CA1: Database Design & Development

Module Title: Database Design & Development

Module Code: B8IT113

Module Leader: Jennifer Byrne

Student Name: Daniel Contero

Student Code: 10583900

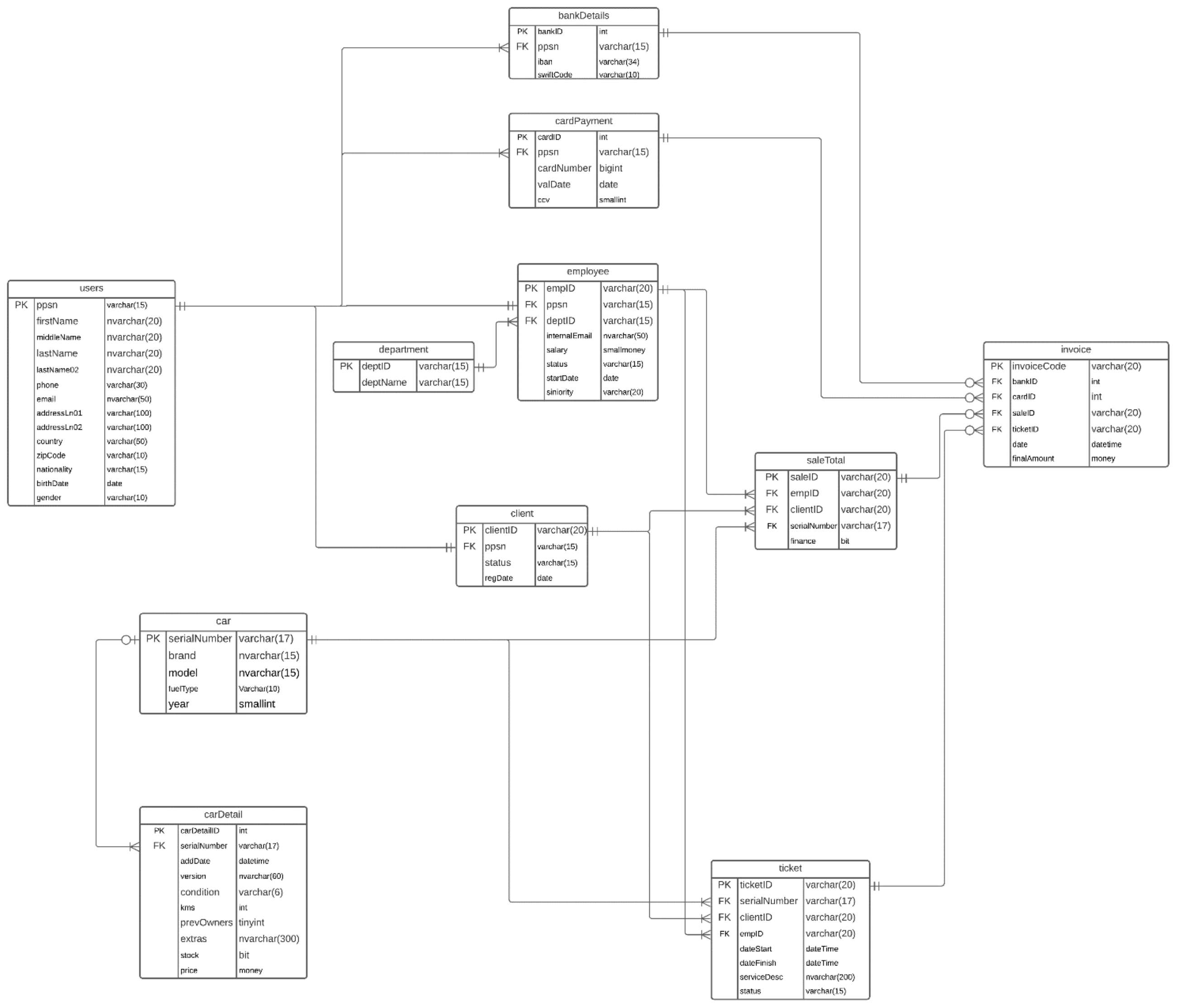
Contents

1. Project Overview/Scope

The objective of this project is providing a technical design of a database as well as the implementation of it for a dealership company. In this phase the main elements of the design are the cars, the customers, the employees, the sales and the tickets. The design also attempts to protect the dealership from losing business information in the event of GDPR hard removal of information.

