

**Autonomous University of Zacatecas**

ACADEMIC UNIT OF ELECTRICAL ENGINEERING

ACADEMIC PROGRAM OF SOFTWARE ENGINEERING



**DATABASE SYSTEMS LABORATORY II PRACTICE 4 -  
DDL2**

**PROFESSOR:  
ALDONSO BECERRA SÁNCHEZ**

**STUDENT:  
DANIEL ALEJANDRO MORALES CASTILLO**

# 1 Introduction

The Oracle DDL language is transcendental in the handling of SQL statements at the level of both administrator and database programmer, since it allows the definition of database schemes regardless of the platform used to generate it.

The DDL (Data definition Language) is a language provide by the database management system and it allows the programmers do fundamental works handling of SQL statements. This practice works for review some theoretical concepts

During these days we saw the topic of synonyms, indices and sequences, in this practice we will apply these new concepts for the creation of databases and our tables

## 2 Development

### Activity 1

Write the section that describes the work developed in the following activities. Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question. Explain the reason for your answer.

#### CREATE PRIVATE AND PUBLIC SYNONYMS

**1. What are distinguishing characteristics of a public synonym rather than a private synonym?**

- A. Public synonyms are always visible to all users.
- B. Public synonyms can be accessed by name without a schema name qualifier.
- C. Public synonyms can be selected from without needing any permission.
- D. Public synonyms can have the same names as tables or views.

My answer: A or B

In theory class we learned about synonyms and a public synonym characteristic. is that all users can see it, also it is not necessary to specify a name to access the public synonym

**2. Consider these three statements:**

```
create synonym s1 for staff;  
create public synonym s1 for warehouse;  
select * from s1;
```

Which of the following statements is correct? (Choose the best answer.)

- A. The second statement will fail because an object S1 already exists.
- B. The third statement will show the contents of warehouse.
- C. The third statement will show the contents of staff.
- D. The third statement will show the contents of the table S1, if such a table exists in the current schema.

My answer: C

You can create public and private synonyms, the public synonym will be useful To use it in other schemas, you can create a public and private synonym with the same name, but when you select all of the synonym, the query will show you the synonym private information because those synonyms are only for that schema and you are querying from that schema, so the private synonym will show it first

**3. A view and a synonym are created as follows: create view dept v as select \* from dept; create synonym dept s for dept v; Subsequently the table DEPT is dropped. What will happen if you query the synonym DEPT S? (Choose the best answer.)**

- A. There will not be an error because the synonym addresses the view, which still exists, but there will be no rows returned.
- B. There will not be an error if you first recompile the view with the command ALTER VIEW DEPT V COMPILE FORCE;
- C. There will be an error because the synonym will be invalid.
- D. There will be an error because the view will be invalid.
- E. There will be an error because the view will have been dropped implicitly when the table was dropped.

My answer: D

There will be an error creating a table and a synonym for that table and query the synonym after deleting the table, there will be an error. I think the same will happen after removing the table because a view is a representation of a table and if you remove it the view will be invalid.

#### **CREATE, MAINTAIN, AND USE SEQUENCES**

**4. A sequence is created as follows: create sequence seq1 maxvalue 100; If the current value is already 100, when you attempt to select SEQ1.NEXTVAL what will happen? (Choose the best answer.)**

- A. The sequence will cycle and issue 0.
- B. The sequence will cycle and issue 1.
- C. The sequence will reissue 100.
- D. There will be an error.

My answer: D

When you create a sequence, you must specify the conditions for the sequence, there are some optional conditions like loop, if you didn't specify a loop for your sequence then, by default, will not cycle, so there will be an error.

**5. You create a sequence as follows: create sequence seq1 start with 5; After selecting from it a few times, you want to reinitialize it to reissue the numbers already generated. How can you do this? (Choose the best answer.)**

- A. You must drop and re-create the sequence.
- B. You can't. Under no circumstances can numbers from a sequence be reissued once they have been used.
- C. Use the command ALTER SEQUENCE SEQ1 START WITH 5; to reset the next value to 5.
- D. Use the command ALTER SEQUENCE SEQ1 CYCLE; to reset the sequence to its starting value.

My answer: A

The reason for my answer: We talked about sequences in theory class and the easiest way to reset a sequence is to drop it and recreate it. so you can use it started again

**6. Assuming that the sequence SEQ1 was created with the option ORDER and INCREMENT BY set to 1, what value will be returned by the final SELECT statement? (Choose the best answer.)**

- A. 2
- B. 3
- C. 4
- D. It will depend on whether any other sessions are selecting from the sequence while the statements in the exhibit are being run.

My answer: A

I think it is 2 because a rollback discard all the DB changes and the last change in the data base was the insert using the sequence so it is like the sequence was never used.

## CREATE AND MAINTAIN INDEXES

**7. A UNIQUE constraint on a column requires an index. Which of the following scenarios is correct? (Choose one or more correct answers.)**

- A. If a UNIQUE index already exists on the column, it will be used.
- B. If a NONUNIQUE index already exists it will be used.
- C. If a NONUNIQUE index already exists on the column, a UNIQUE index will be created implicitly.
- D. If any index exists on the column, there will be an error as Oracle attempts to create another index implicitly.

My answer: A

The primary key and unique constraints on the columns require an index, so if a The UNIQUE index already exists we can use it, if we need another index we can create it.

**8. This statement will fail: create unique bitmap index on employees(department id,hire date); Why? (Choose the best answer.)**

- A. Bitmap indexes cannot be unique.
- B. The two columns are of different data types.
- C. A bitmap index can be on only one column.
- D. There is already a B\*Tree index on DEPARTMENT\_ID

My answer: D

The syntax is correct, it can be UNIQUE, it can use different columns, the only one reason to fail it is because a different index already exist.

**9. You have created an index with this statement: create index ename i on employees(last name,first name); How can you adjust the index to include the employees' birthdays, which is a date type column called DOB? (Choose the best answer.)**

- A. Use ALTER INDEX ENAME I ADD COLUMN DOB;
- B. You can't do this because of the data type mismatch.
- C. You must drop the index and re-create it.
- D. This can only be done if the column DOB is NULL in all existing rows.

My answer: C

The easy way to add a column is dropping the index and later re create it again to re write the code of the CREATE INDEX. I was researching and i did not find how to add a column to the index using ALTER INDEX.

## Activity 2:

Consider the following context issue:

SHOP (*shop<sub>i</sub>d, address, manager*).

PRODUCT (*productid, pname, sale<sub>p</sub>rice, purchaseprice, provider*).

CHANNEL (*channelid, cname* ).

EMPLOYEE (*empid, empname, emplastn, bossid, address, dateofbirth, gender, beneficiaries*)

- Write code to create the tables (only columns data type specifications);
- Create indexes (choose appropriate type);
- Create constraints (all that it needs).
- Create sequences to be used for primary keys where necessary with the best options.
- Create short name synonymous for each table.

