

Universidad Nacional Autónoma de México

Facultad de ingeniería

Ingeniería en Computación

Laboratorio de Bases de Datos

Práctica 7: Lenguaje de definición de datos

15 de marzo de 2021

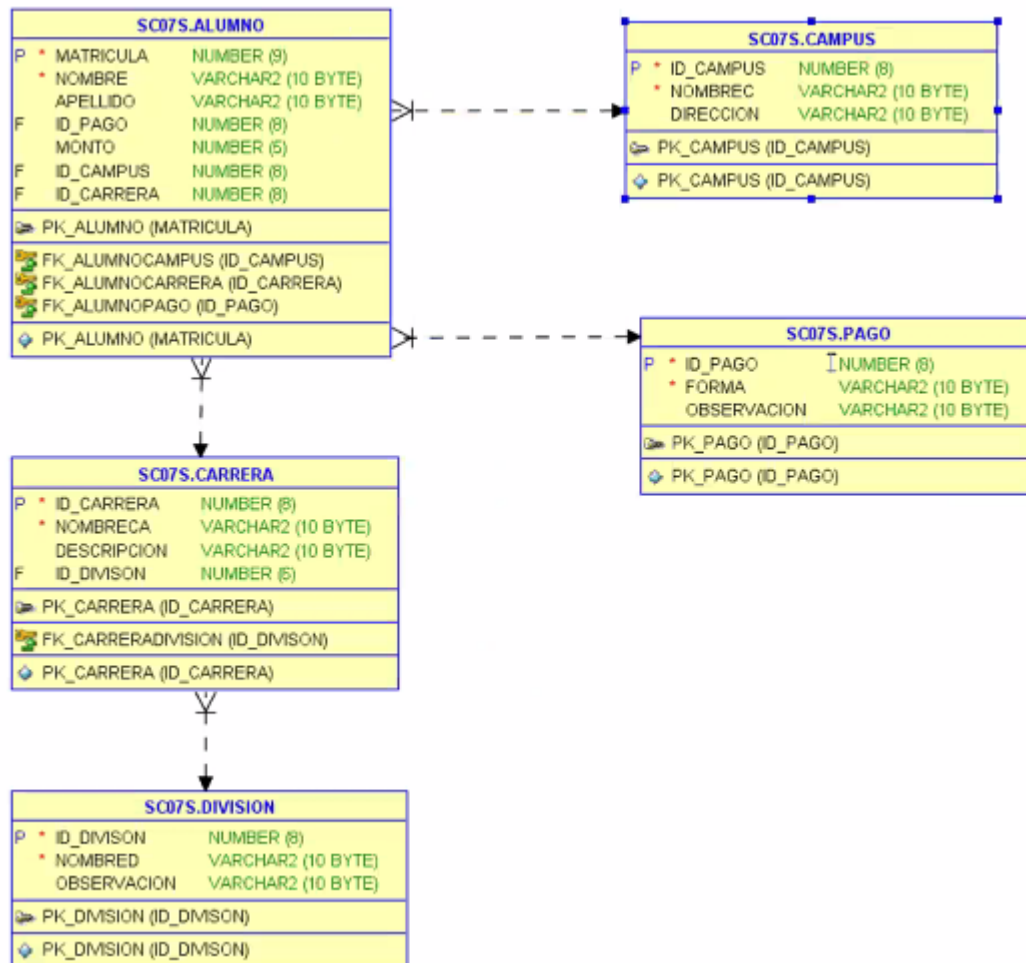
22 de marzo de 2021

Alumno: Correa González Alfredo

A handwritten signature in black ink, appearing to be 'AG' with a stylized flourish above the 'G'.

