Fish Market Database Design Document

Version 1.0 Revision 15

Submittal Date: 10/29/19

Version history

Version	Description
1.0 rev 12	First released draft
1.0 rev 13	Summary of changes: 1. Added issuance parameter to "bymostrecent" command 2. Merged the "bymagazineID" and "bybooknumber" commands into "byID" 3. Add an expiration attribute on the credential when it is returned 4. Fix comment in KXO to indicate the correct key 5. Add an image element to the library information block. 6. Fix the "byauthorbegin" erratum 7. Add the List Series command
1.0 rev 14	Summary of changes: 1. Added an explanatory note regarding the `itemURL` in 112 General Format of `<resultitem>` Section 2. Corrected the examples in 15.5.2 By Book or Magazine ID 3. Added new call: Get User Guides (15.8) 4. Replaced login ID and password with credential in 16.1.10 Authorize Device. Added some explanatory text in that section.
1.0 rev 15	Summary of changes 1. Added section 8.1 Direct Command Arguments and added a heading for section 8.2 Optional Parameters. 2. Added `<uid>` element to specification of `<resultitem>` in sections 12.2 `<resultitem>` Section for a Book, 12.3 `<resultitem>` Section for a Magazine, and elsewhere as needed. 3. Corrected all sequence numbers to four digits. 4. Added Result code 7, Subcode 5: Request exceeds AO quota 5. Changed implementation schedule for 15.8 Get User Guides to unspecified future phase 6. Updated 16.1.10 Authorize Device response block

Table of Contents

Topic **Page**

Purpose

The purpose of this Database Design Documentation (DBDD) is to keep track of everything that happens between the fishermen collecting the fish to the Tampa Fishing Company and Market (TFCM) selling their fish to restaurants. As restaurants often purchase large quantities of fish over a given period of time, it’s important to document the amount of fish being processed through this entire process within an organized database.

Narrative

A Tampa Fishing Company and Market (TFCM) is trying to track all of its moving parts involved in the process of catching fish and selling it restaurants in order to gain insights about its business.

Fishermen who work for TFCM are tracked by a fisherman ID, name, address, email, phone, and salary (hourly rate). A fisherman contracts with a specific boat on a yearly basis. The fishing boats are tracked by Boat ID, and boat name. Some fishermen serve as captains of these boats and each fishing boat must have a captain on board when it leaves the dock. Each boat captain may have one or more fishermen to supervise but a fisherman does not have more than one captain.

A boat targets only one species of fish per trip and they release all the bycatch. Each fish has a unique fish ID, fish type and a price per pound. Each fishing trip is tracked by a unique fishing trip ID, the date leaving the port, the date returning, total days at sea and pounds of fish caught. The fishermen that go on these trips are paid by the number of hours on the boat. Trips are tracked by type of fish caught, and the amount of money generated from the catch (price per pound paid by Fishmonger).

After each trip, the fish are brought back and delivered to the fishmongers (fishmongers work for the market on salary) at the market who are responsible for cleaning and preparing the fish for sale to restaurants. The fishmongers are tracked by an employee ID, name (FN, LN), address, email, and phone number.

Restaurants buy fish in bulk from the fishmongers (price per pound paid by the restaurant). They are identified by restaurant ID, name, address, phone number and email. Restaurants can purchase one or more varieties of fish at a time. Restaurants can buy fish from multiple fishmongers and fishmongers can sell fish to multiple restaurants.

Requirements (Actors and Roles)

Fishermen: Fishermen work on one and only one boat per season. Fishermen may not work on a boat or may work on a different boat in a different season.

Boats: A boat must have a captain, and may have one or more than one fisherman who are not captains on the boat. Boats sell their fish to one or more than one Fishmonger.

Fish: Fish are caught by the fishermen and sold to the fishmonger.

Trip: A trip is taken by a boat and its crew to catch fish.

Fishmongers: Fishmongers clean and prepare the fish. They buy fish from one or more than one boat. They also sell the fish to restaurants.

Restaurants: Restaurants buy the fish from fishmongers.

Entities

- Fishermen
- Boats
- Fish
- Trip
- Fishmongers
- Restaurants

Entities w/ Nested Attributes

- Fishermen
    - Fisherman ID
    - Name
    - Address
    - E-mail
    - Phonenumber
    - Salary (hourly rate)
- Boats
    - Boat ID
    - Boat name
- Fish
    - Fish ID
    - Price per pound
- Trip
    - Fishing Trip ID
    - Date Left
    - Returning Date
    - Total days at sea.
- Fishmongers
    - Fishmonger ID
    - Name
    - Address
    - E-mail
    - Phone number
    - Salary
- Restaurants
    - Restaurant ID
    - Name
    - Address
    - Phone number
    - E-mail

Business Rules

Fishermen: Fishermen work on only one boat a year. Fishermen may also serve as a captain of a boat.

Boats: Boats are staffed by one and only one captain, and at least one or more than one fisherman.

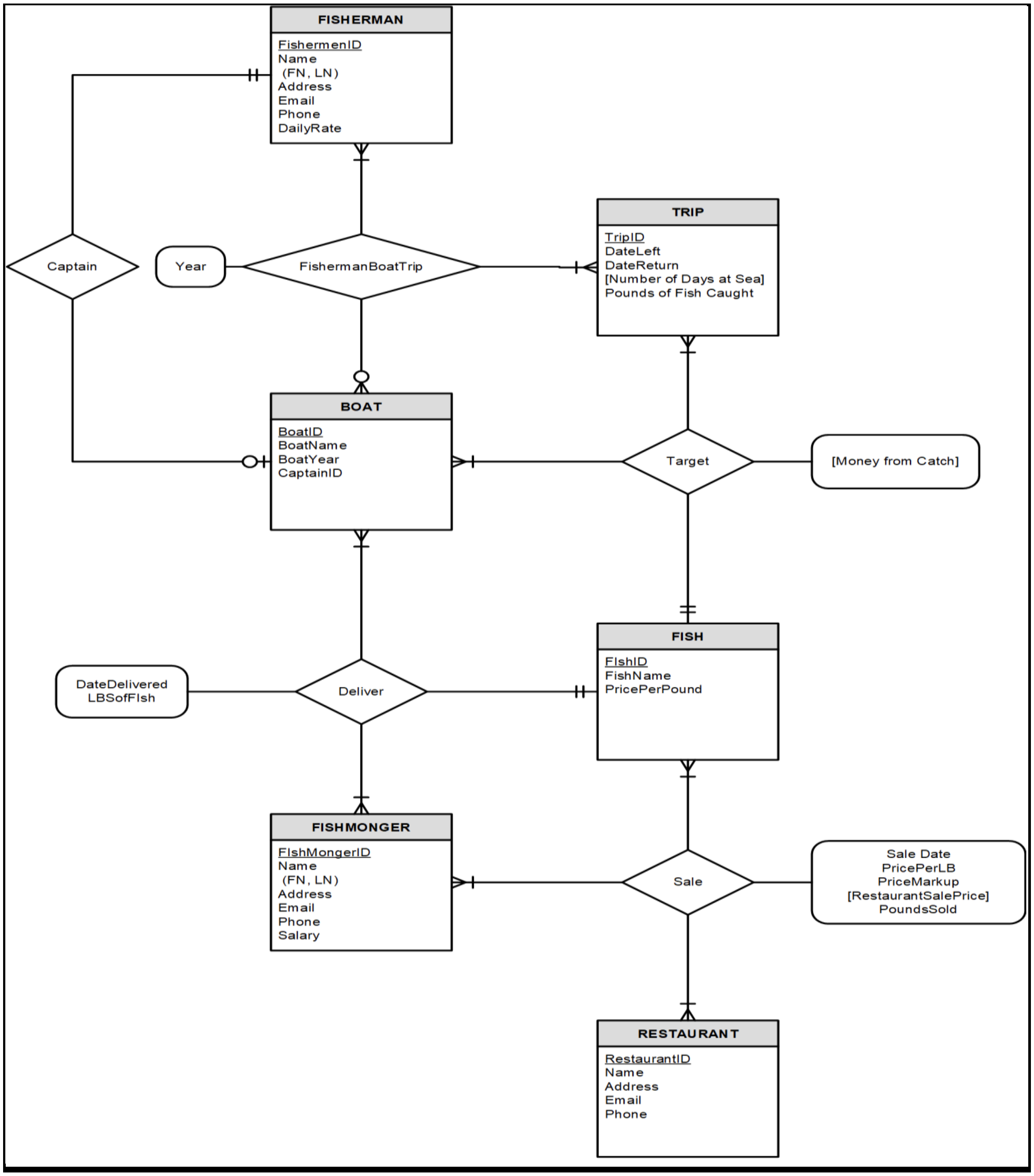
Fish: Fish are caught by the fishermen, and transported to Fishmongers via boat. There is one and only one type of fish caught by each boat.

Trip: Boats and Fishermen take trips to catch fish.

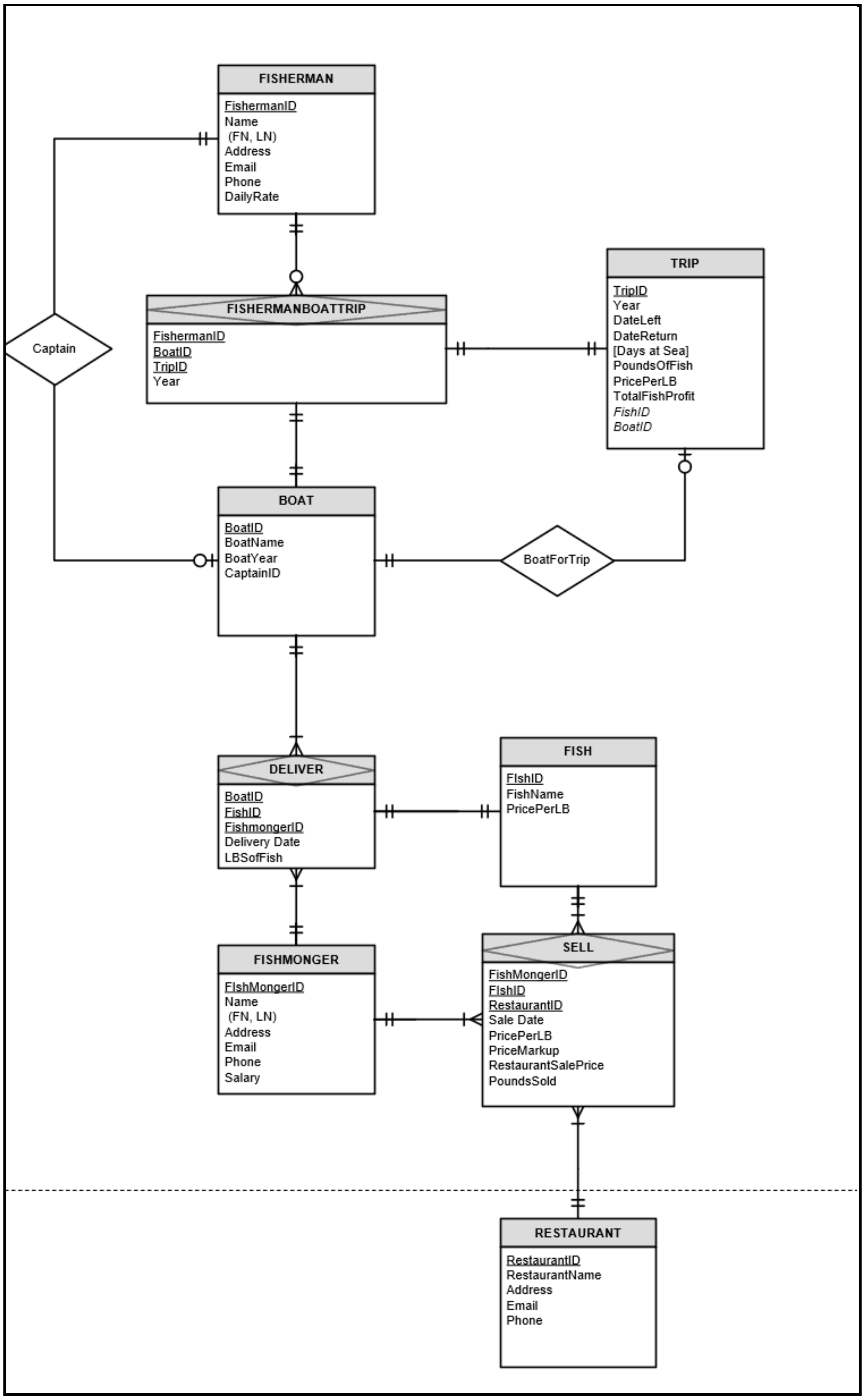
Fishmongers: Fishmongers purchase fish from one or more than one boat. Fishmongers sell fish to one or more than one restaurant.

