



Tasca S4.01

Creació de Base de Dades

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

NIVELL 1

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:

```
CREATE SCHEMA `sprint4` ;
-- Creem un nou esquema i el definim com a predeterminat.

CREATE TABLE transactions(
-- Crear una taula amb el nom transactions.
    id varchar(255),
    -- Crear camps de data type varchar. Introduïm NOT NULL
    -- perquè no hi pugui haver camp sense informació.
    card_id varchar(255),
    business_id varchar(255),
    timestamp datetime,
    amount decimal(20,2),
    -- Assignem decimal al preu. que podra tenir fins a 10
    -- numeros(introduïm 20 ja que son 10 postius i 10 negatius) i 2
    -- decimals.
    declined bit,
    -- Assignem bit perquè només pot ser, 0, 1 o NULL.
    products_id varchar(255),
    user_id int,
    lat varchar(255),
    longitude varchar(255),
    PRIMARY KEY (id)
-- Assignem la primary key al camp ID.
);

LOAD DATA LOCAL
-- Carreguem les dades de la taula.
INFILE '/Users/kiku/Desktop/ItAcademy/S4 SQL/Resources/transactions.csv'
-- Informem sobre quina és la ruta d'accés a l'arxiu.
INTO TABLE transactions;
-- Assignem la taula on volem afegir la informació.

-- Podríem fer aquest procés per totes les taules però és més
-- ràpid importar-les directament amb el import wizard.
```



Tasca S4.01

Creació de Base de Dades

NIVELL 1

Exercici 1: Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
SELECT users.name, COUNT(transactions.id) AS Transaccions
-- Select dels camps que volem que es mostrin.
FROM sprint4.users
-- Taula d'on obtenim les dades.
JOIN sprint4.transactions
-- Join amb la taula transactions.
ON users.id = transactions.user_id
GROUP BY users.name
HAVING COUNT(transactions.id) > 30;
-- Filtre per contar els id de transaccio i despres agru-
par-los per users.id
```

name	Transaccions
Lynn	39
Ocean	52
Hedwig	76
Kenyon	48



Tasca S4.01

Creació de Base de Dades

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?

```
SELECT *
FROM (
    SELECT credit_cards.id AS Card_ID,
    CASE
        WHEN COUNT(transactions.declined) > 3
        THEN "Active"
        ELSE "Not active"
    END AS "Card Status"
    FROM credit_cards
    LEFT JOIN transactions ON credit_cards.id = transac-
tions.card_id
    WHERE transactions.declined IN (
        SELECT transactions.declined
        FROM transactions
        WHERE transactions.declined = "1")
    GROUP BY credit_cards.id
) card_status
WHERE "Card Status" = "Active";
```

Posem la query de l'exercici anterior dins del FROM i creem un filtre where on definim l'estatus de la credit_card en Active.



Tasca S4.01

Creació de Base de Dades

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:

```
CREATE TABLE products( -- Crear una taula amb el nom transac-
tions.
    id varchar(255), -- Crear camps de data type varchar. Intro-
duim NOT NULL perquè no hi pugui haver camp sense informa-
ció.
    product_name varchar(255),
    price varchar(255),
    colour varchar(255),
    weight varchar(255), -- Assignem decimal al preu. que podra
tenir fins a 10 numeros(introduïm 20 ja que son 10 postius is 10
negatius) i 2 decimals.
    warehouse_id varchar(255),
    PRIMARY KEY (id) -- Assignem la primary key al camp ID.
);
```

```
ALTER TABLE transactions
ADD FOREIGN KEY (product_ids) REFERENCES products(id);
-- Afegim el foreign key a la taula transactions
```

```
LOAD DATA -- Carreguem les dades de la taula.
INFILE '/Users/kiku/Desktop/ItAcademy/S4 SQL/Resources/pro-
ducts.csv' -- Informem sobre quina es la ruta d'accés a l'arxiu.
INTO TABLE products; -- Assignem la taula on volem afegir la in-
formació.
```

```
SELECT * -- Comprobem que la taula s'hagi creat correctament.
FROM products;
```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	\$161.11	#7c7c7c	1	WH-4
10	Karstark Dorne	\$119.52	#f4f4f4	2.4	WH--5
100	south duel	\$40.43	#6d6d6d	3	WH--95
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	\$147.52	#e4e4e4	2	WH--9



Tasca S4.01

Creació de Base de Dades

```
CREATE VIEW Users AS
```

```
-- Creem una nova vista anomenada Users on farem una Union de les  
tres taules. Així tindrem tots els users en una sola taula.
```

```
SELECT *  
FROM users_usa  
UNION ALL  
SELECT *  
FROM users_uk  
UNION ALL  
SELECT *  
FROM users_ca;
```

```
SELECT *  
FROM users;
```

```
-- Comprobem que la vista amb tots els users ha estat creada co-  
rrectament.
```

