

CURSO : 2393-Base de Datos Avanzado I
DOCENTE : RAMOS CASTAÑEDA Daniel Alexis
SEMESTRE : 2022-I
CICLO : Tercero
SECCIÓN : T3UB
FECHA : martes, 26 de abril del 2022
DURACIÓN : hasta las 10:00 am del miércoles 27 de abril.

NOTA

ALUMNO (A) : CESAR HUMBERTO HERRERA VILLACORTA
CODIGO : I202117188

EVALUACION DE LABORATORIO (CL1)

Logro

Tenga presente que el logro de esta evaluación es que usted, Implemente la creación de una base de datos y sus esquemas, además que implemente las tablas para almacenar los datos de un proceso de negocio real, así como manejar índices.

Consideraciones generales:

- Considerar el orden, la limpieza y la claridad de las respuestas.
- NO olvide subir su archivo dentro del plazo establecido por el Docente.
- No se aceptará el envío del examen CL1 por correo.
- No se aceptarán exámenes enviados posterior al plazo establecido por el Docente
- Enviar sus respuestas o soluciones (los pantallazos de sus soluciones, es opcional) en este mismo documento de **Word** consignado sus nombres completos y RENOMBRANDO el archivo **Word**: CL1_Nombre_Apellido_T3XX.docx

Consolidado

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	04			
2	08			
3	04			
4	02			
5	02			
Nota Recalificada				

PREGUNTA 1

- a. Implemente una base de datos llamada **BD_LAP** la cual está conformada 01 DataFile y 01 LogFile con las siguientes propiedades:

	DATAFILE	LOGFILE
Name	LAP_Data	LAP_Log
FileName	D:\Exam\LAP_Data.MDF	D:\Exam\LAP_Log.LDF
Size	10MB	7MB
MaxSize	800MB	300MB
FileGrowth	10MB	20%

```
/*
* PREGUNTA 1a
*/
IF EXISTS(SELECT name FROM sys.databases WHERE name='DB_LAP')
    DROP DATABASE DB_LAP
GO
CREATE DATABASE DB_LAP
ON (
    name = LAP_Data,
    filename = 'D:\Exam\LAP_Data.MDF',
    size = 10,
    maxsize = 800,
    filegrowth = 10
)
LOG ON (
    name = LAP_Log,
    filename = 'D:\Exam\LAP_Log.LDF',
    size = 7,
    maxsize = 300,
    filegrowth = 20%
)
GO
EXEC sp_helpdb DB_LAP
GO
```

100 %								
Results Messages								
	name	db_size	owner	dbid	created	status	compatibility_level	
1	DB_LAP	17.00 MB	sa	6	Apr 26 2022	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	150	
	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	LAP_Data	1	D:\Exam\LAP_Data.MDF	PRIMARY	10240 KB	819200 KB	10240 KB	data only
2	LAP_Log	2	D:\Exam\LAP_Log.LDF	NULL	7168 KB	307200 KB	20%	log only

- b. El administrador de datos desea adicionar tres FileGroups: FGLAP01, FGLAP02 y FGLAP03 a la base de datos creada y debe agregar tres DataFile NDF ((LAP_DataN1, LAP_DataN2 y LAP_DataN3), estarán ubicados en **D:\Eval**, debiendo ser mapeados cada archivo en sus FILEGROUP respectivamente. (Puede ser cualquier unidad lógica de su computador).

```

/*
* PREGUNTA 1b
*/
-- Creacion de filegroups
ALTER DATABASE DB_LAP ADD FILEGROUP FGLAP01
GO
ALTER DATABASE DB_LAP ADD FILEGROUP FGLAP02
GO
ALTER DATABASE DB_LAP ADD FILEGROUP FGLAP03
GO
--Asignacion de archivos a los filegroups
ALTER DATABASE DB_LAP
ADD FILE(
    name = LAP_DataN1,
    filename = 'D:\Eval\LAP_DataN1.NDF',
    size = 10,
    maxsize = 300,
    filegrowth = 10
) TO FILEGROUP FGLAP01
GO
ALTER DATABASE DB_LAP
ADD FILE(
    name = LAP_DataN2,
    filename = 'D:\Eval\LAP_DataN2.NDF',
    size = 10,
    maxsize = 300,
    filegrowth = 10
) TO FILEGROUP FGLAP02
GO
ALTER DATABASE DB_LAP
ADD FILE(
    name = LAP_DataN3,
    filename = 'D:\Eval\LAP_DataN3.NDF',
    size = 10,
    maxsize = 300,
    filegrowth = 10
) TO FILEGROUP FGLAP03
GO
EXEC sp_helpdb DB_LAP
GO