Code ins SQL DEVELOPER as evidence of my work developed



```
/*CHANNEL TABLE*/
CREATE TABLE F4_CHANNEL(
  CHANNEL_ID NUMBER(4),
  CNAME VARCHAR2(25)
);
--ALTERS FOR CREATE CONSTRAINTS OF PRIMARY KEYS
ALTER TABLE F4_CHANNEL ADD CONSTRAINT CHANNEL_ID_PK PRIMARY KEY(CHANNEL_ID);

/*EMPLOYEE TABLE*/
CREATE TABLE F4_EMPLOYEE(
  EMP_ID NUMBER(4),
  EMP_NAME VARCHAR2(25),
  EMP_LASTN VARCHAR2(25),
  BOSS_ID NUMBER(4),
  ADDRESS VARCHAR2(50),
  DATE_OF_BIRTH DATE,
  GENDER VARCHAR2(15),
  BENEFICIARIES VARCHAR2(25)
);
--ALTERS FOR CREATE CONSTRAINTS OF PRIMARY KEYS
ALTER TABLE F4_EMPLOYEE ADD CONSTRAINT EMPLOYEE_ID_PK PRIMARY KEY(EMP_ID);

/*PRODUCT TABLE*/
CREATE TABLE F4_PRODUCT(
  PRODUCT_ID NUMBER(4),
  PNAME VARCHAR2(25),
  SALE_PRICE VARCHAR2(25),
  PURCHASE_PRICE VARCHAR2(25),
  PROVIDER VARCHAR2(25)
);
--ALTERS FOR CREATE CONSTRAINTS OF PRIMARY KEYS
ALTER TABLE F4_PRODUCT ADD CONSTRAINT PRODUCT_ID_PK PRIMARY KEY(PRODUCT_ID);

CREATE TABLE F4_SHOP(
  SHOP_ID NUMBER(4),
  ADDRESS VARCHAR2(25),
  MANAGER_SHOP VARCHAR2(25)
);
--ALTERS FOR CREATE CONSTRAINTS OF PRIMARY KEYS
ALTER TABLE F4_SHOP ADD CONSTRAINT SHOP_ID_PK PRIMARY KEY(SHOP_ID);

/*SALES TABLE*/
CREATE TABLE F4_SALES(
  SALE_ID NUMBER(4),
```



```

EMP_ID NUMBER(4),
SALE_DATE DATE
);

--ALTERS FOR CREATE CONSTRAINTS OF PRIMARY KEYS
ALTER TABLE P4_SALES ADD CONSTRAINT SALES_ID_PK PRIMARY KEY(SALE_ID);

--ALTERS FOR CREATE CONSTRAINTS OF FOREIGN KEYS
ALTER TABLE P4_SALES ADD CONSTRAINT CHANNEL_ID_FK FOREIGN KEY (CHANNEL_ID) REFERENCES P4_CHANNEL(CHANNEL_ID);
ALTER TABLE P4_SALES ADD CONSTRAINT PRODUCT_ID_FK FOREIGN KEY (PRODUCT_ID) REFERENCES P4_PRODUCT(PRODUCT_ID);
ALTER TABLE P4_SALES ADD CONSTRAINT SHOP_ID_FK FOREIGN KEY (SHOP_ID) REFERENCES P4_SHOP(SHOP_ID);
ALTER TABLE P4_SALES ADD CONSTRAINT EMP_ID_FK FOREIGN KEY (EMP_ID) REFERENCES P4_EMPLOYEE(EMP_ID);

--CREATE INDEX AND SYNONYMS
CREATE INDEX P4_EMP_NOM_INDX ON P4_EMPLOYEE(EMP_NAME);
CREATE INDEX P4_PRO_NAM_INDX ON P4_PRODUCT(ENAME);

CREATE SEQUENCE SEQUENCE_SALE_ID
INCREMENT BY 1
START WITH 1;

CREATE SEQUENCE SEQUENCE_CHANNEL_ID
INCREMENT BY 1
START WITH 1;

CREATE SEQUENCE SEQUENCE_PRODUCT_ID
INCREMENT BY 1
START WITH 1;

CREATE SEQUENCE SEQUENCE_SHOP_ID
INCREMENT BY 1
START WITH 1;

CREATE SEQUENCE SEQUENCE_EMPLOYEE_ID
INCREMENT BY 1
START WITH 1;

CREATE SYNONYM P4_CH FOR P4_CHANNEL;
CREATE SYNONYM P4_EMP FOR P4_EMPLOYEE;
CREATE SYNONYM P4_PROD FOR P4_PRODUCT;
CREATE SYNONYM P4_SH FOR P4_SHOP;
CREATE SYNONYM P4_S FOR P4_SALES;

```

### Activity 3:

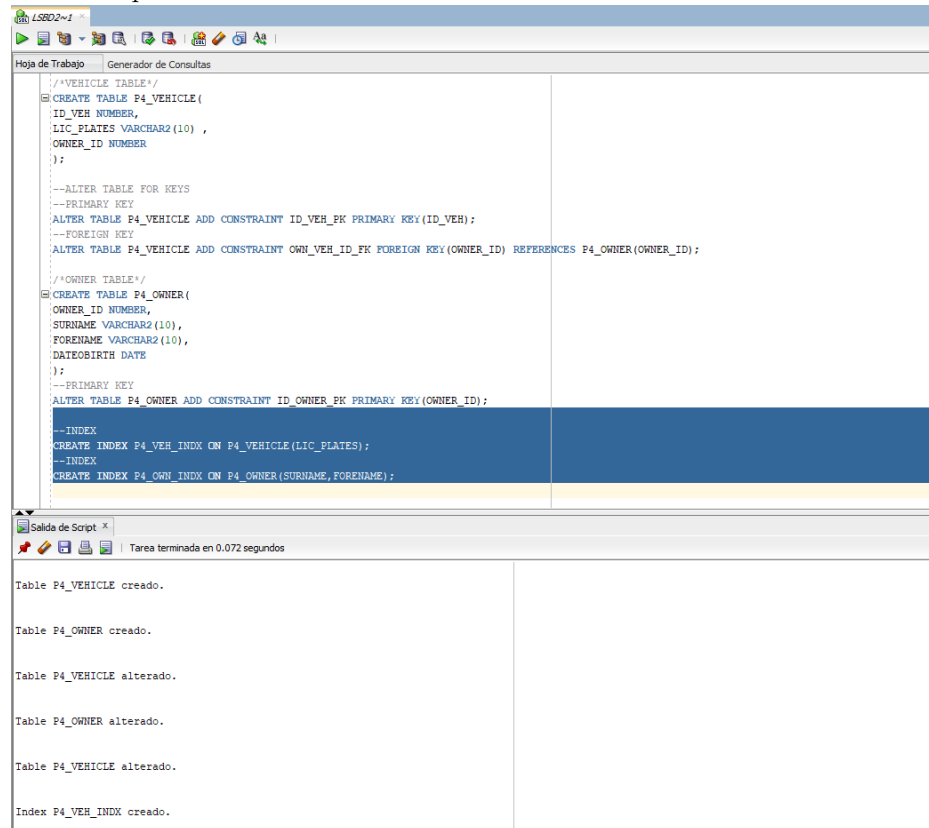
Consider these tables:

```
create table vehicle(id_veh number, lic_plates varchar2(10),
owner_id number);

create table owner(owner_id number, surname varchar2(10),
forename varchar2(10), dateobirth date);
```

Reports often require information from surname and forename of owners, reports from vehicle license plates, vehicle license plates of specific people.

1. Define required indexes (choose adequate type).
2. Add required constraints.



The screenshot shows the SQL Developer interface with a script titled 'LSRD2~1'. The script defines two tables, P4\_VEHICLE and P4\_OWNER, with their respective columns and data types. It then adds primary and foreign key constraints. Finally, it creates two indexes: P4\_VEH\_INDX on the P4\_VEHICLE table for the LIC\_PLATES column, and P4\_OWN\_INDX on the P4\_OWNER table for the SURNAME and FORENAME columns. The execution log at the bottom shows the successful completion of these operations.

```
/*VEHICLE TABLE*/
CREATE TABLE P4_VEHICLE(
  ID_VEH NUMBER,
  LIC_PLATES VARCHAR2(10) ,
  OWNER_ID NUMBER
);
--ALTER TABLE FOR KEYS
--PRIMARY KEY
ALTER TABLE P4_VEHICLE ADD CONSTRAINT ID_VEH_FK PRIMARY KEY(ID_VEH);
--FOREIGN KEY
ALTER TABLE P4_VEHICLE ADD CONSTRAINT OWN_VEH_ID_FK FOREIGN KEY(OWNER_ID) REFERENCES P4_OWNER(OWNER_ID);

/*OWNER TABLE*/
CREATE TABLE P4_OWNER(
  OWNER_ID NUMBER,
  SURNAME VARCHAR2(10),
  FORENAME VARCHAR2(10),
  DATEOBIRTH DATE
);
--PRIMARY KEY
ALTER TABLE P4_OWNER ADD CONSTRAINT ID_OWNER_FK PRIMARY KEY(OWNER_ID);

--INDEX
CREATE INDEX P4_VEH_INDX ON P4_VEHICLE(LIC_PLATES);
--INDEX
CREATE INDEX P4_OWN_INDX ON P4_OWNER(SURNAME,FORENAME);
```

Salida de Script x

Tarea terminada en 0.072 segundos

Table P4\_VEHICLE creado.

