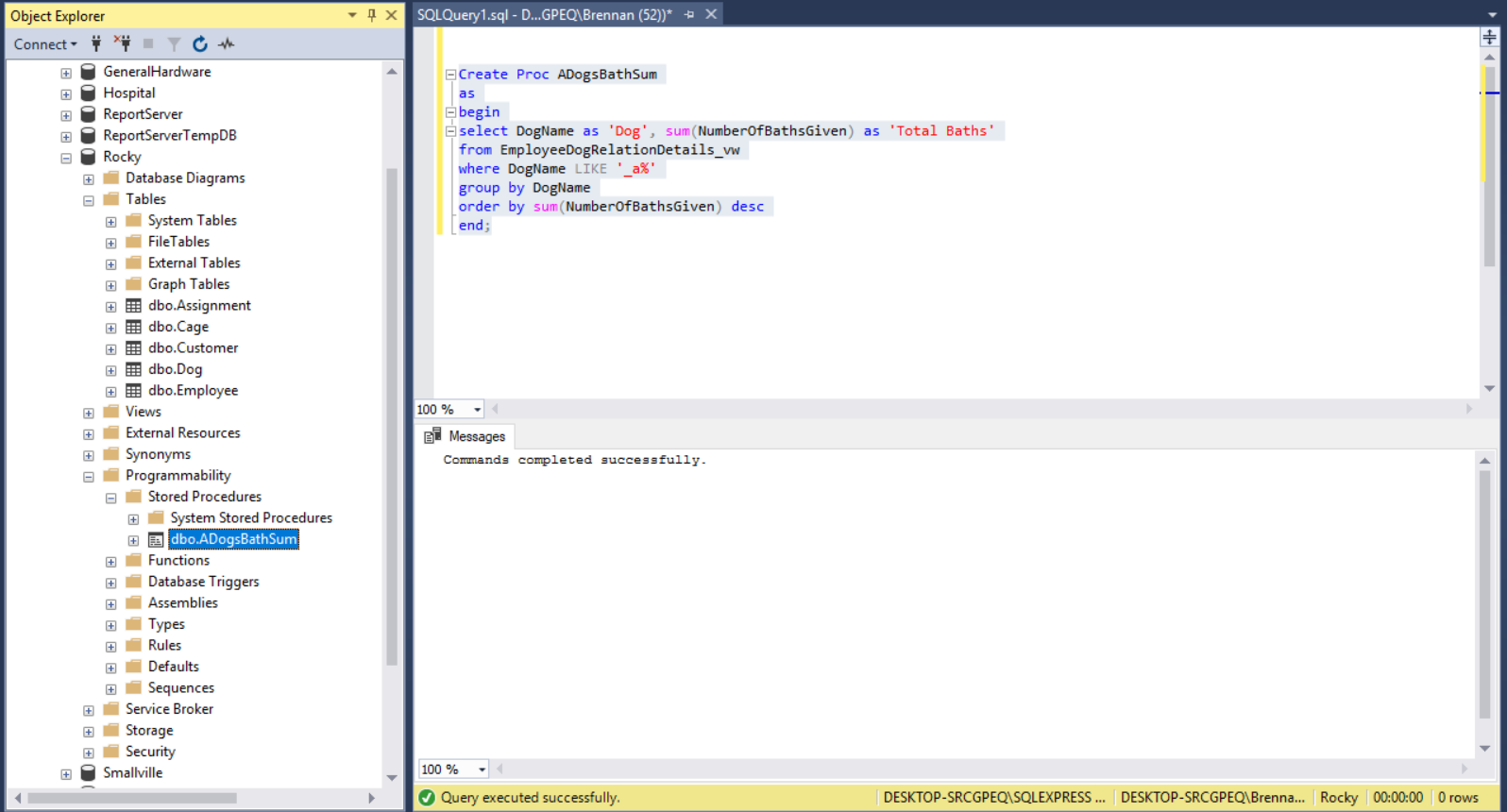
The Employee Dog Relation Details table view joins together the Employee, Assignment, and Dog tables on their primary and foreign key connections. Each table is given an alias for reader comprehension. The columns that are matched in this table view are Employee ID, Type, Number of Bowls Fed, Number of Baths Given, Dog Name, and Breed. The PowerShell Rangers chose these columns to do more narrowed searches on records using SPROCS.

