

## De la creació a la manipulació d'una base de dades relacional

**NOM I COGNOMS:** Cristian Galán Augé

### EXERCICI 1 (25%)

Es pot veure la resolució a l'exercici correspondent.



## EXERCICI 2 (25%)

a)

```
SELECT athlete_id, name, country, substitute_id FROM tb_athlete
WHERE country IN ('ESP') AND name LIKE ('PE%')
ORDER BY athlete_id DESC
```

	athlete_id [PK] character (7)	name character varying (50)	country character (3)	substitute_id character (7)
1	1358251	PENA Inigo	ESP	1302771
2	1349249	PEREZ Maria Isabel	ESP	1307950
3	1349246	PEREZ Marta	ESP	1324396
4	1349245	PEREZ Maria	ESP	1334678
5	1349243	PELETEIRO Ana	ESP	1304891
6	1325882	PEREZ BLANCO Jimena	ESP	1302772

- b) Realitzo combinació natural entre *tb\_athlete* i *tb\_play*, ja que *athlete\_id* guarda la mateixa informació a les dues taules i així aconseguixo tenir el codi de la prova amb els noms dels atletes participants. Després, faig una combinació entre aquesta taula resultant i *tb\_discipline* segons els IDs de la disciplina. Ara només queda seleccionar els atletes que siguin francesos i participin en modalitat de salt.

```
SET search_path TO olimpic;
SELECT d.name, a.name, a.athlete_id
FROM (tb_athlete a NATURAL JOIN tb_play p) JOIN tb_discipline d ON d.discipline_id = p.discipline_id
WHERE country IN ('FRA') AND d.type IN ('JUMP')
ORDER BY d.discipline_id, a.name DESC
```

	name character varying (50)	name character varying (50)	athlete_id character (7)
1	BMX Cycling	VALENTINO Manon	1319979
2	BMX Cycling	MAHIEU Romain	1319977
3	BMX Cycling	ETIENNE Axelle	1319978
4	BMX Cycling	DAUDET Joris	1319973
5	BMX Cycling	ANDRE Sylvain	1319971
6	Trampoline	MORANTE Allan	1334953
7	Trampoline	LABROUSSE Lea	1334991



- c) Faig que compti les aparicions de cada *discipline\_id* i les ordeno de forma descendent, quedant-me amb la primera aparició.

```
SET search_path TO olímpic;
SELECT d.discipline_id, d.name, COUNT(*) AS participant_num
FROM (tb_athlete a NATURAL JOIN tb_play p) JOIN tb_discipline d ON d.discipline_id = p.discipline_id
GROUP BY d.discipline_id
ORDER BY participant_num DESC
LIMIT 1
```

	discipline_id [PK] integer	name character varying (50)	participant_num bigint
1	2	Athletics	2041

- d) Amb la mateixa combinació que l'apartat B, faig un output que només contingui aquells atletes que tinguin més d'una participació.

```
SET search_path TO olímpic;
SELECT a.athlete_id, a.name, a.country, COUNT(a.athlete_id) AS participations
FROM (tb_athlete a NATURAL JOIN tb_play p) JOIN tb_discipline d ON d.discipline_id = p.discipline_id
GROUP BY a.athlete_id
HAVING COUNT(a.athlete_id) > 1
```

	athlete_id [PK] character (7)	name character varying (50)	country character (3)	participations bigint
1	1467919	HAVIK Yoeri	NED	2
2	1330503	BRENNAUER Lisa	GER	2
3	1346205	GANNA Filippo	ITA	2
4	1330510	KLEIN Lisa	GER	2
5	1342530	DYGERT Chloe	USA	2
6	1393561	KOPECKY Lotte	BEL	2

