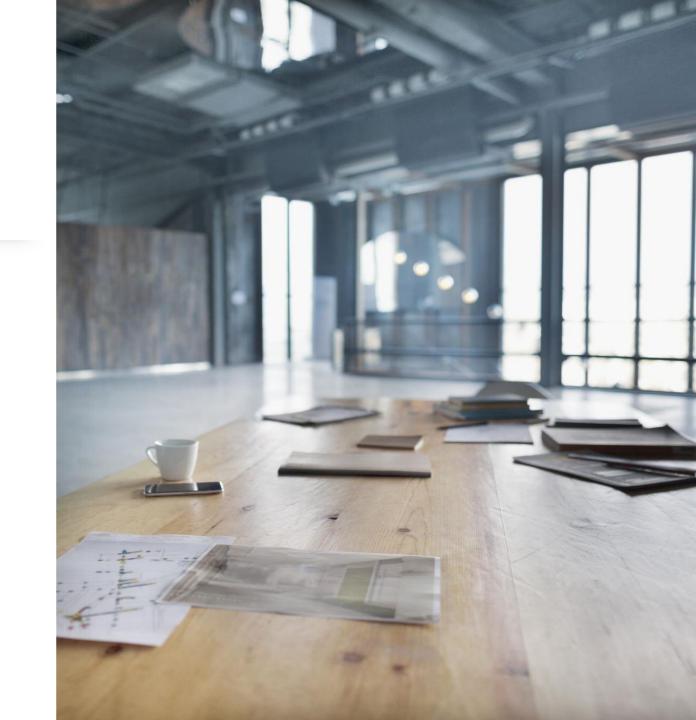
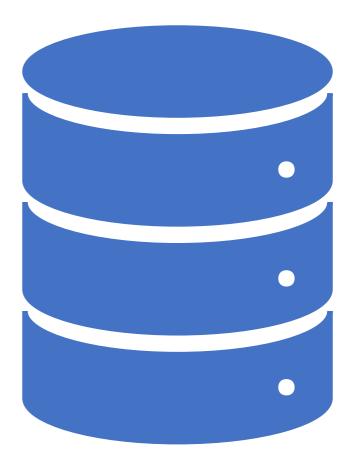


Session's Objective

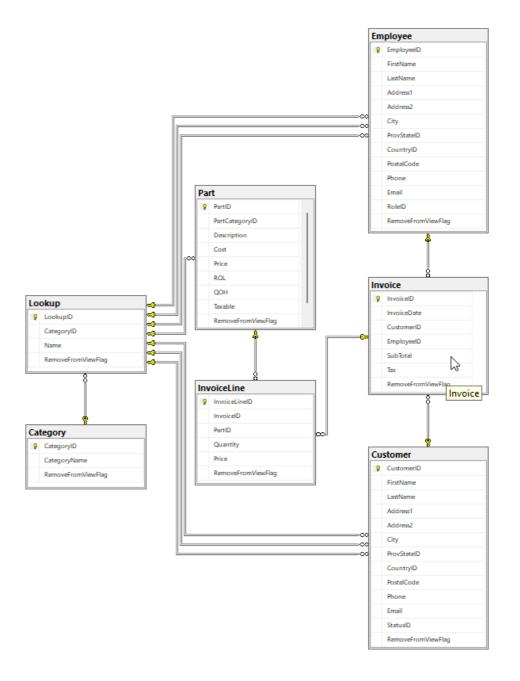
- Understanding the Hog-Wild Database Schema.
- Looking at the requirements of the client regarding interacting with customer data.
- Delving deep into the Customer table.
- Introduction to tailored models for effective customer data operations.



Hog-Wild Database Schema.



Database Schema



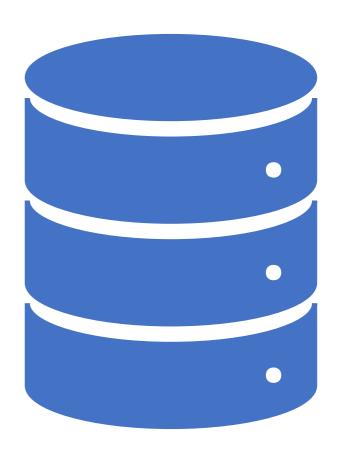
Category Table

- CategoryID: Identifier for each category.
 (Primary Key)
- CategoryName: Name of the category.
- RemoveFromViewFlag: Indicates whether to remove from view or not. The default value is 0 (false).
- Category Name is the parent reference for the "Lookup" data.

