

# SQL SERVER - IMPLEMENTACIÓN

## LECCIÓN 03

### CREACIÓN DE BASE DE DATOS CASO PRACTICO



**Eric Gustavo Coronel Castillo**

[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)

[gcoronel@uni.edu.pe](mailto:gcoronel@uni.edu.pe)

## INDICE

|   |    |
|---|----|
| REQUERIMIENTO .....                         | 4  |
| Descripción .....                           | 4  |
| Modelo de Datos.....                        | 4  |
| CREACIÓN DE LA BASE DE DATOS .....          | 5  |
| Sintaxis .....                              | 5  |
| Creación de la base de datos EDUCA.....     | 5  |
| CREACIÓN DE UNA TABLA .....                 | 5  |
| Sintaxis .....                              | 5  |
| Tabla: ALUMNO .....                         | 5  |
| Tabla: CURSO.....                           | 6  |
| Tabla: MATRICULA.....                       | 6  |
| Tabla: PAGO .....                           | 6  |
| CLAVES PRIMARIAS .....                      | 7  |
| Sintaxis .....                              | 7  |
| Creación de Claves Primarias.....           | 7  |
| CLAVES FORANEAS .....                       | 8  |
| Sintaxis .....                              | 8  |
| Creación de Claves Foráneas.....            | 8  |
| RESTRICCIONES TIPO UNIQUE .....             | 9  |
| Sintaxis .....                              | 9  |
| Creación de Restricciones tipo UNIQUE ..... | 9  |
| RESTRICCIONES TIPO CHECK .....              | 10 |
| Sintaxis .....                              | 10 |
| Creación de Restricciones tipo CHECK.....   | 10 |

|  |    |
|--|----|
| Creación de Restricciones tipo DEFAULT ..... | 11 |
| Sintaxis .....                               | 11 |
| Ejemplo .....                                | 11 |
| INSERTAR DATOS.....                          | 11 |
| Tabla Curso .....                            | 11 |
| Tabla Alumno.....                            | 12 |
| Tabla Matricula .....                        | 12 |
| Actualizar Tabla Curso .....                 | 13 |
| Tabla Pago.....                              | 13 |
| SECUENCIAS.....                              | 14 |
| Sintaxis .....                               | 14 |
| Ejemplo.....                                 | 14 |

### REQUERIMIENTO

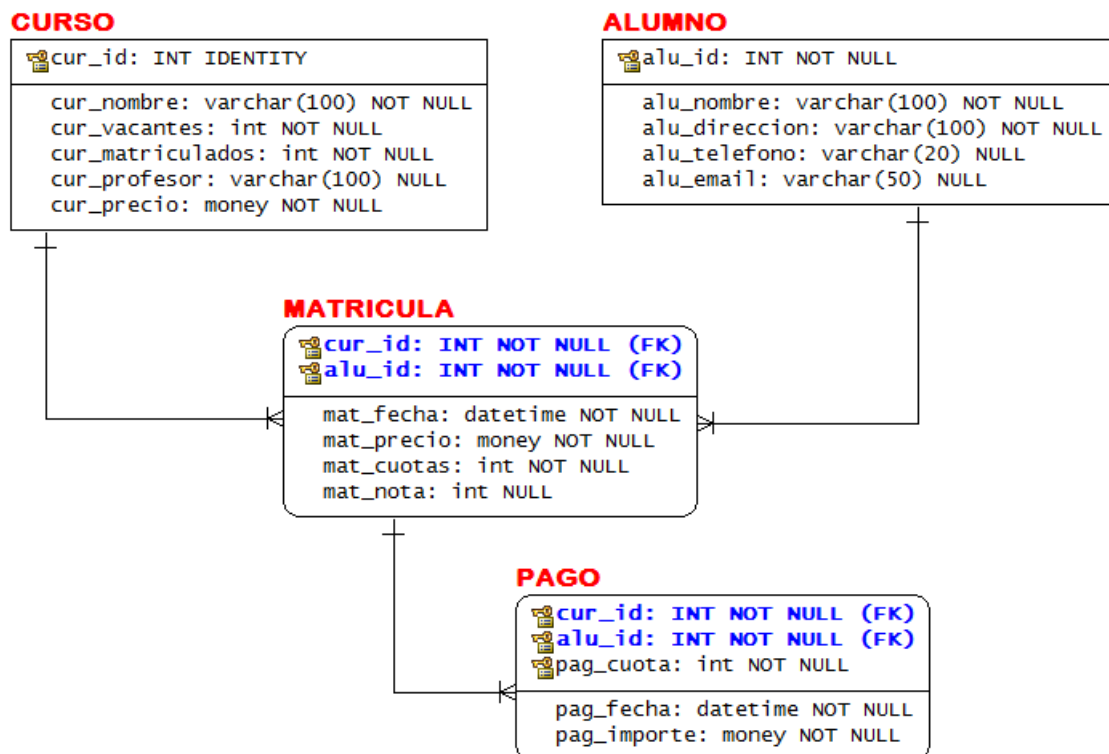
#### Descripción

La institución **Perú Training** necesita un modelo de datos para desarrollar un prototipo para llevar el control de:

1. Los alumnos que se matriculan en cada uno de sus cursos.
2. Los pagos que realizan los alumnos.
3. La nota que el alumno obtiene en el curso.

