# Practical Assignment Three

## Database Design

In practical assignment three, you evaluated the analysis phase deliverables for an information system that would support a small package delivery company. Continuing with this scenario, below you will find the database design that was derived from the previous deliverables (from Practical Assignment Two). **In this assignment, you will evaluate the database design based on the analysis from the previous assignments, and provide recommendations based on your analysis.**

Below, you will find the physical database design, with relationships. This may look slightly different than some of the syntax in our lecture/text, as this was generated by the database code (Data Definition Language “DDL” in the Structured Query Language “SQL” syntax) [*also included*] using one Microsoft’s reverse engineering tools. This is very similar to what you might expect to see in the field when reviewing design documentation. However, you should be able to identify each table, the primary keys, foreign keys, relationships, constraints, and datatypes.

When evaluating the design, try to understand the database schema, and square this against the analysis deliverables you previously evaluated. For example, I’d suggest using these criteria:

1. Review the design and determine if it is adequately normalized to support the business rules.
	1. Are there any enumerated fields (for example, how would multiple phone numbers for a single customer be handled? And what is the business rule?)
	2. Other normalization issues to look for are missing tables that would associate entities, for example, can a customer ship multiple packages in one order?
	3. Are there any redundancies that would affect either performance or data integrity?
2. Review the relationships for both semantic and technical accuracy.
	1. Is the relationship aligned with the analysis E-R diagram?
	2. Do the primary/foreign key relationships in the physical design reflect that relationship?
3. Look at the constraints and data types and determine if they support the business rules and data integrity of the system.
	1. Do they ensure the integrity of data that is critical to the process/business rules?
	2. Do datatypes accurately reflect how the data will be used, and any processing requirements, for example, dates, numbers, etc.?
	3. Are there constraints where needed to enforce business rules?
4. How well do you think this design will support the business?
	1. Will it perform adequately? (This usually requires adequate indexes on commonly used fields)
	2. Are the fields designed in such a way to maximize efficient use of data storage? (Look at data types and lengths.)
5. Will this design support accountability requirements?
	1. Your expertise as an IT auditor might lead you to recommendations of designs/tools that would help with accountability. For example,
		1. How do we know when payments are received, processed, etc.?
		2. How do we know which employees handled which packages or company assets?
		3. Are there types of fraud that would be difficult to trap/investigate based on this design?

## Using the deliverables below, you will provide two deliverables for this assignment:

1. Annotate on the diagrams (you can use a printout and scan physical copies for your assignment submission) and issues you identified, and proposed modifications.
2. Provide a report that explains proposed changes and any other details you believe would be relevant. (When you mark changes, you can include a reference, for example “A”, and use that marking to refer to the error/change in your report.)

Grading rubric:

Students should expect to find at least 2 major issues (for example, missing tables, relationships or primary/foreign keys), and 10 (*or a lot more!*) minor changes that would support business rules or IT audit concerns (for example, wrong datatypes, field lengths, missing constraints, indexes, etc.). For most students, the report will be approximately 1 page.

Some students with background in SQL DDL may choose to also include a revised design, but this is not required, and should not be used in place of the annotations and report.

For each proposed change or error, you should note either a best-practice or findings from the analysis to justify your change.

## Database Diagram (generated from SQL DML):



## Database DML Code:

CREATE TABLE [dbo].[BuildingAddress](

 [buildingAddressID] [int] IDENTITY(1,1) NOT NULL,

 [address] [nvarchar](50) NULL,

 [city] [nvarchar](50) NULL,

 [state] [nchar](10) NULL,

 [ZIP] [nchar](10) NULL,

 CONSTRAINT [PK\_BuildingAddress] PRIMARY KEY CLUSTERED

(

 [buildingAddressID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Customer](

 [customerNumber] [varchar](50) NOT NULL,

 [customerName] [varchar](50) NULL,

 [customerAddress] [int] NULL,

 [customerPhone] [int] NULL,

 [customerEmail] [nvarchar](50) NULL,

 CONSTRAINT [PK\_Customer] PRIMARY KEY CLUSTERED

(

 [customerNumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[CustomerAddress](

 [customerAddressID] [int] IDENTITY(1,1) NOT NULL,

 [address] [nvarchar](50) NULL,

 [city] [nvarchar](50) NULL,

 [state] [nchar](10) NULL,

 [ZIP] [nchar](10) NULL,

 CONSTRAINT [PK\_Address] PRIMARY KEY CLUSTERED

(

 [customerAddressID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[DeliveryAddress](

 [deliveryAddressID] [int] IDENTITY(1,1) NOT NULL,

 [address] [nvarchar](50) NULL,

 [city] [nvarchar](50) NULL,

 [state] [nchar](10) NULL,

 [ZIP] [nchar](10) NULL,

 CONSTRAINT [PK\_DeliveryAddress] PRIMARY KEY CLUSTERED

(

 [deliveryAddressID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Employee](

 [employeeNumber] [nchar](10) NOT NULL,

 [employeeName] [varchar](50) NULL,

 [workAddress] [int] NULL,

 [workEmail] [nvarchar](50) NULL,

 [workPhone] [int] NULL,

 [mobilePhone] [nchar](16) NULL,

 CONSTRAINT [PK\_Employee] PRIMARY KEY CLUSTERED

(

 [employeeNumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Phone](

 [phoneID] [int] IDENTITY(1,1) NOT NULL,

 [countryCode] [int] NULL,

 [phoneNumber] [int] NULL,

 CONSTRAINT [PK\_Phone] PRIMARY KEY CLUSTERED

(

 [phoneID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PickupAddress](

 [pickupAddressID] [int] IDENTITY(1,1) NOT NULL,

 [address] [nvarchar](50) NULL,

 [city] [nvarchar](50) NULL,

 [state] [nchar](10) NULL,

 [ZIP] [nchar](10) NULL,

 CONSTRAINT [PK\_PickupAddress] PRIMARY KEY CLUSTERED

(

 [pickupAddressID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

CREATE TABLE [dbo].[ServiceOrder](

 [serviceOrderNumber] [int] NOT NULL,

 [customerNumber] [varchar](50) NOT NULL,

 [orderDate] [date] NULL,

 [pickupDate] [date] NULL,

 [solineID] [int] NOT NULL,

 [pickupAddress] [int] NULL,

 [deliveryAddress] [int] NULL,

 [employeeID] [nchar](10) NULL,

 CONSTRAINT [PK\_ServiceOrder] PRIMARY KEY CLUSTERED

(

 [serviceOrderNumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[ServiceOrderLine](

 [solineID] [int] NOT NULL,

 [packageID] [nchar](26) NULL,

 [packageDescription] [varchar](50) NULL,

 [serviceOrderNumber] [int] NOT NULL,

 [packageQuantity] [nchar](10) NULL,

 [packageWeight] [nchar](10) NULL,

 [packageSize] [nchar](10) NULL,

 [packageValue] [nchar](10) NULL,

 [isPerishable] [nchar](10) NULL,

 [buildingCode] [int] NULL,

 [binNumber] [nchar](10) NULL,

 CONSTRAINT [PK\_ServiceOrderLine] PRIMARY KEY CLUSTERED

(

 [solineID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Vehicle](

 [vinNumber] [nchar](17) NOT NULL,

 [registrationNumber] [varbinary](50) NULL,

 [licensePlateNumber] [nchar](10) NULL,

 [grossVehicleWeight] [nchar](10) NULL,

 [lastInspectionDate] [nchar](10) NULL,

 [vehicleType] [nchar](10) NULL,

 [vehicleStatus] [nchar](10) NULL,

 [employeeID] [nchar](10) NULL,

 CONSTRAINT [PK\_Vehicle] PRIMARY KEY CLUSTERED

(

 [vinNumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Warehouse](

 [buildingCode] [int] NOT NULL,

 [buildingAddress] [int] NULL,

 [hoursofOperation] [nchar](10) NULL,

 [buildingContract] [varchar](max) NULL,

 [binNumber] [nchar](10) NULL,

 CONSTRAINT [PK\_Warehouse] PRIMARY KEY CLUSTERED

(

 [buildingCode] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

CREATE TABLE [dbo].[WorkAddress](

 [workAddressID] [int] IDENTITY(1,1) NOT NULL,

 [address] [nvarchar](50) NULL,

 [city] [nvarchar](50) NULL,

 [state] [nchar](10) NULL,

 [ZIP] [nchar](10) NULL,

 CONSTRAINT [PK\_WorkAddress] PRIMARY KEY CLUSTERED

(

 [workAddressID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Customer] WITH CHECK ADD CONSTRAINT [FK\_Customer\_CustomerAddress] FOREIGN KEY([customerAddress])

