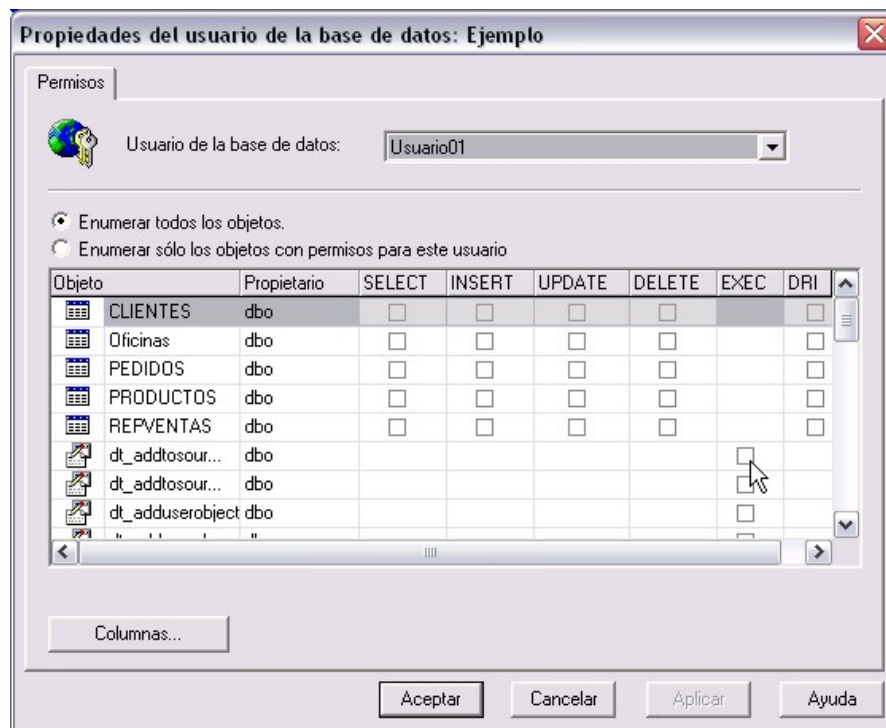


Luego de ello podrá observar una ventana donde podrá permitir o negar accesos sobre los objetos. Tal como lo muestra la siguiente imagen:



EJERCICIOS PROPUESTOS

Crear la base de datos Matriculas con el archivo de datos de 10 Mb., tamaño máximo de 15 Mb. y un crecimiento de 1 Mb. El archivo de log debe tener inicialmente un tamaño de 3 Mb, con un tamaño máximo de 5 Mb. y un crecimiento de 1 Mb.

Una vez creada la base de datos defina la creación de las siguientes tablas:

ALUMNOS

Diseñar tabla 'Alumnos' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
codalu	char	4	
nom	varchar	20	
pat	varchar	20	
mat	varchar	20	

Columnas

Descripción

Valor predeterminado

Precisión 0

Escala 0

Identidad No

Inicialización de identidad

Incremento de identidad

Es RowGuid No

Fórmula

Intercalación SQL_Latin1_General_CP1_CI_AS

CALIFICACIONES

Diseñar tabla 'Calificaciones' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
codalu	char	4	
sec	char	4	
n1	int	4	
n2	int	4	

Columnas

Descripción

Valor predeterminado

Precisión 0

Escala 0

Identidad No

Inicialización de identidad

Incremento de identidad

Es RowGuid No

Fórmula

Intercalación SQL_Latin1_General_CP1_CI_AS

CATEGORIAS

Diseñar tabla 'Categorias' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
categ	char	3	
pagoporhora	money	8	

Columnas

Descripción

Valor predeterminado

Precisión

Escala

Identidad

Inicialización de identidad

Incremento de identidad

Es RowGuid

Fórmula

Intercalación

SQL_Latin1_General_CP1_CI_AS

INSCRITOS

Diseñar tabla 'Inscritos' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
codalu	char	4	
sec	char	4	

Columnas

Descripción

Valor predeterminado

Precisión

Escala

Identidad

Inicialización de identidad

Incremento de identidad

Es RowGuid


Fórmula

Intercalación

SQL_Latin1_General_CP1_CI_AS

PROFESORES

Diseñar tabla 'Profesores' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
 codprofe	char	4	
nombre	varchar	40	
categ	char	3	
fec_ing	datetime	8	
direc	varchar	50	
telef	char	8	

Columnas

Descripción

Valor predeterminado

Precisión

Escala

Identidad

Inicialización de identidad

Incremento de identidad

Es RowGuid


Fórmula

Intercalación

SQL_Latin1_General_CP1_CI_AS

SECCIONES

Diseñar tabla 'Secciones' en 'Matriculas' en '(LOCAL)'

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
 sec	char	4	
codprofe	char	4	
inicio	datetime	8	
pension	money	8	
vacantes	int	4	
numhoras	int	4	

Columnas

Descripción

Valor predeterminado

Precisión

Escala

Identidad

Inicialización de identidad

Incremento de identidad

Es RowGuid

Fórmula

Intercalación

SQL_Latin1_General_CP1_CI_AS

Una vez que termine recupere el script Alumnado.sql, revise el script y ejecútelo con la finalidad de poblar la base de datos.

