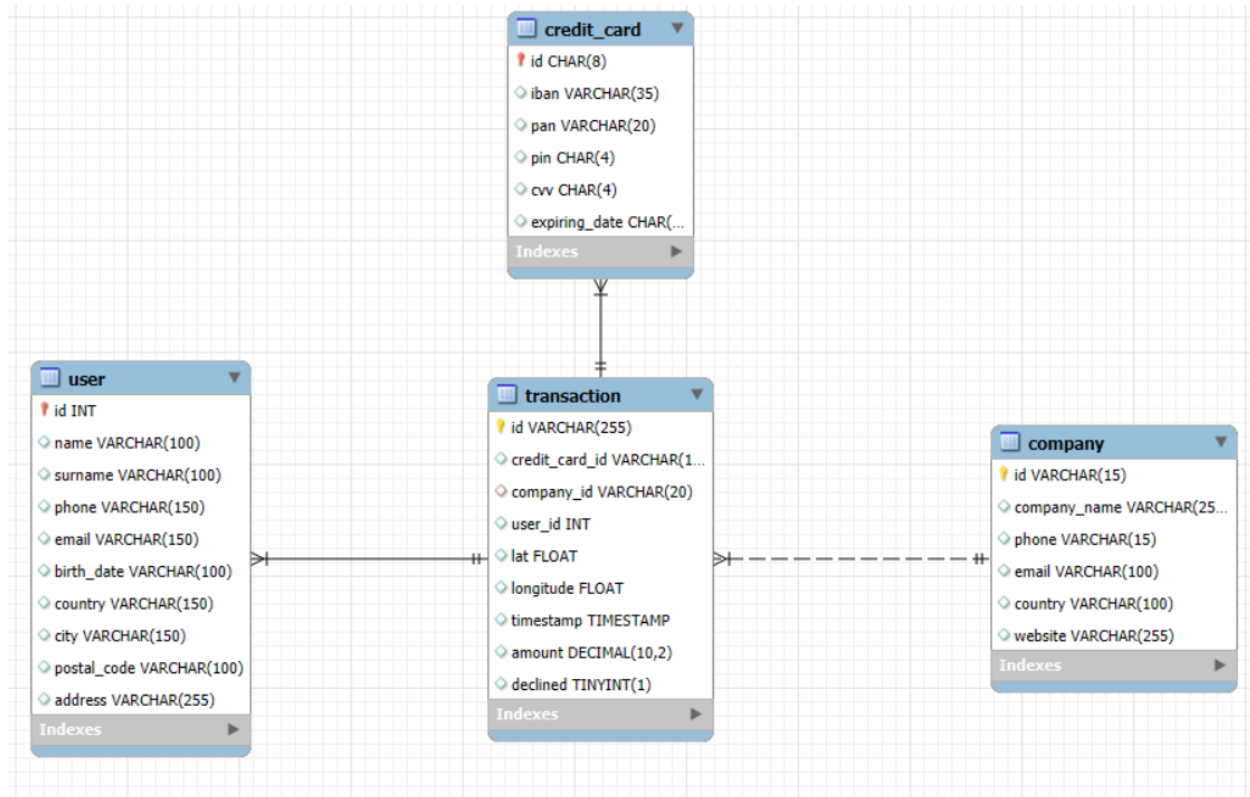


Nivel 1 - Ejercicio 1

Cree la tabla `credit_card` y defina el tipo de datos para cada campo. Además importe los datos para completar la tabla y vinculela con la tabla `transaction`.



Nivel 1 - Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.

```

13 # Nivel 1 - Ejercicio 2
14
15 • UPDATE credit_card
16   SET iban = 'R323456312213576817699999'
17   WHERE id = 'CcU-2938';
18
19 • SELECT *
20   FROM credit_card
21   WHERE id = 'CcU-2938';
  
```

Result Grid						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
*	NULL	NULL	NULL	NULL	NULL	NULL

credit_card 2 x

Output

Action Output

#	Time	Action	Message
✓ 657	11:50:20	UPDATE credit_card SET iban = 'R3...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Nivel 1 - Ejercicio 3

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:

```
23 #Nivel 1 - Ejercicio 3
24
25 • SET FOREIGN_KEY_CHECKS=0;
26
27 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
28 VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', 829.999, -117.999, 111.11, 0);
29
30 • SELECT *
31 FROM transaction
32 WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
```

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	NULL	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 5 x

Output

Action Output

#	Time	Action	Message
663	12:08:23	INSERT INTO transaction (id, credit...	1 row(s) affected
664	12:08:30	SELECT * FROM transaction WHER...	1 row(s) returned

Nivel 1 - Ejercicio 4

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

```
35 #Nivel 1 - Ejercicio 4
36
37 • ALTER table credit_card
38 DROP COLUMN pan;
39
40 • SELECT *
41 FROM credit_card
42 LIMIT 1;
```

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL

credit_card 15 x

Output

Action Output

#	Time	Action	Message
962	12:33:36	ALTER table credit_card DROP COL...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
963	12:33:43	SELECT * FROM credit_card LIMIT 10	10 row(s) returned

Nivel 2 - Ejercicio 1

Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.

```
44 #Nivel 2 - Ejercicio 1
45
46 • DELETE FROM transaction
47 WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
48
49 • SELECT *
50 FROM transaction
51 WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02';
52
53
```

Result Grid									
Filter Rows: <input type="text"/>									
Edit: Export/Import: Wrap Cell Content:									
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 16 x

Output

Action Output

#	Time	Action	Message
✓ 966	12:38:34	DELETE FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected
✓ 967	12:39:18	SELECT * FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, ...	0 row(s) returned

Nivel 2 - Ejercicio 2

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 promedio de compra.

```
68 #Nivel 2 - Ejercicio 2
69
70 • CREATE VIEW `vistamarketing` AS
71 SELECT company.company_name, company.phone, company.country, AVG(transaction.amount) AS AVGtransaction
72 FROM company
73 LEFT JOIN transaction ON company.id=transaction.company_id
74 WHERE declined = 0
75 GROUP BY company.company_name, company.phone, company.country
76 ORDER BY AVGtransaction DESC;
77
78 • SELECT * FROM transactions.vistamarketing;
```

Result Grid				
Filter Rows: <input type="text"/> Export: Wrap Cell Content:				
	company_name	phone	country	AVGtransaction
▶	Eget Ipsum Ltd	03 67 44 56 72	United States	481.860000
	Sed Id Limited	07 28 18 18 13	United States	477.510000
	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.100000
	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.830000
	Non Magna LLC	06 71 73 13 17	United Kingdom	458.740000
	Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	451.290000
	Erat LLP	03 18 88 77 79	Netherlands	448.440000
	Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.110000
	Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.160000
	Pede Cum Ltd	07 62 26 48 38	Norway	442.320000
	Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.400000
	Mauris Tincidunt	05 20 60 26 07	Sweden	427.710000

vistamarketing 5 x

Output

Action Output

#	Time	Message
✓ 37	11:38:46	100 row(s) returned

Nivel 2 - Ejercicio 3

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

```
80      #Nivel 2 - Ejercicio 3
81
82      SELECT *
83      FROM transactions.vistamarketing
84      WHERE country='Germany';
85
```

Result Grid

Export:

Wrap Cell Content:

	company_name	phone	country	AVGtransaction
▶	Ac Industries	09 34 65 40 60	Germany	396.150000
	Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
	Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
	Aliquam PC	01 45 73 52 16	Germany	280.340000
	Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.947692
	Convallis In Incorporated	06 66 57 29 50	Germany	60.990000
	Augue Foundation	06 88 43 15 63	Germany	15.050000

vistamarketing 6

×

Output

Action Output

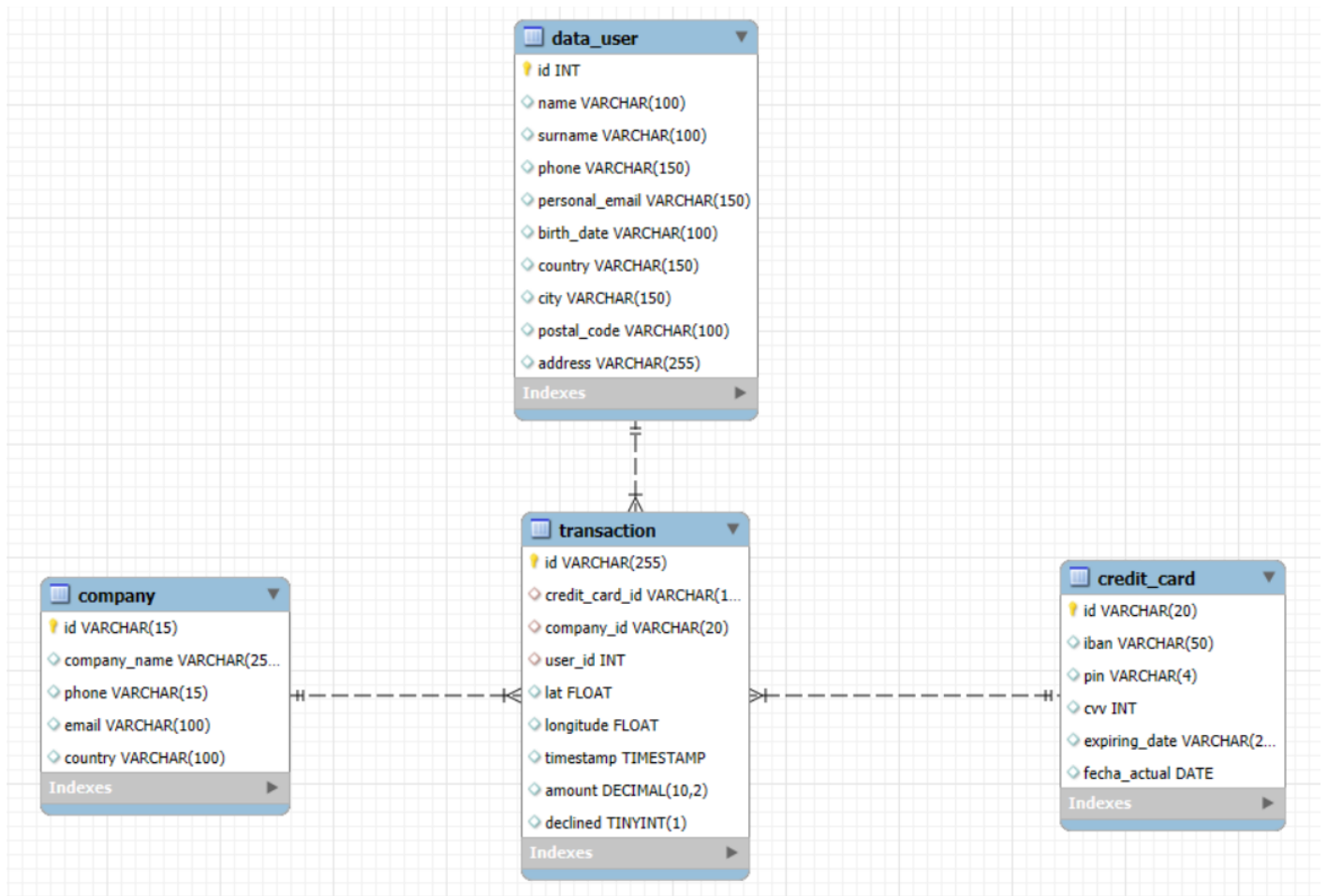
	#	Time	Message
✓	38	11:38:50	100 row(s) returned
✓	39	11:40:46	8 row(s) returned

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 cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama: ...

```
64 #Nivel 3 - Ejercicio 1
65 • ALTER TABLE company
66 DROP COLUMN website;
67 • ALTER TABLE user
68 RENAME COLUMN email TO personal_email;
69 • ALTER TABLE user
70 RENAME data_user;
71 • SET FOREIGN_KEY_CHECKS = 0;
72 • ALTER TABLE credit_card
73 MODIFY COLUMN id VARCHAR(20);
74 • ALTER TABLE credit_card
75 MODIFY COLUMN iban VARCHAR(50);
76 • ALTER TABLE credit_card
77 MODIFY COLUMN pin VARCHAR(4);
78 • ALTER TABLE credit_card
79 MODIFY COLUMN cvv INT;
80 • ALTER TABLE credit_card
81 MODIFY COLUMN expiring_date VARCHAR(20);
82 • ALTER TABLE credit_card
83 ADD COLUMN fecha_actual DATE;

92 • SET FOREIGN_KEY_CHECKS=0;
93 • ALTER TABLE credit_card
94 DROP FOREIGN KEY credit_card_ibfk_1;
95 • ALTER TABLE data_user
96 DROP FOREIGN KEY data_user_ibfk_1;
97 • ALTER TABLE transaction
98 DROP FOREIGN KEY transaction_ibfk_1;
99
100 • ALTER TABLE transaction
101 ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
102 • ALTER TABLE transaction
103 ADD FOREIGN KEY (user_id) REFERENCES data_user(id);
104 • ALTER TABLE transaction
105 ADD FOREIGN KEY (company_id) REFERENCES company(id);
```



Nivel 3 - Ejercicio 2

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información: ID de la transacción, Nombre del usuario/a, Apellido del usuario/a, IBAN de la tarjeta de crédito usada, Nombre de la compañía de la transacción realizada. Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario. Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

```
107 #Nivel 3 - Ejercicio 2
108
109 • CREATE VIEW `informetecnico` AS
110 SELECT transaction.id AS TransactionID, user.name AS UserName, user.surname AS UserSurname,
111 credit_card.iban AS Creditcardiban, company.company_name AS Companyname
112 FROM user
113 LEFT JOIN transaction ON user.id=transaction.user_id
114 RIGHT JOIN company ON transaction.company_id=company.id
115 RIGHT JOIN credit_card ON transaction.credit_card_id=credit_card.id;
116
117 • SELECT *
118 FROM transactions.informetecnico
119 ORDER BY TransactionID DESC;
```

Result Grid					
Filter Rows:		Export:		Wrap Cell Content:	
TransactionID	UserName	UserSurname	Creditcardiban	Companyname	
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries	
FE809ED4-2DB6-55AC-C915-929516E46468	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated	
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated	
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC	
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.	
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated	
FBD7E0D6-BA6B-F5BC-OCA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.	
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP	

informetecnico 7 x

Output

Action Output

#	Time	Message
39	11:40:46	8 row(s) returned
40	11:45:53	586 row(s) returned