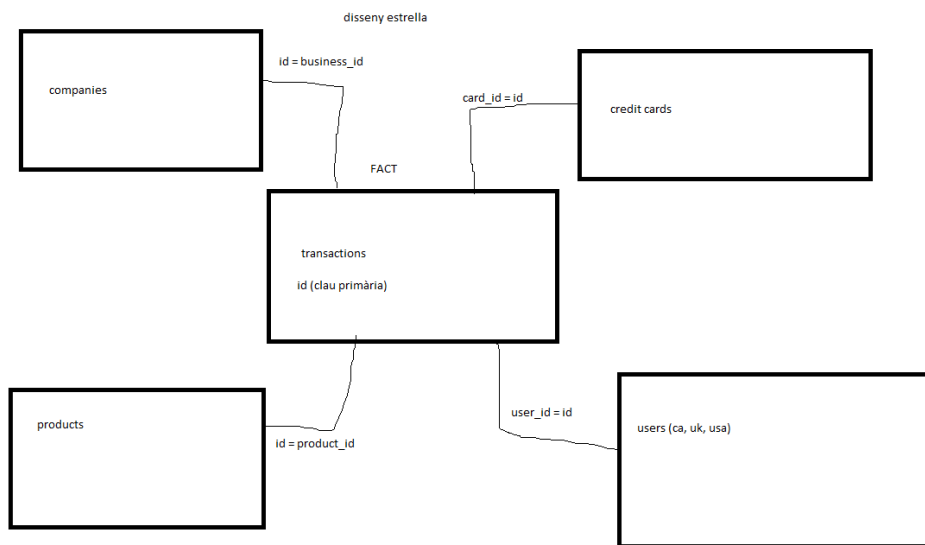


## Tasca S4.01

### # Nivell 1

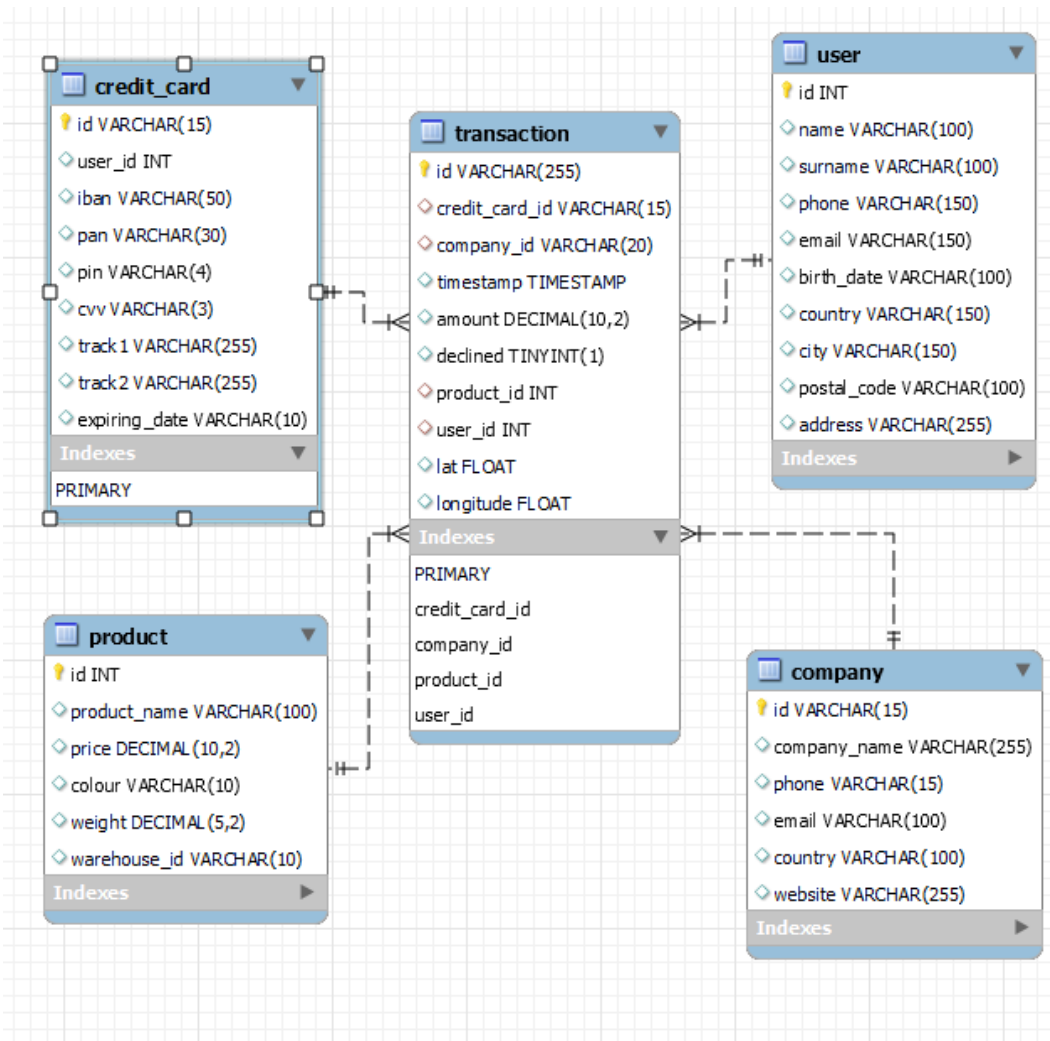


```
6 -- Creem la base de dades i les taules
7 • CREATE DATABASE IF NOT EXISTS database_1;
8 • USE database_1;
9
10 -- M'ha sembla més senzill crear primer les taules secundàries
11 -- i després la taula principal amb les claus foranes
12 -- així m'estalvio haver de crear els índex
13
14 -- Taula companyies
15 • CREATE TABLE IF NOT EXISTS database_1.company (
16     id VARCHAR(15) PRIMARY KEY,
17     company_name VARCHAR(255),
18     phone VARCHAR(15),
19     email VARCHAR(100),
20     country VARCHAR(100),
21     website VARCHAR(255)
22 );
23
24 -- Taula usuaris
25 • CREATE TABLE IF NOT EXISTS database_1.user (
26     id INT PRIMARY KEY,
27     name VARCHAR(100),
28     surname VARCHAR(100),
29     phone VARCHAR(150),
30     email VARCHAR(150),
31     birth_date VARCHAR(100),
32     country VARCHAR(150),
33     city VARCHAR(150),
34     postal_code VARCHAR(100),
35     address VARCHAR(255)
36 );
37
```

```

38      -- Taula tarjetes
39  ● ○ CREATE TABLE IF NOT EXISTS database_1.credit_card (
40      id VARCHAR(15) PRIMARY KEY,
41      user_id INT REFERENCES user(id),
42      iban VARCHAR(50),
43      pan VARCHAR(30),
44      pin VARCHAR(4),
45      cvv VARCHAR(3),
46      track1 VARCHAR(255),
47      track2 VARCHAR(255),
48      expiring_date VARCHAR(10)