	CategoryID	CategoryName	RemoveFromViewFlag
1	1	Province	0
2	2	Country	0
3	3	Customer Status	0
4	4	Employee Role	0
5	5	Part Categories	0

Importance of Soft Delete (Remove From View Flag)

- **1. Data Recovery**: Enables easy restoration of accidentally deleted records, reducing potential data loss consequences.
- **2. Audit Trail**: Maintains a historical record of data changes, aiding in understanding user behaviors and actions over time.
- **3. Minimize Disruptions**: Prevents immediate permanent changes, allowing for continuity in ongoing tasks and reducing potential errors or disruptions.
- **4. Compliance and Regulations**: Helps meet data retention standards in industries with strict regulations regarding data deletion.



Lookup Table

- •LookupID: Identifier for each lookup. (Primary Key)
- •CategoryID: References a Category ID to categorize the lookup.
- •Name: Name of the lookup.
- •RemoveFromViewFlag: Indicates whether to remove from view or not. The default value is 0 (false).
- •Lookup references Category via CategoryID.
- •Uses as lookup references.

	LookupID	CategoryID	Name	RemoveFromViewFlag
1	1	1	Ontario	0
2	2	1	Quebec	0
3	3	1	Nova Scotia	0
4	4	1	New Brunswick	0
5	5	1	Manitoba	0
6	6	1	British Columbia	0
7	7	1	Prince Edward Island	0
8	8	1	Saskatchewan	0
9	9	1	Alberta	0
10	10	1	Newfoundland and Labrador	0
11	11	2	Canada	0
12	12	2	USA	0
13	13	2	Mexico	0
14	14	3	Bronze	0
15	15	3	Silver	0
16	16	3	Gold	0
17	17	3	Platinum	0
18	18	4	Sale Person	0
19	19	4	Parts Person	0
20	20	4	Manager	0
21	21	4	Owner	0
22	22	5	Accessories	0
23	23	5	Parts	0
24	24	5	Appearal	0
25	25	5	Servicing	0

Customer Table

- CustomerID: Identifier for each customer. (Primary Key)
- FirstName, LastName: Customer's first and last names.
- Address1, Address2: Customer's addresses. Address2 is optional.
- **City**: City of the customer.
- **ProvStateID**: References a Lookup ID that corresponds to the province/state of the customer.
- CountryID: References a Lookup ID corresponding to the customer's country.
- PostalCode: Postal code of the customer.
- **Phone, Email**: Contact details of the customer.
- **StatusID**: References a Lookup ID to indicate the status of the customer.
- **RemoveFromViewFlag**: Indicates whether to remove from view or not. The default value is 0 (false).
- The **Customer** table reference **Lookup** using CountryID, ProvStateID, and StatusID.

	CustomerID	FirstName	LastName	Address1	Address2	City	ProvStateID	CountryID	PostalCode	Phone	Email	StatusID	RemoveFromViewFlag
1	1	Sam	Smith	12345 - 67 St		Edmonton	9	11	T5J1X1	780444444	ssmith@hotmail.com	14	0
2	2	John	Jones	23456 - 78 St		Edmonton	9	11	T5J1X2	7804322222	jjones@hotmail.com	14	0
3	3	Vem	Green	11111 - 33 St		Edmonton	9	11	T5J1X3	7804333333	vgreen@hotmail.com	14	0
4	4	Dave	Davids	22222 - 22 St		Edmonton	9	11	T5J1X4	7804334344	ddavids@hotmail.com	14	0
5	5	Abe	Appleby	33333 - 33 St		Edmonton	9	11	T5J1X5	7804324545	aappleby@hotmail.com	14	0
6	6	Fred	Foster	44444 - 44 St		Edmonton	9	11	T5J2X5	7804326565	ffoster@hotmail.com	14	0
7	7	Fem	Forest	55555 - 55 St		Edmonton	9	11	T5J3X2	7804327676	fforest@hotmail.com	14	0
8	8	Roger	Doger	66666 - 66 St		Edmonton	9	11	T5J4X5	7804323547	rdoger@hotmail.com	14	0
9	9	Filly	Ford	77777 - 77 St		Edmonton	9	11	T5J5X4	7804324756	fford@hotmail.com	14	0
10	10	Dan	Decker	88888 - 88 St		Edmonton	9	11	T5J6X3	7804326768	ddecker@hotmail.com	14	0
11	13	SHylizhu	Peraehetuca	20 Lomotedi St	South of Flag Pole	Xaexusaetysu	9	11	T0A1A2	83264362656	shylizhu.peraehetuca@maeqokyja.com	14	0