```

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	LAP_Data	1	D:\Exam\LAP_Data.MDF	PRIMARY	10240 KB	819200 KB	10240 KB	data only
2	LAP_Log	2	D:\Exam\LAP_Log.LDF	NULL	7168 KB	307200 KB	20%	log only
3	LAP_DataN1	3	D:\Eval\LAP_DataN1.NDF	FGLAP01	10240 KB	307200 KB	10240 KB	data only
4	LAP_DataN2	4	D:\Eval\LAP_DataN2.NDF	FGLAP02	10240 KB	307200 KB	10240 KB	data only
5	LAP_DataN3	5	D:\Eval\LAP_DataN3.NDF	FGLAP03	10240 KB	307200 KB	10240 KB	data only

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 3 puntos	Regular: 2 puntos	Deficiente: 1 punto
04 Puntos	Crea la Base de Datos con opciones del datafile MDF y NDF organizados en Filegroups	Crea la Base de Datos con opciones del datafile MDF y NDF.	Crea la Base de Datos con opciones del datafile MDF. O existe ambigüedades.	Crea la Base de Datos sin definir opciones del datafile. O existe muchas ambigüedades.

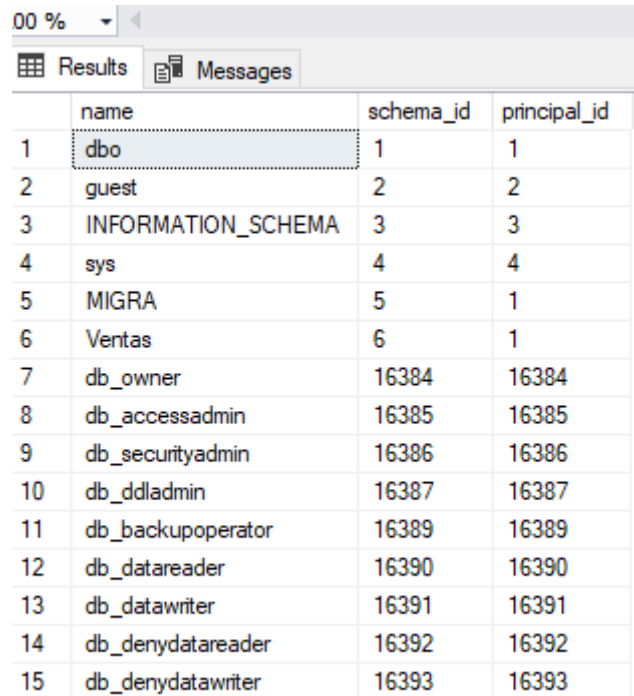
PREGUNTA 2

- a. En la Base de datos, cree los siguientes Schemas: MIGRA y Ventas.

```

/*
* PREGUNTA 2a
*/
CREATE SCHEMA MIGRA
GO
CREATE SCHEMA Ventas
GO
SELECT * FROM sys.schemas
GO

```



	name	schema_id	principal_id
1	dbo	1	1
2	guest	2	2
3	INFORMATION_SCHEMA	3	3
4	sys	4	4
5	MIGRA	5	1
6	Ventas	6	1
7	db_owner	16384	16384
8	db_accessadmin	16385	16385
9	db_securityadmin	16386	16386
10	db_ddladmin	16387	16387
11	db_backupoperator	16389	16389
12	db_datareader	16390	16390
13	db_datawriter	16391	16391
14	db_denydatareader	16392	16392
15	db_denydatawriter	16393	16393

- b. Cree los siguientes tipos de datos definidos por el usuario:

Tipo de Dato de Usuario		Tipo de dato del sistema	Característica
TD_CADENA	basado en	VARCHAR(80)	NOT NULL
TD_FECHA	basado en	DATE	NULL
TD_MONEDA	basado en	SMALLMONEY	NOT NULL

```

/*
* PREGUNTA 2b
*/
USE DB_LAP
CREATE TYPE TD_CADENA FROM VARCHAR(80) NOT NULL
GO
CREATE TYPE TD_FECHA FROM DATE NULL
GO
CREATE TYPE TD_MONEDA FROM SMALLMONEY NOT NULL
GO
exec sp_help
GO

```

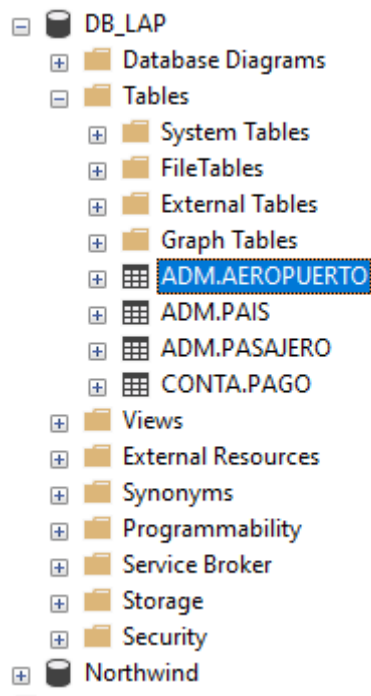
	User_type	Storage_type	Length	Prec	Scale	Nullable	Default_name	Rule_name	Collation
1	TD_CADENA	varchar	80	80	NULL	no	none	none	Modem_Spanish_CI_AS
2	TD_FECHA	date	3	10	0	yes	none	none	NULL
3	TD_MONEDA	smallmoney	4	10	4	no	none	none	NULL

- c. Cree las tablas en el schema **ADM** y **CONTA**, en el Filegroup **[FGLAP02]**, con llaves primarias, llaves foráneas, relaciones. Utilice los 3 tipos de datos creado por Ud. en algunos o todos los campos.