Restaurants: Restaurants buy fish from one or more than one Fishmonger. Fishmongers sell fish to one or more than one restaurant.

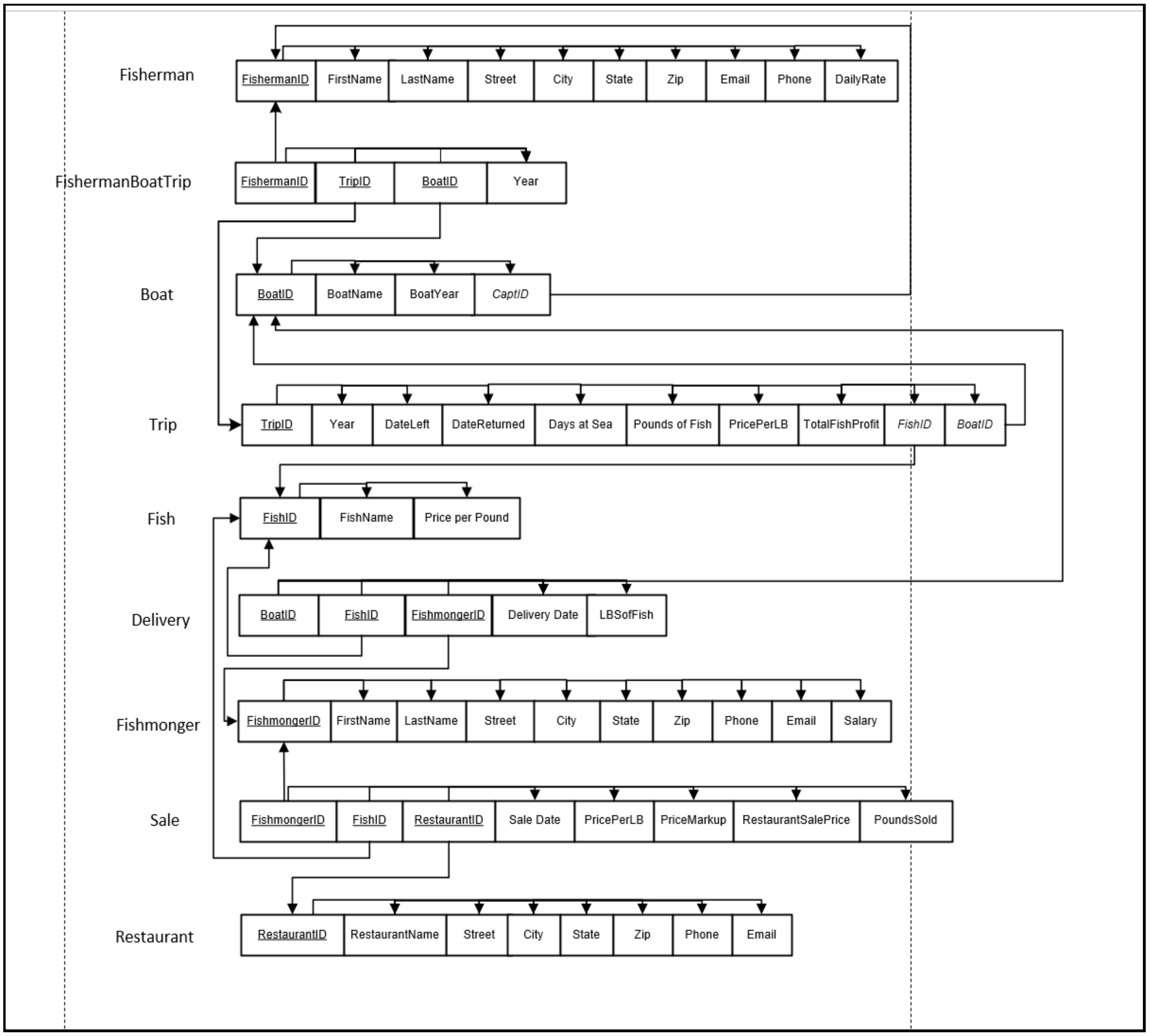
ERD


FISHERMAN
FishermenID
Name
(FN, LN)
Address
Email
Phone
DailyRate
TRIP
TripID
DateLeft
DateReturn
[Number of Days at Sea]
Pounds of Fish Caught
Captain
Year
FishermanBoatTrip
BOAT
BoatID
BoatName
BoatYear
CaptainID
Target
[Money from Catch]
FISH
FIshID
FishName
PricePerPound
DateDelivered
LBSofFIsh
Deliver
FISHMONGER
FIshMongerID
Name
(FN, LN)
Address
Email
Phone
Salary
Sale
Sale Date
PricePerLB
PriceMarkup
[RestaurantSalePrice]
PoundsSold
RESTAURANT
RestaurantID
Name
Address
Email
Phone

EERD


FISHERMAN
FishermanID
Name
(FN, LN)
Address
Email
Phone
DailyRate
Captain
FISHERMANBOATTRIP
FishermanID
BoatID
TripID
Year
TRIP
TripID
Year
DateLeft
DateReturn
[Days at Sea]
PoundsOfFish
PricePerLB
TotalFishProfit
FishID
BoatID
BOAT
BoatID
BoatName
BoatYear
CaptainID
BoatForTrip
DELIVER
BoatID
FishID
FishmongerID
Delivery Date
LBSofFish
FISH
FIshID
FishName
PricePerLB
FISHMONGER
FIshMongerID
Name
(FN, LN)
Address
Email
Phone
Salary
SELL
FishMongerID
FIshID
RestaurantID
Sale Date
PricePerLB
PriceMarkup
RestaurantSalePrice
PoundsSold
RESTAURANT
RestaurantID
RestaurantName
Address
Email
Phone

Relational Schema



Data Dictionary

Table: **Boat**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
BoatID	**PK**; Unique boat ID number	smallint		Y					Y
BoatName	Name of boat	varchar	20						Y
BoatYear	What year is the boat	char	4			‘Contact Fish Market’	([BoatYear] like '[0-9][0-9][0-9][0-9][0-9]')	Y	
CaptainID	**FK** to Captain	smallint					>=100	Y	

Table: **Deliver**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
BoatID	**CPK**; **FK** to Boat Table	smallint					>=100		Y
FishID	**CPK**; **FK** to Fish Table	tinyint							Y
FishmongerID	**CPK**; **FK** to Fishmonger Table	smallint				'Contact Fish Monger'			Y
DeliveryDate	Date of delivery	date							Y
LBSofFish	Number of pounds of fish delivery	smallint							

Table: **Fish**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
FishID	**PK**; Unique fish ID number								Y
FishName	Name of fish							Y	
PricePerLB	Price per pound of fish					'Contact Fish Monger'	>=1		

Table: **Fisherman**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
CustomerID	**PK**; Unique fisherman ID number	int		Y					Y
FirstName	Fisherman first name	varchar	20						
LastName	Fisherman last name	varchar	20						Y
Street	Fisherman street	varchar	20						
City	Fisherman city	varchar	20						

State	Fisherman state	char	2			‘FL’	([State] like '[A-Z][A-Z]')		
Zip	Fisherman zip	char	5				([Zip] like '[0-9][0-9][0-9][0-9][0-9]')		
Phone	Fisherman phone number	char	14						
DailyRate	smallmoney								

Table: **FishermanBoatTrip**

Column Name	**Description**	**Data Type**	**Size**	**Identity**	**Unique**	**Default**	**Check**	**Allow Nulls**	**Index**
FishermanID	**CPK**; **FK** to Fisherman Table	smallint					FishermanID >=100		Y
BoatID	**CPK**; **FK** to Boat Table	smallint							Y
TripID	**CPK; FK** to Trip Table	smallint							Y
Year	Year of boat assignment	char	4			‘Contact Fish Market’	([Year] like '[0-9][0-9][0-9][0-9]')		

Table: **Fishmonger**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
FishmongerID	**PK**; Unique fishmonger ID number	smallint		Y					Y
FirstName	Fishmonger first name	varchar	20						
LastName	Fishmonger last name	varchar	20						
Street	Fishmonger street	varchar	20						
City	Fishmonger city	varchar	20						
State	Fishmonger state	char	2			‘FL’	([State] like '[A-Z][A-Z]')		
Zip	Fishmonger zip	char	5				([Zip] like '[0-9][0-9][0-9][0-9][0-9]')		
Email	Fishmonger email	varchar	20						

Phone	Fishmonger phone number	char	14						
Salary	Fishmonger salary	smallmoney							

Table: **Restaurant**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
RestaurantID	**PK**; Unique restaurant ID	int		Y					Y
RestaurantName	Restaurant name	varchar	20						
Street	Restaurant street	varchar	20						
City	Restaurant city	varchar	20						
State	Restaurant state	char	2						
Zip	Restaurant zip	char	5				([Zip] like '[0-9][0-9][0-9][0-9][0-9]')		
Email	Restaurant email	varchar	20						
Phone	Restaurant phone	varchar	14			'Contact HR'			

