

Nivel 1

Ejercicio 1. Tu tarea es diseñar y crear una tabla llamada "credit_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva mesa tiene que ser capaz de identificar de manera única cada tarjeta y establecer una relación adecuada con las otras dos mesas ("transaction" y "compañero"). Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credito". Recuerda mostrar el diagrama y realizar una breve descripción de este.

```

4  -- i "establir una relació adequada amb les altres dues taules" ("transaction" i "company"). Després de
5  -- crear la taula serà necessari que ingressis la informació del document denominat
6  -- "dades_introduir_credit".Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.
7
8  * CREATE TABLE credit_card (
9      id VARCHAR(50) PRIMARY KEY, -- NOT NULL UNIQUE
10     iban VARCHAR(34) UNIQUE,
11     pan VARCHAR(19),
12     pin CHAR(6),
13     cvv CHAR(4),
14     expiring_date VARCHAR(10)
15 );
16
17 -- para modificar el formato de fecha a DATE primero

```

Output

#	Time	Action	Message	Duration / Fetch
100105	12:40:24	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amo...	1 row(s) affected	0.000 sec
100106	12:47:34	CREATE TABLE credit_card (id VARCHAR(50) PRIMARY KEY, -- NOT NULL UNIQUE iba...	0 row(s) affected	0.156 sec

Figura 1. Creación de tabla credit_card

Tabla 1. Estándares internacionales de longitudes de datos bancarios.

Campo	Estándar	Longitud / reglas
IBAN	ISO 13616	hasta 34 caracteres
PAN	ISO/IEC 7812	hasta 19 dígitos
IIN/BIN	ISO/IEC 7812	6–8 dígitos
PIN	ISO 9564	4–12 dígitos (general: 4 o 6)
CVV/CVC	PCI DSS + reglas de las marcas	3 (Visa/Mastercard), 4 (Amex)

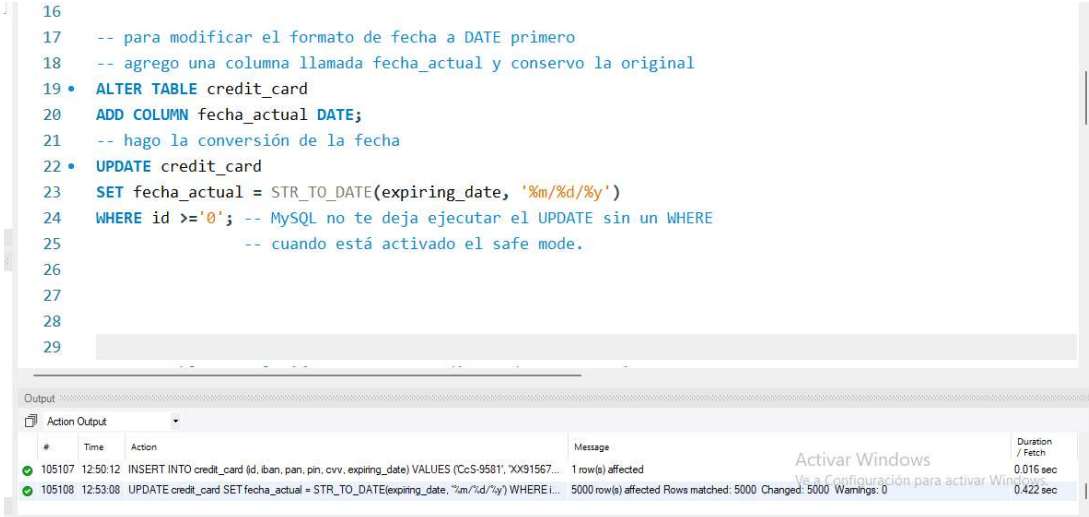
Campo	Estándar	Longitud / reglas
Expiración	PCI DSS / EMV	MM/YY
BIC/SWIFT ISO 9362		8 o 11 caracteres

El código crea la tabla **credit_card** en MySQL definiendo cada campo con su tipo y restricciones: un id como clave primaria, un iban único, y valores como PAN, PIN, CVV y fecha de expiración con longitudes basadas en estándares oficiales. La estructura está preparada para añadir relaciones con las tablas **transaction** y **company**, formando un modelo de datos organizado y consistente.

```

16
17 -- para modificar el formato de fecha a DATE primero
18 -- agrego una columna llamada fecha_actual y conservo la original
19 • ALTER TABLE credit_card
20   ADD COLUMN fecha_actual DATE;
21 -- hago la conversión de la fecha
22 • UPDATE credit_card
23   SET fecha_actual = STR_TO_DATE(expiring_date, '%m/%d/%y')
24   WHERE id >='0'; -- MySQL no te deja ejecutar el UPDATE sin un WHERE
25                   -- cuando está activado el safe mode.
26
27
28
29

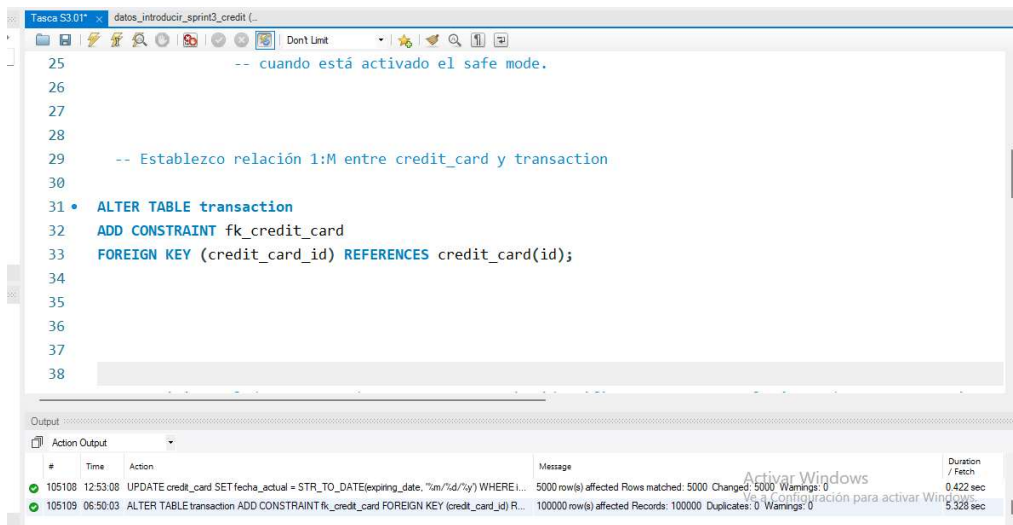
```



#	Time	Action	Message	Duration / Fetch
105107	12:50:12	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('Cc5-9581', 'XX91567...')	1 row(s) affected	0.016 sec
105108	12:53:08	UPDATE credit_card SET fecha_actual = STR_TO_DATE(expiring_date, '%m/%d/%y') WHERE id >='0';	5000 row(s) affected Rows matched: 5000 Changed: 5000 Warnings: 0	0.422 sec

Figura 2. Agregación de columna **fecha_actual** y conversión de fecha al formato **DATE**.