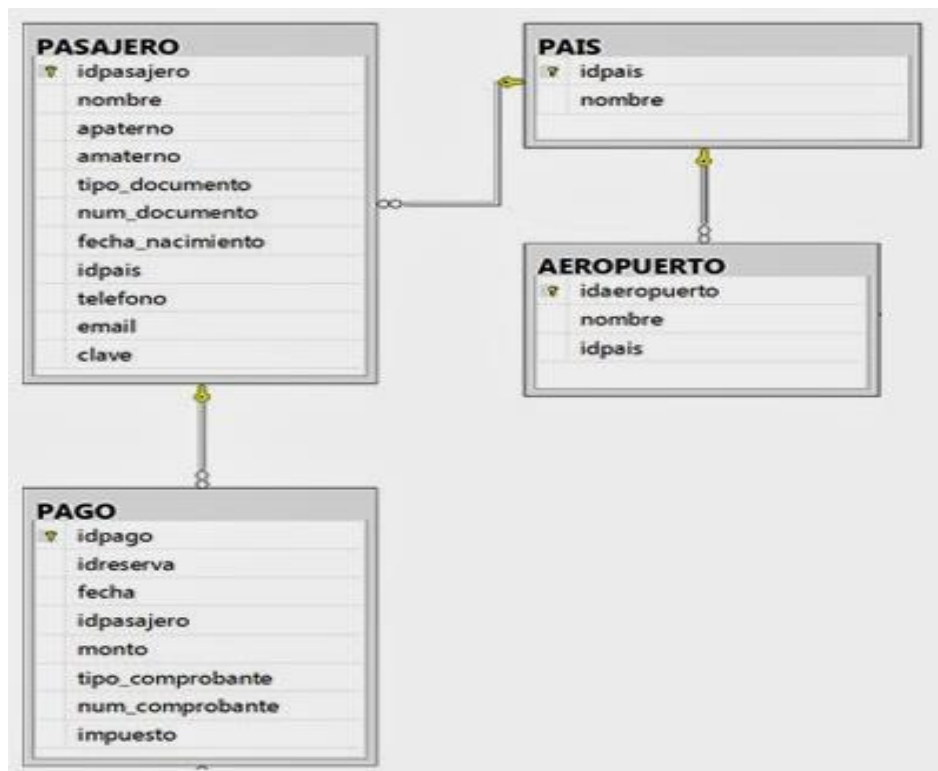
```

/*
* PREGUNTA 2c
*/
-- Creando los schemas ADM y CONTA
CREATE SCHEMA ADM
GO
CREATE SCHEMA CONTA
GO
-- Creando las tablas
-- En el schema ADM (Administrativo)
CREATE TABLE ADM.PAIS(
    idpais INT IDENTITY PRIMARY KEY,
    nombre TD_CADENA NOT NULL
) ON FGLAP02
GO
CREATE TABLE ADM.AEROPUERTO(
    idaeropuerto INT IDENTITY PRIMARY KEY,
    nombre TD_CADENA NOT NULL,
    idpais INT FOREIGN KEY REFERENCES ADM.PAIS(idpais) NOT NULL
) ON FGLAP02
GO
CREATE TABLE ADM.PASAJERO(
    idpasajero INT IDENTITY PRIMARY KEY,
    nombre TD_CADENA NULL,
    apaterno TD_CADENA NULL,
    amaterno TD_CADENA NULL,
    tipo_documento TD_CADENA NOT NULL,
    num_documento TD_CADENA UNIQUE NOT NULL,
    fecha_nacimiento TD_FECHA NULL,
    idpais INT FOREIGN KEY REFERENCES ADM.PAIS(idpais) NOT NULL,
    telefono VARCHAR(15) NULL,
    email TD_CADENA UNIQUE NOT NULL,
    clave TD_CADENA NULL
) ON FGLAP02
GO
-- En el schema CONTA (Contabilidad)
CREATE TABLE CONTA.PAGO(
    idpago INT IDENTITY PRIMARY KEY,
    idreserva INT NOT NULL,
    fecha TD_FECHA NULL,
    idpasajero INT FOREIGN KEY REFERENCES ADM.PASAJERO(idpasajero) NOT NULL,
    monto TD_MONEDA NULL,
    tipo_comprobante TD_CADENA NULL,
    num_comprobante TD_CADENA UNIQUE NOT NULL,
    impuesto TD_MONEDA NULL
) ON FGLAP02
GO

```



Se tiene el siguiente modelo de datos el cuál debe representar:



- d. Así mismo, implemente restricciones para los campos Monto de la tabla PAGO para que el valor sea MAYOR o IGUAL 0 y el campo Impuesto debe ser MAYOR O IGUAL A CERO.

```

/*
* PREGUNTA 2d
*/
ALTER TABLE CONTA.PAGO
ADD CONSTRAINT monto_mayor_igual_0 CHECK (monto >= 0);
GO
ALTER TABLE CONTA.PAGO
ADD CONSTRAINT impuesto_mayor_igual_0 CHECK (impuesto >= 0);
GO
EXEC sp_help 'CONTA.PAGO'
GO

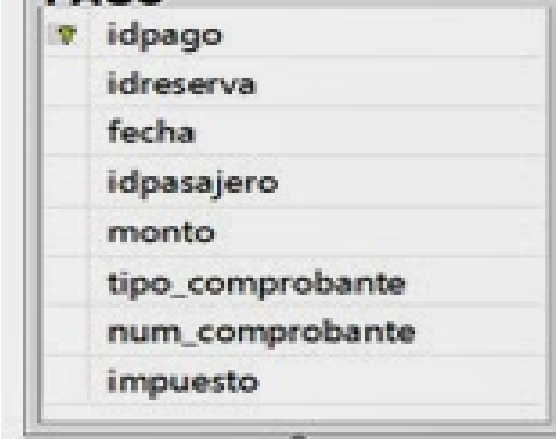
```

