

UNIVERSIDAD TECNOLOGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA

Carrera de Ingeniería en Sistemas de Información

Materia: BASE DE DATOS APLICADA

Trabajo Práctico Integrador Final

Elaborado por:

Acosta Luis

Benitez Peressi, Gonzalo

Cedaro, Gonzalo

Cuzziol, Eric

Núñez, Darío Germán

Presentado ante:

Ing. Carlos Perez

Ing. Andrés Fantín

~2015



Índice

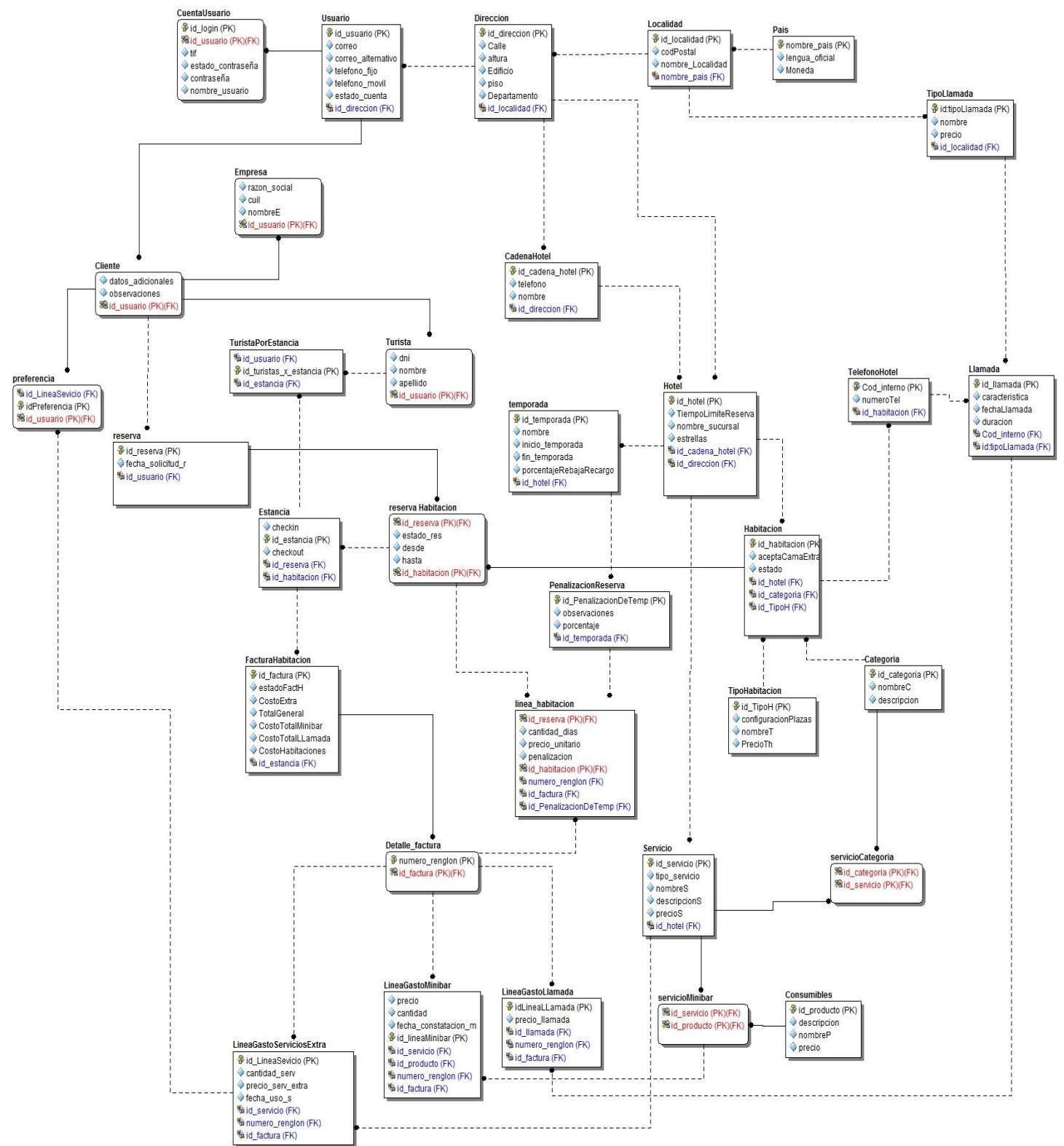
Modelo Físico de la Base de Datos-----	3
Diccionario de Datos-----	4
Script de creación de la Base de Datos-----	8
Objetos de la Base de Datos-----	17
Consultas SQL-----	22
Vistas-----	27
Índices-----	29
Triggers-----	32
Procedimientos Almacenados-----	57
XML-----	67
Seguridad-----	75
Backup-----	85



Modelo Físico de la Base de Datos

A continuación se muestra el modelo físico de la base de datos implementada en la herramienta CASE ER-Studio.

Para la carga de datos se utilizó la aplicación Data Generator de la Empresa Red Gate.





Diccionario de Datos

Nombre Campo	Descripción	Entidad/es a la que pertenece	PK,FK	Dominio/ Longitud	Acepta nulos
aceptaCamaExtra	Indica si la Habitación acepta cama extra. aceptaCamaExtra=[SI NO]	Habitacion		varchar(3)	NO
altura	Altura de la Dirección	Direccion		smallint	NO
apellido	Apellido/s del Turista	Turista		varchar(60)	NO
Calle	Calle de la Dirección	Direccion		varchar(100)	NO
cantidad	Cantidad de productos comprados	LineaGastoMinibar		int	NO
cantidad_dias	Cantidad de días efectivos de la Estancia	linea_habitacion		int	NO
cantidad_serv	Cantidad de veces consumido el Servicio Extra	LineaGastoServiciosExtr		int	NO
caracteristica	Característica de la zona de la Llamada	Llamada		varchar(10)	NO
checkin	Fecha y hora de ingreso a la Estancia	Estancia		datetime	NO
checkout	Fecha y hora de egreso de la Estancia	Estancia		datetime	NO
Cod_interno	Identificación del Teléfono del Hotel desde el cual se realizó la Llamada	Llamada - TelefonoHotel	FK	varchar(10)	NO
Cod_interno	Identificación del Teléfono del Hotel	TelefonoHotel	PK	varchar(10)	NO
codPostal	Código Postal de la Localidad	Localidad		varchar(15)	NO
configuracionPlazas	Cantidad de Plazas del Tipo de Habitación	TipoHabitacion		varchar(10)	NO
contraseña	Contraeña de la Cuenta del Usuario	CuentaUsuario		varchar(50)	NO
correo	Dirección de correo electrónico del Usuario	Usuario		varchar(100)	NO
correo_alternativo	Dirección de correo electrónico alternativo del Usuario	Usuario		varchar(100)	NO
CostoExtra	Total de Gastos de Servicios Extra	FacturaHabitacion		money	NO
CostoHabitaciones	Costo de la Habitacion	FacturaHabitacion		money	NO
CostoTotalLlamada	Total de Gastos de las Llamadas	FacturaHabitacion		money	NO
CostoTotalMinibar	Total de Gastos del Minibar	FacturaHabitacion		money	NO
cuil	Cuil de la Empresa	Empresa		varchar(40)	NO
datos_adicionales	Datos adicionales del Cliente	Cliente		XML	SI
Departamento	Identificación del Departamento de la Dirección	Direccion		varchar(10)	SI
descripcion	Descripción de la Categoría de la Habitación	Categoria		varchar(20)	NO
descripcion	Descripción del Producto/Consumible	Consumibles		varchar(200)	SI
descriptions	Descripción del Servicio	Servicio		varchar(200)	NO
desde	Fecha y hora de inicio de la Reserva	reservaHabitacion		datetime	NO
dni	Documento de identidad del Turista	Turista		int	NO
duracion	Duración de la Llamada	Llamada		time(7)	NO
Edificio	Nombre del Edificio de la Dirección	Direccion		varchar(50)	SI
estado	Estado de la Habitación. estado = [LIB OCU RES]	Habitacion		varchar(3)	NO
estado_contraseña	Estado de la contraseña de la Cuenta del Usuario. estado_contraseña=[A V C] (Activa, Vencida, a Confirmar)	CuentaUsuario		varchar(3)	NO
estado_cuenta	Estado de la Cuenta del Usuario. estado_cuenta=[act ven con des]	Usuario		char(3)	NO
estado_res	Estado de la Reserva. estado_res=[pendiente cancelado En_Uso incumplido]	reservaHabitacion		varchar(10)	NO



estadoFactH	Estado de la Factura. estadoFactH= [Paga Impaga]	FacturaHabitacion		varchar(3)	NO
estrellas	Estrellas del Hotel	Hotel		smallint	NO
fecha_constatacion_m	Fecha de constatación de la Línea de Gasto de Minibar	LineaGastoMinibar		datetime	NO
fecha_solicitud_r	Fecha de solicitud de la Reserva	reserva		datetime	NO
fecha_uso_s	Fecha de registro de la Línea de Gasto de Servicios Extra	LineaGastoServiciosExtra		datetime	NO
fechaLlamada	Fecha en que se realizó la Llamada	Llamada		datetime	NO
fin_temporada	Fecha de fin de la Temporada	temporada		datetime	NO
hasta	Fecha y hora de fin de la Reserva	reservaHabitacion		datetime	NO
id:tipoLlamada	Identificación de Tipo de llamada al cual pertenece la Llamada	Llamada - TipoLlamada	FK	varchar(18)	NO
id:tipoLlamada	Identificación del Tipo de Llamada	TipoLlamada	PK	varchar(18)	NO
id_cadena_hotel	Identificación de la Cadena de Hotel	CadenaHotel	PK	int	NO
id_cadena_hotel	Identificación de la Cadena de Hotel a la que pertenece el Hotel	Hotel - CadenaHotel	FK	int	NO
id_categoria	Identificación de la Categoría de la Habitación	Categoria	PK	varchar(20)	NO
id_categoria	Identificación de la Cartegoría de la Habitación	Habitacion - Categoria	FK	varchar(20)	NO
id_categoria	Identificación de la Categoría de la Habitación	servicioCategoria - Categoria	PK - FK	varchar(20)	NO
id_direccion	Dirección de la Cadena Hotelera	CadenaHotel - Direccion	FK	int	NO
id_direccion	Identificación de la Dirección	Direccion	PK	int	NO
id_direccion	Identificación de la Dirección del Hotel	Hotel - Direccion	FK	int	NO
id_direccion	Identificación de la Dirección del Usuario	Usuario - Direccion	FK	int	NO
id_estancia	Identificación de la Estancia	Estancia	PK	int	NO
id_estancia	Identificación de la Estancia a la que pertenece la Factura	FacturaHabitacion - Estancia	FK	int	NO
id_estancia	Identificación de Estancia en la que se alojó el turista	TuristaPorEstancia - Estancia	FK	int	NO
id_factura	Identificación de la Factura a la que pertenece el renglón	Detalle_factura - FacturaHabitacion	PK - FK	int	NO
id_factura	Identificación de la Factura	FacturaHabitacion	PK	int	NO
id_factura	Identificación de la Factura a la que pertenece la Línea de gasto de Habitación	linea_habitacion - Detalle_factura	FK	int	NO
id_factura	Identificación de la Factura a la cual pertenece la Línea de Gasto de Llamada	LineaGastoLlamada - Detalle_factura	FK	int	NO
id_factura	Identificación de la Factura a la cual pertenece la Línea de Gasto de Minibar	LineaGastoMinibar - Detalle_factura	FK	int	NO
id_factura	Identificación de la Factura a la cual pertenece la Línea de Gasto de Servicios Extra	LineaGastoServiciosExtra - Detalle_factura	FK	int	NO
id_habitacion	Identificación de la Habitación de la Estancia	Estancia - reservaHabitacion	FK	char(10)	NO
id_habitacion	Identificación de la Habitación	Habitacion	PK	char(10)	NO
id_habitacion	Identificación de la Habitación a la que pertenece la Línea de gasto de Habitación	linea_habitacion - reservaHabitacion	PK - FK	char(10)	NO
id_habitacion	Identificación de la Habitación a la que pertenece la Reserva de la Habitación	reservaHabitacion - Habitacion	PK - FK	char(10)	NO
id_habitacion	Identificación de la Habitación en la Cual se	TelefonoHotel -	FK	char(10)	NO



	encuentra el Teléfono del Hotel	Habitacion			
id_hotel	Identificación del Hotel de la Habitación	Habitacion - Hotel	FK	int	NO
id_hotel	Identificación del Hotel	Hotel	PK	int	NO
id_hotel	Identificación del Hotel al cual pertenece el Servicio	Servicio - Hotel	FK	int	NO
id_hotel	Identificación del Hotel a la que pertenece la Temporada	temporada - Hotel	FK	int	NO
id_lineaMinibar	Identificación de la Línea de Gasto de Minibar	LineaGastoMinibar	PK	int	NO
id_LineaServicio	Identificación de la Linea de Gasto de Servicio Extra	LineaGastoServiciosExtra	PK	char(10)	NO
id_LineaServicio	Identificación del Servicio al cual pertenece esa Preferencia	preferencia - LineaGastoServiciosExtra	FK	char(10)	NO
id_llamada	Identificación de la llamada a la que pertenece la Línea de Gasto de Llamada	LineaGastoLlamada - Llamada	FK	int	NO
id_llamada	Identificación de la Llamada	Llamada	PK	int	NO
id_localidad	Identificación de la Localidad de la Dirección	Direccion - Localidad	FK	int	NO
id_localidad	Identificación de la Localidad	Localidad	PK	int	NO
id_localidad	Identificación de la Localidad a la que pertenece el Tipo de Llamada	TipoLlamada - Localidad	FK	int	NO
id_login	Identificación del Login del Usuario	CuentaUsuario	PK	int	NO
id_PenalizacionDeTemporada	Identificación de la Penalización de la Línea de gasto de Habitación	linea_habitacion - PenalizacionReserva	FK	smallint	NO
id_PenalizacionDeTemporada	Identificación de la Penalización de la Temporada	PenalizacionReserva	PK	smallint	NO
id_preferencia	Identificación de la Preferencia	preferencia	PK	int	NO
id_producto	Identificación del Producto/Consumible	Consumibles	PK	int	NO
id_producto	Identificación del Producto al que pertenece la Línea de Gasto de Minibar	LineaGastoMinibar - servicioMinibar	FK	int	NO
id_producto	Identificación del Consumible que contiene el Minibar	servicioMinibar - Consumibles	PK - FK	int	NO
id_reserva	Identificación de la Reserva a la que pertenece la Estancia	Estancia - reservaHabitacion	FK	int	NO
id_reserva	Identificación de la Reserva a la que pertenece la Línea de Habitación	linea_habitacion - reservaHabitacion	PK - FK	int	NO
id_reserva	Identificación de la Reserva	reserva	PK	int	NO
id_reserva	Identificación de la Reserva a la que pertenece la Reserva de la Habitación	reservaHabitacion - reserva	PK - FK	int	NO
id_servicio	Identificación del Servicio al que pertenece la Línea de Gasto de Minibar	LineaGastoMinibar - servicioMinibar	FK	smallint	NO
id_servicio	Identificación del Servicio al que pertenece la Línea de Gasto de Servicios Extra	LineaGastoServiciosExtra - Servicio	FK	smallint	NO
id_servicio	Identificación del Servicio	Servicio	PK	smallint	NO
id_servicio	Identificación del Servicio que Incluye la Categoría de la Habitación	servicioCategoria - Servicio	PK - FK	smallint	NO
id_servicio	Identificación del Servicio de Minibar	servicioMinibar - Servicio	PK - FK	smallint	NO
id_temporada	Identificación de la Temporada a la que pertenece la Penalización	PenalizacionReserva - temporada	FK	smallint	NO
id_temporada	Identificación de la Temporada	temporada	PK	smallint	NO
id_TipoH	Identificación del Tipo de Habitación al que	Habitacion -	FK	smallint	NO



	pertenece la Habitación	TipoHabitacion			
id_TipoH	Identificación del Tipo de Habitación	TipoHabitacion	PK	smallint	NO
id_turistas_x_estancia	Identificación de Turista por Estancia	TuristaPorEstancia	PK	int	NO
id_usuario	Identificación del Usuario del Cliente	Cliente - Usuario	PK - FK	int	NO
id_usuario	Identificación del Usuario	CuentaUsuario - Usuario	PK - FK	int	NO
id_usuario	Identificación del Usuario de la Empresa	Empresa - Usuario	PK - FK	int	NO
id_usuario	Identificación del Usuario al cual pertenece esa Preferencia	preferencia - Cliente	FK	int	NO
id_usuario	Identificación del Usuario que realiza la reserva	reserva - Cliente	FK	int	NO
id_usuario	Identificación del Usuario del Turista	Turista - Usuario	PK - FK	int	NO
id_usuario	Identificación del Usuario del Turista	TuristaPorEstancia - Usuario	FK	int	NO
id_usuario	Identificación del Usuario	Usuario	PK	int	NO
idLineaLlamada	Identificación de la Línea de gasto de Llamada	LineaGastoLlamada	PK	char(10)	NO
inicio_temporada	Fecha de inicio de la Temporada	temporada		datetime	NO
lengua_oficial	Lengua/s oficial/es del País	Pais		varchar(50)	SI
Moneda	Moneda oficial del País	Pais		varchar(50)	SI
nombre	Nombre de la Cadena Hotelera	CadenaHotel		varchar(50)	NO
nombre	Nombre de la Temporada	temporada		varchar(25)	NO
nombre	Nombre del Tipo de Llamada	TipoLlamada		varchar(10)	NO
nombre	Nombre/s del Turista	Turista		varchar(60)	NO
nombre_Localidad	Nombre de la Localidad	Localidad		varchar(100)	NO
nombre_pais	Nombre del País de la Localidad	Localidad - Pais	FK	varchar(100)	NO
nombre_pais	Nombre del País	Pais	PK	varchar(100)	NO
nombre_sucursal	Nombre del Hotel o Sucursal	Hotel		varchar(50)	NO
nombre_usuario	Nombre de la Cuenta del Usuario	CuentaUsuario		varchar(50)	NO
nombreC	Nombre de la Categoría de la Habitación	Categoría		varchar(50)	NO
nombreE	Nombre de la Empresa	Empresa		varchar(80)	NO
nombreP	Nombre del Producto/Consumible	Consumibles		varchar(50)	NO
nombreS	Nombre del Servicio	Servicio		varchar(100)	NO
nombreT	Nombre del Tipo de Habitación	TipoHabitacion		varchar(50)	NO
numero_renglon	Número de renglón de la Factura	Detalle_factura	PK	int	NO
numero_renglon	Número de renglón de la Factura a la cual pertenece la Línea de gasto de Habitación	linea_habitacion - Detalle_factura	FK	int	NO
numero_renglon	Identificación del número de renglón al cual pertenece la Línea de Gasto de Llamada	LineaGastoLlamada - Detalle_factura	FK	int	NO
numero_renglon	Identificación del número de renglón al cual pertenece la Línea de Gasto de Minibar	LineaGastoMinibar - Detalle_factura	FK	int	NO
numero_renglon	Identificación del número de renglón al cual pertenece la Línea de Gasto de Servicios Extra	LineaGastoServiciosExtra - Detalle_factura	FK	int	NO
numeroTel	Número del Teléfono del Hotel	TelefonoHotel		varchar(30)	NO
observaciones	Observaciones del Cliente	Cliente		varchar(300)	SI
observaciones	Observaciones de la Penalización	PenalizacionReserva		varchar(100)	NO
penalizacion	Monto congelado de penalización sobre la habitación	linea_habitacion		money	NO
piso	Piso del Edificio de la Dirección	Direccion		smallint	SI



porcentaje	Porcentaje de la Penalización. Varía entre 0 y 2. 0,1 se lo ve como un Descuento del 10%. 1,1 se lo ve como un aumento del 10%	PenalizacionReserva		decimal(3,2)	NO
porcentajeRebajaRecargo	Porcentaje de Rebaja/Recargo de la Temporada	temporada		decimal(3,2)	NO
precio	Precio del Producto/Consumible	Consumibles		money	NO
precio	Precio congelado de la Línea de Gasto de Minibar	LineaGastoMinibar		money	NO
precio	Precio del Tipo de Llamada	TipoLlamada		money	NO
precio_llamada	Precio congelado de la llamada perteneciente a la Linea de Gasto de Llamada	LineaGastoLlamada		money	NO
precio_serv_extra	Precio congelado de la Línea de Gasto de Servicios Extra	LineaGastoServiciosExtr		money	NO
precio_unitario	Precio unitario de la Habitación	linea_habitacion		money	NO
precioS	Precio del Servicio	Servicio		money	NO
PrecioTh	Precio del Tipo de Habitación	TipoHabitacion		money	NO
razon_social	Razon social de la Empresa	Empresa		varchar(60)	NO
telefono	Teléfono de la Cadena Hotelera	CadenaHotel		varchar(25)	NO
telefono_fijo	Teléfono fijo del Usuario	Usuario		varchar(30)	NO
telefono_movil	Teléfono móvil del Usuario	Usuario		varchar(30)	NO
TiempoLimiteReserva	Tiempo de espera antes de que sea cancelada la Reserva	Hotel		time(7)	NO
tif	Imagen de la Cuenta del Usuario	CuentaUsuario		image	SI
tipo_servicio	Tipo de Servicio del Servicio	Servicio		varchar(50)	NO
TotalGeneral	Total General de la Factura. TotalGeneral= CostoExtra + CostoTotalMinibar + CostoTotalLlamada + CostoHabitaciones	FacturaHabitacion		money	NO

Script de Creación de la Base de Datos:

```

USE BD_TPI;
GO
/*
 * ER/Studio 8.0 SQL Code Generation
 * Project :      IntegradorBDA-v1.02.dml
 *
 * Date Created : Wednesday, December 02, 2015
16:35:41
 * Target DBMS : Microsoft SQL Server 2008
 */

/*
 * TABLE: CadenaHotel
 */

CREATE TABLE CadenaHotel (
    id_cadena_hotel    int          NOT NULL,
    telefono            varchar(25)  NOT NULL,
    nombre               varchar(50)  NOT NULL,
    id_direccion        int          NOT NULL,
    CONSTRAINT PK72 PRIMARY KEY NONCLUSTERED
(id_cadena_hotel)
)
go

IF OBJECT_ID('CadenaHotel') IS NOT NULL
    PRINT '<<< CREATED TABLE CadenaHotel >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE CadenaHotel
>>>'

go

/*
 * TABLE: Categoria
 */

CREATE TABLE Categoria(
    id_categoria        varchar(20)  NOT NULL,
    nombreC              varchar(50)  NOT NULL,
    descripcion          varchar(20)  NOT NULL,
    CONSTRAINT PK38 PRIMARY KEY NONCLUSTERED
(id_categoria)
)
go

IF OBJECT_ID('Categoria') IS NOT NULL

```



```
PRINT '<<< CREATED TABLE Categoria >>>'  
ELSE  
    PRINT '<<< FAILED CREATING TABLE Categoria  
>>>'  
GO  
  
/*  
 * TABLE: Cliente  
 */  
  
CREATE TABLE Cliente(  
    datos_adicionales xml NULL,  
    observaciones varchar(300) NULL,  
    id_usuario int NOT NULL,  
    CONSTRAINT PK6 PRIMARY KEY NONCLUSTERED  
(id_usuario)  
)  
GO  
  
IF OBJECT_ID('Cliente') IS NOT NULL  
    PRINT '<<< CREATED TABLE Cliente >>>'  
ELSE  
    PRINT '<<< FAILED CREATING TABLE Cliente  
>>>'  
GO  
  
/*  
 * TABLE: Consumibles  
 */  
  
CREATE TABLE Consumibles(  
    id_producto int NOT NULL,  
    descripcion varchar(200) NULL,  
    nombreP varchar(50) NOT NULL,  
    precio money NOT NULL,  
    CONSTRAINT PK67 PRIMARY KEY NONCLUSTERED  
(id_producto)  
)  
GO  
  
IF OBJECT_ID('Consumibles') IS NOT NULL  
    PRINT '<<< CREATED TABLE Consumibles >>>'  
ELSE  
    PRINT '<<< FAILED CREATING TABLE Consumibles  
>>>'  
GO  
  
/*  
 * TABLE: CuentaUsuario  
 */  
  
CREATE TABLE CuentaUsuario(  
    id_login int NOT NULL,  
    id_usuario int NOT NULL,  
    tif image NULL,  
    estado_contraseña varchar(3) NOT NULL,  
    contraseña varchar(50) NOT NULL,  
    nombre_usuario varchar(50) NOT NULL,  
    CONSTRAINT PK2 PRIMARY KEY NONCLUSTERED  
(id_login, id_usuario)  
)  
GO  
  
IF OBJECT_ID('CuentaUsuario') IS NOT NULL  
    PRINT '<<< CREATED TABLE CuentaUsuario >>>'  
ELSE
```

```
    PRINT '<<< FAILED CREATING TABLE  
CuentaUsuario >>>'  
GO  
  
/*  
 * TABLE: Detalle_factura  
 */  
  
CREATE TABLE Detalle_factura(  
    numero_renglon int NOT NULL,  
    id_factura int NOT NULL,  
    CONSTRAINT PK75 PRIMARY KEY NONCLUSTERED  
(numero_renglon, id_factura)  
)  
GO  
  
IF OBJECT_ID('Detalle_factura') IS NOT NULL  
    PRINT '<<< CREATED TABLE Detalle_factura  
>>>'  
ELSE  
    PRINT '<<< FAILED CREATING TABLE  
Detalle_factura >>>'  
GO  
  
/*  
 * TABLE: Direccion  
 */  
  
CREATE TABLE Direccion(  
    id_direccion int NOT NULL,  
    Calle varchar(100) NOT NULL,  
    altura smallint NOT NULL,  
    Edificio varchar(50) NULL,  
    piso smallint NULL,  
    Departamento varchar(10) NULL,  
    id_localidad int NOT NULL,  
    CONSTRAINT PK3 PRIMARY KEY NONCLUSTERED  
(id_direccion)  
)  
GO  
  
IF OBJECT_ID('Direccion') IS NOT NULL  
    PRINT '<<< CREATED TABLE Direccion >>>'  
ELSE  
    PRINT '<<< FAILED CREATING TABLE Direccion  
>>>'  
GO  
  
/*  
 * TABLE: Empresa  
 */  
  
CREATE TABLE Empresa(  
    razon_social varchar(60) NOT NULL,  
    cuil varchar(40) NOT NULL,  
    nombreE varchar(80) NOT NULL,  
    id_usuario int NOT NULL,  
    CONSTRAINT PK74 PRIMARY KEY NONCLUSTERED  
(id_usuario)  
)  
GO  
  
IF OBJECT_ID('Empresa') IS NOT NULL  
    PRINT '<<< CREATED TABLE Empresa >>>'  
ELSE
```



```
PRINT '<<< FAILED CREATING TABLE Empresa
>>>
go

/*
 * TABLE: Estancia
 */

CREATE TABLE Estancia(
    checkin      datetime      NOT NULL,
    id_estancia  int          NOT NULL,
    checkout     datetime      NOT NULL,
    id_reserva   int          NOT NULL,
    id_habitacion char(10)    NOT NULL,
    CONSTRAINT PK19 PRIMARY KEY NONCLUSTERED
(id_estancia)
)
go

IF OBJECT_ID('Estancia') IS NOT NULL
    PRINT '<<< CREATED TABLE Estancia >>>
ELSE
    PRINT '<<< FAILED CREATING TABLE Estancia
>>>
go

/*
 * TABLE: FacturaHabitacion
 */

CREATE TABLE FacturaHabitacion(
    id_factura   int          NOT NULL,
    estadoFactH  varchar(3)   NOT NULL,
    CostoExtra   money        NOT NULL,
    TotalGeneral money        NOT NULL,
    CostoTotalMinibar money        NOT NULL,
    CostoTotalLLamada money        NOT NULL,
    CostoHabitaciones money        NOT NULL,
    id_estancia  int          NOT NULL,
    CONSTRAINT PK26 PRIMARY KEY NONCLUSTERED
(id_factura)
)
go

IF OBJECT_ID('FacturaHabitacion') IS NOT NULL
    PRINT '<<< CREATED TABLE FacturaHabitacion
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
FacturaHabitacion >>>
go

/*
 * TABLE: Habitacion
 */

CREATE TABLE Habitacion(
    id_habitacion char(10)    NOT NULL,
    aceptaCamaExtra varchar(3) NOT NULL,
    estado         varchar(3)  NOT NULL,
    id_hotel       int          NOT NULL,
    id_categoria  varchar(20) NOT NULL,
    id_TipoH       smallint    NOT NULL,
    CONSTRAINT PK20 PRIMARY KEY NONCLUSTERED
(id_habitacion)
)
```

```
go

IF OBJECT_ID('Habitacion') IS NOT NULL
    PRINT '<<< CREATED TABLE Habitacion >>>
ELSE
    PRINT '<<< FAILED CREATING TABLE Habitacion
>>>
go

/*
 * TABLE: Hotel
 */

CREATE TABLE Hotel(
    id_hotel      int          NOT NULL,
    TiempoLimiteReserva time(7)    NOT NULL,
    nombre_sucursal varchar(50) NOT NULL,
    estrellas      smallint
IDENTITY(1,1),
    id_cadena_hotel int          NOT NULL,
    id_direccion   int          NOT NULL,
    CONSTRAINT PK8 PRIMARY KEY NONCLUSTERED
(id_hotel)
)
go

IF OBJECT_ID('Hotel') IS NOT NULL
    PRINT '<<< CREATED TABLE Hotel >>>
ELSE
    PRINT '<<< FAILED CREATING TABLE Hotel >>>
go

/*
 * TABLE: linea_habitacion
 */

CREATE TABLE linea_habitacion(
    id_reserva     int          NOT NULL,
    cantidad_dias int          NOT NULL,
    precio_unitario money        NOT NULL,
    penalizacion  money        NOT NULL,
    id_habitacion  char(10)    NOT NULL,
    numero_renglon int          NOT NULL,
    id_factura    int          NOT NULL,
    id_PenalizacionDeTemp smallint NOT NULL
)
go

IF OBJECT_ID('linea_habitacion') IS NOT NULL
    PRINT '<<< CREATED TABLE linea_habitacion
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
linea_habitacion >>>
go

/*
 * TABLE: LineaGastoLlamada
 */

CREATE TABLE LineaGastoLlamada(
    idLineaLLamada char(10)    NOT NULL,
    precio_llamada  money        NOT NULL,
    id_llamada      int          NOT NULL,
    numero_renglon int          NOT NULL,
    id_factura      int          NOT NULL,
)
```



```
        CONSTRAINT PK57 PRIMARY KEY NONCLUSTERED
        (idLinealLlamada)
    )
go

IF OBJECT_ID('LineaGastoLlamada') IS NOT NULL
    PRINT '<<< CREATED TABLE LineaGastoLlamada'
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
LineaGastoLlamada >>>'
go

/*
 * TABLE: LineaGastoMinibar
 */

CREATE TABLE LineaGastoMinibar(
    precio                      money      NOT NULL,
    cantidad                     int       NOT NULL,
    fecha_constatacion_m        datetime   NOT NULL,
    id_lineaMinibar             int       NOT NULL,
    id_servicio                  smallint   NOT NULL,
    id_producto                 int       NOT NULL,
    numero_renglon              int       NOT NULL,
    id_factura                  int       NOT NULL,
    CONSTRAINT PK70 PRIMARY KEY NONCLUSTERED
    (id_lineaMinibar)
)
go

IF OBJECT_ID('LineaGastoMinibar') IS NOT NULL
    PRINT '<<< CREATED TABLE LineaGastoMinibar'
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
LineaGastoMinibar >>>'
go

/*
 * TABLE: LineaGastoServiciosExtra
 */

CREATE TABLE LineaGastoServiciosExtra(
    id_LineaSevicio            char(10)  NOT NULL,
    cantidad_serv                int       NOT NULL,
    precio_serv_extra           money     NOT NULL,
    fecha_uso_s                 datetime  NOT NULL,
    id_servicio                 smallint  NOT NULL,
    numero_renglon              int       NOT NULL,
    id_factura                  int       NOT NULL,
    CONSTRAINT PK59 PRIMARY KEY NONCLUSTERED
    (id_LineaSevicio)
)
go

IF OBJECT_ID('LineaGastoServiciosExtra') IS NOT
NULL
    PRINT '<<< CREATED TABLE
LineaGastoServiciosExtra >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE
LineaGastoServiciosExtra >>>'
go

/*
```

```
 * TABLE: Llamada
 */

CREATE TABLE Llamada(
    id_llamada                  int       NOT NULL,
    caracteristica              varchar(10) NOT NULL,
    fechaLlamada                datetime  NOT NULL,
    duracion                     time(7)   NOT NULL,
    Cod_interno                 varchar(10) NOT NULL,
    [id:tipoLlamada]            varchar(18) NOT NULL,
    CONSTRAINT PK33 PRIMARY KEY NONCLUSTERED
    (id_llamada)
)
go

IF OBJECT_ID('Llamada') IS NOT NULL
    PRINT '<<< CREATED TABLE Llamada >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE Llamada
>>>'
go

/*
 * TABLE: Localidad
 */

CREATE TABLE Localidad(
    id_localidad                int       NOT NULL,
    codPostal                   varchar(15) NOT NULL,
    nombre_Localidad            varchar(100) NOT NULL,
    nombre_pais                 varchar(100) NOT NULL,
    CONSTRAINT PK4 PRIMARY KEY NONCLUSTERED
    (id_localidad)
)
go

IF OBJECT_ID('Localidad') IS NOT NULL
    PRINT '<<< CREATED TABLE Localidad >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE Localidad
>>>'
go

/*
 * TABLE: Pais
 */

CREATE TABLE Pais(
    nombre_pais                 varchar(100) NOT NULL,
    lengua_oficial              varchar(50)  NULL,
    Moneda                      varchar(50)  NULL,
    CONSTRAINT PK5 PRIMARY KEY NONCLUSTERED
    (nombre_pais)
)
go

IF OBJECT_ID('Pais') IS NOT NULL
    PRINT '<<< CREATED TABLE Pais >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE Pais >>>'
go

/*
 * TABLE: PenalizacionReserva
 */

/*
```



```
CREATE TABLE PenalizacionReserva(
    id_PenalizacionDeTemp     smallint NOT NULL,
    observaciones             varchar(100) NULL,
    porcentaje                decimal(3, 2) NOT NULL,
    id_temporada              smallint NOT NULL,
    CONSTRAINT PK36 PRIMARY KEY NONCLUSTERED
    (id_PenalizacionDeTemp)
)
go

IF OBJECT_ID('PenalizacionReserva') IS NOT NULL
    PRINT '<<< CREATED TABLE PenalizacionReserva'
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
PenalizacionReserva >>>'
go

/*
 * TABLE: preferencia
 */

CREATE TABLE preferencia(
    id_LineaSevicio    char(10)      NOT NULL,
    id_usuario          int           NOT NULL,
    CONSTRAINT PK30 PRIMARY KEY NONCLUSTERED
    (id_usuario)
)
go

IF OBJECT_ID('preferencia') IS NOT NULL
    PRINT '<<< CREATED TABLE preferencia >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE preferencia
>>>'
go

/*
 * TABLE: reserva
 */

CREATE TABLE reserva(
    id_reserva          int           NOT NULL,
    fecha_solicitud_r   datetime     NOT NULL,
    precio_venta_reserva money        NOT NULL,
    id_usuario          int           NOT NULL,
    CONSTRAINT PK18 PRIMARY KEY NONCLUSTERED
    (id_reserva)
)
go

IF OBJECT_ID('reserva') IS NOT NULL
    PRINT '<<< CREATED TABLE reserva >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE reserva
>>>'
go

/*
 * TABLE: [reserva Habitacion]
 */

CREATE TABLE [reserva Habitacion] (
    id_reserva          int           NOT NULL,
    estado_res          varchar(10)   NOT NULL,
    desde               datetime     NOT NULL,
```

```
hasta                 datetime     NOT NULL,
    id_habitacion       char(10)     NOT NULL,
    CONSTRAINT PK55 PRIMARY KEY NONCLUSTERED
    (id_reserva, id_habitacion)
)
go

IF OBJECT_ID('reserva Habitacion') IS NOT NULL
    PRINT '<<< CREATED TABLE reserva Habitacion
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE reserva
Habitacion >>>'
go

/*
 * TABLE: Servicio
 */

CREATE TABLE Servicio(
    id_servicio         smallint
    IDENTITY(1,1),
    nombreS              varchar(100)  NOT NULL,
    descripcionS        varchar(200)  NOT NULL,
    precioS              money        NOT NULL,
    id_hotel             int           NOT NULL,
    CONSTRAINT PK37 PRIMARY KEY NONCLUSTERED
    (id_servicio)
)
go

IF OBJECT_ID('Servicio') IS NOT NULL
    PRINT '<<< CREATED TABLE Servicio >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE Servicio
>>>'
go

/*
 * TABLE: servicioCategoria
 */

CREATE TABLE servicioCategoria(
    id_categoria        varchar(20)  NOT NULL,
    id_servicio          smallint
    NOT NULL,
    CONSTRAINT PK64 PRIMARY KEY NONCLUSTERED
    (id_categoria, id_servicio)
)
go

IF OBJECT_ID('servicioCategoria') IS NOT NULL
    PRINT '<<< CREATED TABLE servicioCategoria
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
servicioCategoria >>>'
go

/*
 * TABLE: servicioMinibar
 */

CREATE TABLE servicioMinibar(
    id_servicio         smallint
    NOT NULL,
    id_producto         int           NOT NULL,
```



```
CONSTRAINT PK66 PRIMARY KEY NONCLUSTERED
(id_servicio, id_producto)
)
go

IF OBJECT_ID('servicioMinibar') IS NOT NULL
    PRINT '<<< CREATED TABLE servicioMinibar'
>>>
ELSE
    PRINT '<<< FAILED CREATING TABLE
servicioMinibar >>>'
go

/*
 * TABLE: TelefonoHotel
 */

CREATE TABLE TelefonoHotel(
    Cod_interno      varchar(10)      NOT NULL,
    numeroTel        varchar(30)      NOT NULL,
    id_habitacion    char(10)        NOT NULL,
    CONSTRAINT PK32 PRIMARY KEY NONCLUSTERED
(Cod_interno)
)
go

IF OBJECT_ID('TelefonoHotel') IS NOT NULL
    PRINT '<<< CREATED TABLE TelefonoHotel >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE
TelefonoHotel >>>'
go

/*
 * TABLE: temporada
 */

CREATE TABLE temporada(
    id_temporada     smallint        NOT NULL,
    nombre            varchar(25)     NOT NULL,
    inicio_temporada datetime        NOT NULL,
    fin_temporada    time(7)        NOT NULL,
    porcentajeRebaja decimal(3, 2)
NULL,
    id_hotel          int            NOT NULL,
    CONSTRAINT PK35 PRIMARY KEY NONCLUSTERED
(id_temporada)
)
go

IF OBJECT_ID('temporada') IS NOT NULL
    PRINT '<<< CREATED TABLE temporada >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE temporada
>>>'
go

/*
 * TABLE: TipoHabitacion
 */

CREATE TABLE TipoHabitacion(
    id_TipoH           smallint        NOT NULL,
    configuracionPlazas varchar(10)     NOT NULL,
    nombreT             varchar(50)     NOT NULL,
    PrecioTh            money          NOT NULL,
```

```
CONSTRAINT PK44 PRIMARY KEY NONCLUSTERED
(id_TipoH)
)
go

IF OBJECT_ID('TipoHabitacion') IS NOT NULL
    PRINT '<<< CREATED TABLE TipoHabitacion >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE
TipoHabitacion >>>'
go

/*
 * TABLE: TipoLlamada
 */

CREATE TABLE TipoLlamada (
    [id:tipoLlamada]  varchar(18)     NOT NULL,
    caracteristica   varchar(20)     NOT NULL,
    nombre            varchar(10)     NULL,
    precio            money          NOT NULL,
    id_localidad     int            NOT NULL,
    CONSTRAINT PK34 PRIMARY KEY NONCLUSTERED
([id:tipoLlamada])
)
go

IF OBJECT_ID('TipoLlamada') IS NOT NULL
    PRINT '<<< CREATED TABLE TipoLlamada >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE TipoLlamada
>>>'
go

/*
 * TABLE: Turista
 */

CREATE TABLE Turista(
    dni               int            NOT NULL,
    nombre            varchar(60)     NOT NULL,
    apellido          varchar(60)     NOT NULL,
    id_usuario        int            NOT NULL,
    CONSTRAINT PK73 PRIMARY KEY NONCLUSTERED
(id_usuario)
)
go

IF OBJECT_ID('Turista') IS NOT NULL
    PRINT '<<< CREATED TABLE Turista >>>'
ELSE
    PRINT '<<< FAILED CREATING TABLE Turista
>>>'
go

/*
 * TABLE: TuristaPorEstancia
 */

CREATE TABLE TuristaPorEstancia(
    id_usuario          int            NOT NULL,
    id_turistas_x_estancia int          NOT NULL,
    id_estancia         int            NOT NULL,
    CONSTRAINT PK77 PRIMARY KEY NONCLUSTERED
(id_turistas_x_estancia)
)
```



```
go

IF OBJECT_ID('TuristaPorEstancia') IS NOT NULL
    PRINT '<<< CREATED TABLE TuristaPorEstancia
>>>'  

ELSE
    PRINT '<<< FAILED CREATING TABLE
TuristaPorEstancia >>>'  

go

/*
 * TABLE: Usuario
 */

CREATE TABLE Usuario(
    id_usuario          int      NOT NULL,
    correo              varchar(100) NOT NULL,
    correo_alternativo varchar(100)  NULL,
    telefono_fijo       varchar(30)   NULL,
    telefono_movil     varchar(30)  NOT NULL,
    estado_cuenta       char(3)    NOT NULL,
    id_direccion        int      NOT NULL,
    CONSTRAINT PK11 PRIMARY KEY NONCLUSTERED
(id_usuario)
)
go

IF OBJECT_ID('Usuario') IS NOT NULL
    PRINT '<<< CREATED TABLE Usuario >>>'  

ELSE
    PRINT '<<< FAILED CREATING TABLE Usuario
>>>'  

go

/*
 * TABLE: CadenaHotel
 */

ALTER TABLE CadenaHotel ADD CONSTRAINT
RefDireccion214
    FOREIGN KEY (id_direccion)
    REFERENCES Direccion(id_direccion)
go

/*
 * TABLE: Cliente
 */

ALTER TABLE Cliente ADD CONSTRAINT RefUsuario236
    FOREIGN KEY (id_usuario)
    REFERENCES Usuario(id_usuario)
go

/*
 * TABLE: CuentaUsuario
 */

ALTER TABLE CuentaUsuario ADD CONSTRAINT
RefUsuario14
    FOREIGN KEY (id_usuario)
    REFERENCES Usuario(id_usuario)
go

/*
 * TABLE: Detalle_factura
 */

ALTER TABLE Detalle_factura ADD CONSTRAINT
RefFacturaHabitacion242
    FOREIGN KEY (id_factura)
    REFERENCES FacturaHabitacion(id_factura)
go

/*
 * TABLE: Direccion
 */

ALTER TABLE Direccion ADD CONSTRAINT
RefLocalidad5
    FOREIGN KEY (id_localidad)
    REFERENCES Localidad(id_localidad)
go

/*
 * TABLE: Empresa
 */

ALTER TABLE Empresa ADD CONSTRAINT RefCliente235
    FOREIGN KEY (id_usuario)
    REFERENCES Cliente(id_usuario)
go

/*
 * TABLE: Estancia
 */

ALTER TABLE Estancia ADD CONSTRAINT
Refreserva_Habitacion217
    FOREIGN KEY (id_reserva, id_habitacion)
    REFERENCES [reserva Habitacion](id_reserva,
id_habitacion)
go

/*
 * TABLE: FacturaHabitacion
 */

ALTER TABLE FacturaHabitacion ADD CONSTRAINT
RefEstancia220
    FOREIGN KEY (id_estancia)
    REFERENCES Estancia(id_estancia)
go

/*
 * TABLE: Habitacion
 */

ALTER TABLE Habitacion ADD CONSTRAINT
RefCategoria107
    FOREIGN KEY (id_categoria)
    REFERENCES Categoria(id_categoria)
go

ALTER TABLE Habitacion ADD CONSTRAINT
RefTipoHabitacion128
    FOREIGN KEY (id_TipoH)
```



```
REFERENCES TipoHabitacion(id_TipoH)
go

ALTER TABLE Habitacion ADD CONSTRAINT RefHotel48
    FOREIGN KEY (id_hotel)
        REFERENCES Hotel(id_hotel)
go

/*
 * TABLE: Hotel
 */
ALTER TABLE Hotel ADD CONSTRAINT
RefCadenaHotel211
    FOREIGN KEY (id_cadena_hotel)
        REFERENCES CadenaHotel(id_cadena_hotel)
go

ALTER TABLE Hotel ADD CONSTRAINT RefDireccion216
    FOREIGN KEY (id_direccion)
        REFERENCES Direccion(id_direccion)
go

/*
 * TABLE: linea_habitacion
 */
ALTER TABLE linea_habitacion ADD CONSTRAINT
RefReserva_Habitacion243
    FOREIGN KEY (id_reserva, id_habitacion)
        REFERENCES [reserva Habitacion](id_reserva,
id_habitacion)
go

ALTER TABLE linea_habitacion ADD CONSTRAINT
RefDetalle_factura244
    FOREIGN KEY (numero_renglon, id_factura)
        REFERENCES Detalle_factura(numero_renglon,
id_factura)
go

ALTER TABLE linea_habitacion ADD CONSTRAINT
RefPenalizacionReserva246
    FOREIGN KEY (id_PenalizacionDeTemp)
        REFERENCES PenalizacionReserva(id_PenalizacionDeTemp)
go

/*
 * TABLE: LineaGastoLlamada
 */
ALTER TABLE LineaGastoLlamada ADD CONSTRAINT
RefDetalle_factura241
    FOREIGN KEY (numero_renglon, id_factura)
        REFERENCES Detalle_factura(numero_renglon,
id_factura)
go

ALTER TABLE LineaGastoLlamada ADD CONSTRAINT
RefLlamada164
    FOREIGN KEY (id_llamada)
        REFERENCES Llamada(id_llamada)
go

go

/*
 * TABLE: LineaGastoMinibar
 */
ALTER TABLE LineaGastoMinibar ADD CONSTRAINT
RefDetalle_factura240
    FOREIGN KEY (numero_renglon, id_factura)
        REFERENCES Detalle_factura(numero_renglon,
id_factura)
go

ALTER TABLE LineaGastoMinibar ADD CONSTRAINT
RefServicioMinibar195
    FOREIGN KEY (id_servicio, id_producto)
        REFERENCES servicioMinibar(id_servicio,
id_producto)
go

/*
 * TABLE: LineaGastoServiciosExtra
 */
ALTER TABLE LineaGastoServiciosExtra ADD
CONSTRAINT RefDetalle_factura239
    FOREIGN KEY (numero_renglon, id_factura)
        REFERENCES Detalle_factura(numero_renglon,
id_factura)
go

ALTER TABLE LineaGastoServiciosExtra ADD
CONSTRAINT RefServicio168
    FOREIGN KEY (id_servicio)
        REFERENCES Servicio(id_servicio)
go

/*
 * TABLE: Llamada
 */
ALTER TABLE Llamada ADD CONSTRAINT
RefTelefonoHotel195
    FOREIGN KEY (Cod_interno)
        REFERENCES TelefonoHotel(Cod_interno)
go

ALTER TABLE Llamada ADD CONSTRAINT
RefTipoLlamada96
    FOREIGN KEY ([id:tipoLlamada])
        REFERENCES TipoLlamada([id:tipoLlamada])
go

/*
 * TABLE: Localidad
 */
ALTER TABLE Localidad ADD CONSTRAINT RefPais8
    FOREIGN KEY (nombre_pais)
        REFERENCES Pais(nombre_pais)
go
```



```
/*
 * TABLE: PenalizacionReserva
 */

ALTER TABLE PenalizacionReserva ADD CONSTRAINT
RefTemporada100
    FOREIGN KEY (id_temporada)
        REFERENCES temporada(id_temporada)
go

/*
 * TABLE: preferencia
 */

ALTER TABLE preferencia ADD CONSTRAINT
RefLineaGastoServiciosExtra228
    FOREIGN KEY (id_LineaSevicio)
        REFERENCES LineaGastoServiciosExtra(id_LineaSevicio)
go

ALTER TABLE preferencia ADD CONSTRAINT
RefCliente81
    FOREIGN KEY (id_usuario)
        REFERENCES Cliente(id_usuario)
go

/*
 * TABLE: reserva
 */

ALTER TABLE reserva ADD CONSTRAINT RefCliente43
    FOREIGN KEY (id_usuario)
        REFERENCES Cliente(id_usuario)
go

/*
 * TABLE: [reserva Habitacion]
 */

ALTER TABLE [reserva Habitacion] ADD CONSTRAINT
RefReserva159
    FOREIGN KEY (id_reserva)
        REFERENCES reserva(id_reserva)
go

ALTER TABLE [reserva Habitacion] ADD CONSTRAINT
RefHabitacion160
    FOREIGN KEY (id_habitacion)
        REFERENCES Habitacion(id_habitacion)
go

/*
 * TABLE: Servicio
 */

ALTER TABLE Servicio ADD CONSTRAINT RefHotel110
    FOREIGN KEY (id_hotel)
        REFERENCES Hotel(id_hotel)
go
```

```
/*
 * TABLE: servicioCategoria
 */

ALTER TABLE servicioCategoria ADD CONSTRAINT
RefCategoria185
    FOREIGN KEY (id_categoria)
        REFERENCES Categoria(id_categoria)
go

ALTER TABLE servicioCategoria ADD CONSTRAINT
RefServicio186
    FOREIGN KEY (id_servicio)
        REFERENCES Servicio(id_servicio)
go

/*
 * TABLE: servicioMinibar
 */

ALTER TABLE servicioMinibar ADD CONSTRAINT
RefServicio192
    FOREIGN KEY (id_servicio)
        REFERENCES Servicio(id_servicio)
go

ALTER TABLE servicioMinibar ADD CONSTRAINT
RefConsumibles193
    FOREIGN KEY (id_producto)
        REFERENCES Consumibles(id_producto)
go

/*
 * TABLE: TelefonoHotel
 */

ALTER TABLE TelefonoHotel ADD CONSTRAINT
RefHabitacion94
    FOREIGN KEY (id_habitacion)
        REFERENCES Habitacion(id_habitacion)
go

/*
 * TABLE: temporada
 */

ALTER TABLE temporada ADD CONSTRAINT RefHotel99
    FOREIGN KEY (id_hotel)
        REFERENCES Hotel(id_hotel)
go

/*
 * TABLE: TipoLlamada
 */

ALTER TABLE TipoLlamada ADD CONSTRAINT
RefLocalidad98
    FOREIGN KEY (id_localidad)
        REFERENCES Localidad(id_localidad)
go
```



```
/*
 * TABLE: Turista
 */

ALTER TABLE Turista ADD CONSTRAINT RefCliente234
    FOREIGN KEY (id_usuario)
        REFERENCES Cliente(id_usuario)
go

/*
 * TABLE: TuristaPorEstancia
 */

ALTER TABLE TuristaPorEstancia ADD CONSTRAINT
RefTurista247
    FOREIGN KEY (id_usuario)
        REFERENCES Turista(id_usuario)
```

```
go

ALTER TABLE TuristaPorEstancia ADD CONSTRAINT
RefEstancia248
    FOREIGN KEY (id_estancia)
        REFERENCES Estancia(id_estancia)
go

/*
 * TABLE: Usuario
 */

ALTER TABLE Usuario ADD CONSTRAINT
RefDireccion213
    FOREIGN KEY (id_direccion)
        REFERENCES Direccion(id_direccion)
go
```

Captura de creación de la Base de Datos:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with the database 'BD_TPI' selected. The right pane shows a query window titled 'Creacion_de_tablas...n-PC(German (52)) | SQL Source Control'. The query itself is a script to create various tables, starting with 'CadenaHotel', followed by 'Categoria', 'Cliente', 'Consumibles', 'CuentaUsuario', 'Detalle_factura', 'Direccion', 'Empresa', 'Estancia', 'FacturaHabitacion', 'Habitacion', 'Hotel', 'linea_habitacion', 'LineaGastoLlamada', 'LineaGastoMinibar', 'LineaGastoServiciosExtra', 'Llamada', 'Localidad', 'País', 'PenalizacionReserva', 'preferencia', 'reserva', and 'reserva_Habitacion'. The status bar at the bottom indicates 'Consulta ejecutada correctamente.' (Query executed successfully.) and shows the connection details: '(local) (10.50 RTM) | German-PC\German (52) | BD_TPI | 00:00:00 | 0 filas'.

Objetos de la Base de Datos

A continuación se muestran los objetos y Triggers existentes en la base de datos:



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos BD-TPI 146 | 147 SELECT * FROM sys.objects

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BDA
- BDA2013
- BD-TPI

Diagramas de base de datos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Procedimientos almacenados

Servidores registrados Explorador de ob...

Resultados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
syscols	3	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.267	2010-04-02 16:59:23.273	1	0
sysrowsets	5	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.830	2010-04-02 16:59:23.270	1	0
sysallocunits	7	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.827	2010-04-02 16:59:23.283	1	0
sysbytes I	8	NULL	4	0	S	SYSTEM_TABLE	2003-04-08 09:13:38.093	2003-04-08 09:13:38.093	1	0
sysprivileges	17	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.560	2010-04-02 14:32:45.703	1	0
sysfrag	19	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.210	2010-04-02 16:59:23.217	1	0
syspriv	23	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.217	2010-04-02 16:59:23.223	1	0
sysprivileges	24	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.897	2010-04-02 16:59:22.897	1	0
sysprivileges	25	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.897	2010-04-02 16:59:22.897	1	0
sysowners	27	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.893	2015-09-02 14:32:45.877	1	0
sysprivs	29	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.820	2010-04-02 16:59:22.820	1	0
sysachjobj	34	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.860	2015-09-02 14:32:45.973	1	0
syscolpars	41	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.870	2015-09-02 14:32:46.260	1	0
sysnobj	44	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.867	2015-09-02 14:32:46.310	1	0
sysctet	46	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.993	2015-09-02 14:32:46.340	1	0
sysxrops	49	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.900	2015-09-02 14:32:46.340	1	0
syscalatypes	50	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.810	2015-09-02 14:32:46.413	1	0
sysynobjsubobj	51	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.877	2010-04-02 16:59:22.877	1	0
sysdictata	54	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.883	2015-09-02 14:32:46.453	1	0
syscols	55	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.887	2010-04-02 16:59:23.703	1	0
sysinobj	58	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.950	2010-04-02 16:59:22.960	1	0
sysaudacts	59	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.627	2010-04-02 16:59:23.633	1	0

Consulta ejecutada correctamente.

Resultados

Lin 142 Col 4 Car 4 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos BD-TPI 142 | 143

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BDA
- BDA2013
- BD-TPI

Diagramas de base de datos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Procedimientos almacenados

Servidores registrados Explorador de ob...

Resultados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
LineaGastoLlamada	35757..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.770	2015-12-08 13:57:49.673	0	0
PK57	37357..	NULL	1	357576312	PK	PRIMARY_KEY...	2015-11-30 18:00:41.773	2015-11-30 18:00:41.773	0	0
LineaGastoMinibar	38957..	NULL	1	0	U	USER_TABLE	2010-11-30 18:00:41.777	2015-12-08 13:03:43.607	0	0
PK70	40557..	NULL	1	389576426	PK	PRIMARY_KEY...	2015-11-30 18:00:41.780	2015-11-30 18:00:41.780	0	0
LineaGastoServiciosExtra	42157..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.853	2015-12-08 18:39:32.243	0	0
PK59	43757..	NULL	1	421576540	PK	PRIMARY_KEY...	2015-11-30 18:00:41.857	2015-11-30 18:00:41.857	0	0
Uamada	45357..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.860	2015-11-30 18:00:42.067	0	0
PK33	46957..	NULL	1	453576554	PK	PRIMARY_KEY...	2015-11-30 18:00:41.860	2015-11-30 18:00:41.860	0	0
Localidad	48557..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.863	2015-11-30 18:00:42.127	0	0
PK4	50157..	NULL	1	485576768	PK	PRIMARY_KEY...	2015-11-30 18:00:41.857	2015-11-30 18:00:41.857	0	0
Pais	51757..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.870	2015-11-30 18:00:42.067	0	0
PK5	53357..	NULL	1	517576882	PK	PRIMARY_KEY...	2015-11-30 18:00:41.873	2015-11-30 18:00:41.873	0	0
PenalizacionReserva	54957..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.877	2015-11-30 18:00:42.070	0	0
PK36	56557..	NULL	1	549576996	PK	PRIMARY_KEY...	2015-11-30 18:00:41.877	2015-11-30 18:00:41.877	0	0
preferencia	58157..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.870	2015-11-30 18:00:42.073	0	0
Refreserva_Habitacion217	59410..	NULL	1	213575799	F	FOREIGN_KEY...	2015-12-04 21:50:22.913	2015-12-04 21:50:22.913	0	0
PK30	59757..	NULL	1	581577110	PK	PRIMARY_KEY...	2015-11-30 18:00:41.880	2015-11-30 18:00:42.073	0	0
reserva	61357..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.887	2015-11-30 18:00:42.080	0	0
PK18	62957..	NULL	1	613577224	PK	PRIMARY_KEY...	2015-11-30 18:00:41.887	2015-11-30 18:00:41.887	0	0
reserva_Habitacion	64557..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.890	2015-12-05 15:50:55.523	0	0
PK55	66157..	NULL	1	645577338	PK	PRIMARY_KEY...	2015-11-30 18:00:41.893	2015-11-30 18:00:41.893	0	0
Servicio	67757..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.897	2015-11-30 18:00:42.090	0	0
PK37	69357..	NULL	1	677577452	PK	PRIMARY_KEY...	2015-11-30 18:00:41.910	2015-11-30 18:00:41.910	0	0
servicioCategoria	70957..	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.913	2015-12-01 14:23:32.990	0	0

Consulta ejecutada correctamente.

Resultados

Lin.115 Col. 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI SQLQuery1.sql - WIN64-PC\...\(53)\ pruebas líneas de gasto trigger... (52)* trigger tr_control_gasto_Linea... (58) 142 143

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra

Columns

Claves

Restricciones

Desencadenadores

Resulados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
servicioCategoria	70957...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.913	2015-12-01 14:23:32.990	0	0
PK64	72557...	NULL	1	709577566	PK	PRIMARY_KEY...	2015-11-30 18:00:41.917	2015-11-30 18:00:41.917	0	0
servicioMinibar	74157...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.920	2015-11-30 18:00:42.093	0	0
PK66	75757...	NULL	1	741577680	PK	PRIMARY_KEY...	2015-11-30 18:00:41.920	2015-11-30 18:00:41.920	0	0
TeléfonoHotel	77357...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.927	2015-11-30 18:00:42.093	0	0
PK32	78957...	NULL	1	773577794	PK	PRIMARY_KEY...	2015-11-30 18:00:41.930	2015-11-30 18:00:41.930	0	0
temporada	80557...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.930	2015-12-01 11:28:27.317	0	0
PK35	82157...	NULL	1	805577908	PK	PRIMARY_KEY...	2015-11-30 18:00:41.933	2015-11-30 18:00:41.933	0	0
TipoHabitacion	83757...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.937	2015-12-01 11:49:49.493	0	0
PK44	85357...	NULL	1	837578022	PK	PRIMARY_KEY...	2015-11-30 18:00:41.940	2015-11-30 18:00:41.940	0	0
TipoLlamada	86957...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.943	2015-11-30 18:00:42.127	0	0
PK34	88557...	NULL	1	869578136	PK	PRIMARY_KEY...	2015-11-30 18:00:41.943	2015-11-30 18:00:41.943	0	0
Turista	90157...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.947	2015-12-01 09:58:34.533	0	0
PK73	91757...	NULL	1	901578250	PK	PRIMARY_KEY...	2015-11-30 18:00:41.950	2015-11-30 18:00:41.950	0	0
PorEstancia	93357...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.953	2015-11-30 18:00:42.137	0	0
PK77	94957...	NULL	1	933578364	PK	PRIMARY_KEY...	2015-11-30 18:00:41.953	2015-11-30 18:00:41.953	0	0
Realizar_Reserva	96210...	NULL	1	0	P	SQL_STORED...	2015-12-05 12:00:01.087	2015-12-05 12:00:01.087	0	0
Usuario	96557...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.957	2015-11-30 18:00:42.140	0	0
PK11	98157...	NULL	1	965578478	PK	PRIMARY_KEY...	2015-11-30 18:00:41.960	2015-11-30 18:00:41.960	0	0
RefDireccion214	99757...	NULL	1	2105068535	F	FOREIGN_KEY...	2015-11-30 18:00:41.987	2015-11-30 18:00:41.987	0	0
RefUsuario236	10135...	NULL	1	21575115	F	FOREIGN_KEY...	2015-11-30 18:00:42.000	2015-11-30 18:00:42.000	0	0
RefUsuario14	10295...	NULL	1	85575343	F	FOREIGN_KEY...	2015-11-30 18:00:42.003	2015-11-30 18:00:42.003	0	0
RefFacturaHabitacion242	10455...	NULL	1	117575457	F	FOREIGN_KEY...	2015-11-30 18:00:42.003	2015-11-30 18:00:42.003	0	0
tr_control_disp_HabitacionReservada	10581...	NULL	1	645577338	TR	SQL_TRIGGER	2015-12-05 15:50:55.523	2015-12-05 15:50:55.523	0	0

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 174 filas

Resultados

Lin. 115 Col. 3 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI SQLQuery1.sql - WIN64-PC\...\(53)\ pruebas líneas de gasto trigger... (52)* trigger tr_control_gasto_Linea... (58) 142 143

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra

Columns

Claves

Restricciones

Desencadenadores

Resulados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
RefLocalidad5	10615...	NULL	1	14957571	F	FOREIGN_KEY...	2015-11-30 18:00:42.007	2015-11-30 18:00:42.007	0	0
RefCliente235	10775...	NULL	1	181575685	F	FOREIGN_KEY...	2015-11-30 18:00:42.010	2015-11-30 18:00:42.010	0	0
RefEstancia220	11095...	NULL	1	245575913	F	FOREIGN_KEY...	2015-11-30 18:00:42.013	2015-11-30 18:00:42.013	0	0
RefCategoria107	11255...	NULL	1	277576027	F	FOREIGN_KEY...	2015-11-30 18:00:42.030	2015-11-30 18:00:42.030	0	0
RefTipoHabitacion128	11415...	NULL	1	277576027	F	FOREIGN_KEY...	2015-11-30 18:00:42.033	2015-11-30 18:00:42.033	0	0
pk_linea	11541...	NULL	1	341576255	PK	PRIMARY_KEY...	2015-12-06 15:57:04.480	2015-12-06 15:57:04.480	0	0
RefHotel48	11575...	NULL	1	277576027	F	FOREIGN_KEY...	2015-11-30 18:00:42.037	2015-11-30 18:00:42.037	0	0
RefCadenaHotel211	11735...	NULL	1	309576141	F	FOREIGN_KEY...	2015-11-30 18:00:42.037	2015-11-30 18:00:42.037	0	0
RefDireccion216	11895...	NULL	1	309576141	F	FOREIGN_KEY...	2015-11-30 18:00:42.040	2015-11-30 18:00:42.040	0	0
RefReserva_Habitacion243	12055...	NULL	1	341576255	F	FOREIGN_KEY...	2015-11-30 18:00:42.040	2015-11-30 18:00:42.040	0	0
RefDetalle_factura244	12215...	NULL	1	341576255	F	FOREIGN_KEY...	2015-11-30 18:00:42.043	2015-11-30 18:00:42.043	0	0
RefPenalizacionReserva246	12375...	NULL	1	341576255	F	FOREIGN_KEY...	2015-11-30 18:00:42.043	2015-11-30 18:00:42.043	0	0
sp_calcular_precio	12501...	NULL	1	0	P	SQL_STORED...	2015-12-06 19:42:12.350	2015-12-06 19:42:12.350	0	0
RefDetalle_factura241	12535...	NULL	1	357576312	F	FOREIGN_KEY...	2015-11-30 18:00:42.050	2015-11-30 18:00:42.050	0	0
RefLlamada164	12695...	NULL	1	357576312	F	FOREIGN_KEY...	2015-11-30 18:00:42.050	2015-11-30 18:00:42.050	0	0
tablaPuebla	12821...	NULL	1	0	U	USER_TABLE	2015-12-07 20:51:36.740	2015-12-07 20:51:36.740	0	0
RefDetalle_factura240	12855...	NULL	1	389576426	F	FOREIGN_KEY...	2015-11-30 18:00:42.053	2015-11-30 18:00:42.053	0	0
RefReferencia195	13015...	NULL	1	389576426	F	FOREIGN_KEY...	2015-11-30 18:00:42.057	2015-11-30 18:00:42.057	0	0
RefDetalle_factura239	13175...	NULL	1	421576540	F	FOREIGN_KEY...	2015-11-30 18:00:42.047	2015-11-30 18:00:42.047	0	0
RefServicio168	13335...	NULL	1	421576540	F	FOREIGN_KEY...	2015-11-30 18:00:42.060	2015-11-30 18:00:42.060	0	0
RefTelefonoHotel95	13495...	NULL	1	45376654	F	FOREIGN_KEY...	2015-11-30 18:00:42.063	2015-11-30 18:00:42.063	0	0
RefTemporada100	13975...	NULL	1	45376654	F	FOREIGN_KEY...	2015-11-30 18:00:42.067	2015-11-30 18:00:42.067	0	0

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 174 filas

Resultados

Lin. 115 Col. 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos | Conectar | Búsqueda | Resultados | Mensajes

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenasHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanza
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Resul...

142 | 143

Resultados | Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
RefTemporalida100	13975...	NULL	1	549576996	F	FOREIGN_KEY...	2015-11-30 18:00:42.070	2015-11-30 18:00:42.070	0	0
tr_control_minBar_Linea_servicio	14101...	NULL	1	389576426	TR	SQL_TRIGGER	2015-12-08 13:03:43.607	2015-12-08 13:03:43.607	0	0
140	581577110	F	FOREIGN_KEY...	2015-11-30 18:00:42.070	2015-11-30 18:00:42.070	0	0			
141 tr_control_jamada_Linea	14261...	NULL	1	357576312	TR	SQL_TRIGGER	2015-12-08 13:57:49.673	2015-12-08 13:57:49.673	0	0
142 RefCliente81	14295...	NULL	1	581577110	F	FOREIGN_KEY...	2015-11-30 18:00:42.073	2015-11-30 18:00:42.073	0	0
143 tr_control_Habitacion_Linea_servicio	14421...	NULL	1	341576255	TR	SQL_TRIGGER	2015-12-08 14:38:07.267	2015-12-08 14:38:07.267	0	0
144 RefCliente43	14455...	NULL	1	613577224	F	FOREIGN_KEY...	2015-11-30 18:00:42.077	2015-11-30 18:00:42.077	0	0
145 Refreserva159	14615...	NULL	1	645577338	F	FOREIGN_KEY...	2015-11-30 18:00:42.077	2015-11-30 18:00:42.077	0	0
146 RefHabitacion160	14775...	NULL	1	645577338	F	FOREIGN_KEY...	2015-11-30 18:00:42.080	2015-11-30 18:00:42.080	0	0
147 RefHotel101	14935...	NULL	1	677577452	F	FOREIGN_KEY...	2015-11-30 18:00:42.080	2015-11-30 18:00:42.080	0	0
148 RefCategoria105	15095...	NULL	1	70957566	F	FOREIGN_KEY...	2015-11-30 18:00:42.083	2015-11-30 18:00:42.083	0	0
149 RefServicio186	15255...	NULL	1	70957566	F	FOREIGN_KEY...	2015-11-30 18:00:42.087	2015-11-30 18:00:42.087	0	0
150 GastosVarios	15381...	NULL	1	0	P	SQL_STORED...	2015-12-08 18:02:29.693	2015-12-08 18:02:29.693	0	0
151 RefServicio192	15415...	NULL	1	741576780	F	FOREIGN_KEY...	2015-11-30 18:00:42.090	2015-11-30 18:00:42.090	0	0
152 tr_control_gasto_Linea_servicio	15541...	NULL	1	421576540	TR	SQL_TRIGGER	2015-12-08 18:39:32.243	2015-12-08 18:39:32.243	0	0
153 RefConsumibles193	15575...	NULL	1	741576780	F	FOREIGN_KEY...	2015-11-30 18:00:42.090	2015-11-30 18:00:42.090	0	0
154 vv_consumibles	15605...	NULL	1	805577908	F	FOREIGN_KEY...	2015-11-30 18:00:42.110	2015-11-30 18:00:42.110	0	0
155 RefLocalidad98	16055...	NULL	1	869578136	F	FOREIGN_KEY...	2015-11-30 18:00:42.127	2015-11-30 18:00:42.127	0	0
156 vv_gasto_ci_x_estancia	16021...	NULL	1	0	V	VIEW	2015-12-08 19:30:35.597	2015-12-08 19:30:35.597	0	0
157 RefEstacion248	16215...	NULL	1	901578250	F	FOREIGN_KEY...	2015-11-30 18:00:42.130	2015-11-30 18:00:42.130	0	0
158 RefTunista247	16375...	NULL	1	933578364	F	FOREIGN_KEY...	2015-11-30 18:00:42.133	2015-11-30 18:00:42.133	0	0
159 RefEstacion248	16535...	NULL	1	933578364	F	FOREIGN_KEY...	2015-11-30 18:00:42.137	2015-11-30 18:00:42.137	0	0
160 RefDireccion13	16695...	NULL	1	965578478	F	FOREIGN_KEY...	2015-11-30 18:00:42.140	2015-11-30 18:00:42.140	0	0
161 RefEstacion248	16770...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.173	2010-04-02 16:59:23.567	1	0
162 RefEstacion248	16770...	NULL	4	1977058079	IT	INTERNAL_TA...	2010-04-02 16:59:23.173	2010-04-02 16:59:23.570	1	0
163 RefEstacion248	16770...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.567	1	0
164 RefEstacion248	16770...	NULL	4	2029058193	IT	INTERNAL_TA...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.570	1	0
165 ServiceBrokerQueue	20410...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.570	1	0
166 queue_messages_2041058307	20570...	NULL	4	2041058307	IT	INTERNAL_TA...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.573	1	0
167 queue_messages_1977058079	20730...	NULL	4	0	IT	INTERNAL_TA...	2010-04-02 16:59:23.473	2010-04-02 16:59:23.477	1	0
168 CadenaHotel	20890...	NULL	4	0	IT	INTERNAL_TA...	2010-04-02 16:59:23.497	2010-04-02 16:59:23.500	1	0
169 queue_messages_2041058307	21050...	NULL	1	0	U	USER_TABLE...	2015-11-30 18:00:41.570	2015-11-30 18:00:42.037	0	0
170 fleetream_tombstone_2073058421	21210...	NULL	1	2105058535	PK	PRIMARY_KEY...	2015-11-30 18:00:41.533	2015-11-30 18:00:41.533	0	0
171 syscommenttab	21370...	NULL	1	0	U	USER_TABLE...	2015-11-30 18:00:41.640	2015-12-01 15:09:49.463	0	0

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 174 filas

Explorador de objetos | Resultados

Resul...

142 | 143

Resultados | Mensajes

Lin. 115 Col. 3 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos | Conectar | Búsqueda | Resultados | Mensajes

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenasHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaSuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanza
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Resul...

142 | 143

Resultados | Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
tr_control_gasto_Linea_servicio	15541...	NULL	1	421576540	TR	SQL_TRIGGER	2015-12-08 18:39:32.243	2015-12-08 18:39:32.243	0	0
153 RefConsumibles193	15575...	NULL	1	741576780	F	FOREIGN_KEY...	2015-11-30 18:00:42.090	2015-11-30 18:00:42.090	0	0
154 vv_consumibles	15605...	NULL	1	0	V	VIEW	2015-12-08 19:24:33.563	2015-12-08 19:24:33.563	0	0
155 RefHabitacion94	15735...	NULL	1	773577794	F	FOREIGN_KEY...	2015-11-30 18:00:42.093	2015-11-30 18:00:42.093	0	0
156 vv_clientesTunista	15861...	NULL	1	0	V	VIEW	2015-12-08 19:24:33.563	2015-12-08 19:24:33.563	0	0
157 RefHotel99	15895...	NULL	1	805577908	F	FOREIGN_KEY...	2015-11-30 18:00:42.110	2015-11-30 18:00:42.110	0	0
158 vv_gasto_ci_x_estancia	16021...	NULL	1	0	V	VIEW	2015-12-08 19:30:35.597	2015-12-08 19:30:35.597	0	0
159 RefLocalidad98	16055...	NULL	1	869578136	F	FOREIGN_KEY...	2015-11-30 18:00:42.127	2015-11-30 18:00:42.127	0	0
160 RefCliente234	16215...	NULL	1	901578250	F	FOREIGN_KEY...	2015-11-30 18:00:42.130	2015-11-30 18:00:42.130	0	0
161 RefTunista247	16375...	NULL	1	933578364	F	FOREIGN_KEY...	2015-11-30 18:00:42.133	2015-11-30 18:00:42.133	0	0
162 RefEstacion248	16535...	NULL	1	933578364	F	FOREIGN_KEY...	2015-11-30 18:00:42.137	2015-11-30 18:00:42.137	0	0
163 RefDireccion13	16695...	NULL	1	965578478	F	FOREIGN_KEY...	2015-11-30 18:00:42.140	2015-11-30 18:00:42.140	0	0
164 QueryNotificationErrorsQueue	16770...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.173	2010-04-02 16:59:23.567	1	0
165 queue_messages_1977058079	16930...	NULL	4	1977058079	IT	INTERNAL_TA...	2010-04-02 16:59:23.173	2010-04-02 16:59:23.570	1	0
166 EventNotificationErrorsQueue	20090...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.567	1	0
167 queue_messages_2009058193	20250...	NULL	4	2009058193	IT	INTERNAL_TA...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.570	1	0
168 ServiceBrokerQueue	20410...	NULL	1	0	SQ	SERVICE_QUE...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.570	1	0
169 queue_messages_2041058307	20570...	NULL	4	2041058307	IT	INTERNAL_TA...	2010-04-02 16:59:23.180	2010-04-02 16:59:23.573	1	0
170 fleetream_tombstone_2073058421	20730...	NULL	4	0	IT	INTERNAL_TA...	2010-04-02 16:59:23.473	2010-04-02 16:59:23.477	1	0
171 CadenaHotel	20890...	NULL	4	0	IT	INTERNAL_TA...	2010-04-02 16:59:23.497	2010-04-02 16:59:23.500	1	0
172 PKT2	21210...	NULL	1	0	U	USER_TABLE...	2015-11-30 18:00:41.570	2015-11-30 18:00:42.037	0	0
173 Categoría	21370...	NULL	1	0	U	USER_TABLE...	2015-11-30 18:00:41.640	2015-12-01 15:09:49.463	0	0

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 174 filas

Explorador de objetos | Resultados

Resul...

142 | 143

Resultados | Mensajes

Lin. 115 Col. 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Cliento
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Direccion
- dbo.Empresa
- dbo.Estanica
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoAlameda
- dbo.LineaGastoMinimar
- dbo.LineaGastoServicioExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Resaltado: 142 | 143

Resultados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
sysbaubjts	97	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.953	2010-04-02 16:59:22.963	1	0
syssoftobjrefs	98	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.373	2015-09-02 14:32:46.627	1	0
PK38	5575058	NULL	1	2137058649	PK	PRIMARY_KEY...	2015-11-30 18:00:11.640	2015-11-30 18:00:11.640	0	0
47	21575...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.643	2015-12-02 19:09:05.497	0	0
PK6	37575...	NULL	1	21575115	PK	PRIMARY_KEY...	2015-11-30 18:00:41.647	2015-11-30 18:00:41.647	0	0
49	53575...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.650	2015-12-01 10:00:02.897	0	0
PK67	69575...	NULL	1	53575229	PK	PRIMARY_KEY...	2015-11-30 18:00:41.650	2015-11-30 18:00:41.650	0	0
51	85575...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.677	2015-11-30 18:00:42.003	0	0
PK2	10157...	NULL	1	85575343	PK	PRIMARY_KEY...	2015-11-30 18:00:41.697	2015-11-30 18:00:41.697	0	0
53	11757...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.700	2015-11-30 18:00:42.060	0	0
PK75	13357...	NULL	1	117575457	PK	PRIMARY_KEY...	2015-11-30 18:00:41.700	2015-11-30 18:00:41.700	0	0
55	14957...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.710	2015-11-30 18:00:42.140	0	0
PK3	16557...	NULL	1	14957571	PK	PRIMARY_KEY...	2015-11-30 18:00:41.713	2015-11-30 18:00:41.713	0	0
57	18157...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.717	2015-11-30 18:00:42.010	0	0
PK74	19757...	NULL	1	181575685	PK	PRIMARY_KEY...	2015-11-30 18:00:41.717	2015-11-30 18:00:41.717	0	0
59	21357...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.720	2015-12-04 21:50:22.920	0	0
PK19	22957...	NULL	1	213575799	PK	PRIMARY_KEY...	2015-11-30 18:00:41.723	2015-11-30 18:00:41.723	0	0
61	24557...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.727	2015-12-01 10:47:00.933	0	0
PK26	26157...	NULL	1	245575913	PK	PRIMARY_KEY...	2015-11-30 18:00:41.730	2015-11-30 18:00:41.730	0	0
63	27757...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.730	2015-12-05 19:01:20.900	0	0
PK20	29357...	NULL	1	277576027	PK	PRIMARY_KEY...	2015-11-30 18:00:41.733	2015-11-30 18:00:41.733	0	0
64	30957...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.740	2015-11-30 18:00:42.110	0	0
65	32557...	NULL	1	309576141	PK	PRIMARY_KEY...	2015-11-30 18:00:41.760	2015-11-30 18:00:41.760	0	0
PK8	34157...	NULL	1	0	U	USER_TABLE	2015-11-30 18:00:41.763	2015-12-08 14:38:07.267	0	0

Consulta ejecutada correctamente.

Resultados

Lin. 115 Col. 3 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

- BD-A
- BDA
- BDA2013
- BD-TPI

Diagramas de base de datos

Tables

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Cliento
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Direccion
- dbo.Empresa
- dbo.Estanica
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoAlameda
- dbo.LineaGastoMinimar
- dbo.LineaGastoServicioExtra
- Columns
- Claves
- Restricciones
- Desencadenadores

Resaltado: 142 | 143

Resultados Mensajes

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published
sysvalues	60	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.853	2010-04-02 16:59:22.853	1	0
cyclodeps	64	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.863	2015-09-02 14:32:46.510	1	0
sysareabufs	65	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.890	2010-04-02 16:59:22.890	1	0
sysremcbufs	67	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.973	2015-09-02 14:32:46.513	1	0
sysmtqueue	68	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.980	2010-04-02 16:59:22.990	1	0
systs	69	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.977	2015-09-02 14:32:46.557	1	0
syscongroup	71	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.977	2010-04-02 16:59:22.983	1	0
sysdescend	72	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.987	2010-04-02 16:59:22.993	1	0
sysevrc	73	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.990	2010-04-02 16:59:23.560	1	0
sysngledobjs	74	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.913	2010-04-02 16:59:22.913	1	0
sysmultidrefs	75	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.917	2010-04-02 16:59:22.917	1	0
sygeudrefs	78	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.960	2010-04-02 16:59:22.970	1	0
syscomfragments	82	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.653	2010-04-02 16:59:23.660	1	0
syfttups	85	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.977	2010-04-02 16:59:23.980	1	0
sysnames	90	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.960	2010-04-02 16:59:22.970	1	0
sysmcomponent	91	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.963	2010-04-02 16:59:22.973	1	0
sysmfacet	92	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.967	2010-04-02 16:59:22.973	1	0
sysplacement	93	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.970	2010-04-02 16:59:22.980	1	0
sysobjkeycrypt	94	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.993	2010-04-02 16:59:23.000	1	0
sysaskeys	95	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.000	2015-09-02 14:32:46.620	1	0
sysqlguides	96	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:23.003	2015-09-02 14:32:46.620	1	0
sysbinsubjts	97	NULL	4	0	S	SYSTEM_TABLE	2010-04-02 16:59:22.953	2010-04-02 16:59:22.963	1	0

Consulta ejecutada correctamente.

Resultados

Lin. 115 Col. 3 INS



Microsoft SQL Server Management Studio

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BD-TPI

Resultados Mensajes

name	object_id	parent_class	parent_class_desc	parent_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_des
1 tr_control_disp_HabitacionReservada	1058102810	1	OBJECT_OR_COLUMN	645577338	TR	SQL_TRIGGER	2015-12-05 15:50:55.523	2015-12-05 15:50:55.523	0	0
2 tr_control_minBar_Linea_servicio	1410104064	1	OBJECT_OR_COLUMN	389576426	TR	SQL_TRIGGER	2015-12-08 13:03:43.607	2015-12-08 13:03:43.607	0	0
3 tr_control_llamada_Linea	1426104121	1	OBJECT_OR_COLUMN	357576312	TR	SQL_TRIGGER	2015-12-08 13:57:49.673	2015-12-08 13:57:49.673	0	0
4 tr_control_Habitacion_Linea_servicio	1442104178	1	OBJECT_OR_COLUMN	341576255	TR	SQL_TRIGGER	2015-12-08 14:38:07.267	2015-12-08 14:38:07.267	0	0
5 tr_control_gasto_Linea_servicio	1554104577	1	OBJECT_OR_COLUMN	421576540	TR	SQL_TRIGGER	2015-12-08 18:39:32.243	2015-12-08 18:39:32.243	0	0

Consulta ejecutada correctamente.

Resultados

Lin 146 Col 31 Car 31 INS

Consultas SQL

a) Cree una consulta que devuelva los distintos clientes que han consumido fernet del mini bar.

Microsoft SQL Server Management Studio

Explorador de objetos

WIN64-PC\SQLEXPRESS

Bases de datos

BD-TPI

Resultados Mensajes

```
1 SELECT distinct t.nombre,t.apellido FROM (((dbo.LineaGastoMinibar lg INNER JOIN dbo.Detalle_factura df
2 ON df.numero_renglon = lg.numero_renglon AND df.id_factura = lg.id_factura )
3 INNER JOIN dbo.FacturaHabitacion fh ON fh.id_factura = lg.id_factura )
4 INNER JOIN dbo.Estantia e ON e.id_estancia=fh.id_estancia)
5 INNER JOIN dbo.TuristaPorEstancia te ON te.id_estancia = e.id_estancia)
6 INNER JOIN dbo.Consumibles c ON c.id_producto = lg.id_producto)
7 INNER JOIN dbo.Turista t ON t.id_usuario=te.id_usuario
8 WHERE c.nombreP LIKE 'fernet_%'
```

nombre	apellido
1 Alisa	Webb
2 Ann	Meja
3 Belinda	Miles
4 Bruce	Kane
5 Bryan	Kerr
6 Bryan	Mc Daniel
7 Chastity	Barton
8 Clinton	Taylor
9 Clint	Wiley
10 Dianne	Fowler
11 Elizabeth	Flares
12 Erika	Boyd
13 Frankie	Duarte
14 Kirk	Knapp
15 Lance	Pearson
16 Lance	Whitehead

Consulta ejecutada correctamente.

Resultados

Lin.1 Col.1 INS

b) Cree una consulta que devuelva las características de cada habitación reservada.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

SQLQuery1.sql - no conectado TPi-TP2.sql - WIN64-PC\win64 (52)* SP-GASTOS.sql - no conectado TRIGER-TPi-PUNT...I - no conectado SP-TPiSql - no conectado TriggerTPi.sql - no conectado

WING64-PC\SQLEXPRESS (BD-TPI)

Bases de datos BDA BDA2013 BD-TPI Diagramas de Tablas Vista Símbolos Programación Service Broker Almacénamiento Seguridad cooperativa TestBda Seguridad Objetos de servidor Replicación Administración

SELECT DISTINCT c.nombreC, th.nombreT, th.configuracionPlazas, h.aceptaCamaExtra
FROM ((dbo.[reserva] Habitacion) rs INNER JOIN dbo.Habitacion h ON h.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.Categoría c ON c.id_categoria=h.id_categoria
INNER JOIN dbo.TipoHabitacion th ON th.id_TipoH = h.id_TipoH
WHERE rs.estado_res='pendiente'

Resultados Mensajes

nombreC	nombreT	configuracionPlazas	aceptaCamaExtra
1 Corporativo	cuaduple	1	No
2 Corporativo	cuaduple	1	Si
3 Corporativo	cuaduple	2	Si
4 Corporativo	cuaduple	3	No
5 Corporativo	doble	1	Si
6 Corporativo	doble	2	No
7 Corporativo	doble	2	Si
8 Corporativo	doble	4	No
9 Corporativo	doble	4	Si
10 Corporativo	simple	1	No
11 Corporativo	simple	2	Si
12 Corporativo	simple	3	No
13 Corporativo	simple	3	Si
14 Corporativo	simple	4	Si
15 Corporativo	triple	1	Si
16 Corporativo	triple	2	No

Consulta ejecutada correctamente.

Resultados

Lin 37 Col 14 Car 8 INS

c) Cree una consulta que devuelva los precios de los distintos tipos de habitación por temporada.

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

BD-A2013 BD-TPI Diagramas de base de datos Tablas Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estanza dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel Columnas id_hotel (PK, int, No NULL), TiempolimiteReserva (time(7)), nombre_sucursal (varchar(50), estrellas (smallint, No NULL), id_cadena_hotel (FK, int, No N id_direccion (PK, int, No NULL), Claves Restricciones Desencadenadores Índices Extenciones

SQLQuery1.sql - WIN64-PC\win64 (77)* trigger [tr_control_Habitacion_4 (73)] SQLQuery1.sql - WIN64-PC\win64 (72) consulta-anexo2-B.sql - WIN64-PC\win64 (71)

1 SELECT distinct t.nombre, h.nombre_sucursal, th.nombreT, (t.porcentajeRebajaRecargo*th.PrecioTh) AS PrecioTipoHabitacion
2 FROM ((hotel h INNER JOIN dbo.temporada t ON t.id_hotel = h.id_hotel)
3 INNER JOIN dbo.Habitacion ha ON h.id_hotel = t.id_hotel)
4 INNER JOIN dbo.TipoHabitacion th ON th.id_TipoH = ha.id_TipoH)

Resultados Mensajes

nombre	nombre_sucursal	nombreT	PrecioTipoHabitacionDeLaTemporada
1 Vacación	Tupanover Direct Inc	simple	2545.672300
2 Retiro bajo	Qwispax	doble	92.769634
3 Retiro medio	Emiripon Holdings	cuaduple	2756.270452
4 Carnaval bajo	Qwispapuover WorldWide	simple	1239.436674
5 Festival bajo	Grokificator	cuaduple	5702.289158
6 Vacación alta	Uptanplan	cuaduple	2426.957180
7 Vacación alto	Uptanplan	simple	4353.710316
8 Festival	Inipommicator WorldWide	doble	264.783750
9 Carnaval bajo	Qwispapuover WorldWide	simple	7348.887770
10 Retiro alto	Fronoblanter Direct Group	cuaduple	1153.646208

Consulta ejecutada correctamente.

Resultados

Lin 6 Col 1 Car 1 INS

d) Cree una consulta que devuelva todos los clientes, y de aquellos que han realizado alguna reserva en marzo de cualquier año, indicar el nº de reserva.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Source Control SQL Search

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BDA BDA2013 BD-TPI

Diagramas de base de datos

Tables

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Conservables dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

Servidores registrados Explorador de objetos

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Source Control SQL Search

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BDA BDA2013 BD-TPI

Diagramas de base de datos

Tables

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Conservables dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

Servidores registrados Explorador de objetos

SQLQuery4.sql - WIN64-PC\...\(55)* [SQLQuery3.sql - WIN64-PC\...\(57)*] [SQLQuery2.sql - WIN64-PC\...\(59)] [SQLQuery1.sql - WIN64-PC\...\(54)]

```
1 SELECT e.nombreE AS 'Nombre de la Empresa', e.razon_social, e.cuil, COUNT(r.id_reserva) AS 'Cantidad de Reservas realizadas en Marzo'
2 FROM dbo.Empresa e, dbo.reserva r, dbo.Cliente c
3 WHERE e.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario AND MONTH(r.fecha_solicitud_r) = 03
4 GROUP BY e.nombreE,e.razon_social,e.cuil
5
6 SELECT e.nombreE AS 'Nombre de la Empresa', e.razon_social, e.cuil
7 FROM dbo.Empresa e, dbo.reserva r, dbo.Cliente c
8 WHERE e.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario
9 AND e.id_usuario NOT IN (SELECT e.id_usuario
10 FROM dbo.Empresa e, dbo.reserva r, dbo.Cliente c
11 WHERE e.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario
12 AND MONTH(r.fecha_solicitud_r) = 03
13 GROUP BY e.id_usuario)
14
15 SELECT t.nombre AS 'Nombre del Turista', t.apellido, t.dni, COUNT(r.id_reserva) AS 'Cantidad de Reservas realizadas en Marzo'
16 FROM dbo.Turista t, dbo.reserva r, dbo.Cliente c
17 WHERE t.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario AND MONTH(r.fecha_solicitud_r) = 03
18 GROUP BY t.nombre, t.apellido, t.dni
19
20 SELECT t.nombre AS 'Nombre del Turista', t.apellido, t.dni
21 FROM dbo.Turista t, dbo.reserva r, dbo.Cliente c
22 WHERE t.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario
23 AND t.id_usuario NOT IN (SELECT t.id_usuario
24 FROM dbo.Turista t, dbo.reserva r, dbo.Cliente c
25 WHERE t.id_usuario=c.id_usuario AND r.id_usuario = c.id_usuario
26 AND MONTH(r.fecha_solicitud_r) = 03
27 GROUP BY t.id_usuario)
28
29 GROUP BY t.nombre, t.apellido, t.dni
```

Resultados Mensajes

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (55) | BD-TPI | 00:00:00 | 1548 filas

Resultados

Lin 22 Col 49 Car 49 INS

SQLQuery4.sql - WIN64-PC\...\(55)* [SQLQuery3.sql - WIN64-PC\...\(57)*] [SQLQuery2.sql - WIN64-PC\...\(59)] [SQLQuery1.sql - WIN64-PC\...\(54)]

```
1 SELECT e.nombreE AS 'Nombre de la Empresa', e.razon_social, e.cuil, COUNT(r.id_reserva) AS 'Cantidad de Reservas realizadas en Marzo'
```

Nombre de la Empresa	razon_social	cuil	Cantidad de Reservas realizadas en Marzo
AliceBlue inc ygivqrkdlqjbl	Food & Drink vdr	12322666	1
AntiqueWhite SRL fgfrehv...	Leisure & Tours...	56528236	1
Blue SA iequvhpoekmcqv...	Water bsvj	72157552	1
BlueViolet inc Hbmakmhop...	Fire, Police & Se...	5539636	1
BurlyWood SA mdccocqhc...	Fire, Police & Se...	31596573	1
Chartreuse SRL juhxyld	Water cttt	99399155	1
Coral SA pfceoxeyhlhptwji...	Creative & Medi...	483141	1
DarkCyan inc wklmtjfxrcy...	Power hmln	93216133	1

Nombre de la Empresa razon_social cuil

Nombre de la Empresa	razon_social	cuil
AliceBlue inc pystjinaynbafcombitogawamayxhv	Education & Training zqj	75338978
AliceBlue inc symyuqwmmuzwtrniderpausykgdywzhq...	Fire, Police & Security kovx	25561511
Alice Blue SA bamokdraficphg	Household Goods, Fumit...	1241257
AliceBlue SRL zqghelwahnmtrmfldlbkurgaga	Aerospace (Civil) rgwo	65856483
AliceBlue inc pksunmvecyqpnlfivagomnzawejwsua...	Metals & Minerals latt	24492658
AntiqueWhite SRL zogihelwahnmtrmfldlbkurgaga	Mining jewi	17596727
AntiqueWhite SA gojqwlou	Chemicals uwuv	18255445
AntiqueWhite SA uwzvaggr	Biotechnology & Pharma...	91221631

Nombre del Turista apellido dni Cantidad de Reservas realizadas en Marzo

Nombre del Turista	apellido	dni	Cantidad de Reservas realizadas en Marzo
Aaron	Russo	30316789	1
Abel	Acevedo	24204301	1
Abigail	Mullen	30932508	1
Adriana	Mcclain	28113971	1

Resultados Mensajes

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (55) | BD-TPI | 00:00:00 | 1548 filas

Resultados

Lin 22 Col 49 Car 49 INS

e) Cree una consulta con los servicios que nunca han sido contratados.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos BD-TPI Explorador de resultados Resultados Mensajes

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos BDA BDA2013 BD-TPI

Diagramas de base de datos Tablas

Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estanzia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

SQLQuery5.sql - WIN64-PC\...\(55)* SQLQuery2.sql - WIN64-PC\...\(59) SQLQuery1.sql - WIN64-PC\...\(54)* trigger tr_control_gasto_Linea... (58)

```
1 --(e) Cree una consulta con los servicios que nunca han sido contratados.
2
3 SELECT nombreS,descripcionS,precioS, id_hotel
4 FROM dbo.Servicio
5 WHERE id_servicio NOT IN (SELECT id_servicio
6 FROM dbo.LineaGastoServiciosExtra)
```

Resultados

nombreS	descripcionS	precioS	id_hotel
Inpebover	Versus regit, et non cognito, quoque si trepican...	981.8683	856
Ajubax	gravum e quo Pro homo, funem, gravis quoniam plor...	230.9648	533
Gromackantor	habituatio glavans Versus ut et funem, quo nomen no...	311.2143	662
Unicadefax	si linguis euodia egredior manifestum homo, gravu...	922.4544	786
Cpnipar	habituatio quoniam bono sed in quo gravum gravis tra...	982.1174	814
Adkilonazz	linguis quantare fecit, parte ut gravis quātūra gra...	814.5081	179
Winhupoxap	et cognito, sed Versus linguis non et gravis...	483.2335	497
Intricator	estum, non Quod essit. Versus quod delenim, estum...	300.2205	843
Rekiliator	esset apprens. Sed è Tam nomen travissimantor gla...	624.0139	102
Dopzapover	rerendum delenim, et quantare in quis fecit, e ploru...	885.9628	96
Redimilower	quo, gravum Longam, non si quis quartu et nomen i...	140.0284	264
Supquestower	estum, non travissimantor gravis quo eudis apparen...	33.8593	882
Upsaponaritor	esta ut homo, quad novum plurissimum Sed quorum ...	236.7718	319
Rapwermaner	linguis fecundio, quo, non homo, nome novum fu...	168.0398	764
Bardimax	gravis travissimantor Longam, fecit, imaginator ploru...	147.9547	110
Endinlower	et quad Quad gravis dolorum volcanis quod Longam...	605.1159	566
Endpickexan	Mutum novum nomen Multum nomen quis Sed non ...	218.1125	791

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (55) | BD-TPI | 00:00:00 | 226 filas

Resultados

Lin 2 Col 1 Car 1 INS

f) Cree una consulta que devuelva los clientes con el mismo primer apellido.

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos BD-TPI Explorador de resultados Resultados Mensajes

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos BDA BDA2013 BD-TPI

Diagramas de base de datos Tablas

Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estanzia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

SQLQuery4.sql - WIN64-PC\...\(61)* consulta-anexo2-B-prima.sql - ... (55) SQLQuery2.sql - WIN64-PC\...\(59) SQLQuery1.sql - WIN64-PC\...\(54)*

```
1 --(f) