

**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS**

**DIVISIÓN DE INGENIERÍAS DEPARTAMENTO DE INFORMÁTICA**



Seminario de solución de problemas de Bases de Datos

Martes y jueves

**Actividad 5**

**Actividad de Aprendizaje**

**ALUMNO: LOPEZ RODRIGUEZ CHRISTIAN ADRIAN**

**Código: 218022125**

**Ciclo escolar: 2023-A**

**Profesor: Murillo Leño Maria Magdalena. Sección: D01**

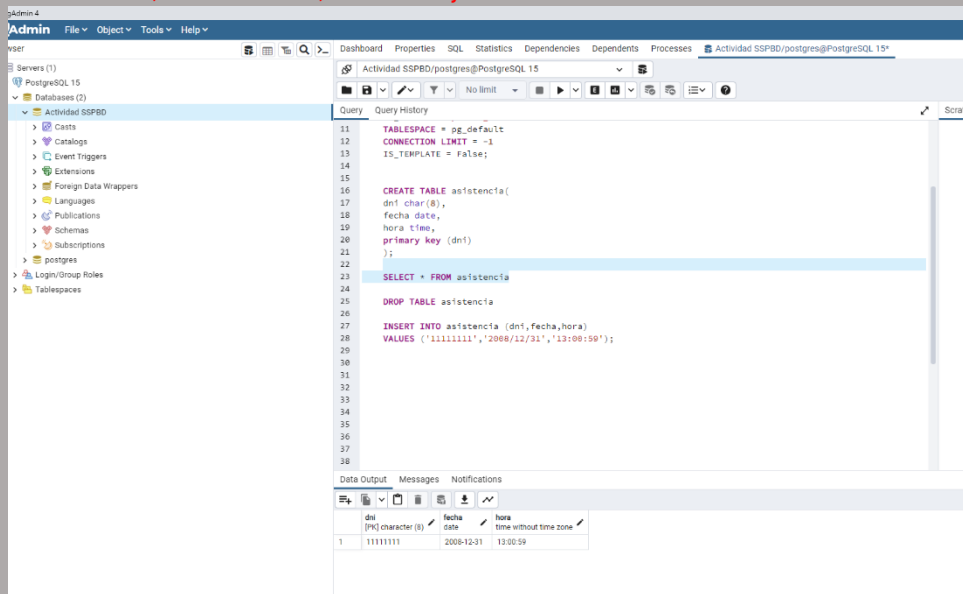
## PROBLEMA 1.-

1.-Trabajamos con la tabla asistencia, donde almacenamos la fecha y hora de ingreso de un empleado:

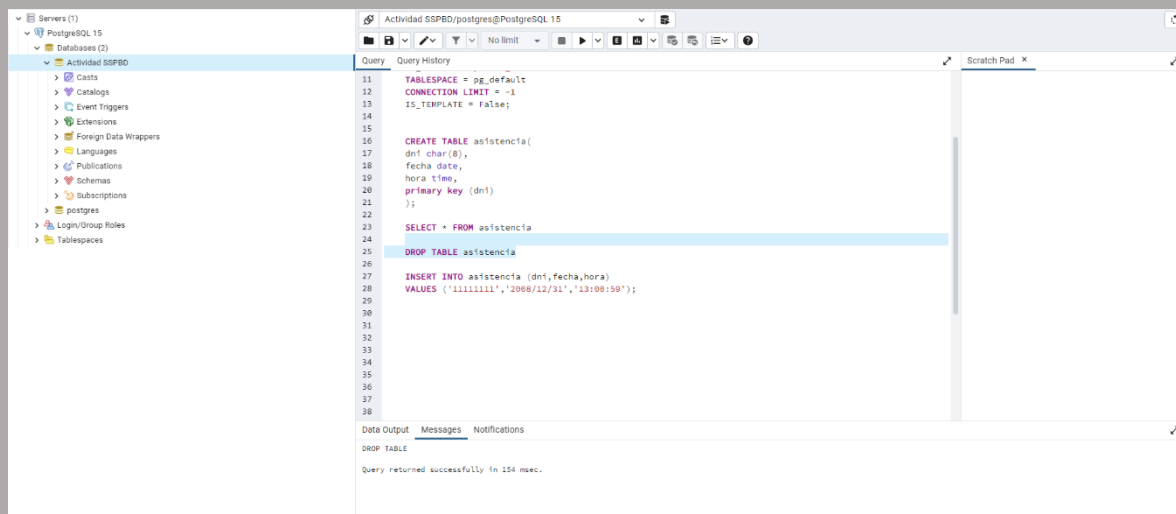
asistencia(dni char(8),fecha date,hora time, primary key (dni))

2.-Ingresamos un registro y lo mostramos:

'11111111','2008/12/31','13:00:59'}



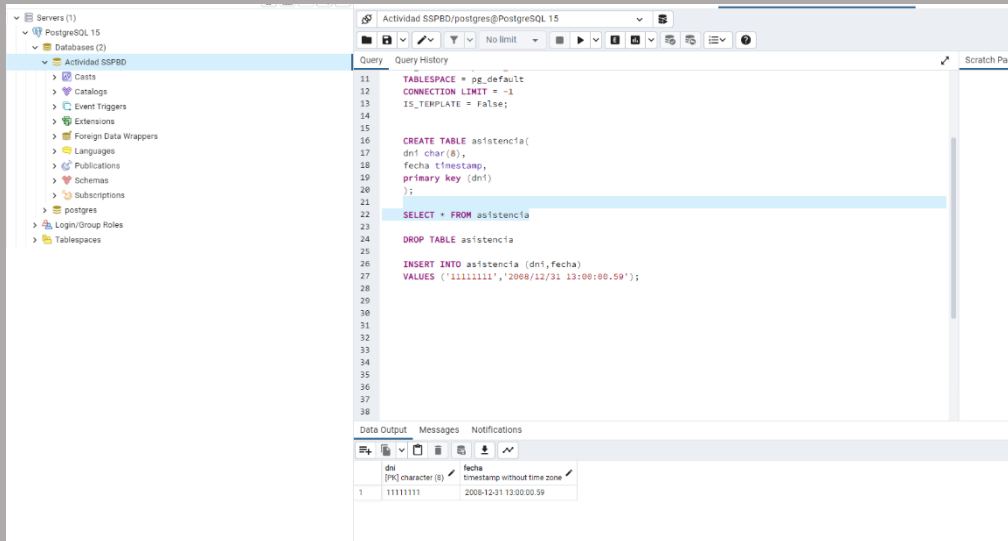
3.-Borramos la tabla:



4.-Creamos nuevamente la tabla pero definimos un solo campo para almacenar la fecha y hora de ingreso del empleado:

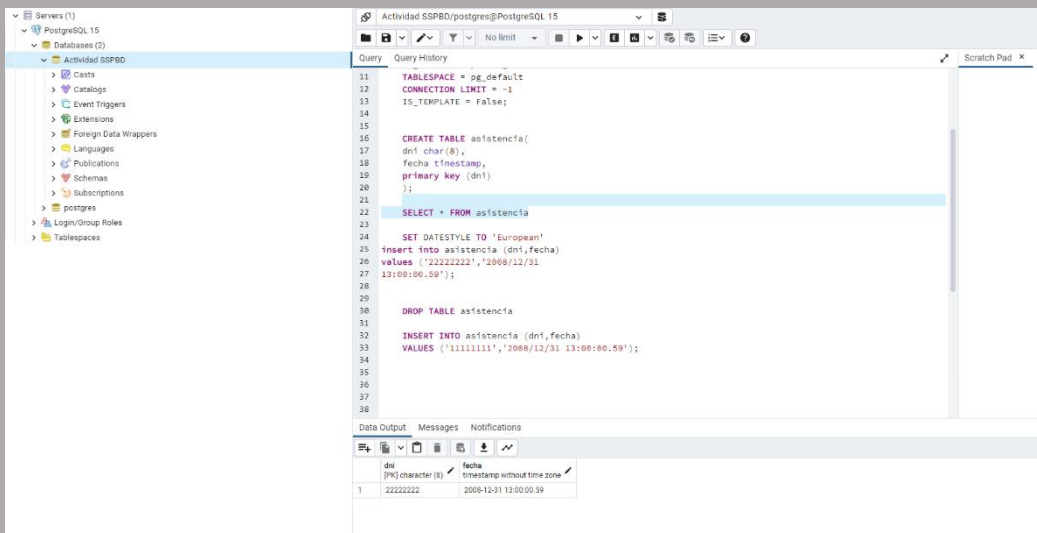
5.-Ingresamos un registro y lo mostramos:

'11111111','2008/12/31 13:00:00.59'

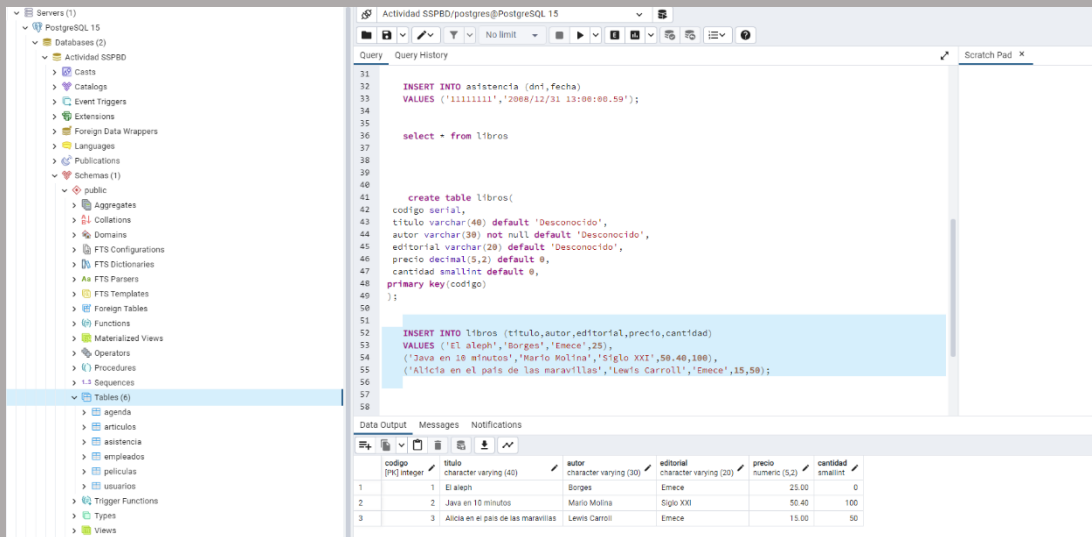


6.-Cambiamos el seteo de fecha para ingresar con el formato dia/mes/año: set datestyle to 'European'; e ingresamos un nuevo registro con el nuevo formato:

insert into asistencia (dni,fechahora) values ('22222222','2008/12/31  
13:00:00.59');



## PROBLEMA 2.-



The screenshot shows the PostgreSQL IDE interface. On the left, the 'Servers' tree is expanded to show the 'public' schema. The main query editor displays the following SQL code:

```

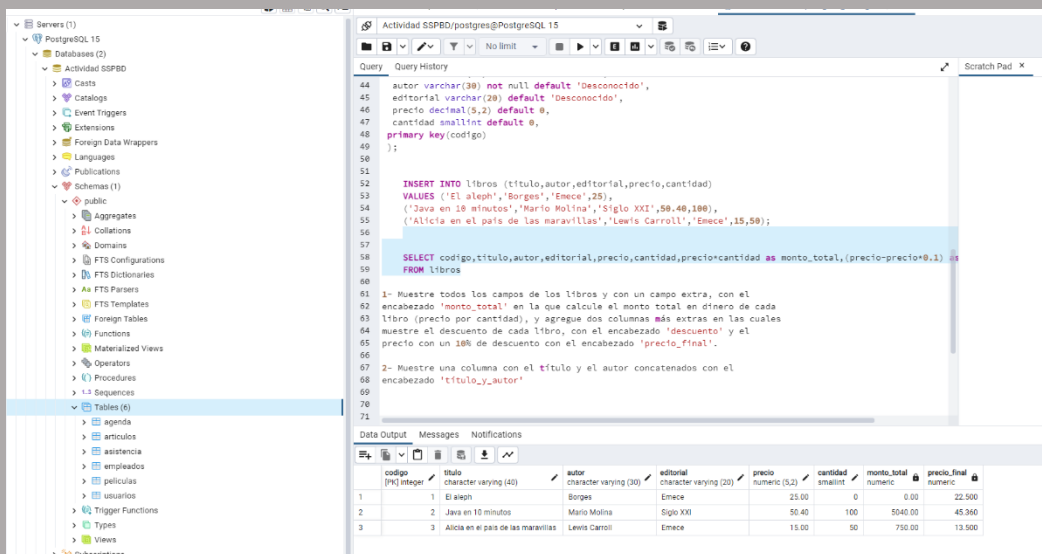
31
32 INSERT INTO asistancia (dni, fecha)
33 VALUES ('11111111', '2008/12/31 13:00:00.59');
34
35
36 select * from libros
37
38
39
40
41 create table libros(
42     codigo serial,
43     titulo varchar(40) default 'Desconocido',
44     autor varchar(30) not null default 'Desconocido',
45     editorial varchar(20) default 'Desconocido',
46     precio decimal(5,2) default 0,
47     cantidad smallint default 0,
48     primary key(codigo)
49 );
50
51
52 INSERT INTO libros (titulo, autor, editorial, precio, cantidad)
53 VALUES ('El aleph', 'Borges', 'Emece', 25),
54 ('Java en 10 minutos', 'Mario Molina', 'Siglo XXI', 50.40, 100),
55 ('Alicia en el país de las maravillas', 'Lewis Carroll', 'Emece', 15, 50);
56
57
58

```

The 'Data Output' pane at the bottom shows the result of the insert query:

	codigo [PK] integer	titulo character varying (40)	autor character varying (30)	editorial character varying (20)	precio numeric (5,2)	cantidad smallint
1	1	El aleph	Borges	Emece	25.00	0
2	2	Java en 10 minutos	Mario Molina	Siglo XXI	50.40	100
3	3	Alicia en el país de las maravillas	Lewis Carroll	Emece	15.00	50

- 1- Muestre todos los campos de los libros y con un campo extra, con el encabezado 'monto\_total' en la que calcule el monto total en dinero de cada libro (precio por cantidad), y agregue dos columnas más extras en las cuales muestre el descuento de cada libro, con el encabezado 'descuento' y el precio con un 10% de descuento con el encabezado 'precio\_final'.