Bien una vez que hemos creado las tablas y hemos cargado los valores, debemos implementar las restricciones (*CONSTRAINTS*) correspondientes a las reglas de integridad de cada tabla. Iniciaremos asignando las restricciones de tipo clave Principal (*PRIMARY KEY*):

```
Use Matriculas
GO
ALTER TABLE Alumnos
ADD CONSTRAINT PK_Alumnos_codalu
PRIMARY KEY (codalu)
GO

ALTER TABLE Secciones
ADD CONSTRAINT PK_Secciones_sec
PRIMARY KEY (sec)
GO

ALTER TABLE Profesores
ADD CONSTRAINT PK_Profesores_codprofe
PRIMARY KEY (codprofe)
GO
ALTER TABLE Categorías
ADD CONSTRAINT PK_Categorías_categ
PRIMARY KEY (categ)
GO

ALTER TABLE Inscritos
ADD CONSTRAINT PK_Inscritos_alusec
PRIMARY KEY (codalu, sec)
GO

ALTER TABLE Calificaciones
ADD CONSTRAINT PK_Calificaciones_alusec
PRIMARY KEY (codalu, sec)
GO
```

Después de crear las definiciones para asegurar la unicidad de cada fila, nos toca implementar las restricciones de claves foráneas (*FOREIGN KEY constraint*).

```
ALTER TABLE Inscritos
ADD CONSTRAINT FK_Alumnos_codalu
FOREIGN KEY (codalu)
REFERENCES Alumnos(codalu)
GO

ALTER TABLE Inscritos
ADD CONSTRAINT FK_Secciones_sec
FOREIGN KEY (sec)
REFERENCES Secciones(sec)
GO

ALTER TABLE Secciones
ADD CONSTRAINT FK_Profesores_codprofe
```

```

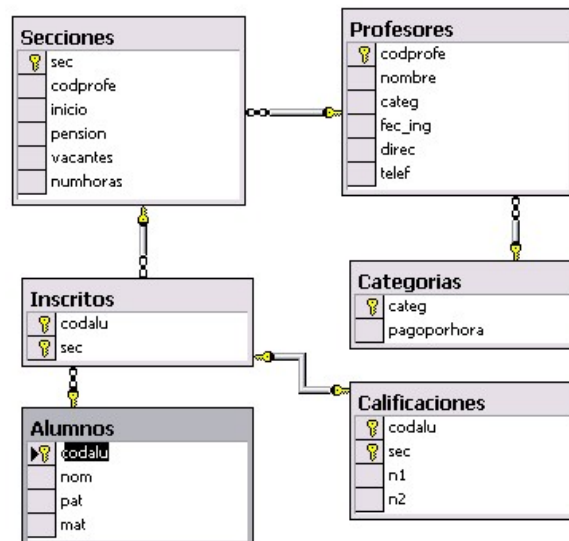
FOREIGN KEY (codprofe)
REFERENCES profesores(codprofe)
GO

ALTER TABLE Calificaciones
ADD CONSTRAINT FK_Inscritos_codalu
FOREIGN KEY (codalu, sec)
REFERENCES Inscritos(codalu, sec)
GO

ALTER TABLE Profesores
ADD CONSTRAINT FK_Categorias_categ
FOREIGN KEY (categ)
REFERENCES Categorias(categ)
GO

```

Finalmente genere el diagrama de base de datos que debe observarse como el siguiente:



Para probar el funcionamiento de los constraints implementados ejecute las siguientes instrucciones:

```

Insert Inscritos Values ('A001', '1116')
GO

```

```

/* El problema se debe a que la llave de inscripcion es duplicada por tanto podemos
deducir que el alumno ya estudio en esa aula. En este caso trabaja PK_Inscritos_alusec.
*/

```

```

Insert Calificaciones Values ('A003', '2315', 10,10)
GO

```

```

/* El problema es que el alumno no ha sido inscrito en la sección especificada, en este
caso trabaja el FK_Inscritos_codalu. */

```

```

Insert Secciones Values ('2318', 'D009', '5/15/2001', 600, 15, 40)

```