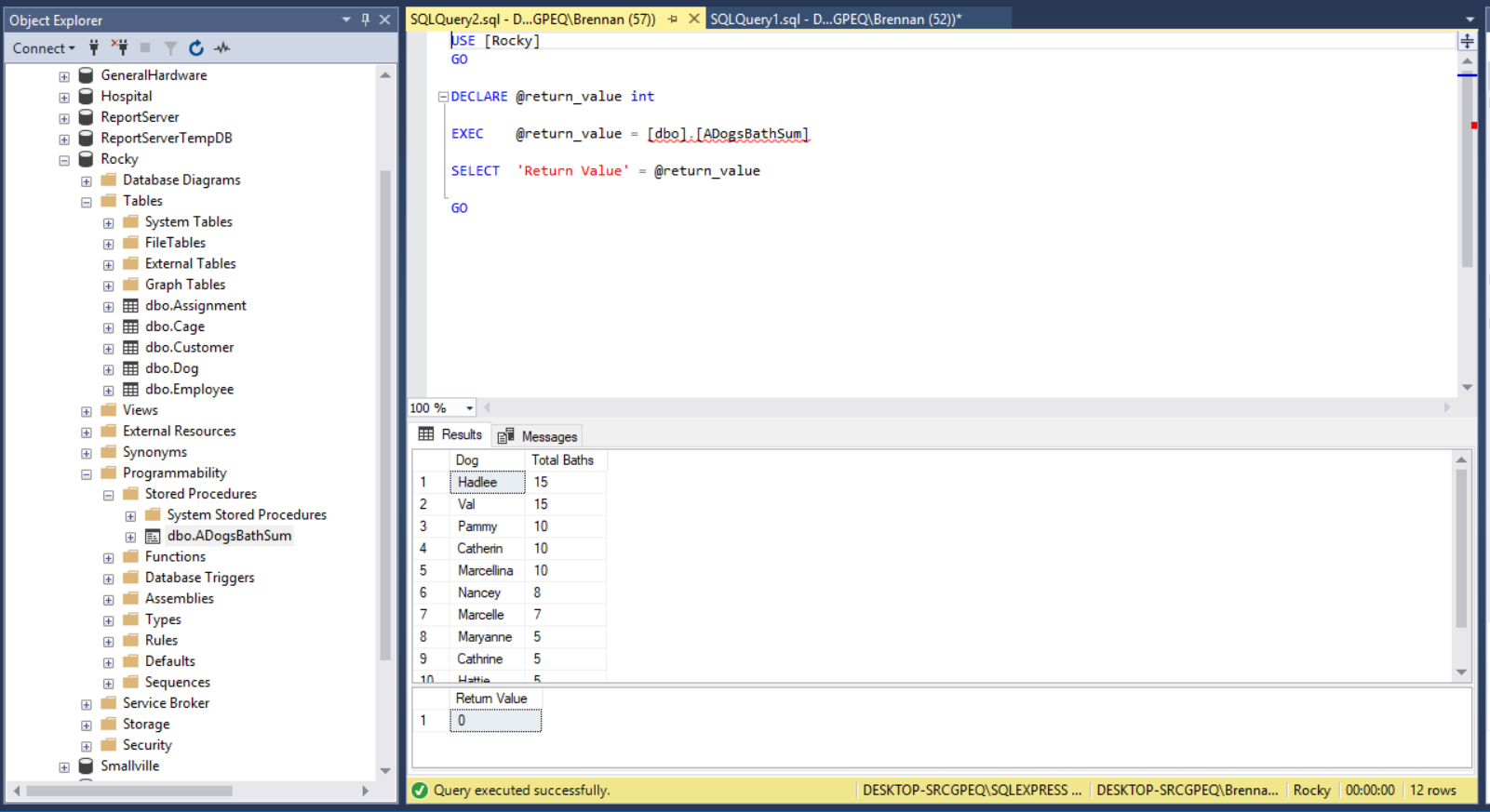
**Customer Dog Relation Details Table View**



**Customer Dog Relation Details Table View Description**

The Customer Dog Relation Details table view joins together the Dog, Cage, and Customer tables on their primary and foreign key connections. Each table is given an alias for reader comprehension. The columns that are matched in this table view are Dog ID, Drop Off date, Cage ID, Aisle Number, First Name, and Last Name. The PowerShell Rangers chose these columns to answer interesting questions utilizing SPROCS.

