



PROYECTO MODELADO CAMPING_NFR

Índice

[Introducción](#)
[Información](#)
[Entidades y Relaciones](#)
[DATA MODELER](#)
[Modelo Lógico](#)
[Modelo Relacional](#)
[Notas](#)
[SQL SERVER](#)
[Database Diagram](#)
[Script](#)
[BACKUP](#)
[Modelo NoSQL - MongoDB](#)
[Colecciones](#)

Introducción

Este proyecto se basa en camping, una opción de vacaciones para todo tipo de personas con todas las opciones para disfrutar de su estancia.

Desde el típico caso de la familia o pareja que quiere disfrutar del aire libre en una tienda de campaña, como los que prefieren disfrutar de las opciones de una estancia más cómoda en bungalow.

Información

Entidades y Relaciones

Tenemos las entidades:

Clientes, donde guardaremos los datos de los clientes y a nombre de quien irá la factura.

Empleados, aquí irán los empleados y se relacionaran con otras tablas como puesto de trabajo o en qué instalación del camping desarrollan su función.

Reserva Huéspedes: Para poder saber siempre cuantas personas hay en el camping y poder filtrarlas tanto por edad como por género

Para aplicar la regla de la normalización , los campos de la dirección los hemos sacado para otra tabla.

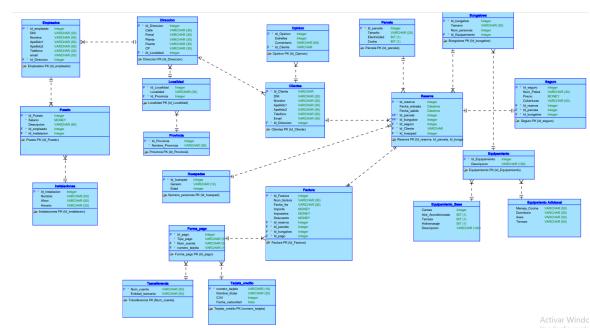
Nos interesa también registrar las opiniones de nuestros clientes., para poder ver en qué podemos mejorar.

En cuanto al Cliente, puede hacer una o varias reservas (la cual lleva siempre asociada un seguro). De la reserva, sacaremos una factura y una forma de pago (que puede ser por transferencia o tarjeta)

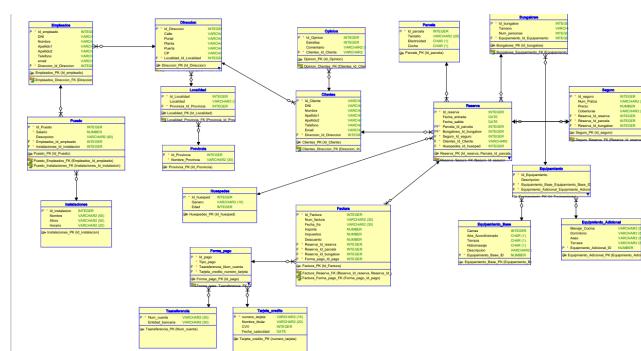
Las reservas pueden ser de parcelas o de bungalow, no pudiendo reservar por cliente mas de una de cada.

DATA MODELER

Modelo Lógico



Modelo Relacional

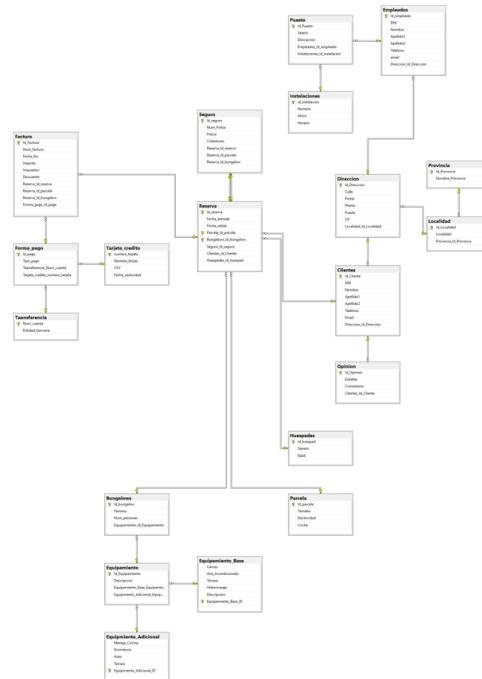


Notas

Me he dado cuenta de que me han quedado campos marcados como "obligatorios" que no me interesa. En la siguiente parte, cuando introduzca datos, lo cambiaré.

