

```

1 ORACLE
2
3 CREATE TABLE NOMBRE_TABLA (ATRIBUTO TIPO(RANGO)
4     NULIDAD,.....);
5 ALTER TABLE NOMBRE_TABLA ADD CONSTRAINT
6     PK_NOMBRE_TABLA
7     PRIMARY KEY (ATRIBUTOS);
8 ALTER TABLE NOMBRE_TABLA ADD CONSTRAINT
9     FK_NOMBRE_TABLAS
10    FOREIGN KEY (ATRIBUTO) REFERENCES
11    TABLA_PRIMARIA(ATRIBUTO);
12 ALTER TABLE NOMBRE_TABLA ADD CONSTRAINT
13    UK_NOMBRE_TABLA_ATRIBUTO
14    UNIQUE(ATRIBUTO);
15
16 ATRIBUTOS
17
18 ALTER TABLE NOMBRE_TABLA MODIFY (ATRIBUTO CHECK (
19     CONDICION DEL ATRIBUTO));
20
21 INSERT INTO NOMBRE_TABLA (ATRIBUTOS) VALUES (
22     ATRIBUTOS,....,.);
23
24 INTEGRIDAD DECLARATIVA
25
26 * DOMINIO (DE TIPO)
27
28 CREATE DOMAIN TNOMBRE AS TIPO(RANGO) CONSTRAINT
29     CHECK(VALUE CONDITION);
30
31 * TUPLA (DE ATRIBUTO DE TUPLA)
32
33 ALTER TABLE NOMBRE_TABLA ADD CONSTRAINT
34     CK_NOMBRE_TABLA
35     CHECK(CONDICION);
36
37 * ASERCIÓN GENERAL (DE TABLA DE BASE DE DATOS)
38
39 CREATE ASSERTION NOMBRE
40     CHECK(NOT EXISTS (CONSULTA));
41
42 UPADTE, BORRAR
43
44 UPDATE NOMBRE_TABLA SET ATRIBUTO= POR LO QUE
45     CAMBIO
46     WHERE ATRIBUTO= EL QUE QUIERO CAMBIAR;
47
48 DELETE FROM NOMBRE_TABLA WHERE CONDICION;
49
50 ERROR, DECLARE, CURSOR
51
52 RAISE_APPLICATION_ERROR(NUMERO[-20001,-20999],
53     MENSAJE);
54
55 SELECT ATRIBUTO INTO VARIABLE FROM NOMBRE_TABLA
56     WHERE CONDICION;
57
58 CURSOR DATOS IS SELECT ESTADO,CIUDAD FROM
59     VENDEDORES WHERE ESTADO > 10;
60
61 ACCIONES DE REFERENCIA
62
63 CONSTRAINT FK_NOMBRE FOREIGN KEY NOMBRE_ATRIBUTO
64     REFERENCES NOMBRE_TABLA(NOMBRE_ATRIBUTO)
65     [ON DELETE (RESTRICT,CASCADE,SET NULL,NO ACTION
66     )]
67     [ON UPDATE (RESTRICT,CASCADE,SET NULL, NO
68     ACTION)]
69
70 ALTER TABLE PERTENECE_GRUPO DROP CONSTRAINT
71     FK_PERTENECE_GRUPO_USUARIO;
72
73 ALTER TABLE PERTENECE_GRUPO ADD CONSTRAINT
74     FK_PERTENECE_GRUPO_USUARIO
75     FOREIGN KEY (ID_USUARIO) REFERENCES USUARIOS(
76
77     USERNAME) ON DELETE CASCADE;
78
79 DISPARADORES
80
81 CREATE OR REPLACE TRIGGER NOMBRE
82     [BEFORE|AFTER|INSTEAD OF]
83     [DELETE,INSERT,UPADTE[OF NOMBRE_COLUMNNA]]
84     ON NOMBRE_TABLA
85     [FOR EACH ROW [WHEN(CONDICION)]]
86     DECLARE
87     NOMBRE_VARIABLE TIPO;
88 BEGIN
89     SELECT ATRIBUTO INTO NOMBRE_VARIABLE FROM TABLA
90     ;
91     IF (CONDICION) THEN
92         QUE HACE
93     END IF;
94 END NOMBRE;
95
96 := <ASIGNAR>
97 :NEW. :OLD.
98
99 ELIMINAR TODO
100 BEGIN
101     --Bye Tables!
