



INSTITUTO DE INVESTIGACIÓN FIIS-UNI
Facultad de Ingeniería Industrial y de Sistemas

TALLER DE PROGRAMACION CON ORACLE PL/SQL

Capítulo 03 Crear un Esquema

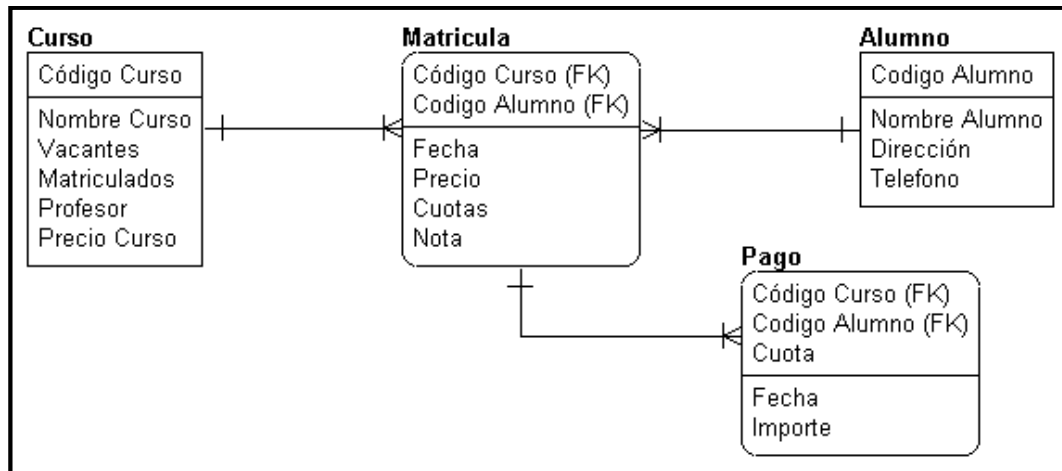
Contenido

- Caso a Desarrollar
- Creación del Usuario para el Esquema
- Creación de Tablas
- Restricción Primary Key (PK)
- Restricción Foreign Key (FK)
- Restricción Default (Valores por Defecto)
- Restricción NOT NULL (Nulidad de una Columna)
- Restricción Unique (Valores Únicos)
- Restricción Check (Reglas de Validación)
- Asignar Privilegios a Usuarios

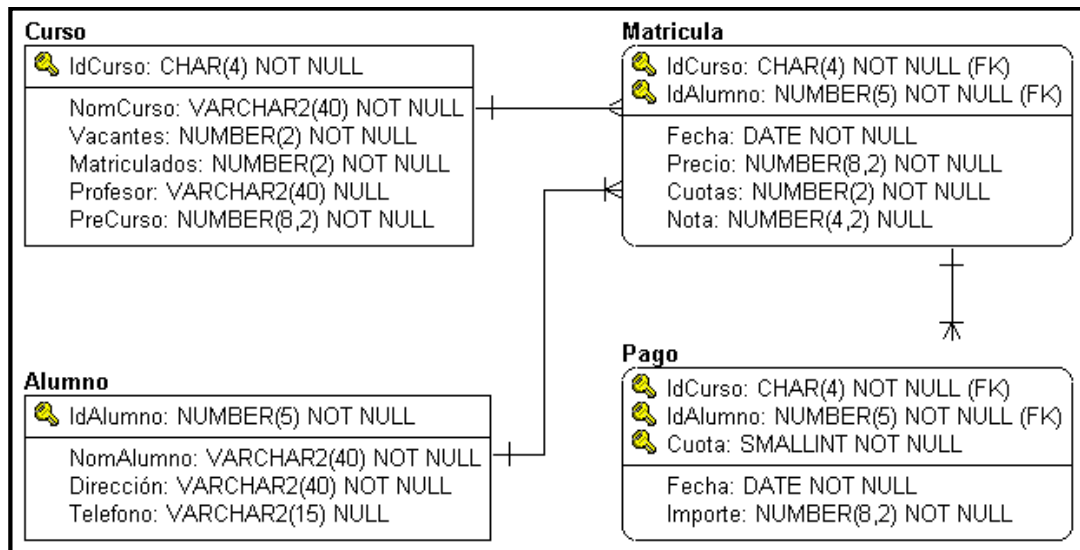
Caso a Desarrollar

El siguiente modelo trata de una empresa que ofrece cursos de extensión, los participantes tienen la libertad de matricularse sin ninguna restricción, y pueden tener facilidades de pago.

