

SPRINT 3

Nivel 1

Ejercicio 1: Crear la tabla “credit_card” y diagrama entidad relación.

Creación de la tabla “credit_card” con el comando CREDIT TABLE y creación de los diferentes campos de la tabla necesarios para poder incluir los registros de la base de datos “datos_introducir_credit”.

Todos los campos los he creado para que se tenga que introducir un valor, con la asignación de “Not Null” pensando que podría ser relevante tener toda la información. Quizás sería más necesario en el caso de una base de datos de un banco, donde definitivamente necesita tener estos datos. Al ser una base de datos de una empresa, a lo mejor, son valores que el usuario no va a facilitar en la transacción (como por ejemplo el “pan”).

Los campos “pin” y “cvv” los que he creado del tipo INT, puesto que son valores enteros. Ha sido arriesgado puesto que si hubiera habido algún número pin o cvv empezando por 0, podría haber leído algún número sin considerar el 0. Habría sido más acertado asignarles una tipología CHAR, para no tener este problema y además fijar la longitud del dato. Para el campo “pan” debido a que presenta espacios fue necesario crear este campo de tipo VARCHAR. La misma situación presentan los campos “id” e “iban”, los cuales presentan caracteres y por esta razón es necesario asignar el tipo VARCHAR.

En el caso del campo “expiring_date”, mi primera idea fue darle el tipo DATE. Sin embargo, no fue posible puesto que el formato datos de las fechas en la base de datos, no se corresponde con el formato que acepta DATE. Así que finalmente, utilicé el formato VARCHAR para poder insertar los datos de fecha. Si es necesario, en el futuro se puede cambiar el formato de la fecha.

```

13 # Ejercicio 1
14 -- Creamos la tabla "credit_card" --
15 CREATE TABLE IF NOT EXISTS credit_card (
16     id varchar(15) NOT NULL PRIMARY KEY,
17     iban VARCHAR(50) NOT NULL,
18     pan VARCHAR (30) NOT NULL,
19     pin INT NOT NULL,
20     cvv INT NOT NULL,
21     expiring_date VARCHAR (15) NOT NULL
22 );
23

```

Output ::::::::::::::::::::

Action	Time	Action	Message	Duration / Fetch
show columns from transaction	695 10:22:17		9 row(s) returned	0.000 sec / 0.000 sec
CREATE TABLE IF NOT EXISTS credit_card (id ...	696 10:26:17		0 row(s) affected	0.031 sec

Realizo una selección de todos los campos de la tabla “credit_card” para comprobar que se han creado correctamente y se puede observar que por el momento no hay registros en la tabla.

```

18 •  SELECT * FROM credit_card;
19
< Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid
| id | iban | pan | pin | cvv | expiring_date |
| * | NULL | NULL | NULL | NULL | NULL |

```

Al mostrar las columnas de la tabla “credit_card” se observan las características que le he asignado a cada uno de los campos. Y también que el campo “id” ha sido elegido como clave primaria (PK).

```

18 •  SELECT * FROM credit_card;
19 •  show columns from credit_card;
20
< Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid
| Field | Type | Null | Key | Default | Extra |
| id | varchar(15) | NO | PRI | NULL |
| iban | varchar(50) | NO | | NULL |
| pan | varchar(30) | NO | | NULL |
| pin | int | NO | | NULL |
| cvv | int | NO | | NULL |
| expiring_date | varchar(15) | NO | | NULL |
Result 6 x
Output:
Action Output
# Time Action Message Duration / Fetch
698 10:28:00 SELECT * FROM credit_card LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec

```

En estos momentos, la tabla está diseñada para poder incluir los datos de la base de datos “datos_introducir_credit”. Así que ejecuto el archivo para introducir todos los registros que se incluyen en el documento.

```

8 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
9 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
10 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
11 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
12 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
13 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
14 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
15 •  INSERT INTO credit card (id, iban, pan, pin, cvv, expiring date) VALUES ('CcU-2973', 'CcU-2980', 'CcU-2987', 'CcU-2994', 'CcU-3001', 'CcU-3008', 'CcU-3015', 'CcU-3022');
< Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid
Output:
Action Output
# Time Action Message Duration / Fetch
973 10:36:59 INSERT INTO credit_card (id, iban, pan, pin, cvv,... 1 row(s) affected 0.000 sec

```

Para chequear que los registros se han incluido correctamente en la tabla, hago un `SELECT * FROM credit_card`. Me devuelve **275 registros insertados** en la tabla.

```

18 •   SELECT * FROM credit_card;
19 •   show columns from credit_card;
20

```

Result Grid

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
	CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
	CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	10/29/24
	CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25

credit_card 7 ×

Action Output

#	Time	Action	Message	Duration / Fetch
974	10:36:59	INSERT INTO credit_card (id, iban, pan, pin, cvv, ...)	1 row(s) affected	0.015 sec
975	10:38:08	SELECT * FROM credit_card LIMIT 0, 1000	275 row(s) returned	0.000 sec / 0.000 sec

Aquí podemos ver las columnas de la tabla “transaction” y vemos cual es su PK (“id”) y las claves externas (FK), en estos momentos solo tiene una FK que es el campo “company_id” (indicado en la columna “key” con las letras **MUL**) y que la relaciona con la tabla “company”.

```

3
4 •   show tables;
5 •   show columns from transaction;
6

```

Result Grid

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(255)	NO	PRI	NULL	
	credit_card_id	varchar(15)	YES		NULL	
	company_id	varchar(20)	YES	MUL	NULL	
	user_id	int	YES		NULL	
	lat	float	YES	YES	NULL	
	longitude	float	YES		NULL	
	timestamp	timestamp	YES		NULL	
	amount	decimal(10,2)	YES		NULL	
	dedined	tinyint(1)	YES		NULL	

Result Grid

Form Editor

Field Types

La nueva tabla, es una tabla de dimensiones que se ha de relacionar con la tabla “transaction” que es la tabla de hechos. Estas dos tablas se relacionan a través del campo “id” de la tabla “credit_card” con el campo “credit_card_id” de la tabla “transaction”. Esta relación tiene una **cardinalidad de 1:N** (por cada registro de la tabla “credit_card_id” podemos tener muchos registros en la tabla “transaction”). Así que he incluido en la tabla “transaction” una clave externa (FK) relacionando estos dos campos.

```

21   -- Modifico la tabla 'transaction' para añadir una relación de 1:N con la tabla 'credit_card'
22 •   ALTER TABLE transaction
23     ADD FOREIGN KEY FK_tarjeta_transaccion(credit_card_id) REFERENCES credit_card(id);
24

```

Action Output

#	Time	Action	Message	Duration / Fetch
975	10:38:08	SELECT * FROM credit_card LIMIT 0, 1000	275 row(s) returned	0.000 sec / 0.000 sec
976	10:41:49	ALTER TABLE transaction ADD FOREIGN KEY ...	587 row(s) affected Records: 587 Duplicates: 0 ...	0.093 sec

Ahora podemos ver en la columna *credit_card_id* que se ha creado una nueva FK para la tabla “transaction”.

19 • show columns from transaction;

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
credit_card_id	varchar(15)	YES	MUL	NULL	
company_id	varchar(20)	YES	MUL	NULL	
user_id	int	YES		NULL	
lat	float	YES		NULL	

Result 8 ×

Output

#	Time	Action	Message	Duration / Fetch
976	10:41:49	ALTER TABLE transaction ADD FOREIGN KEY ...	587 row(s) affected Records: 587 Duplicates: 0 ...	0.093 sec
977	10:43:16	show columns from transaction	9 row(s) returned	0.000 sec / 0.000 sec

El modelo se ha ampliado con una nueva tabla “***credit_card***”, como ya dije anteriormente es una tabla de dimensiones, al igual que la tabla “***company***” que se relacionan con una cardinalidad 1:N con la tabla de métricas o de hechos, “***transaction***”. En la figura del Diagrama Entidad Relación, se pueden observar el color verde los campos que se relacionan con la tabla de métricas.

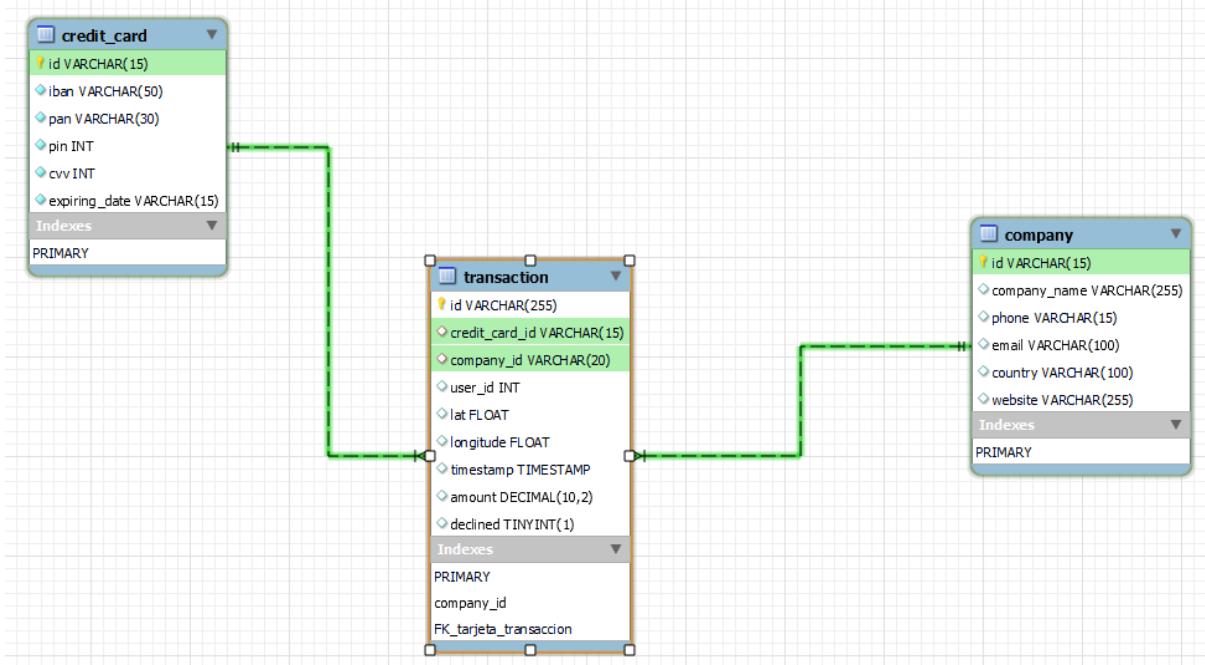


Fig. 1: Diagrama Entidad Relación

Ejercicio 2: Modificar un registro de una tabla.

Para solucionar el error en el número de cuenta del usuario con *ID= CcU-2938*, se ha modificado el registro con la cláusula *UPDATE nombre_tabla SET nuevo_dato WHERE condicion*. Gracias a la condición podemos identificar el registro a modificar, si no actualizaremos todos los registros de la tabla con el mismo iban.

```

25      # Ejercicio 2
26      -- Cambio del registro del usuario con ID CcU-2938 --
27 •   UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';
28

```

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	977	10:43:16	show columns from transaction	9 row(s) returned	0.000 sec / 0.000 sec
	978	10:52:16	UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938';	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec

Si queremos comprobar que el cambio se ha realizado correctamente, buscaremos el registro con un SELECT y vemos que el iban se ha actualizado a R323456312213576817699999

```

32 •   SELECT iban, id FROM credit_card
33      WHERE id = 'CcU-2938';

```

Result Grid:

iban	id
R323456312213576817699999	CcU-2938
NULL	CcU-2938

Ejercicio 3: Ingresar un nuevo registro en una tabla

Para poder ingresar un nuevo registro en la tabla de métricas, es necesario comprobar si existen los registros relacionados con las tablas de dimensiones. Como se puede observar no existen dichos registros y, por lo tanto, debemos incluirlos en las tablas “company” y “credit_card” antes de introducir el registro solicitado.

```

35      # Ejercicio 3
36      -- Insertamos un nuevo registro en la tabla transaction --
37      # Primero comprobamos si existen los registros relacionados con una FK en las tablas de dimensiones.
38 •   SELECT * FROM credit_card WHERE id = 'CcU-9999';
39 •   SELECT id FROM company WHERE id = 'b-9999';

```

Result Grid:

id	iban	pan	pin	cvv	expiring_date
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 14 x

Output:

Action Output	#	Time	Action	Message	Duration / Fetch
	985	11:09:35	SELECT id FROM company WHERE id = 'b-9999' LIMIT 1;	0 row(s) returned	0.000 sec / 0.000 sec
	986	11:10:10	SELECT * FROM credit_card WHERE id = 'CcU-9999' ...	0 row(s) returned	0.000 sec / 0.000 sec

El siguiente paso es introducir el registro *id = “b-9999”* en la tabla de “company” y el registro de una nueva tarjeta de crédito con *id= “CcU-9999”* en la tabla “credit_card”.

```

41      # Como no existen las claves foráneas en las tablas con la relación,
42      # debemos insertar primero estos registros en las tablas de dimensiones.
43 •  INSERT INTO company (id) VALUES ( 'b-9999' );
44 •  UPDATE company SET company_name = 'Unluxe', phone = '34 600 325 017', email = 'unluxe@dmail', country
45 WHERE id = 'b-9999';
46 •  INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
47 VALUES ( 'CcU-9999', 'ES45 4567 8900 1234 4567', '5424465566898765', '5342', '265', '12/29/22');

```

Output:

Action Output				
#	Time	Action	Message	Duration / Fetch
987	11:28:46	INSERT INTO company (id) VALUES ('b-9999')	1 row(s) affected	0.015 sec
988	11:28:51	UPDATE company SET company_name = 'Unluxe', ph...	1 row(s) affected Rows matched: 1 Changed: 1 Wami...	0.016 sec
989	11:29:33	INSERT INTO credit_card (id, iban, pan, pin, cvv, expir...	1 row(s) affected	0.016 sec

Ahora está la tabla preparada para introducir el nuevo registro solicitado en la tabla “transaction”.

```

49      # Ahora podemos insertar el nuevo registro de transacción.
50 •  INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
51 VALUES ( '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0' );

```

Output:

Action Output				
#	Time	Action	Message	Duration / Fetch
988	11:28:51	UPDATE company SET company_name = 'Unluxe', phone = '34 600 3...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
989	11:29:33	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VAL...	1 row(s) affected	0.016 sec
990	11:31:53	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, ...	1 row(s) affected	0.000 sec

Aquí compruebo que se ha incluido correctamente.

```

53 •  SELECT * FROM transaction
54 WHERE company_id = 'b-9999';
55

```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Wrap Cell Content: []

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	HULL	111.11	0
*	NULL	NULL	HULL	HULL	NULL	NULL	HULL	NULL	NULL

Ejercicio 4: Eliminar un campo de una tabla.

Elimino la columna “pan” con el comando `ALTER TABLE nombre_tabla DROP COLUMN nombre_columna`. Y compruebo que se ha eliminado la tabla con un `SELECT` de todos los campos. Se puede observar que el campo “pan” ya no existe en la tabla “credit_card”.

```

56      # Ejercicio 4
57      -- Eliminar la columna 'pan' de la tabla 'credit_card'
58 •  ALTER TABLE credit_card DROP COLUMN pan;
59 •  SELECT * FROM credit_card;

```

Result Grid

	id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22	
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23	
CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21	
CcU-2959	CR7242477244335841535	3583	667	02/24/23	
CcU-2966	KTQ15627628377363	4900	130	10/29/24	

Action Output

#	Time	Action	Message	Duration / Fetch
992	11:34:19	SELECT * FROM transaction WHERE company_id = 'b-9999' LIMIT 0, ...	1 row(s) returned	0.000 sec / 0.000 sec
993	12:13:43	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
994	12:14:16	SELECT * FROM credit_card LIMIT 0, 1000	276 row(s) returned	0.000 sec / 0.000 sec

Nivel 2

Ejercicio 1: Eliminar un registro de la tabla “transaction”.

Antes de proceder a eliminar el registro y para comprobar que la eliminación se ha llevado a cabo, selecciono el registro que voy a eliminar de la tabla “transaction”.

```

6 •  SELECT *
7   FROM transaction
8   WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
9
10 • DELETE FROM transaction
11   WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";

```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92

A continuación, procedo a eliminar el registro deseado, usando el comando DELETE FROM. Y con una condición que identifica únicamente el registro que deseo eliminar gracias a un WHERE y el número identificador de la transacción.

Una vez ejecutada la acción, voy a comprobar si ha sido eliminada, ejecutando de nuevo la consulta anterior de SELECT con el *id* de la transacción que he eliminado. En esta ocasión me devuelve cero filas. Por lo tanto, el registro ha sido eliminado de la base de datos.

```
13 •  SELECT *
14   FROM transaction
15   WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Action Output

Time	Response	Duration / Fetch Time
76 16:17:22	sl 9 row(s) returned	0.056 sec / 0.00011 s...
77 16:22:24	S 1 row(s) returned	0.0034 sec / 0.00000...
78 16:27:28	D 1 row(s) affected	0.0099 sec
79 16:30:09	S 0 row(s) returned	0.00095 sec / 0.000...

Ejercicio 2: Crear una vista llamada “VistaMarketing”

Primero de todo, realizo la consulta que me piden guardar en la vista *VistaMarketing* para comprobar que me devuelve lo que necesito.

```
20  # Crear una vista, llamada VistaMarketing, con los siguientes datos:
21  # Nombre de la compañía, tlf de contacto, país y media de compras realizadas
22
23  # CREATE VIEW VistaMarketing AS
24 •  SELECT company_name AS nombreCompania, phone AS tlfcontacto, country AS pais, round(AVG(amount))
25    FROM company
26    JOIN transaction
27    ON transaction.company_id = company.id
28    WHERE declined = 0
29    GROUP BY nombreCompania, tlfcontacto, pais
30    ORDER BY PromedioCompras DESC;
```

nombreCompania	tlfcontacto	pais	PromedioCompras
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10

Action Output

Time	Response	Duration / Fetch Time
91 18:12:42	U 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.0078 sec
92 18:13:00	S 101 row(s) returned	0.0059 sec / 0.00002...
93 18:13:21	U 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.0019 sec
94 18:14:08	S 101 row(s) returned	0.0063 sec / 0.0000...

Posteriormente, ejecuto la creación de la nueva vista “*VistaMarketing*” con el comando CREATE VIEW. Como se puede observar en la imagen en la columna de “Schemas” dentro del apartado “View” se ha creado una nueva vista con el nombre adjudicado.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows the 'transactions' schema with its tables ('company', 'transaction', 'credit_card'), views ('vistamarketing'), and other objects. The 'Object Info' tab is selected for the 'vistamarketing' view, which has the following definition:

```

18
19  # Ejercicio 2
20  # Crear una vista, llamada VistaMarketing, con los siguientes datos:
21  # Nombre de la compañía, tlf de contacto, país y media de compras realizadas
22
23 • CREATE VIEW VistaMarketing AS
24  SELECT company_name AS nombreCompania, phone AS tlfcontacto, country AS pais, round(AVG(amour
25  FROM company
26  JOIN transaction
27  ON transaction.company_id = company.id
28  WHERE declined = 0
29  GROUP BY nombreCompania, tlfcontacto, pais
30  ORDER BY PromedioCompras DESC;
31
32
33
34
35

```

The 'Action Output' pane at the bottom shows the execution history with rows 92 through 95.

Finalmente, para comprobar que tengo acceso a esta vista, hago una búsqueda con **SELECT * FROM vistamarketing** y accedo a la información.

The screenshot shows the results of the query **SELECT * FROM vistamarketing;**. The results grid displays the following data:

nombreCompania	tlfcontacto	pais	PromedioCompras
Sed id Limited	07 28 10 10 10	United States	777.10
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

The 'Action Output' pane at the bottom shows the execution history with rows 93 through 96.

Ejercicio 3: Filtra la Vista “VistaMarketing” con las compañías alemanas.

Aprovechando la consulta anterior para comprobar si se había creado correctamente la vista, añado un filtro para el país de Alemania con la cláusula WHERE y me da un total de **8 empresas ubicadas en Alemania**.