2. Desarrollo:

1.-Crea las siguientes tablas:



conectándose a la BD.

```
Símbolo del sistema - SQLPLUS

C:\Users\pokem>SQLPLUS

SQL*Plus: Release 11.2.0.2.0 Production on Mar Jun 22 12:07:40 2021

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: SYSTEM
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

Creando las tablas:

```
SQL> CREATE TABLE ALUMNO(  
  2  MATRICULA VARCHAR(9),  
  3  NOMBRE VARCHAR2(10 BYTE),  
  4  APELLIDO VARCHAR2(10 BYTE),  
  5  ID_PAGO NUMBER(8),  
  6  MONTO NUMBER(5),  
  7  ID_CAMPUS NUMBER(8),  
  8  ID_CARRERA NUMBER(8)  
  9 );  
  
Table created.  
  
SQL> CREATE TABLE CAMPUS(  
  2  ID_CAMPUS NUMBER(8),  
  3  NOMBREC VARCHAR2(10 BYTE),  
  4  DIRECCION VARCHAR2(10 BYTE)  
  5 );  
  
Table created.  
  
SQL> CREATE TABLE PAGO(  
  2  ID_PAGO NUMBER(8),  
  3  FORMA VARCHAR2(10 BYTE),  
  4  OBSERVACION VARCHAR(10 BYTE)  
  5 );  
  
Table created.  
  
SQL> CREATE TABLE CARRERA(  
  2  ID_CARRERA NUMBER(8),  
  3  NOMBRECA VARCHAR2(10 BYTE),  
  4  DESCRIPCION VARCHAR2(10 BYTE),  
  5  ID_DIVISION NUMBER(5)  
  6 );  
  
Table created.  
  
SQL> CREATE TABLE DIVISION(  
  2  ID_DIVISION NUMBER(8),  
  3  NOMBRED VARCHAR2(10 BYTE),  
  4  OBSERVACION VARCHAR2(10 BYTE)  
  5 );  
  
Table created.
```

2.- Crear todos los CONSTRAINT.

Hay que recordar que se tienen que crear primero las tablas que solo tengan llaves primarias.

```
SQL> DROP TABLE DIVISION;  
  
Table dropped.  
  
SQL> CREATE TABLE DIVISION(  
  2  ID_DIVISION NUMBER(8),  
  3  NOMBRED VARCHAR2(10 BYTE),  
  4  OBSERVACION VARCHAR2(10 BYTE),  
  5  CONSTRAINT PK_DIVISION PRIMARY KEY (ID_DIVISION)  
  6 );  
  
Table created.
```

```
SQL> CREATE TABLE CARRERA(  
  2  ID_CARRERA NUMBER(8),  
  3  NOMBRECA VARCHAR2(10 BYTE),  
  4  DESCRIPCION VARCHAR2(10 BYTE),  
  5  ID_DIVISION NUMBER(5),  
  6  CONSTRAINT PK_CARRERA PRIMARY KEY (ID_CARRERA),  
  7  CONSTRAINT FK_CARRERDIVISION FOREIGN KEY (ID_DIVISION)  
  8  REFERENCES DIVISION(ID_DIVISION)  
  9  );
```

Table created.

```
SQL> DROP TABLE PAGO;
```

Table dropped.

```
SQL>  
SQL> CREATE TABLE PAGO(  
  2  ID_PAGO NUMBER(8),  
  3  FORMA VARCHAR2(10 BYTE),  
  4  OBSERVACION VARCHAR(10 BYTE),  
  5  CONSTRAINT PK_PAGO PRIMARY KEY(ID_PAGO)  
  6  );
```

Table created.

Table dropped.

```
SQL>  
SQL> CREATE TABLE CAMPUS(  
  2  ID_CAMPUS NUMBER(8),  
  3  NOMBREC VARCHAR2(10 BYTE),  
  4  DIRECCION VARCHAR2(10 BYTE),  
  5  CONSTRAINT PK_CAMPUS PRIMARY KEY (ID_CAMPUS)  
  6  );
```

Table created.

```
SQL> CREATE TABLE ALUMNO(
  2  MATRICULA VARCHAR(9),
  3  NOMBRE VARCHAR2(10 BYTE),
  4  APELLIDO VARCHAR2(10 BYTE),
  5  ID_PAGO NUMBER(8),
  6  MONTO NUMBER(5),
  7  ID_CAMPUS NUMBER(8),
  8  ID_CARRERA NUMBER(8),
  9  CONSTRAINT PK_ALUMNO PRIMARY KEY (MATRICULA),
10  CONSTRAINT FK_ALUMNOCAMPUS FOREIGN KEY (ID_CAMPUS)
11  REFERENCES CAMPUS(ID_CAMPUS),
12  CONSTRAINT FK_ALUMNOCARRERA FOREIGN KEY (ID_CARRERA)
13  REFERENCES CARRERA(ID_CARRERA),
14  CONSTRAINT FK_ALUMNOPAGO FOREIGN KEY (ID_PAGO)
15  REFERENCES PAGO(ID_PAGO)
16 );
```

Table created.

3.- Crear una secuencia para la matrícula de alumnos.

```
SQL> CREATE SEQUENCE SQ_ALUMNO
  2  START WITH 1
  3  INCREMENT BY 1;
```

Sequence created.

4.- Insertar datos: 5 campus, 3 pagos y 2 divisiones.

```
SQL> INSERT INTO CAMPUS VALUES(1,'INGENIERIA', 'CU');
1 row created.

SQL> INSERT INTO CAMPUS VALUES(2,'MEDICINA','COPILCO');
1 row created.
```

```
SQL> INSERT INTO CAMPUS VALUES(3,'CONTA','METROCU');
1 row created.
```

```
SQL> SELECT *FROM CAMPUS;
```

ID_CAMPUS	NOMBREC	DIRECCION
1	INGENIERIA	CU
2	MEDICINA	COPILCO
3	CONTA	METROCU

```
SQL> INSERT INTO PAGO VALUES(1,'CHEQUE','NINGUNA');
1 row created.

SQL> INSERT INTO PAGO VALUES(2,'DEPOSITO','10PESOS');
1 row created.

SQL> INSERT INTO PAGO VALUES(3,'EFFECTIVO','NINGUNA');
1 row created.
```

```
SQL> SELECT *FROM PAGO;
```

ID_PAGO	FORMA	OBSERVACIO
1	CHEQUE	NINGUNA
2	DEPOSITO	10PESOS
3	EFFECTIVO	NINGUNA

```
SQL> INSERT INTO DIVISION VALUES(2,'CTIERRA','NINGUNA');
1 row created.

SQL> SELECT *FROM DIVISION;
```

ID_DIVISION	NOMBRED	OBSERVACIO
2	CTIERRA	NINGUNA

```
SQL> INSERT INTO DIVISION VALUES(1,'DIE','COMPUS');
1 row created.

SQL> SELECT *FROM DIVISION;
```

ID_DIVISION	NOMBRED	OBSERVACIO
2	CTIERRA	NINGUNA
1	DIE	COMPUS

5.- Insertar 5 carreras y 50 alumnos

```
SQL> INSERT INTO CARRERA VALUES(1,'COMPU','PROGRAMAR',1);
```

```
1 row created.
```

```
SQL> INSERT INTO CARRERA VALUES(2,'ELECT','CABLES',1);
```

```
1 row created.
```

```
SQL> INSERT INTO CARRERA VALUES(3,'TRONICA','PROTOBOARD',1);
```

```
1 row created.
```

```
SQL> INSERT INTO CARRERA VALUES(4,'PETRO','PETROLEO',2);
```

```
1 row created.
```

```
SQL> INSERT INTO CARRERA VALUES(5,'GEOLO','ROCAS',2);
```

```
1 row created.
```

```
SQL> SELECT * FROM CARRERA  
2 ;
```

ID_CARRERA	NOMBRECA	DESCRIPCIO	ID_DIVISION
1	COMPU	PROGRAMAR	1
2	ELECT	CABLES	1
3	TRONICA	PROTOBOARD	1
4	PETRO	PETROLEO	2
5	GEOLO	ROCAS	2