Se agregó columna **fecha_actual** como tipo **DATE** y convirtió la fecha almacenada en texto mediante **STR_TO_DATE**. Para poder ejecutar el **UPDATE** incluyó un **WHERE**, ya que el *safe mode* de MySQL lo exige.



```
25 -- cuando está activado el safe mode.
26
27
28
29 -- Establezco relación 1:M entre credit_card y transaction
30
31 • ALTER TABLE transaction
32   ADD CONSTRAINT fk_credit_card
33   FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
34
35
36
37
38
```

#	Time	Action	Message	Duration / Fetch
105108	12:53:08	UPDATE credit_card SET fecha_actual = STR_TO_DATE(expiring_date, "%m/%d/%y") WHERE I...	5000 row(s) affected Rows matched: 5000 Changed: 5000 Warnings: 0	0.422 sec
105109	06:50:03	ALTER TABLE transaction ADD CONSTRAINT fk_credit_card FOREIGN KEY (credit_card_id) R...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	5.328 sec

Figura 3. Creación de relación entre la tabla **transaction** y **credit_card**

Se estableció la relación entre la tabla **transaction** y **credit_card** agregando una clave foránea mediante ALTER TABLE. Para ello creó la restricción **fk_credit_card**, la cual enlaza el campo **credit_card_id** de **transaction** con el campo **id** de **credit_card**. Con esta acción definió una relación de **uno a muchos**, donde una tarjeta puede tener múltiples transacciones asociadas, garantizando la integridad referencial dentro de la base de datos.

EJERCICIO 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a la tarjeta de crédito con ID CcU-2938. La información que tiene que mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

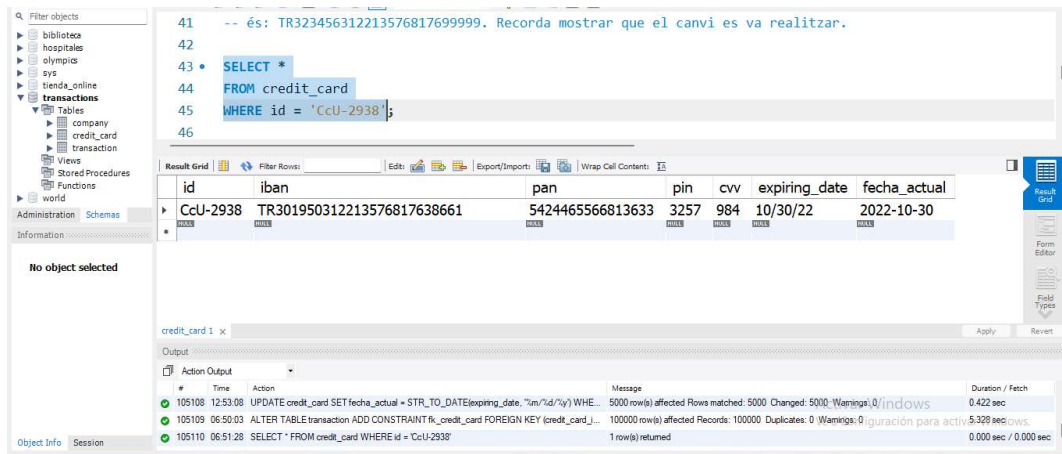


Figura 4. Identificación del error en el IBAN de la tarjeta con ID **CcU-2938**

Detectado el error en el IBAN de la tarjeta con ID **CcU-2938**, figura 4, se procedió a corregirlo mediante una sentencia UPDATE que reemplazó el valor anterior por el nuevo número proporcionado por Recursos Humanos. Después ejecutó un SELECT filtrado por el mismo ID para comprobar que el cambio se aplicó correctamente., ver figura 5. Al visualizar nuevamente el registro en la tabla *credit_card*, pudo confirmar que el IBAN fue actualizado sin afectar el resto de los datos asociados a la tarjeta.

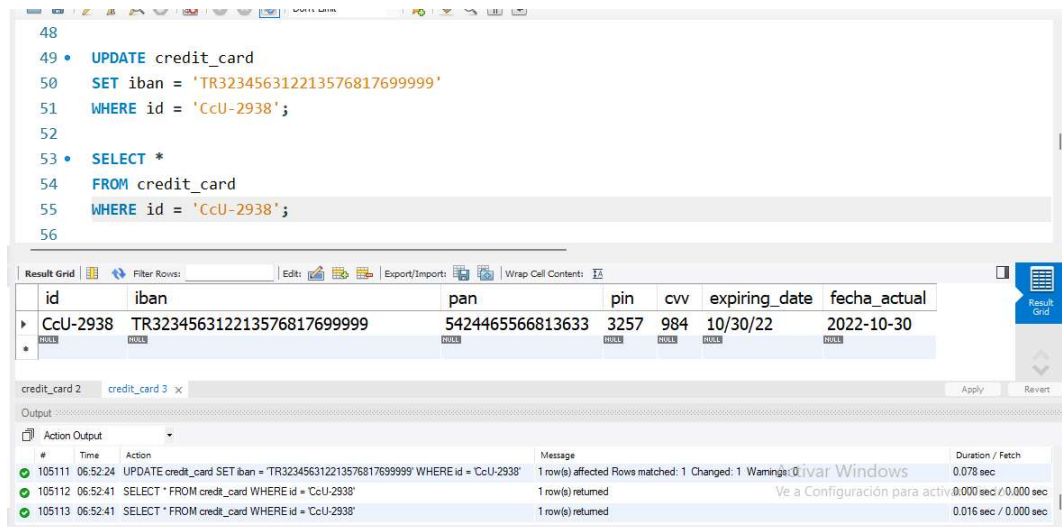


Figura 5 cambio datos ya existentes en registro específico de una tabla *credit_card* y comprobación de cambios..