```
34      # Ejercicio 3
35      # Filtra la vista VistaMarketing y muestra las compañías que residen en Alemania
36 •  SELECT * FROM vistamarketing
37      WHERE pais = 'Germany';
38
39
40
```

The screenshot shows a database query results grid titled "Result Grid". The query executed was: "SELECT * FROM vistamarketing WHERE pais = 'Germany';". The results are displayed in a table with four columns: nombreCompania, tlfcontacto, pais, and PromedioCompras. The data shows eight companies from Germany with their respective contact numbers and average purchases. The interface includes a sidebar with icons for Result Grid, Form Editor, and navigation arrows. Below the grid, there is an "Action Output" section showing the number of rows returned (8) and the execution time (0.0054 sec / 0.00001...).

nombreCompania	tlfcontacto	pais	PromedioCompras
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.95
Convallis In Incorporated	06 66 57 29 50	Germany	60.99

vistamarketing 11

Action Output

Time	Response	Duration / Fetch Time
97	18:25:21	S 8 row(s) returned 0.0054 sec / 0.00001...

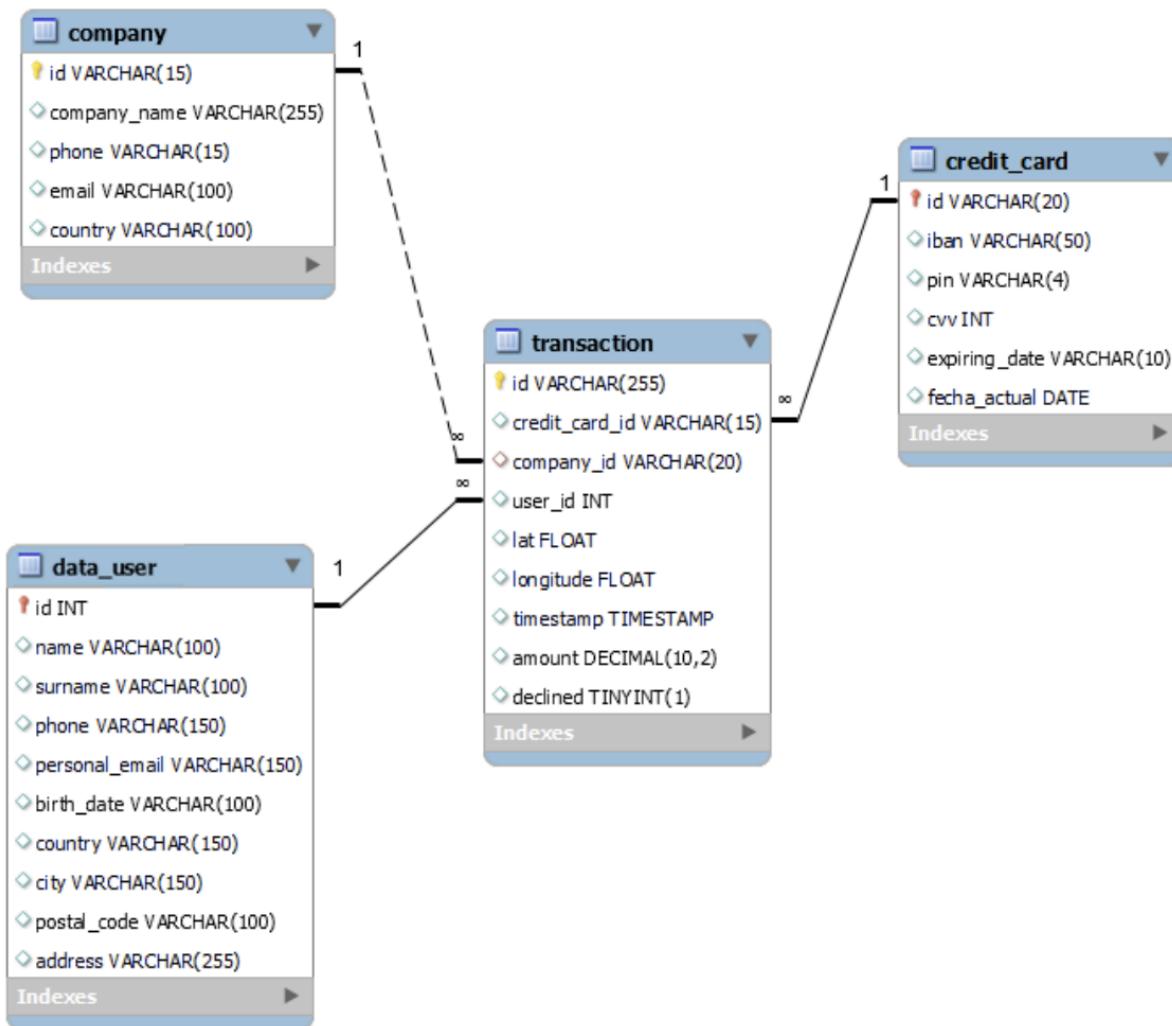
Result Grid

Form Editor

Read Only

Nivel 3

Ejercicio 1: Comandos necesarios para llegar a este Diagrama Entidad Relacional, partiendo del anterior.



Para empezar, empiezo ejecutando los comandos necesarios para hacer los cambios necesarios en las tablas que ya tenemos en el modelo. Posteriormente, crearemos la nueva tabla e incluiremos los datos disponibles para esta tabla.

1A. Tabla “company”

El único cambio detectado en esta tabla es la ausencia de la columna “website” respecto al diagrama anterior. Con la imagen a continuación podemos visualizar el último campo de la tabla “company” que es “website”.

```
12 • SELECT * FROM company;
```

id	company_name	phone	email	country	website
b-2220	Magnis Vtique Inductio	07 77 77 77 77	auditorium.libri@seul.org	Australia	https://whatsapp.com/se
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/se
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/user/1
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/se

Eliminación de la tabla “website” con el comando **ALTER TABLE company DROP COLUMN website**; y comprobación con una visualización de la tabla “company” que ya no existe el campo “website”.