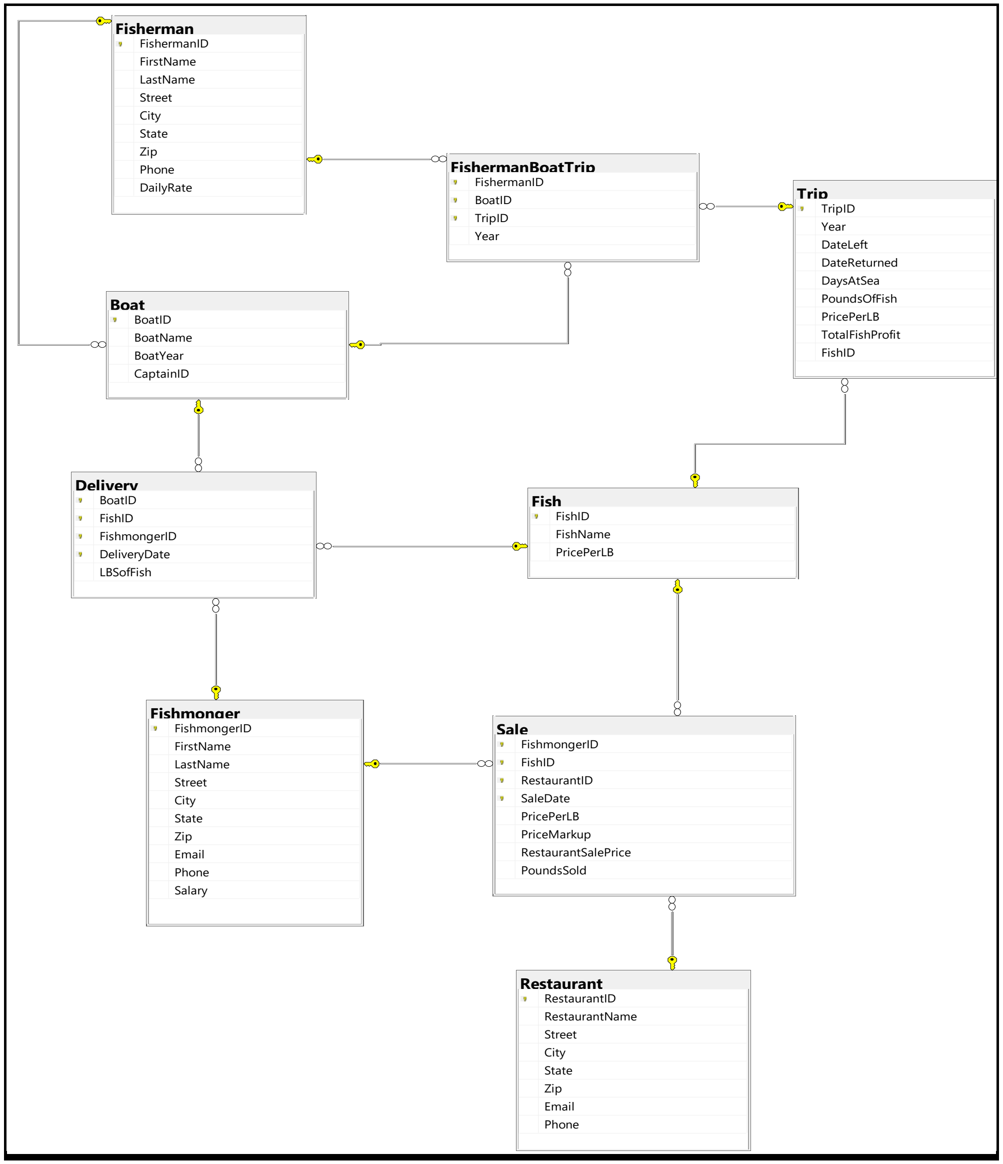
Table: **Sale**

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
FishmongerID	**CPK; FK** to Fishmonger table	smallint				'Contact Fish Monger'	FishmongerID>=100		
FishID	**CPK; FK** to Fish table	tinyiny							
RestaurantID	**CPK; FK** to Restaurant table	smallint							
SaleDate	Date of sale	date							
PricePerLB	Price per pounds	smallmoney							
PriceMarkup	Markup price of fish	decimal	(3,2)						
RestaurantsSalePrice	Sales of fish at restaurant								
PoundsSold	Number of pounds sold	smallint							

Table: **Trip**

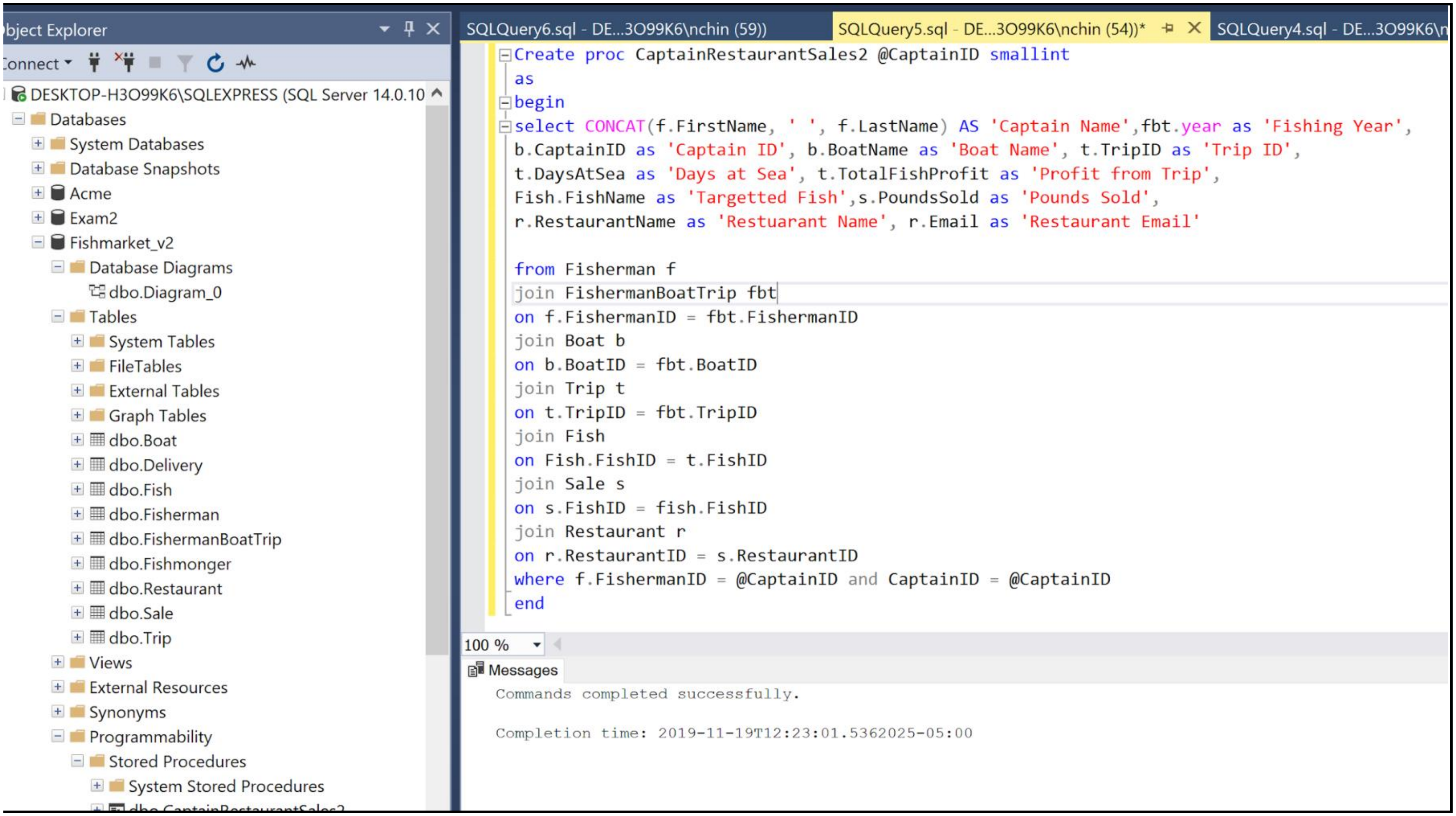
Column Name	**Description**	**Data Type**	**Size**	**Identity**	**Unique**	**Default**	**Check**	**Allow Nulls**	**Index**
TripID	**PK;** Unique trip ID number	int							Y
Year	Trip year	char	4				([Year] like '[0-9][0-9][0-9][0-9]')		
DateLeft	Date boat left for trip	date							
DateReturned	Date boat returned from trip	date					DateReturned > DateLeft		
DaysAtSea	Number of days at sea	computed							
PoundsOfFish	Number of pounds of fish caught	smallint				‘Contact Fish Monger’			
PricePerLB	Price of fish caught per pound	money							
TotalFishProfit	Total profit made from fish	computed							
FishID	**FK** to Fish table	tinyint							
BoatID	**FK** to Boat table	smallint							