49  );
50
51      -- Taula productes
52  ● ○ CREATE TABLE IF NOT EXISTS database_1.product (
53      id INT PRIMARY KEY,
54      product_name VARCHAR(100),
55      price DECIMAL(10, 2),
56      colour VARCHAR(10),
57      weight DECIMAL(5,2),
58      warehouse_id VARCHAR(10)
59  );
60
61      -- Taula principal: transaccions
62  ● ○ CREATE TABLE IF NOT EXISTS database_1.transaction (
63      id VARCHAR(255) PRIMARY KEY,
64      credit_card_id VARCHAR(15),
65      company_id VARCHAR(20),
66      timestamp TIMESTAMP,
67      amount DECIMAL(10, 2),
68      declined BOOLEAN,
69      product_id INT,
70      user_id INT,
71      lat FLOAT,
72      longitude FLOAT,
73      FOREIGN KEY (credit_card_id) REFERENCES credit_card(id),
74      FOREIGN KEY (company_id) REFERENCES company(id),
75      FOREIGN KEY (product_id) REFERENCES product(id),
76      FOREIGN KEY (user_id) REFERENCES user(id)
77  );
78

```



```

INSERT INTO database_1.company (id, company_name, phone, email, country, website) VALUES
('b-2222', 'Ac Fermentum Incorporated', '06 85 56 52 33', 'donec.porrtitor.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-2234', 'Convallis In Incorporated', '06 66 57 29 50', 'mauris.ut@aol.couk', 'Germany', 'https://cnn.com/user/110')
,('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.couk', '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')
  