Employee Table

•EmployeeID: Identifier for each employee. (Primary Key)

•FirstName, LastName: Customer's first and last names.

•Address1, Address2: Customer's addresses. Address2 is optional.

•City: City of the customer.

•ProvStateID: References a Lookup ID that corresponds to the province/state of the customer.

•CountryID: References a Lookup ID corresponding to the customer's country.

•PostalCode: Postal code of the customer.

•Phone, Email: Contact details of the customer.

•RoleID: References a Lookup ID to indicate the employee's role.

•RemoveFromViewFlag: Indicates whether to remove from view or not. The default value is 0 (false).

•The **Employee** table reference **Lookup** using CountryID, ProvStateID, and RoleID.

	EmployeeID	Flyt Name	LastName	Address1	Address2	City	ProvStateID	CountryID	PostalCode	Phone	Email	RoleID	RemoveFromViewFlag
1	1	Nole	Body	3215 - 66 St		Edmonton	9	11	T5J1X1	780.432.9876	nbody@eBikes.com	18	0
2	2	Willie	Work	12345 - 67 St		Edmonton	9	11	T5J1X1	780.444.3535	wwork@eBikes.com	18	0
3	3	Ken	Fixit	23232 - 55 St		Edmonton	9	11	T5J2X2	780.444.3636	kfixit@eBikes.com	18	0
4	4	Brad	Brake	3434 - 56 St		Edmonton	9	11	T5J2X3	780.555.3637	bblake@eBikes.com	18	0
5	5	Brian	Brown	4545 - 57 St		Edmonton	9	11	T5J2X4	780.666.3638	bbrown@eBikes.com	18	0
6	6	Bob	Brooks	5656 - 56 St		Edmonton	9	11	T5J2X5	780.777.3639	bbrooks@eBikes.com	18	0
7	7	Freda	Flash	4354 - 55 St		Edmonton	9	11	T5J2X6	780.765.3630	fflash@eBikes.com	18	0
8	8	Sadie	Star	6565 - 55 St		Edmonton	9	11	T5J2X7	780.654.3631	sstar@eBikes.com	18	0
9	9	Max	Smart	6776 - 55 St		Edmonton	9	11	T5J2X8	780.543.3632	msmart@eBikes.com	18	0
10	10	Sheila	Seller	7887 - 55 St		Edmonton	9	11	T5J2X9	780.432.3633	sseller@eBikes.com	18	0
11	11	Shirly	Ujest	7887 - 55 St		Edmonton	9	11	T5J2X9	780.953.3633	sujest@eBikes.com	18	0
12	12	Brain	Byer	1452 - 155 St		Edmonton	9	11	T5K8X9	780.953.3666	bbyer@eBikes.com	18	0

Part Table

•PartID: Identifier for each part. (Primary Key)

•PartCategoryID: Identifier for categorizing parts.

•Description: Description of the part.

•Cost, Price: Cost and sale price of the part.

•ROL: Reorder Level for the part.

•QOH: Quantity on Hand.

•Taxable: Indicates whether the part is taxable or not. The default value is 1 (true)

•RemoveFromViewFlag: Indicates whether to remove from view or not. The default value is 0 (false)

•The Part tables reference Lookup using PartCategoryID.

