

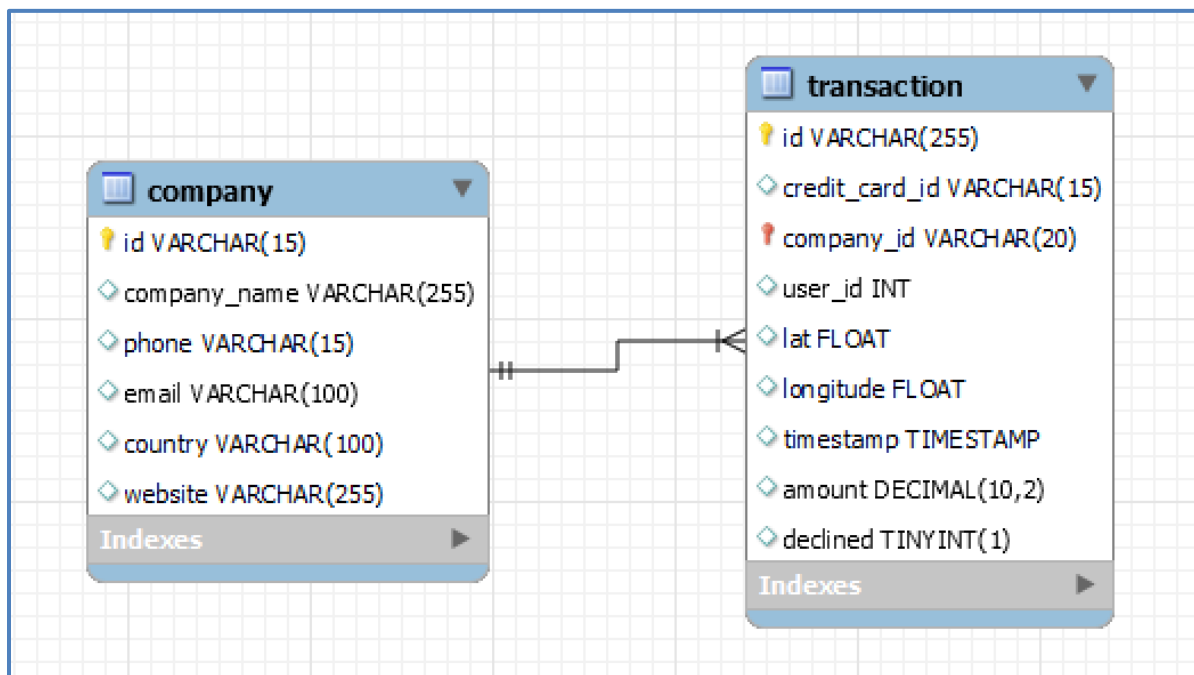
## SPRINT S2.01 – NOCIONES BÁSICAS SQL

## NIVEL 1

**Ejercicio 1.1:**

A partir de los documentos adjuntos (estructura\_dades y dades\_introduir), importar las dos tablas. Mostrar las principales características del esquema creado y explicar las diferentes tablas y variables que existen. Asegurarse de incluir un diagrama que ilustre la relación entre las distintas tablas y variables.

**\*Actualizado** : Se ha incluido el diagrama proporcionado por la opción **Reverse Engineer** del **Workbench**.

**DIAGRAMA DE LA BD TRANSACTIONS**

Tenemos una base de datos denominada **TRANSACTIONS** y está compuesta por 2 tablas. Por un lado, la tabla **company** que contiene como campos toda la información de las compañías con las cuales se han realizado las transacciones. Por otro lado, la tabla **transaction**, la cual contiene información detallada sobre cada transacción realizada.

La tabla **company**, a través de su **PRIMARY KEY id**, se relaciona con la tabla **transaction** y su **FOREIGN KEY** denominado **company\_id**.

La cardinalidad en esta base de datos es de uno a muchos, es decir, una empresa podría haber realizado más de una transacción.

Se observa que se han definido los tipos de variables desde el script de la estructura de datos. También vemos que todas las variables (excepto las **primary key**) admiten valores NULL, ya que de no ser así se hubiera escrito **NOT NULL** junto a la definición de la variable.