SQL SERVER

Database Diagram



Script

Para la creación de la base de datos

```
USE master
GO

DROP DATABASE IF EXISTS Camping_NFR
GO

CREATE DATABASE Camping_NFR
GO

USE Camping_NFR
GO
```

Adjuntamos el script creado en el ddl

```
-- Generado por Oracle SQL Developer Data Modeler 21.2.0.183.1957
-- en:          2022-03-17 20:42:55 CET
-- sitio:       SQL Server 2012
-- tipo:        SQL Server 2012

CREATE TABLE Bungalows
(
    Id_bungalow INTEGER NOT NULL ,
    Tamano VARCHAR (20) ,
    Num_personas INTEGER ,
    Equipamiento_Id_Equipamiento INTEGER NOT NULL
)
GO
```

```

EXEC sp_addextendedproperty 'MS_Description' , 'nombre del tipo de bungalow' , 'USER' , 'dbo' , 'TABLE' , 'Bungalows' , 'COLUMN' , 'Id
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Tamano en metros cuadrados' , 'USER' , 'dbo' , 'TABLE' , 'Bungalows' , 'COLUMN' , 'Tam
GO

EXEC sp_addextendedproperty 'MS_Description' , 'Capacidad de persoas' , 'USER' , 'dbo' , 'TABLE' , 'Bungalows' , 'COLUMN' , 'Num_perso
GO

ALTER TABLE Bungalows ADD CONSTRAINT Bungalows_PK PRIMARY KEY CLUSTERED (Id_bungalow)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Clientes
(
    Id_Cliente VARCHAR NOT NULL ,
    DNI VARCHAR (20) ,
    Nombre VARCHAR (50) ,
    Apellido1 VARCHAR (50) ,
    Apellido2 VARCHAR (50) ,
    Telefono VARCHAR (50) ,
    Email VARCHAR (50) ,
    Direccion_Id_Direccion INTEGER NOT NULL
)
GO

ALTER TABLE Clientes ADD CONSTRAINT Clientes_PK PRIMARY KEY CLUSTERED (Id_Cliente)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Direccion
(
    Id_Direccion INTEGER NOT NULL ,
    Calle VARCHAR (30) ,
    Portal VARCHAR (30) ,
    Planta VARCHAR (30) ,
    Puerta VARCHAR (30) ,
    CP VARCHAR (30) ,
    Localidad_Id_Localidad INTEGER NOT NULL
)
GO

ALTER TABLE Direccion ADD CONSTRAINT Direccion_PK PRIMARY KEY CLUSTERED (Id_Direccion)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Empleados
(
    Id_empleado INTEGER NOT NULL ,
    DNI VARCHAR (20) ,
    Nombre VARCHAR (50) ,
    Apellido1 VARCHAR (50) ,
    Apellido2 VARCHAR (50) ,
    Telefono VARCHAR (20) ,
    email VARCHAR (50) ,
    Direccion_Id_Direccion INTEGER NOT NULL
)
GO

ALTER TABLE Empleados ADD CONSTRAINT Empleados_PK PRIMARY KEY CLUSTERED (Id_empleado)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Equipamiento
(
    Id_Equipamiento INTEGER NOT NULL ,
    Descripcion VARCHAR (100) ,
    Equipamiento_Base_Equipamiento_Base_ID NUMERIC (28) NOT NULL ,
    Equipamiento_Adicional_Equipamiento_Adicional_ID NUMERIC (28) NOT NULL
)
GO

ALTER TABLE Equipamiento ADD CONSTRAINT Equipamiento_PK PRIMARY KEY CLUSTERED (Id_Equipamiento)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )

```

```

        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Equipamiento_Base
(
    Camas INTEGER ,
    Aire_Acondicionado BIT ,
    Terraza BIT ,
    Hidromasaje BIT ,
    Descripcion VARCHAR (100) ,
    Equipamiento_Base_ID NUMERIC (28) NOT NULL IDENTITY NOT FOR REPLICATION
)
GO

ALTER TABLE Equipamiento_Base ADD CONSTRAINT Equipamiento_Base_PK PRIMARY KEY CLUSTERED (Equipamiento_Base_ID)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Equipamiento_Adicional
(
    Menaje_Cocina VARCHAR (50) ,
    Dormitorio VARCHAR (50) ,
    Aseo VARCHAR (50) ,
    Terraza VARCHAR (50) ,
    Equipamiento_Adicional_ID NUMERIC (28) NOT NULL IDENTITY NOT FOR REPLICATION
)
GO

ALTER TABLE Equipamiento_Adicional ADD CONSTRAINT Equipamiento_Adicional_PK PRIMARY KEY CLUSTERED (Equipamiento_Adicional_ID)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Factura
(
    Id_Factura INTEGER NOT NULL ,
    Num_factura VARCHAR (30) ,
    Fecha_fra VARCHAR (30) ,
    Importe MONEY ,
    Impuestos MONEY ,
    Descuento MONEY ,
    Reserva_Id_reserva INTEGER NOT NULL ,
    Reserva_Id_parcela INTEGER NOT NULL ,
    Reserva_Id_bungalow INTEGER NOT NULL ,
    Forma_pago_Id_pago INTEGER NOT NULL
)
GO

ALTER TABLE Factura ADD CONSTRAINT Factura_PK PRIMARY KEY CLUSTERED (Id_Factura)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Forma_pago
(
    Id_pago INTEGER NOT NULL ,
    Tipo_pago VARCHAR (30) NOT NULL ,
    Taansferencia_Num_cuenta VARCHAR (50) NOT NULL ,
    Tarjeta_credito_numero_tarjeta VARCHAR (16) NOT NULL
)
GO

ALTER TABLE Forma_pago ADD CONSTRAINT Forma_pago_PK PRIMARY KEY CLUSTERED (Id_pago)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Huespedes
(
    Id_huesped INTEGER NOT NULL ,
    Genero VARCHAR (10) ,
    Edad INTEGER
)
GO

ALTER TABLE Huespedes ADD CONSTRAINT Huespedes_PK PRIMARY KEY CLUSTERED (Id_huesped)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Instalaciones

```

```

(
    Id_instalacion INTEGER NOT NULL ,
    Nombre VARCHAR (50) ,
    Aforo VARCHAR (50) ,
    Horario VARCHAR (20)
)
GO

ALTER TABLE Instalaciones ADD CONSTRAINT Instalaciones_PK PRIMARY KEY CLUSTERED (Id_instalacion)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Localidad
(
    Id_Localidad INTEGER NOT NULL ,
    Localidad VARCHAR (30) ,
    Provincia_Id_Provincia INTEGER NOT NULL
)
GO

ALTER TABLE Localidad ADD CONSTRAINT Localidad_PK PRIMARY KEY CLUSTERED (Id_Localidad)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Opinion
(
    Id_Opinion INTEGER NOT NULL ,
    Estrellas INTEGER ,
    Comentario VARCHAR (80) ,
    Clientes_Id_Cliente VARCHAR NOT NULL
)
GO

ALTER TABLE Opinion ADD CONSTRAINT Opinion_PK PRIMARY KEY CLUSTERED (Id_Opinion)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Parcela
(
    Id_parcela INTEGER NOT NULL ,
    Tamaño VARCHAR (20) ,
    Electricidad BIT ,
    Coche BIT
)
GO

EXEC sp_addextendedproperty 'MS_Description' , 'tamaño parcela en metros cuadrados' , 'USER' , 'dbo' , 'TABLE' , 'Parcela' , 'COLUMN'
GO

EXEC sp_addextendedproperty 'MS_Description' , 'incluye coche' , 'USER' , 'dbo' , 'TABLE' , 'Parcela' , 'COLUMN' , 'Coche'
GO

ALTER TABLE Parcela ADD CONSTRAINT Parcela_PK PRIMARY KEY CLUSTERED (Id_parcela)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Provincia
(
    Id_Provincia INTEGER NOT NULL ,
    Nombre_Provincia VARCHAR (30) NOT NULL
)
GO

ALTER TABLE Provincia ADD CONSTRAINT Provincia_PK PRIMARY KEY CLUSTERED (Id_Provincia)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Puesto
(
    Id_Puesto INTEGER NOT NULL ,
    Salario MONEY NOT NULL ,
    Descripcion VARCHAR (60) ,
    Empleados_Id_empleado INTEGER NOT NULL ,