#### Modelo de Datos

El modelo de datos es básico, y se muestra en la siguiente imagen:



## CREACIÓN DE LA BASE DE DATOS

### Sintaxis

La sintaxis básica para crear una base de datos es la siguiente:

```
CREATE DATABASE < nombre de la base de datos >  
GO
```

### Creación de la base de datos EDUCA

```
CREATE DATABASE EDUCA;  
GO
```

```
USE EDUCA  
GO
```

## CREACIÓN DE UNA TABLA

### Sintaxis

```
CREATE TABLE [ esquema . ] < nombre de tabla > (  
    < definición de columna >,  
    . . .  
    . . .  
);  
GO
```

### Tabla: ALUMNO

```
CREATE TABLE dbo.ALUMNO  
(  
    alu_id                INT NOT NULL ,  
    alu_nombre            varchar(100) NOT NULL ,  
    alu_direccion          varchar(100) NOT NULL ,  
    alu_telefono           varchar(20)  NULL  ,  
    alu_email              varchar(50)  NULL  
);  
GO
```

**Tabla: CURSO**

```
CREATE TABLE dbo.CURSO
(
    cur_id                INT IDENTITY ( 1,1 ) NOT NULL ,
    cur_nombre            varchar(100) NOT NULL ,
    cur_vacantes          int NOT NULL ,
    cur_matriculados      int NOT NULL ,
    cur_profesor          varchar(100) NULL ,
    cur_precio            money NOT NULL
);
GO
```

**Tabla: MATRICULA**

```
CREATE TABLE dbo.MATRICULA
(
    cur_id                INT NOT NULL ,
    alu_id                INT NOT NULL ,
    mat_fecha             datetime NOT NULL ,
    mat_precio            money NOT NULL ,
    mat_cuotas            int NOT NULL ,
    mat_nota              int NULL
);
GO
```

**Tabla: PAGO**

```
CREATE TABLE dbo.PAGO
(
    cur_id                INT NOT NULL ,
    alu_id                INT NOT NULL ,
    pag_cuota             int NOT NULL ,
    pag_fecha            datetime NOT NULL ,
    pag_importe           money NOT NULL
);
GO
```

## CLAVES PRIMARIAS

### Sintaxis

```
ALTER TABLE [ esquema . ] < nombre de tabla >  
  ADD CONSTRAINT < nombre >  
    PRIMARY KEY CLUSTERED ( < lista de columnas > );  
GO
```

### Creación de Claves Primarias

Crear las claves primarias según el siguiente cuadro:

| TABLA     | NOMBRE       | COLUMNAS                  |
|-----------|--------------|---------------------------|
| CURSO     | PK_CURSO     | cur_id                    |
| ALUMNO    | PK_ALUMNO    | alu_id                    |
| MATRICULA | PK_MATRICULA | cur_id, alu_id            |
| PAGO      | PK_PAGO      | cur_id, alu_id, pag_cuota |

## CLAVES FORANEAS

### Sintaxis

```
ALTER TABLE [ esquema . ] < nombre de tabla >  
  ADD CONSTRAINT < nombre >  
  FOREIGN KEY ( < lista de columnas > )  
  REFERENCES [ esquema . ] < nombre de tabla referenciada > [ ( columnas  
referenciadas ) ]  
  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  
  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ] ;  
GO
```

### Creación de Claves Foráneas

Crear las claves foráneas según el siguiente cuadro:

| TABLA     | NOMBRE              | COLUMNAS       | REFERENCIA |
|-----------|---------------------|----------------|------------|
| MATRICULA | FK_MATRICULA_CURSO  | cur_id         | CURSO      |
| MATRICULA | FK_MATRICULA_ALUMNO | alu_id         | ALUMNO     |
| PAGO      | FK_PAGO_MATRICULA   | cur_id, alu_id | MATRICULA  |



## RESTRICCIONES TIPO UNIQUE

### Sintaxis

```
ALTER TABLE [ esquema . ] < nombre de tabla >  
  ADD CONSTRAINT < nombre >  
    UNIQUE ( < lista de columnas > ) ;  
GO
```

### Creación de Restricciones tipo UNIQUE

Crear las siguientes restricciones de tipo UNIQUE:

| TABLA  | NOMBRE          | COLUMNAS   |
|--------|-----------------|------------|
| CURSO  | U_CURSO_NOMBRE  | cur_nombre |
| ALUMNO | U_ALUMNO_NOMBRE | alu_nombre |
| ALUMNO | U_ALUMNO_EMAIL  | alu_email  |

## RESTRICCIONES TIPO CHECK

### Sintaxis

```
ALTER TABLE [ esquema . ] < nombre de tabla >  
  ADD CONSTRAINT < nombre >  
  CHECK ( < condición > ) ;  
GO
```

### Creación de Restricciones tipo CHECK

| TABLA     | NOMBRE                 | DESCRIPCIÓN   |
|-----------|------------------------|---|
| CURSO     | CHK_CURSO_VACANTES     | Las vacantes deben ser mayor que cero.  |
| CURSO     | CHK_CURSO_MATRICULADOS | Los matriculados deben ser mayor o igual a cero y menor o igual que las vacantes. |
| CURSO     | CHK_CURSO_PRECIO       | El precio debe ser mayor que cero.  |
| MATRICULA | CHK_MATRICULA_PRECIO   | El precio debe ser mayor que cero.  |
| MATRICULA | CHK_MATRICULA_CUOTAS   | Las cuotas deben ser mayor o igual que uno.                                       |
| MATRICULA | CHK_MATRICULA_NOTA     | La nota debe ser NULL o un valor entre 0 y 20.                                    |
| PAGO      | CHK_PAGO_IMPORTE       | El importe debe ser mayor o igual que cero.                                       |

## Creación de Restricciones tipo DEFAULT

### Sintaxis

```
ALTER TABLE [ esquema . ] < nombre de tabla >  
    ADD CONSTRAINT < nombre >  
    DEFAULT ( < valor > ) FOR < nombre de columna > ;  
GO
```

### Ejemplo

Crear una restricción de tipo DEFAULT de nombre D\_CURSO\_MATRICULADOS que asigne un valor cero a la columna CUR\_MATRICULADOS de la tabla CURSO.

## INSERTAR DATOS

### Tabla Curso

```
SET IDENTITY_INSERT dbo.Curso ON;  
GO  
  
INSERT INTO CURSO(CUR_ID,CUR_NOMBRE,CUR_VACANTES,CUR_PRECIO,CUR_PROFESOR)  
VALUES(1,'SQL Server Implementación',24,1000.0,'Gustavo coronel');  
  
INSERT INTO CURSO(cur_id,cur_nombre,cur_vacantes,cur_precio,cur_profesor)  
VALUES(2,'SQL Server Administración',24,1000.0,'Gustavo coronel');  
  
INSERT INTO CURSO(cur_id,cur_nombre,cur_vacantes,cur_precio,cur_profesor)  
VALUES(3,'Inteligencia de Negocios',24,1500.0,'Sergio Matsukawa');  
  
INSERT INTO CURSO(cur_id,cur_nombre,cur_vacantes,cur_precio,cur_profesor)  
VALUES(4,'Programación Transact-SQL',24,1200.0,NULL);  
  
INSERT INTO CURSO(cur_id,cur_nombre,cur_vacantes,cur_precio,cur_profesor)  
VALUES(5,'Java Fundamentos',24,1600.0,'Gustavo Coronel');  
  
INSERT INTO CURSO(cur_id,cur_nombre,cur_vacantes,cur_precio,cur_profesor)  
VALUES(6,'Java Cliente-Servidor',24,1600.0,'Gustavo Coronel');  
  
INSERT INTO CURSO(CUR_ID,CUR_NOMBRE,CUR_VACANTES,CUR_PRECIO,CUR_PROFESOR)  
VALUES(7,'GESTION DE PROYECTOS',24,2200.0,'RICARDO MARCELO');  
GO  
  
SET IDENTITY_INSERT dbo.Curso OFF;  
GO
```