Table P4\_OWNER creado.

Table P4\_VEHICLE alterado.

Table P4\_OWNER alterado.

Table P4\_VEHICLE alterado.

Index P4\_VEH\_INDX creado.

## Activity 4:

Propose a response to the following issues:

- You are involved in designing a database to be used for online order entry and offline financial reporting. What should you consider with regard to data consulting, synonyms, and indexes?
- Should sequences always be used for primary keys?

### ANSWERS:

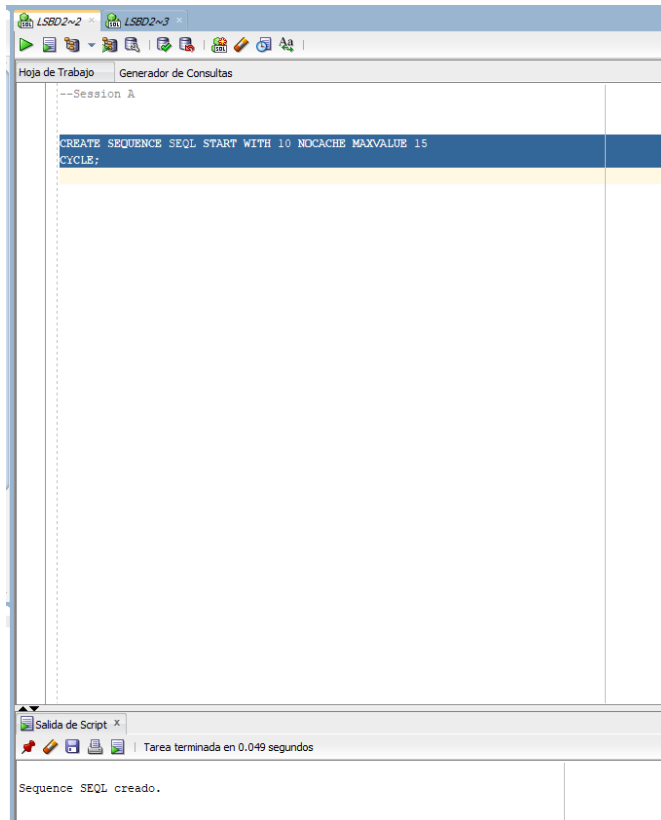
Point 1: In a large database like this or in any relatively large database we must consider these objects as fundamental to work because they help us do a better job be more efficient and create quality databases

Point 2: Yes, because you could be working on a large database and you need to have one setup, the sequence helps us to follow an order with the primary keys.

## Activity 5:

In this exercise, you will create some sequences and use them. You will need two concurrent sessions.

1. Log on to your database twice, as YOURSCHEME in separate sessions. Consider one to be your A session and the other to be your B session.
2. In your A session, create a sequence as follows: create sequence seq1 start with 10 nocache maxvalue 15 cycle; The use of NOCACHE is deleterious for performance. If MAXVALUE is specified, then CYCLE will be necessary to prevent errors when MAXVALUE is reached.



3. Execute the following commands in the appropriate session in the correct order to observe the use of NEXTVAL and CURRVAL and the cycling of the sequence:

The screenshot displays the Oracle SQL Developer interface. At the top, there are two tabs labeled 'LSBD2~2' and 'LSBD2~3'. Below the tabs is a toolbar with various icons. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains a script for 'Session A' with the following SQL commands:

```
--Session A

CREATE SEQUENCE SEQL START WITH 10 NOCACHE MAXVALUE 15
CYCLE;

--COMMANDS PRACS

SELECT SEQL.NEXTVAL FROM DUAL;

SELECT SEQL.NEXTVAL FROM DUAL;

SELECT SEQL.CURRVAL FROM DUAL;











SELECT SEQL.NEXTVAL FROM DUAL;

SELECT SEQL.NEXTVAL FROM DUAL;
```

The last line of the script is highlighted in yellow. Below the script editor, there is a section for the execution results. It includes a 'Salida de Script' tab and a 'Resultado de la Consulta' tab. The 'Resultado de la Consulta' tab shows the results of the last query executed, which is 'SELECT SEQL.NEXTVAL FROM DUAL;'. The results are displayed in a table with one row and one column, showing the value '1'.

NEXTVAL
1

LSBD2~2 × LSBD2~3 ×



Hoja de Trabajo | Generador de Consultas

--Session B

--COMMANDS

SELECT SEQL.NEXTVAL FROM DUAL;





SELECT SEQL.NEXTVAL FROM DUAL;

SELECT SEQL.NEXTVAL FROM DUAL;

SELECT SEQL.CURRVAL FROM DUAL;

SELECT SEQL.NEXTVAL FROM DUAL;

Resultado de la Consulta ×

 SQL | Todas las Filas Recuperadas: 1 en 0.006 segundos

	NEXTVAL
1	2

4. Create a table with a primary key: create table seqtest(c1 number,c2 varchar2(10)); alter table seqtest add constraint seqtest<sub>pk</sub>primarykey(c1);

```
--4
CREATE TABLE SEQTEST (
  C1 NUMBER,
  C2 VARCHAR2(10)
);
ALTER TABLE SEQTEST ADD CONSTRAINT SEQTEST_PK PRIMARY KEY (C1);
```

Salida de Script x  
Tarea terminada en 0.069 segundos

Sequence SEQL creado.  
Table SEQTEST creado.  
Table SEQTEST alterado.

5. Create a sequence to generate primary key values: create sequence seqtest<sub>pk</sub><sub>s</sub>;

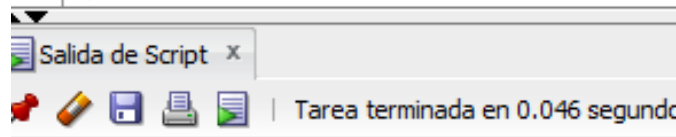
```
--5
CREATE SEQUENCE SEQTEST_PK_S;
--6
INSERT INTO SEQTEST VALUES (
  SEQTEST_PK_S.NEXTVAL, 'FIRST'
);
```

Salida de Script x  
Tarea terminada en 0.047 segundos

Sequence SEQTEST\_PK\_S creado.

6. In your A session, insert a row into the new table and commit: insert into seqtest values(seqtest<sub>pk</sub>.nextval,'first');commit;

```
--6
INSERT INTO SEQTEST VALUES (
  SEQTEST_PK_S.NEXTVAL, 'FIRST'
);
```

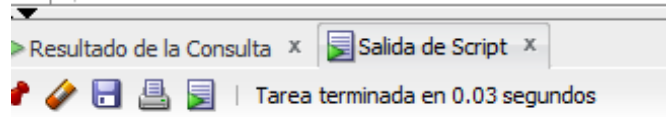


sequence SEQTEST\_PK\_S creado.

. fila insertadas.

7. In your B session, insert a row into the new table and do not commit it: insert into seqtest values(seqtest<sub>pk</sub>.nextval,'second');

```
--7
INSERT INTO SEQTEST VALUES (
  SEQTEST_PK_S.NEXTVAL, 'SECOND'
);
```

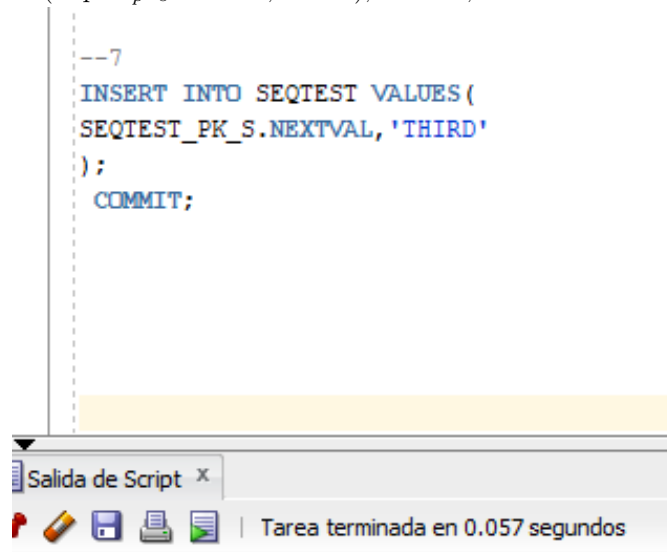


. fila insertadas.



