

## Partint d'alguns arxius CSV dissenyaràs i crearàs la teva base de dades.

```

1  ##Tasca 4.01. Creació de Base de Dades
2
3  /*Nivell 1 SPRINT 4
4  Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents
5  - Exercici 1
6  Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.*/
7
8  • CREATE DATABASE IF NOT EXISTS transactions_2;
9  • USE transactions_2;
10

```

Output

#	Time	Action	Message	Duration / Fetch
1	22:04:33	CREATE DATABASE IF NOT EXISTS transactions_2	1 row(s) affected	0.031 sec

### /\* Nivell 1 SPRINT 4

Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

```

11  ##Creació de taula
12  • CREATE TABLE IF NOT EXISTS Companies
13  (company_id varchar (50) primary key NOT NULL,
14   company_name varchar(100),
15   phone VARCHAR (50),
16   email VARCHAR (100),
17   country VARCHAR (50),      ## Comentar que no es INT por si empieza por 0
18   website varchar (100));
19
20

```

Output

#	Time	Action	Message
1	22:06:54	CREATE TABLE IF NOT EXISTS Companies (company_id varchar (50) primary key NOT NULL, company...	0 row(s) affected

```

34  • CREATE TABLE IF NOT EXISTS Credit_card
35  (id varchar (50) primary key NOT NULL,
36   user_id int,          ##No poner caracteres en INT
37   iban VARCHAR (50),
38   pan VARCHAR (100),
39   pin VARCHAR (50),     ## Comentar que no es INT por si empieza por 0
40   cvv varchar (100),
41   track1 varchar (100),
42   track2 varchar (100),
43   expiring_date DATE);
44

```

Output

#	Time	Action	Message
1	22:07:49	CREATE TABLE IF NOT EXISTS Credit_card (id varchar (50) primary key NOT NULL, user_id int, ##No p...	0 row(s) affected

```
45 • CREATE TABLE IF NOT EXISTS Products
46 (id VARCHAR (50) primary key NOT NULL,
47 product_name varchar (50),
48 price VARCHAR (50),
49 colour VARCHAR (100),
50 weight VARCHAR (50),
51 warehouse_id varchar (155));
```

Output



Action Output

#	Time	Action	Message
✓ 1	22:08:44	CREATE TABLE IF NOT EXISTS Products (id VARCHAR (50) primary key NOT NULL, product_name varc...	0 row(s) affected

```
53 • CREATE TABLE IF NOT EXISTS Users
54 (id varchar (255) primary key NOT NULL,
55 name varchar (50),
56 surname VARCHAR (50),
57 phone VARCHAR (50),
58 email varchar (100),
59 birth_date VARCHAR (50),
60 country VARCHAR (50),
61 city VARCHAR (50),
62 postal_code VARCHAR (50),
63 address VARCHAR (255));
64
```

Output



Action Output

#	Time	Action	Message
✓ 1	22:09:19	CREATE TABLE IF NOT EXISTS Users (id varchar (255) primary key NOT NULL, name varchar (50), sum...	0 row(s) affected

```
65 • CREATE TABLE IF NOT EXISTS Transactions
66 (id varchar (255) NOT NULL,
67 card_id varchar (50),
68 business_id VARCHAR (50),
69 timestamp TIMESTAMP,
70 amount INT,
71 declined boolean,
72 product_ids VARCHAR (50),
73 user_id VARCHAR (50),
74 lat VARCHAR (50),
75 longitude VARCHAR (50),
76 FOREIGN KEY (card_id) REFERENCES credit_card (id),
77 FOREIGN KEY (business_id) REFERENCES companies (company_id),
78 FOREIGN KEY (user_id) REFERENCES users (id));
79
80
```