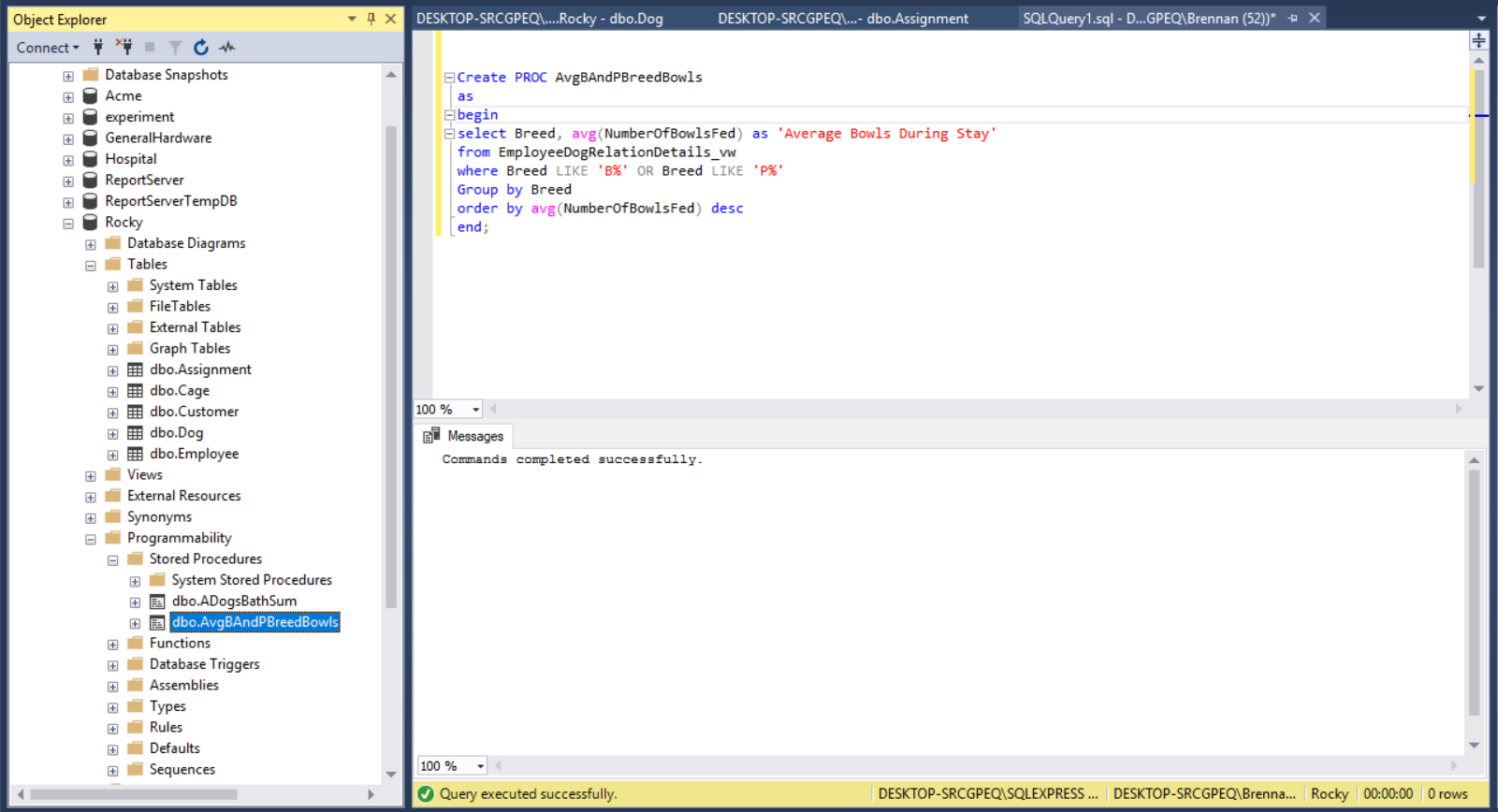
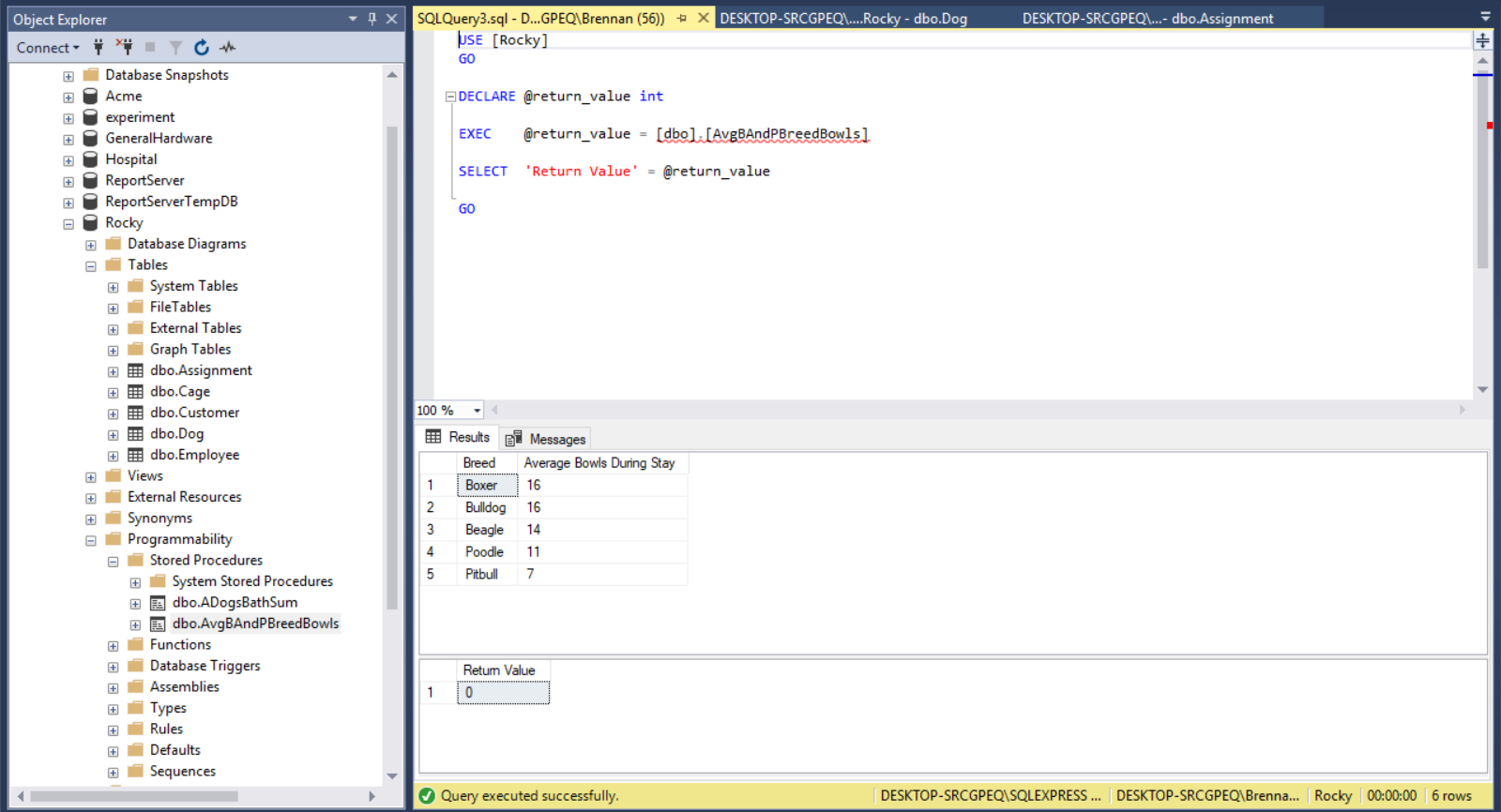
**Total Baths for Dogs With A as 2nd Character SPROC**



