

Sprint 4 Andrés Vidal Monge

Nivel 1

Creamos la nueva base de datos new_transactions

```
38 • create database new_transactions;
```

Output

Action Output	Time	Action	Message
	# 1 00:27:23	create database new_transactions	1 row(s) affected

Creamo tabla companies, todas las tablas las vamos a crear con varchar y null

```
21 • create table if not exists new_transactions.companies (
22     company_id varchar (20) null,
23     company_name varchar (200) null,
24     phone varchar (15) null,
25     email varchar (100) null,
26     country varchar (100) null,
27     website varchar (255) null
28 )
29 ;
30
```

Output

Action Output	Time	Action	Message
	# 1 10:41:41	create table if not exists new_transactions.companies (company_id varchar (20) null, company_name varchar (200) null, ...)	0 row(s) affected

Cargamos los datos del csv en la nueva tabla companies

```
22     # load csv companies
23 •     load data infile
24     'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv'
25     into table new_transactions.companies
26     fields terminated by ','
27     enclosed by ''
28     lines terminated by '\\r\\n'
29     ignore 1 lines (company_id, company_name, phone, email, country, website)
30 ;
31
```

Output

Action Output	Time	Action	Message
	# 1 12:41:34	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\companies.csv' into table new_transactions.c...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
	# 2 12:43:50	SELECT * FROM new_transactions.companies	100 row(s) returned

Creamos tabla credit_cards

```
31     # creamos la siguiente tabla credit_cards
32 •     create table if not exists new_transactions.credit_cards (
33         id varchar (50) null,
34         user_id varchar (100) null,
35         iban varchar (150) null,
36         pan varchar (40) null,
37         pin varchar (4) null,
38         cvv varchar (3) null,
39         track1 varchar (50) null,
40         track2 varchar (50) null,
41         expiring_date varchar(20) null
42     )
43 ;
44
```

Output

Action Output	Time	Action	Message
	# 1 13:11:25	create table if not exists new_transactions.credit_cards (id varchar (50) null, user_id varchar (100) null, iban varchar (1...)	0 row(s) affected

Cargamos los datos a la tabla credit_cards

```
45  # load csv companies
46 •  load data infile
47  'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\credit_cards.csv'
48  into table new_transactions.credit_cards
49  fields terminated by ','
50  enclosed by ''
51  ignore 1 lines (id, user_id, iban, pan, pin, cvv, track1, track2, expiring_date)
52  ;
--
```

Output :::::

Action Output	#	Time	Action	Message
	9	13:24:33	SELECT * FROM new_transactions.credit_cards	275 row(s) returned

Creamos tabla products

```
61  # creamos la siguiente tabla
62 •  create table if not exists new_transactions.products (
63    id varchar(255) null,
64    product_name varchar(255) null,
65    price varchar(255) null,
66    colour varchar(255) null,
67    weight varchar(255) null,
68    warehouse_id varchar(255) null
69  )
70  ;
```

Output :::::

Action Output	#	Time	Action	Message
	1	20:45:01	create table if not exists new_transactions.products (id varchar(255) null, product_name varchar(255) null, price varchar(255) null, colour varchar(255) null, weight varchar(255) null, warehouse_id varchar(255) null)	0 row(s) affected

Cargamos los datos a la tabla products

```
68  # load csv products
69 •  load data infile
70  'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv'
71  into table new_transactions.products
72  fields terminated by ','
73  enclosed by ''
74  ignore 1 lines (id, product_name, price, colour, weight, warehouse_id)
75  ;
```

Output :::::

Action Output	#	Time	Action	Message
	1	20:47:59	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\products.csv' into table new_transactions.products	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```
79  # creamos la tabla transactions
80 •  create table if not exists new_transactions.transactions (
81    id varchar(255) null,
82    card_id varchar(255) null,
83    business_id varchar(255) null,
84    timestamp varchar(255) null,
85    amount varchar(255) null,
86    declined varchar(255) null,
87    product_ids varchar(255) null,
88    user_id varchar(255) null,
89    lat varchar(255) null,
90    longitude varchar(255) null
91  )
92  ;
```

Output :::::

Action Output	#	Time	Action	Message
	1	21:11:33	create table if not exists new_transactions.transactions (id varchar(255) null, card_id varchar(255) null, business_id va...)	0 row(s) affected

Cargamos los datos a la tabla transactions