EJERCICIO 3

En la tabla "transaction" ingresa una nueva transacción con la siguiente información:

IT ACADEMY	S3.01 Fecha entrega: 24-11-25	Responsable: Ana C. Urdaneta F.
-------------------	--	---

Tabla 2. Datos de transacción a ingresar.

<i>id</i>	108B1D1D-5B23-A76C-55EF-C568E49A99DD
<i>credit_card_id</i>	CcU-9999
<i>company_id</i>	b-9999
<i>user_id</i>	9999
<i>lat</i>	829.999
<i>longitude</i>	-117.999
<i>amount</i>	111.11
<i>declined</i>	0


```

64
65 • INSERT INTO company (id)
66   VALUES ('b-9999');
67
68 • INSERT INTO credit_card (id)
69   VALUES ('CcU-9999');
70
71 -- Incerto los datos, coloco NULL en timestamp pq no me proporcionan la fecha y
72 • INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,timestamp ,amount,declined )
73   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999','b-9999',9999,829.999,-117.999,NULL,111.11,0);
74
75
76

```


#	Time	Action	Message	Duration / Fetch
1	06:59:35	INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,timestamp ,amount,declined)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions`tr...	0.016 sec
2	07:05:26	INSERT INTO credit_card (id) VALUES ('CcU-9999')	1 row(s) affected	0.031 sec
3	07:05:39	INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,timestamp ,amount,declined)	1 row(s) affected	0.046 sec

Figura 6. Ingreso de nueva transacción

Se intentó insertar el registro inicialmente, pero el sistema devolvió el error 1452, lo que indicó que todavía no existían los valores correspondientes al *credit_card_id* o *company_id* requeridos por la transacción. Para cumplir con las claves foráneas, primero se insertaron los datos de una tarjeta en la tabla *credit_card* con el id 'CcU-9999' y los de una compañía en la tabla *company* con el id 'b-9999'. Una vez creados estos registros, se añadió la nueva transacción mediante `INSERT INTO transaction`, verificando que *credit_card_id* y *company_id* coincidieran con los valores ya existentes. Al corregir el orden de inserción, se evitó el error de integridad referencial y se pudo registrar

correctamente la operación con todos sus campos, incluyendo coordenadas, monto e indicador de rechazo.

EJERCICIO 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la mesa credit_card. Recuerda mostrar el cambio realizado.

The screenshot shows a database management interface with the following SQL commands executed:

```

81 • ALTER TABLE credit_card
82   DROP COLUMN pan;
83
84 • DESCRIBE credit_card;
  
```

Below the commands, a table displays the structure of the `credit_card` table:

Field	Type	Null	Key	Default	Extra
id	varchar(50)	NO	PRI		
iban	varchar(34)	YES	UNI		
pin	char(6)	YES			
cvv	char(4)	YES			
expiring_date	varchar(10)	YES			
fecha_actual	date	YES			

The bottom section shows the output of the executed commands:

#	Time	Action	Message	Duration / Fetch
3	07:05:39	INSERT INTO transaction(id,credit_card_id,company_id,user_id,lat,longitude,timestamp,amount)	1 row(s) affected	0.046 sec
4	07:07:45	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
5	07:07:45	DESCRIBE credit_card	6 row(s) returned	0.032 sec / 0.000 sec

Figura 7.- Eliminación de la columna PAN

Para atender la solicitud de Recursos Humanos, se procedió a eliminar la columna pan de la tabla `credit_card` utilizando el comando `ALTER TABLE credit_card DROP COLUMN pan;`. Después se ejecutó `DESCRIBE credit_card` para mostrar la estructura actualizada de la tabla y confirmar que la columna había sido removida correctamente, tal y como lo muestra la figura 7. Con esto, dejó la tabla ajustada a los nuevos requerimientos sin afectar el resto de los datos existentes.

NIVEL 2

EJERCICIO 1

Elimina de la Tabla transaction el registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.

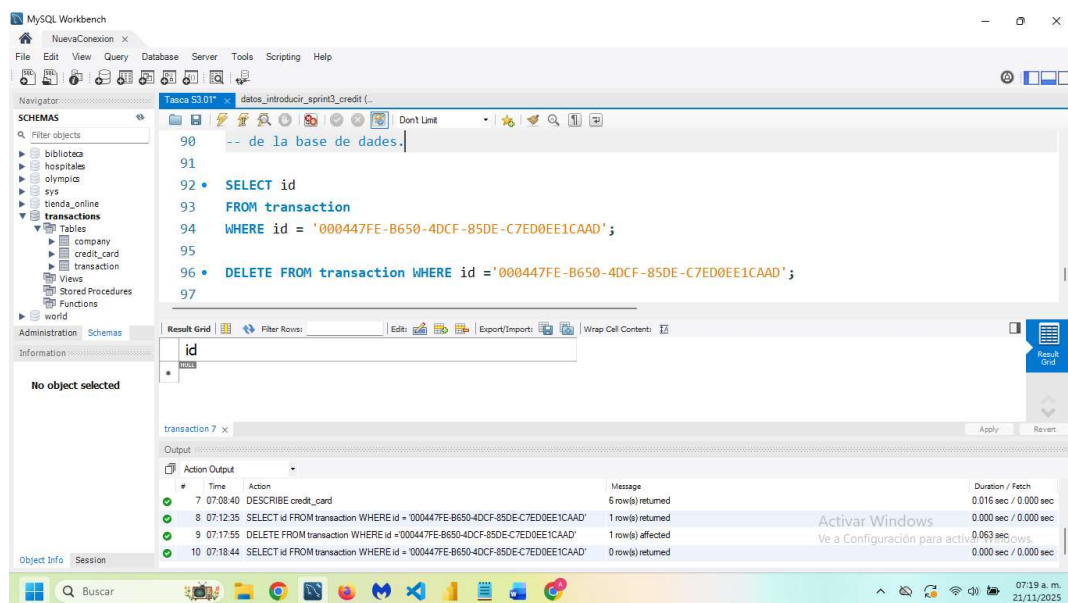


Figura 8. Eliminación de la Tabla transaction registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD

Se procedió a localizar el registro de la tabla transaction donde id era 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD mediante una consulta SELECT, verificando así que el dato existía. Una vez confirmado, ejecutó la sentencia DELETE FROM transaction con el mismo identificador para eliminarlo de la base de datos. Finalmente se volvió a ejecutar un SELECT para comprobar que el registro ya no aparecía, ver figura 8, confirmando que la eliminación se realizó correctamente.