Para finalizar, dentro del esquema mostrado, la **tabla de hechos** estaría representada por **transaction** ya que podemos hallar diferentes métricas a través de sus datos; mientras que la tabla **company** vendría a ser una **tabla de dimensiones** ya que describe a través de sus campos a las compañías.

### Ejercicio 1.2:

Listado de países que están haciendo compras. **Asumimos** 2 cosas: (1) Una compra se entiende como una transacción concretada (declined = 0) y (2) Que en el listado no se repitan los países (usar **distinct**)

Se utiliza **JOIN** para las siguientes consultas:

1.2.A Listado de los países que están haciendo compras.

The screenshot shows a SQL IDE interface. At the top, a script editor contains the following SQL query:

```
16 #Ejercicio 1.2:
17 # 1.2.1 - Listado de países que están haciendo compras. Asumimos 2 cosas: (1) Una compra se entiende como una transacción
18 # concretada (declined = 0) y (2) Que en el listado no se repitan los países (distinct)
19
20 • SELECT DISTINCT country FROM company
21 JOIN transaction
22 WHERE declined = '0'
23 ORDER BY country ASC;
```

Below the script editor, the 'Result Grid' tab is active, displaying a list of countries. The countries are: Australia, Belgium, Canada, China, France, Germany, Ireland, Italy, Netherlands, New Zealand, Norway, Spain, Sweden, United Kingdom, and United States. The list is sorted alphabetically.

At the bottom, the 'Output' tab is active, showing a message: 'SELECT DISTINCT country FROM company JOIN transaction WHERE declined = '0' ORDER BY c... 15 row(s) returned'.

1.2.B Desde cuántos países se realizan las compras.

**\*Actualizado:** Se ha incluido la condición de unión entre las tablas a través del uso de ON.

```

25
26 # 1.2.2 - Desde cuántos países se realizan compras.
27 #*Actualizado: Se ha incluido la condición de unión entre las tablas a través del uso de ON*
28
29 • SELECT COUNT(DISTINCT company.country) FROM company
30 JOIN transaction ON company.id = transaction.company_id
31 WHERE transaction.declined = '0';
32

```

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

	COUNT(DISTINCT company.country)
▶	15

Result 13 x

Output

Action Output

#	Time	Action	Message
✓ 712	23:43:04	SELECT COUNT(DISTINCT company.country) FROM company JOIN transaction ON company.id = ...	1 row(s) returned

1.2.C Identificar a la compañía con la mayor media de ventas.

**\*Actualizado:** Se ha dado más claridad al código utilizando más alias para identificar a los campos y sus tablas.

```

32 # 1.2.3 - Compañía con mayor media de ventas:
33 #*Actualizado: Se ha dado más claridad al código utilizando más alias para identificar a los campos y sus tablas *
34
35 • SELECT t.company_id, c.company_name, AVG(t.amount) AS media_ventas
36 FROM transaction t
37 JOIN company c
38 ON c.id = t.company_id
39 WHERE t.declined = '0'
40 GROUP BY t.company_id, c.company_name
41 ORDER BY media_ventas DESC
42 LIMIT 1;
43
44

```

Result Grid | Filter Rows:  | Export: | Wrap Cell Content: | Fetch rows:

	company_id	company_name	media_ventas
▶	b-2398	Eget Ipsum Ltd	481.860000

Result 11 x

Output

Action Output

#	Time	Action	Message
✓ 710	23:41:08	SELECT t.company_id, c.company_name, AVG(t.amount) AS media_ventas FROM transaction t JO...	1 row(s) returned

### Ejercicio 1.3:

Se utiliza solo subconsultas (sin utilizar JOIN). Con **transacción** asumimos que se pide tratar todo aquel registro (con id) en la tabla transaction; indistintamente de si fue declined o no.

