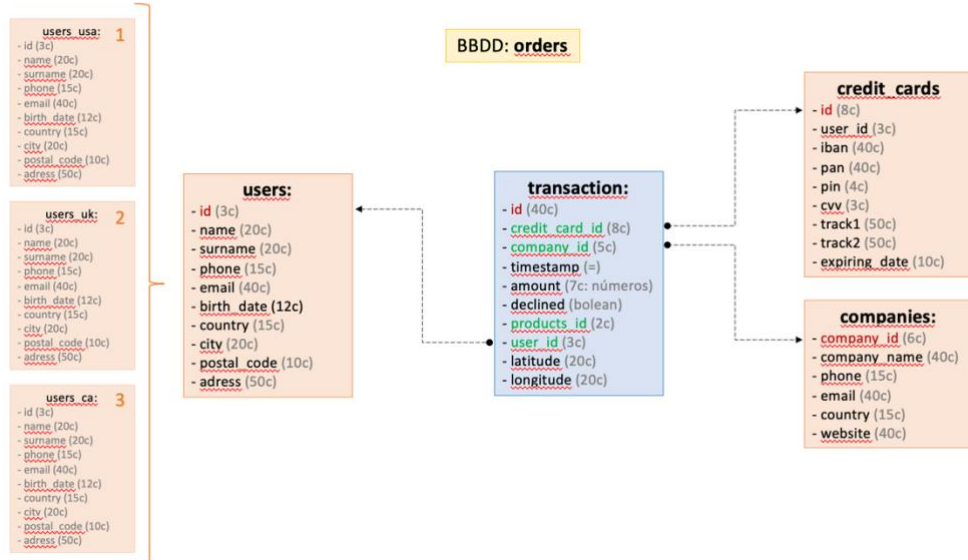


Nivell 1

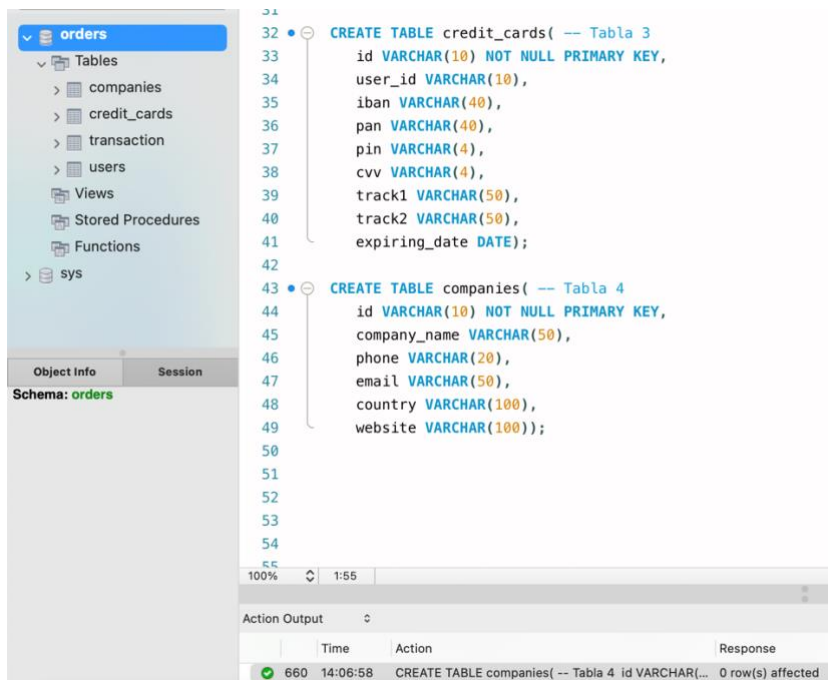
Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:



- Creamos la BBDD y las tablas:

```
SCHMAS
Filter objects
orders
sys

1 -- Nivel 1: Dissenya una base de dades amb un esquema d'estrella:
2
3 • CREATE DATABASE IF NOT EXISTS orders -- Creo la base de datos
4 DEFAULT CHARACTER SET = 'utf8mb4' DEFAULT COLLATE 'utf8mb4_general_ci';
5
6 • use orders; -- y la selecciono para comenzar a crear las tablas que contendrá
7
8
9
10
11
12
13
14
15
16
17
18
19 • CREATE TABLE transaction( -- Tabla 1
20 id VARCHAR(40) NOT NULL PRIMARY KEY, -- a cada tabla le asigno su Primary Key
21 credit_card_id VARCHAR(10),
22 company_id VARCHAR(10),
23 timestamp TIMESTAMP,
24 amount DECIMAL(10,2),
25 declined BOOLEAN,
26 products_id VARCHAR(10),
27 user_id VARCHAR(10),
28 latitude VARCHAR(20),
29 longitude VARCHAR(20));
30
31 • CREATE TABLE users( -- Tabla 2
32 id VARCHAR(10) NOT NULL PRIMARY KEY,
33 name VARCHAR(20),
34 surname VARCHAR(20),
35 phone VARCHAR(20),
36 email VARCHAR(40),
37 birth_date DATE,
38 country VARCHAR(100),
39 city VARCHAR(100),
40 postal_code VARCHAR(10),
41 address VARCHAR(100));
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2
```



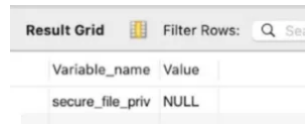
En la tabla 'transaction' elegí la mayoría de datos en formato VARCHAR (con una amplitud dependiente de los caracteres que tuviera nuestro campo) porque la mayoría de campos contenían números y letras o caracteres. En el caso del campo 'timestamp' como su nombre indica su tipo de dato ha de ser TIMESTAMP, 'amount' es decimal porque se trata de dinero y declined es un booleano porque solo acepta valores de 0 o 1.

En la tabla 'users' todos los campos son VARCHAR por lo mismo que en la anterior, excepto el campo 'birth_date' que como su nombre indica es una fecha, ya que así será más fácil de trabajar y manipular en el futuro. Nombre, apellido y país los dejo como VARCHAR en lugar de texto por si existiera alguno de ellos que contuviera un carácter especial no contemplado en otros países (por ejemplo la 'ñ' de 'España' que no existe en los teclados de países de habla inglesa).