In this design and implementation is not included the marketing department, just leaving the output they could use as is specify in the documentation. Also, HR department is not included on itself but many functionalities as the payslip are included, allowing in the future an extension of the design to include a more detailed version.

2. Entity Relationship Diagram



3. Assumptions Made

It would be needed an extended module with encryption and a connection with a payment platform to allow to keep payment records safe and check if they are correct. Also, to allow direct payments

An HR module would need to be develop and added to handle more precisely employee information as well as the payslip.

The model uses the concept of user to reduce duplicity of data as it would be common that employees use facilities as clients.

An extended module to track detailed information about the cars in stock would need to be develop further down the line to ensure centralization of the information as it is assumed now that the information there is not exhaustive and is compared with a brand catalogue for accuracy with the customer.

The serial number is changed to VARCHAR(17) as 17 is the maximum length for cars in Europe.

The column finance on the table saleTotal is just to keep track of any verbal agreement that the salesperson could have made during negotiations with the client. A module with extended information on this area would be highly recommended.

The set up of permissions of access, visibility and modification of the data depending on the user that access the information.

4. Data Dictionary

Attribute	Datatype	Optional	Description
[department]			
[deptID]	VARCHAR(15)	NO	Unique, the primary key in format MCH001
[deptName]	VARCHAR(15)	NO	Descriptive Name
[users]			
[ppsn]	VARCHAR(15)	NO	Primary key in format 000000AA
[firstName]	NVARCHAR(20)	NO	Persons first name
[middleName]	NVARCHAR(20)	NO	Persons middle name
[lastName]	NVARCHAR(20)	NO	Persons surname
[lastName02]	NVARCHAR(20)	NO	Secondary surname
[phone]	VARCHAR(30)	NO	Contact number
[email]	NVARCHAR(50)	NO	Contact email
[addressLn01]	VARCHAR(100)	NO	Persons address
[addressLn02]	VARCHAR(100)	NO	Persons address line two
[country]	VARCHAR(50)	NO	Persons country of residence

[zipCode]	VARCHAR(10)	NO	Zipcode of above address
[nationality]	VARCHAR(15)	NO	Persons Nationality
[birthDate]	DATE	NO	Persons date of birth
[gender]	VARCHAR(10)	YES	Persons gender

[bankDetails]			
[bankID]	INT	NO	Unique, the primary key, starts at one with one incremental
[ppsn]	VARCHAR(15)	NO	Foreign Key in the format 000000AA links to users entity, see above, as default has "GDPR"
[iban]	VARCHAR(34)	NO	SEPA standard iban code
[swiftCode]	VARCHAR(10)	NO	Standard bank swiftcode

[cardPayment]			
[cardID]	INT	NO	Unique, the primary key, starts at one with one incremental
[ppsn]	VARCHAR(15)	NO	Foreign Key in the format 000000AA links to users entity, see above, as default has "GDPR"
[cardNumber]	BIGINT	NO	Credit card number
[valDate]	DATE	NO	expiry date on the card, looks like YYYY-MM-01, day is always 01
[ccv]	SMALLINT	NO	Last three digits on back of card

[employee]			
[ID]	INT	NO	Identity, starts at one with one incremental
[empID]	VARCHAR(20)	NO	primary key, computed value between EMP and ID in the format EMP-000000
[ppsn]	VARCHAR(15)	NO	Foreign Key in the format 000000AA links to users entity, see above, as default has "GDPR"
[deptID]	VARCHAR(15)	NO	Unique, the foreign key in format MCH001, links to department entity, see above
[internalEmail]	NVARCHAR(60)	NO	computed field between the first name and the last name of the employee plus the email domain of the company, looks like firstnamelastname@company.com
[salary]	SMALLMONEY	NO	salary of the employee
[status]	VARCHAR(15)	NO	binary field between active and inactive, defaults to active
[startDate]	DATE	NO	manual input of the day the employee starts employment, looks like YYYY-MM-DD
[siniority]	VARCHAR(20)	NO	level of seniority such as junior, senior etc.