**Total Baths for Dogs With A as 2nd Character SPROC Description**

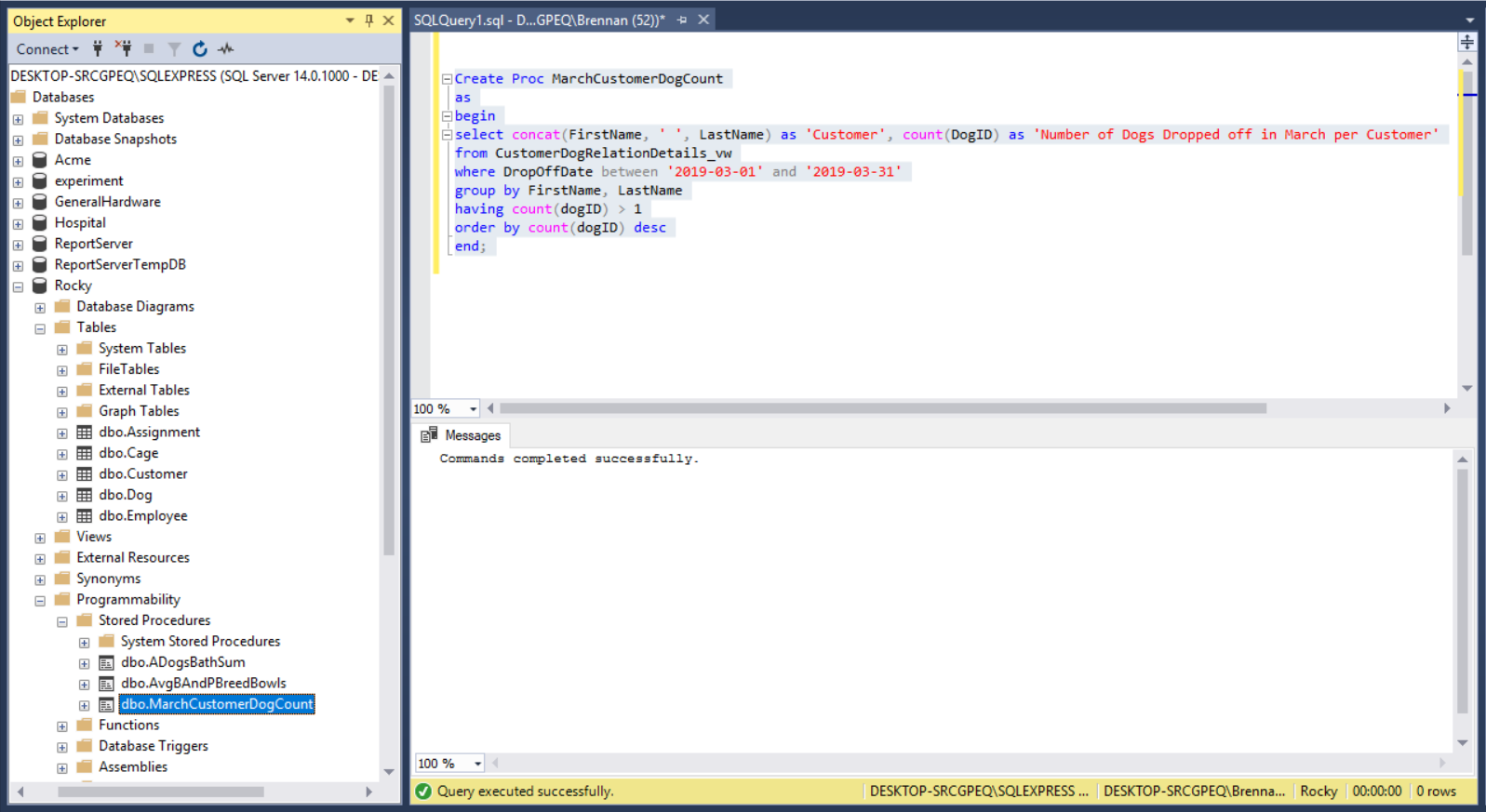
The SPROC named ADogsBathSum was created using the EmployeeDogRelationDetails table view and is searching for a list of all dogs whose name contains any first letter —but requires the letter ‘a’ as the second letter— and then any remaining letters. The aggregate sum of the number of baths each dog has received is also recorded. Each column uses a relative header for their output. The results are grouped by dog name, and then finally ordered by the aggregate sum of the number of baths given, in descending order.