REFERENCES [dbo].[CustomerAddress] ([customerAddressID])

GO

ALTER TABLE [dbo].[Customer] CHECK CONSTRAINT [FK\_Customer\_CustomerAddress]

GO

ALTER TABLE [dbo].[Customer] WITH CHECK ADD CONSTRAINT [FK\_Customer\_Phone] FOREIGN KEY([customerPhone])

REFERENCES [dbo].[Phone] ([phoneID])

GO

ALTER TABLE [dbo].[Customer] CHECK CONSTRAINT [FK\_Customer\_Phone]

GO

ALTER TABLE [dbo].[Employee] WITH CHECK ADD CONSTRAINT [FK\_Employee\_WorkAddress] FOREIGN KEY([workAddress])

REFERENCES [dbo].[WorkAddress] ([workAddressID])

GO

ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK\_Employee\_WorkAddress]

GO

ALTER TABLE [dbo].[ServiceOrder] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrder\_Customer] FOREIGN KEY([customerNumber])

REFERENCES [dbo].[Customer] ([customerNumber])

GO

ALTER TABLE [dbo].[ServiceOrder] CHECK CONSTRAINT [FK\_ServiceOrder\_Customer]

GO

ALTER TABLE [dbo].[ServiceOrder] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrder\_DeliveryAddress] FOREIGN KEY([deliveryAddress])

REFERENCES [dbo].[DeliveryAddress] ([deliveryAddressID])

GO

ALTER TABLE [dbo].[ServiceOrder] CHECK CONSTRAINT [FK\_ServiceOrder\_DeliveryAddress]

GO

ALTER TABLE [dbo].[ServiceOrder] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrder\_Employee] FOREIGN KEY([employeeID])

REFERENCES [dbo].[Employee] ([employeeNumber])

GO

ALTER TABLE [dbo].[ServiceOrder] CHECK CONSTRAINT [FK\_ServiceOrder\_Employee]

GO

ALTER TABLE [dbo].[ServiceOrder] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrder\_PickupAddress] FOREIGN KEY([pickupAddress])

REFERENCES [dbo].[PickupAddress] ([pickupAddressID])

GO

ALTER TABLE [dbo].[ServiceOrder] CHECK CONSTRAINT [FK\_ServiceOrder\_PickupAddress]

GO

ALTER TABLE [dbo].[ServiceOrder] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrder\_ServiceOrderLine] FOREIGN KEY([solineID])

REFERENCES [dbo].[ServiceOrderLine] ([solineID])

GO

ALTER TABLE [dbo].[ServiceOrder] CHECK CONSTRAINT [FK\_ServiceOrder\_ServiceOrderLine]

GO

ALTER TABLE [dbo].[ServiceOrderLine] WITH CHECK ADD CONSTRAINT [FK\_ServiceOrderLine\_Warehouse] FOREIGN KEY([buildingCode])

REFERENCES [dbo].[Warehouse] ([buildingCode])

GO

ALTER TABLE [dbo].[ServiceOrderLine] CHECK CONSTRAINT [FK\_ServiceOrderLine\_Warehouse]

GO

ALTER TABLE [dbo].[Vehicle] WITH CHECK ADD CONSTRAINT [FK\_Vehicle\_Employee] FOREIGN KEY([employeeID])

REFERENCES [dbo].[Employee] ([employeeNumber])

GO

ALTER TABLE [dbo].[Vehicle] CHECK CONSTRAINT [FK\_Vehicle\_Employee]

GO

ALTER TABLE [dbo].[Warehouse] WITH CHECK ADD CONSTRAINT [FK\_Warehouse\_BuildingAddress] FOREIGN KEY([buildingAddress])

REFERENCES [dbo].[BuildingAddress] ([buildingAddressID])

GO

ALTER TABLE [dbo].[Warehouse] CHECK CONSTRAINT [FK\_Warehouse\_BuildingAddress]

GO