	PartID	PartCategoryID	Description	Cost	Price	ROL	QOH	Taxable	RemoveFromViewFlag
1	9	23	Air filter	10.00	25.00	80	10	1	0
2	41	23	Air filter	10.00	25.00	80	11	1	0
3	47	23	Air filter	10.00	25.00	80	11	1	0
4	18	24	Aviator goggle	45.00	75.00	10	15	1	0
5	54	22	Beep Beep Hom	65.00	70.00	85	65	1	0
6	27	25	Brake Oil, pint	3.65	4.95	70	125	1	0
7	52	22	Brass Hom	45.00	50.00	55	45	1	0
8	32	23	bulbs, front	2.95	4.25	100	185	1	0
9	31	23	bulbs, running	1.95	2.65	100	185	1	0
10	30	23	bulbs, signal	2.95	3.65	100	200	1	0

Invoice Table

- •InvoiceID: Identifier for each invoice. (Primary Key)
- •InvoiceDate: Date of the invoice.
- •CustomerID: References a Customer ID indicating which customer the invoice belongs to.
- •EmployeeID: References an Employee ID indicating which employee processed the invoice.
- •SubTotal, Tax: Monetary details of the invoice.
- •RemoveFromViewFlag: Indicates whether to remove from view or not. Default value is 0 (false).
- •Invoice references Customer via CustomerID and Employee via EmployeeID.

	InvoiceID	InvoiceDate	CustomerID	EmployeeID	SubTotal	Tax	RemoveFromViewFlag
1	1	2023-08-02	2	8	3061.00	153.05	0
2	2	2023-08-02	8	2	2063.95	103.20	0
3	3	2023-08-02	2	9	2501.20	125.06	0
4	4	2023-08-02	1	6	8388.25	419.41	0
5	5	2023-08-02	10	3	8477.20	423.86	0
6	6	2023-08-02	6	11	1980.00	99.00	0
7	7	2023-08-02	7	11	300.00	15.00	0
8	8	2023-08-02	7	1	3858.40	192.92	0
9	9	2023-08-03	6	8	2653.15	132.66	0
10	10	2023-08-03	1	6	3314.60	165.73	0

InvoiceLine Table

- •InvoiceLineID is the identifier for each invoice line. (Primary Key)
- •InvoiceID: References an Invoice ID indicating which invoice this line belongs to.
- •PartID: References a Part ID to indicate the part being invoiced.
- •Quantity: Quantity of the part.
- •Price: Price of the part.
- •RemoveFromViewFlag: Indicates whether to remove from view or not. The default value is 0 (false).
- •InvoiceLine references Invoice via InvoiceID and Part via PartID.

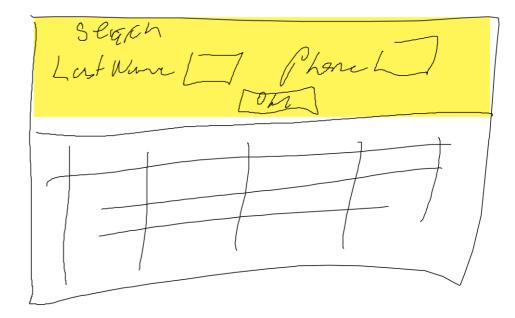
_	InvoiceLineID	InvoiceID	PartID	Quantity	Price	RemoveFromViewFlag
1/3	1	1	27	5	4.95	0
2	2	1	53	2	60.00	0
3	3	1	51	17	40.00	0
4	4	1	5	7	60.00	0
5	5	1	38	5	60.00	0
6	6	1	18	7	75.00	0
7	7	1	32	5	4.25	0
8	8	1	2	5	50.00	0
9	9	1	17	8	90.00	0
10	10	2	54	4	70.00	0
11	11	2	11	7	220.00	0

Customer Requirements

Customer Search

- Customer Search (Minimum of 1 field required)
 - Last Name (partial name)
 - Phone Number (partial)
- Search Button
- New Button (Requirement Added Afterwards) - Calls "Customer Edit Screen" for creating new customers.