- e) Realitzo la combinació natural entre *tb\_register* i *tb\_athlete*. Entenc que el que es vol retornar és el nom de la disciplina, per tant combino aquesta taula resultant amb *tb\_discipline*.  
Després, agrupo per les tres primeres condicions que diu l'enunciat.. Finalment, les ordeno de forma descendent, de forma que a dalt quedin els atletes que han participat a més rondes. Per seleccionar les màximes observo el resultat i després afegeixo "LIMIT 2".



```
SELECT r.athlete_id, a.name, d.name, COUNT (r.round_number) AS total_rounds
FROM (tb_register r NATURAL JOIN tb_athlete a ) JOIN tb_discipline d ON d.discipline_id = r.discipline_id
GROUP BY r.athlete_id, a.name, d.name
ORDER BY total_rounds DESC
LIMIT 2
```

	athlete_id character (7)	name character varying (50)	name character varying (50)	total_rounds bigint
1	1478152	PROSTOROV Mykola	Trampoline	4
2	1429192	KOCHESOK Susana	Trampoline	4



### EXERCICI 3 (25%)

a)

```

1  --A
2  SET search_path TO olympic;
3  INSERT INTO olympic.tb_athlete(athlete_id, name, country, substitute_id)
4  VALUES('0000001', 'REMBRAND Luc', 'FRA', NULL);
5
6  INSERT INTO olympic.tb_athlete(athlete_id, name, country, substitute_id)
7  VALUES('0000002', 'SMITH Mike', 'ENG', NULL);
8
9  INSERT INTO olympic.tb_athlete(athlete_id, name, country, substitute_id)
10 VALUES('0000003', 'LEWIS Carl', 'USA', NULL);
11
12 SELECT * FROM tb_athlete
13 WHERE athlete_id LIKE ('000000%')
14 |
15

```

Data Output	Explain	Messages	Notifications
athlete_id [PK] character (7)	name character varying (50)	country character (3)	substitute_id character (7)
1 0000001	REMBRAND Luc	FRA	[null]
2 0000002	SMITH Mike	ENG	[null]
3 0000003	LEWIS Carl	USA	[null]

b) Utilitzem la forma NOT per marcar que hem de fer check que no hi hagi cap atleta espanyol que no tingui substitut. Per comprovar-ho, creem un atleta fictici espanyol sense substitut i veiem que retorna que es viola la restricció abans posada.

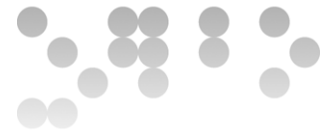
```

16 --B
17 SET search_path TO olympic;
18 ALTER TABLE tb_athlete
19 ADD CONSTRAINT spanish_substitute CHECK(NOT (country = 'ESP' AND substitute_id IS NULL));
20
21 SET search_path TO olympic;
22
23 INSERT INTO olympic.tb_athlete(athlete_id, name, country, substitute_id)
24 VALUES('0000004', 'RAJOY Mariano', 'ESP', NULL);
25

```

Data Output	Explain	Messages	Notifications
ERROR: new row for relation "tb_athlete" violates check constraint "spanish_substitute"			
DETAIL: Failing row contains (0000004, RAJOY Mariano, ESP, null).			
SQL state: 23514			

c)



```

32 -- C
33 CREATE VIEW olimpico.exercise33 AS (
34     SELECT * FROM tb_athlete WHERE country IN ('ESP') AND name LIKE ('PE%') ORDER BY athlete_id DESC)
35 SELECT * FROM olimpico.exercise33;

```

	athlete_id	name	country	substitute_id	date_add
	character (7)	character varying (50)	character (3)	character (7)	date
1	1358251	PENA Inigo	ESP	1302771	2021-11-03
2	1349249	PEREZ Maria Isabel	ESP	1307950	2021-11-03
3	1349246	PEREZ Marta	ESP	1324396	2021-11-03
4	1349245	PEREZ Maria	ESP	1334678	2021-11-03
5	1349243	PELETEIRO Ana	ESP	1304891	2021-11-03
6	1325882	PEREZ BLANCO Jimena	ESP	1302772	2021-11-03

d) S'ha d'explicitar que els valors no poden ser NULL, ja que si només s'entrés "DEFAULT CURRENT DATE", l'usuari podria entrar NULL com a valor.

```

38 SET search_path TO olimpico;
39 ALTER TABLE tb_athlete
40 ADD COLUMN date_add DATE NOT NULL DEFAULT CURRENT_DATE
41
42 SELECT * FROM tb_athlete
43 LIMIT 5