EJERCICIO 2

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas. Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia. Media de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

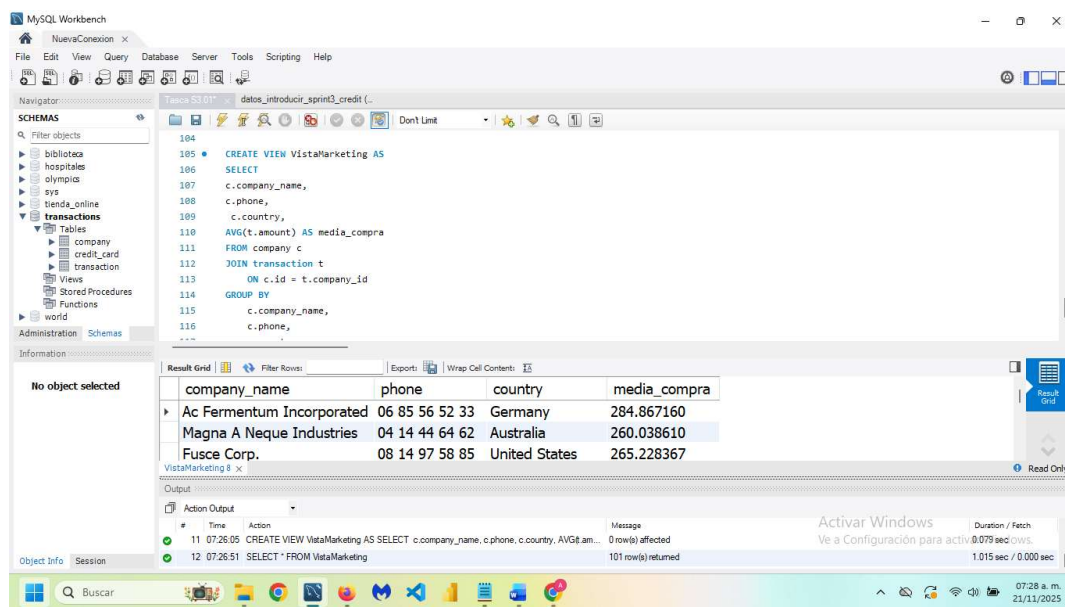


Figura 9. Creación de vista VistaMarketing

En la figura 9 se observa que se creó la vista VistaMarketing para proporcionar al área de marketing información relevante sobre cada compañía y sus transacciones. La vista muestra el nombre de la empresa, su teléfono de contacto, el país de residencia y la media de compra obtenida a partir de sus transacciones, utilizando una combinación de las tablas company y transaction. Después de generar la vista, se ordenó los resultados de mayor a menor según la media de compra, permitiendo que el departamento de marketing visualice de inmediato qué compañías realizan mayores volúmenes de compra y facilitando así sus análisis y estrategias.

EJERCICIO 3

Filtra la vista VistaMarketing para mostrar solo las compañías que tienen su país de residencia en "Germany"

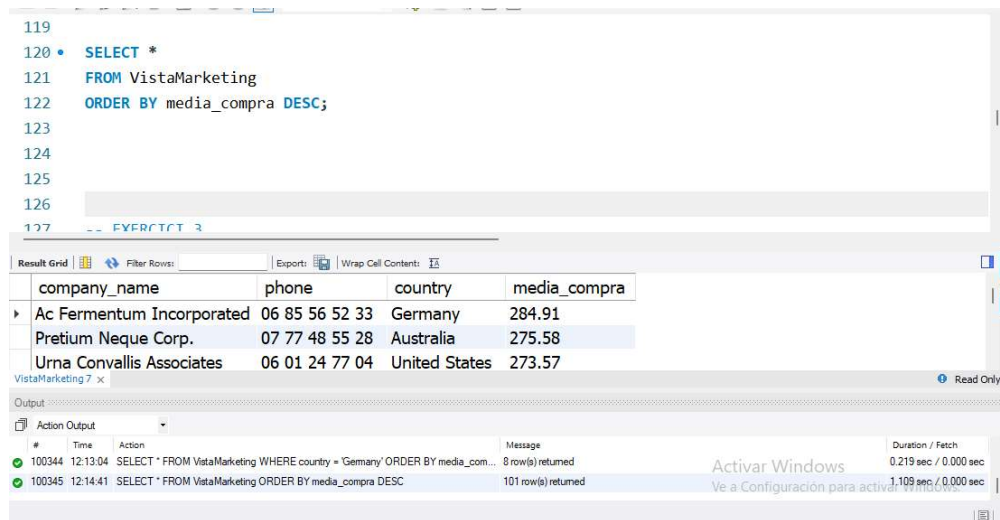


Figura 10 Filtro de la vista VistaMarketing por país "Germany".

Filtró la vista **VistaMarketing** para mostrar exclusivamente las compañías cuyo país de residencia es *Germany*, utilizando una sentencia SELECT con la condición WHERE country = 'Germany'. Luego ordenó los resultados de mayor a menor según la media de compra.