Modelo Lógico



Modelo Físico



Creación del Usuario para el Esquema

Creación del Usuario

Script 9.1

```
SQL> conn / as sysdba
Connected.

SQL> create user egcc
      2 identified by admin;

User created.
```

Asignar Privilegios

Asignaremos privilegios al usuario **egcc** a través de los roles **connect** y **resource**, los cuales le otorgan los privilegios necesarios para que pueda crear sus objetos.

Script 9.2

```
SQL> grant connect, resource to egcc;

Grant succeeded.
```

Ahora ya podemos ingresar como usuario **egcc** y crear los objetos que corresponden a su esquema.

Creación de Tablas

Sintaxis

```
Create Table NombreTabla(  
  Columna1 Tipo1 [ NULL | NOT NULL ],  
  Columna2 Tipo2 [ NULL | NOT NULL ],  
  Columna2 Tipo2 [ NULL | NOT NULL ],  
  . . .  
  . . .  
);
```

Tabla Curso

Script 9.3

```
SQL> connect egcc/admin  
Connected.  
  
SQL> CREATE TABLE Curso (  
2      IdCurso          CHAR(4) NOT NULL,  
3      NomCurso         VARCHAR2(40) NOT NULL,  
4      Vacantes         NUMBER(2) NOT NULL,  
5      Matriculados     NUMBER(2) NOT NULL,  
6      Profesor         VARCHAR2(40) NULL,  
7      PreCurso         NUMBER(8,2) NOT NULL  
8  );  
  
Table created.
```

Tabla Alumno

Escriba el script para crear la tabla Alumno.

Tabla Matricula

Escriba el script para crear la tabla Matricula.

Tabla Pago

Escriba el script para crear la tabla Pago.

Restricción Primary Key (PK)

La restricción Primary Key se utiliza para definir la clave primaria de una tabla, en el siguiente cuadro se especifica la(s) columna(s) que conforman la PK de cada tabla.

Tabla	Primary Key
Curso	Incurso
Alumno	IdAlumno
Matricula	IdCurso, IdAlumno
Pago	IdCurso, IdAlumno, Cuota

Sintaxis

```
Alter Table NombreTabla  
Add Constraint PK_NombreTabla  
Primary Key ( Columna1, Columna2, . . . );
```

Tabla Curso

Script 9.4

```
SQL> Alter Table Curso  
2      Add Constraint PK_Curso  
3      Primary Key ( IdCurso );  
  
Table altered.
```

Tabla Alumno

Escriba el script para crear la PK de la tabla Alumno.

Tabla Matricula

Escriba el script para crear la PK de la tabla Matricula.

Tabla Pago

Escriba el script para crear la PK de la tabla Pago.

Restricción Foreign Key (FK)

La restricción Foreign Key se utiliza para definir la relación entre dos tablas, en el siguiente cuadro se especifica la(s) columna(s) que conforman la FK de cada tabla.

Tabla	Foreign Key	Tabla Referenciada
Matricula	IdCurso	Curso
	IdAlumno	Alumno
Pago	IdCurso, IdAlumno	Matricula

Sintaxis

```
Alter Table NombreTabla
Add Constraint FK_NombreTabla_TablaReferenciada
Foreign Key ( Columna1, Columna2, . . . )
References TablaReferenciada;
```

Es necesario que en la tabla referenciada esté definida la PK, por que la relación se crea entre la PK de la tabla referenciada y las columnas que indicamos en la cláusula Foreign Key.

Tabla Matricula

1ra FK

La primera FK de esta tabla es IdCurso y la tabla referenciada es Curso, el script para crear esta FK es el siguiente:

Script 9.5

```
SQL> Alter table Matricula
2   Add Constraint FK_Matricula_Curso
3   Foreign Key ( IdCurso )
4   References Curso;

Table altered.
```


2da FK

La segunda FK de esta tabla es **IdAlumno** y la tabla referenciada es **Alumno**, escriba usted el script para crear ésta FK.

Tabla Pago

Esta tabla solo tiene una FK y está compuesta por dos columnas: **IdCurso** e **IdAlumno**, y la tabla referenciada es **Matricula**, escriba usted el script para crear ésta FK.

Restricción Default (Valores por Defecto)

El *Valor por Defecto* es el que toma una columna cuando no especificamos su valor en una sentencia insert.

Sintaxis

```
Alter Table NombreTabla  
Modify ( NombreColumna Default Expresión );
```

Ejemplo

El número de vacantes por defecto para cualquier curso debe ser 20.

Script 9. 6

```
SQL> Alter Table Curso  
2 Modify ( Vacantes default 20 );  
  
Table altered.
```

Para probar el default insertemos un registro en la tabla curso.