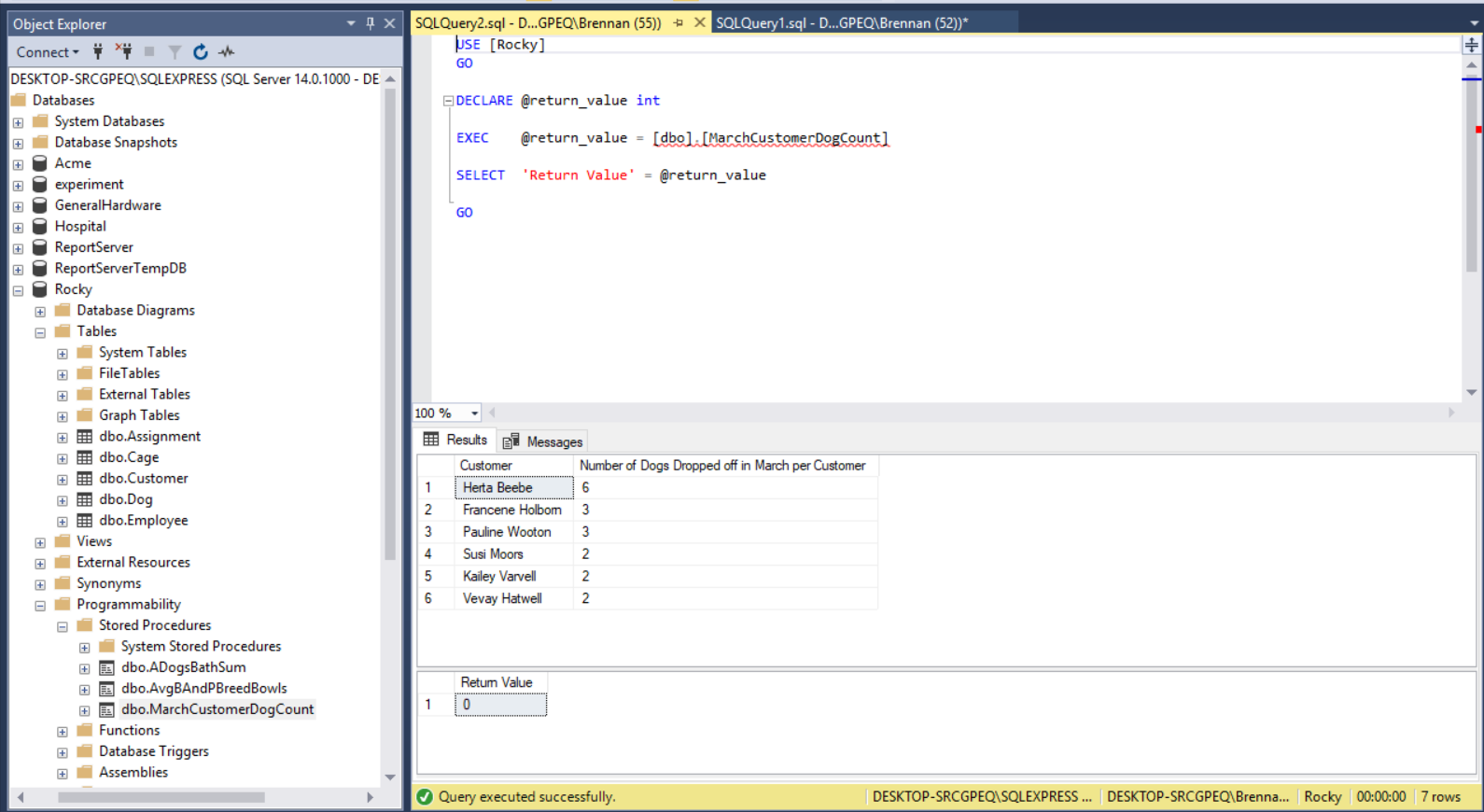
**Average Bowls Fed for Breeds that Start With B and P SPROC**