8. In your A session, insert a third row and commit: insert into seqtest values(seqtest<sub>pk<sub>s</sub></sub>.nextval,'third');commit;

```
--7
INSERT INTO SEQTEST VALUES (
SEQTEST_PK_S.NEXTVAL, 'THIRD'
);
COMMIT;
```



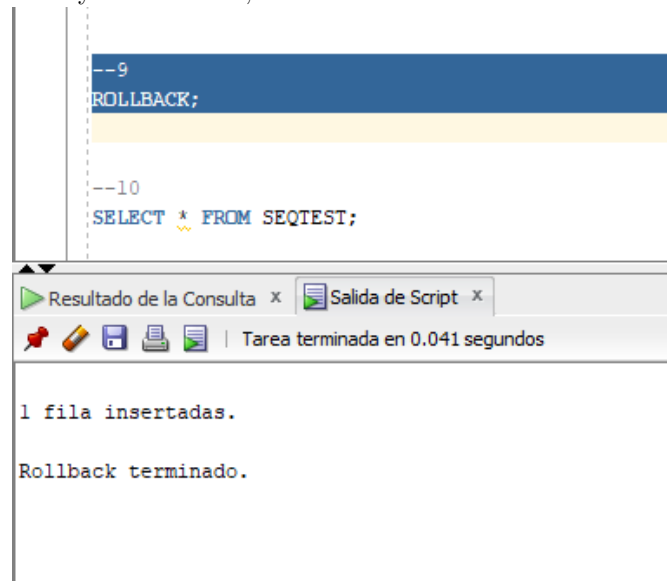
fila insertadas.

confirmación terminada.

9. In your B session, roll back the second insertion: rollback;

```
--9
ROLLBACK;
```

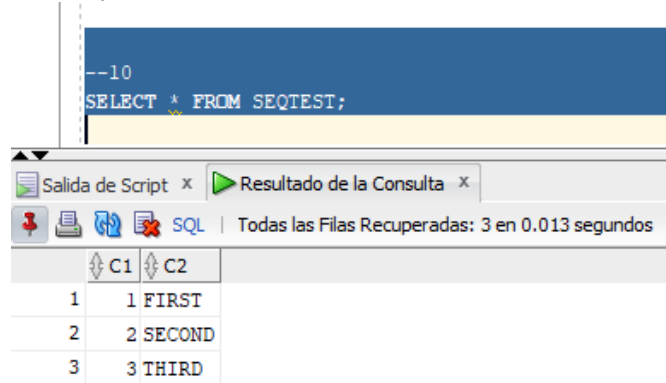
```
--10
SELECT * FROM SEQTEST;
```



1 fila insertadas.

Rollback terminado.

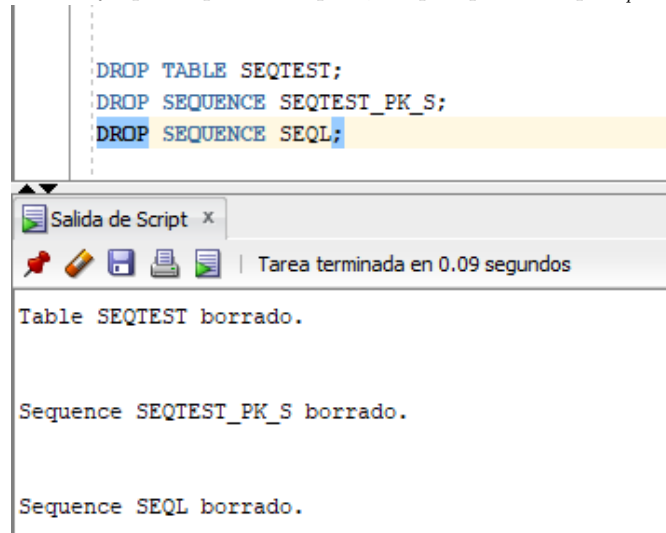
10. In your B session, see the contents of the table: `select * from seqtest;` This demonstrates that sequences are incremented and the next value published immediately, outside the transaction control mechanism.



The screenshot shows a SQL IDE window with a script editor and a results pane. The script editor contains the query `SELECT * FROM SEQTEST;`. The results pane shows a table with two columns, C1 and C2, and three rows of data.

	C1	C2
1	1	FIRST
2	2	SECOND
3	3	THIRD

11. Tidy up: `drop table seqtest;` `drop sequence seqtestpks;` `drop sequence seq1;`



The screenshot shows a SQL IDE window with a script editor and a results pane. The script editor contains the following statements: `DROP TABLE SEQTEST;`, `DROP SEQUENCE SEQTEST_PK_S;`, and `DROP SEQUENCE SEQ1;`. The results pane shows the execution status of these statements.

Statement	Result
<code>DROP TABLE SEQTEST;</code>	Table SEQTEST borrado.
<code>DROP SEQUENCE SEQTEST_PK_S;</code>	Sequence SEQTEST_PK_S borrado.
<code>DROP SEQUENCE SEQ1;</code>	Sequence SEQ1 borrado.

## Activity 6:

Execute the following sentences and include an image for each one:  
**Creating Indexes**

1. Connect to your database as your user.
2. Create a table that is a copy of HR.EMPLOYEES: create table emps as select \* from hr.employees; This table will have neither indexes nor primary, unique, or foreign key constraints, because these are not copied by a CREATE TABLE AS command. The NOT NULL constraints will have been copied. Confirm this by describing the table: describe emps;

The screenshot shows the SQL Developer interface. The top pane displays the following SQL commands:

```
--1  
--2  
CREATE TABLE EMPS AS SELECT * FROM HR.EMPLOYEES;  
DESCRIBE EMPS;
```

The bottom pane shows the output of the commands:

```
Tarea terminada en 0.104 segundos  
  
SALARY          NUMBER(8,2)  
COMMISSION_PCT  NUMBER(2,2)  
MANAGER_ID      NUMBER(6)  
DEPARTMENT_ID   NUMBER(4)  
  
Table EMPS creado.  
  
Nombre          ¿Nulo?  Tipo  
-----  
EMPLOYEE_ID     NUMBER(6)  
FIRST_NAME      VARCHAR2(20)  
LAST_NAME       NOT NULL VARCHAR2(25)  
EMAIL           NOT NULL VARCHAR2(25)  
PHONE_NUMBER    VARCHAR2(20)  
HIRE_DATE       NOT NULL DATE  
JOB_ID          NOT NULL VARCHAR2(10)  
SALARY          NUMBER(8,2)  
COMMISSION_PCT  NUMBER(2,2)  
MANAGER_ID      NUMBER(6)  
DEPARTMENT_ID   NUMBER(4)
```

3. Create an index to be used for the primary key constraint: create unique index emps empid i on emps(employee id);

The screenshot shows the Oracle SQL Developer interface. The top pane contains the following SQL commands:

```

describe employees;
--1
--2
CREATE TABLE EMPS AS SELECT * FROM HR.EMPLOYEES;
DESCRIBE EMPS;
CREATE UNIQUE INDEX EMPS_EMPID_I ON EMPS(EMPLOYEE_ID);