```
102 # load csv transactions
103 • load data infile
104 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv'
105 into table new_transactions.transactions
106 fields terminated by ';'
107 lines terminated by '\\n'
108 ignore 1 lines (id, card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, longitude)
109 ;
110
```

Output :::::

Action Output	Time	Action	Message
	1 21:50:32	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\transactions.csv' into table new_transactions.transactions tr...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

Creamos tabla users_ca, users_uk y users_usa

```
101 • create table if not exists new_transactions.users_ca (
102     id varchar(255) null,
103     name varchar(255) null,
104     surname varchar(255) null,
105     phone varchar(255) null,
106     email varchar(255) null,
107     birth_date varchar(255) null,
108     country varchar(255) null,
109     city varchar(255) null,
110     postal_code varchar(255) null,
111     address varchar(255) null
112 )
113 ;
114
```

Output :::::

Action Output	Time	Action	Message
	1 22:03:37	create table if not exists new_transactions.users_ca (id varchar(255) null, name varchar(255) null, surname varchar(255) null, phone varchar(255) null, email varchar(255) null, birth_date varchar(255) null, country varchar(255) null, city varchar(255) null, postal_code varchar(255) null, address varchar(255) null)	0 row(s) affected

```
127 # creamos tabla users_uk
128 • create table if not exists new_transactions.users_uk (
129     id varchar(255) null,
130     name varchar(255) null,
131     surname varchar(255) null,
132     phone varchar(255) null,
133     email varchar(255) null,
134     birth_date varchar(255) null,
135     country varchar(255) null,
136     city varchar(255) null,
137     postal_code varchar(255) null,
138     address varchar(255) null
139 )
140 ;
```

Output :::::

Action Output	Time	Action	Message
	1 22:57:48	create table if not exists new_transactions.users_uk (id varchar(255) null, name varchar(255) null, surname varchar(255) null, phone varchar(255) null, email varchar(255) null, birth_date varchar(255) null, country varchar(255) null, city varchar(255) null, postal_code varchar(255) null, address varchar(255) null)	0 row(s) affected

```
153 # creamos tabla users_usa
154 • create table if not exists users_usa (
155     id varchar(255) null,
156     name varchar(255) null,
157     surname varchar(255) null,
158     phone varchar(255) null,
159     email varchar(255) null,
160     birth_date varchar(255) null,
161     country varchar(255) null,
162     city varchar(255) null,
163     postal_code varchar(255) null,
164     address varchar(255) null
165 )
166 ;
```

Output :::::

Action Output	Time	Action	Message
	1 23:04:13	create table if not exists users_usa (id varchar(255) null, name varchar(255) null, surname varchar(255) null, phone varchar(255) null, email varchar(255) null, birth_date varchar(255) null, country varchar(255) null, city varchar(255) null, postal_code varchar(255) null, address varchar(255) null)	0 row(s) affected

Cargamos los datos a la tablas users_ca, users_uk y users_usa

```
114  # load csv users_ca
115 •  load data infile
116  'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_ca.csv'
117  into table new_transactions.users_ca
118  fields terminated by ','
119  enclosed by ''
120  lines terminated by '\\r\\n'
121  ignore 1 lines (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
122 ;
```

Output

Action Output	Time	Action	Message
	1 22:47:38	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_ca.csv' into table new_transactions.user...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

```
141  # load csv users_uk
142 •  load data infile
143  'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv'
144  into table new_transactions.users_uk
145  fields terminated by ','
146  enclosed by ''
147  lines terminated by '\\r\\n'
148  ignore 1 lines (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
149 ;
```

Output

Action Output	Time	Action	Message
	1 22:58:36	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_uk.csv' into table new_transactions.user...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

```
167  # load csv users_usa
168 •  load data infile
169  'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_usa.csv'
170  into table new_transactions.users_usa
171  fields terminated by ','
172  enclosed by ''
173  lines terminated by '\\r\\n'
174  ignore 1 lines (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
175 ;
```

Output

Action Output	Time	Action	Message
	1 23:06:50	load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\users_usa.csv' into table new_transactions.user...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0

Quitamos el simbolo \$ de la columna price en la tabla products

```
199  # quitar el simbolo $ de price para poder cambiar a decimal el data type
200 •  update products
201  set price = replace(price, '$', '')
202  where price like '$%'
203 ;
```

Output

Action Output	Time	Action	Message
	1 00:54:54	update products set price = replace(price, '\$', '') where price like '\$%'	100 row(s) affected Rows matched: 100 Changed: 100 Warnings: 0

Actualizamos los data type a decimal

