

BASE DE DATOS II

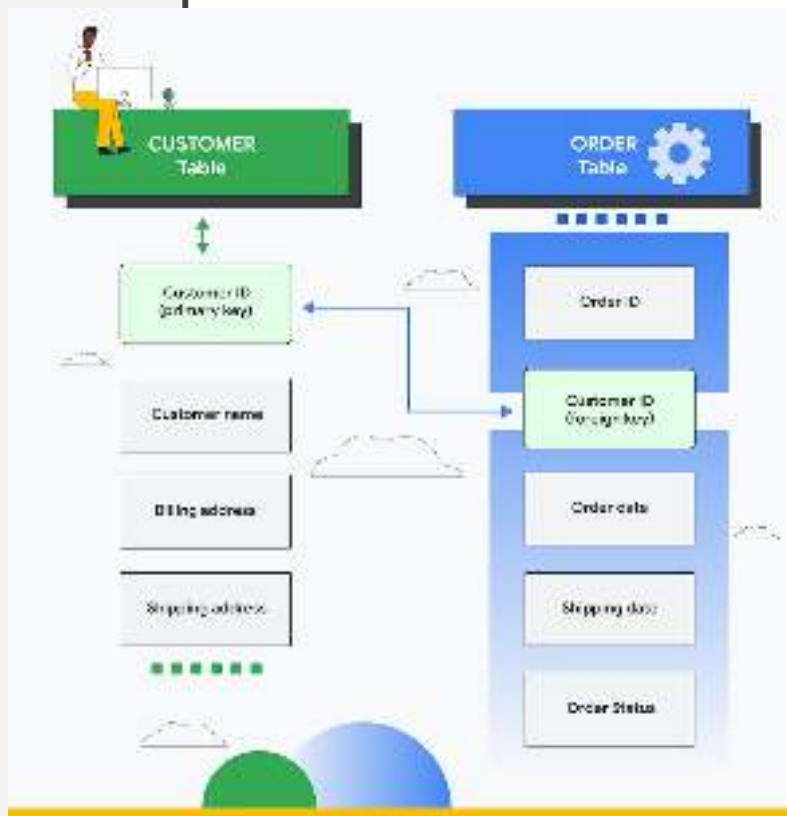
Estudiante : Elvin Braxail Cussi Aranibar

Semestre : 3er

Año : 2023

MANEJO DE CONCEPTOS

I. ¿A QUE SE REFIERE CUANDO SE HABLA DE BASES DE DATOS RELACIONALES?



Conjunto de información relacionada entre sí, referente a un tema o propósito en particular.

2. ¿A QUE SE REFIERE CUANDO SE HABLA DE BASES DE DATOS NO RELACIONALES?



Conjunto de datos que funcionan de forma independientemente a esto se le añade que esto solo se usa en sitios web como puede ser Facebook

3. ¿QUÉ ES MYSQL Y MARIADB?. EXPLIQUE SI EXISTEN DIFERENCIAS O SON IGUALES, ETC

MariaDB ha agregado varias características y funcionalidades adicionales en comparación con MySQL, como la capacidad de manejar una mayor cantidad de conexiones concurrentes, soporte mejorado para la replicación y clústeres de alta disponibilidad, y nuevas funciones de seguridad.



4. ¿QUÉ SON LAS FUNCIONES DE AGREGACIÓN?

Nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos



5. ¿QUÉ LLEGARÍA A SER XAMPP, WAMP SERVER O LAMP?



XAMPP, WAMP Server y LAMP son herramientas de software para crear entornos de desarrollo y servidores web, que incluyen los componentes básicos necesarios para crear aplicaciones y sitios web, tales como un servidor web, una base de datos y un lenguaje de programación. La elección entre estas herramientas dependerá del sistema operativo y las necesidades específicas del usuario.

6. ¿CUAL ES LA DIFERENCIA ENTRE LAS FUNCIONES DE AGRESIÓN Y FUNCIONES CREADOS POR EL DBA? ES DECIR FUNCIONES CREADAS POR EL USUARIO.



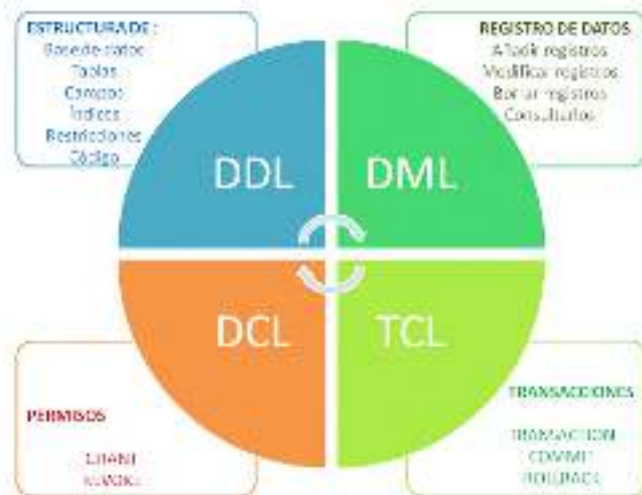
La principal diferencia entre las funciones de agregación y las funciones creadas por el usuario es que las funciones de agregación son funciones integradas en el DBMS y están disponibles para su uso inmediato, mientras que las funciones creadas por el usuario deben ser creadas y definidas antes de poder ser utilizadas en la base de datos.

7. ¿PARA QUÉ SIRVE EL COMANDO USE?

```
1 CREATE DATABASE biblioteca;
2 USE biblioteca;
3
4 CREATE TABLE autor (
5     id INTEGER AUTO_INCREMENT PRIMARY KEY,
6     nombre VARCHAR(50) NOT NULL,
7     nacionalidad VARCHAR(50),
8     fecha_nacimiento DATE
9 );
10 CREATE TABLE usuario (
11     id INTEGER AUTO_INCREMENT PRIMARY KEY,
12     nombre VARCHAR(50) NOT NULL,
13     email VARCHAR(100) NOT NULL,
14     fecha_nacimiento DATE,
15     direccion VARCHAR(100)
16 );
```

El comando USE es un comando de SQL utilizado para cambiar la base de datos en la que se está trabajando en un servidor de bases de datos.

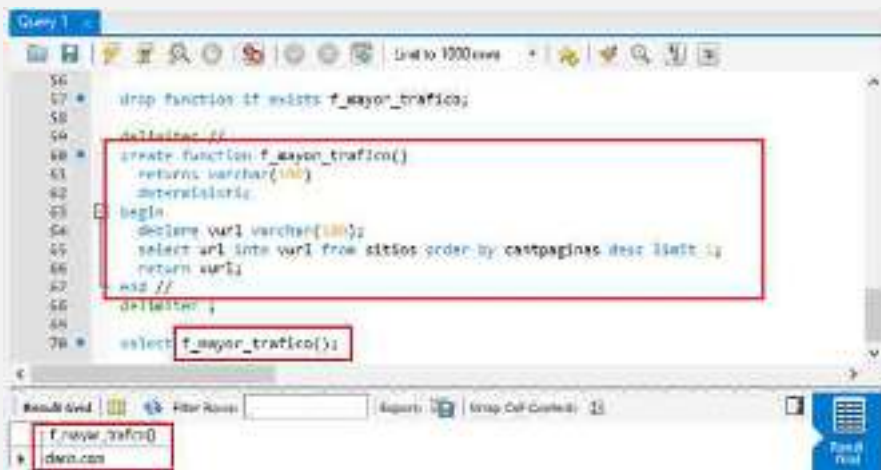
8. QUE ES DML Y DDL?



DML (Data Manipulation Language) Estos comandos incluyen SELECT, INSERT, UPDATE y DELETE. La función principal de los comandos DML es recuperar, agregar, modificar o eliminar datos en una o más tablas de una base de datos.

DDL (Data Definition Language) es un conjunto de comandos utilizados para definir y administrar la estructura de una base de datos. Estos comandos incluyen CREATE, ALTER y DROP.

9. ¿QUÉ COSAS CARACTERÍSTICAS DEBE DE TENER UNA FUNCIÓN? EXPLIQUE SOBRE EL NOMBRE, EL RETURN, PARAMETROS, ETC.?



```
56
57 * drop function if exists f_mayor_trafico;
58
59
60 * create function f_mayor_trafico()
61     returns varchar(100)
62     deterministic
63     begin
64         declare url varchar(100);
65         select url into url from sitios order by castpaginas desc limit 1;
66         return url;
67     end //
68
69 * select f_mayor_trafico();
```

The screenshot shows a SQL query editor with a toolbar at the top. The query is written in a monospaced font. A red rectangle highlights the function definition block (lines 60-67). Below the query editor, the results pane shows the output of the function call: 'f_mayor_trafico()' and 'denn.com'.

Nombre: Una función debe tener un nombre descriptivo que indique su función

Parámetros: Una función puede recibir uno o más que se utilizan para pasar datos a la función

Cuerpo de la función: El cuerpo de la función es donde se define la tarea que se realizará. Esto puede incluir declaraciones de variables, estructuras de control de flujo (como if, while, for

Retorno: Una función puede devolver un valor de salida a la parte del programa que la llamó. Este valor de retorno puede ser de cualquier tipo de datos, como enteros, cadenas, objetos, etc. Si una función no devuelve ningún valor, se dice que su tipo de retorno es void.

10.¿CÓMO CREAR, MODIFICAR Y CÓMO ELIMINAR UNA FUNCIÓN?

```
CREATE FUNCTION promedio(num1 INT, num2 INT)
RETURNS FLOAT
BEGIN
    DECLARE resultado FLOAT;
    SET resultado = (num1 + num2) / 2;
    RETURN resultado;
END;
```

```
ALTER FUNCTION promedio(num1 INT, num2 INT, num3 INT)
RETURNS FLOAT
BEGIN
    DECLARE resultado FLOAT;
    SET resultado = (num1 + num2 + num3) / 3;
    RETURN resultado;
END;
```

PARTE PRACTICA

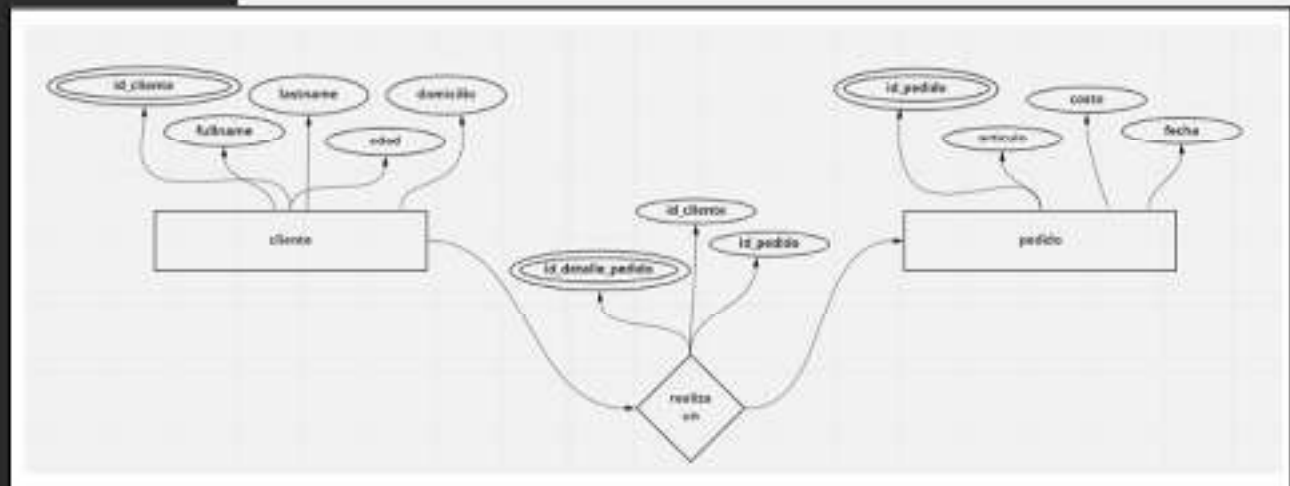
II. CREAR LAS TABLAS Y 2 REGISTROS PARA CADA TABLA PARA EL SIGUIENTE MODELO ER

```
CREATE DATABASE Pizzeria;
USE Pizzeria;

CREATE TABLE cliente (
  id_cliente VARCHAR(10) PRIMARY KEY,
  fullname VARCHAR(50) NOT NULL,
  lastname VARCHAR(50),
  email VARCHAR(100) NOT NULL,
  direccion VARCHAR(100) NOT NULL
);

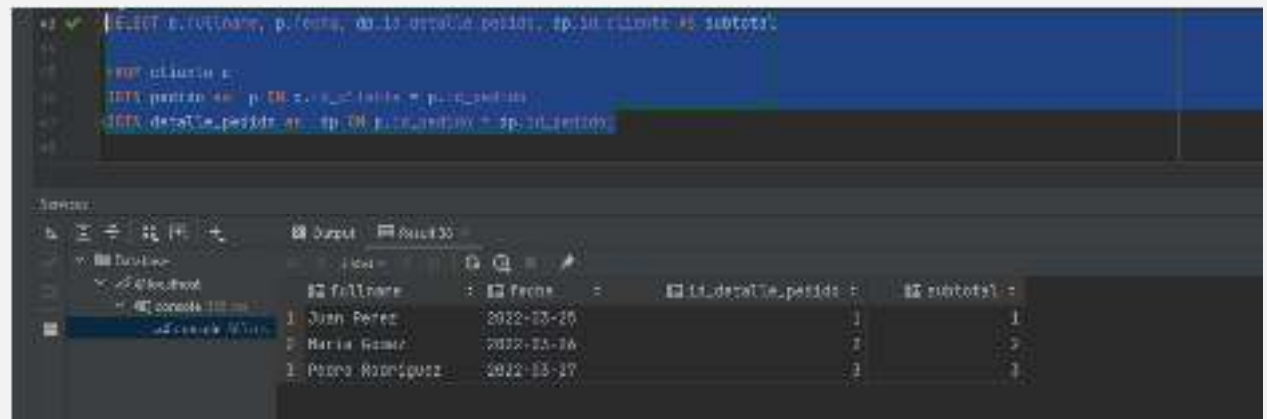
CREATE TABLE pedido (
  id_pedido VARCHAR(10) PRIMARY KEY,
  articulo VARCHAR(50) NOT NULL,
  valor DECIMAL(10,2) NOT NULL,
  fecha DATE
);

CREATE TABLE pedido_cliente (
  id_pedido VARCHAR(10) PRIMARY KEY,
  id_cliente VARCHAR(10) PRIMARY KEY,
  FOREIGN KEY (id_pedido) REFERENCES pedido (id_pedido),
  FOREIGN KEY (id_cliente) REFERENCES cliente (id_cliente)
);
```