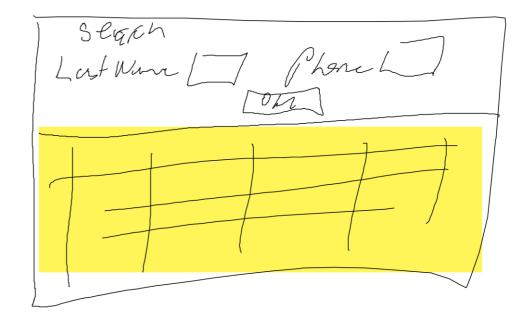
Custane List

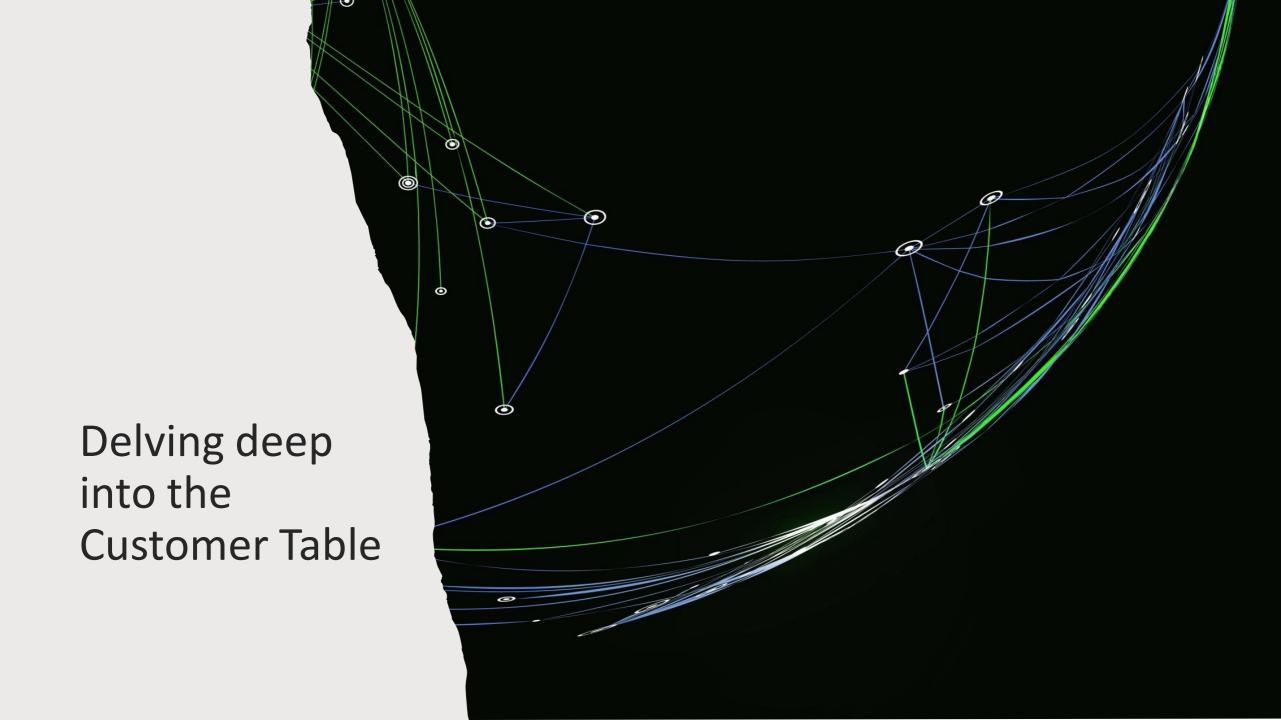


Customer Search Results

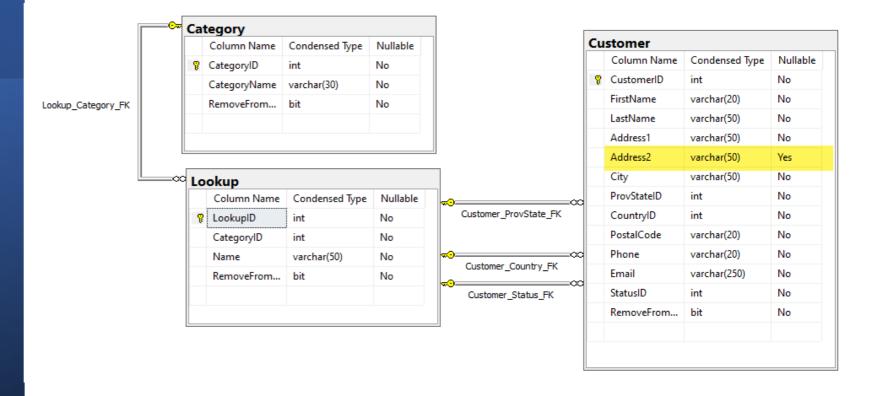
- First Name: The first name of the customer.
- Last Name: The last name of the customer.
- **City**: The city where the customer resides.
- **Phone**: The customer's contact phone number.
- **Email**: The customer's email address.
- **Status**: Customer's Status. (Bronze, Silver, Gold, and Platinum).
- **Total Sales**: The total sales associated with the customer.
- Rules
 - Only customers with the property of RemoveFromViewFlag = false can be retrieved.
- Edit Button for edit a customer in the customer search list results.

Costane List





Customer Schema

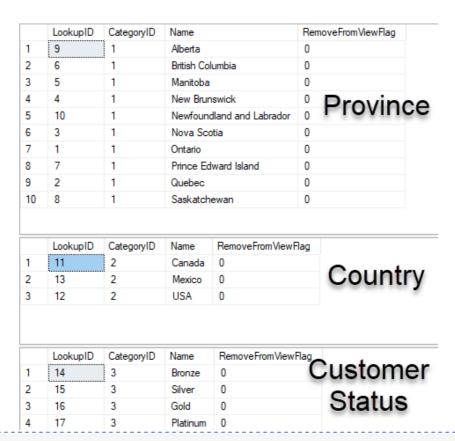


Categories and Lookup Data

Hog-Wild Categories

	CategoryID	CategoryName	RemoveFromViewFlag
1	1	Province	0
2	2	Country	0
3	3	Customer Status	0
4	4	Employee Role	0
5	5	Part Categories	0

Customer Lookups



Introduction to Models

Strongly Type Data

Models Overview

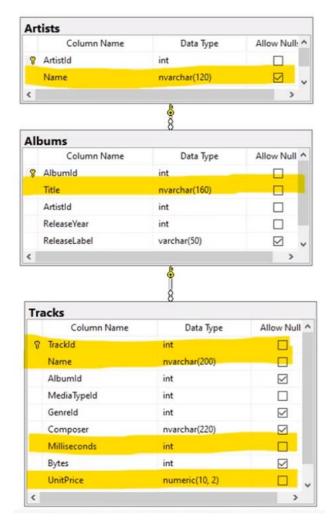
- Customer Search Model (CustomerSearchView)
 - Used to provide a list of customers that means specific search criteria.
 - Includes "Total Sales."
- Customer Add/Edit Model (CustomerEditView)
 - Allows for adding or editing customer data.
 - Provides RemoveFromViewFlag, which allows for a "soft delete."
- **NOTE**: The naming schema for all models is:
 - **1. Subject name** i.e. Customer, Invoice, Employee
 - **2. Action** i.e. Search, Add/Edit -> Edit, Lookup
 - 3. Follow by the word "View"
- Searching for customers is CustomerSearchView
- Adding or editing a customer is CustomerEditView

Model vs. Database – Clarifying Misconceptions

- The Model isn't just the database table or the ORM (e.g., Entity Framework).
- It represents the structure and logic revolving around the data.
- The database is a part of the Model but not its entirety.