[client]			
[ID]	INT	NO	Identity, starts at one with one incremental
[clientID]	VARCHAR(20)	NO	primary key, computed value between CLI and ID in the format CLI-000000

[ppsn]	VARCHAR(15)	NO	Foreign Key in the format 000000AA links to users entity, see above, as default has "GDPR"
[status]	VARCHAR(15)	NO	binary field between active and inactive, defaults to active
[regDate]	DATE	NO	date of initial registration autopopulated by defaul with getdate()

[car]			
[serialNumber]	VARCHAR(17)	NO	primary key, the serial number of the car, in format following the industrial regulations
[brand]	VARCHAR(15)	NO	the brand name of the car
[model]	VARCHAR(15)	NO	the model of the car
[fuelType]	VARCHAR(10)	NO	binary field between gasoline and diesel
[year]	SMALLINT	NO	Production year of the car in the format YYYY

[carDetail]			
[carDetailID]	INT	NO	Identity, starts at one with one incremental
[serialNumber]	VARCHAR(17)	NO	foreign key, the serial number of the car, in format following the industrial regulations, links to car entity, see above
[addDate]	DATETIME	NO	date of initial registration autopopulated by defaul with getdate()
[version]	VARCHAR(60)	NO	detailed variant of the model of the car
[condition]	VARCHAR(6)	NO	binary field between used and new, default to new
[kms]	INT	NO	kms done by the car
[prevOwners]	TINYINT	NO	number of owners previously attached to registration
[extras]	VARCHAR(300)	NO	description of all the extras that the car has
[stock]	BIT	NO	binary field between in stock or out of stock, represented with 1 or 0
[price]	MONEY	NO	price of the car

[saleTotal]			
[ID]	INT	NO	Identity, starts at one with one incremental
[saleID]	VARCHAR(20)	NO	primary key, computed value between SLS and ID in the format SLS-000000
[empID]	VARCHAR(20)	NO	foreign key, computed value between EMP and ID in the format EMP-000000, links to entity employee, see above
[clientID]	VARCHAR(20)	NO	foreign key, computed value between CLI and ID in the format CLI-000000, links to entity client, see above
[serialNumber]	VARCHAR(17)	NO	foreign key, the serial number of the car, in format following the industrial regulations, links to car entity, see above

[finance]	BIT	NO	binary field between having finance (1) and not having finance (0)

[ticket]			
[ID]	INT	NO	Identity, starts at one with one incremental
[ticketID]	VARCHAR(20)	NO	primary key, computed value between SRV and ID in the format SRV-000000
[serialNumber]	VARCHAR(17)	NO	foreign key, the serial number of the car, in format following the industrial regulations, links to car entity, see above
[clientID]	VARCHAR(20)	NO	foreign key, computed value between CLI and ID in the format CLI-000000, links to entity client, see above
[empID]	VARCHAR(20)	NO	foreign key, computed value between EMP and ID in the format EMP-000000, links to entity employee, see above
[dateStart]	DATETIME	NO	autopopulated when the ticket is created with getdate()
[dateFinish]	DATETIME	NO	manually input date in the format YYYY-MM-DD
[serviceDesc]	VARCHAR(200)	NO	description of services provided to the car
[status]	VARCHAR(15)	NO	binary field between active and closed, defaults to active