```

	athlete_id	name	country	substitute_id	date_add
	[PK] character (7)	character varying (50)	character (3)	character (7)	date
1	1346266	AALERUD Katrine	NOR	[null]	2021-11-01
2	1344792	ABASS Abobakr	SUD	[null]	2021-11-01
3	1328854	ABDALLA Abubaker H...	QAT	[null]	2021-11-01
4	1306332	ABDEL LATIF Radwa	EGY	[null]	2021-11-01
5	1306339	ABDEL RAZEK Samy	EGY	1306332	2021-11-01

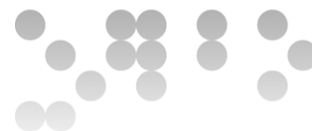
e) Creo un rol que permeti connectar-se a l'esquema olímpic i li dono els permisos que es demanen. Després, creo l'usuari i li dono el rol "regist" que conté els permisos demanats.



```
--E
```

```
CREATE ROLE regist;  
GRANT USAGE ON SCHEMA olimpico TO regist;  
GRANT SELECT, INSERT, UPDATE, DELETE ON tb_register TO regist;  
GRANT SELECT ON tb_athlete TO regist;  
CREATE USER registerer WITH PASSWORD '1234';  
GRANT regist TO registerer;
```

A l'obrir sessió amb el nou usuari, es poden veure les diferents taules i columnes però no s'hi pot accedir excepte a les que té permís. Tampoc es poden crear usuaris.



## EXERCICI 4 (25%)

a)

GROUPING SET: A SQL, la funció GROUP BY permet agrupar les tuples segons diferents atributs i quantificar segons aquests atributs.

Per exemple, podem agafar com exemple una botiga que vengui electrodomèstics i vulgui calcular quants electrodomèstics han venut i la seva marca. Els electrodomèstics poden ser de diferents marques (Bosch, Balay, TEKA, etc.) i les diferents marques poden vendre diferents electrodomèstics (neveres, microones, rentadores, etc). GROUP BY permetria fer, per ell sol, l'agrupació per marca, per tipus d'electrodomèstic o per totes dues coses al mateix temps. El que no permet per si sol, és agrupar per les tres permutacions al mateix temps.

GROUPING SET permet agrupar per aquestes dues variables per separat i juntes de forma ràpida sense haver de recórrer a altres construccions de codis, en aquest cas, tan sols s'hauria de fer el següent:

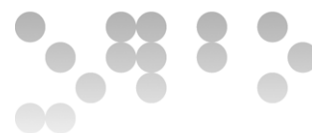
```
GROUP BY
GROUPING SETS (
    (marca, tipus_electrodomèstic)
    (marca)
    (tipus_electrodomèstic)
    ())
```

L'última línia serviria per si volguéssim tenir la informació sense agrupar per cap de les dues categories.

El resultat que rebriem seria el següent, prenent com a marques BOSCH, BALAY, TEKA i com a tipus\_electrodomèstic BOSCH, NEVERES, MICROONES, RENTADORES

marca	Tipus_electrodomèstic	Ventes
NULL	NULL	VALOR 1
NULL	NEVERES	VALOR 2
NULL	MICROONES	VALOR 3
NULL	RENTADORES	VALOR 4
BOSCH	NULL	VALOR 5
BALAY	NULL	VALOR 6
TEKA	NULL	VALOR 7
BOSCH	NEVERES	VALOR 8
BOSCH	MICROONES	VALOR 9
BOSCH	RENTADORES	VALOR 10
BALAY	NEVERES	VALOR 11
BALAY	MICROONES	VALOR 12
BALAY	RENTADORES	VALOR 13
TEKA	NEVERES	VALOR 14





TEKA	MICROONES	VALOR 15
TEKA	RENTADORES	VALOR 16

La primera línia correspon a l'agrupació tant per marca com per tipus d'electrodomètic, les següents 3 agrupant per marca, les següents a aquestes a l'agrupació per tipus d'electrodomètic i, les últimes 8 files corresponen als resultats sense aplicar agrupació.