En el caso de la tabla 'credit_card', al igual que en 'users' he definido todos los campos como VARCHAR exceptuando el campo 'expiring_date' que es una fecha. En este caso tanto 'cvv' como 'pin' podrían ser considerados INT (enteros) pero entonces tendríamos problemas si existiera algún código que fuera, por ejemplo: 0001 porque lo detectaría como 1. El IBAN (International Bank Account Number), utilizado para identificar de forma estandarizada cualquier cuenta sea del país que sea, es una serie alfanumérica formada por un máximo de 34 caracteres dependiendo del país, por eso lo he definido como VARCHAR (40) para darle algo de margen. El CVV o código de verificación de la tarjeta es un código de tres o cuatro números utilizado para verificar que el usuario de la compra es el poseedor de la tarjeta por eso lo he definido como VARCHAR (4) porque es el máximo número de caracteres que puede aceptar. El PIN es el código de seguridad de la tarjeta que he definido como VARCHAR (4) porque suele ser de 4 números.

Por último, en la tabla 'companies' todos los datos son de tipo VARCHAR por lo anteriormente explicado.

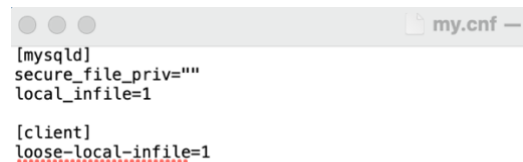
- Al intentar cargar los csv para añadirlos a las tablas me daba error ya que la exportación e importación de datos estaba desactivada porque las versiones de MySQL anteriores a la 8.0 no tienen un archivo llamado my.cnf, y por eso al ejecutar: SHOW VARIABLES LIKE "secure_file_priv" Nos devuelve el valor NULL.



Variable_name	Value
secure_file_priv	NULL

Para solucionar esto:

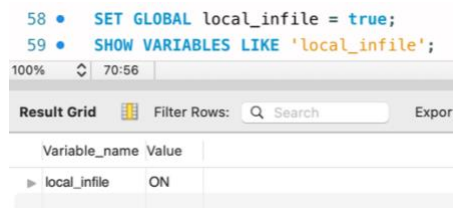
1º Creé el archivo /etc/my.cnf que no existía y le añadí los siguientes valores:



```
[mysqld]
secure_file_priv=""
local_infile=1

[client]
loose-local-infile=1
```

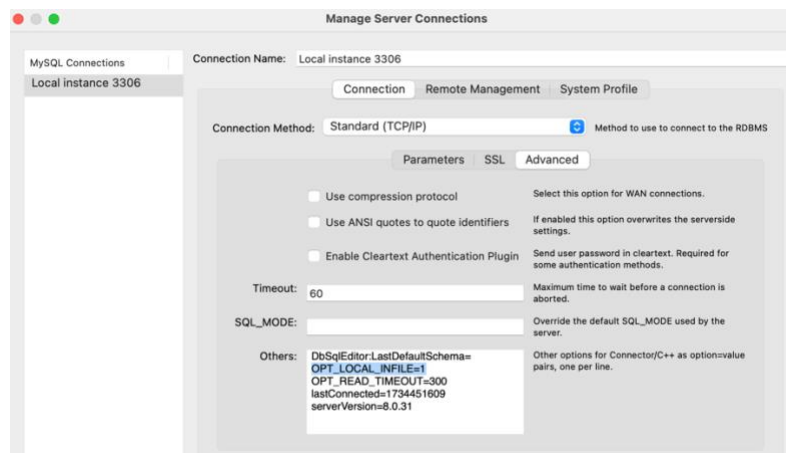
2º Me aseguré que la variable 'local_infile' estuviera activa igualando su valor a true:



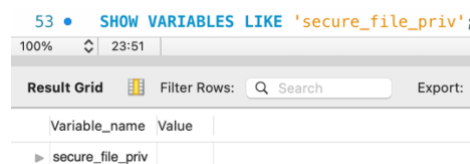
```
58 • SET GLOBAL local_infile = true;
59 • SHOW VARIABLES LIKE 'local_infile';
```

Variable_name	Value
local_infile	ON

3º Como continuaba dándome error, edité la conexión en la subpestaña 'Advanced' de la casilla 'Others' añadiendo: OPT_LOCAL_INFILE=1



Y entonces funcionó:



```
53 • SHOW VARIABLES LIKE 'secure_file_priv';
```

Variable_name	Value
secure_file_priv	

- Ahora sí, cargo los csv para añadirlos a las tablas:

Tabla 1: 'transaction':

Al intentar cargar los datos el campo 'timestamp' me salen valores a 0, por eso cambio el campo de TIMESTAMP a VARCHAR(50) en la tabla creada por mí, importo los datos de los archivos csv y modifíco el formato de los datos del campo para volver a cambiar la variable de VARCHAR a TIMESTAMP:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'orders' database is expanded, showing the 'transaction' table. The table's columns are listed: id (varchar(40) PK), credit_card_id (varchar(10)), company_id (varchar(10)), timestamp (timestamp), amount (decimal(10,2)), declined (tinyint(1)), products_id (varchar(10)), user_id (varchar(10)), latitude (varchar(20)), and longitude (varchar(20)).

On the right, a SQL script is shown with the following steps:

```

61 -- Al intentar cargar los datos el campo 'timestamp' (de 'transactions') tengo valores 0, cambio el tipo a VARCHAR(50) en la tabla:
62 ALTER TABLE orders.transaction
63 CHANGE COLUMN timestamp
64 timestamp VARCHAR(50);
65
66 -- Ahora importo los datos desde 'transaction.csv' a la tabla creada:
67 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/ISprints/54/Datos/transactions.csv'
68 INTO TABLE orders.transaction
69 FIELDS TERMINATED BY ';' -- Delimitador de las columnas
70 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)
71 IGNORE 1 ROWS; -- Ignorar la primera línea (encabezados)
72
73 -- Y modifíco el formato del campo 'timestamp' para volver a cambiar la variable de VARCHAR a TIMESTAMP
74 SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura, porque sino no me deja realizar el cambio
75
76 UPDATE orders.transaction
77 SET timestamp = DATE_FORMAT(STR_TO_DATE(timestamp, '%d/%m/%y %H:%i'), '%y-%m-%d %H:%i'); -- str_to_date cambia las fechas de formato
78
79 SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
80
81 ALTER TABLE orders.transaction -- y vuelvo a cambiar el tipo de variable
82 CHANGE COLUMN timestamp
83 timestamp TIMESTAMP;
84

```

The 'Action Output' pane at the bottom shows the execution of the script, indicating that 587 rows were affected and no warnings or duplicates were found.