## Tabla Alumno

```
INSERT INTO ALUMNO
(alu_id, alu_nombre, alu_direccion, alu_telefono, alu_email )
VALUES
( 1, 'YESENIA VIRHUEZ', 'LOS OLIVOS', '986412345', 'yesenia@hotmail.com'),
( 2, 'OSCAR ALVARADO FERNANDEZ', 'MIRAFLORES', NULL, 'oscar@gmail.com'),
( 3, 'GLADYS REYES CORTIJO', 'SAN BORJA', '875643562', 'gladys@hotmail.com'),
( 4, 'SARA RIEGA FRIAS', 'SAN ISIDRO', NULL, 'sara@yahoo.com'),
( 5, 'JHON VELASQUEZ DEL CASTILLO', 'LOS OLIVOS', '78645345', 'jhon@movistar.com'),
( 6, 'RODRIGUEZ ROJAS, RENZO ROBERT', 'SURCO', '673465235', 'rrodriguez@gmail.com'),
( 7, 'CALERO MORALES, EMELYN DALILA', 'LA MOLINA', '896754652', 'ecalero@peru.com'),
( 8, 'KAREN FUENTES', 'San Isidro', '555-5555', 'KAFUENTES@HOTMAIL.COM'),
( 9, 'Yamina Ruiz', 'San Isidro', '965-4521', 'yami_ruiz@gmail.com'),
(10, 'MARIA EULALIA VELASQUEZ TORVISCO', 'SURCO', '6573456', 'mvelasques@gmail.com'),
(11, 'FIORELLA LIZET VITELLA REYES', 'SAN BORJA', '5468790', 'fvitela@outlook.com');
GO
```

## Tabla Matricula

```
SET DATEFORMAT DMY
GO

DECLARE @ANIO VARCHAR(10);
SET @ANIO = cast(year(getdate()) as varchar);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(1, 5, '15-04-' + @ANIO + ' 10:30', 800.0, 1, 15);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(1, 3, '16-04-' + @ANIO + ' 11:45', 1000.0, 2, 18);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(1, 4, '18-04-' + @ANIO + ' 08:33', 1200.0, 3, 12);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(2, 1, '15-04-' + @ANIO + ' 12:33', 800.0, 1, 16);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(2, 2, '01-05-' + @ANIO + ' 15:34', 1000.0, 2, 10);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(2, 3, '03-05-' + @ANIO + ' 16:55', 1300.0, 3, 14);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(2, 4, '04-05-' + @ANIO + ' 17:00', 400.0, 1, 18);

INSERT INTO dbo.MATRICULA ( cur_id, alu_id, mat_fecha, mat_precio, mat_cuotas,
mat_nota ) VALUES(2, 5, '06-05-' + @ANIO + ' 13:12', 750.0, 1, 17);

GO
```

## Actualizar Tabla Curso

Se debe actualizar la columna **cur\_matriculados** en la tabla Curso.

```
UPDATE dbo.CURSO
SET cur_matriculados = (
    SELECT COUNT(*) FROM dbo.MATRICULA
    WHERE dbo.MATRICULA.cur_id = dbo.CURSO.cur_id );
GO
```

## Tabla Pago

```
SET DATEFORMAT DMY
GO

DECLARE @ANIO VARCHAR(10)
SET @ANIO = CAST(YEAR(GETDATE()) AS VARCHAR)

INSERT INTO DBO.PAGO VALUES(1,3,1,'16-04-' + @ANIO,500)
INSERT INTO DBO.PAGO VALUES(1,3,2,'16-05-' + @ANIO,500)
INSERT INTO DBO.PAGO VALUES(1,4,1,'18-04-' + @ANIO,400)
INSERT INTO DBO.PAGO VALUES(1,4,2,'18-05-' + @ANIO,400)
INSERT INTO DBO.PAGO VALUES(2,1,1,'15-04-' + @ANIO,800)
INSERT INTO DBO.PAGO VALUES(2,2,1,'01-05-' + @ANIO,500)
INSERT INTO DBO.PAGO VALUES(2,3,1,'03-05-' + @ANIO,430)
INSERT INTO DBO.PAGO VALUES(2,3,2,'03-06-' + @ANIO,430)
INSERT INTO DBO.PAGO VALUES(2,4,1,'04-05-' + @ANIO,400)
INSERT INTO DBO.PAGO VALUES(2,5,1,'06-05-' + @ANIO,750)
GO
```

## SECUENCIAS

### Sintaxis

```
CREATE SEQUENCE [ esquema . ] < nombre de secuencia >  
START WITH < valor de inicio >  
INCREMENT BY < valor de incremento > ;  
GO
```

### Ejemplo

```
CREATE SEQUENCE dbo.sq_alumno  
    START WITH 20  
    INCREMENT BY 1;  
GO  
  
INSERT INTO ALUMNO (alu_id, alu_nombre, alu_direccion, alu_telefono, alu_email )  
VALUES ( next value for dbo.sq_alumno, 'SHARON TINEO', 'SURCO', '956389564',  
'karla@gmail.com' );  
go  
  
INSERT INTO ALUMNO (alu_id, alu_nombre, alu_direccion, alu_telefono, alu_email )  
VALUES ( next value for dbo.sq_alumno , 'KARLA MORALES', 'LOS OLIVOS', '93564356',  
'claudia@gmail.com' );  
GO  
  
SELECT * FROM ALUMNO;  
GO
```