The screenshot shows the PostgreSQL IDE interface. The main query editor displays the following SQL code:

```

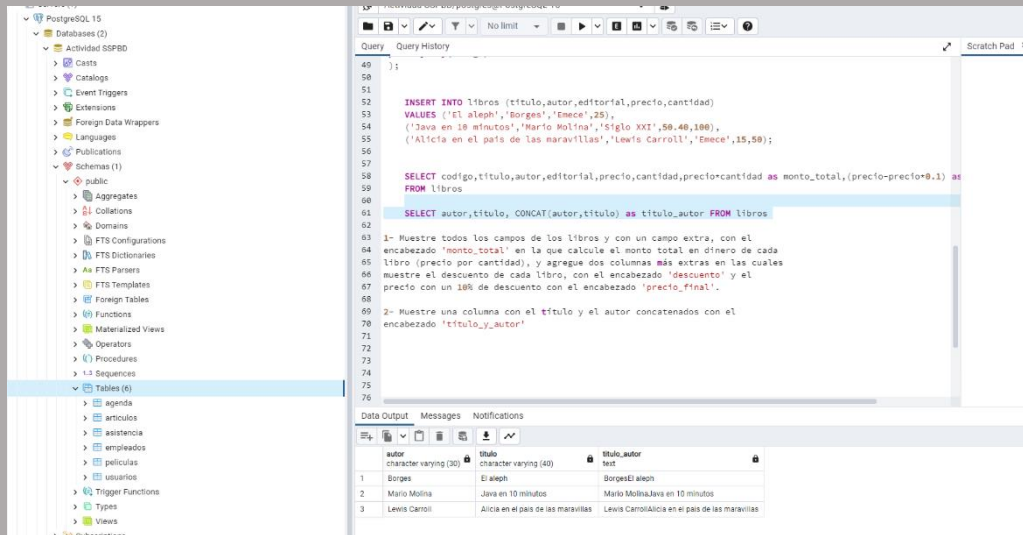
44 autor varchar(30) not null default 'Desconocido',
45 editorial varchar(20) default 'Desconocido',
46 precio decimal(5,2) default 0,
47 cantidad smallint default 0,
48 primary key(codigo)
49 );
50
51
52 INSERT INTO libros (titulo, autor, editorial, precio, cantidad)
53 VALUES ('El aleph', 'Borges', 'Emece', 25),
54 ('Java en 10 minutos', 'Mario Molina', 'Siglo XXI', 50.40, 100),
55 ('Alicia en el país de las maravillas', 'Lewis Carroll', 'Emece', 15, 50);
56
57
58
59 SELECT codigo, titulo, autor, editorial, precio, cantidad, precio*cantidad as monto_total, (precio-precio*0.1) as
60 FROM libros
61
62
63 1- Muestre todos los campos de los libros y con un campo extra, con el
64 encabezado 'monto_total' en la que calcule el monto total en dinero de cada
65 libro (precio por cantidad), y agregue dos columnas más extras en las cuales
66 muestre el descuento de cada libro, con el encabezado 'descuento' y el
67 precio con un 10% de descuento con el encabezado 'precio_final'.
68
69 2- Muestre una columna con el titulo y el autor concatenados con el
70 encabezado 'titulo_y_autor'
71
72

```

The 'Data Output' pane at the bottom shows the result of the query:

	codigo [PK] integer	titulo character varying (40)	autor character varying (30)	editorial character varying (20)	precio numeric (5,2)	cantidad smallint	monto_total numeric	precio_final numeric
1	1	El aleph	Borges	Emece	25.00	0	0.00	22.500
2	2	Java en 10 minutos	Mario Molina	Siglo XXI	50.40	100	5040.00	45.360
3	3	Alicia en el país de las maravillas	Lewis Carroll	Emece	15.00	50	750.00	13.500

## 2- Muestre una columna con el título y el autor concatenados con el encabezado 'título\_y\_autor'



The screenshot shows the PostgreSQL 15 interface. On the left is the 'Database Explorer' tree with 'public' selected. The main window displays a SQL query and its results. The query is as follows:

```
40 }
41
42 INSERT INTO libros (titulo,autor,editorial,precio,cantidad)
43 VALUES ('El aleph','Borges','Emece',25),
44 ('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100),
45 ('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);
46
47
48 SELECT codigo,titulo,autor,editorial,precio,cantidad,precio*cantidad as monto_total,(precio-precio*0.1) as
49 FROM libros
50
51
52 SELECT autor,titulo, CONCAT(autor,titulo) as titulo_autor FROM libros;
```

The results of the second query are shown in the 'Data Output' pane:

	autor	titulo	titulo_autor
	character varying (20)	character varying (40)	text
1	Borges	El aleph	BorgesEl aleph
2	Mario Molina	Java en 10 minutos	Mario MolinaJava en 10 minutos
3	Lewis Carroll	Alicia en el país de las maravillas	Lewis CarrollAlicia en el país de las maravillas

## PROBLEMA 3.-

Trabaje con la tabla 'libros' de una librería.

Crear la tabla:

```
create table libros(
    codigo serial,
    titulo varchar(40) not null,
    autor varchar(20) default 'Desconocido',
    editorial varchar(20),
    precio decimal(6,2),
    cantidad smallint default 0,
    primary key (codigo)
);
```

Ingresa los siguientes registros:

```
insert into libros (titulo,autor,editorial,precio)
values('El aleph','Borges','Emece',25);
```

insert into libros (titulo,autor,editorial,precio,cantidad)

values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);

insert into libros (titulo,autor,editorial,precio,cantidad)

values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);

The screenshot shows the PostgreSQL 15 interface. On the left, the 'Databases (2)' tree is expanded to 'Tables (6)'. The main query editor displays the following SQL code:

```
38
39
40
41 create table libros(
42     codigo serial,
43     titulo varchar(40) default 'Desconocido',
44     autor varchar(30) not null default 'Desconocido',
45     editorial varchar(20) default 'Desconocido',
46     precio decimal(6,2) default 0,
47     cantidad smallint default 0,
48     primary key(codigo)
49 );
50
51 insert into libros (titulo,autor,editorial,precio)
52 values('El aleph','Borges','Emece',25);
53 insert into libros (titulo,autor,editorial,precio,cantidad)
54 values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
55 insert into libros (titulo,autor,editorial,precio,cantidad)
56 values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);
57
58
59
60
61
62
63
64
65
```

The 'Data Output' tab shows the resulting data in the 'libros' table:

codigo [PK] integer	titulo character varying (40)	autor character varying (30)	editorial character varying (20)	precio numeric (6,2)	cantidad smallint
1	El aleph	Borges	Emece	25.00	0
2	Java en 10 minutos	Mario Molina	Siglo XXI	50.40	100
3	Alicia en el pais de las maravillas	Lewis Carroll	Emece	15.00	50

1.- Mostrar sólo los 12 primeros caracteres de los títulos de los libros y sus autores,

(revise que función deberá utilizar para este problema):

The screenshot shows the PostgreSQL 15 interface. The main query editor displays the following SQL code:

```
38
39
40
41 create table libros(
42     codigo serial,
43     titulo varchar(40) default 'Desconocido',
44     autor varchar(30) not null default 'Desconocido',
45     editorial varchar(20) default 'Desconocido',
46     precio decimal(6,2) default 0,
47     cantidad smallint default 0,
48     primary key(codigo)
49 );
50
51 insert into libros (titulo,autor,editorial,precio)
52 values('El aleph','Borges','Emece',25);
53 insert into libros (titulo,autor,editorial,precio,cantidad)
54 values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
55 insert into libros (titulo,autor,editorial,precio,cantidad)
56 values('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);
57
58
59
60
61
62
63
64
65
```

The 'Data Output' tab shows the result of the query:

result text
1 El aleph
2 Java en 10 minutos
3 Alicia en el pais de las maravillas

2.- Mostrar los títulos, autores y editoriales de todos libros, al último campo lo queremos en mayúsculas:

The screenshot shows the PostgreSQL IDE interface. On the left, the 'Databases (2)' tree is expanded to 'Tables (6)'. The main editor displays the following SQL code:

```
38
39
40
41
42 create table libros(
43     codigo serial,
44     titulo varchar(40) default 'Desconocido',
45     autor varchar(30) not null default 'Desconocido',
46     editorial varchar(20) default 'Desconocido',
47     precio decimal(6,2) default 0,
48     cantidad smallint default 0,
49     primary key(codigo)
50 );
51
52 insert into libros (titulo,autor,editorial,precio)
53 values('El aleph','Borges','Emece',25);
54 insert into libros (titulo,autor,editorial,precio,cantidad)
55 values('Java en 10 minutos','Mario Molina','Siglo XXI',50-40,100);
56 insert into libros (titulo,autor,editorial,precio,cantidad)
57 values('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);
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
```

The Data Output pane shows the following results:

titulo	autor	upper
El aleph	Borges	EMECE
Java en 10 minutos	Mario Molina	SIGLO XXI
Alicia en el país de las maravillas	Lewis Carroll	EMECE

3.- Mostrar todos los títulos de los libros rellenando con el caracter '-' a la derecha.

The screenshot shows the PostgreSQL IDE interface. The main editor displays the same SQL code as in the previous screenshot, but the query is now:

```
38
39
40
41
42 create table libros(
43     codigo serial,
44     titulo varchar(40) default 'Desconocido',
45     autor varchar(30) not null default 'Desconocido',
46     editorial varchar(20) default 'Desconocido',
47     precio decimal(6,2) default 0,
48     cantidad smallint default 0,
49     primary key(codigo)
50 );
51
52 insert into libros (titulo,autor,editorial,precio)
53 values('El aleph','Borges','Emece',25);
54 insert into libros (titulo,autor,editorial,precio,cantidad)
55 values('Java en 10 minutos','Mario Molina','Siglo XXI',50-40,100);
56 insert into libros (titulo,autor,editorial,precio,cantidad)
57 values('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);
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
```

The Data Output pane shows the following results:

pad
El aleph-----
Java en 10 minutos-----
Alicia en el país de las maravillas-----

4.- Imprimir todo los libros que contienen la cadena 'en' en alguna parte del título del libro.

The screenshot shows the PostgreSQL Enterprise Studio interface. On the left, the 'Servers' tree is expanded to 'PostgreSQL 15' and 'Databases (2)', with 'Actividad SSPBD' selected. The main editor displays a SQL script with the following content:

```
38
39
40
41 create table libros(
42     codigo serial,
43     titulo varchar(40) default 'Desconocido',
44     autor varchar(30) not null default 'Desconocido',
45     editorial varchar(20) default 'Desconocido',
46     precio decimal(6,2) default 0,
47     cantidad smallint default 0,
48     primary key(codigo)
49 );
50
51 insert into libros (titulo,autor,editorial,precio)
52 values('El aleph','Borges','Emece',25);
53 insert into libros (titulo,autor,editorial,precio,cantidad)
54 values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
55 insert into libros (titulo,autor,editorial,precio,cantidad)
56 values('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);
57
58
59 SELECT titulo FROM libros
60 WHERE titulo LIKE 'René';
61
62
63
64
65
```

The 'Data Output' pane at the bottom shows the results of the last query:

titulo
Java en 10 minutos
Alicia en el país de las maravillas

5.- Imprimir todos los libros que tienen un título con 10 o más caracteres:

The screenshot shows the PostgreSQL Enterprise Studio interface. On the left, the 'Servers' tree is expanded to 'PostgreSQL 15' and 'Databases (2)', with 'Actividad SSPBD' selected. The main editor displays a SQL script with the following content:

```
38
39
40
41 create table libros(
42     codigo serial,
43     titulo varchar(40) default 'Desconocido',
44     autor varchar(30) not null default 'Desconocido',
45     editorial varchar(20) default 'Desconocido',
46     precio decimal(6,2) default 0,
47     cantidad smallint default 0,
48     primary key(codigo)
49 );
50
51 insert into libros (titulo,autor,editorial,precio)
52 values('El aleph','Borges','Emece',25);
53 insert into libros (titulo,autor,editorial,precio,cantidad)
54 values('Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
55 insert into libros (titulo,autor,editorial,precio,cantidad)
56 values('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);
57
58
59 SELECT titulo FROM libros
60 WHERE LENGTH(titulo) > 10;
61
62
63
64
65
```

The 'Data Output' pane at the bottom shows the results of the last query:

titulo
Java en 10 minutos
Alicia en el país de las maravillas



#### PROBLEMA 4.-

Una facultad almacena los datos de sus alumnos en una tabla denominada “alumnos”;

Cree la tabla eligiendo el tipo de dato adecuado para cada campo:

```
create table alumnos(  
    apellido varchar(30),  
    nombre varchar(30),  
    documento char(8),  
    domicilio varchar(30),  
    fechaingreso date,  
    fechanacimiento date  
);
```

1.- Setee el formato para entrada de datos de tipo fecha para que acepte valores día-mes-año

```
SET DATESTYLE TO 'European'
```

2- Ingrese un alumno empleando distintos separadores para las fechas

```
INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
```

```
VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',  
    TO_DATE('01.01.2022', 'DD.MM.YYYY'),  
    TO_DATE('12.03.2000', 'DD.MM.YYYY'));
```

The screenshot shows the PostgreSQL 15 interface. On the left, the 'Servers' tree is expanded to show the 'public' schema. The 'alumnos' table is listed under 'Tables (6)'. The main window displays the SQL query history, which includes the following SQL statements:

```
55 insert into libros (titulo,autor,editorial,precio,cantidad)  
56 values('Alicia en el país de las maravillas','Lewis Carroll','Emece',15,50);  
57  
58  
59 SELECT titulo FROM libros  
60 WHERE LENGTH(titulo) > 10;  
61  
62 1- Setee el formato para entrada de datos de tipo fecha para que acepte valores día-mes-año  
63  
64 create table alumnos(  
65     apellido varchar(30),  
66     nombre varchar(30),  
67     documento char(8),  
68     domicilio varchar(30),  
69     fechaingreso date,  
70     fechanacimiento date  
71 );  
72  
73  
74  
75 SET DATESTYLE TO 'European'  
76  
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)  
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',  
79         TO_DATE('01.01.2022', 'DD.MM.YYYY'),  
80         TO_DATE('12.03.2000', 'DD.MM.YYYY'));  
81  
82
```