```

```

        Instalaciones_Id_instalacion INTEGER NOT NULL
    )
GO

ALTER TABLE Puesto ADD CONSTRAINT Puesto_PK PRIMARY KEY CLUSTERED (Id_Puesto)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Reserva
(
    Id_reserva INTEGER NOT NULL ,
    Fecha_entrada DATETIME ,
    Fecha_salida DATETIME ,
    Parcela_Id_parcela INTEGER NOT NULL ,
    Bungalows_Id_bungalow INTEGER NOT NULL ,
    Seguro_Id_seguro INTEGER NOT NULL ,
    Clientes_Id_Cliente VARCHAR NOT NULL ,
    Huespedes_Id_huesped INTEGER NOT NULL
)
GO

EXEC sp_adddextendedproperty 'MS_Description' , 'Fecha entrada' , 'USER' , 'dbo' , 'TABLE' , 'Reserva' , 'COLUMN' , 'Fecha_entrada'
GO

EXEC sp_adddextendedproperty 'MS_Description' , 'Fechas salida' , 'USER' , 'dbo' , 'TABLE' , 'Reserva' , 'COLUMN' , 'Fecha_salida'
GO

CREATE UNIQUE NONCLUSTERED INDEX
    Reserva__IDX ON Reserva
    (
        Seguro_Id_seguro
    )
GO

ALTER TABLE Reserva ADD CONSTRAINT Reserva_PK PRIMARY KEY CLUSTERED (Id_reserva, Parcela_Id_parcela, Bungalows_Id_bungalow)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Seguro
(
    Id_seguro INTEGER NOT NULL ,
    Num_Poliza VARCHAR (30) ,
    Precio MONEY ,
    Coberturas VARCHAR (50) ,
    Reserva_Id_reserva INTEGER NOT NULL ,
    Reserva_Id_parcela INTEGER NOT NULL ,
    Reserva_Id_bungalow INTEGER NOT NULL
)
GO

CREATE UNIQUE NONCLUSTERED INDEX
    Seguro__IDX ON Seguro
    (
        Reserva_Id_reserva ,
        Reserva_Id_parcela ,
        Reserva_Id_bungalow
    )
GO

ALTER TABLE Seguro ADD CONSTRAINT Seguro_PK PRIMARY KEY CLUSTERED (Id_seguro)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Taansferencia
(
    Num_cuenta VARCHAR (50) NOT NULL ,
    Entidad_bancaria VARCHAR (50)
)
GO

ALTER TABLE Taansferencia ADD CONSTRAINT Taansferencia_PK PRIMARY KEY CLUSTERED (Num_cuenta)