```
206  # cambiamos data type de price y weight en la tabla products
207 •  alter table new_transactions.products
208  modify column price decimal(10,2),
209  modify column weight decimal(10,2)
210 ;
211
```

Output

Action Output	Time	Action	Message
	1 00:57:15	alter table new_transactions.products modify column price decimal(10,2), modify column weight decimal(10,2)	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0

Creamos una nueva tabla de user para unir las tablas anteriores

```
215  # unir las tablas de user a una sola tabla
216 • create table if not exists new_transactions.users (
217     id varchar(255) null,
218     name varchar(255) null,
219     surname varchar(255) null,
220     phone varchar(255) null,
221     email varchar(255) null,
222     birth_date varchar(255) null,
223     country varchar(255) null,
224     city varchar(255) null,
225     postal_code varchar(255) null,
226     address varchar(255) null
227 )
228 ;
229
```

Output

Action Output	Time	Action	Message
1 01:16:05	create table if not exists new_transactions.users (id varchar(255) null, name varchar(255) null, surname varchar(255) n...	0 row(s) affected	

Añadimos (insertamos) los datos a la tabla user poniendo entre comillas los datos que cambian

```
230  # insertamos los datos a la tabla users cambiando y poniendo en comillas los valores que son distintos.
231 • insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
232 select id, name, surname, phone, email, birth_date, 'Canada' ,city,postal_code,address
233 from users_ca
234 ;
235
236 • insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
237 select id, name, surname, phone, email, birth_date, 'United Kingdom' ,city,postal_code,address
238 from users_uk
239 ;
240
241 • insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address)
242 select id, name, surname, phone, email, birth_date, 'United States' ,city,postal_code,address
243 from users_usa
244 ;
```

Output

Action Output	Time	Action	Message
1 01:18:54	insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address) select id, name, surname, p...	75 row(s) affected	Records: 75 Duplicates: 0 Warnings: 0
2 01:21:58	insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address) select id, name, surname, p...	50 row(s) affected	Records: 50 Duplicates: 0 Warnings: 0
3 01:23:02	insert into users (id, name, surname, phone, email, birth_date,country,city,postal_code,address) select id, name, surname, p...	150 row(s) affected	Records: 150 Duplicates: 0 Warnings: 0

Borramos las tablas que ya no necesitamos

```
246  # borramos las tablas de usuarios que ya no necesitamos y aprovecho en borrar los bk
247 • drop table users_ca;
248 • drop table users_uk;
249 • drop table users_usa;
250 • drop table companies_backup;
251 • drop table credit_cards_bk;
252 • drop table products_bk;
253 • drop table transactions_bk;
```

Output

Action Output	Time	Action	Message
1 01:27:34	drop table users_ca	0 row(s) affected	
2 01:27:37	drop table users_uk	0 row(s) affected	
3 01:27:40	drop table users_usa	0 row(s) affected	
4 01:28:31	drop table companies_backup	0 row(s) affected	
5 01:28:33	drop table credit_cards_bk	0 row(s) affected	
6 01:28:35	drop table products_bk	0 row(s) affected	
7 01:28:36	drop table transactions_bk	0 row(s) affected	

Modificamos los data type de la tabla transactions

```
217 # cambiamos los data type de la tabla transactions
218 • alter table new_transactions.transactions
219 modify column timestamp timestamp,
220 modify column amount decimal(10,2),
221 modify column declined tinyint(1),
222 modify column lat decimal(10,2),
223 modify column longitude decimal(10,2)
224 ;
225
```

Output

#	Time	Action	Message
1	01:41:50	alter table new_transactions.transactions modify column timestamp timestamp, modify column amount decimal(10,2), modify ...	

Hacemos un update a la tabla credit cards para que mysql pueda leer el formato de la fecha de la DB y despues podemos modificar la columna expiring_date en la tabla credit_cards y añadir el (data type) date

```
270 # arreglamos la columna expiring_date para que mysql pueda leer el formato date
271 • update credit_cards
272 set expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%Y')
273 where STR_TO_DATE(expiring_date, '%m/%d/%Y') IS NOT NULL
274 ;
275
276 • alter table credit_cards
277 modify column expiring_date date
278 ;
```

Output

#	Time	Action	Message
1	01:50:53	update credit_cards set expiring_date = STR_TO_DATE(expiring_date, "%m/%d/%Y") where STR_TO_DATE(expiring_date, "%m/%d/%Y") IS NOT NULL	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
2	01:52:07	alter table credit_cards modify column expiring_date date	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Agregamos PK en transactions