Model vs. Database – Clarifying Misconceptions (Multiple Tables -> Single Model)



```
public class TrackSelectionView
{
    public int TrackId { get; set; }
    public string SongName { get; set; }
    public string AlbumTitle { get; set; }
    public string ArtistName { get; set; }
    public int Milliseconds { get; set; }
    public decimal Price { get; set; }
    public string Length
    {
        get { return $"{(int)Milliseconds / 60 / 1000}:{Milliseconds / 1000 % 60}"; }
    }
}
```

Customer Search View



Please think about what properties are needed for the Customer Search Model



Cu	stomer		
	Column Name	Data Type	Allow Nulls
8	CustomerID	int	
	FirstName	varchar(20)	
	LastName	varchar(50)	
	Address1	varchar(50)	
	Address2	varchar(50)	<u>~</u>
	City	varchar(50)	
	ProvStateID	int	
	CountryID	int	
	PostalCode	varchar(20)	
	Phone	varchar(20)	
	Email	varchar(250)	
	StatusID	int	
	${\sf RemoveFromViewFlag}$	bit	

Invoice

InvoiceD

InvoiceDate

CustomerID

EmployeeID

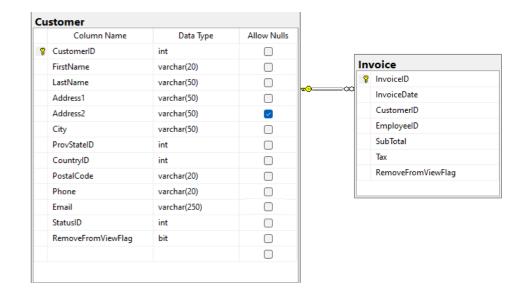
SubTotal

Tax

RemoveFromViewFlag

Database Customer Search Model Vs Customer & Invoice Table

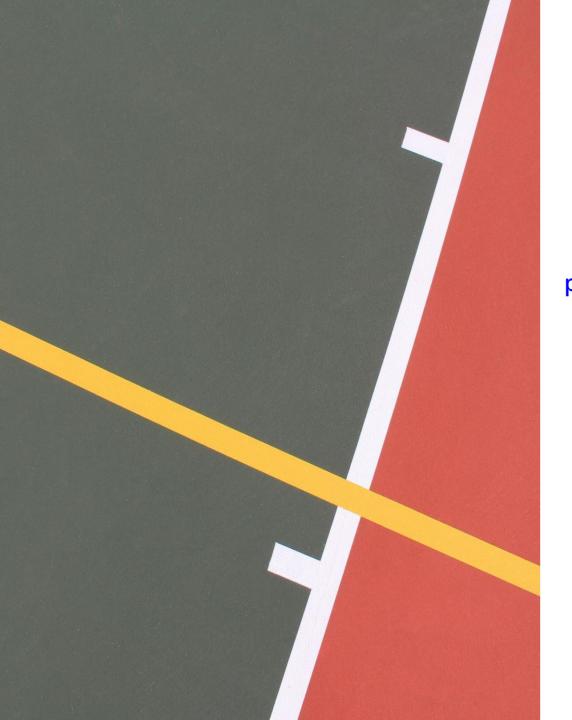
- Model
- Customer ID
- •First Name
- Last Name
- City
- Phone
- •Email
- Status ID
- Total Sales



Customer Search Model (Customer Search View)

- Please create a "Customer Search" model based on the following properties.
 - Customer ID
 - First Name
 - Last Name
 - City
 - Phone
 - Email
 - Status ID
 - Total Sales





Customer Search Model (CustomerSearchView)

```
public class CustomerSearchView
      public int CustomerID { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }
      public string City { get; set; }
      public string Phone { get; set; }
      public string Email { get; set; }
      public int StatusID { get; set; }
      public decimal TotalSales { get; set; }
```

Creating Folders Structure and Save Model

- 1. Goto your DMIT 2018 workbook.
- 2. Create a folder called BlazorDemo
- 3. In the BlazorDemo folder, create LINQ
- 4. In the LINQ folder, create three folders called Queries, UnitTest & ViewModels
- 5. Save your model as BlazorDemo\LINQ\ViewModels \CustomerSearchView.cs



Customer Edit Ford First Name: Filly Last Name: Ford Address1: 77777 - 77 St Address2: City: Edmonton Province: Alberta Country: Canada Postal Code: T5J5X4 SAVE CLOSE