```

The bottom pane shows the output of the last command, indicating that the table and index were created successfully. The output includes the structure of the EMPS table and the creation of the unique index.

```

DEPARTMENT_ID          NUMBER(4)

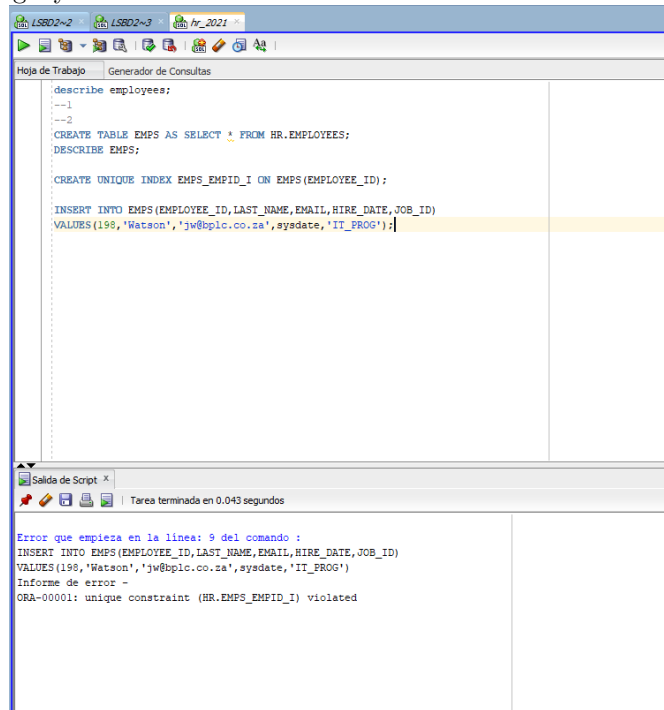
Table EMPS creado.

Nombre      ¿Nulo?  Tipo
-----
EMPLOYEE_ID          NUMBER(6)
FIRST_NAME           VARCHAR2(20)
LAST_NAME            NOT NULL VARCHAR2(25)
EMAIL                NOT NULL VARCHAR2(25)
PHONE_NUMBER         VARCHAR2(20)
HIRE_DATE            NOT NULL DATE
JOB_ID               NOT NULL VARCHAR2(10)
SALARY               NUMBER(8,2)
COMMISSION_PCT        NUMBER(2,2)
MANAGER_ID           NUMBER(6)
DEPARTMENT_ID        NUMBER(4)

INDEX EMPS_EMPID_I creado.

```

4. Demonstrate that a unique index cannot accept duplicates, even before a constraint is defined: This will return an error because the index cannot insert a second employeeid 198. Index uniqueness is an attribute of the index that can exist without a constraint but should not be relied upon to enforce data integrity.



The screenshot shows the Oracle SQL Developer interface. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains a SQL script with the following lines:

```
--1  
--2  
CREATE TABLE EMPS AS SELECT * FROM HR.EMPLOYEES;  
DESCRIBE EMPS;  
  
CREATE UNIQUE INDEX EMPS_EMPID_I ON EMPS(EMPLOYEE_ID);  
  
INSERT INTO EMPS(EMPLOYEE_ID, LAST_NAME, EMAIL, HIRE_DATE, JOB_ID)  
VALUES(198, 'Watson', 'jw@bpic.co.za', sysdate, 'IT_PROG');
```

The script is executed, and the output window at the bottom shows the following error message:

```
Error que empieza en la línea: 9 del comando :  
INSERT INTO EMPS(EMPLOYEE_ID, LAST_NAME, EMAIL, HIRE_DATE, JOB_ID)  
VALUES(198, 'Watson', 'jw@bpic.co.za', sysdate, 'IT_PROG')  
Informe de error -  
ORA-00001: unique constraint (HR.EMPS_EMPID_I) violated
```

The error message indicates that the unique constraint (HR.EMPS\_EMPID\_I) was violated because the EMPLOYEE\_ID 198 already exists in the table.

5. Create additional indexes on columns that are likely to be used in WHERE clauses, using B\*Tree for columns of high cardinality and bitmap for columns of low cardinality: create index emps name i on emps(last name,first name); create index emps tel i on emps(phone number); create bitmap index emps mgr i on emps(manager id); create bitmap index emps dept i on emps(department id);

```

--1
--2
--3
--4
--5
--6
--7
--8
--9
--10
--11
--12
--13
--14
--15
--16
--17
--18
--19
--20
--21
--22
--23
--24
--25
--26
--27
--28
--29
--30
--31
--32
--33
--34
--35
--36
--37
--38
--39
--40
--41
--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

```

I did not know that bitmap was not available for the standart edition and I got those errors

6. Define some constraints: alter table emps add constraint emps pk primary key (employee id); alter table emps add constraint emps email uk unique(email); alter table emps add constraint emps tel uk unique(phone number);

```
--1
--2
CREATE TABLE EMPS AS SELECT * FROM HR.EMPLOYEES;
DESCRIBE EMPS;
--3
CREATE UNIQUE INDEX EMPS_EMPID_I ON EMPS(EMPLOYEE_ID);
--4
INSERT INTO EMPS(EMPLOYEE_ID, LAST_NAME, EMAIL, HIRE_DATE, JOB_ID)
VALUES(198, 'Watson', 'jw@bplic.co.za', sysdate, 'IT_PROG');
--5
CREATE INDEX EMPS_NAME_I ON EMPS(LAST_NAME, FIRST_NAME);
CREATE INDEX EMPS_TEL_I ON EMPS(PHONE_NUMBER);
CREATE BITMAP INDEX EMPS_MGR_I ON EMPS(MANAGE_ID);
CREATE BITMAP INDEX EMPS_DEPT_I ON EMPS(DEPARTMENT_ID);

--6DEFINE CONSTRAINTS
ALTER TABLE EMPS ADD CONSTRAINT EMPS_PK PRIMARY KEY(EMPLOYEE_ID);
ALTER TABLE EMPS ADD CONSTRAINT EMPS_EMAIL_UK UNIQUE(EMAIL);
ALTER TABLE EMPS ADD CONSTRAINT EMPS_TEL_UK UNIQUE(PHONE_NUMBER);
```

```
Salida de Script x
Tarea terminada en 0.086 segundos

Error que empieza en la línea: 15 del comando :
CREATE BITMAP INDEX EMPS_MGR_I ON EMPS(MANAGE_ID)
Informe de error -
ORA-00439: feature not enabled: Bit-mapped indexes
00439. 00000 - "feature not enabled: %s"
*Cause: The specified feature is not enabled.
*Action: Do not attempt to use this feature.

Error que empieza en la línea: 16 del comando :
CREATE BITMAP INDEX EMPS_DEPT_I ON EMPS(DEPARTMENT_ID)
Informe de error -
ORA-00439: feature not enabled: Bit-mapped indexes
00439. 00000 - "feature not enabled: %s"
*Cause: The specified feature is not enabled.
*Action: Do not attempt to use this feature.

Table EMPS alterado.

Table EMPS alterado.

Table EMPS alterado.
```

7. Display the index names and their type: select index name,index type,uniqueness from user indexes where table name='EMPS'; The view USER INDEXES shows details of all indexes in your current schema. Note that in addition to the five indexes explicitly created in steps 3 and 5, there is also an index created implicitly with the name of the constraint defined on EMAIL. Note also that the unique constraint on PHONE NUMBER is being enforced with a nonunique index; this is perfectly possible, because although the constraint mechanism uses indexes, it is independent of the structure of the index.

The screenshot shows an Oracle SQL Developer window with the following content:

**SQL Code:**

```

INSERT INTO EMPS (EMPLOYEE_ID, LAST_NAME, EMAIL, HIRE_DATE, JOB_ID)
VALUES (198, 'Watson', 'jw@bplic.co.za', sysdate, 'IT_FROG');

--5
CREATE INDEX EMPS_NAME_I ON EMPS (LAST_NAME, FIRST_NAME);
CREATE INDEX EMPS_TEL_I ON EMPS (PHONE_NUMBER);
CREATE BITMAP INDEX EMPS_MGR_I ON EMPS (MANAGE_ID);
CREATE BITMAP INDEX EMPS_DEPT_I ON EMPS (DEPARTMENT_ID);

--DEFINE CONSTRAINTS
ALTER TABLE EMPS ADD CONSTRAINT EMPS_PK PRIMARY KEY (EMPLOYEE_ID);
ALTER TABLE EMPS ADD CONSTRAINT EMPS_EMAIL_UK UNIQUE (EMAIL);
ALTER TABLE EMPS ADD CONSTRAINT EMPS_TEL_UK UNIQUE (PHONE_NUMBER);

--7
SELECT INDEX_NAME, INDEX_TYPE, UNIQUENESS FROM USER_INDEXES
WHERE TABLE_NAME='EMPS';