```

10      # Nivel 3
11      # Ejercicio 1
12      -- Eliminar la columna "website" de la tabla "company" --
13 • ALTER TABLE company DROP COLUMN website;
14 • SELECT * FROM company;

```

id	company_name	phone	email	country
b-2220	magistra Neque industries	04 14 44 04 02	risus.dicendum@icloud.org	Australia
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany
b-2238	Ante laculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand
b-2240	Diam sed euismod	01 05 51 07 07	etiam.suscipit@hotmail.co.uk	Namibia

Action Output

Time	Action	Response	Duration / Fetch Time
100 16:42:03	SELECT * FROM company	101 row(s) returned	0.00000 sec / 0.000...
104 16:42:19	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0...	0.028 sec
105 16:42:35	SELECT * FROM company	101 row(s) returned	0.00084 sec / 0.000...

1B. Tabla “credit_card”

- Crear una nueva columna llamada “fecha_actual” de tipo DATE

Para ello se utiliza el comando **ALTER TABLE credit_card ADD fecha_actual DATE**; donde se especifica el nombre de la nueva columna detrás de la cláusula ADD seguida del tipo de dato, en este caso DATE.

```

17      -- Crear una nueva columna en la tabla "credit_card" llamada "fecha_actual" de tipo DATE --
18 • ALTER TABLE credit_card ADD fecha_actual DATE;
19 • show columns from credit_card;

```

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
iban	varchar(50)	NO		NULL	
pin	varchar(4)	NO		NULL	
cvv	int	NO		NULL	
expiring_date	varchar(10)	NO		NULL	
fecha_actual	date	YES		NULL	

Action Output

Time	Action	Response	Duration / Fetch Time
113 17:11:38	ALTER TABLE credit_card ADD fecha_actual DA...	0 row(s) affected Records: 0 Duplicates: 0...	0.047 sec
114 17:11:55	show columns from credit_card	6 row(s) returned	0.012 sec / 0.000020...

- Cambiar la tipología de algunos campos:

Para ello se utiliza el comando **ALTER TABLE credit_card CHANGE pin pin VARCHAR(4) NOT NULL**; donde se especifica el nombre de la tabla que se va a modificar y seguido de la cláusula CHANGE se escribe el nombre la vieja columna y el nombre de la nueva columna, en este caso hemos renombrado las columnas con el nombre anterior, por eso aparecer dos veces repetido el nombre de la columna. Seguidamente, se especifica el tipo de dato y además he añadido un NOT NULL para que sea un campo obligatorio.

- “pin” de INT a VARCHAR (4)
- “id” de VARCHAR(15) a VARCHAR (20)
- “expiring_date” de VARCHAR(15) a VARCHAR (10)

```

11 -- Cambiar los tipos de datos de algunos campos en la tabla "credit_card"
12 • ALTER TABLE credit_card CHANGE pin pin VARCHAR(4) NOT NULL;
13 • ALTER TABLE credit_card CHANGE id id VARCHAR(20) NOT NULL;
14 • ALTER TABLE credit_card CHANGE expiring_date expiring_date VARCHAR(10) NOT NULL;
15 • show columns from credit_card;
16

```

Result Grid

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
iban	varchar(50)	NO		NULL	
pin	varchar(4)	NO		NULL	
cvv	int	NO		NULL	
expiring_date	varchar(10)	NO		NULL	

Action Output

Time	Action	Response	Duration / Fetch Time
107 16:56:15	show columns from credit_card	5 row(s) returned	0.0025 sec / 0.00001...
108 16:56:41	ALTER TABLE credit_card CHANGE pin pin VAR...	0 row(s) affected Records: 0 Duplicates: 0...	0.029 sec
109 16:56:44	show columns from credit_card	5 row(s) returned	0.0019 sec / 0.00001...
110 16:57:45	ALTER TABLE credit_card CHANGE id id VARCH...	0 row(s) affected Records: 0 Duplicates: 0...	0.011 sec
111 16:58:21	ALTER TABLE credit_card CHANGE expiring_dat...	276 row(s) affected Records: 276 Duplicate...	0.025 sec
112 16:58:24	show columns from credit_card	5 row(s) returned	0.0018 sec / 0.00001...

1C. Tabla “data_user”

- Crear la tabla “data_user”

Ejecuto el archivo “estructura_datos_user” con el cual creamos un índice en la tabla “transaction” y creamos la tabla “user”.

```

2   |- Creamos la tabla user
3
4 • CREATE INDEX idx_user_id ON transaction(user_id);
5
6 • CREATE TABLE IF NOT EXISTS user (
7     id INT PRIMARY KEY,
8     name VARCHAR(100),
9     surname VARCHAR(100),
10    phone VARCHAR(150),
11    email VARCHAR(150),
12    birth_date VARCHAR(100),
13    country VARCHAR(150),
14    city VARCHAR(150),
15    postal_code VARCHAR(100),
16    address VARCHAR(255),
17    FOREIGN KEY(id) REFERENCES transaction(user_id)
18 );
19

```

Action Output

Time	Action	Response	Duration / Fetch Time
114 17:11:55	show columns from credit_card	6 row(s) returned	0.012 sec / 0.000020...
115 17:21:12	CREATE INDEX idx_user_id ON transaction(user_id)	0 row(s) affected Records: 0 Duplicates: 0...	0.023 sec
116 17:21:12	CREATE TABLE IF NOT EXISTS user (id IN...	0 row(s) affected	0.00060 sec

Muestro todas las columnas de la nueva tabla “user” para comprobar que se han generado correctamente. Se han generado un total de 10 columnas (campos) de tipo VARCHAR y con la PK el “id”.