102     FOR i IN (SELECT ut.table_name
103         FROM USER_TABLES ut) LOOP
104         EXECUTE IMMEDIATE 'drop table '|| i.
105             table_name ||' CASCADE CONSTRAINTS ';
106     END LOOP;
107 END;
108
109 ELIMINAR
110
111 ALTER TABLE NOMBRE_TABLA DROP CONSTRAINT
112     NOMBRE_CONSTRAINT;
113
114 DROP TABLE NOMBRE_TABLA;
115
116 CONCAT(STRING1,STRING2);
117
118 SELECT *
119     FROM TABLA
120     WHERE CONDICION
121     GROUP BY
122     HAVING FUNCIONES(MAX,MIN,COUNT,AVG)
123     ORDER BY;
124
125 REPLACE("MAMA-","-'','/''); = MAMA/
126 SUBSTR(CADENA,INICIA,LONGITUD);
127
128 CONTIENE DOS PALABRAS MINIMO:
129 ALTER TABLE NOMBRE ADD CONSTRAINT
130     CK_NOMBRE_TABLA
131     CHECK (REGEXP_LIKE(LTRIM(M_NAME),'\S\s\S'));
132
133 TO_DATE('1999/03/12','YYYY/MM/DD');
134 SELECT SYSDATE FROM DUAL;
135
136 VISTAS
137
138 CREATE VIEW NOMBRE_VISTA AS
139     CONSULTA;
140
141 SECUENCIAS
142
143 declare
144     l_new_seq INTEGER;
145 begin
146     select count(*) + 1
147     into l_new_seq
148     from grupos;

```

```

126     execute immediate 'Create sequence seq_grupos
127         start with ' || l_new_seq
128         ' increment by 1';
129 end;
130
131 //CRUD_E
132 CREATE OR REPLACE PACKAGE PC_GRUPO IS
133     /*GRUPO*/
134     PROCEDURE AD_GRUPO (xNOMBRE IN VARCHAR,
135         xCREADOR IN VARCHAR);
136     PROCEDURE MOD_GRUPO (xID_GRUPO IN NUMBER,
137         xNOMBRE IN VARCHAR);
138     PROCEDURE EL_PERTENECE (xID_USUARIO IN VARCHAR,
139         xID_GRUPO IN NUMBER);
140
141     FUNCTION CO_GRUPO RETURN SYS_REFCURSOR;
142     FUNCTION CO_EQUI (xNOMBRE IN VARCHAR) RETURN
143         SYS_REFCURSOR;
144
145 END PC_GRUPO;
146
147 CRUD_I
148 CREATE OR REPLACE PACKAGE BODY PC_GRUPO IS
149     PROCEDURE AD_GRUPO (xNOMBRE IN VARCHAR,
150         xCREADOR IN VARCHAR) IS
151     BEGIN
152         INSERT INTO GRUPOS (ID_GRUPO, NOMBRE, CREADOR)
153         VALUES (SEQ_GRUPOS.NEXTVAL, xNOMBRE, xCREADOR);
154         COMMIT;
155         EXCEPTION
156         WHEN OTHERS THEN
157             ROLLBACK;
158             RAISE_APPLICATION_ERROR(-20032, 'No se puede
159             insertar en grupos. ');
160     END;
161
162     PROCEDURE MOD_GRUPO (xID_GRUPO IN NUMBER,
163         xNOMBRE IN VARCHAR) IS
164     BEGIN
165         UPDATE GRUPOS SET NOMBRE= xNOMBRE WHERE
166         ID_GRUPO= xID_GRUPO;
167         COMMIT;
168         EXCEPTION
169         WHEN OTHERS THEN
170             ROLLBACK;
171             RAISE_APPLICATION_ERROR (-20021, 'No se puede
172             modificar el grupo. ');
173     END;
174
175     PROCEDURE EL_PERTENECE (xID_USUARIO IN VARCHAR,
176         xID_GRUPO IN NUMBER) IS
177     BEGIN
178         DELETE FROM PERTENECE_GRUPO WHERE ID_GRUPO=
179         xID_GRUPO AND ID_USUARIO= xID_USUARIO;
180         COMMIT;
181         EXCEPTION
182         WHEN OTHERS THEN
183             ROLLBACK;
184             RAISE_APPLICATION_ERROR(-20013, 'No se puede
185             eliminar este usuario. ');
186     END;
187
188     FUNCTION CO_GRUPO RETURN SYS_REFCURSOR IS
189     CON_GRUPO SYS_REFCURSOR;
190     BEGIN
191         OPEN CON_GRUPO FOR
192             SELECT DISTINCT USERNAME, U.NOMBRE, ID_GRUPO,
193             G.NOMBRE FROM USUARIOS U, GRUPOS G
194             WHERE G.CREADOR= USERNAME;
195         RETURN CON_GRUPO;
196
197     END;
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

```

```

239 WHEN OTHERS THEN
240     ROLLBACK;
241     RAISE_APPLICATION_ERROR(-20999,SQLERRM);