Output



Action Output

#	Time	Action	Message
✓ 1	22:10:10	CREATE TABLE IF NOT EXISTS Transactions (id varchar (255) NOT NULL, card_id varchar (50), busine...	0 row(s) affected

```
81  ## Càrrega d'arxius##
82  •  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
83  INTO TABLE transactions_2.companies
84  FIELDS TERMINATED BY ','
85  ENCLOSED BY '"'
86  LINES TERMINATED BY '\n'
87  IGNORE 1 ROWS;
88
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:10:48	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```
--
89  •  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
90  INTO TABLE transactions_2.credit_card
91  FIELDS TERMINATED BY ','
92  ENCLOSED BY '"'
93  LINES TERMINATED BY '\n'
94  IGNORE 1 ROWS
95  (id, user_id, iban, @pan, pin, cvv, track1, track2, @expiring_date)
96  SET
97  pan = REPLACE (@pan, ' ', ''),
98  expiring_date = STR_TO_DATE (@expiring_date, '%m/%d/%y');
99
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:14:00	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0

```
100  •  ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE; ##CONVIERTO EN DATE##
101
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:15:27	ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```
102  •  LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
103  INTO TABLE transactions_2.products
104  FIELDS TERMINATED BY ','
105  ENCLOSED BY '"'
106  LINES TERMINATED BY '\n'
107  IGNORE 1 ROWS;
108
```

Output

Action Output

#	Time	Action	Message
✓ 1	22:15:57	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INTO TABLE tra...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```

109 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv'
110 INTO TABLE transactions_2.users
111 FIELDS TERMINATED BY ','
112 ENCLOSED BY '"'
113 LINES TERMINATED BY '\r\n' ##Como es un archivo WIndows CRLF cada línea, de manera oculta, acaba en \r\n;##
114 IGNORE 1 ROWS;

```

---

Output

Action Output

#	Time	Action	Message
1	22:16:24	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE tra...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

```

116 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv'
117 INTO TABLE transactions_2.users
118 FIELDS TERMINATED BY ','
119 ENCLOSED BY '"'
120 LINES TERMINATED BY '\r\n' ##Como es un archivo WIndows CRLF cada línea, de manera oculta, acaba en \r\n;##
121 IGNORE 1 ROWS;
122

```

---

Output

Action Output

#	Time	Action	Message
1	22:16:48	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' INTO TABLE tra...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0

```

123 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv'
124 INTO TABLE transactions_2.users
125 FIELDS TERMINATED BY ','
126 ENCLOSED BY '"'
127 LINES TERMINATED BY '\r\n' ##Como es un archivo WIndows CRLF cada línea, de manera oculta, acaba en \r\n;##
128 IGNORE 1 ROWS;
129

```

---

Output

Action Output

#	Time	Action	Message
1	22:17:17	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' INTO TABLE tra...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0

**IMPORTANT:** No he volgut en aquesta ocasió a l'hora de carregar les dades a la taula **USERS** nomenar la columna **birth\_date** como **DATE** (com sí he fet amb les dates de la taula **credit\_card**, entenent que vull saber com transformar la columna a **DATE** si ja hem donessin la taula:

```

242 • SET SQL_SAFE_UPDATES = 0;
243 • UPDATE transactions_2.users SET birth_date_format = STR_TO_DATE(birth_date, '%b %e, %Y');
244 • SET SQL_SAFE_UPDATES = 1; ##Se vuelve a activar el modo seguro después##
245

```

---

Output

Action Output

#	Time	Action	Message
1	23:40:35	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
2	23:40:49	UPDATE transactions_2.users SET birth_date_format = STR_TO_DATE(birth_date, '%b %e, %Y')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
3	23:40:53	SET SQL_SAFE_UPDATES = 1	0 row(s) affected

```

135 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
136 INTO TABLE transactions_2.transactions
137 FIELDS TERMINATED BY ';'
138 LINES TERMINATED BY '\r\n'
139 IGNORE 1 ROWS;
140

```

---

Output

Action Output

#	Time	Action	Message
1	22:28:43	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE ...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0