```
341 # agregamos la pk en transactions
342 • alter table new_transactions.transactions
343 add constraint PK_id primary key (id)
344 ;
345
```

Output

#	Time	Action	Message
1	12:41:44	alter table new_transactions.transactions add constraint PK_id primary key (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Agregamos PK en products

```
330 # agregamos pk en products
331 • alter table new_transactions.products
332 add constraint PK_id_products primary key (id)
333 ;
334
```

Output

#	Time	Action	Message
1	12:48:41	alter table new_transactions.products add constraint PK_id_products primary key (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Modificamos los data type de la tabla transactions

```
359 # agregar pk a las tablas que faltan
360 • alter table companies
361 add constraint PK_companies primary key (company_id)
362 ;
363 • alter table credit_cards
364 add constraint PK_credit_card primary key (id)
365 ;
366 • alter table products_in_transactions
367 add primary key (transaction_id, product_id)
368 ;
369 • alter table users
370 add constraint PK_users primary key (id)
371 ;
```

Output

Action Output			Message
#	Time	Action	
1	12:58:18	alter table companies add constraint PK_companies primary key (company_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	12:58:40	alter table credit_cards add constraint PK_credit_card primary key (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
3	12:58:43	alter table products_in_transactions add primary key (transaction_id, product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
4	12:58:45	alter table users add constraint PK_users primary key (id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Crear las FK en la tablas transacations

```
373 # crear claves fk en la tabla transactions
374 • alter table transactions
375 add constraint fk_card_id
376 foreign key (card_id)
377 references credit_cards (id)
378 ;
379
```

Output

Action Output			Message
#	Time	Action	
1	13:00:28	alter table transactions add constraint fk_card_id foreign key (card_id) references credit_cards (id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

```
380 • alter table transactions
381 add constraint fk_business_id
382 foreign key (business_id)
383 references companies (company_id)
384 ;
385
```

Output

Action Output			Message
#	Time	Action	
1	13:03:03	alter table transactions add constraint fk_business_id foreign key (business_id) references companies (company_id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

```
386 • alter table transactions
387 add constraint fk_users
388 foreign key (user_id)
389 references users (id)
390 ;
```

Output

Action Output			Message
#	Time	Action	
1	13:04:51	alter table transactions add constraint fk_users foreign key (user_id) references users (id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

```
392 • alter table transactions
393 add constraint fk_product
394 foreign key (id)
395 references products_in_transactions (transaction_id)
396 ;
397
```

Output

Action Output			Message
#	Time	Action	
1	13:07:56	alter table transactions add constraint fk_product foreign key (id) references products_in_transactions (transaction_id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Creamos una nueva tabla puente para separar los productos en transactions