```

**Execution Results:**

Resultado de la Consulta: Todas las Filas Recuperadas: 4 en 0.009 segundos

	INDEX_NAME	INDEX_TYPE	UNIQUENESS
1	EMPS_EMPID_I	NORMAL	UNIQUE
2	EMPS_NAME_I	NORMAL	NONUNIQUE
3	EMPS_TEL_I	NORMAL	NONUNIQUE
4	EMPS_EMAIL_UK	NORMAL	UNIQUE



8. Tidy up by dropping the EMPS table, and confirm that all the indexes have also gone: drop table emps; select index name from user indexes where table name='EMPS';

```
--8
DROP TABLE EMPS;
SELECT INDEX_NAME FROM USER_INDEXES WHERE TABLE_NAME='EMPS';
```

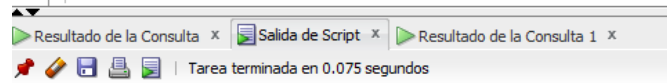
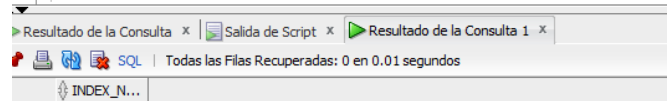


Table EMPS borrado.

```
--8
DROP TABLE EMPS;
SELECT INDEX_NAME FROM USER_INDEXES WHERE TABLE_NAME='EMPS';
```



## Activity 7:

Write the section that describes the work developed in the following activities. Consider the following context issues employed in past practices. For each one:

- Analyze the use of “sequences”, “indexes”, and “synonyms” objects for a new formulation of the solution according to the user requirements appreciated in each case.
- Write code to create the tables (only the create table statements with basic constraints: default, check, not null, . . . , and column data type specifications).
- Create indexes (choose appropriate type).
- Create constraints (unique, primary key, foreign keys, . . . ).
- Create sequences to be used for primary keys where necessary with the best options.
- Create short name synonymous for each table

User case 1. Bookstore. The establishment of activity 1/practice 1. There is an establishment that provides online book sales. Such books are always interested in knowing their author (s) and their publisher, as well as the warehouse where they are stored. Suppose the bookstore adds music cassettes, CDs and Blue-ray to its collection. The same musical element can be present in cassette or compact disc format with different prices. The user requirements are:

- The list of customers who buy the most items.
- The list of products that customers currently have in the shopping cart. •
- Know the best-selling product.
- Know what items are in stock on a specific date.
- Obtain the lists of the books with their complete data.
- Know the author of the best-selling book.
- Know the best-selling musical medium.
- Know the least sold type of musical medium.
- Know today's sales.
- Know the best-selling electronic files

Hoja de Trabajo	Generador de Consultas
<pre> /*ESTABLISHMENT TABLE*/ CREATE TABLE ESTABLISHMENT_F4( ESTABLISHMENT_ID NUMBER, NAME_ESTABLISHMENT VARCHAR2(25) CONSTRAINT NAME_ESTABLISHMENT_NN NOT NULL );  --ALTER PRIMARY KEY ALTER TABLE ESTABLISHMENT_F4 ADD CONSTRAINT EST_ID_PK PRIMARY KEY;  /*PRODUCT TABLE*/ CREATE TABLE PRODUCT_F4( PRODUCT_ID NUMBER CONSTRAINT PRODUCT_ID_UNIQUE UNIQUE, AUTHOR VARCHAR2(25), TITLE_PRODUCT VARCHAR2(25), PRICE_PRODUCT VARCHAR2(25) );  --ALTER PRIMARY KEY ALTER TABLE PRODUCT_F4 ADD CONSTRAINT PROD_ID_PK PRIMARY KEY;  /*WAREHOUSE TABLE*/ CREATE TABLE WAREHOUSE_F4( WAREHOUSE_ID NUMBER, NAME_WAREHOUSE VARCHAR2(25), ADDRESS_WAREHOUSE VARCHAR2(50) );  --ALTER PRIMARY KEY ALTER TABLE WAREHOUSE_F4 ADD CONSTRAINT WAREHOUSE_ID_PK PRIMARY KEY;  /*CUSTOMER TABLE*/ CREATE TABLE CUSTOMER_F4( CUSTOMER_ID NUMBER, NAME_CUSTOMER VARCHAR2(25), FIRST_LAST_NAME VARCHAR2(25), SECOND_LAST_NAME VARCHAR2(25), EMAIL VARCHAR2(50) );  --ALTER PRIMARY KEY ALTER TABLE CUSTOMER_F4 ADD CONSTRAINT CUST_ID_PK PRIMARY KEY;  --PROCESS /*SHOPPINGCART TABLE*/ CREATE TABLE SHOPPINGCART_F4( CUSTOMER_ID NUMBER, PRODUCT_ID NUMBER ); </pre>	

Hoja de Trabajo	Generador de Consultas
<pre> --ALTER FOREIGN KEY ALTER TABLE PURCHASE_P4 ADD CONSTRAINT CUST_PURCH_FK FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID); ALTER TABLE PURCHASE_P4 ADD CONSTRAINT PROD_PURCH_FK FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT(PRODUCT_ID); ALTER TABLE PURCHASE_P4 ADD CONSTRAINT ESTA_PURCH_FK FOREIGN KEY (ESTABLISHMENT_ID) REFERENCES ESTABLISHMENT(ESTABLISHMENT_ID);  --STORED TABLE/ CREATE TABLE STORED_P4(     PRODUCT_ID NUMBER,     WAREHOUSE_ID NUMBER ); --ALTER FOREIGN KEY ALTER TABLE STORED_P4 ADD CONSTRAINT PROD_S_FK FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT(PRODUCT_ID); ALTER TABLE STORED_P4 ADD CONSTRAINT WARE_S_FK FOREIGN KEY (WAREHOUSE_ID) REFERENCES WAREHOUSE(WAREHOUSE_ID);  --Index CREATE INDEX ESTABLISH_P4_IDX ON ESTABLISHMENT_P4(ESTABLISHMENT_ID); CREATE INDEX PURCH_P4_IDX ON PURCHASE_P4(PURCHASE_NUMBER);  --SEQUENCE CREATE SEQUENCE SEQUENCE_ESTABLISHMENT_ID     INCREMENT BY 1     START WITH 1     MAXVALUE 5     CYCLE;  CREATE SEQUENCE SEQUENCE_PRODUCT_ID     INCREMENT BY 1     START WITH 1;  CREATE SEQUENCE SEQUENCE_WAREHOUSE_ID     INCREMENT BY 1     START WITH 1     ORDER;  CREATE SEQUENCE SEQUENCE_CUSTOMER_ID     INCREMENT BY 1     START WITH 1     MAXVALUE 1000     CYCLE;  --SYNONYMS CREATE SYNONYM P4_EST FOR ESTABLISHMENT_P4; CREATE SYNONYM P4_PROD FOR PRODUCT_P4; CREATE SYNONYM P4_WH FOR WAREHOUSE_P4; CREATE SYNONYM P4_CUST FOR CUSTOMER_P4; CREATE SYNONYM P4_SC FOR SHOPPINGCART_P4; CREATE SYNONYM P4_PURCH FOR PURCHASE_P4; CREATE SYNONYM P4_STO FOR STORED_P4; </pre>	