**IMPORTANTE:** Ahora creo una tabla intermedia entre *Products* y *Transactions*, porque en esta última el *id\_products* que nos serviría de clave foránea se encuentra concatenando uno, dos, tres o cuatro id de productos (según la fila), en la misma columna. Para ello, se crea una tabla para separarlos por columnas, con el comando `SUBSTRING_INDEX`.

```

168 • CREATE TABLE products_transactions
169     (id_transaction VARCHAR(255),
170     product_1 VARCHAR(255),
171     product_2 VARCHAR(255),
172     product_3 VARCHAR(255),
173     product_4 VARCHAR(255),
174     PRIMARY KEY (id_transaction, product_1),
175     FOREIGN KEY (product_1) REFERENCES products(id),
176     FOREIGN KEY (product_2) REFERENCES products(id),
177     FOREIGN KEY (product_3) REFERENCES products(id),
178     FOREIGN KEY (product_4) REFERENCES products(id));
179

```

Output

#	Time	Action	Message
1	22:34:32	CREATE TABLE products_transactions (id_transaction VARCHAR(255), product_1 VARCHAR(255), ...	0 row(s) affected

```

180 ##Separar los productos separados en columnas##
181 • INSERT INTO products_transactions (id_transaction, product_1, product_2, product_3, product_4)
182 SELECT id AS id_transaction,
183     TRIM(SUBSTRING_INDEX(product_ids, ',', 1)) AS product_1,
184     CASE WHEN LENGTH(product_ids) - LENGTH(REPLACE(product_ids, ',', '')) + 1 >= 2
185     THEN TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', 2), ',', -1))
186     ELSE NULL END AS product_2,
187     CASE WHEN LENGTH(product_ids) - LENGTH(REPLACE(product_ids, ',', '')) + 1 >= 3
188     THEN TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', 3), ',', -1))
189     ELSE NULL END AS product_3,
190     CASE WHEN LENGTH(product_ids) - LENGTH(REPLACE(product_ids, ',', '')) + 1 >= 4
191     THEN TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(product_ids, ',', 4), ',', -1))
192     ELSE NULL END AS product_4
193 FROM transactions;

```

Output

#	Time	Action	Message
1	22:34:56	INSERT INTO products_transactions (id_transaction, product_1, product_2, product_3, product_4) SELEC...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

Trabajando la tabla, la primera versión de ésta es:

```

195 • SELECT * FROM products_transactions; ##DROP TABLE products_transactions;*/
196

```

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

	id_transaction	product_1	product_2	product_3	product_4
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	71	1	19	NULL
	0466A42E-47CF-8D24-FD01-C0B689713128	47	97	43	NULL
	063FBA79-99EC-66FB-29F7-25726D1764A5	47	67	31	5
	0668296C-CDB9-A883-76BC-2E4C44F8C8AE	89	83	79	NULL
	06CD9AA5-9B42-D684-DDDD-ASE394FEB9A99	43	31	NULL	NULL

products\_transactions 6 x

Output

#	Time	Action	Message
1	22:35:42	SELECT * FROM products_transactions	587 row(s) returned



Transformaremos las columnas en filas, para obtener una sola columna del id de products, creando una nueva tabla, a la que ya a adido las claves de uni n. Antes en transactions, hay que a adir la clave principal id de transactions:

```
207 • SHOW INDEXES FROM transactions;
208 • ALTER TABLE transactions ADD INDEX idx_id (id);
209
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
transactions	1	card_id	1	card_id	A	275	NULL	NULL	YES	BTREE			YES	NULL
transactions	1	business_id	1	business_id	A	100	NULL	NULL	YES	BTREE			YES	NULL
transactions	1	user_id	1	user_id	A	216	NULL	NULL	YES	BTREE			YES	NULL
transactions	1	idx_id	1	id	A	587	NULL	NULL		BTREE			YES	NULL

Result 9 x

#	Time	Action	Message
1	23:06:12	SHOW INDEXES FROM transactions	4 row(s) returned