	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	FK_PAGO_idpasajero__2F10007B	No Action	No Action	Enabled	Is_For_Replication	idpasajero
2							REFERENCES DB_LAP.ADM.PASAJERO (idpasajero)
3	CHECK on column impuesto	impuesto_mayor_igual_0	(n/a)	(n/a)	Enabled	Is_For_Replication	((impuesto)>=(0))
4	CHECK on column monto	monto_mayor_igual_0	(n/a)	(n/a)	Enabled	Is_For_Replication	((monto)>=(0))
5	PRIMARY KEY (clustered)	PK_PAGO_55A035DE17809BA0	(n/a)	(n/a)	(n/a)	(n/a)	idpago
6	UNIQUE (non-clustered)	UQ_PAGO_3D2B865742F815BA	(n/a)	(n/a)	(n/a)	(n/a)	num_comprobante

Rúbrica				
Puntos	Excelente: 8 puntos	Bueno: 6 puntos	Regular: 4 puntos	Deficiente: 2 puntos
08 Puntos	Crea las tablas del modelo en el schema y filegroup señalado, utilizando los tipos de datos definidos por el usuario, llaves primarias, foráneas y las restricciones requeridas.	Crea las tablas del modelo en el schema y filegroup señalado, utilizando los tipos de datos definidos por el usuario, llaves primarias, foráneas y las restricciones requeridas. Existen máximo 3 errores simples.	Crea las tablas del modelo en el schema y filegroup señalado, utilizando los tipos de datos definidos por el usuario, llaves primarias, foráneas y las restricciones requeridas. Existen máximo 4 errores simples.	Existen muchos errores de funcionalidad.

Pregunta 3

Cree una tabla particionada llamada **AUDITORIA_PAGO**, contenida en el Schema **CONTA**, la cual está definida:



idpago
idreserva
fecha
idpasajero
monto
tipo_comprobante
num_comprobante
impuesto

Dicha partición deberá definirse bajo el siguiente intervalo: los registros corresponden a fechas menores al año 2017 las cuales se almacenarán en partición 1 (FGLAP01), entre el año 2017 y 2019 en la partición 2 (FGLAP02) y el resto en partición 3 (FGLAP03). Debe crear el esquema de partición.

```

/*
* PREGUNTA 3
*/
-- Creando la funcion de particion
/*NOTA IMPORTANTE:
    range left (1,100) divide la particion en p1<=1; 1<p2<=100; p3>100
    range right (1,100) divide la particion en p1<1; 1<=p2<100; p3>=100
    Por ende uno de los lados siempre sera INCLUSIVO
    Existen 2 opciones:
        range left for values ('2016-12-31','2019-12-31')
        range right for values ('2017-01-01','2020-01-01')
*/
CREATE PARTITION FUNCTION fnpFechas(DATE)
AS RANGE right FOR VALUES ('2017-01-01','2020-01-01')
GO
-- Creando el esquema de particion
CREATE PARTITION SCHEME scpFechas
AS PARTITION fnpFechas
to('FGLAP01','FGLAP02','FGLAP03')
GO
-- Creando la tabla AUDITORIA_PAGO (copia de CONTA.PAGO)
CREATE TABLE CONTA.AUDITORIA_PAGO(
    idpago INT IDENTITY,
    idreserva INT NOT NULL,
    fecha DATE NULL,
    idpasajero INT FOREIGN KEY REFERENCES ADM.PASAJERO(idpasajero) NOT NULL,
    monto TD_MONEDA NULL,
    tipo_comprobante TD_CADENA NULL,
    num_comprobante TD_CADENA NOT NULL,
    impuesto TD_MONEDA NULL
) ON scpFechas(fecha)
GO
ALTER TABLE CONTA.AUDITORIA_PAGO
ADD CONSTRAINT aud_monto_mayor_igual_0 CHECK (monto >= 0);
GO
ALTER TABLE CONTA.AUDITORIA_PAGO
ADD CONSTRAINT aud_impuesto_mayor_igual_0 CHECK (impuesto >= 0);
GO
EXEC sp_help 'CONTA.AUDITORIA_PAGO'
GO

```

Data_located_on_filegroup							
1	scpFechas						
	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	CHECK on column impuesto	aud_impuesto_mayor_igual_0	(n/a)	(n/a)	Enabled	Is_For_Replication	((impuesto)>=(0))
2	CHECK on column monto	aud_monto_mayor_igual_0	(n/a)	(n/a)	Enabled	Is_For_Replication	((monto)>=(0))
3	FOREIGN KEY	FK_AUDITORIA_idpas...	No Action	No Action	Enabled	Is_For_Replication	idpasajero
4							REFERENCES DB_LAP.ADM.PASAJERO (idpasajero)

