

**CURSO** : 4686 - Base de Datos Avanzado I  
**PROFESOR** : RENZO MENDEZ MORENO  
**SEMESTRE** : 2024 – 1  
**CICLO** : Tercero  
**SECCIÓN** : T3FL - 00  
**FECHA** : 23 de Junio del 2024  
**DURACIÓN** : 90 minutos

NOTA

**ALUMNO (A): Raúl Ichiro Rosas Chinen**

1

## **SEGUNDA EVALUACIÓN DE LABORATORIO (T2)**

### **Consideraciones generales:**

- Considerar el orden, la correcta sintaxis de las sentencias, y la claridad al momento de definir sus respuestas.
- La evaluación es **estrictamente individual**.
- **No se puede compartir respuestas** por ningún medio durante la evaluación.
- Los teléfonos móviles deben permanecer en silencio, y sobre el escritorio.
- Estar pendiente a la hora final de la presente evaluación ya que, luego terminado el horario final, no les permitirá subir su evaluación por la plataforma blackboard.
- No está permitido el envío de la solución de su evaluación por otro medio que no sea blackboard.
- Leer detenidamente las preguntas, antes de iniciar el desarrollo de esta evaluación.
- **Recopilar en un archivo de texto, los comandos utilizados como parte de la solución.**
- **Así mismo, recopilar las capturas de pantalla de la ejecución y resultados obtenidos según cada pregunta, en este mismo documento.**

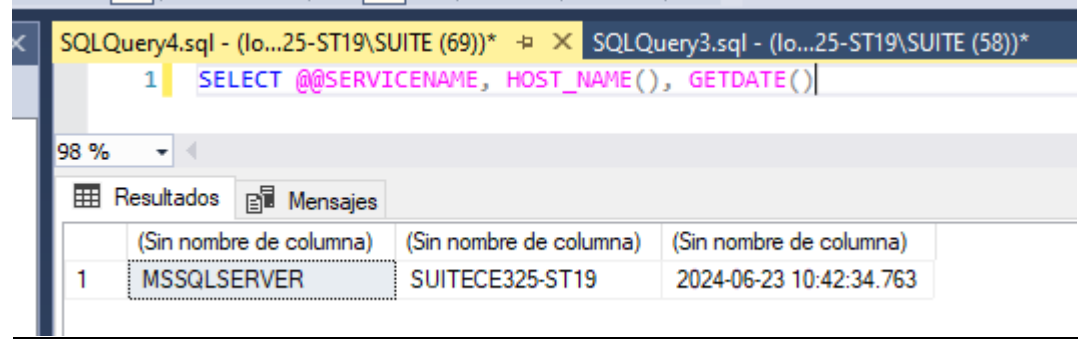
### **Consolidado**

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	06			
2	08			
3	03			
4	03			
Nota Recalificada				