```
281  # creamos una nueva tabla usando los datos de la columna products_ids de la tabla transactions
282  create table if not exists new_transactions.products_in_transactions (
283      transaction_id varchar(255) not null,
284      product_id varchar(255) not null
285  )
286  ;
287  ;
288
```

Output

Action Output	Time	Action	Message
1	11:10:03	create table if not exists new_transactions.products_in_transactions (transaction_id varchar(255) not null, product_id varchar(255) not null)	0 row(s) affected

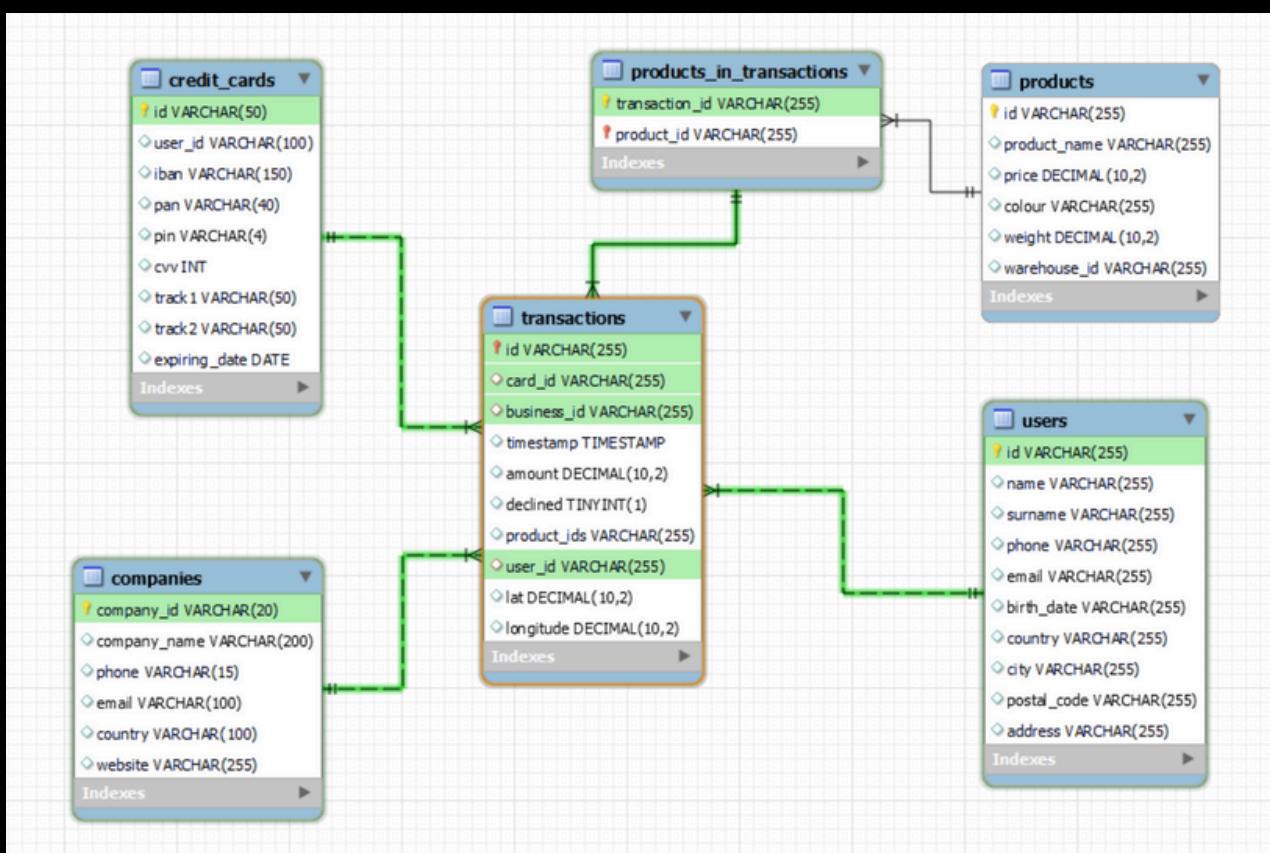
Insertamos los datos a la nueva tabla usando recursive para poder separar los valores de la columna product_ids

```
311  # insertamos los datos a la nueva tabla
312  with recursive numbers as (select 1 as n
313      union all
314      select n + 1
315      from numbers
316      where n < 500
317  ), split_products as(
318      select t.id as transaction_id,
319          trim(substring_index(substring_index(t.product_ids, ',', n), ',', -1)) as product_id
320      from transactions t
321      join numbers
322      on n <= 1 + length(t.product_ids) - length(replace(t.product_ids, ',', '')) 
323  )
324  select *
325  from split_products
326  where product_id <> '';
327
```

Output

Action Output	#	Time	Action	Message
1	12:19:42	with recursive numbers as (select 1 as n union all select n + 1 from numbers where n < 20), split_products as (select 1 as n union all ...)	1457 row(s) returned	
2	12:22:34	insert into products_in_transactions (transaction_id, product_id) with recursive numbers as (select 1 as n union all ...)	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0	

Revisamos como ha quedado el diagrama modelado



Nivel 1

Ejercicio 1

Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.

```
402  # Ejercicio 1
403  # Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.
404
405 • select *
406   from users u
407   where u.id in (select t.user_id
408     from transactions t
409     group by t.user_id
410     having count(*) > 30)
411   order by u.id desc
412 ;
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
92	Lynn	Riddle	1-387-885-4057	vitae.aliquet@outlook.edu	Sep 21, 1984	United States	Bozeman	61871	P.O. Box 712, 7907 Est St.
275	Kenyon	Hartman	082-871-7248	convallis.ante.lectus@yahoo.com	Aug 3, 1982	Canada	Richmond	R8H 2K2	8564 Facilisi. St.
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	Apr 16, 1991	Canada	Tuktoyaktuk	Q4C 3G7	P.O. Box 496, 5145 Sapien Road
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	Dec 26, 1991	Canada	Charlottetown	85X 3P4	Ap #732-8357 Pede, Rd.
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

users 23 x

Output

Action Output

#	Time	Action	Message
1	13:46:14	select * from users u where u.id in (select t.user_id from transactions t group by t.user_id having count(...))	4 row(s) returned