Database Diagram



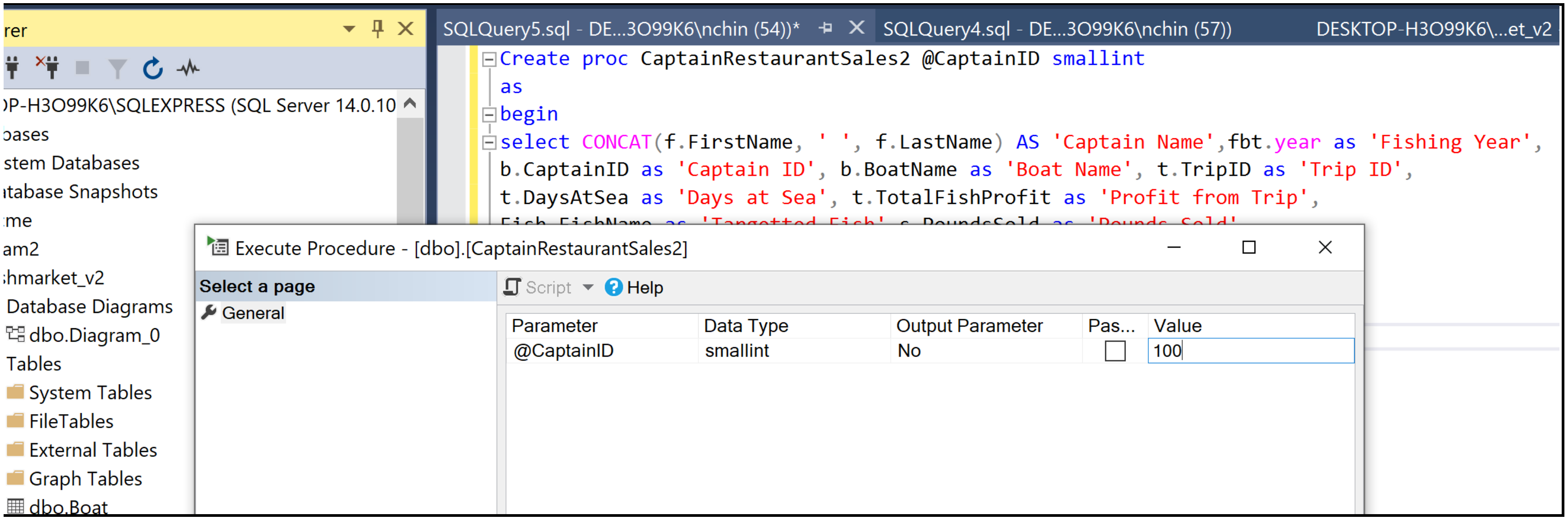
SPROCs

CaptainRestaurantSales SPROC



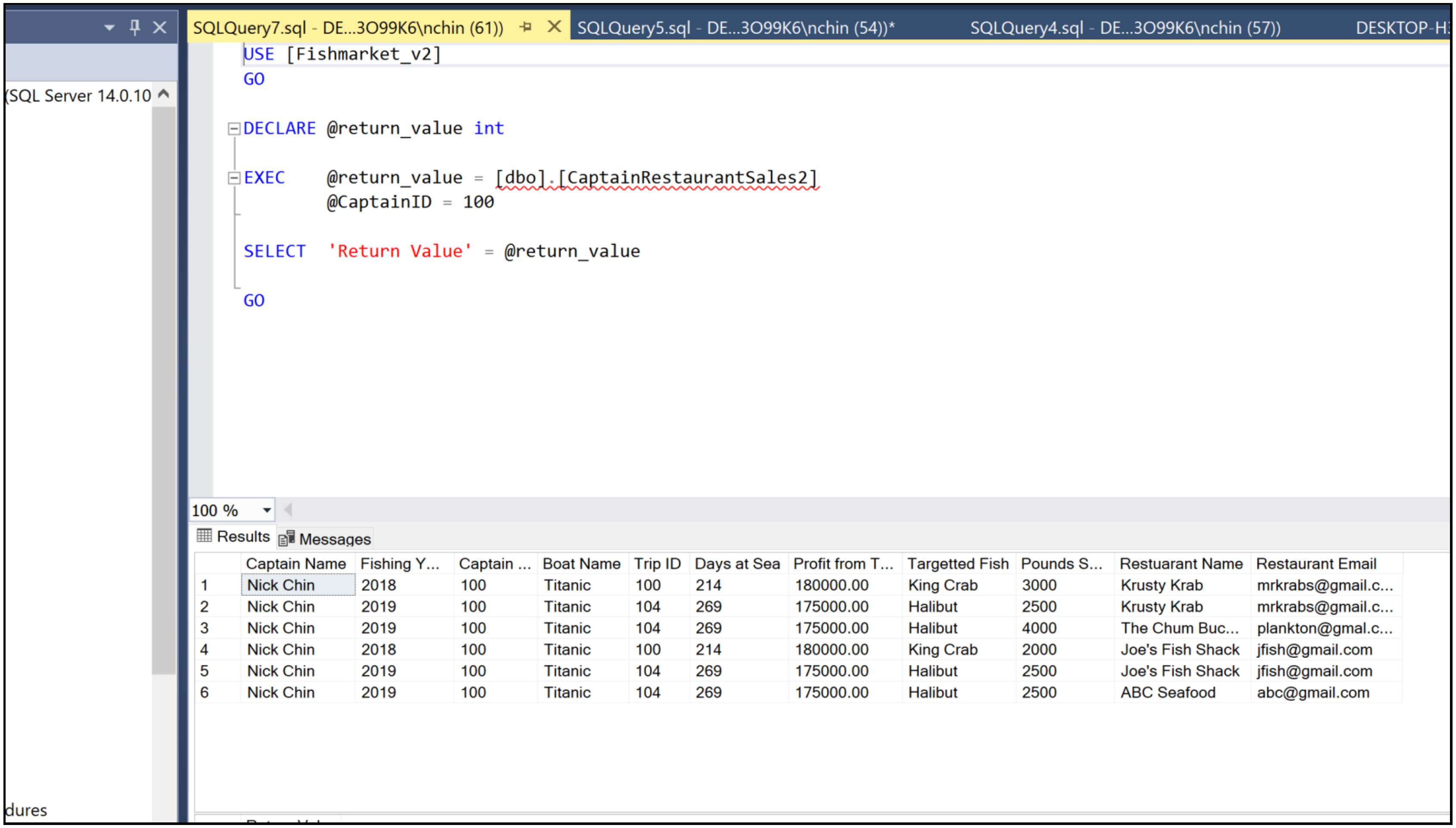
```sql
Create proc CaptainRestaurantSales2 @CaptainID smallint
as
begin
select CONCAT(f.FirstName, ' ', f.LastName) AS 'Captain Name',fbt.year as 'Fishing Year',
b.CaptainID as 'Captain ID', b.BoatName as 'Boat Name', t.TripID as 'Trip ID',
t.DaysAtSea as 'Days at Sea', t.TotalFishProfit as 'Profit from Trip',
Fish.FishName as 'Targetted Fish',s.PoundsSold as 'Pounds Sold',
r.RestaurantName as 'Restuarant Name', r.Email as 'Restaurant Email'

from Fisherman f
join FishermanBoatTrip fbt
on f.FishermanID = fbt.FishermanID
join Boat b
on b.BoatID = fbt.BoatID
join Trip t
on t.TripID = fbt.TripID
join Fish
on Fish.FishID = t.FishID
join Sale s
on s.FishID = fish.FishID
join Restaurant r
on r.RestaurantID = s.RestaurantID
where f.FishermanID = @CaptainID and CaptainID = @CaptainID
end
```

```
Commands completed successfully.

Completion time: 2019-11-19T12:23:01.5362025-05:00
```



```
Create proc CaptainRestaurantSales2 @CaptainID smallint
as
begin
select CONCAT(f.FirstName, ' ', f.LastName) AS 'Captain Name',fbt.year as 'Fishing Year',
b.CaptainID as 'Captain ID', b.BoatName as 'Boat Name', t.TripID as 'Trip ID',
t.DaysAtSea as 'Days at Sea', t.TotalFishProfit as 'Profit from Trip',
Fish.FishName as 'Targetted Fish', s.PoundsSold as 'Pounds Sold',
```

Parameter	Data Type	Output Parameter	Pas...	Value
@CaptainID	smallint	No	□	100



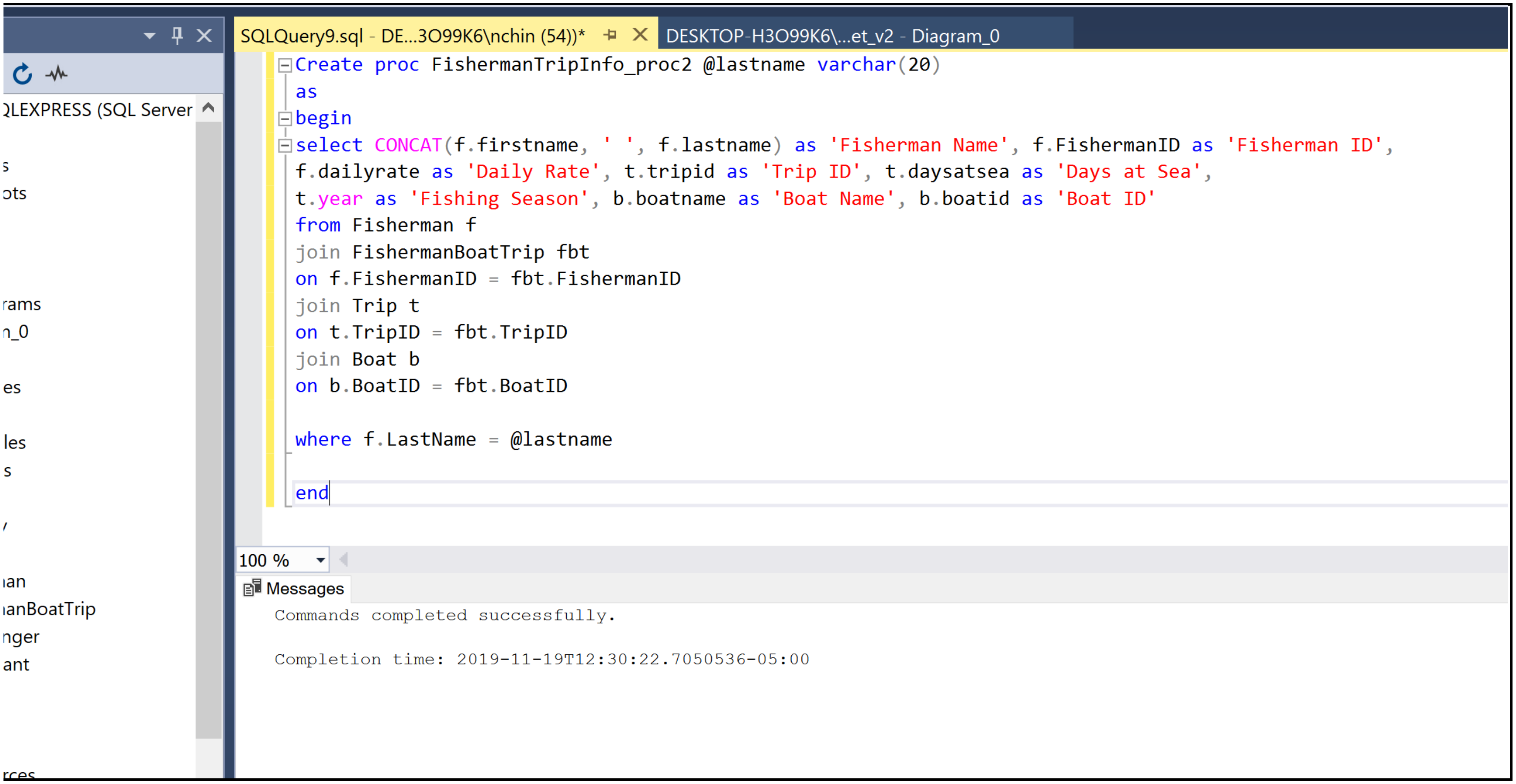
SQLQuery7.sql - DE...3O99K6\nchin (61))
SQLQuery5.sql - DE...3O99K6\nchin (54))*
SQLQuery4.sql - DE...3O99K6\nchin (57))
DESKTOP-H
(SQL Server 14.0.10
dures
100 %
Results
Messages

```sql
USE [Fishmarket_v2]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[CaptainRestaurantSales2]
        @CaptainID = 100

SELECT  'Return Value' = @return_value

GO
```

	Captain Name	Fishing Y...	Captain ...	Boat Name	Trip ID	Days at Sea	Profit from T...	Targetted Fish	Pounds S...	Restuarant Name	Restaurant Email
1	Nick Chin	2018	100	Titanic	100	214	180000.00	King Crab	3000	Krusty Krab	mrkrabs@gmail.c...
2	Nick Chin	2019	100	Titanic	104	269	175000.00	Halibut	2500	Krusty Krab	mrkrabs@gmail.c...
3	Nick Chin	2019	100	Titanic	104	269	175000.00	Halibut	4000	The Chum Buc...	plankton@gmal.c...
4	Nick Chin	2018	100	Titanic	100	214	180000.00	King Crab	2000	Joe's Fish Shack	jfish@gmail.com
5	Nick Chin	2019	100	Titanic	104	269	175000.00	Halibut	2500	Joe's Fish Shack	jfish@gmail.com
6	Nick Chin	2019	100	Titanic	104	269	175000.00	Halibut	2500	ABC Seafood	abc@gmail.com

The CaptainRestaurantSales Sproc pulls together columns from the Boat, Trip, Fish, and Restaurant tables. It easily brings up the fish caught on a trip, the profit made by the boat, what restaurant/s bought the fish and how much they purchased. There is a parameter on this sproc: the captain ID which is identified via a small integer.