## TAREAS PREVIA

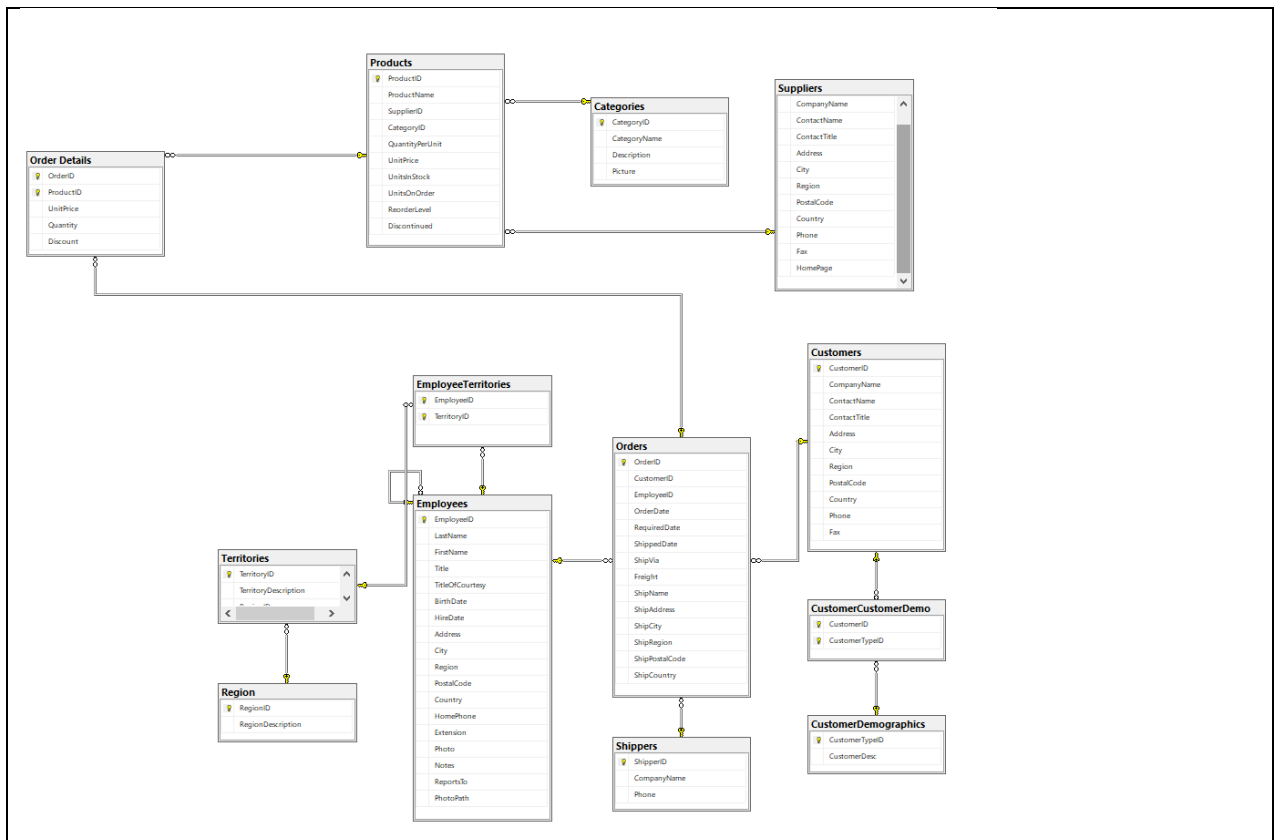
- A. Antes de iniciar a resolver las preguntas, abre una consulta en la instancia, ejecuta la siguiente sentencia, captura el resultado y cópialo aquí.

```
SELECT @@SERVICENAME, HOST_NAME(), GETDATE()
```



	(Sin nombre de columna)	(Sin nombre de columna)	(Sin nombre de columna)
1	MSSQLSERVER	SUITECE325-ST19	2024-06-23 10:42:34.763

- B. Descargar el archivo NORTHWIND\_T3HL.sql proporcionado por el docente, ejecutarlo en su instancia, elaborar el diagrama de la base de datos creada y copiarlo en el siguiente recuadro.



### **PREGUNTA 1: SENTENCIAS BULK INSERT / MERGE**

Desarrolle 3 archivos de texto (de tipo csv) considerando las siguientes características. Cada archivo debe contener:

- i. al menos 5 registros, y
- ii. al menos 2 de ellos se deben repetir en los 3 archivos, y
- iii. el resto de registros debe ser diferente.

<b>Archivo: Inmuebles_2024_FEB</b>		
<i>Columna</i>	<i>Tipo Dato</i>	<i>Descripción</i>
COD_AGN	CHAR(4)	Codificación del Agente Inmobiliario
NOM_AGN	VARCHAR(100)	Nombre completo del Agente Inmobiliario
VENTAS	INT	Número de Ventas Concretadas

<b>Archivo: Inmuebles_2024_MAR</b>		
<b>Columna</b>	<b>Tipo Dato</b>	<b>Descripción</b>
CODIGO	CHAR(4)	Codificación del Agente Inmobiliario
NOMBRECOMPLETO	VARCHAR(100)	Nombre completo del Agente Inmobiliario
NRO_VENTAS	INT	Número de Ventas Concretadas