Below the query history, the 'Data Output' tab shows the result of the insertion:

apellido	nombre	documento	domicilio	fechaingreso	fechanacimiento
Pérez	Juan	12345678	Calle 123, Ciudad	2022-01-01	2000-03-12

### 3- Ingrese otro alumno empleando solamente un dígito para día y mes y 2 para el año

Query History

```
59 SELECT titulo FROM libros
60 WHERE LENGTH(titulo) > 10;
61
62 1- Setee el formato para entrada de datos de tipo fecha para que acepte valores día-mes-año
63
64 create table alumnos(
65     apellido varchar(30),
66     nombre varchar(30),
67     documento char(8),
68     domicilio varchar(30),
69     fechaingreso date,
70     fechanacimiento date
71 );
72
73 SELECT * FROM alumnos
74
75 SET DATESTYLE TO 'European'
76
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',
79         TO_DATE('01-01-2022', 'DD-MM-YYYY'),
80         TO_DATE('12-03-2000', 'DD-MM-YYYY'));
81
82 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
83 VALUES ('Gómez', 'Ana', '23456789', 'Calle 456, Ciudad',
84         TO_DATE('01/01/22', 'DD/MM/YY'),
85         TO_DATE('2/3/90', 'DD/MM/YY'));
```

Data Output

apellido	nombre	documento	domicilio	fechaingreso	fechanacimiento
Pérez	Juan	12345678	Calle 123, Ciudad	2022-01-01	2000-03-12
Gómez	Ana	23456789	Calle 456, Ciudad	2022-01-01	1990-03-02

### 4- Ingrese un alumnos empleando 2 dígitos para el año de la fecha de ingreso y 'null' en 'fechanacimiento'

Query History

```
67 documento char(8);
68 domicilio varchar(30),
69 fechaingreso date,
70 fechanacimiento date
71 );
72
73 SELECT * FROM alumnos
74
75 SET DATESTYLE TO 'European'
76
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',
79         TO_DATE('01-01-2022', 'DD-MM-YYYY'),
80         TO_DATE('12-03-2000', 'DD-MM-YYYY'));
81
82 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
83 VALUES ('Gómez', 'Ana', '23456789', 'Calle 456, Ciudad',
84         TO_DATE('01/01/22', 'DD/MM/YY'),
85         TO_DATE('2/3/90', 'DD/MM/YY'));
```

Data Output

apellido	nombre	documento	domicilio	fechaingreso	fechanacimiento
Pérez	Juan	12345678	Calle 123, Ciudad	2022-01-01	2000-03-12
Gómez	Ana	23456789	Calle 456, Ciudad	2022-01-01	1990-03-02
González	María	34567890	Calle 789, Ciudad	2022-01-01	[null]

## 5- Muestre todos los alumnos que ingresaron antes del '1-1-91'

The screenshot shows a PostgreSQL IDE with a query editor on the left and a data output table on the right. The query editor contains the following SQL code:

```
72
73 SELECT * FROM alumnos
74
75 SET DATESTYLE TO 'European'
76
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',
79         TO_DATE('01-01-2022', 'DD-MM-YYYY'),
80         TO_DATE('12-03-2000', 'DD-MM-YYYY'));
81
82 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
83 VALUES ('Gómez', 'Ana', '23456789', 'Calle 456, Ciudad',
84         TO_DATE('01/01/22', 'DD/MM/YY'),
85         TO_DATE('2/3/90', 'DD/MM/YY'));
86
87 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
88 VALUES ('González', 'María', '34567890', 'Calle 789, Ciudad',
89         TO_DATE('01/01/22', 'DD/MM/YY'),
90         NULL);
91
92
93 5- Muestre todos los alumnos que ingresaron antes del '1-1-91'
94
95 SELECT apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento FROM alumnos
96 WHERE fechaingreso < '1-1-91'
97
98
99
```

The data output table shows the following results:

apellido	nombre	documento	domicilio	fechaingreso	fechanacimiento
Pérez	Juan	12345678	Calle 123, Ciudad	2022-01-01	2000-03-12
Gómez	Ana	23456789	Calle 456, Ciudad	2022-01-01	1990-03-02
González	María	34567890	Calle 789, Ciudad	2022-01-01	[null]