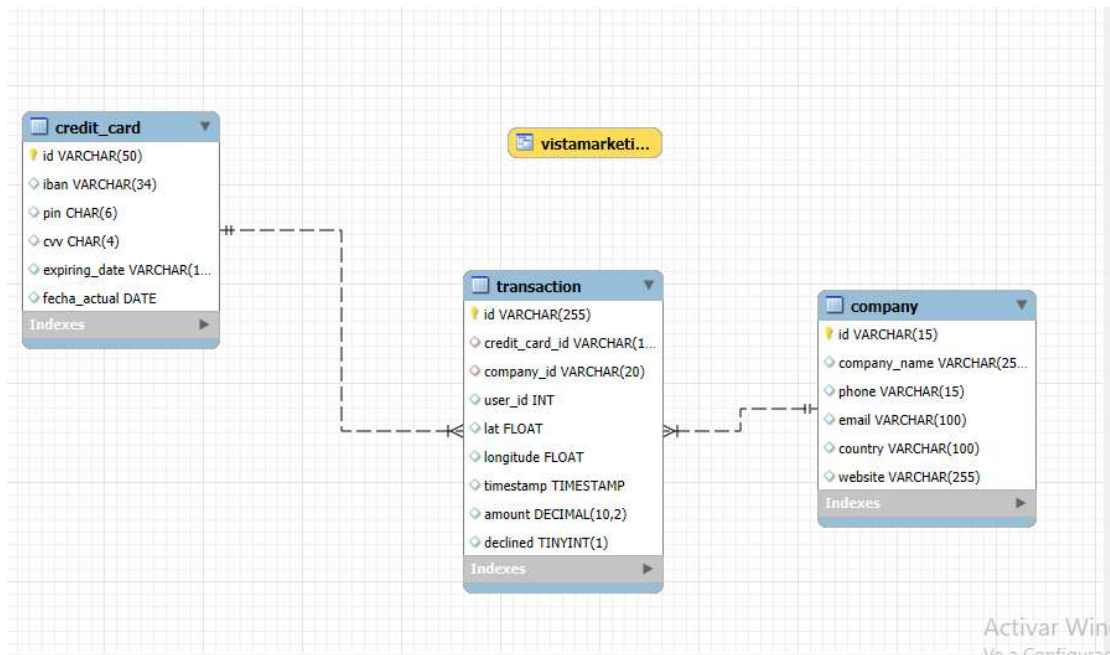


Figura 11. Diagrama resultado del nivel 2

NIVEL 3

Ejercicio 1

La próxima semana tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo **realizó modificaciones en la base de datos**, pero no recuerda como las realizó. Te pide que lo ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:

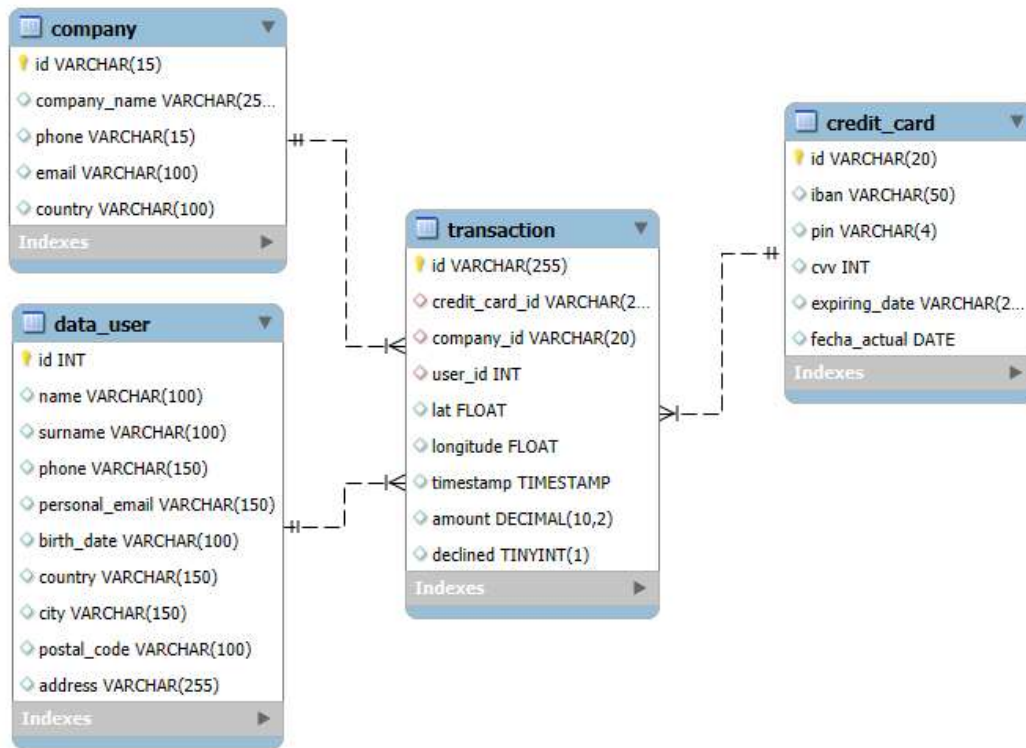


Figura 12. Diagrama ejercicio nivel 3

Para verificar los cambios realizados se verificó el diagrama actual con el diagrama resultados hechos hasta el nivel 2. Evidenciándose la creación de una nueva tabla inicialmente, data_user

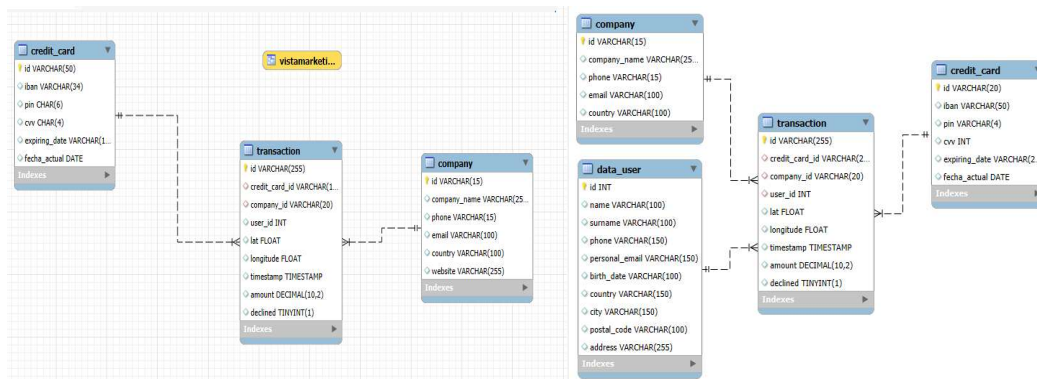


Figura 13. Diagramas antes y después de los cambios.