[invoice]			
[ID]	INT	NO	Identity, starts at one with one incremental
[invoiceCode]	VARCHAR(20)	NO	primary key, computed value between INV and ID in the format INV-000000
[bankID]	INT	YES	Unique, the foreign key, starts at one with one incremental, links to entity bankDetails
[cardID]	INT	YES	Unique, the foreign key, starts at one with one incremental, links to entity cardPayment
[saleID]	VARCHAR(20)	YES	foreign key, computed value between SLS and ID in the format SLS-000000, links to entity client, see above
[ticketID]	VARCHAR(20)	YES	foreign key, computed value between SRV and ID in the format SRV-000000, links to entity client, see above
[date]	DATETIME	NO	autopopulated when the invoice is created with getdate()
[finalAmount]	MONEY	NO	final cost to be paid on invoice

5. Technology Used

LuciChart (https://www.lucidchart.com/)	Diagraming
SQL Server Management Studio	Transact SQL
Microsoft Word	Documentation
Moodle	Module Notes and Videos
SublimeText 3	Formatted Text Editing
https://www.mockaroo.com/	Data Generation

6. Test Plan

Item Tested	Test Run	Expected Result	Actual Result
newCustomer	EXEC newCustomer @ppsn = '0213123DB', @firstName = 'Gustavo', @middleName = NULL, @lastName = 'Cerezo', @lastName02 = 'Gila', @phone = '+353445345656', @email = 'gustavo69@hotmail.com', @addressLn01 = '45, somewhere somewhere, somewhere', @addressLn02 = NULL, @country = 'Ireland', @zipCode = 'D08F998', @nationality = 'Italian', @birthDate = '1984-05-12', @gender = 'Male', @status = 'Active' GO	New Customer Added, added register in users table and computed new entry in client table	New Customer Added, added register in users table and computed new entry in client table
newCustomer	EXEC newCustomer @ppsn = '0213123DB', @firstName = 'Gustavo', @middleName = NULL, @lastName = 'Cerezo', @lastName02 = 'Gila', @phone = '+353445345656', @email = 'gustavo69@hotmail.com', @addressLn01 = '45, somewhere somewhere, somewhere', @addressLn02 = NULL, @country = 'Ireland', @zipCode = 'D08F998', @nationality = 'Italian', @birthDate = 'TEST01', @gender = 'Male', @status = 'Active' GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 241 16 1 22 newCustomer Conversion failed when converting date and/or time from character string.

newEmployee	EXEC newEmployee @ppsn = '564345345FG', @firstName = 'Sara', @middleName = NULL, @lastName = 'Galindo', @lastName02 = NULL, @phone = '+353657058545', @email = 'theSara33@gmail.com', @addressLn01 = '67, somewhere around here, somewhere', @addressLn02 = NULL, @country = 'Ireland', @zipCode = 'D08F998', @nationality = 'Venezuelan', @birthDate = '1994-05-12', @gender = 'Female', @status = 'Active', @deptID = 'SLS002', @salary = '20000', @startDate = '2005-02-12', @siniority = 'Mid' GO	New employee added, added register in users table and computed new entry in employee table	New employee added, added register in users table and computed new entry in employee table
newEmployee	EXEC newEmployee @ppsn = '564345345FG', @firstName = NULL, @middleName = NULL, @lastName = 'Galindo', @lastName02 = NULL, @phone = '+353657058545', @email = 'theSara33@gmail.com', @addressLn01 = '67, somewhere around here, somewhere', @addressLn02 = NULL, @country = 'Ireland', @zipCode = 'D08F998', @nationality = 'Venezuelan', @birthDate = '1994-05-12', @gender = 'Female', @status = 'Active', @deptID = 'SLS002', @salary = '20000', @startDate = '2005-02-12',		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 515 16 2 41 newEmployee Cannot insert the value NULL into column 'firstName', table 'conteroDaniel_CA_carDealer.dbo.users'; column does not allow nulls. INSERT fails.