242 END;
243
244 XPATH
245
246 /Course_Catalog/Department/Chair[not(Professor/
    Last_Name/text() = parent::*/Course/
    Instructors/Professor/Last_Name/text())]/
    Professor/Last_Name/text()
247 /Course_Catalog/Department/Course/Instructors[
    count(*)=min(/Course_Catalog/Department/
    Course/Instructors/count(*))]/parent::Course/
    @Numbe
248
249 EN SQL
250 SELECT A.INCIDENTE,T.concecuencia
251 FROM ACCIDENTES A, XMLTABLE('/Tdetalle/
    Consecuencias/Consecuencia'
252     passing A.detalle
253     columns concecuencia VARCHAR(200)
    PATH 'text()') T;
254
255
256 SELECT CASE
257     WHEN condicion THEN accion cuando cumple
    el when END;
258 FROM name_table
259
260
261 CREATE ASSERTION REPARTIDO
262 CHECK (NOT EXISTS (SELECT * FROM ENVIOS
263     GROUP BY VENDEDOR HAVING COUNT(*) > 100))
264
265 ALTER TABLE USUARIOS MODIFY ( ESTADO CHECK(
    ESTADO LIKE 'Pasivo' OR ESTADO LIKE 'Activo')
    );
266
267
268 FUNCTION CO_MUSICO RETURN XMLTYPE;
269 FUNCTION CO_MUSICO RETURN XMLTYPE IS CO_XML
    XMLTYPE;
270 BEGIN
271     CO_XML := dbms_xmlgen.getxmltype(
272         'SELECT * FROM MUSICIAN');
273     RETURN CO_XML;
274 END;

```

```

24 <!ELEMENT empresa (#PCDATA)>
25 <!--ATTLIST empresa
26     nombre CDATA #REQUIRED
27     pais CDATA #REQUIRED
28     ciudad CDATA #REQUIRED>
29 <!--ELEMENT proyectos (proyecto*)>
30 <!--ELEMENT proyecto (fecha,pagina?,detalle?)>
31 <!--ATTLIST proyecto
32     nombre CDATA #REQUIRED
33     tema CDATA #REQUIRED>
34 <!--ELEMENT pagina (#PCDATA)>
35 <!--ATTLIST pagina
36     nombre CDATA #REQUIRED
37     url CDATA #REQUIRED>
38 <!--ELEMENT detalle (#PCDATA)>
39 ]>

1 /*TODAS LAS ASIGNACIONES SON A = " A" ENTRE LAS
    ETIQUETAS*/
2 <?xml version="1.0"?>
3 <Perfil>
4     <estudios>
5         <estudio nombre = Ingenieria de sistemas>
6             <universidad
7                 nombre= Escuela Colombiana de Ingenieria
8                 pais= Colombia
9                 ciudad = Bogota >
10             </universidad>
11             <fecha
12                 inicio= 2016-2 >
13             </fecha>
14         </estudio>
15     </estudios>
16     <proyectos>
17         <proyecto nombre= Procommunity tema= Bases de
            datos>
18             <fecha
19                 inicio= 2017-2 >
20             </fecha>
21             <detalle>
22                 Nuestra mision es almacenar preguntas y
                soluciones, para que la comunidad de
                programadores competitivos tenga una
                retroalimentacion entre ellos y de esta
                forma mejorar el rendimiento de las personas,
                de igual manera se tendran
                registros de los procesos de cada usuario
                , para que las organizaciones conozcan el
                avance de cada persona.
23             </detalle>
24         </proyecto>
25         <proyecto nombre= TEAM tema= Programacion>
26             <fecha
27                 inicio= 2017-2 >
28             </fecha>
29             <detalle>
30                 Nuestra mision es tener un repositorio
31                 con algunos problemas de programacion
32                 y sus respectivas soluciones con videos
33                 de explicacion para generar una nueva
34                 forma de ayuda a los nuevos y antiguos
35                 programadores.
36             </detalle>
37         </proyecto>
38     </proyectos>
39 </Perfil>

```