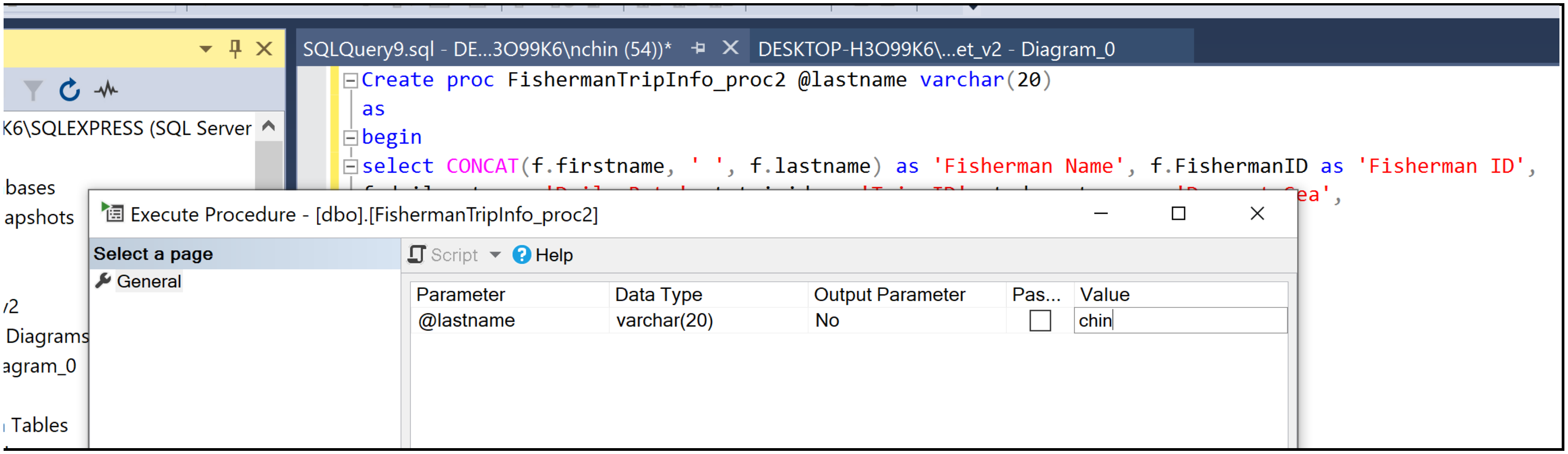
FishermanTrip SPROC



SQLQuery9.sql - DE...3O99K6\nchin (54))*
DESKTOP-H3O99K6\...et_v2 - Diagram_0
QLEXPRESS (SQL Server
s
ots
rams
n_0
es
les
s
ian
ianBoatTrip
nger
ant
rces
100 %
Messages

```
Create proc FishermanTripInfo_proc2 @lastname varchar(20)
as
begin
select CONCAT(f.firstname, ' ', f.lastname) as 'Fisherman Name', f.FishermanID as 'Fisherman ID',
f.dailyrate as 'Daily Rate', t.tripid as 'Trip ID', t.daysatsea as 'Days at Sea',
t.year as 'Fishing Season', b.boatname as 'Boat Name', b.boatid as 'Boat ID'
from Fisherman f
join FishermanBoatTrip fbt
on f.FishermanID = fbt.FishermanID
join Trip t
on t.TripID = fbt.TripID
join Boat b
on b.BoatID = fbt.BoatID

where f.LastName = @lastname

end
```

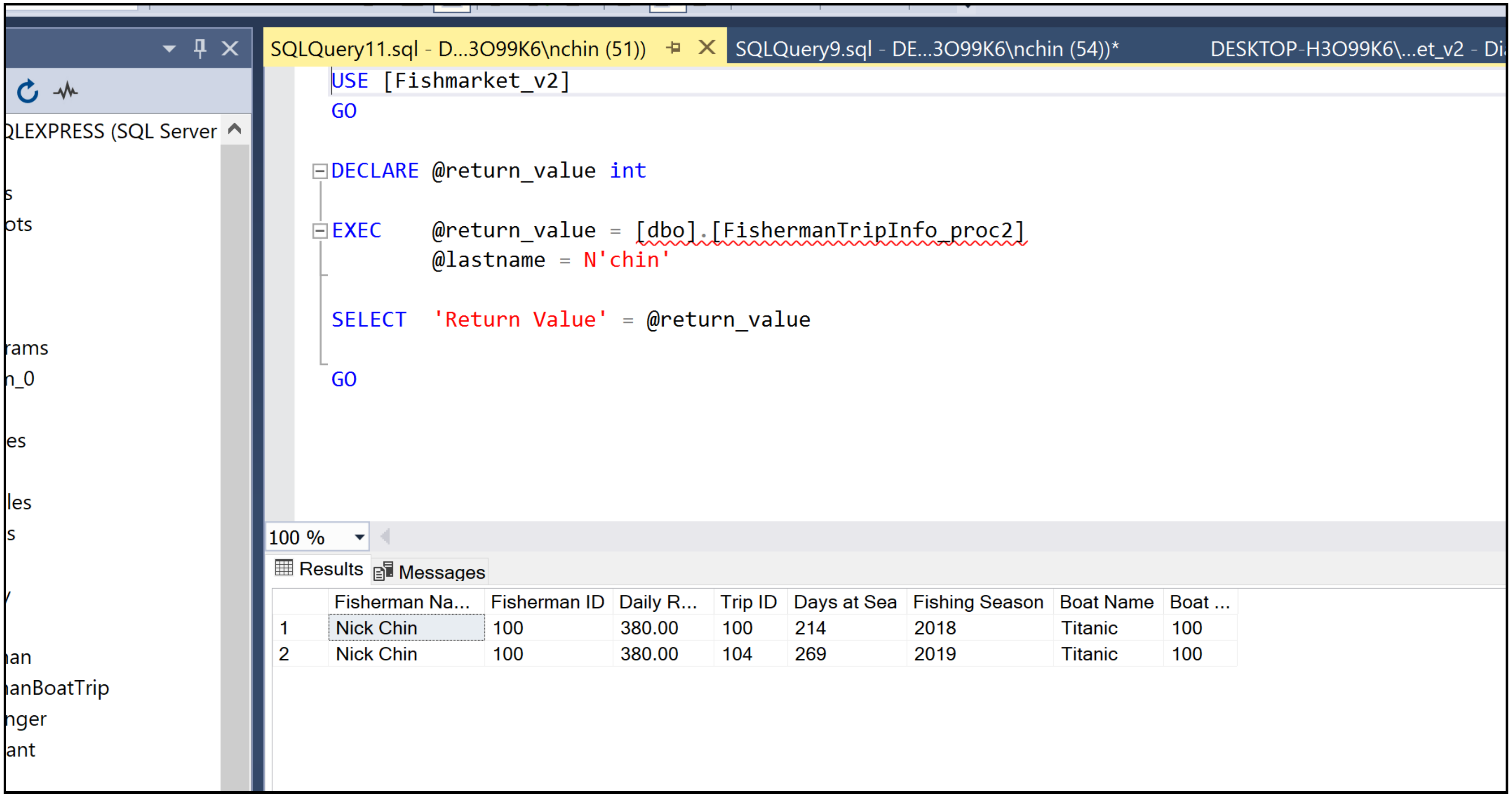
```
Commands completed successfully.

Completion time: 2019-11-19T12:30:22.7050536-05:00
```


SQLQuery9.sql - DE...3O99K6\nchin (54))*
DESKTOP-H3O99K6\...et_v2 - Diagram_0
K6\SQLEXPRESS (SQL Server
bases
apshots
/2
Diagrams
agram_0
Tables
Select a page
General
Script
Help

```
Create proc FishermanTripInfo_proc2 @lastname varchar(20)
as
begin
select CONCAT(f.firstname, ' ', f.lastname) as 'Fisherman Name', f.FishermanID as 'Fisherman ID',
                                                                                     ea',
```

Execute Procedure - [dbo].[FishermanTripInfo_proc2]

Parameter	Data Type	Output Parameter	Pas...	Value
@lastname	varchar(20)	No	□	chin

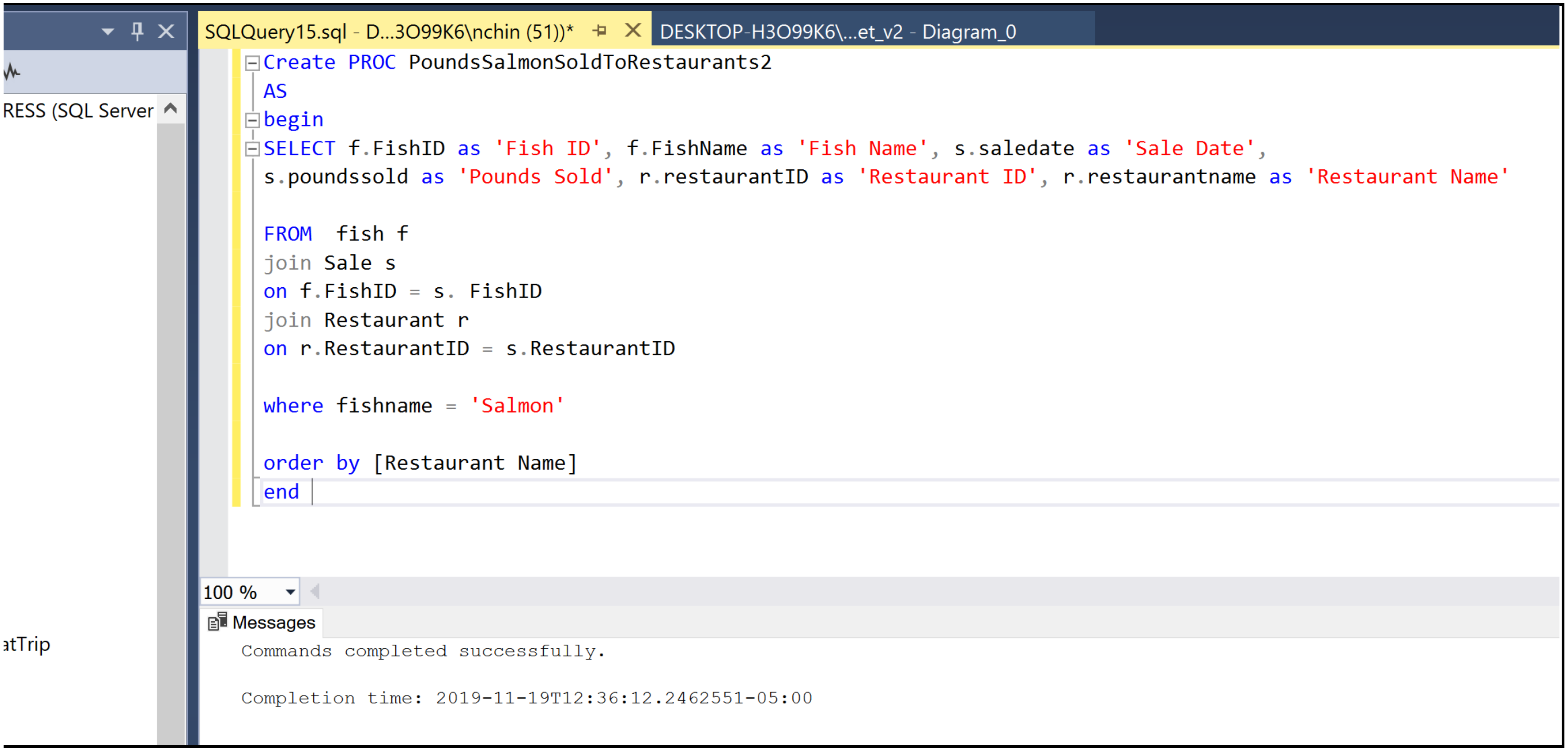

SQLQuery11.sql - D...3O99K6\nchin (51))
SQLQuery9.sql - DE...3O99K6\nchin (54))*
DESKTOP-H3O99K6\...et_v2 - Di
LEXPRESS (SQL Server
s
ots
rams
n_0
es
les
s
/
an
anBoatTrip
nger
ant
100 %
Results
Messages

```
USE [Fishmarket_v2]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[FishermanTripInfo_proc2]
        @lastname = N'chin'

SELECT  'Return Value' = @return_value

GO
```

	Fisherman Na...	Fisherman ID	Daily R...	Trip ID	Days at Sea	Fishing Season	Boat Name	Boat ...
1	Nick Chin	100	380.00	100	214	2018	Titanic	100
2	Nick Chin	100	380.00	104	269	2019	Titanic	100