**Average Bowls Fed for Breeds that Start With B and P SPROC Description**

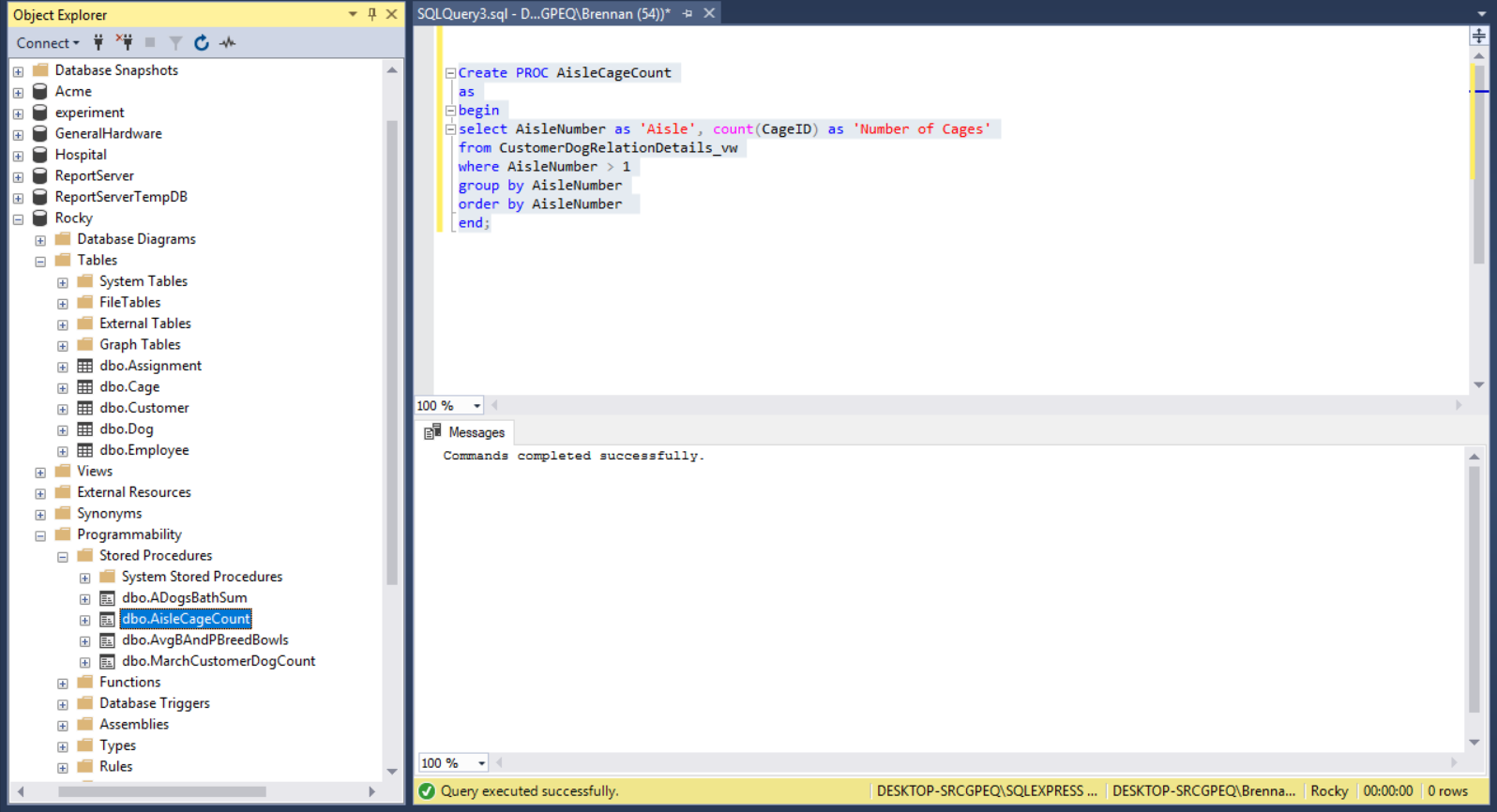
The SPROC named AvgBAndPBreedBowls was created using the EmployeeDogRelationDetails table view and is searching for the breed of dogs and the average number of bowls they consume during their stay. The where clause determines the breed must start with the letter ‘B’ or with ‘P’, and can contain any number of letters afterwards. The avg aggregate is used to find the average number of bowls the dogs are fed during their stay. The results are grouped by Breed and then finally ordered by the average number of bowls fed, in descending order.

