



# Sprint 4 - Creació de Base de Dades

## 1 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:

```
1 • CREATE TABLE companies (  
2     id VARCHAR(10) PRIMARY KEY,  
3     name VARCHAR(50),  
4     address VARCHAR(100),  
5     phone VARCHAR(20),  
6     email VARCHAR(100),  
7     country VARCHAR(20),  
8     city VARCHAR(20),  
9     postal_code VARCHAR(10));  
10  
11 • CREATE TABLE products (  
12     id VARCHAR(10) PRIMARY KEY,  
13     product_name VARCHAR(50),  
14     price DECIMAL(10, 2),  
15     colour VARCHAR(20),  
16     weight DECIMAL(10, 2),  
17     warehouse_id VARCHAR(12));
```

```

19 • CREATE TABLE transactions (
20     id VARCHAR(10) PRIMARY KEY,
21     user_id VARCHAR(10),
22     product_ids VARCHAR(50),
23     amount DECIMAL(10, 2),
24     transaction_date DATETIME,
25     payment_method VARCHAR(20));
26
27 • CREATE TABLE user_data (
28     id VARCHAR(10),
29     name VARCHAR(20),
30     surname VARCHAR(40),
31     phone VARCHAR(20),
32     email VARCHAR(100),
33     birth_date VARCHAR(12),
34     country VARCHAR(20),
35     city VARCHAR(20),
36     postal_code VARCHAR(5),
37     address VARCHAR(100));

39 • CREATE TABLE credit_cards (
40     id VARCHAR(11) PRIMARY KEY,
41     user_id VARCHAR(10),
42     card_number VARCHAR(16),
43     card_type VARCHAR(20),
44     expiry_date DATE,
45     cvv VARCHAR(4));
46

```

En algunos casos tuve que cambiar el tipo de dato

```

49 • alter table transactions
50     modify column amount decimal(10,2);
51 • alter table transactions
52     modify column product_ids varchar(50);
53 • alter table transactions
54     modify column user_id varchar(11);
55 • alter table credit_cards
56     modify column id varchar(11) PRIMARY KEY;
57

```

Para importar los .csv, me salía el error 2068, relacionado a permisos de FILE. Después de muchísimas búsquedas y estos códigos, pude hacer el LOAD FILE. (via cambios en el archivo my.ini)

```

59 • show global variables like "local_infile";
60 • set global local_infile=1;
61 --
62 • show grants;
63 • GRANT FILE on *.* to 'root'@'localhost';
64 --
65 • SHOW SESSION VARIABLES LIKE 'local_infile';
66 • show variables like 'pid_file';

```

---

## Load local infile:

```
70 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/users_usa.csv'
71 INTO TABLE user_data
72 FIELDS TERMINATED BY ','
73 ENCLOSED BY '"'
74 LINES TERMINATED BY '\r\n'
75 IGNORE 1 ROWS;
76
77 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/users_uk.csv'
78 INTO TABLE user_data
79 FIELDS TERMINATED BY ','
80 ENCLOSED BY '"'
81 LINES TERMINATED BY '\r\n'
82 IGNORE 1 ROWS;
83
84 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/users_ca.csv'
85 INTO TABLE user_data
86 FIELDS TERMINATED BY ','
87 ENCLOSED BY '"'
88 LINES TERMINATED BY '\r\n'
89 IGNORE 1 ROWS;
90
91 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/transactions.csv'
92 INTO TABLE transactions
93 FIELDS TERMINATED BY ';'
94 ENCLOSED BY '"'
95 LINES TERMINATED BY '\r\n'
96 IGNORE 1 ROWS;
97
98 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/products.csv'
99 INTO TABLE products
100 FIELDS TERMINATED BY ','
101 ENCLOSED BY '"'
102 IGNORE 1 ROWS;
103
104 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/credit_cards.csv'
105 INTO TABLE credit_cards
106 FIELDS TERMINATED BY ','
107 ENCLOSED BY '"'
108 IGNORE 1 ROWS;
109
110 • LOAD DATA LOCAL INFILE 'C:/Users/gbrlc/Downloads/companies.csv'
111 INTO TABLE companies
112 FIELDS TERMINATED BY ','
113 ENCLOSED BY '"'
114 LINES TERMINATED BY '\r\n'
115 IGNORE 1 ROWS;
```