- User Case 2. Accounts and loans. The Bank scenario of activity 8/practice
2. • Check the primary key columns.
    - Then correct the necessary attributes.
    - User requirements:
      - o List of clients order by name and account.
      - o List of loans and balances.
      - o List of employees and their boss.
      - o List of loans and their payments.
      - o List of branches and their corresponding loans and accounts

```
--ACTIVIDAD 8

CREATE TABLE CUSTOMER(
  CUSTOMER_ID NUMBER(4) CONSTRAINT CUSTOMER_ID_PK NOT NULL,
  CUSTOMER_NAME VARCHAR2(25),
  CUSTOMER_FIRST_LAST_NAME VARCHAR2(25),
  CUSTOMER_SECOND_LAST_NAME VARCHAR2(25),
  CUSTOMER_ADDRESS VARCHAR2(50),
  CUSTOMER_EMAIL VARCHAR2(50),
  CUSTOMER_PHONE_NUMBER VARCHAR2(10) CONSTRAINT C_P_N_UNIQUE UNIQUE
);

CREATE TABLE BRANCH(
  BRANCH_ID NUMBER(4),
  BRANCH_NAME VARCHAR(25),
  BRANCH_ADDRESS VARCHAR2(25),
  ASSETS VARCHAR2(25)
);

CREATE TABLE LOAN_BRANCH(
  BRANCH_ID NUMBER(4),
  LOAN_ID NUMBER(4)
);

CREATE TABLE LOAN(
  LOAN_ID NUMBER(4),
  LOAN_NUMBER VARCHAR2(25),
  AMOUNT VARCHAR(25)
);

CREATE TABLE CUST_BANKER(
  TYPE_CUST VARCHAR2(25),
  CUSTOMER_ID NUMBER(4),
  ACCOUNT_NUMBER VARCHAR2(25) CONSTRAINT ACCOUNT_N_PK NOT NULL
);

CREATE TABLE EMPLOYEE(
  EMPLOYEE_ID NUMBER(4),
  DEPENDENT_NAME VARCHAR2(25),
  PHONE_NUMBER VARCHAR2(10),
  EMPLOYEE_NAME VARCHAR2(25),
  EMPLOYEEMENT_LENGTH CHAR(9),
  -----
);

CREATE TABLE LOAN_PAYMENT(
  LOAN_ID NUMBER(4),
  PAYMENT_ID NUMBER(4)
);

CREATE TABLE PAYMENT(
  PAYMENT_ID NUMBER(4),
  PAYMENT_NUMBER CHAR(8),
  PAYMENT_DATE DATE,
  PAYMENT_ACCOUNT VARCHAR2(25)
);

CREATE TABLE ACCOUNT_BRANCH(
  BRANCH_ID NUMBER(4),
  ACCOUNT_NUMBER CHAR(8)
);

CREATE TABLE ACCOUNT(
  ACCOUNT_NUMBER CHAR(8),
  BALANCE VARCHAR2(25)
);

CREATE TABLE SAVINGS_ACCOUNT(
  INTERES_RATE CHAR(6),
  CHECKING_RATE CHAR(10),
  ACCOUNT_NUMBER CHAR(8)
);

CREATE TABLE CHECKING_ACCOUNT(
  OVERDRAFT_AMOUNT VARCHAR2(16),
  ACCOUNT_NUMBER CHAR(8)
);

ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER_ID_PK PRIMARY KEY(CUSTOMER_ID);
ALTER TABLE BRANCH ADD CONSTRAINT BRANCH_ID_PK PRIMARY KEY(BRANCH_ID);
ALTER TABLE LOAN ADD CONSTRAINT LOAN_ID_PK PRIMARY KEY(LOAN_ID);
ALTER TABLE CUST_BANKER ADD CONSTRAINT CUST_BANKER_ID_PK PRIMARY KEY(TYPE_CUST);
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_ID_PK PRIMARY KEY(EMPLOYEE_ID);
ALTER TABLE PAYMENT ADD CONSTRAINT PAYMENT_ID_PK PRIMARY KEY(PAYMENT_ID);
ALTER TABLE ACCOUNT ADD CONSTRAINT ACCOUNT_ID_PK PRIMARY KEY(ACCOUNT_NUMBER);
```

```

ALTER TABLE CUST_BANKER ADD CONSTRAINT CUSTOMER_ID_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE EMPLOYEE ADD CONSTRAINT CUST_BANKER_TYPE_ID_FK FOREIGN KEY(CUST_BANKER_TYPE) REFERENCES CUST_BANKER(TYPE_COST);
ALTER TABLE LOAN_BRANCH ADD CONSTRAINT BRANCH_ID_FK FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID);
ALTER TABLE LOAN_BRANCH ADD CONSTRAINT LOAN_ID_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE BORROWER ADD CONSTRAINT CUSTOMER_ID_BORROWER_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE BORROWER ADD CONSTRAINT LOAN_ID_BORROWER_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE LOAN_PAYMENT ADD CONSTRAINT LOAN_ID_PAYMENT_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE LOAN_PAYMENT ADD CONSTRAINT PAYMENT_ID_PAYMENT_FK FOREIGN KEY(PAYMENT_ID) REFERENCES PAYMENT(PAYMENT_ID);
ALTER TABLE DEPOSITOR ADD CONSTRAINT CUSTOMER_DEPOSITOR_ID_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE DEPOSITOR ADD CONSTRAINT ACCOUNT_DEPOSITOR_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);
ALTER TABLE SAVINGS_ACCOUNT ADD CONSTRAINT SAVINGS_ACCOUNT_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);
ALTER TABLE CHECKING_ACCOUNT ADD CONSTRAINT CHECKING_ACCOUNT_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_EMPLOYEE_ID_FK FOREIGN KEY(EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID);

ALTER TABLE ACCOUNT_BRANCH ADD CONSTRAINT BRANCH_ACCOUNT_ID_FK FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID);
ALTER TABLE ACCOUNT_BRANCH ADD CONSTRAINT ACCOUNT_NUMBER_BRANCH_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);

--INDEX --
CREATE INDEX CUST_IDX ON CUSTOMER(CUSTOMER_NAME);
CREATE INDEX CUST_AP_IDX ON CUSTOMER(CUSTOMER_FIRST_NAME, CUSTOMER_SECOND_LAST_NAME);

CREATE INDEX BRANCH_INDEX ON BRANCH(BRANCH_NAME);

CREATE INDEX EMP_IDX ON EMPLOYEE(EMPLOYEE_NAME);

--SEQUENCE
CREATE SEQUENCE SEQUENCE_COST_ID
  INCREMENT BY 1
  START WITH 1;

CREATE SEQUENCE SEQUENCE_BRANCH_ID
  INCREMENT BY 1
  START WITH 1;

CREATE SEQUENCE SEQUENCE_LOAN_ID
  INCREMENT BY 1
  START WITH 1;

CREATE SEQUENCE SEQUENCE_CUSTCUST_ID
  INCREMENT BY 1
  START WITH 1;

CREATE SEQUENCE SEQUENCE_EMPLOYEE_ID
  INCREMENT BY 1
  START WITH 1;

--SYNONYMS
CREATE SYNONYM CUST FOR CUSTOMER;
CREATE SYNONYM BR FOR BRANCH;
CREATE SYNONYM L_B FOR LOAN_BRANCH;
CREATE SYNONYM LOA FOR LOAN;
CREATE SYNONYM C_B FOR CUST_BANKER;
CREATE SYNONYM EMPL FOR EMPLOYEE;
CREATE SYNONYM DEP FOR DEPOSITOR;
CREATE SYNONYM BORR FOR BORROWER;
CREATE SYNONYM L_P FOR LOAN_PAYMENT;
CREATE SYNONYM PAYM FOR PAYMENT;
CREATE SYNONYM AC_B FOR ACCOUNT_BRANCH;
CREATE SYNONYM ACCT FOR ACCOUNT;
CREATE SYNONYM S_A FOR SAVINGS_ACCOUNT;
CREATE SYNONYM CH_A FOR CHECKING_ACCOUNT;

```

User case 4. Documents. User requirements:

- How many paragraphs has a document?
- What is the most used template?
- What type of template is the least used?
- List the name of the templates and documents.

```

-- CREATE TABLE REF_TEMPLATE_TYPES(
  TEMPLATE_TYPE_CODE CHAR(15),
  TEMPLATE_TYPE_DESCRIPTION VARCHAR2(255)
);

--ALTER PRIMARY KEY
ALTER TABLE REF_TEMPLATES_TYPES ADD CONSTRAINT TEMPLATE_TYPE_CODE PRIMARY KEY(TEMPLATE_TYPE_CODE);

-- CREATE TABLE TEMPLATES(
  TEMPLATE_ID NUMBER,
  VERSION_NUMBER NUMBER,
  TEMPLATE_TYPE_CODE CHAR(15),
  DATE_EFFECTIVE_FROM DATE,
  DATE_EFFECTIVE_TO DATE,
  TEMPLATE_DETAILS VARCHAR2(255)
);

--ALTER PRIMARY KEY
ALTER TABLE TEMPLATES ADD CONSTRAINT TEMPLATE_ID_PK PRIMARY KEY(TEMPLATE_ID);
--ALTER FOREIGN KEY
ALTER TABLE TEMPLATES ADD CONSTRAINT T_T_C_FK FOREIGN KEY (TEMPLATE_TYPE_CODE) REFERENCES REF_TEMPLATE_TYPES(TEMPLATE_TYPE_CODE);

-- CREATE TABLE DOCUMENTS(
  DOCUMENT_ID NUMBER,
  TEMPLATE_ID NUMBER,
  DOCUMENT_NAME VARCHAR2(255),
  DOCUMENT_DESCRIPTION VARCHAR2(255),
  OTHER_DETAILS VARCHAR2(255)
);

--ALTER PRIMARY KEY
ALTER TABLE DOCUMENTS ADD CONSTRAINT DOCUMENT_ID_PK PRIMARY KEY(DOCUMENT_ID);
--ALTER FOREIGN KEY
ALTER TABLE DOCUMENTS ADD CONSTRAINT DOC_TEMPL_FK FOREIGN KEY (TEMPLATE_ID) REFERENCES TEMPLATES(TEMPLATE_ID);

-- CREATE TABLE PARAGRAPHS(
  PARAGRAPH_ID NUMBER,
  DOCUMENT_ID NUMBER,
  PARAGRAPH_TEXT VARCHAR2(255),
  OTHER_DETAILS VARCHAR2(255)
);

--ALTER PRIMARY KEY
ALTER TABLE PARAGRAPHS ADD CONSTRAINT PARAGRAPH_ID_PK PRIMARY KEY(PARAGRAPH_ID);

OTHER_DETAILS VARCHAR2(255)
);

--ALTER PRIMARY KEY
ALTER TABLE PARAGRAPHS ADD CONSTRAINT PARAGRAPH_ID_PK PRIMARY KEY(PARAGRAPH_ID);

--ALTER FOREIGN KEY
ALTER TABLE PARAGRAPHS ADD CONSTRAINT D_P_FK FOREIGN KEY (DOCUMENT_ID) REFERENCES DOCUMENTS(DOCUMENT_ID);

--Index
CREATE INDEX TEMPLATES_INDX ON TEMPLATES(VERSION_NUMBER);
CREATE INDEX DOCUMENT_INDX ON DOCUMENTS(DOCUMENT_NAME);

--SEQUENCE
--CREATE SEQUENCE SEQUENCE_TEMPLATES_ID
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 5
  CYCLE;

--CREATE SEQUENCE SEQUENCE_DOCUMENTS_ID
  INCREMENT BY 1
  START WITH 1;

--CREATE SEQUENCE SEQUENCE_PARAGRAPHS_ID
  INCREMENT BY 1
  START WITH 1
  ORDER;

--SYNONYMS
CREATE SYNONYM TEMP FOR TEMPLATES;
CREATE SYNONYM REF_T_TV FOR REF_TEMPLATE_TYPE;
CREATE SYNONYM DOCS FOR DOCUMENTS;
CREATE SYNONYM PARAS FOR PARAGRAPHS;

```

User case 5. Jobs. The jobs scenario of activity 7/practice 2. User requirements:

- List of employees.
- List of jobs.
- List of branches.
- List of managers.
- Which jobs are being done.
- The jobs date are being managed

```
CREATE TABLE EMPLOYEE(  
    EMPLOYEE_ID NUMBER(4) CONSTRAINT EMPLOYEE_ID_NN NOT NULL,  
    START_DATE DATE,  
    DESCRIPTION_JOB VARCHAR2(25),  
    STATUS VARCHAR(25),  
    FIRST_NAME VARCHAR(25),  
    LAST_NAME VARCHAR(25)  
);  
  
CREATE TABLE JOB(  
    ID_JOB NUMBER(4) CONSTRAINT ID_JOB_NN NOT NULL,  
    ID_EMPLOYEE NUMBER(4) CONSTRAINT ID_EMPLOYEE_NN NOT NULL,  
    JOB_NAME VARCHAR(25),  
    JOB_DESCRIPTION VARCHAR2(25)  
);  
  
CREATE TABLE BRANCH(  
    ID_BRANCH NUMBER(4),  
    NAME_BRANCH VARCHAR2(25),  
    ID_EMPLOYEE NUMBER(4),  
    ID_JOB NUMBER(4),  
    ID_MANAGER NUMBER(4)  
);  
  
CREATE TABLE MANAGER(  
    ID_MANAGER NUMBER(4),  
    FIRST_NAME VARCHAR(25),  
    LAST_NAME VARCHAR2(25)  
);  
  
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_ID_PK PRIMARY KEY (EMPLOYEE_ID);  
ALTER TABLE JOB ADD CONSTRAINT JOB_ID_PK PRIMARY KEY (ID_JOB);  
ALTER TABLE BRANCH ADD CONSTRAINT BRANCH_ID_PK PRIMARY KEY (ID_BRANCH);  
ALTER TABLE MANAGER ADD CONSTRAINT MANAGER_ID_PK PRIMARY KEY (ID_MANAGER);  
  
ALTER TABLE JOB ADD CONSTRAINT ID_EMPLOYEE_FK FOREIGN KEY (ID_EMPLOYEE) REFERENCES EMPLOYEE (ID_EMPLOYEE);  
ALTER TABLE BRANCH ADD CONSTRAINT ID_EMPLOYEE_FK FOREIGN KEY (ID_EMPLOYEE) REFERENCES EMPLOYEE (EMPLOYEE_ID);  
ALTER TABLE BRANCH ADD CONSTRAINT ID_JOB_FK FOREIGN KEY (ID_JOB) REFERENCES JOB (ID_JOB);  
ALTER TABLE BRANCH ADD CONSTRAINT ID_MANAGER_FK FOREIGN KEY (ID_MANAGER) REFERENCES MANAGER (ID_MANAGER);  
  
--Index  
CREATE INDEX EMPLOYEE_INDEX ON EMPLOYEE (FIRST_NAME, LAST_NAME);  
CREATE INDEX JOB_INDEX ON JOB (JOB_NAME);  
CREATE INDEX MANAGER_INDEX ON MANAGER (FIRST_NAME, LAST_NAME);  
  
--SEQUENCE  
CREATE SEQUENCE SEQUENCE_EMPLOYEE_ID  
    INCREMENT BY 1  
    START WITH 1;  
  
CREATE SEQUENCE SEQUENCE_JOB_ID  
    INCREMENT BY 1  
    START WITH 1;  
  
CREATE SEQUENCE SEQUENCE_BRANCH_ID  
    INCREMENT BY 1  
    START WITH 1;  
  
CREATE SEQUENCE SEQUENCE_MANAGER_ID  
    INCREMENT BY 1  
    START WITH 1;  
  
--SYNONYMS  
CREATE SYNONYM EMPL FOR EMPLOYEE;  
CREATE SYNONYM JB FOR JOB;  
CREATE SYNONYM BR FOR BRANCH;  
CREATE SYNONYM MGER FOR MANAGER;
```



### 3 PRE-EVALUATION

Practices pre-Assessment for Database Systems Laboratory II Pre-Assessment  
PRACTICE 4 carried out by student

1 COMPLIES WITH THE REQUESTED FUNCTIONALITY  
YES

4 HAS THE CORRECT INDENTATION  
YES

6 HAS AN EASY WAY TO ACCESS THE PROVIDED FILES  
YES

7 HAS A REPORT WITH IDC FORMAT  
YES

8 REPORT INFORMATION IS FREE OF SPELLING ERRORS  
YES

9 DELIVERED IN TIME AND FORM  
YES

10 IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)  
YES,90-95 percent

## 4 Conclusion

The new topics / concepts that we saw are important like any other, they can help us a lot in the creation of databases, on the one hand the indexes can help us to find a record when a database is very large and with many records, on the other hand we have the sequences that in a real situation can help us to filter and put some parameters to search and insert records, and finally the synonyms, these can help us create our shortcuts or shorter words with which we can manipulate our tables.