```

```

WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Tarjeta_credito
(
    numero_tarjeta VARCHAR (16) NOT NULL ,
    Nombre_titular VARCHAR (20) ,
    CVV INTEGER ,
    Fecha_caducidad DATE
)
GO

ALTER TABLE Tarjeta_credito ADD CONSTRAINT Tarjeta_credito_PK PRIMARY KEY CLUSTERED (numero_tarjeta)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO

ALTER TABLE Bungalows
ADD CONSTRAINT Bungalows_Equipamiento_FK FOREIGN KEY
(
    Equipamiento_Id_Equipamiento
)
REFERENCES Equipamiento
(
    Id_Equipamiento
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Clientes
ADD CONSTRAINT Clientes_Direccion_FK FOREIGN KEY
(
    Direccion_Id_Direccion
)
REFERENCES Direccion
(
    Id_Direccion
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Direccion
ADD CONSTRAINT Direccion_Localidad_FK FOREIGN KEY
(
    Localidad_Id_Localidad
)
REFERENCES Localidad
(
    Id_Localidad
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Empleados
ADD CONSTRAINT Empleados_Direccion_FK FOREIGN KEY
(
    Direccion_Id_Direccion
)
REFERENCES Direccion
(
    Id_Direccion
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Equipamiento
ADD CONSTRAINT Equipamiento_Equipamiento_Base_FK FOREIGN KEY
(
    Equipamiento_Base_Equipamiento_Base_ID
)
REFERENCES Equipamiento_Base
(
    Equipamiento_Base_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Equipamiento
ADD CONSTRAINT Equipamiento_Equipamiento_Adicional_FK FOREIGN KEY

```

```

(
    Equipment_Adicional_Equipment_Adicional_ID
)
REFERENCES Equipment_Adicional
(
    Equipment_Adicional_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Factura
ADD CONSTRAINT Factura_Forma_pago_FK FOREIGN KEY
(
    Forma_pago_Id_pago
)
REFERENCES Forma_pago
(
    Id_pago
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Factura
ADD CONSTRAINT Factura_Reserva_FK FOREIGN KEY
(
    Reserva_Id_reserva,
    Reserva_Id_parcela,
    Reserva_Id_bungalow
)
REFERENCES Reserva
(
    Id_reserva ,
    Parcela_Id_parcela ,
    Bungalows_Id_bungalow
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Forma_pago
ADD CONSTRAINT Forma_pago_Taansferencia_FK FOREIGN KEY
(
    Taansferencia_Num_cuenta
)
REFERENCES Taansferencia
(
    Num_cuenta
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Forma_pago
ADD CONSTRAINT Forma_pago_Tarjeta_credito_FK FOREIGN KEY
(
    Tarjeta_credito_numero_tarjeta
)
REFERENCES Tarjeta_credito
(
    numero_tarjeta
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Localidad
ADD CONSTRAINT Localidad_Provincia_FK FOREIGN KEY
(
    Provincia_Id_Provincia
)
REFERENCES Provincia
(
    Id_Provincia
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Opinion
ADD CONSTRAINT Opinion_Clientes_FK FOREIGN KEY
(
    Clientes_Id_Cliente
)
REFERENCES Clientes
(

```

```

        Id_Cliente
    )
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Puesto
ADD CONSTRAINT Puesto_Empleados_FK FOREIGN KEY
(
    Empleados_Id_empleado
)
REFERENCES Empleados
(
    Id_empleado
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Puesto
ADD CONSTRAINT Puesto_Instalaciones_FK FOREIGN KEY
(
    Instalaciones_Id_instalacion
)
REFERENCES Instalaciones
(
    Id_instalacion
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Reserva
ADD CONSTRAINT Reserva_Bungalows_FK FOREIGN KEY
(
    Bungalows_Id_bungalow
)
REFERENCES Bungalows
(
    Id_bungalow
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Reserva
ADD CONSTRAINT Reserva_Clientes_FK FOREIGN KEY
(
    Clientes_Id_Cliente
)
REFERENCES Clientes
(
    Id_Cliente
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Reserva
ADD CONSTRAINT Reserva_Huespedes_FK FOREIGN KEY
(
    Huespedes_Id_huesped
)
REFERENCES Huespedes
(
    Id_huesped
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Reserva
ADD CONSTRAINT Reserva_Parcela_FK FOREIGN KEY
(
    Parcela_Id_parcela
)
REFERENCES Parcela
(
    Id_parcela
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Reserva
ADD CONSTRAINT Reserva_Seguro_FK FOREIGN KEY
(