Customer Edit Model Phone: 7804324756

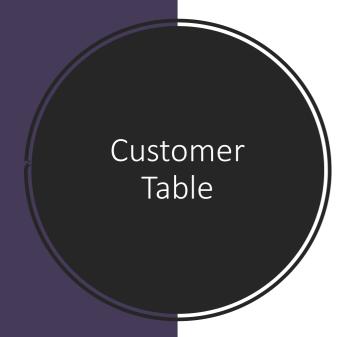
Email: fford@hotmail.com

Customer Status: Bronze

Previous Purchases

	Invoice ID		Invoice Date	Total
VIEW	30	Invoice Medal	2023-08-07	\$4,252.29
VIEW	34	Invoice Model	2023-08-07	\$2,123.84
VIEW	37		2023-08-08	\$7,113.91

Please think about what properties are needed for the Customer Edit Model



	Column Name	Data Type	Allow Nulls
P	CustomerID	int	
	FirstName	varchar(20)	
	LastName	varchar(50)	
	Address1	varchar(50)	
	Address2	varchar(50)	\checkmark
	City	varchar(50)	
	ProvStateID	int	
	CountryID	int	
	PostalCode	varchar(20)	
	Phone	varchar(20)	
	Email	varchar(250)	
	StatusID	int	
	RemoveFromViewFlag	bit	

Database Customer Edit Model Vs Customer

Model

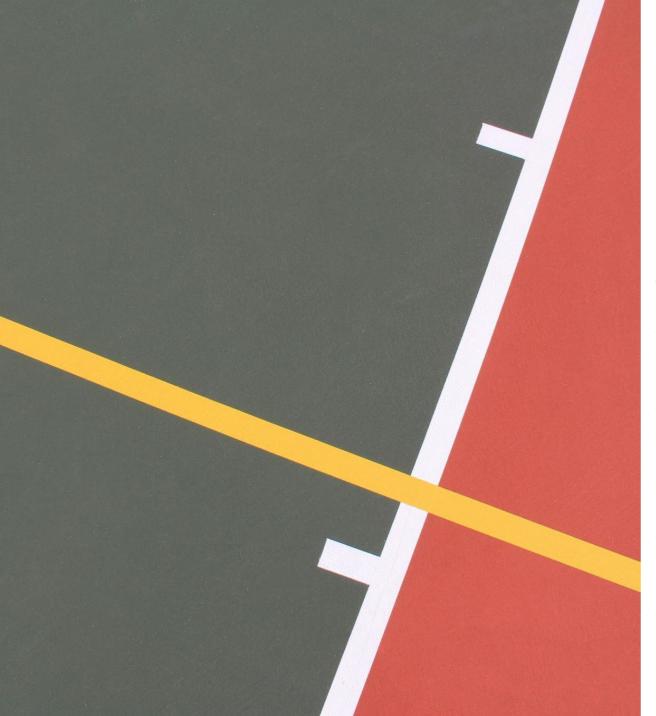
- •First Name
- •Last Name
- Address 1
- Address 2
- City
- Province/State ID
- Country ID
- Postal Code
- Phone
- •Email
- •Status ID
- •Remove From View Flag

	Column Name	Data Type	Allow Nulls
P	CustomerID	int	
	FirstName	varchar(20)	
	LastName	varchar(50)	
	Address1	varchar(50)	
	Address2	varchar(50)	\checkmark
	City	varchar(50)	
	ProvStateID	int	
	CountryID	int	
	PostalCode	varchar(20)	
	Phone	varchar(20)	
	Email	varchar(250)	
	StatusID	int	
	${\sf RemoveFromViewFlag}$	bit	

Customer Search Model (Customer Search View)

- Please create a "Customer Search" model based on the following properties.
- First Name
- Last Name
- Address 1
- Address 2
- City
- Province/State ID
- Country ID
- Postal Code
- Phone
- Email
- Status ID
- Remove From View Flag





Customer Model (CustomerEditView)

```
public class CustomerEditView
     public int CustomerID { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }
      public string Address1 { get; set; }
      public string Address2 { get; set; }
      public string City { get; set; }
     public int ProvStateID { get; set; }
      public int CountryID { get; set; }
      public string PostalCode { get; set; }
      public string Phone { get; set; }
      public string Email { get; set; }
      public int StatusID { get; set; }
      public bool RemoveFromViewFlag { get; set; }
```