<b>Archivo: Inmuebles_2024_ABR</b>		
<b>Columna</b>	<b>Tipo Dato</b>	<b>Descripción</b>
CODIGO_AGEN	CHAR(4)	Codificación del Agente Inmobiliario
NOMBRE_AGEN	VARCHAR(100)	Nombre completo del Agente Inmobiliario
VENTAS_AGEN	INT	Número de Ventas Concretadas

- a. Crear 1 tabla por cada archivo, tomando como referencia los nombres y tipos de datos en los cuadros anteriores.

```
23 )
24
25 SELECT * FROM sys.tables
26
27 BULK INSERT Inmuebles_2024_FEB
```

98 %

	name	object_id	principal
1	Employees	901578250	NULL
2	Categorias	965578478	NULL
3	Customers	997578592	NULL
4	Shippers	1029578706	NULL
5	Suppliers	1061578820	NULL
6	Orders	1093578934	NULL
7	Products	1189579276	NULL
8	Order Details	1397580017	NULL
9	CustomerCustomerDemo	1557580587	NULL
10	CustomerDemographics	1573580644	NULL
11	Region	1589580701	NULL
12	Territories	1605580758	NULL
13	EmployeeTerritories	1621580815	NULL
14	sysdiagrams	1813581499	NULL
15	Inmuebles_2024_FEB	1973582069	NULL
16	Inmuebles_2024_MAR	1989582126	NULL
17	Inmuebles_2024_ABR	2005582183	NULL

- b. Cargar los datos desde cada archivo csv en su respectiva tabla usando la sentencia adecuada.

```
27 BULK INSERT Inmuebles_2024_FEB
28 FROM 'D:\Descargas\bulk_1.csv'
29 WITH
30 (
31 FIELDTERMINATOR=',',
32 ROWTERMINATOR='\n'
33 )
```

98 %

Mensajes

(5 filas afectadas)

Hora de finalización: 2024-06-23T11:08:37.4953885-05:00

```
34
35 BULK INSERT Inmuebles_2024_MAR
36 FROM 'D:\Descargas\bulk_2.csv'
37 WITH
38 (
39 FIELDTERMINATOR=',',
40 ROWTERMINATOR='\n'
41 )
```

98 %

Mensajes

(5 filas afectadas)

Hora de finalización: 2024-06-23T11:09:46.9723134-05:00

```

42
43 BULK INSERT Inmuebles_2024_ABR
44 FROM 'D:\Descargas\bulk_3.csv'
45 WITH
46 (
47 FIELDTERMINATOR=';',
48 ROWTERMINATOR='\n'
49 )

```

98 %

Mensajes

(5 filas afectadas)

Hora de finalización: 2024-06-23T11:11:11.9395597-05:00

- c. Confirmar a través de una consulta simple que los datos hayan sido cargados por cada tabla

```

50
51 SELECT * FROM Inmuebles_2024_FEB
52 SELECT * FROM Inmuebles_2024_MAR
53 SELECT * FROM Inmuebles_2024_ABR

```

98 %

Resultados Mensajes

	COD_AGN	NOM_AGN	VENTAS
1	AG01	AGENTE JOSE	10
2	AG02	AGENTE MIGUEL	5
3	AG03	AGENTE ARTURO	8
4	AG04	AGENTE GANADOR	9
5	AG05	AGENTE VENDE MAX	15

	CODIGO	NOMBRECOMPLETO	NRO_VENTAS
1	AG01	AGENTE JOSE	10
2	AG02	AGENTE MIGUEL	5
3	AG06	AGENTE MEJOR CASA	20
4	AG07	AGENTE SUERTE	2
5	AG08	AGENTE SUPER CO...	6

