

Nivel 1

Ejercicio 1

A partir de los documentos adjuntos (estructura_datos y datos_introducir), importa las dos tablas. Muestra las principales características del esquema creado y explica las diferentes tablas y variables que existen. Asegúrate de incluir un diagrama que ilustre la relación entre las distintas tablas y variables.

Las tablas creadas con los documentos adjuntos son 'transaction' y 'company'. Hay una relación de 1 a muchos entre 'company' y 'transaction' ya que una compañía puede hacer muchas transacciones, pero cada transacción es de una sola compañía.

La primary key de 'company' es **id** que es el id de cada compañía y de 'transaction' es **id** que son los id de cada transacción, es decir son los valores únicos de cada tabla. La relación se establece entre la foreign key **company_id** de la tabla 'transaction' y la primary key **id** de la tabla 'company'.

La tabla 'company' contiene información de las compañías y está formada por las columnas: **id**, **company_name**, **phone**, **email**, **country**, **website**. todas las columnas son de tipo VARCHAR es decir guardan cadenas de texto que pueden incluir números, símbolos o letras, con diferentes longitudes.

La tabla 'transaction' contiene las transacciones realizadas por las compañías con tarjeta de crédito y está formada por las columnas: **id**, **credit_card_id**, **company_id**, **user_id**, **lat**, **longitude**, **timestamp**, **amount**, **declined**

De la tabla 'transaction' las columnas contienen diferentes tipos de datos según el tipo de información:

id, **credit_card_id** y **company_id** son de tipo VARCHAR.

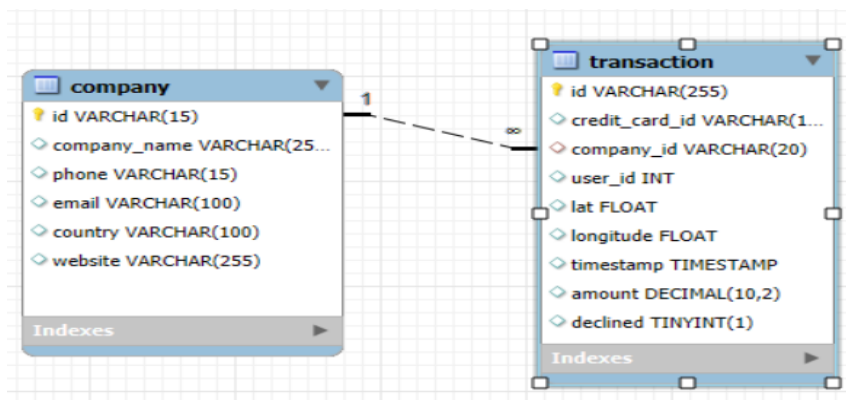
la columna **user_id** es de tipo int, que guarda números enteros.

lat y **longitude** son de tipo float ya que no son decimales exactos.

timestamp guarda fecha y hora de la transacción.

amount es de tipo decimal, tiene el monto de las transacciones, que no son números enteros.

declined es de tipo tinyint(1) para representar valores booleanos donde 1 representa rechazada y 0 aprobada.



Ejercicio 2

Utilizando JOIN realizarás las siguientes consultas:

- Listado de los países que están generando ventas.

The screenshot shows a SQL IDE window titled 'ejemplos codigos'. The query editor contains the following SQL code:

```
5
6 # Listado de los países que están generando ventas.
7 • SELECT DISTINCT c.country ventas_por_paises
8 FROM company c
9 INNER JOIN transaction t ON c.id = t.company_id
10 WHERE t.declined = 0;
```

Below the query editor, the 'Result Grid' shows the results of the query:

ventas_por_paises
Netherlands
Sweden
Ireland
United States
Belgium
Canada
Germany

The 'Output' pane shows the execution details:

#	Time	Action	Message
1	11:19:11	SELECT DISTINCT c.country ventas_por_paises FROM company c INNER JOIN transaction t ON c.id = t.com...	15 row(s) returned

Nota: Se unen las tablas company y transaction mediante un INNER JOIN que permite buscar los datos que coincidan de las 2 tablas y se filtran las transacciones aprobadas.