#### 1.3.A Transacciones realizadas por empresas de Alemania

```

51 # 1.3.A - Transacciones realizadas por empresas de Alemania.
52
53 • SELECT *
54 FROM transaction
55 WHERE company_id
56 IN (SELECT id
57 FROM company
58 WHERE company.country = 'Germany') ;
59

```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A05DD	CcU-2938	b-2222	275	83.7839	-178.86	2021-07-07 17:43:16	293.57	0
EA2C3281-C9C1-A387-44F8-729FB4B51C76	CcU-2938	b-2222	275	20.2004	-116.84	2021-05-09 10:25:08	119.36	1
0DD2E608-5C9E-D1B3-4999-B99F43AD735A	CcU-2959	b-2234	275	9.68811	130.282	2021-04-17 05:30:17	252.47	1
AB069F53-965E-A2A8-CE06-CA8C4FD92501	CcU-2959	b-2234	275	1.64819	-158.007	2021-04-15 13:37:18	60.99	0
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	170	-43.9695	-117.525	2021-07-26 07:29:18	49.53	0

transaction 14 x

Output

Action Output

#	Time	Action	Message
713	23:43:05	SELECT COUNT(DISTINCT company.country) FROM company JOIN transaction ON company.id = ...	1 row(s) returned

1.3.B Lista de empresas que han realizado transacciones por una cantidad superior a la media de todas las transacciones.

\*La media de todas las transacciones es 256.74, como se muestra en la tabla calculada

```

71 # Media de transacciones:
72 • SELECT AVG(transaction.amount)
73 FROM transaction;
74

```

AVG(transaction.amount)
256.735520

Se ha **actualizado** la query para incluir más campos y así describir con detalle a las empresas que cumplen la condición.

```

60 # 1.3.B - Lista de empresas que han realizado transacciones por una cantidad superior a la media de todas las transacciones.
61 # * Actualizado: se ha incluido toda la información de las empresas en la query *
62
63 • select id, company_name, phone, email, country, website from company where id in
64 (SELECT company_id
65 FROM transaction
66 WHERE amount >=
67 (SELECT AVG(transaction.amount)
68 FROM transaction)
69 GROUP BY company_id)
70 ;
71
72

```

Result Grid

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.co.uk	Norway	https://nytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States	https://ebay.com/sub
b-2258	Vestibulum Lorem PC	02 02 87 33 40	aenean.massa.integer@aol.net	Belgium	https://pinterest.com/sub/cars

company 56 x

Output

Action Output

#	Time	Action	Message
785	01:18:40	select id, company_name, phone, email, country, website from company where id in (SELECT comp...	70 row(s) returned

1.3.C Eliminar del sistema las empresas que carecen de transacciones registradas, entregar el listado de tales empresas:

No se procederá a eliminar ninguna empresa, dado que no existe empresas sin transacciones registradas, según se observa. Hemos **actualizado** la query obviando el uso de DISTINCT e incluyendo la clausula WHERE dentro de la subquery.

```
81 # 1.3.C - Eliminar del sistema las empresas que carecen de transacciones registradas, entregar el listado de tales empresas:
82 # *Se ha obviado el uso de DISTINCT e incluido la clausula WHERE dentro de la subquery*
83
84 • SELECT company_name
85 FROM company c
86 WHERE NOT EXISTS
87 (SELECT company_id
88 FROM transaction t
89 WHERE c.id = t.company_id );
90
```

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

company_name
--------------

company 68 x

Output

Action Output

#	Time	Action	Message
799	11:11:59	SELECT company_name FROM company c WHERE NOT EXISTS (SELECT company_id FROM tr...	0 row(s) returned

## NIVEL 2

### Ejercicio 2.1:

Identificar los 5 días que se generó la mayor cantidad de ingresos en la empresa por ventas. Mostrar la fecha de cada transacción junto con el total de las ventas.

**\*Actualización:** Se ha eliminado el uso del JOIN y la subquery por sugerencia, ya que realmente no eran necesarios. Ahora tenemos una versión más directa y optimizada del código

```

98 • SELECT SUM(transaction.amount) AS total_dia, DATE(transaction.timestamp) AS dia
99 FROM transaction
100 WHERE declined = '0'
101 GROUP BY DATE(transaction.timestamp)
102 ORDER BY total_dia DESC
103 LIMIT 5;
104

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	total_dia	dia
▶	1532.36	2021-12-20
	1397.96	2021-04-22
	1344.37	2021-05-09
	1337.62	2022-02-26
	1325.12	2021-03-29

Result 77 x

Output

Action Output

#	Time	Action	Message
✓ 808	22:53:46	SELECT SUM(transaction.amount) AS total_dia, DATE(transaction.timestamp) AS dia FROM transa...	5 row(s) returned

## Ejercicio 2.2:

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

**\*Actualización:** Ya que declined es un campo booleano, se utilizará la condición `declined = 0`. Observamos una ganancia en rendimiento frente al inicial `declined = '0'`. Asumimos que se entiende por ventas a las transacciones efectivamente concretadas.