	CODIGO_AGEN	NOMBRE_AGEN	VENTAS_AGEN
1	AG01	AGENTE JOSE	10
2	AG02	AGENTE MIGUEL	5
3	AG09	AGENTE NUEVE	9
4	AG10	AGENTE SUPR...	8
5	AG11	AGENTE SECR...	4

- d. **MERGE:** Tomando como base la tabla del archivo **Inmuebles\_2024\_ABR:**
- Y como referencia la tabla del archivo **Inmuebles\_2024\_FEB**, si encuentra coincidencias (REFERENCIA Codificación del Agente), actualizar el número de ventas, de lo contrario, insertar el registro

```

54
55 MERGE Inmuebles_2024_FEB AS Target
56 USING Inmuebles_2024_ABR AS Source
57 ON (Target.COD_AGN = Source.CODIGO_AGEN)
58 WHEN MATCHED THEN
59 UPDATE SET VENTAS=Source.VENTAS_AGEN
60 WHEN NOT MATCHED THEN
61 INSERT VALUES (Source.CODIGO_AGEN, Source.NOMBRE_AGEN, Source.VENTAS_AGEN);

```

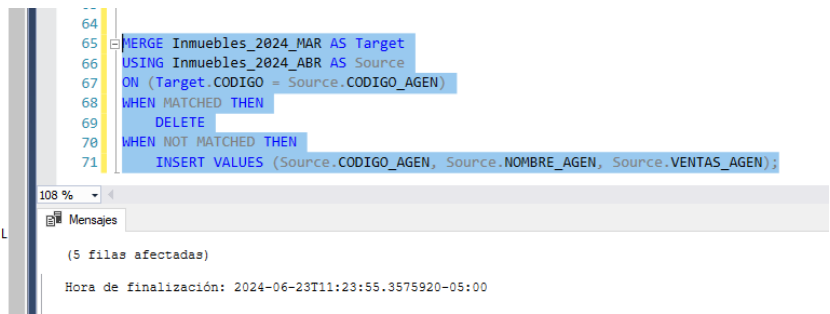
108 %

Mensajes

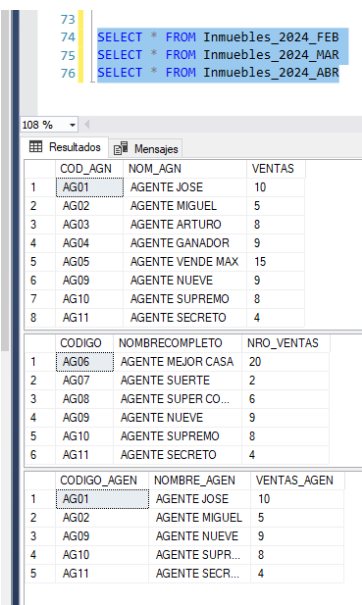
(5 filas afectadas)

Hora de finalización: 2024-06-23T11:21:23.1247532-05:00

II. Y como referencia la tabla del archivo **Inmuebles\_2024\_MAR**, si encuentra coincidencias (REFERENCIA Codificación del Agente), eliminar el registro, de lo contrario, insertar el registro



e. Consultar los datos de las 3 tablas, luego de aplicar los cambios en el punto d.



Rúbrica				
Puntos	Excelente: 6 puntos	Bueno: 4 puntos	Regular: 2 puntos	Deficiente: 1 punto
6 Puntos	Completa los puntos a, b, c, d y e. (*)	Completa los puntos a, b, c. (*)	Completa los puntos a y b. (*)	Completa el punto a. (*)

(\*) Por cada ejecución, capture la pantalla de la ejecución y el resultado obtenido, y cópielas aquí.

## USANDO LA BASE DE DATOS **NORTHWIND\_T3HL** DESARROLLE LO SIGUIENTE:

### **PREGUNTA 2: CURSORES**

Implemente un cursor que permita listar los países registrados en la tabla de Proveedores (Suppliers).

Por cada país, listar sus Proveedores asociados, mostrando el Nombre de la Compañía y Región.

A su vez, por cada Proveedor, debe mostrar los productos asociados, considerando el Nombre del Producto y el Nombre de su Categoría, así como el Total de Pedidos donde se despachó este producto.

Cada producto debe tener un índice de posición (Nro 1, Nro 2, Nro 3...)

Por cada Proveedor, incluir el Total de Productos Vendidos.

#### Referencia:

País: PERU

Proveedor: MI PROVEEDOR #100 | Región: AAA

#1	producto 1 – categoría A – Nro. Pedidos: 10
#2	producto 10 – categoría A – Nro. Pedidos: 4
#3	producto 5 – categoría B – Nro. Pedidos: 1
#4	producto 3 – categoría B – Nro. Pedidos: 1
+ Máximo Nro de Pedidos: 10	

Proveedor: MI FAVORITO #200 | Región: BBB

#1	producto 55 – categoría D – Nro. Pedidos: 8
#2	producto 33 – categoría F – Nro. Pedidos: 22
+ Máximo Nro de Pedidos: 22	

```

77
78
79
80 -- PREGUNTA 2
81
82 --CURSOR COMPLETO
83
84 DECLARE c_paises_suppliers CURSOR FOR
85     SELECT DISTINCT Country FROM Suppliers
86 DECLARE @v_country VARCHAR(15)
87 OPEN c_paises_suppliers
88
89 FETCH NEXT FROM c_paises_suppliers INTO @v_country
90 WHILE (@@FETCH_STATUS=0)
91 BEGIN
92     PRINT 'País: ' + @v_country
93     -----
94
95     DECLARE c_proveedores_suppliers CURSOR FOR
96         SELECT SupplierID, CompanyName, Region FROM Suppliers WHERE Country = @v_country
97     DECLARE @v_supplierID INT, @v_companyName VARCHAR(40), @v_region VARCHAR(15)
98     OPEN c_proveedores_suppliers
99
100    FETCH NEXT FROM c_proveedores_suppliers INTO @v_supplierID, @v_companyName, @v_region
101    WHILE (@@FETCH_STATUS=0)
102    BEGIN
103        PRINT ' Proveedor: ' + @v_companyName + ' - Region: ' + ISNULL(@v_region, '/NULL/')
104        -----
105
106        DECLARE c_productos CURSOR FOR
107            SELECT p.ProductName, c.CategoryName, SUM(od.Quantity)
108            FROM Products p, Categories c, [Order Details] od, Suppliers s
109            WHERE p.ProductID = od.ProductID AND c.CategoryID = p.CategoryID AND s.SupplierID=p.SupplierID AND s.SupplierID=@v_supplierID
110            GROUP BY p.ProductName, c.CategoryName
111            ORDER BY 2, 1
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Messages