Me sale un WARNING al importar 'transaction':

The 'Action Output' pane shows a warning message (116) at 11:52:54. The message states: "LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2...' 587 row(s) affected, 102 warning(s): 1265 Data truncated for column 'products_id'".

Pero al visualizar la tabla todo está correcto y como menciona la columna 'products_id' deduzco que es porque esta tiene varios valores separados por comas en algunas de sus filas.

14 • SELECT * FROM orders.transaction;

The screenshot shows the 'transaction' table data in the 'Result Grid' pane. The table has 11 columns: id, credit_card_id, company_id, timestamp, amount, declined, products_id, user_id, latitude, and longitude. The data is displayed in a grid format, showing several rows of transaction records.

Below the grid, the 'Action Output' pane shows the execution of the query 'SELECT * FROM orders.transaction', indicating that 587 row(s) were returned.

Tabla 2: 'users':

Al intentar cargar los datos el campo 'birth_date' me salen valores a 0, por eso cambio el campo de DATE a VARCHAR(20) en la tabla creada por mí. También modifico el campo 'id' de VARCHAR a INT porque me daba errores de ordenación al intentar cargar los datos.

The screenshot shows the 'users' table structure in SQL Server Enterprise Manager. The columns are: id (int PK), name (varchar(20)), surname (varchar(20)), phone (varchar(20)), email (varchar(40)), birth_date (varchar(20)), country (varchar(100)), city (varchar(100)), postal_code (varchar(10)), and adress (varchar(100)).

The SQL commands executed are:

```

84
85 -- Al intentar cargar los datos el campo 'birth_date' (de 'users') me salen valores a 0, cambio el tipo de DATE a VARCHAR(20)
86 ALTER TABLE orders.users
87 CHANGE COLUMN birth_date
88 birth_date VARCHAR(20);
89
90 -- Tambien cambio el campo 'id' de VARCHAR a INT porque me daba errores de ordenación
91 ALTER TABLE orders.users
92 CHANGE COLUMN id
93 id INT;
94
100% 1:94

```

The Action Output shows the execution of the commands:

Time	Action	Response
133 12:14:30	ALTER TABLE orders.users CHANGE COLUMN id INT	150 row(s) affected Records: 150 Duplicates: 0 Warnings: 0

Importo los datos de los archivos csv:

The screenshot shows the 'users' table structure in SQL Server Enterprise Manager. The columns are: id (int PK), name (varchar(20)), surname (varchar(20)), phone (varchar(20)), email (varchar(40)), birth_date (varchar(20)), country (varchar(100)), city (varchar(100)), postal_code (varchar(10)), and adress (varchar(100)).

The SQL commands executed are:

```

95
96
97 -- Luego importo los datos desde los diferentes archivos de csv (users_usa, users_uk y users_ca) a la tabla creada:
98 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints!/S4/Datos/users_usa.csv'
99 INTO TABLE orders.users
100 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
101 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección
102 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)
103 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
104
105 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints!/S4/Datos/users_uk.csv'
106 INTO TABLE orders.users
107 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
108 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección
109 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)
110 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
111
112 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints!/S4/Datos/users_ca.csv'
113 INTO TABLE orders.users
114 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
115 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección
116 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)
117 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
118
100% 1:96

```

The Action Output shows the execution of the commands:

Time	Action	Response
136 12:17:43	LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

Modifico el formato de los datos del campo para volver a cambiar la variable de VARCHAR a DATE y visualizo la tabla:

The screenshot shows the 'users' table structure in SQL Server Enterprise Manager. The columns are: id (int PK), name (varchar(20)), surname (varchar(20)), phone (varchar(20)), email (varchar(40)), birth_date (date), country (varchar(100)), city (varchar(100)), postal_code (varchar(10)), and adress (varchar(100)).

The SQL commands executed are:

```

118 SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura, porque sino no me deja realizar el cambio
119
120 UPDATE orders.users
121 SET birth_date = DATE_FORMAT(STR_TO_DATE(birth_date, '%b %d, %Y'), '%Y-%m-%d') -- str_to_date cambia las fechas de formato
122 WHERE users.id IS NOT null;
123
124 SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
125
126 ALTER TABLE orders.users -- y vuelvo a cambiar el tipo de variable para 'birth date' a DATE
127 CHANGE COLUMN birth_date
128 birth_date DATE;
129
100% 4:126

```

The Action Output shows the execution of the commands:

Time	Action	Response
141 12:23:56	ALTER TABLE orders.users -- y vuelvo a cambiar el tipo de...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

15 • **SELECT * FROM orders.users;**

100% 1:7

Result Grid Filter Rows: Search Edit: Export/Import:

	id	name	surname	phone	email	birth_date	country	city	postal_code	adress
▶ 1	Zeus	Gamble		1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mconnell		(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave
3	Ciaran	Harrison		(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford		1-411-740-3269	omare.egestas@icloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce		1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson		(718) 288-8020	gravida.nunc.sed@yahoo.ca	1989-10-15	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez		(817) 689-0478	eget@outlook.ca	1981-12-04	United States	Hillsboro	29874	8627 Malesuada Rd.

users 5

Action Output

	Time	Action	Response
✓ 142	12:26:34	SELECT * FROM orders.users	275 row(s) returned

Tabla 3: 'credit_cards':

Primero cambio el campo 'user_id' a INT para que coincida con el campo 'id' de la tabla 'users' y el campo 'expiring_date' a VARCHAR para que acepte los datos y luego lo volveré a cambiar a DATE:

Object Info	Session
Table: credit_cards	
Columns:	
id	varchar(10) PK
user_id	int
iban	varchar(40)
pan	varchar(40)
pin	varchar(4)
cvv	varchar(4)
track1	varchar(50)
track2	varchar(50)
expiring_date	date

```

145 -- Cambio el campo 'user_id' (de 'credit_cards') de VARCHAR a INT para que sea igual al campo 'id' de la tabla 'users':
146 • ALTER TABLE orders.credit_cards
147   CHANGE COLUMN user_id
148   user_id INT;
149
150 -- También cambio el tipo de dato de 'expiring_date' de DATE a VARCHAR(20) en la tabla:
151 • ALTER TABLE orders.credit_cards
152   CHANGE COLUMN expiring_date
153   expiring_date VARCHAR(20);