```

```

        Seguro_Id_seguro
    )
REFERENCES Seguro
(
    Id_seguro
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Seguro
ADD CONSTRAINT Seguro_Reserva_FK FOREIGN KEY
(
    Reserva_Id_reserva,
    Reserva_Id_parcela,
    Reserva_Id_bungalow
)
REFERENCES Reserva
(
    Id_reserva ,
    Parcela_Id_parcela ,
    Bungalows_Id_bungalow
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

-- Informe de Resumen de Oracle SQL Developer Data Modeler:
--
-- CREATE TABLE          20
-- CREATE INDEX          2
-- ALTER TABLE          40
-- CREATE VIEW           0
-- ALTER VIEW            0
-- CREATE PACKAGE        0
-- CREATE PACKAGE BODY   0
-- CREATE PROCEDURE      0
-- CREATE FUNCTION       0
-- CREATE TRIGGER        0
-- ALTER TRIGGER         0
-- CREATE DATABASE       0
-- CREATE DEFAULT        0
-- CREATE INDEX ON VIEW  0
-- CREATE ROLLBACK SEGMENT 0
-- CREATE ROLE           0
-- CREATE RULE           0
-- CREATE SCHEMA         0
-- CREATE SEQUENCE       0
-- CREATE PARTITION FUNCTION 0
-- CREATE PARTITION SCHEME 0
--
-- DROP DATABASE         0
--
-- ERRORS                0
-- WARNINGS              0

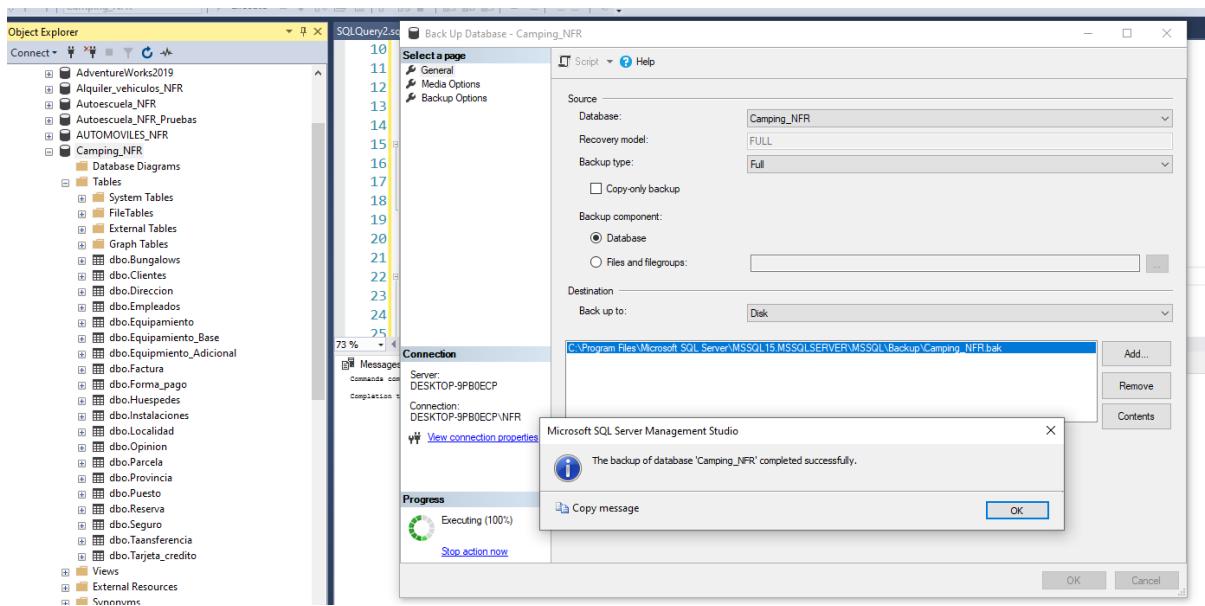
--Commands completed successfully.

--Completion time: 2022-03-17T20:46:25.1664531+01:00

```

BACKUP

Desde aquí generamos el backup, que adjuntamos en la carpeta del proyecto



Modelo NoSQL - MongoDB

Un documento en MongoDB no puede ocupar más de 16 Mb, es embebido e intenta hacer el menor número de operaciones. No hay FK

Primero consultamos la versión

```
db.hostInfo()
db.version()
```

```
campingnfr> db.version()
5.0.5
campingnfr> db.hostInfo
[Function: hostInfo] AsyncFunction {
  apiVersions: [ 0, 0 ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplicaSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 0, 1, 2 ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
campingnfr> .
```

Ahora creamos la BBDD **campingnfr** y compruebo que está creada

```

academianfr_bd> use campingnfr
switched to db campingnfrnot defined
campingnfr> show dbs
PetHotel          41 kB
academianfr_bd   81.9 kB
admin            41 kB
config           111 kB
local            90.1 kB
test             65.5 kB
usuario          57.3 kB
campingnfr> .

```

Colecciones

En este caso, como la BBDD del proyecto es bastante grande, así que he optado por hacer una "parte".

He creado las colecciones clientes y empleados.

En clientes he embebido opinión, reserva y factura, mientras que en empleado he incluído puesto e instalación en la que trabaja.

Creamos las colecciones y incluimos los datos

```

db.clientes.insertOne({
  _id:1,
  "nombre":"Nuria",
  "apellidos":["Ferreiro","Rodriguez"],
  "telefonos":"666666666",
  "email":["nuria@gmail.com"],

  "opinion":[{
    "estrellas":"5",
    "comentario":"el mejor camping"
  }], 

  "reserva":[{
    "Fecha_entrada":"2020-05-05",
    "Fecha_salida":"2020-05-15",
    "Parcela_id_parcela":"1",
    "Seguro_Id_Seguro":"1",
    "Cliente_Id_Cliente":"1",
    "Huespedes_Id_Huespedes":"2",
  }],
  
  "factura":[{
    "id_factura":"5",
    "num_factura":"26",
    "Fecha_fra":"2020-05-15",
    "Importe":"250",
    "Descuento":"30"
  }],
});

db.clientes.insertOne({
  _id:2,
  "nombre":"Juan",
  "apellidos":["Perez","Rodriguez"],
  "telefonos":"666666999",
  "email":["juan@gmail.com"],

  "opinion":[{
    "estrellas":"4",
    "comentario":"casi el mejor camping"
  }],
});

```

```
"reserva": [{  
    "Fecha_entrada": "2020-06-06",  
    "Fecha_salida": "2020-06-06",  
    "Parcela_id_parcela": "2",  
    "Seguro_Id_Seguro": "2",  
    "Cliente_Id_Cliente": "2",  
    "Huespedes_Id_Huespedes": "4",  
},  
  
],  
  
"factura": [{  
    "id_factura": "6",  
    "num_factura": "27",  
    "Fecha_fra": "2020-06-06",  
    "Importe": "400",  
    "Descuento": "50"  
},  
  
],  
  
});  
db.clientes.find().pretty()
```

```
db.clientes.find().pretty()
```