## Creacion de constraints/FK

```

120 • ALTER TABLE credit_cards
121     ADD CONSTRAINT fk_credit_cards_user_id FOREIGN KEY (user_id)
122     REFERENCES user_data(id);
123
124 • ALTER TABLE transactions
125     ADD CONSTRAINT fk_transactions_product_ids FOREIGN KEY (product_ids)
126     REFERENCES products(id);
127
128     -- Entre transactions e credit_cards
129 • ALTER TABLE transactions
130     ADD CONSTRAINT fk_transactions_card_id FOREIGN KEY (card_id)
131     REFERENCES credit_cards(id);
132     -- Relacionamento entre active_cards e credit_cards
133 • ALTER TABLE active_cards
134     ADD CONSTRAINT fk_active_cards_card_id FOREIGN KEY (card_id)
135     REFERENCES credit_cards(id);
136     -- Entre transactions e user_data
137 • ALTER TABLE transactions
138     ADD CONSTRAINT fk_transactions_user_id FOREIGN KEY (user_id)
139     REFERENCES user_data(id);

```

---

```

140     -- Entre transactions e companies
141 • ALTER TABLE transactions
142     ADD CONSTRAINT fk_transactions_business_id FOREIGN KEY (business_id)
143     REFERENCES companies(company_id);
144
145     -- Creo una tabla intermediaria porque muchos productos pueden estar en muchas transacciones,
146     -- y muchas transacciones pueden comprar muchos productos.
147
148 • create table products_transactions (
149     product_id varchar(10),
150     transaction_id varchar(10));
151     -- Entre transactions_products y products
152 • ALTER TABLE products_transactions
153     ADD CONSTRAINT fk_products_transactions_product_id FOREIGN KEY (product_id)
154     REFERENCES products(id);
155     -- Entre transactions_products y transactions
156 • ALTER TABLE products_transactions
157     ADD CONSTRAINT fk_products_transactions_transaction_id FOREIGN KEY (transaction_id)
158     REFERENCES transactions(id);
159

```

## Exercici 1

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

```

170 • SELECT
171     id, name, surname, country,
172     (SELECT COUNT(*) FROM transactions t WHERE t.user_id = u.id) AS transactions
173 FROM user_data u
174 WHERE (SELECT COUNT(*) FROM transactions t WHERE t.user_id = u.id) > 30
175 ORDER BY transactions DESC;
176
177

```

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

	id	name	surname	country	transactions
▶	272	Hedwig	Gilbert	Canada	76
	267	Ocean	Nelson	Canada	52
	275	Kenyon	Hartman	Canada	48
	92	Lynn	Riddle	United States	39

Result 7 x

Output

Action Output

#	Time	Action	Message
✓ 1	16:46:26	SELECT id, name, surname, country, (SELECT COUNT(*) FROM transactions t WHERE t.user_id = u.id) ...	4 row(s) returned

## Con join:

```

129 • SELECT
130     u.id AS id,
131     u.name AS name,
132     u.surname AS surname,
133     u.country AS country,
134     COUNT(user_id) AS transactions
135 FROM
136     transactions t
137     LEFT JOIN
138     user_data u ON t.user_id = u.id
139 GROUP BY id, name, surname, country
140 HAVING transactions > 30
141 ORDER BY transactions DESC;

```

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

	id	name	surname	country	transactions
▶	272	Hedwig	Gilbert	Canada	76
	267	Ocean	Nelson	Canada	52
	275	Kenyon	Hartman	Canada	48
	92	Lynn	Riddle	United States	39

Result 5 x

Output

Action Output

#	Time	Action	Message
✓ 1	10:01:38	SELECT u.id AS id, u.name AS name, u.surname AS surname, u.country AS country, COUNT(us...	4 row(s) returned

## Exercici 2

**Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.**

```

146 • select
147     c.company_id,
148     c.company_name,
149     cc.iban,
150     round(avg(amount), 2) as avg_amount
151 from transactions t
152 join companies c on c.company_id = t.business_id
153 join credit_cards cc on cc.id = t.card_id
154 where c.company_name = 'Donec Ltd'
155 group by c.company_id, iban, c.company_name;

```

## 2 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:

```

CREATE TABLE active_cards AS SELECT card_id, timestamp, declined FROM
transactions;

```

```

with tarjetas_activas as(
    select card_id, declined,
    row_number () over (partition by card_id order by timestamp desc) as partition_time
    from active_cards)

```

## Exercici 1

Quantes targetes estan actives?

```

170 with tarjetas_activas as(
171     select card_id, declined,
172     row_number () over (partition by card_id order by timestamp desc) as partition_time
173     from active_cards)
174
175 Select card_id as tarjeta,
176 (case
177     when sum(declined) < 3 then 'Activa'
178     else 'Inactiva'
179 end) as status
180 from tarjetas_activas
181 group by card_id
182 having status = 'activa';
183

```

tarjeta	status
CcU-2938	Activa
CcU-2945	Activa
CcU-2952	Activa
CcU-2959	Activa
CcU-2966	Activa
CcU-2973	Activa
CcU-2980	Activa

