

Análisis/Justificación carga

Para la carga de datos masiva empezamos por construir un script de python con el cual podemos generar una muy gran cantidad de entradas para la base de datos. Para lograr esto, creamos unos generadores que crean las consultas de inserción para las diferentes tablas y para la información dentro de cada incersion creamos información aleatoria, pero asegurándonos que tuviera sentido para el mundo del problema.

Random Word:

Este es un generador que concatena letras aleatorias. La palabra tiene una longitud igual al parámetro que le pasen.

num_reserva/num_reserva_servicio:

Num reserva y num reserva servicios son arreglos que usamos para guardar los ids de las reservas, para lograr esto, hacemos un for en un rango, asignándole un valor i a cada reserva. Esto lo usaremos más adelante para crear nuestros datos en las tablas.

Ahora, para asegurarnos que en efecto estamos manteniendo el mundo del problema creamos unos arreglos se los cuales sacaremos información aleatoriamente más adelante. Entre estos están tipo_habitacion, planes, usuarios y tipos ids.

```
55 def random_word(length):
56
57     word = ''
58     for i in range(length):
59         word += random.choice(letters)
60     return word
61
62 num_reserva = []
63 for i in range(1,1000):
64     num_reserva.append(i)
65
66 num_reserva_servicios = []
67 for i in range(1,20000):
68     num_reserva_servicios.append(i)
69
70 # Generate a random integer between 0 and 9
71 random_number = random.randint(0, 9)
72
73 # Print the random number
74 print(random_number)
75
76 tipo_habitacion = ["sencilla", "doble", "triple", "suite"]
77 letters = string.ascii_lowercase
78 planes = ["todo_incluido", "continental", "desayuno", "sin_alimentacion"]
79 nombre_elemento = []
80 usuarios = ["administrador", "cliente", "repcionista", "gerente"]
81 ids = ["pasaporte,cedula,tarjeta_de_identidad"]
82
```

Ahora empezamos con las funciones de inserción de datos, en los cuales iteramos para crear múltiples datos en un archivo externo... (las imágenes de lo que hablamos de aquí en adelante estarán adjuntas al final).

Tipos de Habitación:

Para tipos de habitación iteramos sobre los tipos definidos y hacemos entradas sobre la tabla para generar nuevos tipos de habitaciones.

Para los elementos que se encuentran en una habitación primero generamos unas palabras aleatorias que representarán los elementos que puede haber en una habitación.

Para hoteles creamos una instancia de nuestro hotel en la tabla. Como ya mencionamos en el documento de diseño, propusimos que nuestra aplicación funcionara aparte por hotel, por lo cual solo creamos la instancia de un hotel, pero realmente por la forma que está creada la base de datos puede haber más.

Habitaciones:

Para habitaciones creamos 1000 habitaciones (que nos parece un número razonable de habitaciones en un hotel dado). A cada habitación le asignamos un id y le asignamos un tipo aleatoriamente entre la lista de tipos de habitación que creamos anteriormente.

Planes:

Para planes generamos valores entre 0 y 1 para los booleanos que describen los beneficios del plan, para costo generamos un costo entre 100000 y 1000000. Así mismo, generamos un número aleatorio de días de estadía entre 1 y 14 días, que se adapta correctamente al mundo del problema.

Elementos Extra:

Para elementos extra recorremos nuestra lista de elementos creada anteriormente e iteramos para generar entradas de cada elemento en la base de datos.

Elementos extra por habitación:

Para la tabla que relaciona los elementos con el tipo de habitación, iteramos sobre nuestros tipos de habitación y luego añadimos entre 1 y 3 elementos extra por tipo de habitación, los cuales se escogen aleatoriamente entre la lista de elementos extra previamente creada. Creemos que esta es una buena aproximación a la creación de elementos por habitación,

Tipo usuario:

Para el tipo de usuario iteramos sobre la lista de usuarios que tenemos y creamos sus respectivas sentencias.