- Desde cuántos países se generan las ventas.

The screenshot shows a SQL IDE window titled 'ejemplos codigos'. The query editor contains the following SQL code:

```
11
12 # Desde cuántos países se generan las ventas.
13 • SELECT COUNT(DISTINCT c.country) cantidad_paises
14 FROM company c
15 INNER JOIN transaction t ON c.id = t.company_id
16 WHERE declined = 0;
17
```

Below the query editor, the 'Result Grid' shows the results of the query:

cantidad_paises
15

The 'Output' pane shows the execution details:

#	Time	Action	Message
1	11:36:53	SELECT COUNT(DISTINCT c.country) cantidad_paises FROM company c INNER JOIN transaction t ON c.id = ...	1 row(s) returned

Nota: se cuentan todos los países que cumplen las condiciones anteriores solo para saber cuántos son.

- Identifica a la compañía con la mayor media de ventas.

sprint 2 ejemplos codigos

```

18 # Identifica a la compañía con la mayor media de ventas.
19 • SELECT c.company_name compañía, ROUND(AVG(t.amount),2) media_ventas
20 FROM company c
21 INNER JOIN transaction t ON c.id = t.company_id
22 WHERE declined = 0
23 GROUP BY c.id
24 ORDER BY media_ventas DESC
25 LIMIT 1 ;
26

```

Result Grid

compañía	media_ventas
Ac Fermentum Incorporated	284.91

Result 47

Output

Action Output

#	Time	Action	Message
1	11:33:16	SELECT c.company_name compañía, ROUND(AVG(t.amount),2) media_ventas FROM company c INNER JOI...	1 row(s) returned

Nota: se limita a 1 ya que solo se quiere la compañía que tiene la mayor media, pero primero se realiza el calculo de las medias de ventas con la función de AVG para obtener los valores y luego se ordena de manera DESC para mostrar el valor más alto primero.

Ejercicio 3

Utilizando sólo subconsultas (sin utilizar JOIN):

- Muestra todas las transacciones realizadas por empresas de Alemania.

```

31 • SELECT t.*
32 FROM transaction t
33 WHERE declined = 0 AND EXISTS (
34     SELECT c.country
35     FROM company c
36     WHERE c.id = t.company_id
37     AND c.country = 'Germany');
38

```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
00138D3B-206D-4C03-94B7-63A2676EB9B4	CcS-4899	b-2222	318	41.3781	12.447	2020-03-25 10:43:43	426.36	0
0013C1B6-3B84-4D6C-8154-E2B3FEBCA8E9	CcS-5070	b-2222	489	41.3814	2.18176	2020-12-17 18:15:37	316.90	0
00201A11-2E62-44C4-941D-198FC8D877F0	CdU-3512	b-2222	193	55.5704	-3.65129	2021-01-22 23:44:27	453.04	0
00235618-0A5C-4D49-9DCB-83A9405D8923	CcS-8137	b-2222	3556	59.8421	18.729	2020-09-09 15:43:19	263.14	0
005A5A7B-1F1A-4B6C-9B15-1625A78C9C38	CcS-8998	b-2222	4417	41.1591	-8.63905	2024-05-15 09:10:11	442.01	0
00687139-48B2-4FFA-8E73-B20376F04AB4	CcS-4870	b-2222	289	51.1966	10.4669	2019-03-09 19:37:49	524.84	0

transaction 87

Output

Action Output

#	Time	Action	Message
1	13:46:40	SELECT t.* FROM transaction t WHERE declined = 0 AND EXISTS (SELECT c.country FROM company c W...	13269 row(s) returned

Nota: Al no poder utilizar JOIN se realiza una subconsulta en el WHERE donde se utiliza EXISTS para verificar si hay una coincidencia con la condición, se hace una especie de JOIN dentro de esta condición para poder añadir la tabla company y verificar de esta tabla que las transacciones sean de Alemania.

- Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.

```

37
38 # Lista las empresas que han realizado transacciones por un amount superior a la media de todas las transacciones.
39 • SELECT c.company_name
40 FROM company c
41 WHERE id IN (
42     SELECT t.company_id
43     FROM transaction t
44     WHERE t.declined = 0 AND t.amount > (
45         SELECT AVG(t2.amount)
46         FROM transaction t2)
47 );

```

Result Grid

company_name
Ac Fermentum Incorporated
Magna A Neque Industries
Fusce Corp.
Conwallis In Incorporated
Ante Iaculis Nec Foundation
Donec Ltd

company 50 x

Output

Action Output

#	Time	Action	Message
1	11:42:52	SELECT c.company_name FROM company c WHERE id IN (SELECT t.company_id FROM transaction t ...	100 row(s) returned

Nota: Se utilizan subconsultas anidadas, primero se calcula la media de todas las transacciones, luego se escogen las compañías que tienen amounts superiores a esta media y por ultimo se muestra la lista de estas empresas. Con el IN se asocia la tabla company con los resultados de la subconsulta, aunque también se podría hacer con EXISTS.

- Eliminarán del sistema las empresas que carecen de transacciones registradas, entrega el listado de estas empresas.

```

49 # Eliminarán del sistema las empresas que carecen de transacciones registradas,
50 # entrega el listado de estas empresas.
51 • SELECT c.id,c.company_name
52 FROM company c
53 WHERE c.id NOT IN (
54     SELECT t.company_id
55     FROM transaction t);
56

```

Result Grid

id	company_name
NULL	NULL

company 59 x

Output

Action Output

#	Time	Action	Message
1	11:50:23	SELECT c.id,c.company_name FROM company c WHERE c.id NOT IN (SELECT t.company_id FROM trans...	0 row(s) returned

Nota: Se obtiene una lista con valores NULL ya que todas las empresas han realizado transacciones.

Nivel 2

Ejercicio 1

Identifica los cinco días que se generó la mayor cantidad de ingresos en la empresa por ventas. Muestra la fecha de cada transacción junto con el total de las ventas.

```
62 • SELECT DATE(timestamp) fecha, SUM(amount) total_venta
63 FROM transaction
64 WHERE declined = 0
65 GROUP BY fecha
66 ORDER BY total_venta DESC
67 LIMIT 5;
```

fecha	total_venta
2022-12-13	14337.44
2019-11-18	13591.32
2023-02-20	13332.59
2017-12-20	13318.43
2019-03-18	12680.95

Result 92 ×

Output

Action Output

#	Time	Action	Message
1	13:57:30	SELECT DATE(timestamp) fecha, SUM(amount) total_venta FROM transaction WHERE declined = 0 GROUP BY fecha ORDER BY total_venta DESC LIMIT 5;	5 row(s) returned

Nota: se suman los ingresos para saber cual fue el mayor al ordenarlos, se transforma el campo de timestamp con la función DATE y así poder mostrar únicamente las fechas.

Ejercicio 2

¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor medio.

```
67 # Ejercicio 2
68 #¿Cuál es la media de ventas por país? Presenta los resultados ordenados de mayor a menor medio.
69 • SELECT c.country, ROUND(AVG(t.amount),2) media_ventas
70 FROM transaction t
71 JOIN company c ON t.company_id = c.id
72 GROUP BY c.country
73 ORDER BY media_ventas DESC;
74
```

country	media_ventas
Australia	265.19
United States	264.98
Belgium	261.15
Germany	260.84
Ireland	260.64
Spain	260.47
France	259.98
New Zealand	259.59

Result 69 ×

Output

Action Output

#	Time	Action	Message
1	12:11:52	SELECT c.country, ROUND(AVG(t.amount),2) media_ventas FROM transaction t JOIN company c ON t.company_id = c.id GROUP BY c.country ORDER BY media_ventas DESC;	15 row(s) returned

Nota: Se calcula la media de las ventas con la función AVG y mediante un JOIN se agrupan estas ventas por país.