12. CREAR UNA CONSULTA SQL EN BASE AL EJERCICIO ANTERIOR

- Debe de utilizar las 3 tablas creadas anteriormente.
- Para relacionar las tablas utilizar JOINS.
- Adjuntar el código SQL generado.



The screenshot shows a SQL IDE with a query window and a results window. The query window contains the following SQL code:

```
SELECT e.fullname, p.fecha, d.id_detalle_pedido, sp.id_producto, sp.subtotal
FROM cliente e
JOIN pedido p ON e.id_cliente = p.id_cliente
JOIN detalle_pedido d ON p.id_pedido = d.id_pedido
JOIN producto sp ON d.id_producto = sp.id_producto;
```

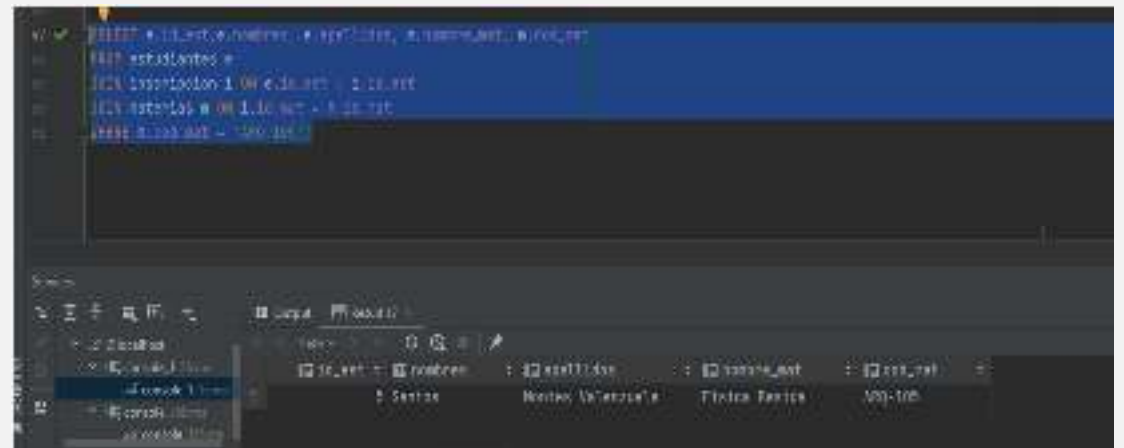
The results window displays the following data:

| id_detalle_pedido | id_producto | subtotal |
|-------------------|-------------|----------|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |

13. RESOLVER LO SIGUIENTE:

- Mostrar los nombres y apellidos de los estudiantes inscritos en la materia ARQ-105, adicionalmente mostrar el nombre de la materia.