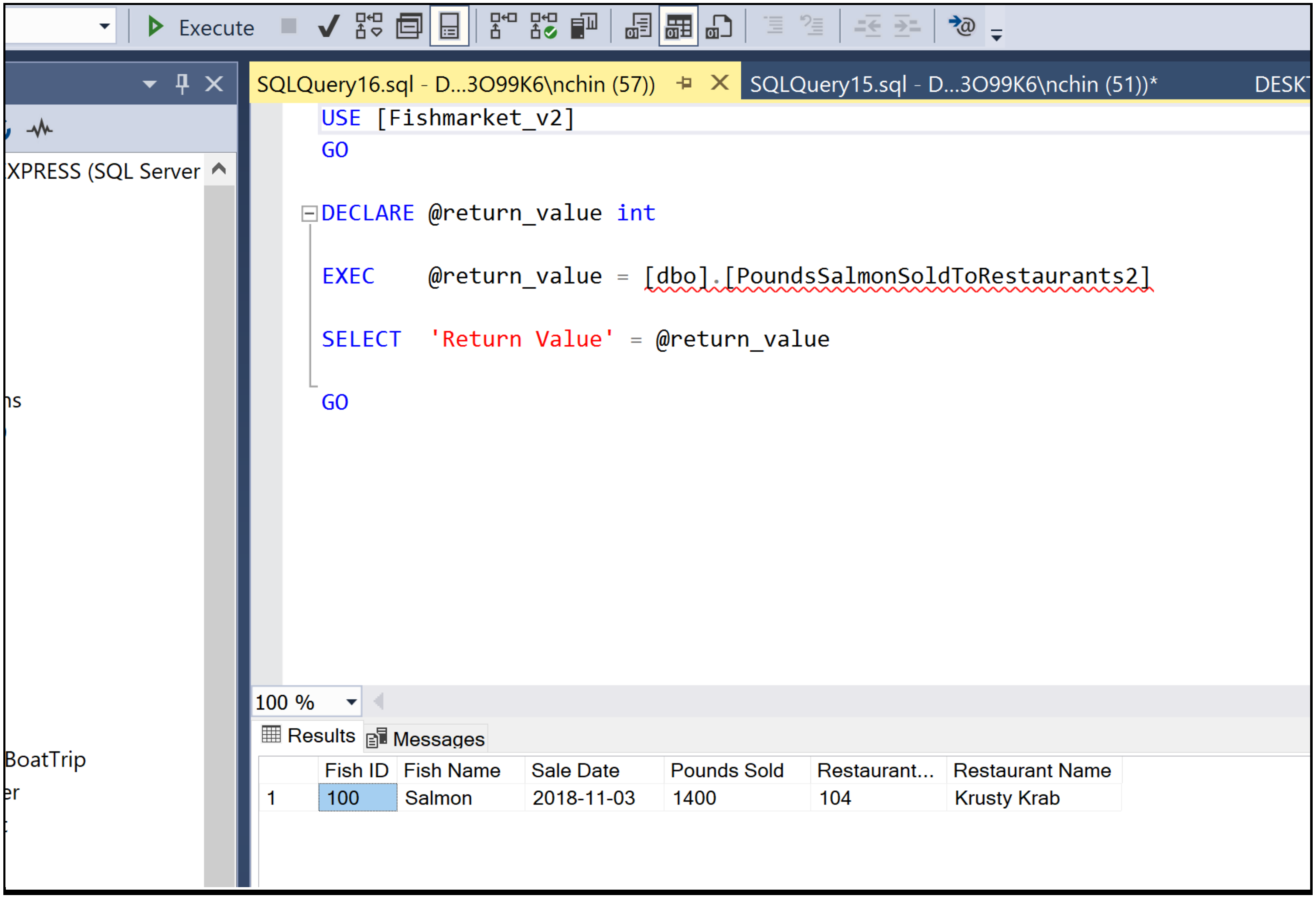
The FishermanTrip Sproc pulls data from the fisherman, trip and boat tables. The goal of this sproc is to pull up information regarding a trip a fisherman worked, what boat he/she was on, how long, and the daily wage for the fisherman. This sproc also has the fisherman last name as a parameter.

PoundsSalmonSold SPROC


SQLQuery15.sql - D...3O99K6\nchin (51))*
DESKTOP-H3O99K6\...et_v2 - Diagram_0
RESS (SQL Server
atTrip
100 %
Messages

```
Create PROC PoundsSalmonSoldToRestaurants2
AS
begin
SELECT f.FishID as 'Fish ID', f.FishName as 'Fish Name', s.saledate as 'Sale Date',
s.poundssold as 'Pounds Sold', r.restaurantID as 'Restaurant ID', r.restaurantname as 'Restaurant Name'

FROM  fish f
join Sale s
on f.FishID = s. FishID
join Restaurant r
on r.RestaurantID = s.RestaurantID

where fishname = 'Salmon'

order by [Restaurant Name]
end
```

```
Commands completed successfully.

Completion time: 2019-11-19T12:36:12.2462551-05:00
```


Execute
SQLQuery16.sql - D...3O99K6\nchin (57))
SQLQuery15.sql - D...3O99K6\nchin (51))*
DESK
XPRESS (SQL Server
ns
BoatTrip
er
100 %
Results
Messages

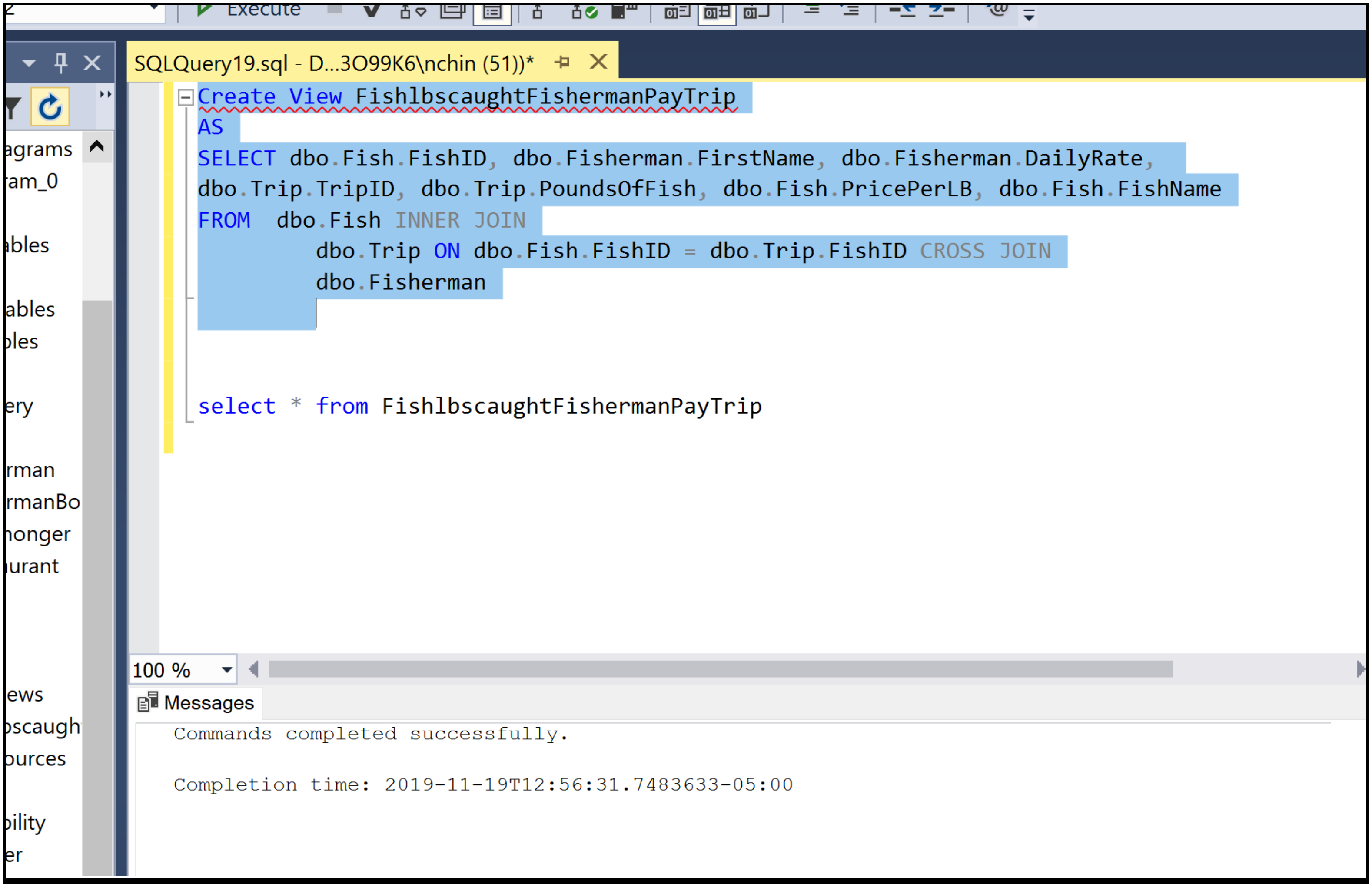
```
USE [Fishmarket_v2]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[PoundsSalmonSoldToRestaurants2]

SELECT  'Return Value' = @return_value

GO
```

	Fish ID	Fish Name	Sale Date	Pounds Sold	Restaurant...	Restaurant Name
1	100	Salmon	2018-11-03	1400	104	Krusty Krab

The PoundsofSalmonSold Sproc pulls data from the fish and restaurant table. It allows the user to see how much salmon was sold to what restaurant.