```

```

INSERT INTO database_1.credit_card (id, user_id, iban, pan, pin, cvv, track1, track2, expiring_date) VALUES
('Ccu-2938', 275, 'TR301950312213576817638661', '5424465566813633', '3257', '984', 'X8B383712448554646XovvsxjDpwievX8604114277', 'X87653663056044187-800716333673', '2022-10-30')
,('Ccu-2945', 274, '0026854763748537475216568689', '5142423821948828', '9080', '887', 'X84621311609958661UftuyfsSeimmX061062824177', 'X84149568437843501-510714033071', '2023-08-24')
,('Ccu-2952', 273, 'B045IVQL52710525608255', '4556 453 55 5287', '4598', '438', 'X82183285104307501CddyrtCuxwfdqX590795543079', 'X86778580257827162-6906859740077', '2021-06-29')
,('Ccu-2959', 272, 'CR7242477244335841535', '372461377349375', '3583', '667', 'X87281111956795320XocddjJbckecdX0901625373', 'X84246154489281853-280522391678', '2023-02-24')
,('Ccu-2966', 271, 'B072LKTQ15627628377363', '448566 886747 7265', '4900', '130', 'X84728932322756223XhlgvsufbmwgjX7202289494377', 'X82318571115599881-890821578475', '2024-10-29')
,('Ccu-2973', 270, 'PT87806228135092429456346', '544 58654 54343 384', '8760', '887', 'X84761405253275637XhjnipoBlefr1X710851563271', 'X87816169831446746-1310277279', '2025-01-30')
,('Ccu-2980', 269, 'DE39241881883086277136', '402400 7145845969', '5075', '596', 'X87320483593870549XookzqrHasedX4901739473', 'X82474313962214151-041221913175', '2022-07-24')
,('Ccu-2987', 268, 'GE89681434837748781813', '3763 747687 76666', '2298', '797', 'X84750646345146674XpjmlyrfHwtrfX8305171276', 'X85441935173418615-410370453677', '2023-10-31')
,('Ccu-2994', 267, 'BH62714428368066765294', '344283273252593', '7545', '595', 'X81583759784015674XmqoyhtUtoqrrX2507851177', 'X84141467473824349-6506800955014', '2022-02-28')
,('Ccu-3001', 266, 'CY49007426654774581266832110', '511722 924833 2244', '9562', '867', 'X86227288756728648XauxilcfmgvdyX208808461178', 'X83429355750963453-530526830573', '2022-09-16')
,('Ccu-3008', 265, 'LU50721693616119230', '4485744464433884', '1856', '748', 'X871824494308529226XlkoutyhTfdvpoX170817547375', 'X86235123731781366-940563816678', '2025-04-05')
  
```

```

INSERT INTO database_1.user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES
(1, 'Zeus', 'Gamble', '1-282-581-0551', 'interdum.enim@protonmail.edu', 'Nov 17, 1985', 'United States', 'Lowell', 73544, '348-7818 Sagittis St.')
,(2, 'Garrett', 'Mcconnell', '(718) 257-2412', 'integer.vitae.nibh@protonmail.org', 'Aug 23, 1992', 'United States', 'Des Moines', 59464, '903 Sit Ave')
,(3, 'Ciaran', 'Harrison', '(522) 598-1365', 'interdum.feugiat@aol.org', 'Apr 29, 1998', 'United States', 'Columbus', 56518, '736-2063 Tellus St.')
,(4, 'Howard', 'Stafford', '1-411-740-3269', 'ornare.egestas@icloud.edu', 'Feb 18, 1989', 'United States', 'Kailua', 77417, 'Ap #545-2244 Erat. Rd.')
,(5, 'Hayfa', 'Pierce', '1-554-541-2077', 'et.malesuada.fames@hotmail.org', 'Sep 26, 1998', 'United States', 'Sandy', 31564, '341-2821 Ultrices Av.')
,(6, 'Joel', 'Tyson', '(718) 288-8020', 'gravida.nunc.sed@yahoo.ca', 'Oct 15, 1989', 'United States', 'Nashville', 96838, '888-2799 Amet Street')
,(7, 'Rafael', 'Jimenez', '(817) 689-0478', 'eget@outlook.ca', 'Dec 4, 1981', 'United States', 'Hillsboro', 29874, '8627 Malesuada Rd.')
,(8, 'Niccim', 'Franko', '(692) 157-3469', 'eestac.aliquam.fringilla@protonmail.ca', 'Aug 1, 1993', 'United States', 'Jackson', 61750, 'An #251-7144 Integer')
  