IT ACADEMY	S3.01 Fecha entrega: 24-11-25	Responsable: Ana C. Urdaneta F.
-------------------	--	---

Se evidenciaron los siguientes cambios entre los diagramas, ver tabla 3

Tabla 3. Cambios realizados al diagrama 1 evidenciados en diagrama 2

Elemento	Diagrama 1	Diagrama 2	Cambio realizado
Nueva tabla agregada	No existe tabla data_user	Aparece tabla data_user	Se agregó una nueva entidad llamada data_user con información personal del usuario.
Campos de la tabla data_user	—	id, name, surname, phone, personal_email, birth_date, country, postal_code, address	Nueva estructura completa de datos del usuario.
Relación transaction → data_user	transaction.user_id existe pero no tenía referencia visual	transaction.user_id se conecta claramente a data_user.id	Se formalizó la relación entre transacciones y usuarios.
Tabla credit_card	id (VARCHAR50), iban (VARCHAR24), pin (CHAR5), cvv (CHAR5), expiring_date VARCHAR/fecha_actual DATE	id VARCHAR(20), iban VARCHAR(50), pin VARCHAR(4), cvv INT, expiring_date VARCHAR(20), fecha_actual DATE	Cambios en tipos de datos y longitudes (reducción en id, ampliación de iban, ajuste de pin y cvv).
Tabla company	Campos iguales pero sin orden estandarizado	Igual estructura pero organizada en nuevo orden No aparece el campo website	Reorganización de campos , sin cambios de estructura. Se borra el campo website
Relación transaction → company	Existía vínculo	Se mantiene vínculo	Se mantiene la relación
Tipos de datos en transaction	id VARCHAR(255), lat FLOAT, long FLOAT, timestamp TIMESTAMP, amount DECIMAL(10,2), declined TINYINT(1)	Igual, sin cambios	Sin modificaciones.
Relación transaction → credit_card	credit_card_id VARCHAR(?) apuntaba a credit_card.id	Se mantiene igual	Sin cambios.

Se comenzó creando la data data_user, sin embargo, inicialmente no se creó con ese nombre, sino con la estructura proporcionada en los archivos adjuntos del ejercicio para poder introducir los valores proporcionados y luego hacer los cambios sobre la tabla user, ver figura 14.

```

136
137 -- Como los datos proporcionados para introducir no coinciden con el nombre de
138 -- tabla del diagrama proporcionado, creé la tabla con la misma estructura de la original y
139 -- luego se realizarán los cambios para ajustarlo al diagrama
140 -- Creación tabla data_user
141 CREATE TABLE IF NOT EXISTS user (
142     id int PRIMARY KEY,
143     name VARCHAR(100),
144     surname VARCHAR(100),
145     phone VARCHAR(150),
146     email VARCHAR(150),
147     birth_date VARCHAR(100),
148     country VARCHAR(150),
149     city VARCHAR(150),
150     postal_code VARCHAR(100),
151     address VARCHAR(255)
152 );
153 SHOW CREATE TABLE transaction;
154
Output
Action Output
# Time Action Message Duration / Fetch
4503 06:25:24 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (1, 'John', 'Doe', '1234567890', 'john.doe@example.com', '1990-01-01', 'USA', 'New York', '10001', '123 Main St') 1 row(s) affected 0.016 sec
4504 06:25:24 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (2, 'Jane', 'Smith', '9876543210', 'jane.smith@example.com', '1992-05-15', 'Canada', 'Toronto', 'M5H 2N2', '456 Queen St W') 1 row(s) affected 0.015 sec
4505 06:25:24 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (3, 'Mike', 'Johnson', '5555555555', 'mike.johnson@example.com', '1988-11-20', 'Australia', 'Sydney', '2000', '789 George St') 1 row(s) affected 0.000 sec
4506 06:25:24 INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (4, 'Emily', 'Brown', '4444444444', 'emily.brown@example.com', '1995-03-10', 'Brazil', 'Rio de Janeiro', '20040-001', '321 Av. Paulista') 1 row(s) affected 0.016 sec

```

Figura 14. Creación de tabla user

En la figura 15 podemos observar el cambio del nombre de la tabla user a data_user y de la columna email a personal_email

```

155
156 RENAME TABLE user TO data_user;
157 ALTER TABLE data_user
158 RENAME COLUMN email TO personal_email;
159
160 -- transaction = data user (muchas transacciones por usuario)

```

Output

#	Time	Action	Message	Duration / Fetch
5001	06:25:30	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (1, 'John', 'Doe', '1234567890', 'john.doe@example.com', '1990-01-01', 'USA', 'New York', '10001', '123 Main St')	1 row(s) affected	0.031 sec
5002	06:25:30	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (2, 'Jane', 'Smith', '9876543210', 'jane.smith@example.com', '1992-05-15', 'Canada', 'Toronto', 'M5H 2N2', '456 Queen St W')	1 row(s) affected	0.016 sec
5003	06:34:39	RENAME TABLE user TO data_user	0 row(s) affected	0.110 sec
5004	06:36:08	ALTER TABLE data_user RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.110 sec

Figura 15. Cambio de nombre de tabla user a data_user y de la columna email a personal_email

En la figura 16 se puede observar la creación de la relación transaction → data_user (muchas transacciones por usuario).

```

213
214 -- transaction → data_user (muchas transacciones por usuario)
215
216 • ALTER TABLE transaction
217   ADD CONSTRAINT fk_transaction_user
218   FOREIGN KEY (user_id)
219   REFERENCES data_user(id)
220   ON DELETE CASCADE
221   ON UPDATE CASCADE;
222
223
224
225
226 -- EJERCICI 2

```

Output

#	Time	Action	Message	Duration / Fetch
3	05:45:58	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(20) PRIMARY KEY, ib...	0 row(s) affected, 1 warning(s): 1050 Table 'credit_card' already exists	0.031 sec
4	05:47:36	CREATE TABLE IF NOT EXISTS transaction (id VARCHAR(255) PRIMARY KEY, ...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be remo...	0.016 sec
5	05:49:12	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card FOREIGN K...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.422 sec
6	05:50:12	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_company FOREIGN KEY...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.297 sec

Figura 16. Creación de relación *transaction* → *data_user* (muchas transacciones por usuario)

El siguiente paso fue hacer los cambios en la tabla *credit_card* estos fueron de tipo de datos y longitudes, ver figura 17.

```

173
174 -- 2. Ajustes de tipos de datos y longitudes
175 • ALTER TABLE credit_card
176   MODIFY id VARCHAR(20),
177   MODIFY iban VARCHAR(50),
178   MODIFY pin VARCHAR(4),
179   MODIFY cvv INT,
180   MODIFY expiring_date VARCHAR(20),
181   MODIFY fecha_actual DATE;
182
183 • SELECT DISTINCT user_id
184 FROM transaction