Script 9. 7

IDCU	NOMCURSO	VACANTES	MATRICULADOS	PROFESOR	PRECURSO
C001	Oracle 9i - Nivel Inicial	20	10	Gustavo Coronel	350

Restricción NOT NULL (Nulidad de una Columna)

Es muy importante determinar la nulidad de una columna, y es muy importante para el desarrollador tener esta información a la mano cuando crea las aplicaciones.

Sintaxis

```
Alter Table NombreTabla  
  Modify ( NombreColumna [NOT] NULL );
```

Ejemplo

En la tabla alumno, la columna **Telefono** no debe aceptar valores nulos.

Script 9. 8

```
SQL> Alter Table Alumno  
  2  Modify ( Telefono NOT NULL );  
  
Table altered.  
  
SQL> describe alumno  
Name                               Null?    Type  
-----  
IDALUMNO                           NOT NULL NUMBER(5)  
NOMALUMNO                           NOT NULL VARCHAR2(40)  
DIRECCIÓN                           NOT NULL VARCHAR2(40)  
TELEFONO                           NOT NULL VARCHAR2(15)
```

Si queremos insertar un alumno tendríamos que ingresar datos para todas las columnas.

Script 9. 9

```
SQL> insert into alumno  
  2  values(10001, 'Ricardo Marcelo', 'Ingeniería', NULL);  
insert into alumno  
*  
ERROR at line 1:  
ORA-01400: cannot insert NULL into ("EGCC"."ALUMNO"."TELEFONO")
```

El mensaje de error claramente nos indica que no se puede insertar valores nulos en la columna TELEFONO, de la tabla ALUMNO, que se encuentra en el esquema EGCC.

Restricción Unique (Valores Únicos)

En muchos casos debemos garantizar que los valores de una columna ó conjunto de columnas de una tabla acepten solo valores únicos.

Sintaxis

```
Alter Constraint NombreTabla  
Add Constraint U_NombreTabla_NombreColumna  
Unique ( Columna1, Columna2, . . . );
```

Ejemplo

No puede haber dos alumnos con nombres iguales.

Script 9.10

```
SQL> Alter Table alumno  
2 Add Constraint U_Alumno_NomAlumno  
3 Unique (NomAlumno);  
  
Table altered.
```

Para probar la restricción insertemos datos.

Script 9.11

```
SQL> Insert Into Alumno  
2 Values( 10001, 'Sergio Matsukawa', 'San Miguel', '456-3456' );  
  
1 row created.  
  
SQL> Insert Into Alumno  
2 Values( 10002, 'Sergio Matsukawa', 'Los Olivos', '521-3456' );  
Insert Into Alumno  
*  
ERROR at line 1:  
ORA-00001: unique constraint (EGCC.U_ALUMNO_NOMALUMNO) violated
```

El mensaje de error del segundo insert nos indica que esta violando el constraint de tipo unique de nombre U_ALUMNO_NOMALUMNO en el esquema EGCC.

Restricción Check (Reglas de Validación)

Las reglas de validación son muy importantes por que permiten establecer una condición a los valores que debe aceptar una columna.

Sintaxis

```
Alter Table NombreTabla  
Add Constraint CK_NombreTable_NombreColumna  
Check ( Condición );
```

Ejemplo

El precio de un curso no puede ser cero, ni menor que cero.

Script 9.12

```
SQL> Alter Table Curso  
2 Add Constraint CK_Curso_PreCurso  
3 Check ( PreCurso > 0 );  
  
Table altered.
```

Probemos el constraint ingresando datos.

Script 9.13

```
SQL> Insert Into Curso  
2 Values( 'C002', 'Asp.NET', 20, 7, 'Ricardo Marcelo', -400.00 );  
Insert Into Curso  
*  
ERROR at line 1:  
ORA-02290: check constraint (EGCC.CK_CURSO_PRECURSO) violated
```

Al intentar ingresar un curso con precio negativo, inmediatamente nos muestra el mensaje de error indicándonos que se está violando la regla de validación.

Asignar Privilegios a Usuarios

Si queremos que otros usuarios puedan operar los objetos de un esquema, debemos darle los privilegios adecuadamente.

Sintaxis

```
Grant Privilegio On Objeto To Usuario;
```

Ejemplo

Por ejemplo, el usuario scott necesita consultar la tabla curso.

Script 9.14

```
SQL> Grant Select On Curso To Scott;

Grant succeeded.
```

Ahora hagamos la prueba respectiva.

Script 9.15

```
SQL> connect scott/tiger
Connected.

SQL> select * from egcc.curso;
```

IDCU	NOMCURSO	VACANTES	MATRICULADOS	PROFESOR	PRECURSO
C001	Oracle 9i - Nivel Inicial	20	10	Gustavo Coronel	350