```

109
110 • SELECT AVG(amount) AS media_venta, country
111 FROM transaction
112 JOIN company ON transaction.company_id = company.id
113 WHERE declined = 0
114 GROUP BY country
115 ORDER BY media_venta DESC;
116

```

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

	media_venta	country
▶	287.531111	United States
	285.825357	Ireland
	276.668382	Sweden
	271.767527	United Kingdom
	261.941930	Canada

Result 84 x

Output

Action Output

#	Time	Action	Message
✓ 815	00:50:11	SELECT AVG(amount) AS media_venta, country FROM transaction JOIN company ON transaction....	15 row(s) returned

### Ejercicio 2.3:

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

#### 2.3.A Aplicando JOIN y subconsultas.

**\*Actualización:** Se elimina el uso del AND id IS NOT NULL al ser redundante. id al ser primary key no puede ser NULL por definición. También se elimina subconsulta innecesaria, reduciendo el código.

```

127 • SELECT *
128 FROM transaction
129 JOIN company ON company.id = transaction.company_id
130 WHERE company.country = (SELECT country FROM company WHERE company_name = 'non institute');
131

```

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

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount
▶	2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74
	BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95

Result 89 x

Output

Action Output

#	Time	Action	Message
✓ 820	01:06:28	SELECT * FROM transaction JOIN company ON company.id = transaction.company_id WHERE co...	100 row(s) returned

#### 2.3.B Aplicando solamente subconsultas.

**\*Actualización:** Se elimina el uso del AND id IS NOT NULL al ser redundante, al igual que en la consulta anterior.



```

132 # Usando solo subconsultas:
133 # Actualización: Se elimina el uso del AND id IS NOT NULL al ser redundante, al igual que en la consulta anterior.
134
135 • SELECT *
136 FROM transaction
137 WHERE company_id
138 IN (SELECT id
139 FROM company
140 WHERE country =
141 (SELECT country
142 FROM company
143 WHERE company_name = "non institute") )
144 ;
145

```

Result Grid

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	2B928E1C-EC14-A760-0A75-871477649D6A	CcU-2980	b-2246	275	-41.0496	161.685	2021-08-10 08:14:49	383.73	0
	ACD2011A-A2B1-C365-41E1-2AB00C65147A	CcU-2980	b-2246	275	-54.4792	-82.7974	2022-03-05 20:41:20	60.07	1
	4334349E-CEB0-3D68-A4D4-FEB7718A1ACE	CcU-3092	b-2310	275	-20.4859	150.87	2021-05-03 22:37:23	458.74	0
	BC2B9A38-77B4-28CD-1FE8-14DED863E773	CcU-3092	b-2310	275	-78.0295	18.5295	2021-10-18 07:27:35	477.95	1
	1479B3D2-87BA-C7BB-4CE3-8D7C2DE85ABB	CcU-2994	b-2326	133	66.2672	172.399	2021-08-09 00:58:07	309.45	0
	152598C2-029D-D684-4B66-91EDF393EBFF	CcU-2994	b-2326	126	-67.0189	-141.672	2021-07-05 03:10:00	395.43	0
	18E2BFB8-A7E8-7C60-D0C0-CE4E3EDAE02B	CcU-2994	b-2326	121	70.2543	12.1235	2021-07-06 09:48:45	105.06	0

transaction 92 x

Output

Action Output

#	Time	Action	Message
823	01:13:57	SELECT * FROM transaction WHERE company_id IN (SELECT id FROM company WHERE coun...	100 row(s) returned

### NIVEL 3

#### Ejercicio 3.1:

Presentar el nombre, teléfono, país, fecha y cantidad de aquellas empresas que realizaron transacciones con un valor comprendido entre 100 y 200 euros y en alguna de estas fechas: 29 de abril del 2021, 20 de julio del 2021 y 13 de marzo del 2022. Ordenar los resultados de mayor a menor cantidad.