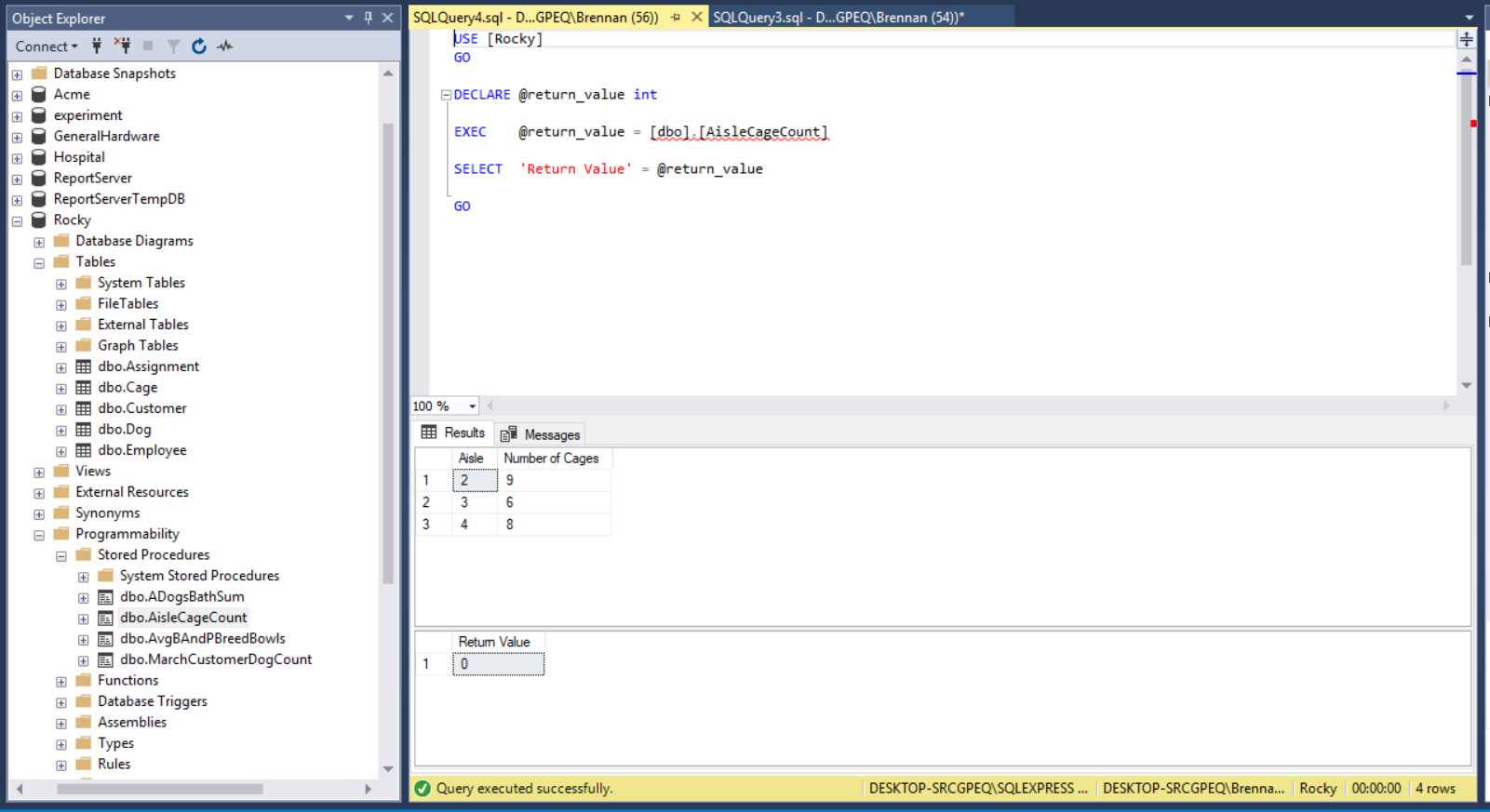
**March Customer Dog Count SPROC**



**March Customer Dog Count SPROC Description**

The SPROC named MarchCustomerDogCount was created using the CustomerDogRelationDetails table view and is searching for customers whose drop off dates fall between ‘2019-03-01’ and ‘2019-03-31’ and the number of dogs those customers dropped off in that range. The count aggregate is used to sum the number of dogs dropped off in March per Customer. The having clause indicates that number of dogs counted per customer must be greater than one. The results are grouped by the concatenation of customers’ first and last names, and then finally ordered by the count of DogID, in descending order.