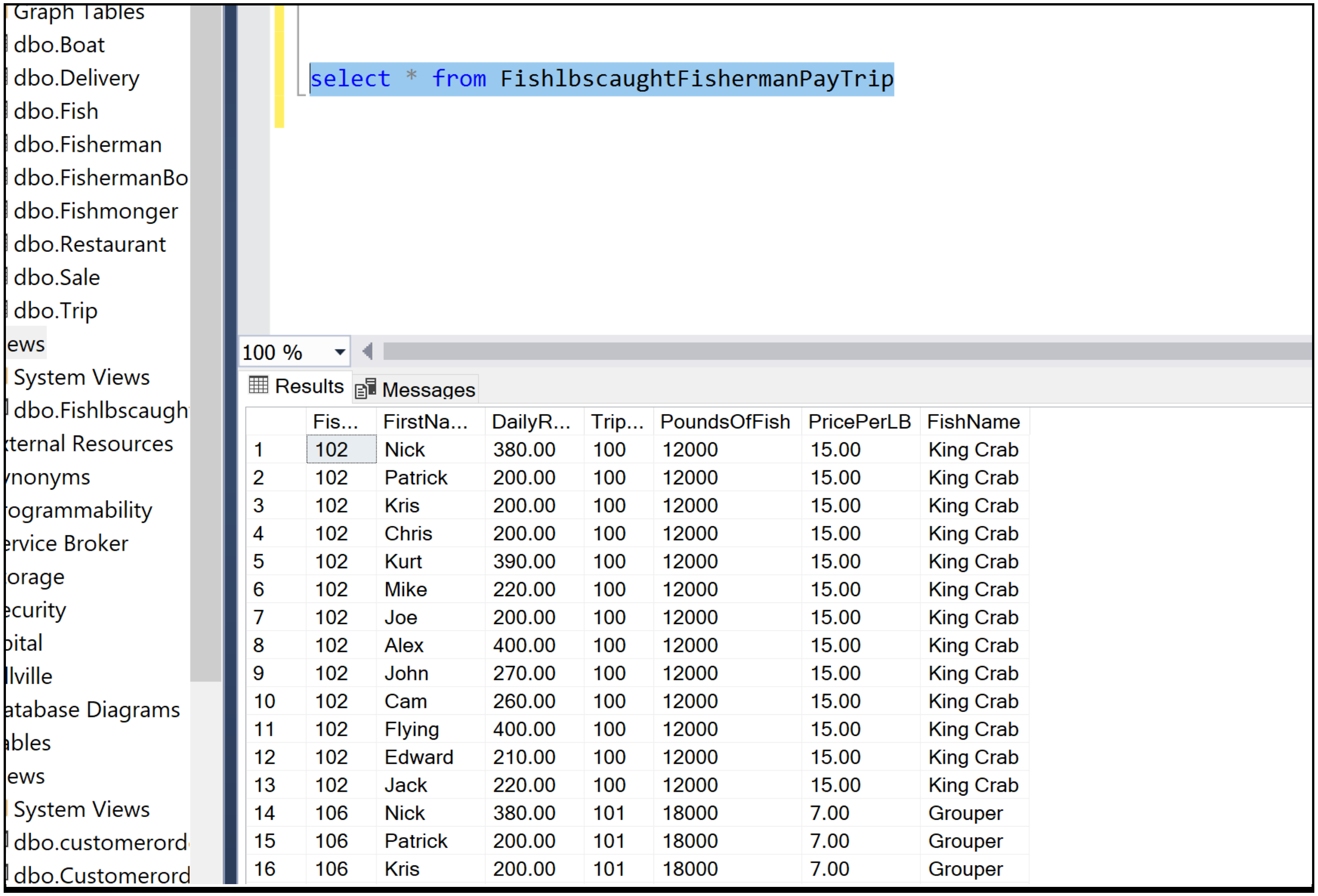
VIEWS

Lbs of Fish Caught/Fisherman Trip/Pay View



```
Create View FishlbscaughtFishermanPayTrip
AS
SELECT dbo.Fish.FishID, dbo.Fisherman.FirstName, dbo.Fisherman.DailyRate,
dbo.Trip.TripID, dbo.Trip.PoundsOfFish, dbo.Fish.PricePerLB, dbo.Fish.FishName
FROM  dbo.Fish INNER JOIN
         dbo.Trip ON dbo.Fish.FishID = dbo.Trip.FishID CROSS JOIN
         dbo.Fisherman


select * from FishlbscaughtFishermanPayTrip
```

```
Commands completed successfully.

Completion time: 2019-11-19T12:56:31.7483633-05:00
```

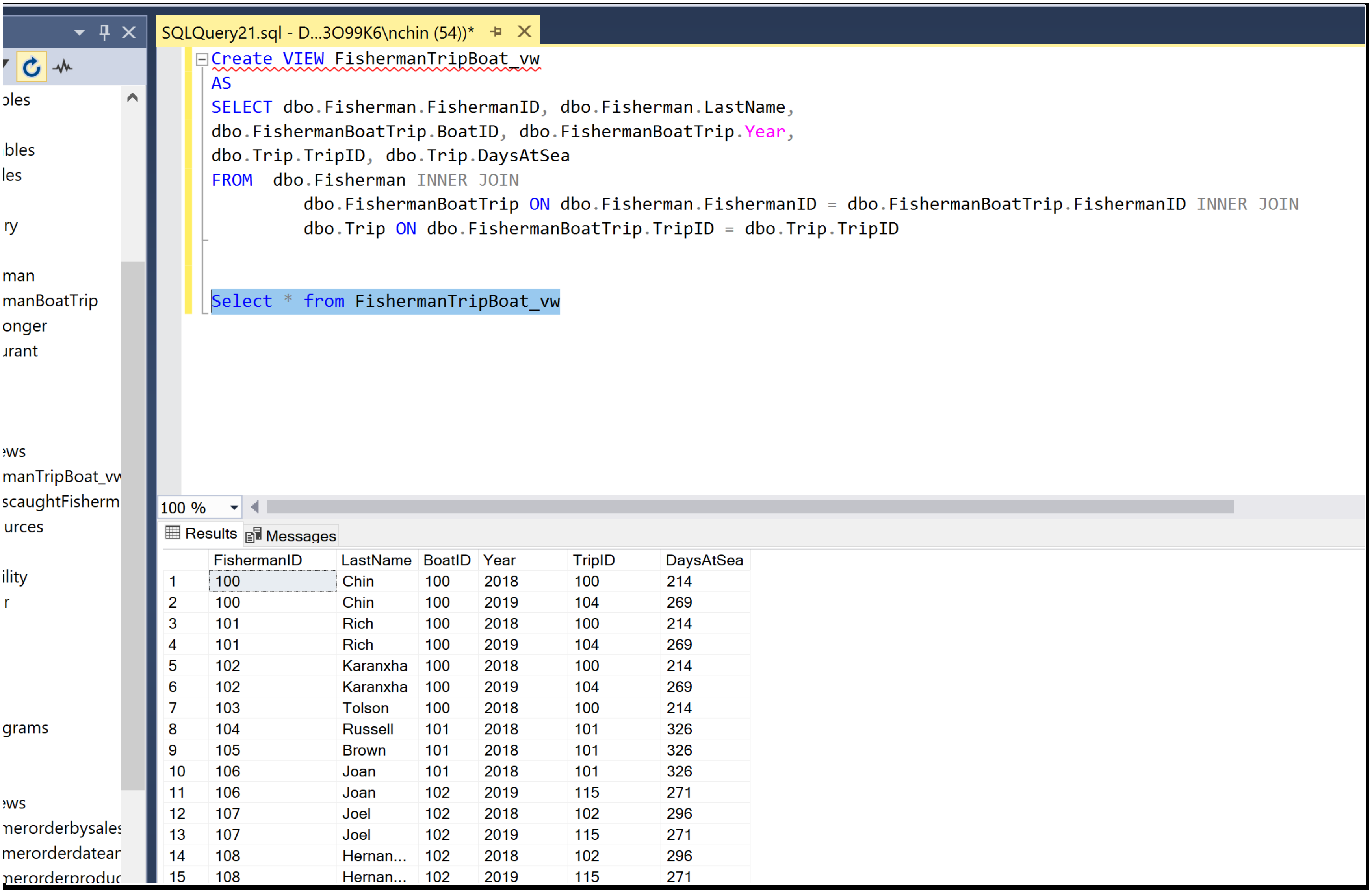


```
select * from FishlbscaughtFishermanPayTrip
```

	Fis...	FirstNa...	DailyR...	Trip...	PoundsOfFish	PricePerLB	FishName
1	102	Nick	380.00	100	12000	15.00	King Crab
2	102	Patrick	200.00	100	12000	15.00	King Crab
3	102	Kris	200.00	100	12000	15.00	King Crab
4	102	Chris	200.00	100	12000	15.00	King Crab
5	102	Kurt	390.00	100	12000	15.00	King Crab
6	102	Mike	220.00	100	12000	15.00	King Crab
7	102	Joe	200.00	100	12000	15.00	King Crab
8	102	Alex	400.00	100	12000	15.00	King Crab
9	102	John	270.00	100	12000	15.00	King Crab
10	102	Cam	260.00	100	12000	15.00	King Crab
11	102	Flying	400.00	100	12000	15.00	King Crab
12	102	Edward	210.00	100	12000	15.00	King Crab
13	102	Jack	220.00	100	12000	15.00	King Crab
14	106	Nick	380.00	101	18000	7.00	Grouper
15	106	Patrick	200.00	101	18000	7.00	Grouper
16	106	Kris	200.00	101	18000	7.00	Grouper

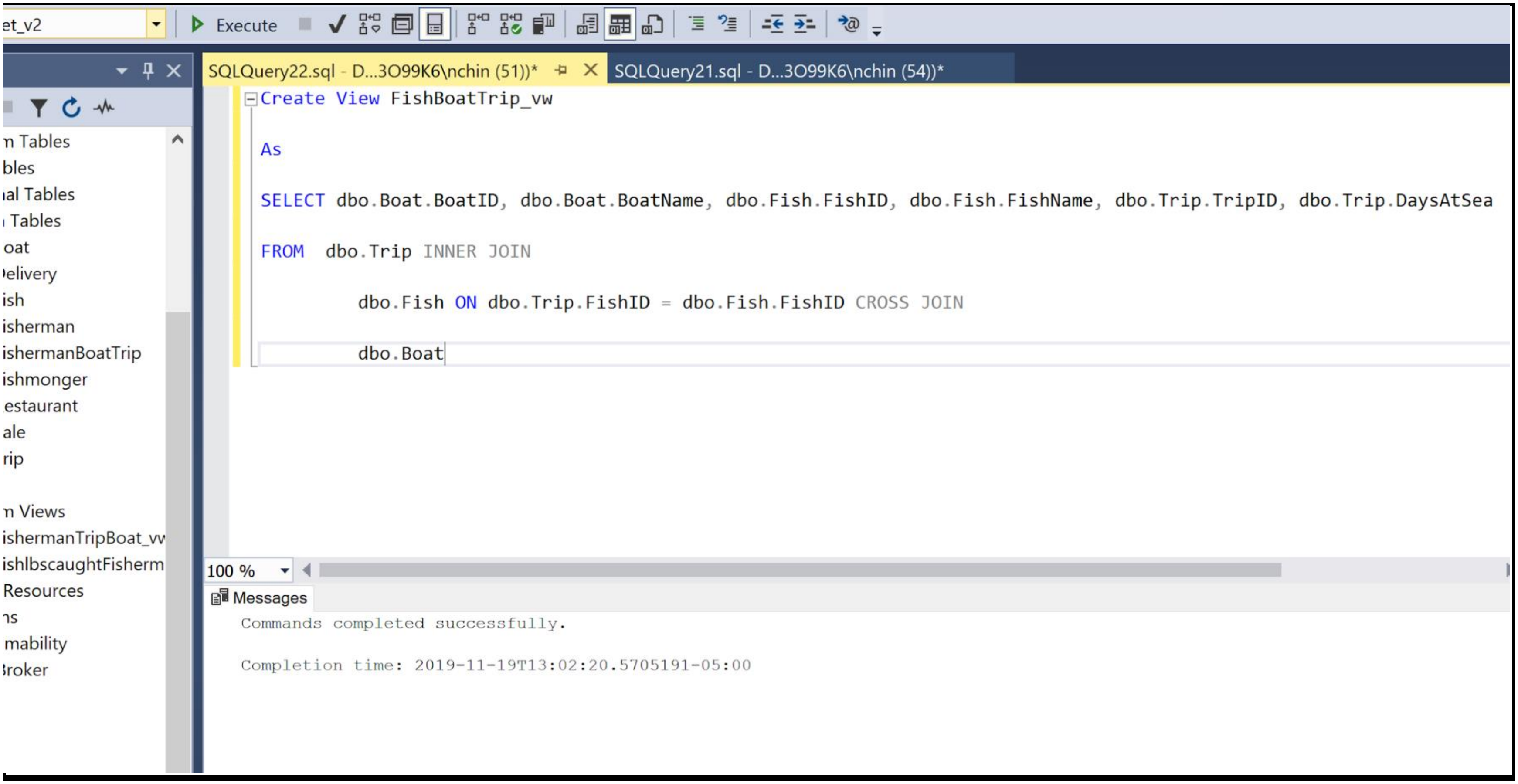
The lbs of fish caught/fisherman/trip view pulls data from the fish, fishermen and trip tables in order to give the user a read out of what fish were caught by fishermen, and the fishermen's daily pay rate.