```

Output

#	Time	Action	Message	Duration / Fetch
1	07:01:38	ALTER TABLE credit_card MODIFY id VARCHAR(...	1 row(s) affected Records: 1 Duplicates: 0 Warnings: 0	0.156 sec

Figura 17. Modificación de longitudes y tipo de datos

Se evaluó la integridad referencial garantizar que las relaciones entre tablas sean **coherentes y válidas**, debido a que en ejercicios anteriores se realizaron modificaciones y se necesita saber si hay datos huérfanos, ver código en figura 18. En este caso se encontró que en la tabla *data_user* no está el id =9999 y se procede a insertar, ya que la tabla padre (*data_user*) debe tener los mismos id que en la tabla hija (*transaction*) y así crear la FK.

Es importante destacar que inicialmente se había presentado la situación de que se pudo crear la FK entre *data_user* y *transaction* sin hacer la validación referencial lo que no debe suceder si están activas las FK. Es por este sentido se borra la FK creada inicialmente y

se activan las restricciones de las Fk y se realiza correctamente la adición de la FK correctamente, ver figura 19.

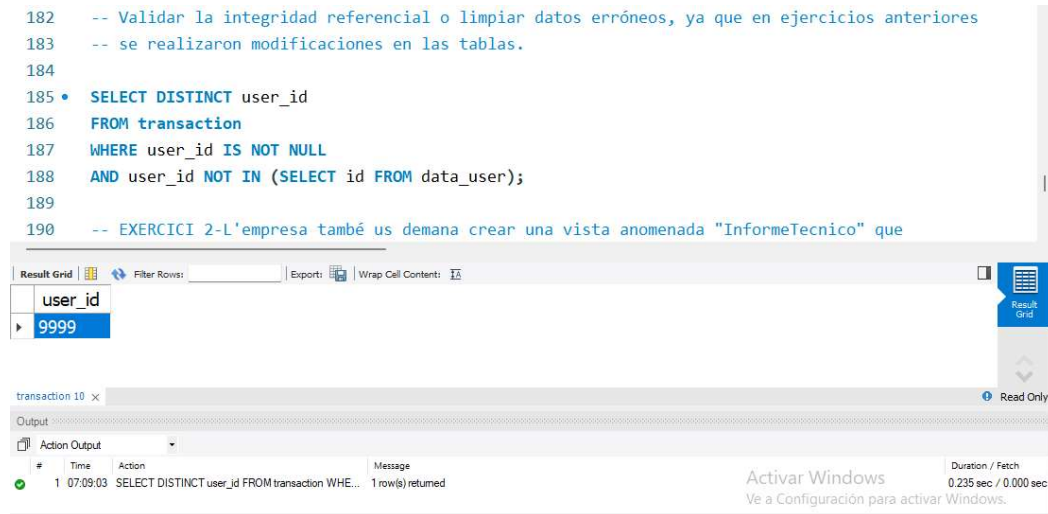


Figura 18. Evaluación de integridad referencial

Evidentemente no me debió dejar crear la FK y es por eso que se evidencia que estaban desactivadas las restricciones de foreign key ya que obliga a que cada transaction.user_id exista en data_user.id y por tanto daría error.

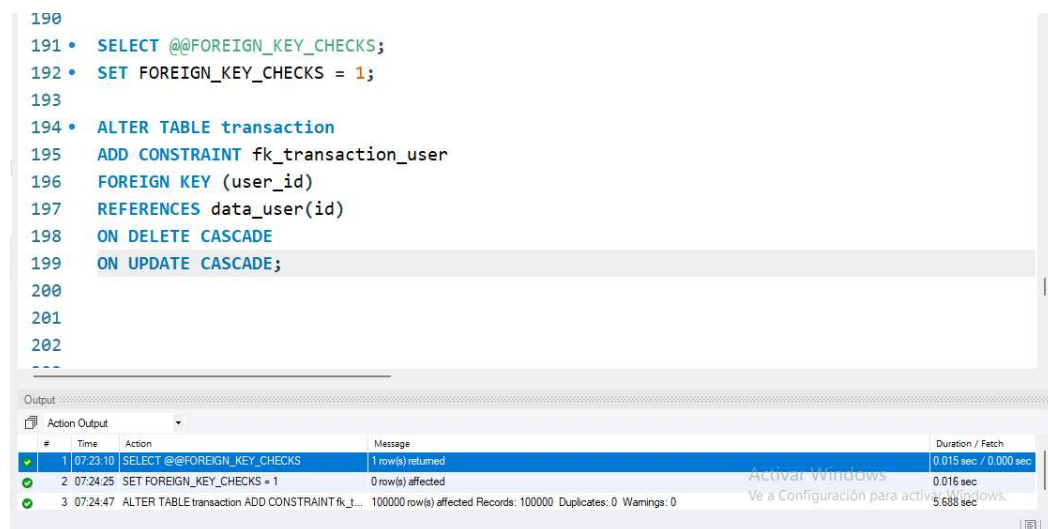


Figura 19. Activación de las restricciones de la FK y creación de la FK entre transacción y data_user



Figura 20. Drop de la columna website

Adicionalmente en los cambios se puede observar que en la tabla company no existe el campo website



Figura 21. Modificación de longitud de dato en la tabla data_user del campo personal_email

En la tabla 21 podemos observar el cambio de la longitud de dato de personal_email

EJERCICIO 2.

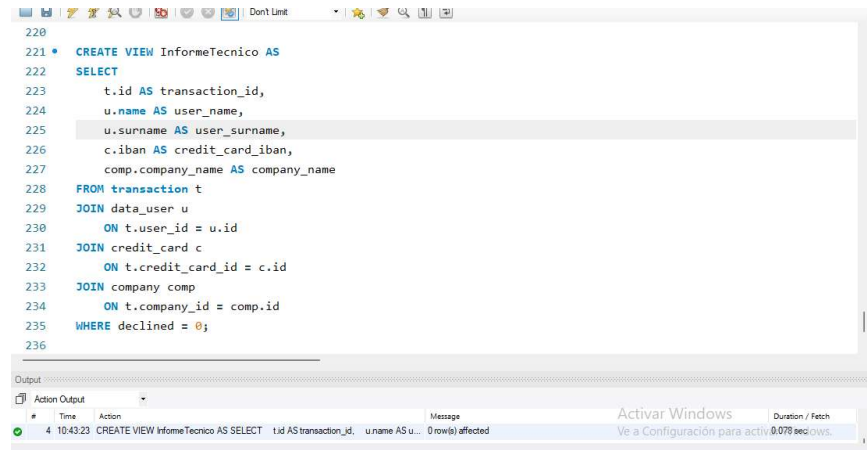
La empresa también os pide crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/aria
- Apellido del usuario/aria
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.

Aseguraos de incluir información relevante de las tablas que conoceréis y utilizáis alias para cambiar de nombre columnas según haga falta.

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

Se creó la vista y se llamó para mostrar la tabla con los resultados. Ver figura 22 y 23.



```

220
221 • CREATE VIEW InformeTecnico AS
222 SELECT
223     t.id AS transaction_id,
224     u.name AS user_name,
225     u.surname AS user_surname,
226     c.iban AS credit_card_iban,
227     comp.company_name AS company_name
228 FROM transaction t
229 JOIN data_user u
230     ON t.user_id = u.id
231 JOIN credit_card c
232     ON t.credit_card_id = c.id
233 JOIN company comp
234     ON t.company_id = comp.id
235 WHERE declined = 0;
236

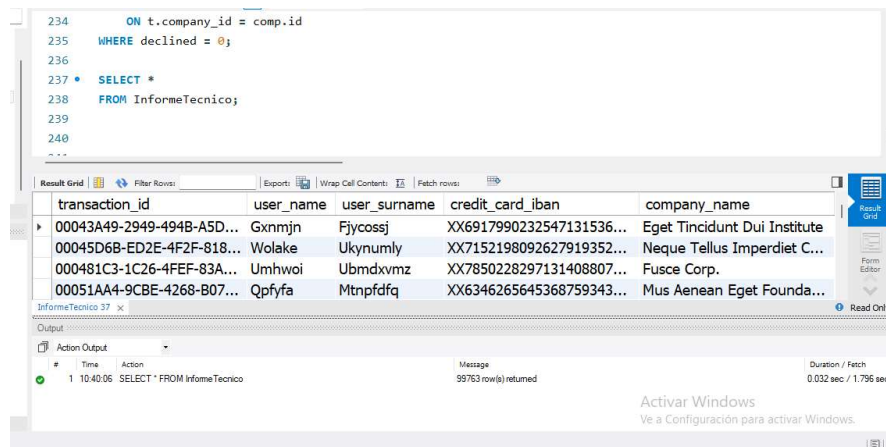
```

Output

#	Time	Action	Message	Duration / Fetch
4	10:43:23	CREATE VIEW InformeTecnico AS SELECT	Lid AS transaction_id, u name AS u...	0 row(s) affected

Activar Windows
Ve a Configuración para activar Windows.

Figura 22. Creación de la vista "InformeTecnico"



```

234     ON t.company_id = comp.id
235 WHERE declined = 0;
236
237 • SELECT *
238 FROM InformeTecnico;
239
240

```

Result Grid

transaction_id	user_name	user_surname	credit_card_iban	company_name
00043A49-2949-494B-A5D...	Gxnmjn	Fjycossj	XX6917990232547131536...	Eget Tincidunt Dui Institute
00045D6B-ED2E-4F2F-818...	Wolake	Ukynumly	XX7152198092627919352...	Neque Tellus Imperdiet C...
000481C3-1C26-4FEF-83A...	Umhwoi	Ubmdxvmz	XX7850228297131408807...	Fusce Corp.
00051AA4-9CBE-4268-B07...	Qpfyfa	Mtnpfdfq	XX6346265645368759343...	Mus Aenean Eget Founda...

Output

#	Time	Action	Message	Duration / Fetch
1	10:40:06	SELECT * FROM InformeTecnico	99763 row(s) returned	0.032 sec / 1.796 sec

Activar Windows
Ve a Configuración para activar Windows.

Figura 23. Muestra de "InformeTecnico"

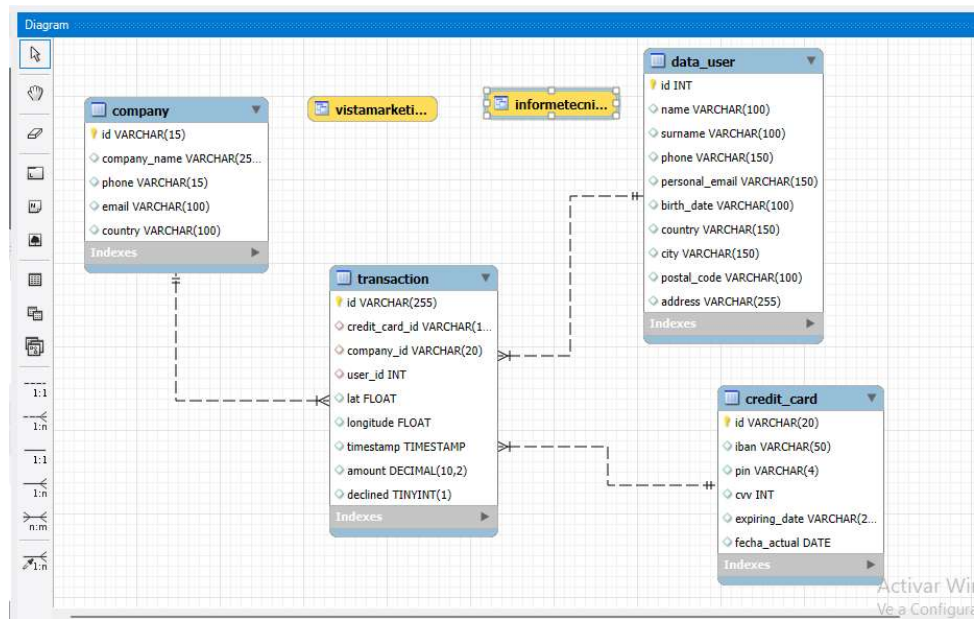


Figura 24. Diagrama Nivel 3