```

100% 1:144

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 677	14:10:04	ALTER TABLE orders.credit_cards CHANGE COLUM...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.045 sec

Importo los datos del archivo csv, cambio el formato de fecha de 'expiring_date', vuelvo a cambiar el tipo de variable a DATE y visualizo mi tabla:

```

144 -- Importo los datos desde 'credit_cards.csv' a la tabla creada:
145 • LOAD DATA LOCAL INFILE '/Users/Liss/Desktop/ItAcademy/2-Especialización/iSprints!/54/Datos/credit_cards.csv'
146   INTO TABLE orders.credit_cards
147   FIELDS TERMINATED BY ',' -- Delimitador de las columnas
148   LINES TERMINATED BY '\n' -- Fin de línea
149   IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
150
151 -- Y modifico el formato del campo 'expiring_date' para volver a cambiar la variable de VARCHAR a DATE
152 • SET SQL_SAFE_UPDATES=0; -- para desabilitar la opción de actualización segura
153
154 • UPDATE orders.credit_cards
155   SET expiring_date = DATE_FORMAT(STR_TO_DATE(expiring_date, '%m/%d/%Y'), '%Y-%m-%d') -- str_to_date cambia las fechas de formato
156   WHERE credit_cards.id IS NOT null;
157
158 • SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
159
160 • ALTER TABLE orders.credit_cards -- y vuelvo a cambiar el tipo de variable de 'expiring_date' a DATE
161   CHANGE COLUMN expiring_date
162   expiring_date DATE;
163

```

100% 1:140

Action Output

	Time	Action	Response
✓ 176	13:06:08	ALTER TABLE orders.credit_cards -- y vuelvo a cambiar el t...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

15

SELECT * FROM orders.credit_cards;

100%

1:5

Result Grid

Filter Rows: Search Edit: Export/Import:

	id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
▶	CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646*WovsxejDpwiev*86041...	%B7653863056044187=800716333673	2022-10-30
▶	CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%B4621311609958661*UftuyfsSeimxn*061062...	%B4149568437843501=510714033071	2023-08-24
▶	CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501*CddytycUxwfdq*59079...	%B6778580257827162=6906859740077	2021-06-29
▶	CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320*XocddyjBkeacd*090162...	%B42461544489281853=280522391678	2023-02-24
▶	CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223*UhlgsuFbmwgj*72022...	%B2318571115599881=890821578475	2024-10-29
▶	CcU-2973	270	PT87806228135092429456346	544 58654 54343 384	8760	887	%B4761405253275637*HjnmipoBlejrt*7108515...	%B7816169831446746=1310277279	2025-01-30
▶	CcU-2980	269	DE39241881883086277136	402400 7145845969	5075	596	%B7320483593870549*OokzqxrHpsaed*49017...	%B2474313962214151=041221913175	2022-07-24

credit_cards 7

Action Output

	Time	Action	Response
✓ 177	13:08:58	SELECT * FROM orders.credit_cards	275 row(s) returned

Tabla 4: 'companies':

Importo los datos del archivo csv y visualizo mi tabla:

Table: companies

Columns:

id

company_name

phone

email

country

website

varchar(10) PK

varchar(50)

varchar(20)

varchar(50)

varchar(100)

varchar(100)

142

-- Y también importo los datos desde 'companies.csv' a la tabla creada:

143

LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints/54/Datos/companies.csv'

144

INTO TABLE orders.companies

145

FIELDS TERMINATED BY ',' -- Delimitador de las columnas

146

LINES TERMINATED BY '\r\n' -- Fin de línea

147

IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)

148

149

100%

60:138

Action Output

Time

Action

Response

158

12:44:00

SELECT * FROM orders.credit_cards

0 row(s) returned

15

SELECT * FROM orders.companies;

100%

1:5

Result Grid

Filter Rows:

Search

Edit:

Export/Import:

id	company_name	phone	email	country	website
▶ b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.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

companies 2

Action Output

	Time	Action	Response
✔ 160	12:45:42	SELECT * FROM orders.companies	100 row(s) returned

- Añado las Foreign Keys que relacionaran las tablas entre ellas en este esquema de estrella:

Functions

> sys

Object Info

Session

Table: transaction

Columns:

id	varchar(40) PK
credit_card_id	varchar(10)
company_id	varchar(10)
timestamp	timestamp
amount	decimal(10,2)
declined	tinyint(1)
products_id	varchar(10)
user_id	int
latitude	varchar(20)
longitude	varchar(20)

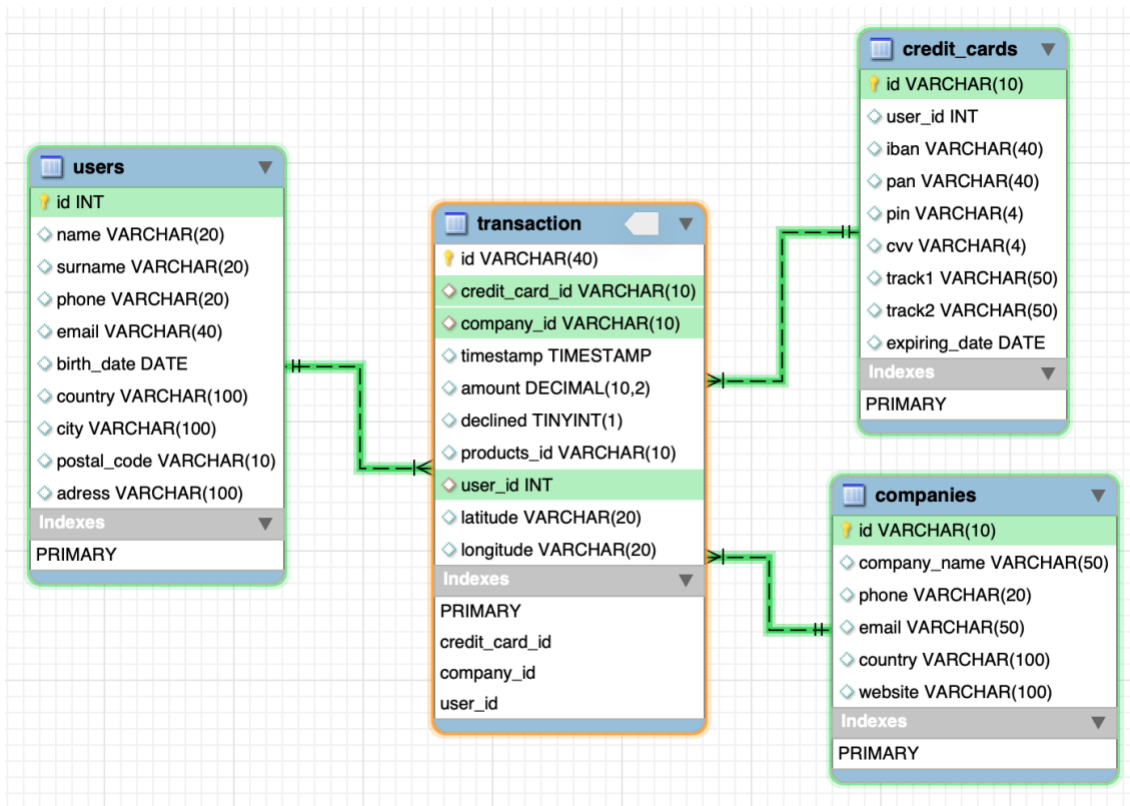
```
167 -- Ahora añado las Foreign Keys que relacionaran las tablas entre ellas en este esquema de estrella
168 • ALTER TABLE orders.transaction
169 ADD FOREIGN KEY (credit_card_id) REFERENCES credit_cards(id);
170
171 • ALTER TABLE orders.transaction
172 ADD FOREIGN KEY (company_id) REFERENCES companies(id);
173
174 • ALTER TABLE orders.transaction
175 CHANGE COLUMN user_id
176 user_id INT; -- cambio el campo 'user_id' a INT para que sea igual al campo 'id' de la tabla 'users'
177
178 • ALTER TABLE orders.transaction
179 ADD FOREIGN KEY (user_id) REFERENCES users(id);
```

100% 4:178

Action Output

	Time	Action	Response
✓	182 17:14:47	ALTER TABLE orders.transaction ADD FOREIGN KEY (user_id) REFERENCES users(id);	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

- Y así queda mi BBDD con esquema de estrella:



En el diagrama observamos que la tabla de hechos sería 'transaction' y las tablas de dimensiones 'companies', 'users' y 'credit_cards'. La relación entre las tres tablas se establece a través de las Primary Keys de las tablas de dimensiones (id de 'companies', id de 'users' e id de 'credit_cards') y las Foreign Keys de la tabla de hechos: company_id que se relaciona con id de 'company', user_id que se relaciona con la id de 'users' y credit_card_id que se relaciona con id de 'credit_card'.

*Aunque también había un archivo de csv que contenía datos de una tabla llamada 'products' por ahora no lo añadido a mi base de datos porque no lo usaré hasta el nivel 3.

Nivell 1

Exercici 1

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

```
185 • SELECT u.name AS nombre_usuario, u.surname AS apellido_usuario, COUNT(t.id) AS transacciones
186 FROM users AS u
187 JOIN transaction AS t
188 ON u.id = t.user_id
189 GROUP BY 1, 2
190 HAVING transacciones > 30;
```

100% 12:182

Result Grid Filter Rows: Search Export:

nombre_usuario	apellido_usuario	transacciones
Lynn	Riddle	39
Ocean	Nelson	52
Hedwig	Gilbert	76
Kenyon	Hartman	48

Result 84

Action Output

	Time	Action	Response
692	15:27:26	SELECT u.name AS nombre_usuario, u.surname AS...	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.

```
193 • SELECT c.company_name AS nombre_compañia, cc.id AS tarjeta_credito, cc.iban, ROUND(AVG(t.amount),2) AS importe
194 FROM credit_cards AS cc
195 JOIN transaction AS t
196 JOIN companies AS c
197 WHERE cc.id = t.credit_card_id
198 AND t.company_id IN (SELECT c.id
199                      FROM companies
200                      WHERE c.company_name = 'Donec Ltd')
201 GROUP BY 1, 2;
```

100% 1:203

Result Grid Filter Rows: Search Export:

nombre_compañia	tarjeta_credito	iban	importe
Donec Ltd	CcU-2973	PT87806228135092429456346	203.72

Result 14

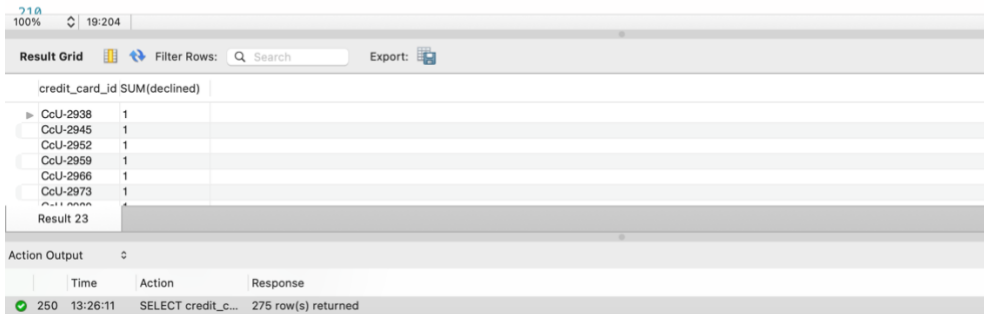
Action Output

	Time	Action	Response
209	19:51:44	SELECT c.company_name AS nombre_compañia, cc.id AS t...	1 row(s) returned

Nivell 2

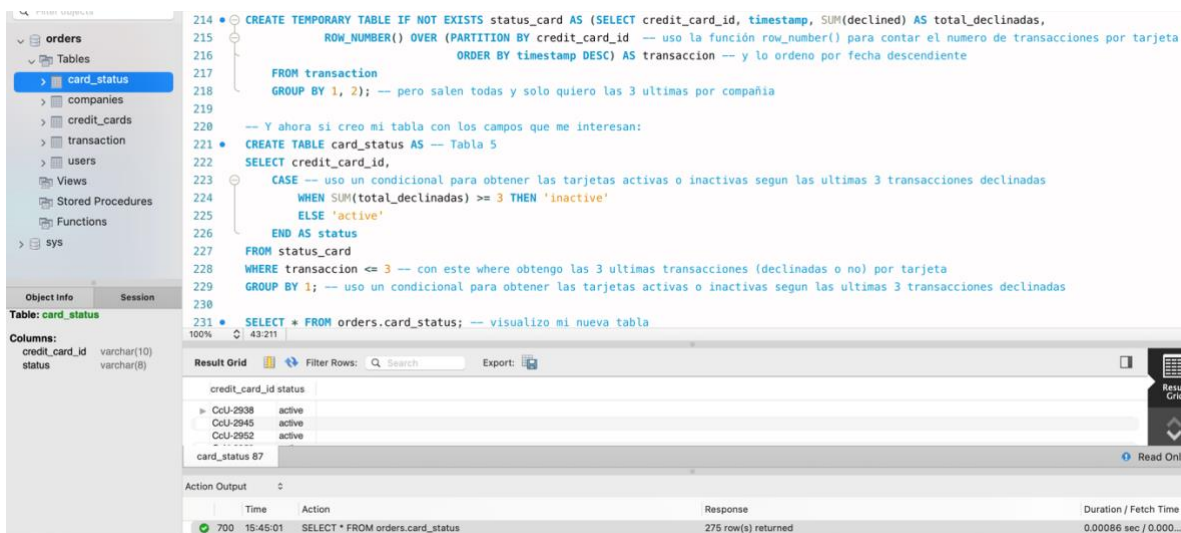
Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

```
206 • SELECT credit_card_id, SUM(declined) -- compruebo cuantas tarjetas han sido declinadas más de una vez en general
207 FROM transaction
208 GROUP BY 1;
209 -- Ninguna tarjeta ha sido declinada más de una vez
```



credit_card_id	SUM(declined)
CcU-2938	1
CcU-2945	1
CcU-2952	1
CcU-2959	1
CcU-2966	1
CcU-2973	1

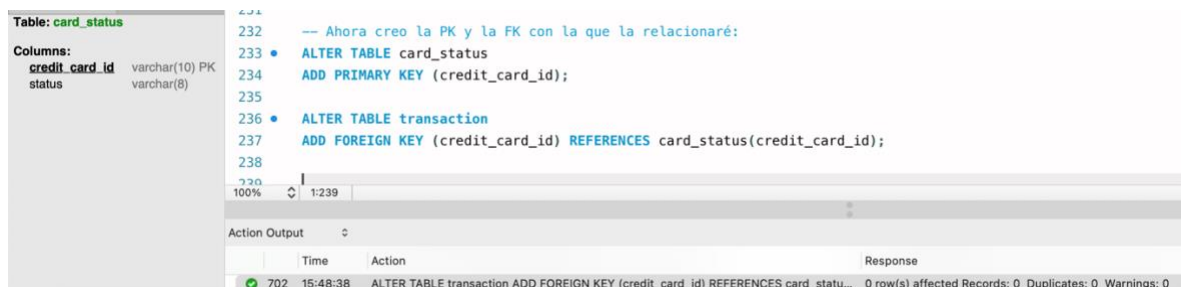
- Primero creo una tabla temporal, luego creo la tabla con los campos que me interesan y la visualizo:



```
214 • CREATE TEMPORARY TABLE IF NOT EXISTS status_card AS (SELECT credit_card_id, timestamp, SUM(declined) AS total_declinadas,
215 ROW_NUMBER() OVER (PARTITION BY credit_card_id -- uso la función row_number() para contar el numero de transacciones por tarjeta
216 ORDER BY timestamp DESC) AS transaccion -- y lo ordeno por fecha descendiente
217 FROM transaction
218 GROUP BY 1, 2); -- pero salen todas y solo quiero las 3 ultimas por compañía
219
220 -- Y ahora si creo mi tabla con los campos que me interesan:
221 • CREATE TABLE card_status AS -- Tabla 5
222 SELECT credit_card_id,
223 CASE -- uso un condicional para obtener las tarjetas activas o inactivas segun las ultimas 3 transacciones declinadas
224 WHEN SUM(total_declinadas) >= 3 THEN 'inactive'
225 ELSE 'active'
226 END AS status
227 FROM status_card
228 WHERE transaccion <= 3 -- con este where obtengo las 3 ultimas transacciones (declinadas o no) por tarjeta
229 GROUP BY 1; -- uso un condicional para obtener las tarjetas activas o inactivas segun las ultimas 3 transacciones declinadas
230
231 • SELECT * FROM orders.card_status; -- visualizo mi nueva tabla
```

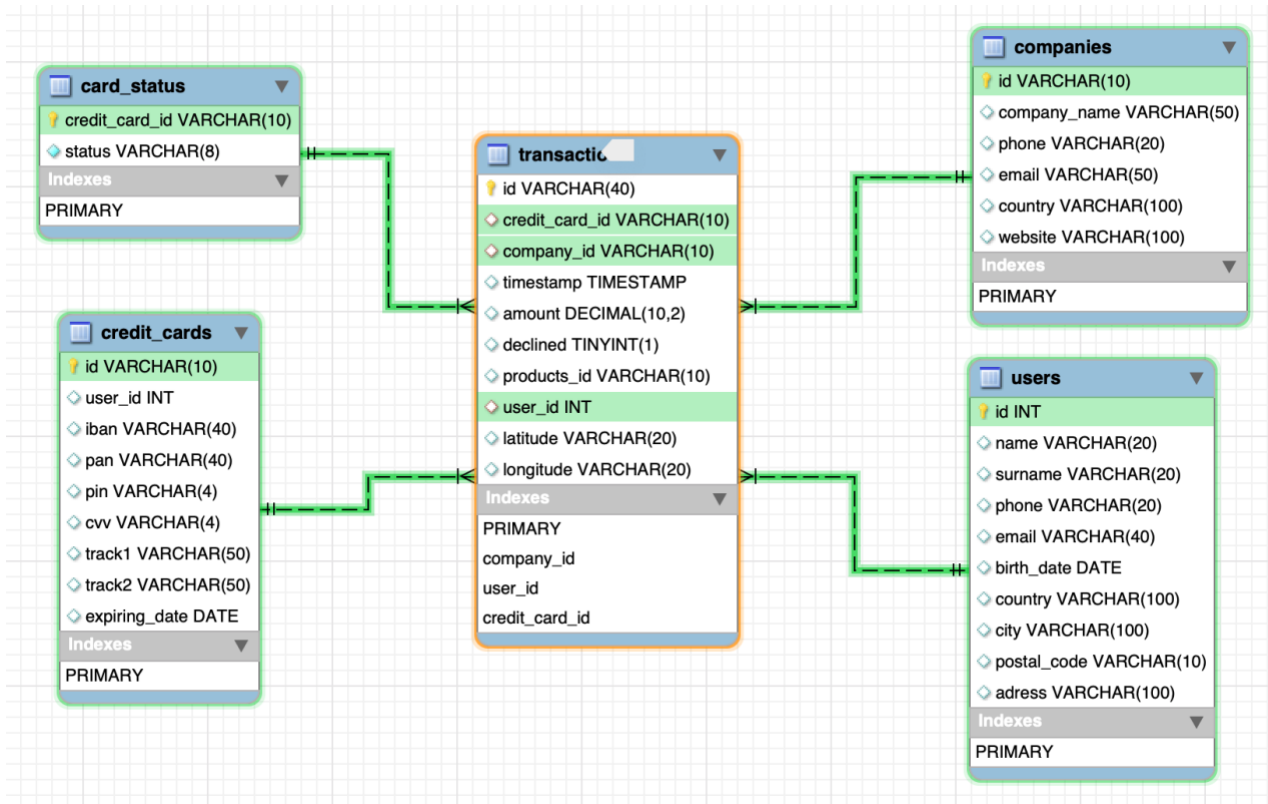
credit_card_id	status
CcU-2938	active
CcU-2945	active
CcU-2952	active

- Por último, añado su PK y la relaciono con una FK de la tabla 'transaction':



```
232 -- Ahora creo la PK y la FK con la que la relacionaré:
233 • ALTER TABLE card_status
234 ADD PRIMARY KEY (credit_card_id);
235
236 • ALTER TABLE transaction
237 ADD FOREIGN KEY (credit_card_id) REFERENCES card_status(credit_card_id);
238
```

- Y así queda nuestra BBDD ahora:



La nueva tabla 'card_status' nos queda relacionada con la tabla de hechos 'transaction' a través de su PK credit_card_id con la FK (del mismo nombre) de la tabla 'transaction', de manera que si se actualizan los datos de las transacciones se actualice el estado de las tarjetas activas o inactivas.

Nivell 2

Exercici 1

Quantes targetes estan actives?

- Todas las tarjetas están activas:

```
259 • SELECT COUNT(credit_card_id) AS tarjetas_activas
260 FROM card_status
261 WHERE status = 'active'; -- todas las tarjetas están activas
262
```

The screenshot shows a database query interface. At the top, the query is displayed: `SELECT COUNT(credit_card_id) AS tarjetas_activas FROM card_status WHERE status = 'active';`. Below the query, the 'Result Grid' shows a single row with the value 275. The 'Action Output' section shows the query execution details: '260 21:52:32 SELECT COUNT(...) 1 row(s) returned'.

Time	Action	Response
260 21:52:32	SELECT COUNT(...)	1 row(s) returned

- Lo comprobamos mirando que no haya tarjetas inactivas:

```
263 • SELECT COUNT(credit_card_id) AS tarjetas_inactivas
264 FROM card_status
265 WHERE status = 'inactive'; -- comprobamos mirando las tarjetas inactivas
266
```

The screenshot shows a database query interface. At the top, the query is displayed: `SELECT COUNT(credit_card_id) AS tarjetas_inactivas FROM card_status WHERE status = 'inactive';`. Below the query, the 'Result Grid' shows a single row with the value 0. The 'Action Output' section shows the query execution details: '261 21:54:56 SELECT COUNT(...) 1 row(s) returned'.

Time	Action	Response
261 21:54:56	SELECT COUNT(...)	1 row(s) returned

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:

- Primero creamos la tabla 'products' y le añadimos los datos del csv con el mismo nombre:

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The 'products' table is selected under the 'orders' schema. The main pane displays SQL code for creating the table and loading data from a CSV file. The 'Result Grid' shows the first four rows of the loaded data.

```
274 CREATE TABLE products( -- Tabla 6
275     id VARCHAR(10),
276     product_name VARCHAR(40),
277     price VARCHAR(10),
278     colour VARCHAR(10),
279     weight DECIMAL(10,2),
280     warehouse_id VARCHAR(10));
281
282 -- cargamos los datos de la tabla desde el archivo csv:
283 LOAD DATA LOCAL INFILE '/Users/Liss/Desktop/ItAcademy/2-Especialización/iSprints!/S4/Datos/products.csv'
284 INTO TABLE orders.products
285 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
286 LINES TERMINATED BY '\n' -- Fin de línea
287 IGNORE 1 ROWS; -- Ignorar la primera línea (encabezados)
288
289 -- visualizamos la tabla:
290 SELECT * FROM orders.products;
```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	\$161.11	#7c7c7c	1.00	WH-4
2	Tarly Stark	\$9.24	#919191	2.00	WH-3
3	duel tourney Lannister	\$171.13	#d8d8d8	1.50	WH-2
4	warden south duel	\$71.89	#111111	3.00	WH-1

- Elimino el símbolo de dólar de la columna 'price' de 'products' para convertirla en decimal y trabajar mejor con sus datos ya que se trata de dinero:

The screenshot shows the same database management interface. The SQL code pane now contains commands to update the 'price' column by removing dollar signs and changing its data type to DECIMAL. The 'Result Grid' shows the updated data, including a new row for 'skywalker ewok'.

```
292 -- sustituimos los símbolos '$' de la columna price y cambiamos el tipo de dato de VARCHAR a DECIMAL:
293 SET SQL_SAFE_UPDATES=0;
294
295 UPDATE products
296 SET price = REPLACE(price, "$", "")
297 WHERE price IS NOT NULL;
298
299 ALTER TABLE orders.products
300 CHANGE COLUMN price
301 price DECIMAL(10,2);
302
303 SET SQL_SAFE_UPDATES=1;
```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	161.11	#7c7c7c	1.00	WH-4
2	Tarly Stark	9.24	#919191	2.00	WH-3
3	duel tourney Lannister	171.13	#d8d8d8	1.50	WH-2
4	warden south duel	71.89	#111111	3.00	WH-1
5	skywalker ewok	171.22	#d8d8d8	3.20	WH-0

