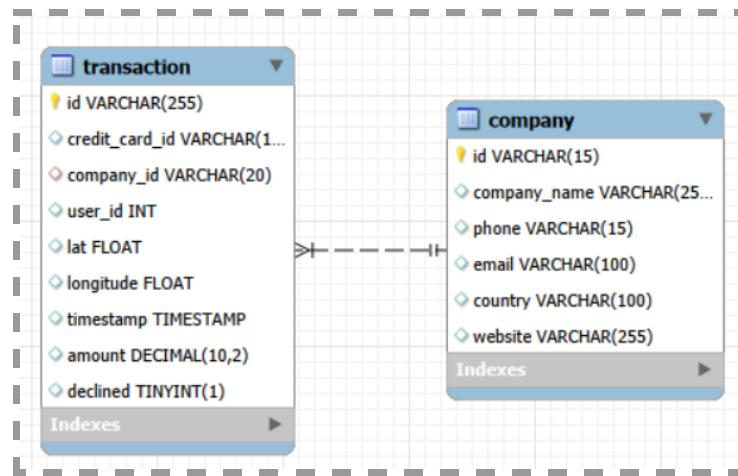


# Tasca S2.01. Nocions bàsiques SQL

## Nivell 1

### Exercici 1

A partir dels documents adjunts (estructura\_dades i dades\_introduir), importa les dues taules. Mostra les característiques principals de l'esquema creat i explica les diferents taules i variables que existeixen. Assegura't d'incloure un diagrama que il·lustri la relació entre les diferents taules i variables.



Creem les taules i importem les dades, des de els documents adjunts, enganxant el text de cada arxiu i executant la totalitat de cada script. Després de comprovar que no han sorgit problemes, i que les taules apareixen al actualitzar les bases de dades, comencem a estudiar-ne el resultat.

En aquest primer esquema de taules podem observar que les dues que conformen la base de dades tenen una relació de 1 a N. La taula de fets (transaction) mostra cada una de les accions comercials de l'empresa. Hi podem trobar una Foreign Key (company\_id) que fa referència a la Primary Key (id) de la taula company, on es detallen les dades de cada companyia (clients). Per tant, cada transacció serà única, però una mateixa companyia pot haver realitzat diverses transaccions.

Entrant a les dades de cada taula, podem observar com cada transacció s'acompanya de característiques com el numero de la tarjeta de credit i l'Id de la companyia (com a valor de text), user\_id com a valor numèric INT, longitud i latitud com a valors FLOAT, La data expressada com a TIMESTAMP, el valor de la transacció, amb dos decimals, al ser un valor monetari i finalment, trobem la columna declined amb característiques booleanes, expressat com a TINYINT.

Per altre banda, totes les característiques de les companyies a la taula de dimensió son característiques qualitatives, i per tant, estan expressades com a dades de tipus VARCHAR.

## Exercici 2

Utilitzant JOIN realitzaràs les següents consultes:

Llistat dels països que estan generant vendes.

```
41 •  SELECT DISTINCT co.country FROM company co
42     JOIN transaction tr ON tr.company_id = co.id;
```

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

country
Germany
Australia
United States
New Zealand
Norway
United Kingdom
Italy
Belgium
Sweden
Ireland
China
Canada
France

Result 3 ×

Output

Action Output

#	Time	Action	Message
2	10:21:14	SELECT COUNT(DISTINCT co.country) AS RecomptePaisos FROM company co JOIN transaction tr ON tr.company_id = co.id	1 row(s) returned
3	10:21:56	SELECT DISTINCT co.country FROM company co JOIN transaction tr ON tr.company_id = co.id	16 row(s) returned

Des de quants països es generen les vendes.

```
47 •  SELECT COUNT(DISTINCT co.country) AS RecomptePaisos FROM company co
48     JOIN transaction tr ON tr.company_id = co.id;
```

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

RecomptePaisos
15

Result 4 ×

Output

Action Output

#	Time	Action	Message
3	10:21:56	SELECT DISTINCT co.country FROM company co JOIN transaction tr ON tr.company_id = co.id	16 row(s) returned
4	10:23:13	SELECT COUNT(DISTINCT co.country) AS RecomptePaisos FROM company co JOIN transaction tr ON ...	1 row(s) returned

El resultat coincideix amb el total de la pregunta anterior.

Identifica la companyia amb la mitjana més gran de vendes.