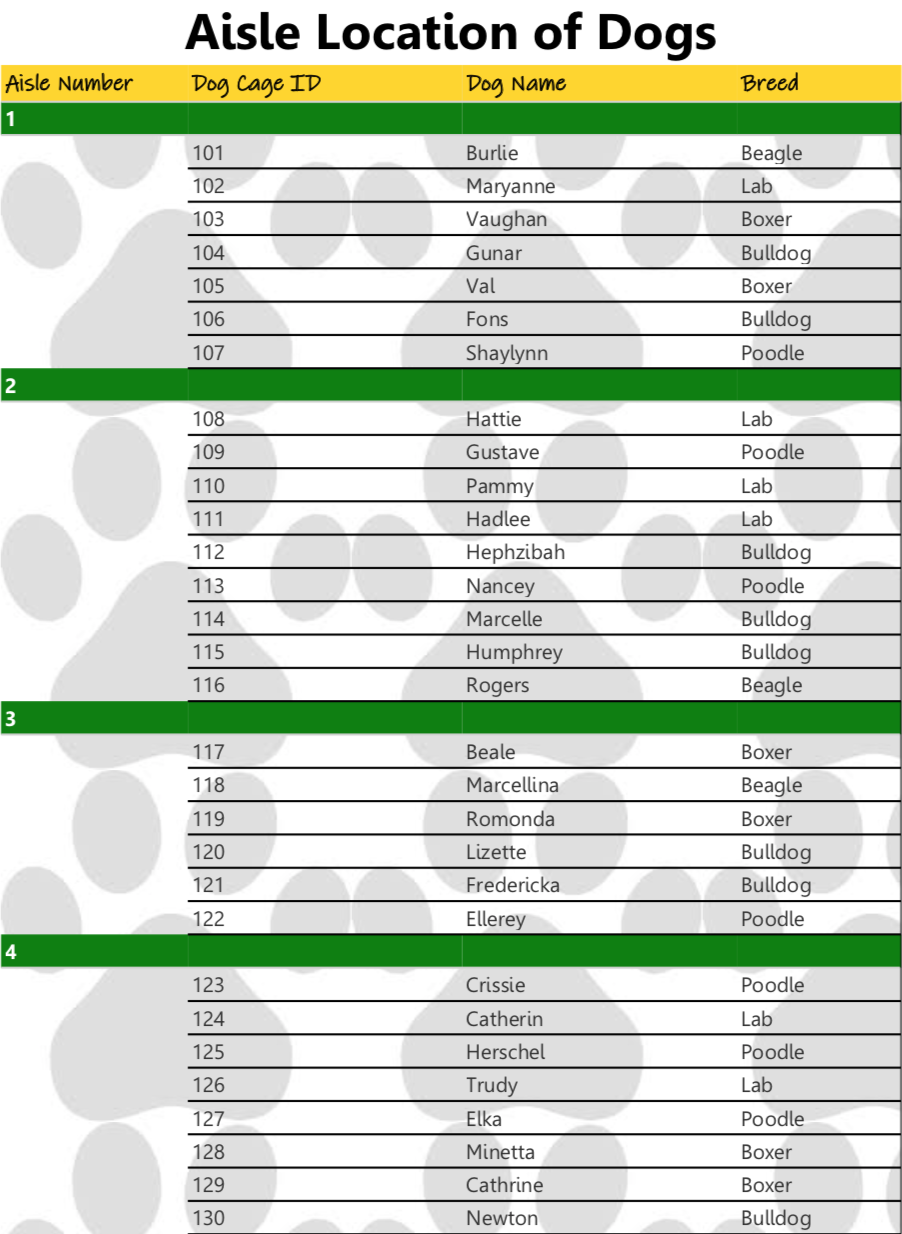
**Aisle Cage Count SPROC**

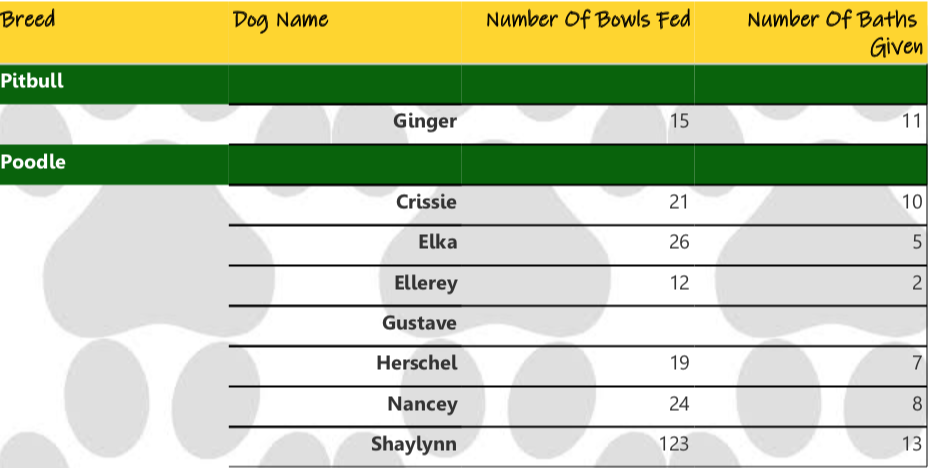


**Aisle Cage Count SPROC Description**

The SPROC named AisleCageCount was created using the CustomerDogRelationDetails table view and is searching for the aisle number and the number of cages in those aisles. The count aggregate is used to count the number of cages within each aisle. The where clause determines that the aisle number must be greater than one. The results are grouped by aisle number, and then finally ordered by aisle number.

**Customer Dog Relation Details Table View Report**



**Employee Dog Relation Details Table View Report** 

**User Acceptance Test Queries**

What is the number of bowls fed and days attended per dog, sorted by highest bowls fed?

select sum(a.[NumberOfBowlsFed]) as 'Bowls Fed',

DATEDIFF(day, d.[DropOffDate],d.[PickUpDate]) as 'Days Attended', d.[DogName] as 'Dog Name'

from [Assignment] a

join [dog] d

on a.[DogID] = d.[DogID]

where a.[NumberOfBowlsFed] is not null and a.[NumberOfWalksTaken] is not null

group by d.[DogName], d.[DropOffDate], d.[PickUpDate]

order by sum(a.[NumberOfBowlsFed]) desc

What customers are missing email addresses, sorted by the customers’ first names?

select concat(c.[FirstName], ' ', c.[LastName]) as 'First Name',

ISNULL(convert(varchar(13),c.[Email]), 'Missing Email') as 'Email'

from [Customer] c

where c.[Email] IS NULL

order by concat(c.[FirstName], ' ', c.[LastName])

How many dogs come from each city, starting from the city with the most dogs?

select count(d.[DogID]) as 'Number of Dogs Per City', c.[City] as 'City'

from [Dog] d

right join [Customer] c

on c.[CustomerID] = d.[CustomerID]

group by c.[City]

order by count(d.[DogID]) desc