```

País: Australia
Proveedor: Pavlova, Ltd. - Region: Victoria
Producto #1: Outback Lager - Categoria: Beverages - Pedidos: 817
Producto #2: Veggie-spread - Categoria: Condiments - Pedidos: 445
Producto #3: Pavlova - Categoria: Confections - Pedidos: 1158
Producto #4: Alice Mutton - Categoria: Meat/Poultry - Pedidos: 978
Producto #5: Carnarvon Tigers - Categoria: Seafood - Pedidos: 539
>>>>>> TOTAL DE PRODUCTOS VENDIDOS: 3937 <<<<<<
Proveedor: G'day, Mate - Region: NSW
Producto #1: Filo Mix - Categoria: Grains/Cereals - Pedidos: 500

```

Rúbrica				
Puntos	Excelente: 8 puntos	Bueno: 4 puntos	Regular: 2 puntos	Deficiente: 1 punto
8 Puntos	Relacionar los 3 cursores para mostrar la información como la Referencia. Así como calcular el Total por cada Cliente. (*)	Relacionar los cursores países y clientes para mostrar la información como la Referencia. (*)	Definir un cursor para países, otro para clientes y otro para pedidos. (*)	Definir un cursor para países. (*)

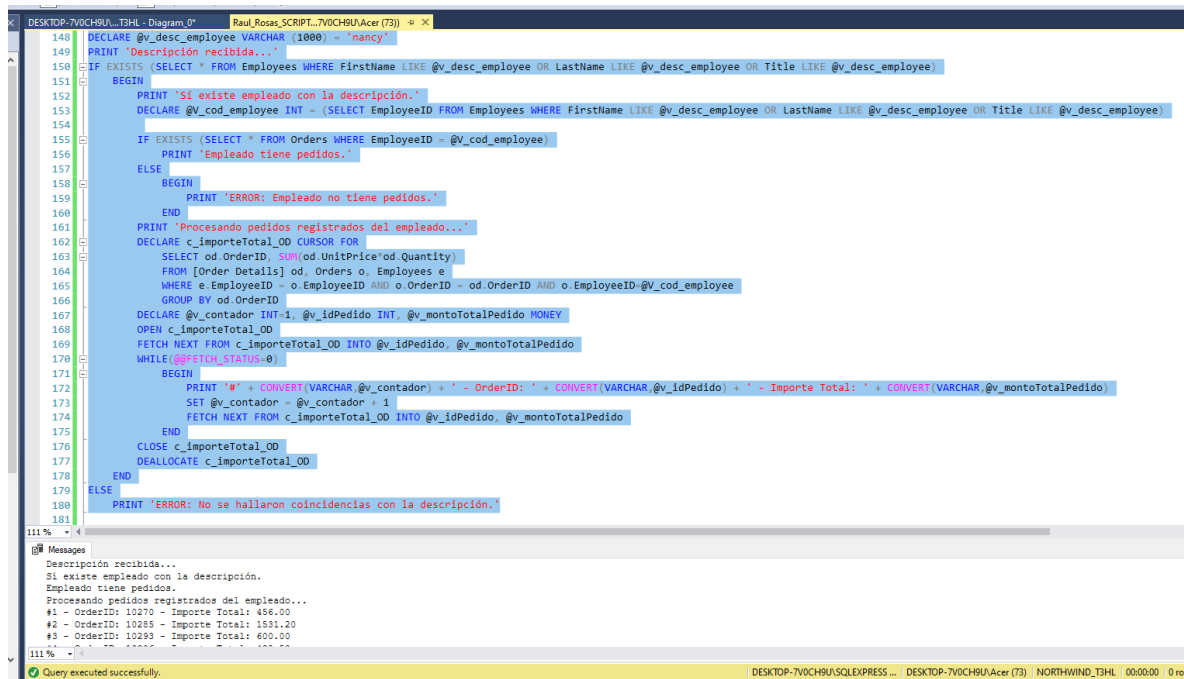
(\*) Por cada ejecución, capture la pantalla de la ejecución y el resultado obtenido, y cópielas aquí.



### **PREGUNTA 3: MANEJO DE VARIABLES Y CONDICIONALES**

Implemente un **bloque de programación T-SQL**, que cumpla con las siguientes validaciones de manera secuencial:

- Que reciba en una variable la descripción de un Empleado (Tabla Employees).
- Validar si exista un Empleado con dicha descripción.  
Si existe, que capture en otra variable el código de dicho Empleado.  
De lo contrario que genere un mensaje de error.
- Sólo si existe el Empleado, validar si hay Pedidos registrados para dicho Empleado.  
Si no hay pedidos, genere un mensaje de error.
- Por cada pedido registrado para el Empleado, calcular el importe total facturado.  
El importe de un pedido se calcula con la sumatoria del resultado de la multiplicación entre la cantidad y el precio unitario de todos los productos involucrados en el pedido.
- Mostrar un mensaje con el importe total facturado por cada pedido.
- Ejecutar el bloque de programación T-SQL, de diferentes maneras, de tal forma que se evidencie cada una de las 5 validaciones desarrolladas (ítems a, b, c, d, e).