```

```

introdur_companies  introdur_creat_cara  introdur_user  introdur_product  introdur_transaction

INSERT INTO database_1.product (id, product_name, price, colour, weight, warehouse_id) VALUES
(1, 'Direwolf Stannis', '161.11', '#7c7c7c', 1, 'WH-4')
(2, 'Tarly Stark', '9.24', '#919191', 2, 'WH-3')
(3, 'duel tourney Lannister', '171.13', '#d8d8d8', 1.5, 'WH-2')
(4, 'warden south duel', '71.89', '#111111', 3, 'WH-1')
(5, 'skywalker ewok', '171.22', '#bdbdbd', 3.2, 'WH-0')
(6, 'dooku solo', '136.60', '#c4c4c4', 0.8, 'WH--1')
(7, 'north of Casterly', '63.33', '#b7b7b7', 0.6, 'WH--2')
(8, 'Winterfell', '32.37', '#383838', 1.4, 'WH--3')
(9, 'Winterfell', '76.40', '#b5b5b5', 1.2, 'WH--4')
(10, 'Karstark Dorne', '119.52', '#f4f4f4', 2.4, 'WH--5')
(11, 'Karstark Dorne', '49.70', '#141414', 2.7, 'WH--6')
(12, 'duel Direwolf', '181.60', '#a8a8a8', 2.1, 'WH--7')
(13, 'palpatine chewbacca', '139.59', '#2b2b2b', 1, 'WH--8')
(14, 'Direwolf', '147.53', '#c4c4c4', 2, 'WH--9')

introdur_companies  introdur_creat_cara  introdur_user  introdur_product  introdur_transaction

INSERT INTO database_1.transaction (id, credit_card_id, company_id, timestamp, amount, declined, product_id, user_id, lat, longitude) VALUES
('10881D1D-5823-A76C-55EF-C568E49A05DD', 'CcU-2938', 'b-2222', '2021-07-07 17:43:16', '293.57', '0', '59', '275', '83.7839152128', '-178.860353536')
('7DC26247-20EC-53FE-E555-B6C2E55CA5D5', 'CcU-2945', 'b-2226', '2022-02-04 15:52:56', '312.5', '0', '71', '41', '275', '58.9367181312')
('72997E96-DC2C-A4D7-7C24-66C302F8AE5A', 'CcU-2952', 'b-2230', '2022-01-30 15:16:36', '239.87', '0', '97', '41', '3', '275')
('AB069F53-965E-A2A8-CE06-CAB4CFD92501', 'CcU-2959', 'b-2234', '2021-04-15 13:37:18', '60.99', '0', '11', '13', '61', '29')
('2F3B6AB6-147D-EB08-FE8D-9AAE2EA9D8D5', 'CcU-2966', 'b-2238', '2021-10-18 06:12:03', '33.81', '0', '47', '37', '11', '1')
('580EEF86-B8A1-EFAA-5EE1-27E7DC8F54A4', 'CcU-2973', 'b-2242', '2022-01-06 01:44:48', '42.82', '0', '23', '19', '71', '275')
('2B928E1C-EC14-A760-0A75-871477649D6A', 'CcU-2980', 'b-2246', '2021-08-10 08:14:49', '383.73', '0', '59', '13', '23', '275')
('063FBA79-99EC-66FB-29F7-25726D1764A5', 'CcU-2987', 'b-2250', '2022-01-06 21:25:27', '92.61', '0', '47', '67', '31', '5')
('D33B21EB-A61E-BD0C-5DE7-1DA4AC210AE6', 'CcU-2994', 'b-2254', '2022-03-02 13:29:50', '20.35', '0', '17', '13', '73', '275')
('841AC1A0-9CA6-2AF7-EBC2-BC9C77C1EBB8', 'CcU-3001', 'b-2258', '2021-12-20 02:01:10', '428.4', '0', '47', '275', '-39.6528282624', '88.7919108096')
('D3470F3E-9683-799A-40F1-E42C143BAC5A', 'CcU-3008', 'b-2262', '2021-11-29 19:06:42', '379.14', '0', '1', '275', '-62.8359631872', '-17.71266816')

```

## ## Exercici 1

```

101  /** Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules. */
102
103  -- Transaccions per user
104  • SELECT user_id, COUNT(user_id) AS num_transactions FROM transaction
105  GROUP BY user_id;
106
107  -- No hi ha usuaris amb més de 30 transaccions.
108  • SELECT user.id, user.name, user.surname, COUNT(transaction.user_id) AS num_transactions FROM database_1.user
109  INNER JOIN database_1.transaction
110  ON user.id = transaction.user_id
111  GROUP BY user_id
112  ORDER BY num_transactions DESC;

```

id	name	surname	num_transactions
67	Kenneth	Morrison	27
13	Allegra	Stanton	24
1	Zeus	Gamble	24
17	Bruce	Gill	23
71	Emerson	Hess	23
83	Dana	Ware	22
79	Preston	Hubbard	22
2	Garrett	Mcconnell	21

```

14  -- Si busquessim més de 25 transaccions...
15  • SELECT user.id, user.name, user.surname, COUNT(transaction.user_id) AS num_transactions FROM database_1.user
16  INNER JOIN database_1.transaction
17  ON user.id = transaction.user_id
18  GROUP BY user_id
19  HAVING num_transactions > 25
20  ORDER BY num_transactions DESC;
21
22  -- Resultat:

```

id	name	surname	num_transactions
67	Kenneth	Morrison	27

## ## Exercici 2

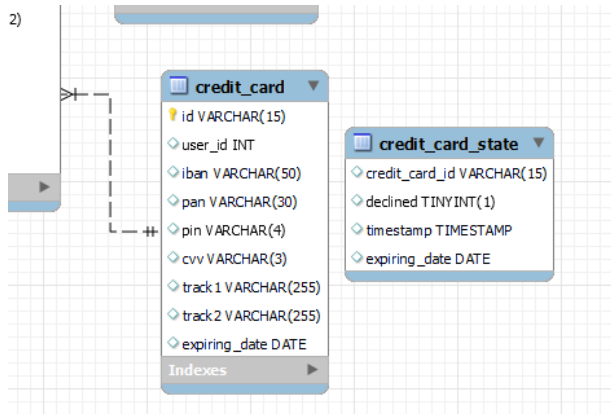
```
137  /** Mostra la mitjana de la suma de transaccions per IBAN de les targetes de crèdit en la companyia Donec Ltd.
138  utilitzant almenys 2 taules. */
139
140  -- Busco la companyia
141  • SELECT * FROM company
142  WHERE company_name LIKE "Donec Ltd"; -- l'id és b-2242
143
144  -- Busco les transaccions i targetes
145  • SELECT * FROM transaction
146  WHERE company_id = "b-2242"; -- Només en tenen una, la CcU-2973
147
148  -- Busco l'iban
149  • SELECT * FROM credit_card
150  WHERE id = "CcU-2973"; -- És el 'PT87806228135092429456346'
151
152  -- La mitjana de les transaccions de la targeta
153  • SELECT AVG(amount) AS avg_amount FROM transaction
154  WHERE credit_card_id = "CcU-2973";
155
156  -- La mitjana de la companyia Donec Ltd
157  • SELECT AVG(amount) AS avg_amount FROM transaction
158  INNER JOIN company
159  ON company.id = transaction.company_id
160  WHERE company_id = (
161  SELECT id FROM company
162  WHERE company_name LIKE "Donec Ltd");
163
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
avg_amount			
203.715000			

## # Nivell 2

```
168  /** Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions
169  van ser declinades i genera la següent consulta: */
170
171  -- Targetes declinades i la seva data de caducitat
172  • SELECT credit_card_id, declined, timestamp, expiring_date
173  FROM transaction
174  INNER JOIN credit_card
175  ON credit_card.id = transaction.credit_card_id
176  WHERE declined = 1
177  ORDER BY timestamp DESC;
178
179  -- Creem la taula
180  • CREATE TABLE credit_card_state AS
181  SELECT credit_card_id, declined, timestamp, expiring_date
182  FROM transaction
183  INNER JOIN credit_card
184  ON credit_card.id = transaction.credit_card_id
185  WHERE declined = 1;
186
187  • SELECT * FROM credit_card_state; -- Comprovem
188
```

	credit_card_id	declined	timestamp	expiring_date
▶	CcU-3225	1	2021-06-28 21:11:42	2022-12-20
	CcU-3071	1	2021-05-11 20:40:06	2023-12-20
	CcU-3141	1	2022-03-04 14:54:35	2024-05-11
	CcU-3309	1	2021-04-10 20:58:41	2020-04-21
	CcU-3435	1	2022-02-02 07:29:36	2021-06-25



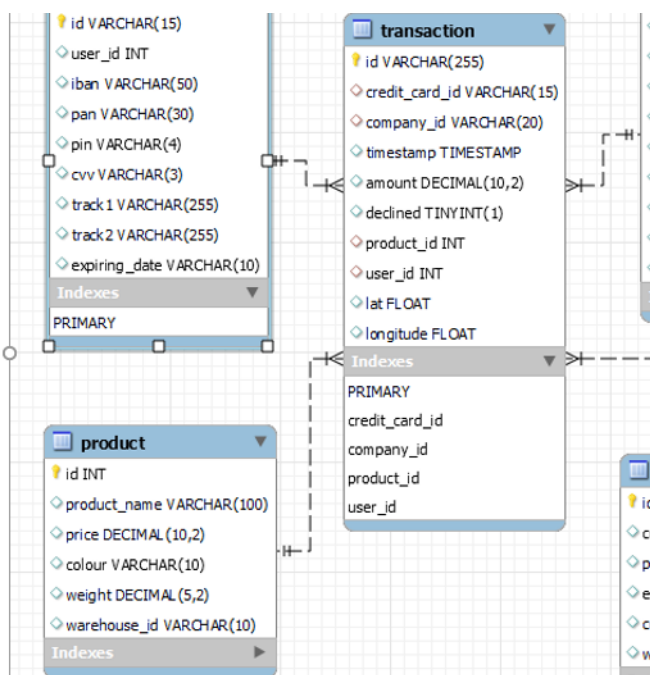
## ## Exercici 1

```

191  /** Quantes targetes estan actives? */
192
193  • SELECT credit_card_id, expiring_date FROM credit_card_state;
194  • SELECT CURRENT_DATE(); -- Millor que NOW(), ja que ens retorna la data en YYYY-MM-DD
195
196  -- Comparo la data de caducitat amb la data actual i mostro les que caduquen avui o més endavant
197  • SELECT credit_card_id, expiring_date FROM credit_card_state
198  WHERE expiring_date >= (SELECT CURRENT_DATE());
199
200  -- Faig recompte de les que estan actives
201  • SELECT COUNT(*) AS num_cards_active
202  FROM (SELECT credit_card_id FROM credit_card_state
203  WHERE expiring_date >= (SELECT CURRENT_DATE())) AS subconsulta;
204  -- No entenc molt bé perquè no funciona amb un sol àlias
205
206  -- Com explico més amunt, he hagut de convertir les dates a format DATE per poder comparar amb la data actual
207
  
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
num_cards_active			
21			

## # Nivell 3



## ## Exercici 1

```
190  /** Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada,
191      tenint en compte que des de transaction tens product_ids. Genera la següent consulta: */
192
193      -- Ja ho he fet des del principi.
194
195      ## Exercici 1
196
197      /** Necessitem conèixer el nombre de vegades que s'ha venut cada producte. */
198
199      • SELECT product.id, product_name, COUNT(product_id) AS num_sales FROM transaction
200        INNER JOIN product
201        ON product.id = transaction.product_id
202        GROUP BY product_id
203        ORDER BY num_sales DESC;
```

Result Grid			
Filter Rows:			
Export: Wrap Cell Content:			
	id	product_name	num_sales
▶	23	riverlands north	32
	79	Direwolf riverlands the	31
	7	north of Casterly	30
	43	duel	29
	61	Winterfell Lannister	28
	17	skywalker ewok sith	26
	47	Tully	26
	53	kingsblood Littlefinger the	26
	89	skywalker ewok	26
	67	Winterfell	24
	37	Direwolf Littlefinger	23
	1	Direwolf Stannis	22