Ejercicio 3

En tu empresa, se plantea un nuevo proyecto para lanzar algunas campañas publicitarias para hacer competencia a la compañía “Non Institute”. Para ello, te piden la lista de todas las transacciones realizadas por empresas que están ubicadas en el mismo país que esta compañía.

- Muestra el listado aplicando JOIN y subconsultas.

```
80 SELECT c.country, c.company_name, t.*
81 FROM transaction t
82 INNER JOIN company c ON t.company_id = c.id
83 WHERE c.country <> 'Non Institute' AND declined = 0
84 AND c.country = (
85     SELECT country
86     FROM company
87     WHERE company_name = 'Non Institute'
88 )
```

country	company_name	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
United Kingdom	Sed Nunc Ltd	00862984-C9A9-406C-A3D2-7F0DA47BC546	CcS-7063	b-2246	2482	45.7666	4.83048	2015-07-30 12:12:42	486.44	0
United Kingdom	Sed Nunc Ltd	00872BA4-54A3-4B8E-B13F-2D57535AA17A	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0
United Kingdom	Sed Nunc Ltd	01F075B1-07AE-4D02-AAD9-5FFD72A43F3C	CcS-8700	b-2246	4119	55.856	-3.15783	2018-01-27 13:44:36	103.73	0
United Kingdom	Sed Nunc Ltd	023FCE8-E618-4938-BF56-C8DF80594ADD	CcS-7816	b-2246	3235	46.3568	1.82755	2016-12-19 11:53:45	219.28	0
United Kingdom	Sed Nunc Ltd	026838EB-EF91-4564-957B-D6F1662AB7C5	CcS-9471	b-2246	4890	42.1332	12.396	2017-01-10 21:09:29	326.87	0
United Kingdom	Sed Nunc Ltd	02C2F29E-CEf2-4C1E-A594-F476E8F279C0	CcS-9082	b-2246	4501	39.4662	-0.373246	2020-05-24 01:17:29	155.72	0

Result 71 x

Output

Action Output

Time Action Message

1 12:24:47 SELECT c.country, c.company_name, t.* FROM transaction t INNER JOIN company c ON t.company_id = c.id ... 12213 row(s) returned

Nota: Con el JOIN se unen las tablas company y transaction para mostrar las transacciones y las empresas que las realizaron, en el WHERE se excluye la empresa Non institute porque se busca la información de la competencia de esta empresa, y con la subconsulta consigue el país de la empresa para mostrar únicamente las transacciones de empresas ubicadas en el mismo país.

- Muestra el listado aplicando solo subconsultas.

```
93 SELECT t.*
94 FROM transaction t
95 WHERE EXISTS (
96     SELECT c.company_name
97     FROM company c
98     WHERE t.company_id = c.id AND t.declined = 0
99     AND c.country <> 'Non Institute'
100     AND c.country = (
101         SELECT c2.country
102         FROM company c2
103         WHERE c2.company_name = 'Non Institute')
104 )
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
00862984-C9A9-406C-A3D2-7F0DA47BC546	CcS-7063	b-2246	2482	45.7666	4.83048	2015-07-30 12:12:42	486.44	0
00872BA4-54A3-4B8E-B13F-2D57535AA17A	CcS-8475	b-2246	3894	55.6212	-3.7546	2017-10-26 22:08:26	414.06	0
01F075B1-07AE-4D02-AAD9-5FFD72A43F3C	CcS-8700	b-2246	4119	55.856	-3.15783	2018-01-27 13:44:36	103.73	0
023FCE8-E618-4938-BF56-C8DF80594ADD	CcS-7816	b-2246	3235	46.3568	1.82755	2016-12-19 11:53:45	219.28	0
026838EB-EF91-4564-957B-D6F1662AB7C5	CcS-9471	b-2246	4890	42.1332	12.396	2017-01-10 21:09:29	326.87	0
02C2F29E-CEf2-4C1E-A594-F476E8F279C0	CcS-9082	b-2246	4501	39.4662	-0.373246	2020-05-24 01:17:29	155.72	0

transaction 96 x

Output

Action Output

Time Action Message

1 14:21:40 SELECT t.* FROM transaction t WHERE EXISTS (SELECT c.company_name FROM company c WHERE ... 12213 row(s) returned

Nota: Mediante EXISTS se une la tabla company con transaction ya que no se puede utilizar JOIN, la subconsulta verifica que la empresa asociada a cada transacción este en el mismo país que Non institute excluyendo a esta compañía.

Nivel 3

Ejercicio 1

Presenta el nombre, teléfono, país, fecha y amount, de aquellas empresas que realizaron transacciones con un valor comprendido entre 350 y 400 euros y en alguna de estas fechas: 29 de abril de 2015, 20 de julio de 2018 y 13 de marzo de 2024. Ordena los resultados de mayor a menor cantidad.