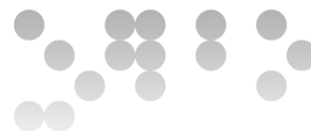
ROLL UP: És una altra funció de GROUP BY, la qual serveix per retornar subtotals de categories amb una jerarquia dins les columnes seleccionades, retornant una columna NULL per aquell valor que estigui agrupant. Amb l'anterior exemple, imaginem per simplificar que els productes TEKA i les rentadores no s'han entrat a la base de dades. Tindrem venda de productes BOSCH, BALAY. D'aquestes marques, tindrem NEVERES i MICROONES. El resultat que rebriem d'executar **GROUP BY ROLLUP (marca, tipus\_electrodomètic)** seria el següent:

marca	Tipus_electrodomètic	Ventes
BOSCH	NEVERES	VALOR 1
BOSCH	MICROONES	VALOR 2
BOSCH	NULL	VALOR 3
BALAY	NEVERES	VALOR 4
BALAY	MICROONES	VALOR 5
BALAY	NULL	VALOR 6
NULL	NULL	VALOR 7

CUBE: De forma similar a ROLL UP, CUBE retorna subtotals segons les categories. La diferencia amb ROLL UP, és que CUBE retorna totes les combinacions possibles i sense aquesta jerarquia. De forma que, fent servir **GROUP BY CUBE (marca, tipus\_electrodomètic)** primer ens retornarà tots els valors per la columna tipus\_electrodomètic amb el valor NEVERES i després per MICROONES. A continuació, ens retornarà els valors agrupats per MARCA.

marca	Tipus_electrodomètic	Ventes
BOSCH	NEVERES	VALOR 1
BALAY	NEVERES	VALOR 2
NULL	NEVERES	VALOR 3
BOSCH	MICROONES	VALOR 4
BALAY	MICROONES	VALOR 5
NULL	MICROONES	VALOR 6
NULL	NULL	VALOR 7
BOSCH	NULL	VALOR 8
BALAY	NULL	VALOR 9

b)



Aquesta clàusula permet generar columnes que siguin operacions d'altres, ja siguin operacions numèriques o a partir d'atributs formats per caràcters.

Un exemple per explicar aquesta variable seria la de formació de correus electrònics a partir dels noms i cognoms del treballador o treballadora. Suposem que tenim la següent taula:

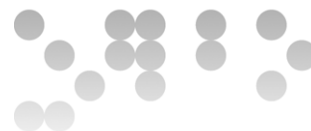
TREBALLADORS (DNI, nom, cognom1, cognom2, email)

Si volguéssim fer la construcció del correu electrònic, podríem elegir fer servir el nom, un punt i el cognom. En el moment de generar la taula a SQL fariem el següent:

```
CREATE TABLE treballadors (  
dni CHAR(9) PRIMARY KEY  
nom VARCHAR (30) NOT NULL  
cognom1 VARCHAR (50) NOT NULL  
cognom2 VARCHAR(50)  
email GENERATED ALWAYS AS (CONCAT(nom,'.',cognom1,'@uoc.edu')));
```

ens generaria el següent:

dni	nom	cognom1	cognom2	email
12345678A	Cristian	Galan	Auge	cristian.galan@uoc.edu



## Criteris de valoració

En l'enunciat s'indica el pes/valoració de cada exercici.

Per aconseguir la puntuació màxima en els exercicis, cal explicar amb claredat la solució que es proposa.

## Format i data de lliurament

Heu d'enviar la PAC a la bústia de «Lliurament i registre d'AC» disponible a l'aula (apartat Avaluació). El format de l'arxiu que conté la vostra solució pot **ser .pdf, .doc i .docx. Per a altres opcions, si us plau, contacteu prèviament amb el vostre consultor.**

La data límit per lliurar la PAC2 és el **8/11/2021**.

### Nota: Propietat intel·lectual

Al presentar una pràctica o PAC que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en el qual es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a cap altra llicència d'ús (*Creative Commons*, llicència GNU, GPL etc.).

L'estudiant s'haurà d'assegurar que la llicència que sigui no impedeixi específicament el seu ús en el marc de la pràctica o PAC. En el cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright. A més, serà necessari adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si així correspon.