Rúbrica				
Puntos	Excelente: 4 puntos	Bueno: 3 puntos		Deficiente: 0 puntos

04 Puntos	Crea la función de partición definiendo los rangos, el esquema de partición y la tabla particionada correctamente.	Crea la función de partición y de esquema con imprecisiones.		No resuelve la pregunta
-----------	--------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------	--	-------------------------

PREGUNTA 4

- Ingrese registros a la tabla particionada.
- Listar los registros de la tabla particionada, donde figura el número de la partición.

```

/*
* PREGUNTA 4
*/
--Insertando datos para otras tablas
INSERT INTO ADM.PAIS VALUES ('Peru')
GO
INSERT INTO ADM.PASAJERO VALUES('Cesar', 'Herrera','Villacorta', 'DNI',
'46656049', '1990-11-23',1,null,'elch.herrera@gmail.com','1234')
GO
-- Insertando datos con fechas claves
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2015-07-18',1,50,'boleta','B0001',5)
GO
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2017-01-01',1,50,'boleta','B0002',5)
GO
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2019-12-31',1,50,'boleta','B0003',5)
GO
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2020-07-14',1,50,'boleta','B0004',5)
GO
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2016-12-31',1,50,'boleta','B0005',5)
GO
INSERT INTO CONTA.AUDITORIA_PAGO VALUES (1,'2020-01-01',1,50,'boleta','B0006',5)
GO
-- Comprobando el filegroup
SELECT *, $partition.fnpFechas(fecha) as NroParticion
FROM CONTA.AUDITORIA_PAGO
GO

```

100 %

Results

Messages

	idpago	idreserva	fecha	idpasajero	monto	tipo_comprobante	num_comprobante	impuesto	NroParticion
1	1	1	2015-07-18	1	50.00	boleta	B0001	5.00	1
2	5	1	2016-12-31	1	50.00	boleta	B0005	5.00	1
3	2	1	2017-01-01	1	50.00	boleta	B0002	5.00	2
4	3	1	2019-12-31	1	50.00	boleta	B0003	5.00	2
5	4	1	2020-07-14	1	50.00	boleta	B0004	5.00	3
6	6	1	2020-01-01	1	50.00	boleta	B0006	5.00	3

Rúbrica				
Puntos	Excelente: 2 puntos	Bueno: 1 punto		Deficiente: 0 puntos

02 Puntos	Ingresa registros a la tabla particionada y los consulta en la partición correspondiente.	Ingresa registros a la tabla particionada y los consulta con imprecisiones.		No resuelve la pregunta
-----------	-------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------	--	-------------------------

Pregunta 5

Cree un Índice compuesto (apPaterno y fecha nacimiento del **PASAJERO**) ordenado de forma descendente, coloque la consulta de comprobación de funcionamiento del índice.

```

/*
* PREGUNTA 5
*/
CREATE INDEX idx_Pat_Fec ON ADM.PASAJERO(apaterno DESC, fecha_nacimiento DESC)
GO
EXEC sp_helpindex 'ADM.PASAJERO'
GO

```

100 %			
Results Messages			
	index_name	index_description	index_keys
1	idx_Pat_Fec	nonclustered located on FGLAP02	apaterno(-), fecha_nacimiento(-)
2	PK__PASAJERO__FB4104BED2E5F070	clustered, unique, primary key located on FGLAP02	idpasajero
3	UQ__PASAJERO__7BBF0F6E8D842DEB	nonclustered, unique, unique key located on FGLAP02	num_documento
4	UQ__PASAJERO__AB6E616416DB0483	nonclustered, unique, unique key located on FGLAP02	email

Rúbrica				
Puntos	Excelente: 2 puntos	Bueno: 1 punto		Deficiente: 0 puntos
02 Puntos	Crea Índice compuesto, con inclusión de campos y ordenado de forma descendente.	Crea Índice compuesto.		No resuelve la pregunta