	@siniority = 'Mid' GO		
addCar	EXEC addCar @serialNumber = '1HGCR2F3XEA153973', @brand = 'Ford', @model = 'Edge', @fuelType = 'Gasoline', @year = '2008', @version = '250cv supercharged', @condition = 'Used', @kms = '200000', @preOwners = '2', @extras = 'Air con, Alloy wheels 21", Tinted windows', @stock = '1', @price = '30000' GO	New car added, added register in car table as well as in the carDetails table	New car added, added register in car table as well as in the carDetails table
addCar	EXEC addCar @serialNumber = '1HGCR2F3XEA153973', @brand = 'Ford', @model = 'Edge', @fuelType = 'Gasoline', @year = '2008', @version = '250cv supercharged', @condition = 'Used', @kms = 'TEST01', @preOwners = '2', @extras = 'Air con, Alloy wheels 21", Tinted windows',		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 245 16 1 20 addCar Conversion failed when converting the varchar value 'TEST01' to data type int.

	@stock = '1', @price = '30000' GO		
findClient	DECLARE @clientPrint VARCHAR(20) EXEC findClient @ppsn = '0213123DB', @clientID = @clientPrint OUTPUT PRINT @clientPrint GO	Returns clientID corresponding to the ppsn, prints the clientID	Returns clientID corresponding to the ppsn, prints the clientID
findClient	DECLARE @clientPrint VARCHAR(20) EXEC findClient @ppsn = 'TEST01', @clientID = @clientPrint OUTPUT PRINT @clientPrint GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 510011 16 1 17 findClient Client Can't be found
newTicket	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newTicket @serialNumber = '1HGCR2F3XEA153973', @ppsn = '0213123DB', @empID = @empID, @serviceDesc = 'TEST 01 Description' GO	Creates a new ticket with the parameters given	Creates a new ticket with the parameters given
newTicket	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newTicket @serialNumber = 'TEST01', @ppsn = '0213123DB',		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 510011 16 1 17 findClient Client Can't be found

	@empID = @empID, @serviceDesc = 'TEST 01 Description' GO		
closeTicket	DECLARE @ticketID VARCHAR(20) SELECT @ticketID = ticketID FROM ticket WHERE serviceDesc='TEST 01 Description' EXEC closeTicket @ticketID GO	closes an active ticket and adds the date of when it has been closed	closes an active ticket and adds the date of when it has been closed
closeTicket	DECLARE @ticketID VARCHAR(20) SELECT @ticketID = ticketID FROM ticket WHERE serviceDesc='' EXEC closeTicket @ticketID GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 51009 16 1 14 closeTicket The Ticket provided can't be found
newSale	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newSale @serialNumber = '1HGCR2F3XEA153973', @ppsn = '0213123DB', @empID = @empID, @finance = '0' GO	Creates a new sale with the parameters given	Creates a new sale with the parameters given
newSale	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newSale @serialNumber = 'TEST01', @ppsn = '0213123DB', @empID = @empID, @finance = '0' GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 510011 16 1 17 findClient Client Can't be found

dbo.stockChecking	DECLARE @result VARCHAR(20) SELECT @result = dbo.stockChecking ('1HGCR2F3XEA153973') PRINT @result GO	checks if the car exists in the car details table from the serial number given	checks if the car exists in the car details table from the serial number given
dbo.stockChecking	DECLARE @result VARCHAR(20) SELECT @result = dbo.stockChecking ('TEST01') PRINT @result GO		No
createDepartment	EXEC createDepartment 'DPM001', 'Department Test' GO	Creates a new department with the given parameters	Creates a new department with the given parameters
createDepartment	EXEC createDepartment 'DPM001', 'Department Test' GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 51007 16 1 12 createDepartment The department already exists

addPaymentMethod	DECLARE @cardID INT
	
	EXEC addPaymentMethod @ppsn = '0213123DB',
	  @paymentMethod = 'Card',
	  @cardNumber = '6546546546546465465',
	  @valDate = '2025-04-01',
	  @ccv = '986',
	  @result = @cardID OUTPUT
	PRINT @cardID
	GO
		Adds a new payment method with the given parameters, returns the unique ID of the created payment method and prints it out	Adds a new payment method with the given parameters, returns the unique ID of the created payment method and prints it out

addPaymentMethod	DECLARE @cardID INT
	
	EXEC addPaymentMethod @ppsn = '0213123DB',
	  @paymentMethod = 'Card',
	  @cardNumber = '6546546546546465465',
	  @valDate = '2025-04-01',
			ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage
			245 16 1 19 addPaymentMethod Conversion failed when converting the varchar value 'TEST' to data type smallint.