En aquest cas hem considerat nomes les comandes acabades, que afecten directament a la mitjana.

```
52 •  SELECT co.company_name, ROUND(AVG(tr.amount), 2) AS MitjaCompanyia FROM company co
53     JOIN transaction tr ON tr.company_id = co.id
54     WHERE declined = 0
55     GROUP BY co.id
56     ORDER BY MitjaCompanyia DESC
57     LIMIT 1;
```

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

company_name	MitjaCompanyia
Ac Fermentum Incorporated	284.91

Result 5 ×

Output

Action Output

#	Time	Action	Message
4	10:23:13	SELECT COUNT(DISTINCT co.country) AS RecomptePaisos FROM company co JOIN transaction tr ON ...	1 row(s) returned
5	10:24:19	SELECT co.company_name, ROUND(AVG(tr.amount), 2) AS MitjaCompanyia FROM company co JOIN tr...	1 row(s) returned

## Exercici 3

Utilitzant només subconsultes (sense utilitzar JOIN):

Mostra totes les transaccions realitzades per empreses d'Alemanya.

```
65 •   SELECT tr.id FROM transaction tr
66   WHERE EXISTS (
67     SELECT co.id FROM company co
68     WHERE co.country = 'Germany'
69     AND co.id = tr.company_id)
70 ;
71
```

The screenshot shows a database interface with a code editor at the top containing the SQL query. Below it is a 'Result Grid' table with one column labeled 'id'. The table contains 13291 rows of transaction IDs. At the bottom, there is an 'Output' section showing two log entries:

#	Time	Action	Message
5	10:24:19	SELECT co.company_name, ROUND(AVG(tr.amount), 2) AS MitjaCompanyia FROM company co JOIN transaction tr ON co.id = tr.company_id GROUP BY co.id	1 row(s) returned
6	10:30:30	SELECT tr.id FROM transaction tr WHERE EXISTS (SELECT co.id FROM company co WHERE co.c...)	13291 row(s) returned

Originalment fet amb IN, però canviat a EXISTS per bone pràctiques, amb l'objectiu de millorar l'eficiència.

Llista les empreses que han realitzat transaccions per un amount superior a la mitjana de totes les transaccions.

Després de realitzar la consulta, observem que totes les companyies han fet transaccions per sobre de la mitjana.

```
74 •   SELECT co.company_name FROM company co
75   WHERE EXISTS (
76     SELECT tr.company_id FROM transaction tr
77     WHERE tr.amount > (SELECT AVG(tr.amount) FROM transaction tr)
78     AND tr.company_id = co.id)
79 ;
```

The screenshot shows a database interface with a code editor at the top containing the SQL query. Below it is a 'Result Grid' table with one column labeled 'company\_name'. The table contains 100 rows of company names. At the bottom, there is an 'Output' section showing two log entries:

#	Time	Action	Message
6	10:30:30	SELECT tr.id FROM transaction tr WHERE EXISTS (SELECT co.id FROM company co WHERE co.c...)	13291 row(s) returned
7	10:32:56	SELECT co.company_name FROM company co WHERE EXISTS (SELECT tr.company_id FROM transac...)	100 row(s) returned

Eliminaran del sistema les empreses que no tenen transaccions registrades, entrega el llistat d'aquestes empreses.

```

83 •   SELECT co.company_name
84     FROM company co
85     WHERE NOT EXISTS (
86       SELECT tr.company_id
87         FROM transaction tr
88        WHERE co.id = tr.company_id)
89

```

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

company_name

company 8 x Read Only

Output

Action Output

#	Time	Action	Message
7	10:32:56	SELECT co.company_name FROM company co WHERE EXISTS (SELECT tr.company_id FROM transaction tr WHERE co.id = tr.company_id)	100 row(s) returned
8	10:33:45	SELECT co.company_name FROM company co WHERE NOT EXISTS (SELECT tr.company_id FROM transaction tr WHERE co.id = tr.company_id)	0 row(s) returned

Observem que no retorna cap resultat. Per tant, hem de deduir que totes les companyies tenen transaccions registrades. Podem comparar el nombre total de companyies (100), amb el total de companyies diferents que han fet alguna transacció.

```

90    -- Totes les companyies tenen transaccions registrades.
91    -- El nombre de diferents companyies amb transaccions i companyies a la base de dades coincideix
92 •   SELECT DISTINCT company_id FROM transaction -- 100 companyies
93 •   SELECT DISTINCT id FROM company -- 100 companyies

```

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

id
b-2222
b-2226
b-2230
b-2234
b-2238
k-7747

company 10 x Apply Revert

Output

Action Output

#	Time	Action	Message
9	10:34:49	SELECT DISTINCT company_id FROM transaction	101 row(s) returned
10	10:34:57	SELECT DISTINCT id FROM company	101 row(s) returned

Nota: El resultat original era de 100 resultats en ambdues cerques. Pero la base de dades ha sigut modificada a posteriori, a demanda de la tasca 3, i ara mostra les 100 originals mes una afegida.

## Nivell 2

### Exercici 1

Identifica els cinc dies que es va generar la quantitat més gran d'ingressos a l'empresa per vendes. Mostra la data de cada transacció juntament amb el total de les vendes.

```

100 •   SELECT SUM(amount) AS TotalVentesDiaries, DATE(timestamp) AS Data
101     FROM transaction
102     WHERE declined = "0" -- Tot i no ser significatiu, filtrem nomes vendes consumades
103     GROUP BY Data
104     ORDER BY TotalVentesDiaries DESC
105     LIMIT 5;

```

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

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

Result 11 x Read Only

Output

Action Output

#	Time	Action	Message
10	10:34:57	SELECT DISTINCT id FROM company	101 row(s) returned
11	10:42:39	SELECT SUM(amount) AS TotalVentesDiaries, DATE(timestamp) AS Data FROM transaction WHERE d...	5 row(s) returned

## Exercici 2

Quina és la mitjana de vendes per país? Presenta els resultats ordenats de major a menor mitjà.

```
110 •   SELECT co.country AS pais, ROUND(AVG(tr.amount), 2) AS MitjanaVendesXpais
111     FROM company co
112     JOIN transaction tr ON tr.company_id = co.id
113     WHERE tr.declined = 0 -- En aquest cas demana per vendes, no per transaccions, per tant, afegim la condició
114     GROUP BY pais
115     ORDER BY MitjanaVendesXpais DESC;
116
```

The screenshot shows a MySQL query interface with the following details:

- Result Grid:** Displays a table with two columns: "pais" and "MitjanaVendesXpais". The data is sorted by average sales per country in descending order. The top 15 countries listed are: Australia (265.54), United States (264.42), Belgium (260.97), Germany (260.83), Ireland (260.39), Spain (260.28), France (259.91), New Zealand (259.59), Norway (259.14), Netherlands (258.34), Italy (258.12), Canada (257.41), Sweden (257.40), United Kingdom (256.68), and China (252.60).
- Action Output:** Shows the history of actions taken on the database. It includes two entries:
  - 11 10:42:39 SELECT SUM(amount) AS TotalVentesDiaries, DATE(timestamp) AS Data FROM transaction WHERE d... 5 row(s) returned
  - 12 10:43:40 SELECT co.country AS pais, ROUND(AVG(tr.amount), 2) AS MitjanaVendesXpais FROM company co JO... 16 row(s) returned

## Exercici 3

En la teva empresa, es planteja un nou projecte per a llançar algunes campanyes publicitàries per a fer competència a la companyia "Non Institute". Per a això, et demanen la llista de totes les transaccions realitzades per empreses que estan situades en el mateix país que aquesta companyia.

Mostra el llistat aplicant JOIN i subconsultes.

```
125 •   SELECT * FROM transaction tr
126     JOIN company co ON co.id = tr.company_id
127     WHERE co.country = (
128         SELECT co.country FROM company co
129         WHERE co.company_name = 'Non institute');
```

The screenshot shows a MySQL query interface with the following details:

- Result Grid:** Displays a table with 13 columns: id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined, id, and company\_name. The data is filtered for transactions where the company's country is 'Non institute'. The results show four rows of transaction data for companies with id 2246.
- Action Output:** Shows the history of actions taken on the database. It includes two entries:
  - 12 10:43:40 SELECT co.country AS pais, ROUND(AVG(tr.amount), 2) AS MitjanaVendesXpais FROM company co JO... 16 row(s) returned
  - 13 10:46:15 SELECT \* FROM transaction tr JOIN company co ON co.id = tr.company\_id WHERE co.country = ( SELE... 13776 row(s) returned

Mostra el llistat applicant solament subconsultes.

```

148 •   SELECT t.*, c.*
149     FROM transaction t, company c
150     WHERE company_id IN (
151         SELECT id FROM company
152         WHERE country = (
153             SELECT country FROM company
154             WHERE company_name = 'Non institute')
155     )
156     AND t.company_id = c.id
157 ;
158

```

The screenshot shows the MySQL Workbench interface with the following details:

- Result Grid:** Shows 15 rows of data from the transaction table, joined with the company table via company\_id. The columns include id, credit\_card\_id, company\_id, user\_id, lat, longitude, timestamp, amount, declined, id, and company\_name.
- Action Output:** Displays two log entries:
  - # 14 10:49:54: SELECT t.\* , c.\* FROM transaction t, company c WHERE company\_id IN ( SELECT id FROM company ... 13776 row(s) returned)
  - # 15 10:51:49: SELECT t.\* , c.\* FROM transaction t, company c WHERE company\_id IN ( SELECT id FROM company ... 13776 row(s) returned)

En aquesta solució, hem fet la cerca a les dues taules al FROM, fent servir el WHERE condicional, per eliminar els resultats incorrectes generats.

## Nivell 3

### Exercici 1

Presenta el nom, telèfon, país, data i amount, d'aquelles empreses que van realitzar transaccions amb un valor comprès entre 350 i 400 euros i en alguna d'aquestes dates: 29 d'abril del 2015, 20 de juliol del 2018 i 13 de març del 2024. Ordena els resultats de major a menor quantitat.

```

166 •   SELECT company_name, phone, country, amount FROM company co
167     JOIN transaction tr ON tr.company_id = co.id
168     WHERE DATE(timestamp) IN ("2015-04-29", "2018-07-20", "2024-03-13") AND amount BETWEEN 350 AND 400
169     GROUP BY company_id, amount
170     ORDER BY amount DESC
171 ;

```

The screenshot shows the MySQL Workbench interface with the following details:

- Result Grid:** Shows 8 rows of company data, joined with the transaction table via company\_id. The columns include company\_name, phone, country, and amount.
- Action Output:** Displays two log entries:
  - # 40 10:59:53: SELECT t.\* , (SELECT company\_name FROM company c WHERE c.id = t.company\_id) AS Nom\_compa... Error Code: 1247. Reference 'website' not supported forward reference
  - # 41 10:59:59: SELECT company\_name, phone, country, amount FROM company co JOIN transaction tr ON tr.company... 8 row(s) returned

## Exercici 2

Necessitem optimitzar l'assignació dels recursos i dependrà de la capacitat operativa que es requereixi, per la qual cosa et demanen la informació sobre la quantitat de transaccions que realitzen les empreses, però el departament de recursos humans és exigent i vol un llistat de les empreses on especifiquis si tenen més de 400 transaccions o menys.

```
179 •   SELECT company_id, company_name AS nom, COUNT(tr.id) AS total,  
180   CASE  
181     WHEN COUNT(tr.id) < 400 THEN 'Menys de 400 transaccions'  
182     ELSE 'Mes de 400 transaccions'  
183   END AS transaccions  
184   FROM transaction tr  
185   JOIN company co ON co.id = tr.company_id  
186   GROUP BY company_id  
187   ORDER BY total DESC  
188 ;  
189
```

company_id	nom	total	transaccions
b-2298	Elit Etiam Laoreet Associates	418	Mes de 400 transaccions
b-2534	Mattis Foundation	418	Mes de 400 transaccions
b-2590	Euismod Mauris Institute	418	Mes de 400 transaccions
b-2526	Amet Luctus Vulputate Foundation	411	Mes de 400 transaccions
b-2226	Magna A Neque Industries	410	Mes de 400 transaccions

Output :::::::

#	Time	Action	Message
100191	13:38:00	SELECT company_name, phone, country, SUM(tr.amount) s_amount, DATE(tr.timestamp) ts_timestamp FROM transaction tr JOIN company co ON co.id = tr.company_id WHERE co.id IN ('b-2226', 'b-2578', 'b-2470', 'b-2286', 'b-2362') GROUP BY co.id ORDER BY s_amount DESC;	6 row(s) returned
100192	13:38:47	SELECT company_id, company_name AS nom, COUNT(tr.id) AS total, CASE WHEN COUNT(tr.id) < 400 THEN 'Menys de 400 transaccions' ELSE 'Mes de 400 transaccions' END AS transaccions FROM transaction tr JOIN company co ON co.id = tr.company_id GROUP BY company_id ORDER BY total DESC;	100 row(s) returned

Només les tres darreres companyies tenen menys de 400 transaccions:

b-2226	Magna A Neque Industries	410	Mes de 400 transaccions
b-2578	Dui Quis Institute	400	Mes de 400 transaccions
b-2470	Nec Luctus LLC	399	Menys de 400 transaccions
b-2286	Fringilla LLC	396	Menys de 400 transaccions
b-2362	Lorem Eu Incorporated	378	Menys de 400 transaccions

Finalment, he optat per no incloure les transaccions declinades. Com que estem parlant de necessitats operatives de l'empresa, totes les transaccions s'han de gestionar, acabin, o no, a bon port.