## 6- Muestre todos los alumnos que tienen 'null' en 'fechanacimiento':

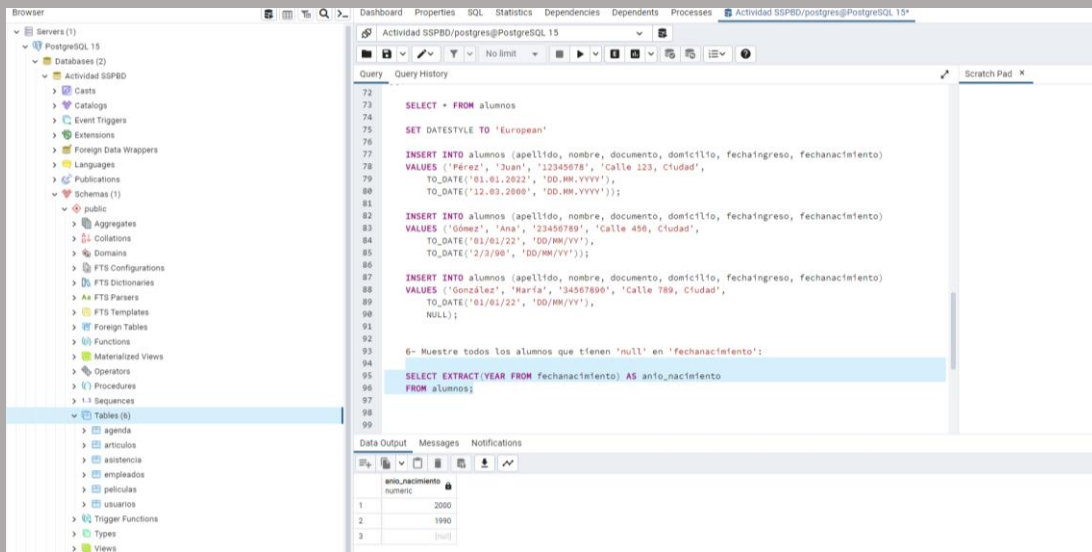
The screenshot shows a PostgreSQL IDE with a query editor on the left and a data output table on the right. The query editor contains the following SQL code:

```
72
73 SELECT * FROM alumnos
74
75 SET DATESTYLE TO 'European'
76
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',
79         TO_DATE('01-01-2022', 'DD-MM-YYYY'),
80         TO_DATE('12-03-2000', 'DD-MM-YYYY'));
81
82 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
83 VALUES ('Gómez', 'Ana', '23456789', 'Calle 456, Ciudad',
84         TO_DATE('01/01/22', 'DD/MM/YY'),
85         TO_DATE('2/3/90', 'DD/MM/YY'));
86
87 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
88 VALUES ('González', 'María', '34567890', 'Calle 789, Ciudad',
89         TO_DATE('01/01/22', 'DD/MM/YY'),
90         NULL);
91
92
93 6- Muestre todos los alumnos que tienen 'null' en 'fechanacimiento':
94
95 SELECT apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento FROM alumnos
96 WHERE fechanacimiento IS NULL
97
98
99
```

The data output table shows the following results:

apellido	nombre	documento	domicilio	fechaingreso	fechanacimiento
González	María	34567890	Calle 789, Ciudad	2022-01-01	[null]

## 7- Muestre el año de nacimiento de todos los alumnos.



The screenshot shows a PostgreSQL IDE interface. On the left, a 'Browser' pane displays the database structure, including 'Actividad SSPBD', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas (1)', 'public' schema, 'Aggregates', 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Operators', 'Procedures', 'Sequences', 'Tables (6)', 'agenda', 'articulos', 'asistencia', 'empleados', 'peliculas', 'usuarios', 'Trigger Functions', 'Types', and 'Views'. The 'Tables (6)' folder is selected.

The main editor displays a SQL query with line numbers 72 to 99. The query includes several INSERT statements and a SELECT statement. The SELECT statement is highlighted in blue:

```
72
73 SELECT * FROM alumnos
74
75 SET DATESTYLE TO 'European'
76
77 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
78 VALUES ('Pérez', 'Juan', '12345678', 'Calle 123, Ciudad',
79         TO_DATE('01.01.2022', 'DD.MM.YYYY'),
80         TO_DATE('12.03.2000', 'DD.MM.YYYY'))
81
82 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
83 VALUES ('Gómez', 'Ana', '23456789', 'Calle 456, Ciudad',
84         TO_DATE('01/01/22', 'DD/MM/YY'),
85         TO_DATE('2/3/98', 'DD/MM/YY'))
86
87 INSERT INTO alumnos (apellido, nombre, documento, domicilio, fechaingreso, fechanacimiento)
88 VALUES ('González', 'María', '34567890', 'Calle 789, Ciudad',
89         TO_DATE('01/01/22', 'DD/MM/YY'),
90         NULL);
91
92
93 6- Muestre todos los alumnos que tienen 'null' en 'fechanacimiento':
94
95 SELECT EXTRACT(YEAR FROM fechanacimiento) AS año_nacimiento
96 FROM alumnos;
97
98
99
```

Below the query editor, the 'Data Output' pane shows the results of the SELECT statement. It displays a table with two columns: 'año\_nacimiento' and 'numeric'. The results are:

año_nacimiento	numeric
1	2000
2	1990
3	[null]