```
SQL> INSERT INTO ALUMNO VALUES(SQ_ALUMNO2.NEXTVAL, 'ALFREDO', 'CORREA', 1, 500, 1, 1);
1 row created.
```

```
SQL> INSERT INTO ALUMNO VALUES(SQ_ALUMNO2.NEXTVAL, 'ALFREDO', 'CORREA', 1, 500, 1, 1);
1 row created.
```

```
SQL> INSERT INTO ALUMNO VALUES(SQ_ALUMNO2.NEXTVAL, 'ALFREDO', 'CORREA', 1, 500, 1, 1);
1 row created.
```

```
SQL> INSERT INTO ALUMNO VALUES(SQ_ALUMNO2.NEXTVAL, 'ALFREDO', 'CORREA', 1, 500, 1, 1);
1 row created.
```

```
SQL> INSERT INTO ALUMNO VALUES(SQ_ALUMNO2.NEXTVAL, 'ALFREDO', 'CORREA', 1, 500, 1, 1);
1 row created.
```

```
SQL> SELECT * FROM ALUMNO;
```

MATRICULA	NOMBRE	APELLIDO	ID_PAGO	MONTO	ID_CAMPUS	ID_CARRERA
1	ALFREDO	CORREA	1	500	1	1
2	ALFREDO	CORREA	1	500	1	1
3	ALFREDO	CORREA	1	500	1	1
4	ALFREDO	CORREA	1	500	1	1
5	ALFREDO	CORREA	1	500	1	1
6	ALFREDO	CORREA	1	500	1	1
7	ALFREDO	CORREA	1	500	1	1
8	ALFREDO	CORREA	1	500	1	1
9	ALFREDO	CORREA	1	500	1	1
10	ALFREDO	CORREA	1	500	1	1
11	ALFREDO	CORREA	1	500	1	1

MATRICULA	NOMBRE	APELLIDO	ID_PAGO	MONTO	ID_CAMPUS	ID_CARRERA
12	ALFREDO	CORREA	1	500	1	1
13	ALFREDO	CORREA	1	500	1	1
14	ALFREDO	CORREA	1	500	1	1
15	ALFREDO	CORREA	1	500	1	1
16	ALFREDO	CORREA	1	500	1	1
17	ALFREDO	CORREA	1	500	1	1
18	ALFREDO	CORREA	1	500	1	1
19	ALFREDO	CORREA	1	500	1	1
20	ALFREDO	CORREA	1	500	1	1
21	ALFREDO	CORREA	1	500	1	1
22	ALFREDO	CORREA	1	500	1	1
MATRICULA	NOMBRE	APELLIDO	ID_PAGO	MONTO	ID_CAMPUS	ID_CARRERA
23	ALFREDO	CORREA	1	500	1	1
24	ALFREDO	CORREA	1	500	1	1
25	ALFREDO	CORREA	1	500	1	1
26	ALFREDO	CORREA	1	500	1	1
27	ALFREDO	CORREA	1	500	1	1
28	ALFREDO	CORREA	1	500	1	1
29	ALFREDO	CORREA	1	500	1	1
30	ALFREDO	CORREA	1	500	1	1
31	ALFREDO	CORREA	1	500	1	1
32	ALFREDO	CORREA	1	500	1	1
33	ALFREDO	CORREA	1	500	1	1
MATRICULA	NOMBRE	APELLIDO	ID_PAGO	MONTO	ID_CAMPUS	ID_CARRERA
34	ALFREDO	CORREA	1	500	1	1
35	ALFREDO	CORREA	1	500	1	1
36	ALFREDO	CORREA	1	500	1	1
37	ALFREDO	CORREA	1	500	1	1
38	ALFREDO	CORREA	1	500	1	1
39	ALFREDO	CORREA	1	500	1	1
40	ALFREDO	CORREA	1	500	1	1
41	ALFREDO	CORREA	1	500	1	1
42	ALFREDO	CORREA	1	500	1	1
43	ALFREDO	CORREA	1	500	1	1
44	ALFREDO	CORREA	1	500	1	1
MATRICULA	NOMBRE	APELLIDO	ID_PAGO	MONTO	ID_CAMPUS	ID_CARRERA
45	ALFREDO	CORREA	1	500	1	1
46	ALFREDO	CORREA	1	500	1	1
47	ALFREDO	CORREA	1	500	1	1
48	ALFREDO	CORREA	1	500	1	1
49	ALFREDO	CORREA	1	500	1	1
50	ALFREDO	CORREA	1	500	1	1

50 rows selected.

6.- Eliminar los registros de la tabla de pago si se puede.

```
SQL> DELETE FROM PAGO
      2 WHERE ID_PAGO = 1;
DELETE FROM PAGO
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.FK_ALUMNOPAGO) violated - child record found
```

No deja eliminar registros.

7.- Eliminar la tabla de pago.

```
SQL> DROP TABLE PAGO;
DROP TABLE PAGO
*
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys
```

9.- Mostrar las estructuras de las tablas.

```
SQL> DESCRIBE ALUMNO;
Name                               Null?    Type
-----
MATRICULA                          NOT NULL VARCHAR2(9)
NOMBRE                             VARCHAR2(10)
APELLIDO                           VARCHAR2(10)
ID_PAGO                            NUMBER(8)
MONTO                              NUMBER(5)
ID_CAMPUS                           NUMBER(8)
ID_CARRERA                          NUMBER(8)
```

```
SQL> DESCRIBE CARRERA;
Name                               Null?    Type
-----
ID_CARRERA                          NOT NULL NUMBER(8)
NOMBRECA                           VARCHAR2(10)
DESCRIPCION                         VARCHAR2(10)
ID_DIVISION                         NUMBER(5)
```

```
SQL> DESCRIBE DIVISION;
Name                               Null?    Type
-----
ID_DIVISION                         NOT NULL NUMBER(8)
NOMBRED                             VARCHAR2(10)
OBSERVACION                         VARCHAR2(10)
```

```
SQL> DESCRIBE CAMPUS;
```

Name	Null?	Type
ID_CAMPUS	NOT NULL	NUMBER(8)
NOMBREC		VARCHAR2(10)
DIRECCION		VARCHAR2(10)

```
SQL> DESCRIBE PAGO;
```

Name	Null?	Type
ID_PAGO	NOT NULL	NUMBER(8)
FORMA		VARCHAR2(10)
OBSERVACION		VARCHAR2(10)

10.- Explicar que es un índice en Oracle.

Los índices sirven para acceder a los registros de una tabla rápidamente, acelerando la localización de la información.

Oracle accede a los datos de dos maneras:

- 1) recorriendo las tablas; comenzando el principio y extrayendo los registros que cumplen las condiciones de la consulta; lo cual implica posicionar las cabezas lectoras, leer el dato, controlar si coincide con lo que se busca (como si pasáramos una a una las páginas de un libro buscando un tema específico).
- 2) empleando índices; recorriendo la estructura de árbol del índice para localizar los registros y extrayendo los que cumplen las condiciones de la consulta (comparando con un libro, diremos que es como leer el índice y luego de encontrar el tema buscado, ir directamente a la página indicada).

Un índice posibilita el acceso directo y rápido haciendo más eficiente las búsquedas. Sin índice, Oracle debe recorrer secuencialmente toda la tabla para encontrar un registro.

3. Conclusiones:

En esta práctica pude retomar el como crear tablas e insertarle valores como vimos en las prácticas anteriores, aun que todavía tengo dudas respecto a cómo manipulara una tabla ya hecha ya que por eso no pude completar exitosamente el punto 6 y 7 ya que tenía que eliminar registros y la llave foránea de la tabla pago pero no lo logré.

Respecto a los demás puntos, al principio tuve problemas porque no sabía cómo referenciar las llaves foráneas, pero al final lo logré. También me quedó en claro el como crear CONSTARINT y como crear secuencias.