```
109 • SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) fecha, t.amount
110 FROM company c
111 INNER JOIN transaction t ON c.id = t.company_id
112 WHERE t.declined = '0'
113 AND t.amount BETWEEN 350 AND 400
114 AND DATE(t.timestamp) IN ('2015-04-29', '2018-07-20', '2024-03-13')
115 ORDER BY t.amount DESC;
116
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

company_name	phone	country	fecha	amount
Aliquam PC	01 45 73 52 16	Germany	2024-03-13	399.84
Auctor Mauris Vel LLP	08 09 28 74 14	United States	2018-07-20	399.51
At Pedo Corp.	06 14 48 33 15	Italy	2015-04-29	390.69
Aliquam PC	01 45 73 52 16	Germany	2024-03-13	388.29
Orci Adipiscing Limited	03 18 00 77 81	United Kingdom	2018-07-20	373.71
Fringilla LLC	08 29 15 93 57	New Zealand	2015-04-29	367.62
Pede Cum Ltd	07 62 26 48 38	Norway	2018-07-20	356.87
Auctor Mauris Vel LLP	08 09 28 74 14	United States	2024-03-13	353.75

Result 73 x

Output

Action Output

#	Time	Action	Message
1	12:29:42	SELECT c.company_name, c.phone, c.country, DATE(t.timestamp) fecha, t.amount FROM company c INNER J...	8 row(s) returned

Nota: se usa BETWEEN para filtrar los montos indicados y se transforma timestamp con la función DATE.

Ejercicio 2

Necesitamos optimizar la asignación de los recursos y dependerá de la capacidad operativa que se requiera, por lo que te piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos humanos es exigente y quiere un listado de las empresas en las que especifiques si tienen más de 400 transacciones o menos.

```
122 • SELECT c.company_name, COUNT(t.id) AS transacciones,
123 CASE
124     WHEN COUNT(t.id) > 400 THEN 'mas de 400'
125     ELSE 'menos de 400'
126 END total_transacciones
127 FROM transaction t
128 JOIN company c ON t.company_id = c.id
129 WHERE declined = 0
130 GROUP BY c.id
131 ORDER BY transacciones DESC;
```

Result Grid

company_name	transacciones	total_transacciones
Ac Fermentum Incorporated	2400	mas de 400
Nunc Interdum Incorporated	1599	mas de 400
Donec Fringilla PC	1590	mas de 400
Mauris Institute	1583	mas de 400
Aliquet Vel Vulputate Incorporated	1581	mas de 400

Result 74 x

Output

Action Output

#	Time	Action	Message
1	12:31:00	SELECT c.company_name, COUNT(t.id) AS transacciones, CASE WHEN COUNT(t.id) > 400 THEN 'mas de ...	100 row(s) returned

Nota: se cuenta cada transacción mediante el id de las mismas y se utiliza CASE para poder clasificar las transacciones en mas o menos de 400.