```
-- Modifiquem el tipus de data type perquè ens deixi assignar PK i  
FK.
```

```
ALTER TABLE transactions
```

```
-- Utilitzem alter table per modificar l'estructura de la taula.
```

```
MODIFY COLUMN id varchar(255);
```

```
-- Modifiquem el tipus de data del camp user_id
```

```
ALTER TABLE companies
```

```
MODIFY COLUMN company_id varchar(255);
```

```
ALTER TABLE credit_cards
```

```
MODIFY COLUMN id varchar(255);
```

```
ALTER TABLE transactions
```

```
MODIFY COLUMN product_ids varchar(255);
```

```
ALTER TABLE transactions
```

```
MODIFY COLUMN card_id varchar(255);
```

```
ALTER TABLE transactions
```

```
MODIFY COLUMN business_id varchar(255);
```

```
ALTER TABLE transactions
```

```
MODIFY COLUMN product_ids varchar(255);
```

```
ALTER TABLE products
```

```
MODIFY COLUMN id varchar(255);
```

```
ALTER TABLE users_ca
```

```
MODIFY COLUMN id varchar(255);
```

```
ALTER TABLE users_usa
```

```
MODIFY COLUMN id varchar(255);
```

```
ALTER TABLE users_uk
```

```
MODIFY COLUMN id varchar(255);
```



Tasca S4.01

Creació de Base de Dades

NIVELL 1

Exercici 2: Mostra la mitjana de la suma de transaccions per IBAN de les targetes de crèdit en la companyia Donec Ltd. utilitzant almenys 2 taules.

```
SELECT credit_cards.id AS Card_ID,  
CASE  
-- Utilitzem case per fer una columna condicional provisional  
    WHEN COUNT(transactions.declined) > 3  
-- Quan el contador de transaccions declinades=1 sigui superior a 3  
    THEN "Active"  
-- Es mostrarà Active  
    ELSE "Not active"  
-- Si no es compleix la condicional, es mostrarà Not Active.  
END AS "Card Status"  
-- Aquest sera el nom de la columna  
FROM credit_cards  
LEFT JOIN transactions ON credit_cards.id = transactions.card_id  
-- Fem un join per obtenir informació de la taula transactions.  
WHERE transactions.declined IN (  
    -- Creem una subquery per seleccionar només les transaccions declinades, així el count serà només d'aquestes transaccions.  
    SELECT transactions.declined  
    FROM transactions  
    WHERE transactions.declined = "1")  
GROUP BY credit_cards.id;  
-- Agrupem el count de les transaccions pel id de les cards.
```

iban	Mitjana
PT87806228135092429456346	203.715



Tasca S4.01

Creació de Base de Dades

-- Crearem totes les Primary keys

```
ALTER TABLE transactions
ADD PRIMARY KEY (id);
ALTER TABLE companies
ADD PRIMARY KEY (company_id);
ALTER TABLE credit_cards
ADD PRIMARY KEY (id);
ALTER TABLE products
ADD PRIMARY KEY (id);
ALTER TABLE users_ca
ADD PRIMARY KEY (id);
ALTER TABLE users_usa
ADD PRIMARY KEY (id);
ALTER TABLE users_uk
ADD PRIMARY KEY (id);
```

```
SET FOREIGN_KEY_CHECKS=0;
```

-- Crearem totes les Foreign keys

```
ALTER TABLE transactions
ADD CONSTRAINT fk_credit_card
FOREIGN KEY (card_id) REFERENCES credit_cards(id);

ALTER TABLE transactions
ADD CONSTRAINT fk_companies
FOREIGN KEY (business_id) REFERENCES companies(company_id);

ALTER TABLE transactions
ADD CONSTRAINT fk_products
FOREIGN KEY (product_ids) REFERENCES products(id);

ALTER TABLE transactions
ADD CONSTRAINT fk_users_ca
FOREIGN KEY (user_id) REFERENCES users_ca(id);

ALTER TABLE transactions
ADD CONSTRAINT fk_users_usa
FOREIGN KEY (user_id) REFERENCES users_usa(id);

ALTER TABLE transactions
ADD CONSTRAINT fk_users_uk
FOREIGN KEY (user_id) REFERENCES users_uk(id);
```

Tasca S4.01

Creació de Base de Dades

NIVELL 3

Exercici 1

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

```
SELECT product_name, COUNT(transactions.product_ids)
-- Creem un contador dels ids de producte de la taula transac-
cions.
FROM products
JOIN transactions
ON products.id = transactions.product_ids
GROUP BY products.product_name;
-- Agrupem el count per els noms de productes.
```

product_name	COUNT(transactions.product_...
Direwolf Stannis	9
Karstark Dorne	3
palpatine chewbacca	3
skywalker ewok sith	4
dooku solo	4
Tarly Stark	3
riverlands north	6
Tully maester Tarly	7
duel tourney Lannister	4
Lannister	3
Direwolf Lannister	9