- También elimino los espacios después de las comas de 'products_id' de la tabla 'transaction', para que al compararla con 'id' de la tabla 'products' no nos tenga en cuenta los espacios vacíos:

```

289 • SET SQL_SAFE_UPDATES=0; -- para desactivar la opción de actualización segura, porque sino no me deja realizar el cambio
290
291 • UPDATE transaction
292   SET products_id = REPLACE(products_id, " ", "")
293   WHERE products_id IS NOT NULL;
294
295 • SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
296
297 • SELECT * FROM orders.transaction; -- visualizo los cambios
298

```

- Y creo la tabla 'sales' que es una tabla puente que me relaciona las tablas 'products' y 'transaction':

```

317 • CREATE TABLE sales AS( -- tabla 7
318   SELECT t.id AS transaction_id, p.id AS product_id
319   FROM products AS p
320   JOIN transaction AS t
321   ON FIND_IN_SET(p.id, t.products_id));
322
323 -- visualizo la nueva tabla:
324 • SELECT * FROM orders.sales;
325
326 -- y compruebo que todos los valores son correctos, sabiendo que tenemos 578 transaction_id diferentes:
327 • SELECT COUNT(DISTINCT transaction_id)
328   FROM sales;

```

- Por último, reo la PK de la tabla 'products' y le añado las FKs a la tabla 'sales' que la relacionan con 'transaction' y 'products':

Table: products

Columns:

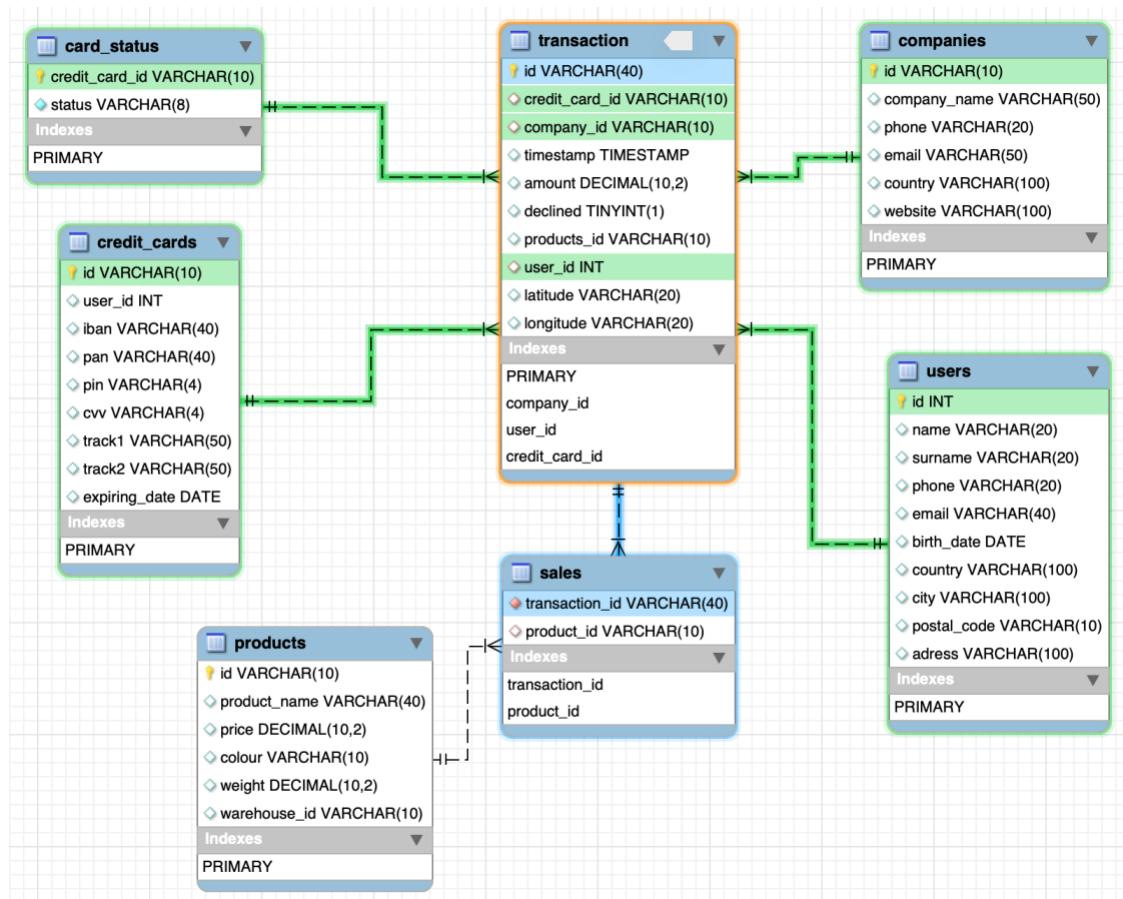
id	varchar(10) PK
product_name	varchar(40)
price	decimal(10,2)
colour	varchar(10)
weight	decimal(10,2)
warehouse_id	varchar(10)

```

314 • ALTER TABLE products
315   ADD PRIMARY KEY (id);
316
317 -- le añado las FKs a la tabla 'sales' que la relacionan con 'transaction' y 'products':
318 • ALTER TABLE sales
319   ADD FOREIGN KEY (transaction_id) REFERENCES transaction (id),
320   ADD FOREIGN KEY (product_id) REFERENCES products (id);

```

- Y así queda nuestra BBDD final:



La tabla puente 'sales' nos queda relacionada con la tabla de hechos 'transaction' a través de una FK llamada transaction_id que se relaciona con la PK (id) de la tabla 'transaction' y también se relaciona a través de la FK producto_id con la PK (id) de la tabla 'products' que es una nueva tabla de dimensiones que añadimos a nuestra BBDD.

Nivell 3

Exercici 1

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

```
325 • SELECT p.product_name AS producto, COUNT(s.product_id) AS cantidad_de_ventas
326 FROM sales AS s
327 JOIN products AS p
328 ON p.id = s.product_id
329 GROUP BY p.id
330 ORDER BY cantidad_de_ventas DESC;
```

100% 65:323

Result Grid Filter Rows: Search Export:

producto	cantidad_de_ventas
riverlands north	65
Winterfell	65
Direwolf riverlands the	63
Tarly Stark	61
skywalker ewok sith	60
Tully	60
duel	57

Result 1

ction Output

	Time	Action	Response
3	18:39:19	SELECT p.product_name AS producto, COUNT(s.product_id) AS cantidad_de_ventas FROM sales AS s JOIN produc...	26 row(s) returned