y vemos el resultado

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
campingnfr> db.clientes.find().pretty()
[
  {
    _id: 1,
    nombre: 'Nuria',
    apellidos: [ 'Ferreiro', 'Rodriguez' ],
    telefonos: '66666666',
    email: [ 'nuria@gmail.com' ],
    opinion: [ { estrellas: '5', comentario: 'el mejor camping' } ],
    reserva: [
      {
        Fecha_entrada: '2020-05-05',
        Fecha_salida: '2020-05-15',
        Parcela_Id_parcela: '1',
        Seguro_Id_Seguro: '1',
        Cliente_Id_Cliente: '1',
        Huespedes_Id_Huespedes: '2'
      }
    ],
    factura: [
      {
        id_factura: '5',
        num_factura: '26',
        Fecha_fra: '2020-05-15',
        Importe: '250',
        Descuento: '30'
      }
    ]
  },
  {
    _id: 2,
    nombre: 'Juan',
    apellidos: [ 'Perez', 'Rodriguez' ],
    telefonos: '666666999',
    email: [ 'juan@gmail.com' ],
    opinion: [ { estrellas: '4', comentario: 'casi el mejor camping' } ],
    reserva: [
      {
        Fecha_entrada: '2020-06-06',
        Fecha_salida: '2020-06-06',
        Parcela_Id_parcela: '2',
        Seguro_Id_Seguro: '2',
        Cliente_Id_Cliente: '2',
        Huespedes_Id_Huespedes: '4'
      }
    ],
    factura: [
      {
        id_factura: '6',
        num_factura: '27',
        Fecha_fra: '2020-06-06',
        Importe: '400',
        Descuento: '50'
      }
    ]
  }
]
campingnfr>

```

ahora lo hacemos con empleados

```

db.empleados.insertOne({
  _id:1,
  "nombre":"Noa",
  "apellidos":["Rey","Ferreiro"],
  "telefonos":"666999999",
  "email":"noa@gmail.com",
  "puesto": [
    {
      "id_puesto":"1",
      "salario":"1200",
      "descripcion":"Vigilante"
    }
  ],
  "instalacion":"camping"
});

db.empleados.insertOne({

```

```

_id:2,
"nombre":"Lola",
"apellidos":["Lopez","Sanchez"],
"telefonos":"666999666",
"email":"lola@gmail.com",
"puesto":
  [
    {
      "id_puesto":"2",
      "salario":"1400",
      "descripcion":"socorrista"
    }
  ],
"instalacion":"piscina"
}
);

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```

campingnfr> db.empleados.insertOne({
...   _id:1,
...   "nombre":"Noa",
...   "apellidos":["Rey","Ferreiro"],
...   "telefonos":"666999999",
...   "email":"noa@gmail.com",
...   "puesto":
...     [
...       {
...         "id_puesto":"1",
...         "salario":"1200",
...         "descripcion":"Vigilante"
...       }
...     ],
...   "instalacion":"camping"
...
...
...
...
{
  acknowledged: true,
  insertedId: 1
}
campingnfr>

campingnfr>
campingnfr> db.empleados.insertOne({
...   _id:2,
...   "nombre":"Lola",
...   "apellidos":["Lopez","Sanchez"],
...   "telefonos":"666999666",
...   "email":"lola@gmail.com",
...   "puesto":
...     [
...       {
...         "id_puesto":"2",
...         "salario":"1400",
...         "descripcion":"socorrista"
...       }
...     ],
...   "instalacion":"piscina"
...
...
...
...
{
  acknowledged: true,
  insertedId: 2
}
campingnfr>

```

Consultamos y nos dice que hay dos y cuales son

```

db.empleados.find().pretty().count()
db.empleados.find().pretty()

```

```
campingnfr> db.empleados.find().pretty().count()
2
campingnfr> db.empleados.find().pretty()
[
  {
    _id: 1,
    nombre: 'Noa',
    apellidos: [ 'Rey', 'Ferreiro' ],
    telefonos: '666999999',
    email: 'noa@gmail.com',
    puesto: [ { id_puesto: '1', salario: '1200', descripcion: 'Vigilante' } ],
    instalacion: 'camping'
  },
  {
    _id: 2,
    nombre: 'Lola',
    apellidos: [ 'Lopez', 'Sanchez' ],
    telefonos: '666999666',
    email: 'lola@gmail.com',
    puesto: [ { id_puesto: '2', salario: '1400', descripcion: 'socorrista' } ],
    instalacion: 'piscina'
  }
]
campingnfr>
```