```
20 • show columns from user; |
```

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
postal_code	varchar(100)	YES		NULL	
address	varchar(255)	YES		NULL	

Result 12 Read Only

Action Output

Time	Action	Response	Duration / Fetch Time
400 17:37:03	SELECT * FROM user	275 row(s) returned	0.00096 sec / 0.000...
401 17:39:56	show columns from user	10 row(s) returned	0.0042 sec / 0.00001...

- Introducir los datos del archivo “datos_introducir_user”

Ejecuto todo el archivo “datos_introducir_user” con el que también se establece un FK. y después muestro la tabla “user” para comprobar que los datos se han introducido en la tabla. Me salen un total de **275 registros de usuarios**.

```
276 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, ad
277 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, ad
278 • INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, ad
279
280 • SET foreign_key_checks = 1;
```

100% 1:1

Action Output

Time	Action	Response	Duration / Fetch Time
398 17:33:15	INSERT INTO user (id, name, ...	1 row(s) affected	0.00011 sec
399 17:33:15	SET foreign_key_checks = 1	0 row(s) affected	0.000030 sec

```
19 -- Comprobar que se han introducido los datos en la tabla "user"
```

```
20 • SELECT * FROM user; |
```

100% 20:20

Result Grid Filter Rows: Search Edit: Export/Import: Apply

id	name	surname	phone	email	birth_date	country	city	postal_code
1	Carroll	Wiseburn	(703) 555-2714	litteratum.mollis@protonmail.com	Aug 29, 1998	United States	Dos Marmos	55101-1234
2	Ward	Wolcott	(703) 555-2715	aliquam.sed@outlook.net	Oct 15, 1989	United States	Port Huron	58501-1234
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518
4	Howard	Stafford	1-411-740-3269	omare.egestas@cloud.edu	Feb 18, 1989	United States	Kailua	77417
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564
6	Joel	Tyson	(718) 288-8020	gravida.nunc sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1999	United States	Jackson	61750
9	Mannix	Mcclain	(590) 883-2184	aliquam.nisi@outlook.com	Jan 24, 1987	United States	Richmond	35987
10	Robert	McCarthy	(324) 746-6771	fermentum@protonmail.com	Apr 30, 1984	United States	Eugene	85526
11	Joan	Baird	(981) 429-8106	et@outlook.net	Feb 25, 1990	United States	Lincoln	35211
12	Benedict	Wheeler	1-515-824-2855	tincidunt.donec.vitae@hotmail.co.uk	Aug 6, 1999	United States	Lewiston	92393
13	Amelia	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
14	Albert	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
15	Elspeth	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
16	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
17	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
18	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
19	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
20	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
21	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
22	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
23	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
24	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
25	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
26	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
27	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
28	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
29	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
30	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
31	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
32	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
33	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
34	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
35	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
36	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
37	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
38	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
39	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
40	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
41	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
42	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
43	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
44	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
45	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
46	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
47	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
48	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
49	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
50	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
51	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
52	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
53	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
54	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
55	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
56	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
57	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
58	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
59	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
60	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
61	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
62	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
63	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
64	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
65	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
66	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
67	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
68	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
69	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
70	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
71	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
72	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
73	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
74	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
75	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
76	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
77	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
78	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
79	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
80	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
81	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
82	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
83	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
84	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
85	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
86	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
87	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
88	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
89	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
90	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
91	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
92	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
93	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
94	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
95	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
96	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
97	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
98	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
99	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122
100	Frederick	Wolpert	1-800-720-4400	quisque@outlook.com	May 10, 1990	United States	Albuquerque	87122</td

- Cambiar nombre de tabla “user” por “data_user”

Cambio el nombre de la tabla “user” por el de “data_user” que es como se llama en el diagrama ER de referencia con el comando **RENAME TABLE user TO data_user;**

```
20 -- Cambiar el nombre de la tabla "user" por "data_user"
21 • RENAME TABLE user TO data_user;
22 • show tables;
23
```

Action	Time	Response	Duration / Fetch Time
116	17:21:12	CREATE TABLE IF NOT EXISTS user (id IN... 0 row(s) affected	0.0060 sec
117	17:26:29	RENAME TABLE user TO data_user 0 row(s) affected	0.022 sec
118	17:26:43	show tables 5 row(s) returned	0.0050 sec / 0.00001...

- Cambiar nombre de la columna “email” por “personal_email”

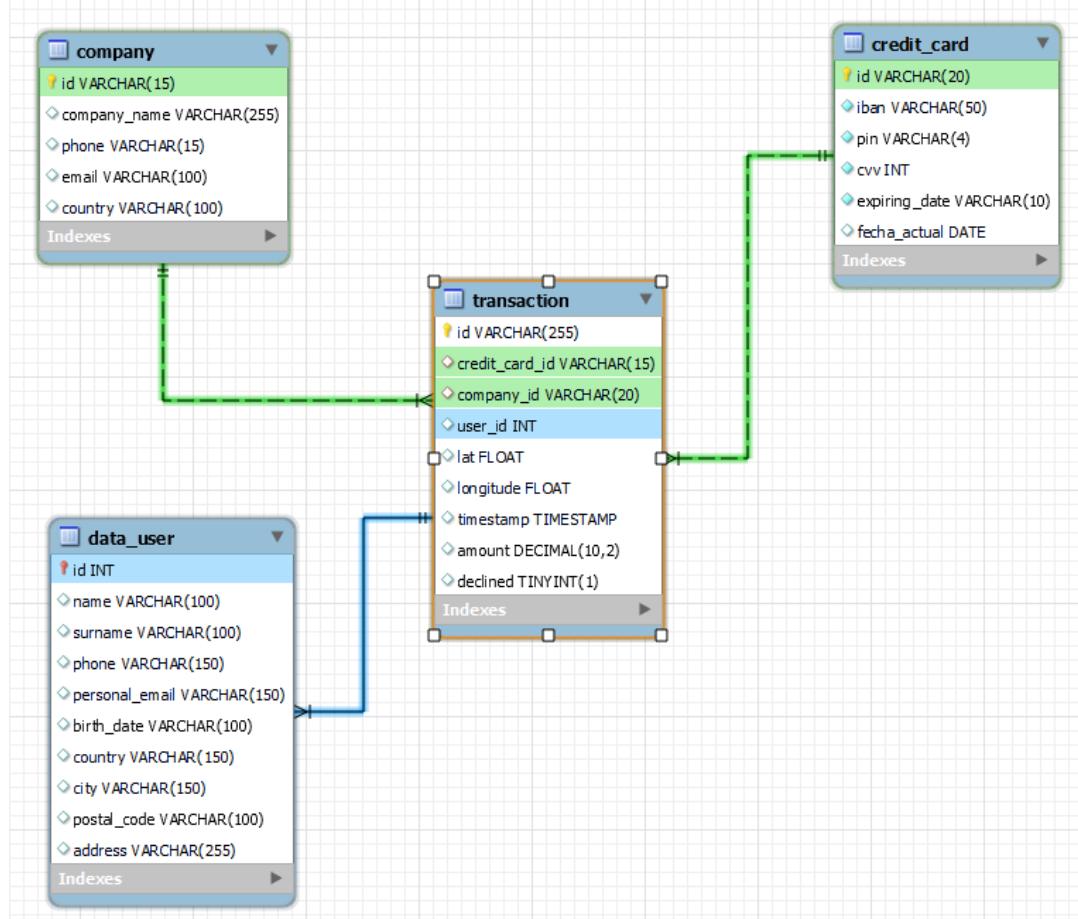
ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);

```
28 -- Cambiar el nombre de la columna "email" por "personal_email" --
29 • ALTER TABLE data_user CHANGE email personal_email VARCHAR(150);
30 • show columns from data_user;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
personal_email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	

Action	Time	Response	Duration / Fetch Time
406	17:59:28	ALTER TABLE data_user CH... 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
407	17:59:49	show columns from data_user 10 row(s) returned	0.0023 sec / 0.00001...

En la siguiente figura se puede ver el Diagrama Entidad Relacional que me ha resultado después de ejecutar todos los anteriores comandos para conseguir el mismo diagrama de referencia.



Ejercicio 2: Crear una vista llamada “InformeTecnico”

Como en el ejercicio anterior se usó el comando CREATE VIEW para crear la vista llamada “InformeTecnico”. En esta vista se han añadido los campos solicitado por la empresa. Además, he considerado que sería relevante también incluir la cantidad de la transacción en el InformeTecnico. Depende la finalidad de este informe quizás también habría sido interesante incluir la columna “declined” para saber si la transacción había sido realizada o no.

```

    transactions
      Tables
        > company
        > credit_card
        > data_user
        > transaction
      Views
        > informetecnico
          vistamarketing
          Stored Procedures
          Functions
    Object Info Session
    View: informetecnico
  
```

```

32  # Ejercicio 2
33  -- Crear una vista llamada "InformeTecnico" --
34
35 • CREATE VIEW InformeTecnico AS
36   SELECT transaction.id AS IdTransaccion, transaction.amount AS Cantidad,
37   data_user.name AS NombreUsuario, data_user.surname AS ApellidoUsuario,
38   credit_card.iban AS IBAN, company.company_name AS NombreCompania
39   FROM company
40   JOIN transaction
41   ON transaction.company_id = company.id
42   JOIN data_user
43   ON transaction.user_id = data_user.id
44   JOIN credit_card
45   ON transaction.credit_card_id = credit_card.id
46   ORDER BY IdTransaccion DESC;
47
  
```

Columns:	
IdTransaccion	varchar(255)
Cantidad	decimal(10,2)
NombreUsuario	varchar(100)
ApellidoUsuario	varchar(100)

Action Output

	Time	Action	Response	Duration / Fetch Time
412	18:18:20	SELECT transaction.id AS IdTransaccion,...	586 row(s) returned	0.0077 sec / 0.0043...
413	18:20:32	CREATE VIEW InformeTecnico AS SELEC...	0 row(s) affected	0.013 sec

Query Completed

Por último, se realiza una SELECT de la vista “*InformeTecnico*” para visualizar que se ha creado con los campos solicitados. Y esta vista nos da un output de **586 registros de transacciones**.

The screenshot shows a database interface with the following details:

Query: `48 • SELECT * FROM Informetecnico;`

Result Grid:

IdTransaccion	Cantidad	NombreUsuario	ApellidoUsuario	IBAN	NombreCompania
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	42.32	Linus	Willis	KW948532754781757886242955643	Nunc Interdum Incorpora
FD89D51B-AE8D-77DC-E450-B8083FBD3187	200.72	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	78.29	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet C
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	335.56	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorpora
FBD7E0D6-BA6B-F5BC-0CA9-E44BB760100C	207.09	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	304.95	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB3B	149.84	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated

Action Output:

Time	Action	Response	Duration / Fetch Time
413 18:20:32	CREATE VIEW InformeTecnico AS SELECT transaction.id AS...	0 row(s) affected	0.013 sec
414 18:22:01	SELECT * FROM Informetecnico	586 row(s) returned	0.010 sec / 0.00024 s...