	@ccv = 'TEST01', @result = @cardID OUTPUT PRINT @cardID GO		
getClientPaymentMethod	EXEC getClientPaymentMethod '0213123DB', 'Bank', '1' GO	Obtains the payment method ID from the given ppsn, allowing to select between bank and card and from the payment methods that belong to the user, returns the ID	Obtains the payment method ID from the given ppsn, allowing to select between bank and card and from the payment methods that belong to the user, returns the ID
getClientPaymentMethod	EXEC getClientPaymentMethod '0213123DB', 'Bank', '10' GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 51001 16 1 14 getClientPaymentMethod The user doesn't exist
newInvoice	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newInvoice @clientPPSN = '0213123DB', @empID = @empID, @goodType = 'Sale', @paymentMethod = 'Card', @newMethod = 1, @cardNumber= '5867864434354312', @valDate = '2026-04-01', @ccv = '563', @finalAmount = '200000' GO	Create a new invoice with the parameters given, and also creates a new payment method with the proc addPaymentMethod	Create a new invoice with the parameters given, and also creates a new payment method with the proc addPaymentMethod

newInvoice	DECLARE @empID VARCHAR(20) SELECT @empID = empID FROM employee WHERE ppsn = '564345345FG' EXEC newInvoice @clientPPSN = '0213123DB', @empID = @empID, @goodType = 'Sale', @paymentMethod = 'Card', @newMethod = 0, @cardNumber= '5867864434354312', @valDate = '2026-04-01', @ccv = '563', @finalAmount = '200000' GO		ErrorNumber ErrorSeverity ErrorState ErrorLine ErrorProcedure ErrorMessage 51001 16 1 14 getClientPaymentMethod The user doesn't exist
customerDeleteGDPR	EXEC customerDeleteGDPR @ppsn = '0213123DB' GO	updates the client to inactive, sensors all payment methods for the given ppsn and then deletes the user from the users table	updates the client to inactive, sensors all payment methods for the given ppsn and then deletes the user from the users table
customerDeleteGDPR	EXEC customerDeleteGDPR @ppsn = '' GO		The user with PPSN N. doesn't exist

7. Reflections on Learning

At the beginning the ERM was really abstract, and it took me a while before I could engage fully with it. I had dived into coding the queries and I got deeper and deeper trying to make the maximum amount of functionality. I ended up making the work more complex than the brief, but it actually motivated me to keep going. This experience felt really enriching and I believe I learned a lot.

I still don’t understand why THROW is the only function that I could find that requires semi colons before and after. This will chase me in my nightmares.

8. References

https://stackoverflow.com/

https://docs.microsoft.com/

https://www.sqlshack.com/

https://www.mssqltips.com/

https://www.techonthenet.com/

https://www.sqlservertutorial.net/

https://www.w3schools.com/

https://www.zentut.com/

https://www.tutorialspoint.com/

https://www.red-gate.com/

https://www.kodyaz.com/

9. SQL

Please, find the SQL files in the following paths. The first 4 ones can be executed selecting all the code, in my test they create all without problem in a single execution.

/conteroDaniel_CA_projectFolder/conteroDaniel_CA_tableCreation.sql

/conteroDaniel_CA_projectFolder/conteroDaniel_CA_storedProcedures.sql

/conteroDaniel_CA_projectFolder/conteroDaniel_CA_views.sql

/conteroDaniel_CA_projectFolder/conteroDaniel_CA_dataDump.sql

For the last one I recommend execute proc by proc and view by view

/conteroDaniel_CA_projectFolder/conteroDaniel_CA_execAndTest.sql