FishermanTripBoat View



```
Create VIEW FishermanTripBoat_vw
AS
SELECT dbo.Fisherman.FishermanID, dbo.Fisherman.LastName,
dbo.FishermanBoatTrip.BoatID, dbo.FishermanBoatTrip.Year,
dbo.Trip.TripID, dbo.Trip.DaysAtSea
FROM  dbo.Fisherman INNER JOIN
          dbo.FishermanBoatTrip ON dbo.Fisherman.FishermanID = dbo.FishermanBoatTrip.FishermanID INNER JOIN
          dbo.Trip ON dbo.FishermanBoatTrip.TripID = dbo.Trip.TripID


Select * from FishermanTripBoat_vw
```

	FishermanID	LastName	BoatID	Year	TripID	DaysAtSea
1	100	Chin	100	2018	100	214
2	100	Chin	100	2019	104	269
3	101	Rich	100	2018	100	214
4	101	Rich	100	2019	104	269
5	102	Karanxha	100	2018	100	214
6	102	Karanxha	100	2019	104	269
7	103	Tolson	100	2018	100	214
8	104	Russell	101	2018	101	326
9	105	Brown	101	2018	101	326
10	106	Joan	101	2018	101	326
11	106	Joan	102	2019	115	271
12	107	Joel	102	2018	102	296
13	107	Joel	102	2019	115	271
14	108	Hernan...	102	2018	102	296
15	108	Hernan...	102	2019	115	271

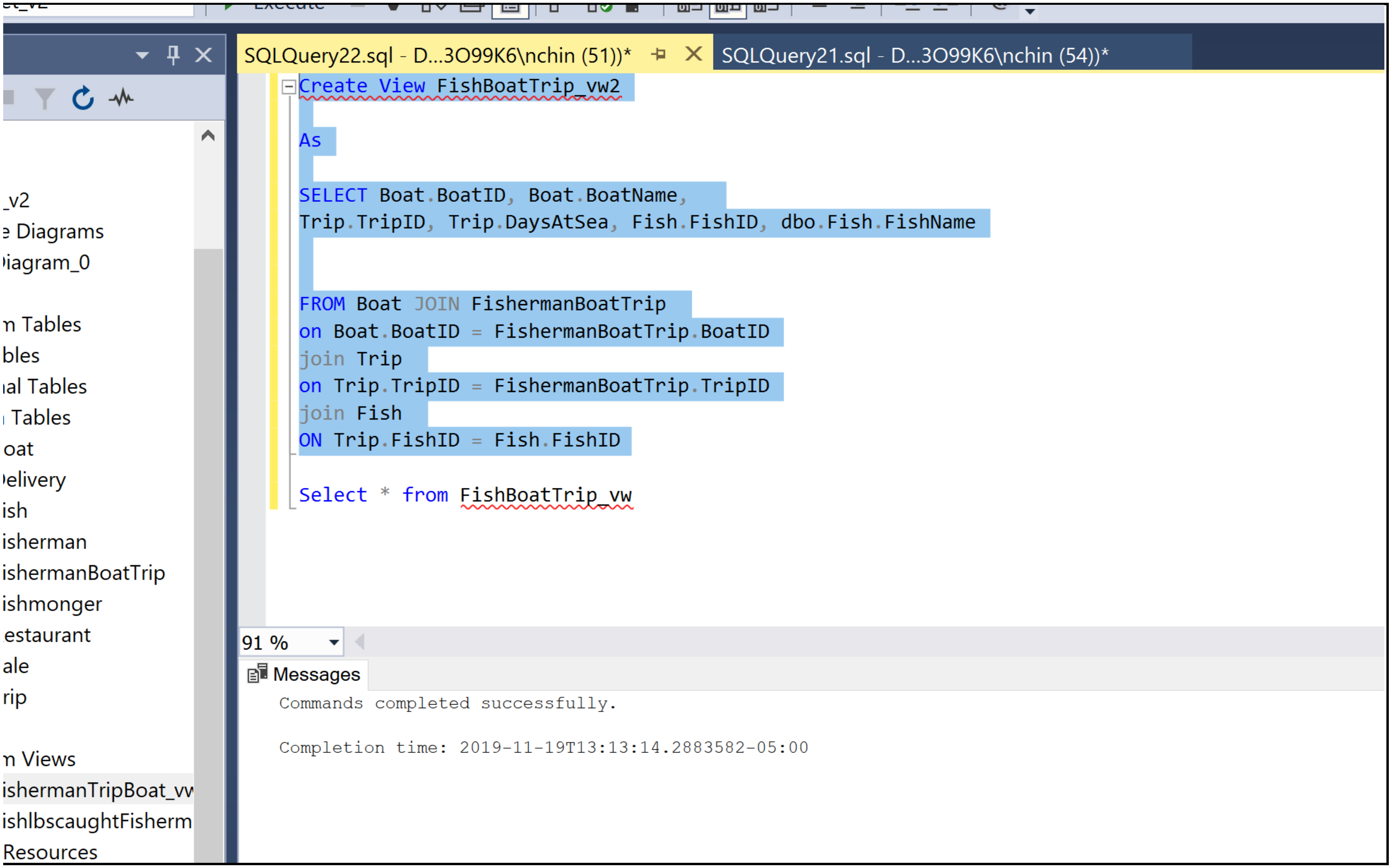


```
Create View FishBoatTrip_vw

As

SELECT dbo.Boat.BoatID, dbo.Boat.BoatName, dbo.Fish.FishID, dbo.Fish.FishName, dbo.Trip.TripID, dbo.Trip.DaysAtSea

FROM  dbo.Trip INNER JOIN

          dbo.Fish ON dbo.Trip.FishID = dbo.Fish.FishID CROSS JOIN

          dbo.Boat
```

```
Commands completed successfully.

Completion time: 2019-11-19T13:02:20.5705191-05:00
```

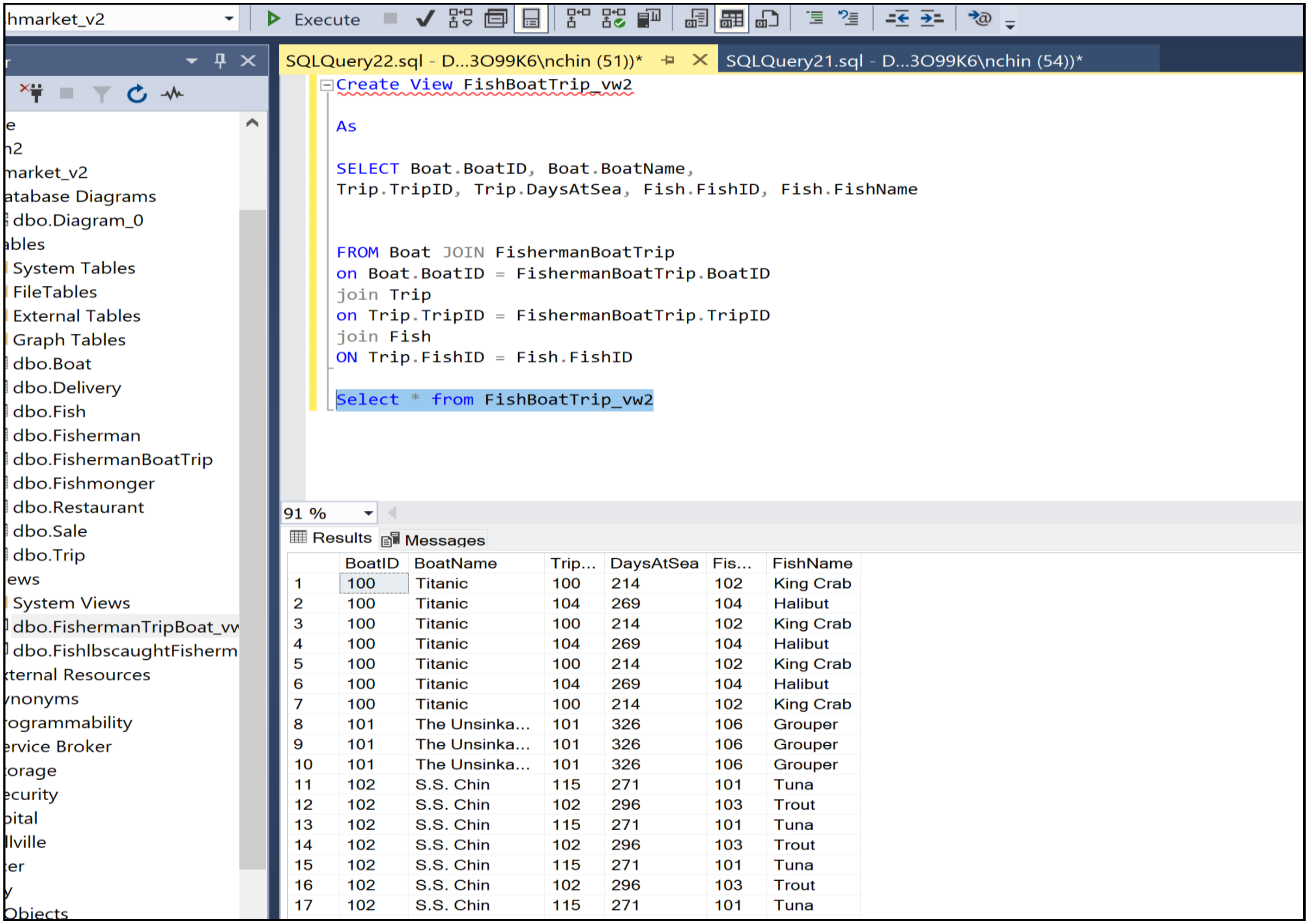
The FishermanTripBoat view pulls data from the fisherman, trip and boat tables. The view allows the user to create a table of what boat a fisherman was on and for how many days the fisherman and boat were at sea.

FishBoatTrip View



```
Create View FishBoatTrip_vw2

As

SELECT Boat.BoatID, Boat.BoatName,
Trip.TripID, Trip.DaysAtSea, Fish.FishID, dbo.Fish.FishName


FROM Boat JOIN FishermanBoatTrip
on Boat.BoatID = FishermanBoatTrip.BoatID
join Trip
on Trip.TripID = FishermanBoatTrip.TripID
join Fish
ON Trip.FishID = Fish.FishID

Select * from FishBoatTrip_vw
```

```
Commands completed successfully.

Completion time: 2019-11-19T13:13:14.2883582-05:00
```



```
Create View FishBoatTrip_vw2

As

SELECT Boat.BoatID, Boat.BoatName,
Trip.TripID, Trip.DaysAtSea, Fish.FishID, Fish.FishName


FROM Boat JOIN FishermanBoatTrip
on Boat.BoatID = FishermanBoatTrip.BoatID
join Trip
on Trip.TripID = FishermanBoatTrip.TripID
join Fish
ON Trip.FishID = Fish.FishID

Select * from FishBoatTrip_vw2
```

	BoatID	BoatName	Trip...	DaysAtSea	Fis...	FishName
1	100	Titanic	100	214	102	King Crab
2	100	Titanic	104	269	104	Halibut
3	100	Titanic	100	214	102	King Crab
4	100	Titanic	104	269	104	Halibut
5	100	Titanic	100	214	102	King Crab
6	100	Titanic	104	269	104	Halibut
7	100	Titanic	100	214	102	King Crab
8	101	The Unsinka...	101	326	106	Grouper
9	101	The Unsinka...	101	326	106	Grouper
10	101	The Unsinka...	101	326	106	Grouper
11	102	S.S. Chin	115	271	101	Tuna
12	102	S.S. Chin	102	296	103	Trout
13	102	S.S. Chin	115	271	101	Tuna
14	102	S.S. Chin	102	296	103	Trout
15	102	S.S. Chin	115	271	101	Tuna
16	102	S.S. Chin	102	296	103	Trout
17	102	S.S. Chin	115	271	101	Tuna

The FishBoatTrip View pulls data from the fish, trip and boat tables. It allows the user to view the type of fish caught by a boat on a specific trip.