Ejercicio 2

Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.

```
412  # Ejercicio 2
413  # Muestra la media de amount por IBAN de las tarjetas de crédito en la compañía Donec Ltd., utiliza por lo menos 2 tablas.
414
415 • select cc.iban, round(avg(t.amount),2) as media_amount
416   from credit_cards cc
417   join transactions t
418   on cc.id = t.card_id
419   join companies c
420   on c.company_id = t.business_id
421   where c.company_name = 'Donec Ltd'
422   group by cc.iban
423 ;
```

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

iban	media_amount
PT87806228135092429456346	203.72

Result 18 x

Output

Action Output

#	Time	Action	Message
1	13:30:57	select iban, round(avg(t.amount),2) as media_amount from credit_cards cc join transactions t on cc.id = t.card_id join comp...	1 row(s) returned

Nivel 2

Creamos una vista con las transacciones ordenadas por fecha y agrupadas por id de tarjeta

```
445  # ordenamos en una vista los datos de credit card
446 • create view credit_Card_reorder as
447   select
448     t.id,
449     t.card_id,
450     t.timestamp,
451     row_number () over (partition by t.card_id order by t.timestamp desc) as id_rows_CreditCard
452   from transactions t
453 ;
454 
```

Output:

#	Time	Action	Message
1	00:29:02	create view credit_Card_reorder as select t.id, t.card_id, t.timestamp, row_number () over (partition by t.card_id order by t.timestamp desc) as id_rows_CreditCard from transactions t;	0 row(s) affected

Creamos una nueva tabla del estado de las tarjetas con los datos ordenados de base

```
447 • create table if not exists estado_tarjetas (
448   select cr.card_id,cc.iban,
449   case
450     when cr.declined = 1 then 'Inactiva'
451     else 'Activa'
452   end as estado
453   from credit_card_reorder cr
454   join credit_cards cc
455   on cc.id = cr.card_id
456   group by cr.card_id, cc.iban, estado)
457 ;
```

Output:

#	Time	Action	Message
1	01:30:24	create table if not exists estado_tarjetas (select cr.card_id,cc.iban, case when cr.declined = 1 then 'Inactiva' else 'Activa'...) group by cr.card_id, cc.iban, estado);	362 row(s) affected Records: 362 Duplicates: 0 Warnings: 0

Ejercicio 1

¿Cuántas tarjetas están activas?

```
459  # hacemos la query para saber cuantas tarjetas quedan activas despues de los filtros previos
460 • select count(*) as 'tarjetas Activas'
461   from estado_tarjetas et
462   where et.estado = 'Activa'
463 ;
```

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

tarjetas Activas
275

Result 158 x

Output:

#	Time	Action	Message
1	01:48:48	select count(*) as 'tarjetas Activas'from estado_tarjetas et where et.estado = 'Activa'	1 row(s) returned

Nivel 3

Ejercicio 1

Necesitamos conocer el número de veces que se ha vendido cada producto.

Usamos la tabla creada puente para poder hacer un conteo de las transacciones por el id del producto

```
467  # Nivel 3
468  # Ejercicio 1
469  # Necesitamos conocer el número de veces que se ha vendido cada producto.
470
471 • select p.id as 'ID del producto', p.product_name as 'Nombre del producto', count(t.id) as 'Cantidad en transacciones'
472   from products_in_transactions pt
473   join products p
474   on pt.product_id = p.id
475   join transactions t
476   on t.product_ids = p.id
477   group by p.id, p.product_name
478   order by 3 desc
479 ;
480
```

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

ID del producto	Nombre del producto	Cantidad en transacciones
23	riverlands north	408
53	kingsblood Littlefinger the	406
43	duel	390
29	Tully maester Tarly	343
5	skywalker ewok	343
79	Direwolf riverlands the	330
37	Direwolf Littlefinger	306
83	duel tourney	285

Result 28 × Read Only

Output:

Action Output	#	Time	Action	Message
1 15:38:13 select p.id as 'ID del producto', p.product_name as 'Nombre del producto', count(t.id) as 'Cantidad en transacciones' from ... 25 row(s) returned	1	15:38:13	select p.id as 'ID del producto', p.product_name as 'Nombre del producto', count(t.id) as 'Cantidad en transacciones' from ... 25 row(s) returned	25 row(s) returned