Usuarios:

Para usuarios creamos varios usuarios con un id que por simplicidad hicimos en secuencia, para el tipo de id seleccionamos un tipo de id aleatoriamente de nuestra lista de tipos de id. Para nombre generamos una palabra aleatoria, para tipo usuario seleccionamos un rol aleatorio de entre nuestra lista de usuarios, para el correo generamos una palabra aleatoria y para el numero de habitación seleccionamos un numero entre el rango que cubren los índices de habitación para asegurarnos que si sea una llave valida.

Servicios en general:

Para los servicios en general generamos un id tipo servicio donde los rangos de los ids dependen de qué servicio sea, por ejemplo, los bares están entre 1 y 9, los gimnasios entre 10 y 14 y así. Esto lo hicimos para mantener unicidad en los ids de servicio (así planteamos nuestra base de datos). De todas formas, a la hora de crear nuevos datos no es como que sea necesario que el índice esté dentro de este rango, el nuevo índice siempre seguirá la secuencia e irá al final de todos los índices de servicios (y no habrá ningún tipo de problema). Además de esto, para parámetros como estilo en bares y restaurantes creamos palabras aleatorias, para aforo creamos números aleatorios de acuerdo al contexto del servicio (es decir que para un servicio como restaurante, va a haber un mayor aforo que para algo como un spa). Para aplica compartido generamos un 0 o un 1 que indica si el plan de compartido tiene algún descuento. Para hoteles nombre incluimos el nombre del hotel y para los que aplica generamos un float aleatorio para el costo de hacer la reserva.

Productos:

Para productos generamos ids de 1 a 999, les generamos nombres aleatorios y asignamos un precio que es generado por un float aleatorio. Similar a los servicios, generamos un 0 o un 1 en caso de que

tenga un descuento por algún plan. Después tenemos la opción de agregar ids de cada servicio donde aparezca, que se generan cogiendo de los rangos de ids que conocemos (por los rangos que explicamos anteriormente), que vincularán un producto a un id.

Productos Supermercado:

Para productos supermercado seleccionamos aleatoriamente un id de algún supermercado y escogemos aleatoriamente el id de un producto para relacionarlo. Con esto conseguimos asignar a los supermercados diferentes productos.

Reservas Hoteles:

Para reservas hoteles generamos los ids de reserva en orden, generamos fechas de entrada y de salida aleatoriamente (nuestros constrains se encargan de manejar datos inconsistentes). Se generan valores entre 0 y 1 para representar si la reserva está paga y si fue ocupada, se asigna el nombre del hotel y se asigna una habitación de entre los ids posibles (el rango asignado a habitaciones) y se asigna una promoción aleatoria de entre las promociones disponibles.

Reserva Servicio:

Para reservas servicios recorremos nuestros servicios reservables y asignamos un id de reserva a cada uno. Además, asignamos una fecha aleatoria, un valor de 0 o 1 para si la reserva fue atendida o no, asignamos un numero de habitación aleatorio y asignamos opcionalmente un id de servicio que representará el servicio al cual se hizo una reserva, en caso de reservar múltiples, hay varios ids de reservas.

Consumos:

Para consumos generamos ids en orden para cada consumo, generamos un valor asociado de forma aleatoria, creamos un booleano (0 o 1) para indicar si algo fue cargado a la habitación. Además generamos una fecha de creación aleatoria y asignamos ids opcionales de los servicios asociados a dicho consumo.

```

194
195 #consumos
196 for i in range(30000):
197     entrada = str(random.randint(1,31))+"-"+str(random.randint(1,12))+"-"+str(random.randint(2018,2023))
198     print("INSERT INTO CONSUMOS (ID,COSTO,CARGADOHABITACION,FECHA,GIMNASIOS_IDTIPOSERVICIO,SALONES_IDTIPOSERVICIO,PRESTAMOS_IDTIPOSERVICIO,SUPE
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