```
200 • CREATE TABLE transc_prod_DEFINITIVA (
201     id_transaction VARCHAR(255),
202     product_id VARCHAR(10),
203     FOREIGN KEY (id_transaction) REFERENCES transactions(id),
204     FOREIGN KEY (product_id) REFERENCES products(id));
205
```

#	Time	Action	Message
1	23:06:43	CREATE TABLE transc_prod_DEFINITIVA ( id_transaction VARCHAR(255), product_id VARCHAR(10...	0 row(s) affected

```
209 • INSERT INTO transc_prod_definitiva (id_transaction, product_id)
210 SELECT id_transaction, product_1 AS product_id FROM products_transactions
211 UNION ALL SELECT id_transaction, product_2 FROM products_transactions
212 WHERE product_2 IS NOT NULL
213 UNION ALL SELECT id_transaction, product_3 FROM products_transactions
214 WHERE product_3 IS NOT NULL
215 UNION ALL SELECT id_transaction, product_4 FROM products_transactions
216 WHERE product_4 IS NOT NULL;
217
218
```

#	Time	Action	Message
1	00:32:17	INSERT INTO transc_prod_definitiva (id_transaction, product_id) SELECT id_transaction, product_1 AS pr...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0

```
218 • SELECT * FROM transc_prod_definitiva;
219
```

id_transaction	product_id
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	1
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	1
2F499B4D-4DC7-B337-010D-8B7471812A80	1
331A8A52-52D4-D323-0388-1A97C982E441	1
41A2942C-D6E1-B164-2A45-D7ED17237A3A	1
4E562A2B-99E9-8C82-2340-B6623BC9FE2F	1
5458A436-88C5-7CBD-CEB6-8A4545B12CCA	1

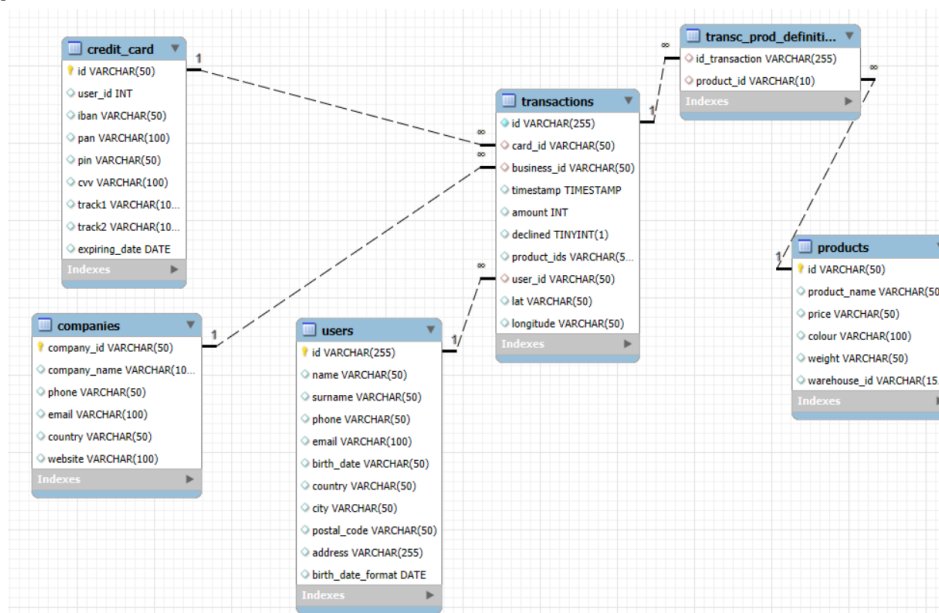
transc\_prod\_definitiva19 x

#	Time	Action	Message
1	00:32:17	INSERT INTO transc_prod_definitiva (id_transaction, product_id) SELECT id_transaction, product_1 AS p...	1457 row(s) affected R
2	00:33:29	SELECT * FROM transc_prod_definitiva	1457 row(s) returned

Respecto a la vinculación que he hecho de la clave id de products, los compañeros me han mostrado otra alternativa con **FIND\_IN\_SET**:

```
INSERT INTO transaction_products (transaction_id, product_id) SELECT tr.id, pr.id FROM transactions t JOIN products p ON FIND_IN_SET(p.id, REPLACE(t.product_ids, ' ', '')).
```

Diagrama estrella, llave foránea en **Transactions** con relación no obligatoria, discontinua, al resto de tablas, excepto a la tabla intermedia, uniéndola con **Products** con una dimensión:



## - Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```

276 • SELECT u.id, u.name, u.surname, COUNT(t.id) nombre_transaccions
277 FROM users u JOIN transactions t ON u.id = t.user_id
278 GROUP BY u.id, u.name, u.surname
279 HAVING COUNT(t.id) > 30;

```

id	name	surname	nombre_transaccions
275	Kenyon	Hartman	48
272	Hedwig	Gilbert	76
267	Ocean	Nelson	52
92	Lynn	Riddle	39

Result 14 x

Output

Action Output

#	Time	Action	Message
1	00:09:20	SELECT u.id, u.name, u.surname, COUNT(t.id) nombre_transaccions FROM users u JOIN transactions t O...	4 row(s) returned

**Nivell 2** Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

**Exercici 1** Quantes targetes estan actives?

```

285 • CREATE TABLE Estat_targetes
286 (card_id VARCHAR(50) PRIMARY KEY,
287  status ENUM('Acceptada', 'Rebutjada'));
288
289

```

Output

Action Output

#	Time	Action	Message
1	00:16:45	CREATE TABLE Estat_targetes (card_id VARCHAR(50) PRIMARY KEY, status ENUM('Acceptada', 'Re...	0 row(s) affected

```

296 • INSERT INTO Estat_targetes (card_id, status)
297 SELECT t.card_id,
298 CASE WHEN (SELECT COUNT(*) FROM transactions t2
299           WHERE t2.card_id = t.card_id
300           ORDER BY t2.timestamp DESC LIMIT 3) =
301           (SELECT COUNT(*) FROM transactions t2
302           WHERE t2.card_id = t.card_id AND t2.declined = 1
303           ORDER BY t2.timestamp DESC LIMIT 3)
304 THEN 'Rebutjada'
305 ELSE 'Acceptada' ##CASE: <Primera Subconsulta> = <Segunda Subconsulta> THEN 'Rebutjada' ELSE 'Acceptada'##
306 END as status
307 FROM transactions t
308 GROUP BY t.card_id;

```

Output

Action Output

#	Time	Action	Message
1	10:25:56	INSERT INTO Estat_targetes (card_id, status) SELECT t.card_id, CASE WHEN (SELECT COUNT(*) FR...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

```

304 ##Comptem les targetes##
305 • SELECT COUNT(*) Targetes_aceptades FROM Estat_targetes
306 WHERE status = 'Acceptada';
307

```

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

Targetes_aceptades
275

Result 16 x

Output

Action Output

#	Time	Action	Message
1	00:21:05	SELECT COUNT(*) Targetes_aceptades FROM Estat_targetes WHERE status = 'Acceptada'	1 row(s) returned

**Nivell 3** Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product\_ids. Genera la següent consulta:



**Exercici 1** Necessitem conèixer el nombre de vegades que s'ha venut cada producte.

**Com ja he vinculat la taula Products al primer exercici, només he de fer la consulta:**

```
315 • SELECT product_id, COUNT(*) Total_vendes_cada_producte
316 FROM transc_prod_definitiva
317 GROUP BY product_id
318 ORDER BY Total_vendes_cada_producte DESC;
319
320
```

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

	product_id	Total_vendes_cada_producte
▶	23	68
	67	68
	79	66
	2	65
	43	65
	47	62
	1	61

Result 20 x

Output

Action Output

#	Time	Action	Message
✓ 1	00:35:57	SELECT product_id, COUNT(*) Total_vendes_cada_producte FROM transc_prod_definitiva GROUP BY ...	26 row(s) returned