- Deberá de crear una función que reciba dos parámetros y esta función deberá ser utilizada en la cláusula WHERE.



The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:

```
SELECT * FROM estudiantes, materias WHERE  
estudiantes.nombre =  
materia.nombre AND materia = 'ARQ-105'  
ORDER BY materia;
```

The results pane shows a table with the following data:

| id_estudiante | nombre | apellido | materia |
|---------------|------------|----------|---------|
| 1 | Estudiante | Apellido | ARQ-105 |

15. CREAR UNA FUNCIÓN QUE PERMITA CONCATENAR 3 CADENAS.

- La función recibe 3 parámetros.
 - Si las cadenas fuesen: ■ Pepito ■ Pep ■ 50
 - La salida debería ser: (Pepito), (Pep), (50)
 - La función creada utilizarlo en una consulta SQL.
- Es decir podría mostrar el nombre, apellidos y la edad de los estudiantes

```
CREATE FUNCTION concat_strings(str1 VARCHAR(50), str2 VARCHAR(50), str3 VARCHAR(50))  
RETURNS VARCHAR(200)  
BEGIN  
    RETURN CONCAT(' ', str1, ' ', ' ', str2, ' ', ' ', str3, ' ');  
END;  
  
SELECT concat_strings( m1.nombres, m1.apellidos, m1.edad) AS concatenated_strings FROM estudiantes;
```



| | concatenated_strings |
|---|-------------------------------------|
| 1 | (Miguel), (Gonzales Veliz), (28) |
| 2 | (Sandra), (Mavir Uriá), (25) |
| 3 | (Joel), (Adubiri Mondar), (38) |
| 4 | (Andrea), (Arias Ballesteros), (21) |
| 5 | (Santos), (Montes Valenzuela), (24) |

16. CREAR LA SIGUIENTE VISTA:

- La vista deberá llamarse **ARQUITECTURA_DIA_LIBRE**
- El día viernes tendrán libre los estudiantes de la carrera de **ARQUITECTURA** debido a su aniversario
 - Este permiso es solo para aquellos estudiantes inscritos en el año 2021.
- La vista deberá tener los siguientes campos. 1. Nombres y apellidos concatenados = **FULLNAME** 2. La edad del estudiante = **EDAD** 3. El año de inscripción = **GESTION** 4. Generar una columna de nombre **DIA_LIBRE** a. Si tiene libre mostrar **LIBRE** b. Caso contrario mostrar **NO LIBRE**

```
CREATE VIEW ARQUITECTURA_DIA_LIBRE AS
SELECT CONCAT(e.nombre, ' ', e.apellidos) AS FULLNAME,
       e.edad AS EDAD,
       e.gestion AS GESTION,
       CASE WHEN DAYOFWEEK('2021-05-04') = 6 THEN 'LIBRE' ELSE 'NO LIBRE' END AS DIA_LIBRE
FROM estudiantes e
INNER JOIN inscripcion i ON e.id_est = i.id_est
INNER JOIN materias m ON i.id_mat = m.id_mat
WHERE m.cod_mat = 'ARQ-105' AND i.gestion = 2021;

create table paralelo (
  id_paralelo integer primary key,
  cantidad_est integer,
  horario varchar (50),
  nombre varchar (50)
);
```

17. CREAR LA SIGUIENTE VISTA:

- Agregar una tabla cualquiera al modelo de base de datos.

- Después generar una vista que maneje las 4 tablas

- La vista deberá llamarse **PARALELO_DBA_I**

```
100 CREATE TABLE paralelo (
101     id_paralelo INT(4) PRIMARY KEY,
102     cat_parc INT(4),
103     fecha DATE(10),
104     comarc VARCHAR(100)
105 );
106
107
108 INSERT INTO paralelo (id_paralelo, cat_parc, fecha, comarc)
109 VALUES
110 (1, 20, '2020-10-01', 'Paralelo 1'),
111 (2, 20, '2020-10-02', 'Paralelo 2'),
112 (3, 20, '2020-10-03', 'Paralelo 3'),
113 (4, 20, '2020-10-04', 'Paralelo 4');
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
```