```

135 # NIVEL 3: -----
136
137 # Ejercicio 3.1:
138
139 # Presentar el nombre, teléfono, país, fecha y cantidad de aquellas empresas que realizaron transacciones con un valor
140 # comprendido entre 100 y 200 euros y en alguna de estas fechas: 29 de abril del 2021, 20 de julio del 2021 y 13 de marzo
141 # del 2022. Ordenar los resultados de mayor a menor cantidad.
142
143 • select company_name, phone, country, date(timestamp), amount
144   from company c
145   join transaction t
146   on c.id = t.company_id
147   where date(timestamp) in ('2021-04-29', '2021-07-20', '2022-03-13') and
148   amount between 100 and 200
149   order by amount desc;
150

```

company_name	phone	country	date(timestamp)	amount
Interdum Feugiat Sed Associates	04 88 40 32 52	United Kingdom	2021-07-20	164.86
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2022-03-13	164.32
Enim Condimentum Ltd	09 55 51 66 25	United Kingdom	2021-04-29	149.89
Lorem Eu Incorporated	01 83 66 62 07	Canada	2021-07-20	133.39
Nunc Interdum Incorporated	05 18 15 48 13	Germany	2021-04-29	111.51

Result 409 ×

Output

Action Output

#	Time	Action	Message
✓ 1299	11:31:38	select company_name, phone, country, date(timestamp), amount from company c join tran...	5 row(s) returned

### Ejercicio 3.2:

Necesitamos optimizar las asignaciones de los recursos y dependerá de la capacidad operativa que se requiera, por lo que nos 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 donde se especifique si tienen más de 4 o menos transacciones. **Asumimos que:** Se pide considerar todas las transacciones, independientemente de si fueron concretadas o no (declined).

```

163 # Ejercicio 3.2:
164
165 # Necesitamos optimizar las asignaciones de los recursos y dependerá de la capacidad operativa que se requiera, por lo que
166 # nos piden la información sobre la cantidad de transacciones que realizan las empresas, pero el departamento de recursos
167 # humanos es exigente y quiere un listado de las empresas donde se especifique si tienen más de 4 o menos transacciones.
168 # Asumimos que: Se pide considerar todas las transacciones, independientemente de si fueron concretadas o no (declined).
169
170 • SELECT table321.num_transacciones, company_id, company_name, country,
171 CASE
172     WHEN table321.num_transacciones > 4 THEN 'Más de 4 transacciones'
173     ELSE 'Menos de 4 transacciones'
174 END AS Cantidad_transacciones
175 FROM (SELECT count(id) AS num_transacciones, company_id
176 FROM transaction
177 GROUP BY company_id) AS table321
178 JOIN company
179 ON company.id = table321.company_id
180 ORDER BY num_transacciones;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [fA](#)

	num_transacciones	company_id	company_name	country	Cantidad_transacciones
1	1	b-2578	Dui Quis Institute	New Zealand	Menos de 4 transacciones
1	1	b-2582	Nibh Phasellus Corporation	China	Menos de 4 transacciones
1	1	b-2574	Ordi Adipiscing Limited	United Kingdom	Menos de 4 transacciones
1	1	b-2570	Aliquet Diam Limited	United States	Menos de 4 transacciones
1	1	b-2586	Tempor Diam Institute	Netherlands	Menos de 4 transacciones
1	1	b-2590	Euismod Mauris Institute	Belgium	Menos de 4 transacciones
1	1	b-2594	Et Magnis Ltd	Belgium	Menos de 4 transacciones
1	1	b-2598	Aliquam Iaculis Lacus Corp.	Belgium	Menos de 4 transacciones

Result 94 ×

Output

Action Output

#	Time	Action	Message
825	01:42:30	SELECT table321.num_transacciones, company_id, company_name, country, CASE WHEN table3...	100 row(s) returned