Result 8 x

Output

Action Output

#	Time	Action	Message
1	11:26:05	with tarjetas_activas as(select card_id, declined, row_number () over (partition by card_id order by timestamp d...	275 row(s) returned

### 3 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:

```

-- ANTES DE LA CONSULTA TENEMOS QUE SEPARAR LOS IDS DE PRODUCTOS, HACER COMO UN "CONCATENACION INVERTIDA"
-- VAMOS A CREAR UNA TABLA TEMPORAL LLAMADA SPLIT_PRODUCTS PARA LOS RESULTADOS DEL PROCESO DE DIVISIÓN DE ID DE PRODUCTOS POR FILAS
-- "RECURSIVE" PERMITE QUE LA TABLA TEMPORAL SE LLAME A SÍ MISMA VARIAS VECES PARA PROCESAR PARTES DE LA COLUMNA REPETIDAMENTE
WITH RECURSIVE split_products AS (
    SELECT
        id,
        TRIM(SUBSTRING_INDEX(product_ids, ',', 1)) AS product_id,
        -- ^ EXTRAE EL PRIMER VALOR DE LA LISTA DE IDS DE PRODUCTO. EN "71, 1, 19" EL VALOR EXTRAÍDO SERÁ 71. "TRIM" ELIMINA LOS ESPACIOS SOBANTES.
        TRIM(SUBSTRING_INDEX(product_ids, ',', -1)) AS remaining_ids,
        -- ^ EXTRAE EL RESTO DE LA LISTA DESPUÉS DE ELIMINAR EL PRIMER VALOR. DE "71, 1, 19" EL RESTO SERÁ 1, 19. "TRIM" NOVAMENTE.
        1 AS level
    FROM transactions t
    UNION ALL
    -- ^ COMBINA EL PRIMER PASO ARRIBA CON LOS SIGUIENTES PASOS DE RECURSIÓN, ES DECIR: "DESPUÉS DE HACER EL PRIMER PASO,
    -- ^ SIGUE LLAMÁNDOSE A TI MISMO PARA PROCESAR EL RESTO DE LA LISTA HASTA QUE NO HAYA MÁS COMAS".
    SELECT
        id,
        TRIM(SUBSTRING_INDEX(remaining_ids, ',', 1)) AS product_id,
        TRIM(SUBSTRING_INDEX(remaining_ids, ',', -1)) AS remaining_ids,
        level + 1
    -- EL +1 ACTUALIZA EL CONTADOR DE LA ITERACION PARA INDICAR QUE ESTAMOS A UN NIVEL MÁS

    FROM split_products
    WHERE remaining_ids LIKE '%,%'
    -- UNA CONDICIÓN PARA QUE CONTINÚE LA RECURSIÓN QUE SÓLO SE PRODUCE MIENTRAS REMAINING_IDS AÚN CONTIENE UNA COMA
)

```

EL METODO ARRIBA NO HABIA CREADO UNA TABLA CON CLAVE COMPUESTA DE ID DE TRANSACCION CON PRODUCT ID. CONSULTÉ ALEIX PARA EL METODO CON JSON PARA TENERLA.

```

258 -- EL METODO ARRIBA NO HABIA CREADO UNA TABLA CON CLAVE COMPUESTA DE ID DE TRANSACCION CON PRODUCT ID.
259 -- CONSULTÉ ALEIX PARA EL METODO CON JSON PARA TENERLA.
260 with split_products as (
261   select id, productes.product_id
262   from transactions
263   join json_table (concat("[", product_ids, "]"),
264     "$[*]" columns (product_id varchar(100) path "$")
265   ) as productes)
266   select * from split_products;
267 • SELECT product_id Product, count(product_id) 'Total sales'
268   FROM split_products
269   group by product_id;
270
271 • create table trasac_products as(
272   select id, productes.product_id
273   from transactions
274   join json_table (concat("[", product_ids, "]"),
275     "$[*]" columns (product_id varchar(100) path "$")
276   ) as productes);
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Result Grid		
Filter Rows:		
Edit:		
Export/Import:		
Wrap Cell Content:		
Fetch rows:		
trasac_products 2 x		
Output		
Action Output		
#	Time	Action
1	12:47:08	SELECT * FROM sprint4.trasac_products
Message		
1457 row(s) returned		

## Exercici 1



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

```
218 SELECT product_id Product, count(product_id) 'Total sales'
219 FROM split_products
220 group by product_id;
221
222
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Product	Total sales
▶	71	18
	47	26
	89	26
	43	29
	67	24
	29	14
	19	20
	83	22
	7	30

Result 14 x

Output

Action Output

#	Time	Action	Message
✓ 1	13:43:31	WITH RECURSIVE split_products AS ( SELECT id, TRIM(SUBSTRING_INDEX(product_ids, ',', 1...)	26 row(s) returned