```

```

111 #elementos extra
112 for ele in nombre_elemento:
113     print("INSERT INTO ELEMENTOEXTRAS (NOMBREELEMENTO) VALUES ('"+ele+"');")
114
115 #elementos de tipo de habitacion
116 for habitacion in tipo_habitacion:
117     for i in range(random.randint(1, 3)):
118         print("INSERT INTO ELEMENTOSTIPOSHABITACIONES (TIPOHABITACION,NOMBREELEMENTO) VALUES ('"+habitacion+"','"+nombre_elemento[random.randint(0, len(nombre_elemento)-1)]+"');")
119
120 #tipo de usuario
121 for users in usuarios:
122     print("INSERT INTO TIPOUSUARIOS (TIPOUSUARIO) VALUES ('"+users+"');")
123
124 #usuarios
125 for i in range(1,5000):
126     print("INSERT INTO USUARIOS (ID,TIPOID,NOMBRE,TIPOUSUARIO,CORREO,HABITACIONES_NUMEROHABITACION) VALUES ('"+str(i)+"','"+ids[random.randint(0, len(ids)-1)]+"','"+nombres[random.randint(0, len(nombres)-1)]+"','"+tipos[random.randint(0, len(tipos)-1)]+"','"+corros[random.randint(0, len(corros)-1)]+"','"+habitaciones[random.randint(0, len(habitaciones)-1)]+"');")
127
128 #bares
129 for i in range(1,10):
130     print("INSERT INTO BARES (IDTIPOSERVICIO,ESTILO,AFORO,APLICACOMPARTIDO,HOTELES_NOMBRE,COSTORESERVA) VALUES ('"+str(i)+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"');")
131
132 #Gimnasio
133 for i in range(10,14):
134     print("INSERT INTO GIMNASIOS (IDTIPOSERVICIO,AFORO,NUMMAQUINAS,HORARIO,APLICACOMPARTIDO,HOTELES_NOMBRE,COSTORESERVA) VALUES ('"+str(i)+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"');")
135
136 #Lavanderia
137 for i in range(14,24):
138     print("INSERT INTO LAVANDERIAS (IDTIPOSERVICIO,APLICACOMPARTIDO,HOTELES_NOMBRE,AFORO,COSTORESERVA) VALUES ('"+str(i)+"','"+str(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"','"+random_word(random.randint(5, 10))+"');")
139
140
141 #tipos de habitacion
142 for tipos in tipo_habitacion:
143     print("INSERT INTO TIPOSHABITACIONES (TIPOHABITACION) VALUES ('"+tipos+"');")
144
145 #Lista de elementos que se pueden encontrar en una habitacion
146 for i in range(1,10):
147     nombre_elemento.append(random_word(random.randint(5, 10)))
148
149 #hoteles
150 print("INSERT INTO HOTELES (NOMBRE) VALUES ('"+'Hotel de los Andes'+"');")
151
152 #habitaciones
153 for i in range(1,1000):
154     print("INSERT INTO HABITACIONES (NUMEROHABITACION,TIPOHABITACION) VALUES ('"+str(i)+"','"+tipo_habitacion[random.randint(0, 3)]+"');")
155
156 #planes
157 for plan in planes:
158     d_alojamiento = random.uniform(0, 1.0)
159     d_comida = random.uniform(0, 1.0)
160     d_servicios = random.uniform(0, 1.0)
161     costo = random.uniform(100000, 1000000)
162     dias_min_estadia = random.randint(1, 14)
163     print("INSERT INTO PROMOCIONES (NOMBREPLAN,DESCUENTOALOJAMIENTO,DESCUENTOCOMIDA,DESCUENTOSERVICIOS,COSTO,DIASMINESTADIA) VALUES ('"+plan+"','"+str(d_alojamiento)+"','"+str(d_comida)+"','"+str(d_servicios)+"','"+str(costo)+"','"+str(dias_min_estadia)+"');")

```