```
148 DECLARE @v_desc_employee VARCHAR (1000) = 'nancy'
149 PRINT 'Descripción recibida...'
150 IF EXISTS (SELECT * FROM Employees WHERE FirstName LIKE @v_desc_employee OR LastName LIKE @v_desc_employee OR Title LIKE @v_desc_employee)
151 BEGIN
152     PRINT 'Si existe empleado con la descripción.'
153     DECLARE @v_cod_employee INT = (SELECT EmployeeID FROM Employees WHERE FirstName LIKE @v_desc_employee OR LastName LIKE @v_desc_employee OR Title LIKE @v_desc_employee)
154     IF EXISTS (SELECT * FROM Orders WHERE EmployeeID = @v_cod_employee)
155     PRINT 'Empleado tiene pedidos.'
156     ELSE
157     BEGIN
158         PRINT 'ERROR: Empleado no tiene pedidos.'
159     END
160     PRINT 'Procesando pedidos registrados del empleado...'
161     DECLARE c_importeTotal_OD CURSOR FOR
162     SELECT od.OrderID, SUM(od.UnitPrice*od.Quantity)
163     FROM [Order Details] od, Orders o, Employees e
164     WHERE e.EmployeeID = o.EmployeeID AND o.OrderID = od.OrderID AND o.EmployeeID=@v_cod_employee
165     GROUP BY od.OrderID
166     DECLARE @v_contador INT=1, @v_idPedido INT, @v_montoTotalPedido MONEY
167     OPEN c_importeTotal_OD
168     FETCH NEXT FROM c_importeTotal_OD INTO @v_idPedido, @v_montoTotalPedido
169     WHILE (@@FETCH_STATUS=0)
170     BEGIN
171         PRINT '#' + CONVERT(VARCHAR,@v_contador) + ' - OrderID: ' + CONVERT(VARCHAR,@v_idPedido) + ' - Importe Total: ' + CONVERT(VARCHAR,@v_montoTotalPedido)
172         SET @v_contador = @v_contador + 1
173         FETCH NEXT FROM c_importeTotal_OD INTO @v_idPedido, @v_montoTotalPedido
174     END
175     CLOSE c_importeTotal_OD
176     DEALLOCATE c_importeTotal_OD
177 END
178 ELSE
179 PRINT 'ERROR: No se hallaron coincidencias con la descripción.'
181
```

Messages

Descripción recibida...

Si existe empleado con la descripción.

Empleado tiene pedidos.

Procesando pedidos registrados del empleado...

#1 - OrderID: 10270 - Importe Total: 456.00

#2 - OrderID: 10285 - Importe Total: 1531.20

#3 - OrderID: 10293 - Importe Total: 600.00

Query executed successfully.

Rúbrica				
Puntos	Excelente: 3 puntos	Bueno: 2 puntos		Deficiente: 1 punto
3 Puntos	Completar los puntos a, b, c, d e f. (*)	Completar los puntos a, b, c, d. (*)		Completar los puntos a y b. (*)

(\*) Por cada ejecución, capture la pantalla de la ejecución y el resultado obtenido, y cópielas aquí.

#### **PREGUNTA 4: TRY / CATCH, @@ERROR, RAISERROR**

Implemente un **procedimiento almacenado** que permita:

- Recibir como parámetro de entrada el código del Categoría.
- Validar que el Categoría exista (Tabla Categories).
- Eliminar un determinado Categoría.
- En caso no permita la eliminación de dicho Categoría, controlar el error con un mensaje personalizado.

```

155
156 ALTER PROCEDURE delete_Category (@p_categoryID INT)
157 AS
158 BEGIN
159     IF EXISTS (SELECT * FROM Categories WHERE CategoryID=@p_categoryID)
160     BEGIN TRY
161         DELETE od
162         FROM [Order Details] od INNER JOIN products p
163         ON (od.ProductID = p.ProductID)
164         WHERE p.CategoryID = @p_categoryID
165
166         DELETE FROM Products WHERE CategoryID = @p_categoryID
167
168         DELETE FROM Categories WHERE CategoryID = @p_categoryID
169     END TRY
170     BEGIN CATCH
171         DECLARE @v_error_message VARCHAR (1000), @v_error_severity INT, @v_error_state INT
172         SELECT @v_error_message = ERROR_MESSAGE(), @v_error_severity = ERROR_SEVERITY(), @v_error_state = ERROR_STATE()
173         RAISERROR (@v_error_message,@v_error_severity,@v_error_state)
174         ROLLBACK TRANSACTION
175         THROW
176     END CATCH
177     ELSE
178         PRINT 'Categoría no existe.'
179 END
180
181 EXECUTE delete_Category 5

```

122 %

Messages

Commands completed successfully.

Completion time: 2024-06-25T23:46:33.7194298-05:00

- Validar el funcionamiento de código usando como referencia 3 Categorías, de los cuales 1 de ellos debe ser creado recientemente. Mostrar el Categoría registrado.

176  
177  
178  
179  
180

END  
INSERT INTO Categories (CategoryName, Description) values ('categoriaNueva','Descripcion nueva')  
EXECUTE delete\_Category 5

122 %  
Messages  
(1 row affected)  
Completion time: 2024-06-25T23:49:18.3790904-05:00

178  
179  
180  
181  
182

INSERT INTO Categories (CategoryName, Description) values ('categoriaNueva','Descripcion nueva')  
select \* from Categories  
EXECUTE delete\_Category 5

122 %  
Results  
Messages  

	CategoryID	CategoryName	Description	Picture
1	3	Confections	Desserts, candies, and sweet breads	0x151C2F002000000000000000E0014002100FFFFFFF4269746...
2	4	Dairy Products	Cheeses	0x151C2F002000000000000000E0014002100FFFFFFF4269746...
3	6	Meat/Poultry	Prepared meats	0x151C2F002000000000000000E0014002100FFFFFFF4269746...
4	7	Produce	Dried fruit and bean curd	0x151C2F002000000000000000E0014002100FFFFFFF4269746...
5	8	Seafood	Seaweed and fish	0x151C2F002000000000000000E0014002100FFFFFFF4269746...
6	10	categoriaNueva	Descripcion nueva	NULL

180  
181  
182

select \* from Categories  
EXECUTE delete\_Category 10

122 %  
Messages  
(0 rows affected)  
(0 rows affected)  
(1 row affected)  
Completion time: 2024-06-25T23:50:47.5493004-05:00

182  
183  
184

EXECUTE delete\_Category 10  
EXECUTE delete\_Category 3  
EXECUTE delete\_Category 4

122 %  
Messages  
(334 rows affected)  
(13 rows affected)  
(1 row affected)  
Completion time: 2024-06-25T23:51:17.7579014-05:00

```

182 EXECUTE delete_Category 10
183 EXECUTE delete_Category 3
184 EXECUTE delete_Category 4

```

Messages

(366 rows affected)

(10 rows affected)

(1 row affected)

Completion time: 2024-06-25T23:51:26.8556550-05:00

```

186 select * from Categories
187

```

Results Messages

	CategoryID	CategoryName	Description	Picture
1	6	Meat/Poultry	Prepared meats	0x151C2F00020000000000E0014002100FFFFFFFFF4269746...
2	7	Produce	Dried fruit and bean curd	0x151C2F00020000000000E0014002100FFFFFFFFF4269746...
3	8	Seafood	Seaweed and fish	0x151C2F00020000000000E0014002100FFFFFFFFF4269746...

Rúbrica				
Puntos	Excelente: 3 puntos		Regular: 2 puntos	Deficiente: 1 punto
3 Puntos	Desarrolle el procedimiento almacenado con la eliminación de la Categoría, con validaciones y ejecutado para 3 categorías. (*)		Desarrolle el procedimiento almacenado con la eliminación de la Categoría, sin validaciones. (*)	Define las variables. (*)

(\*) Por cada ejecución, capture la pantalla de la ejecución y el resultado obtenido, y cópielas aquí.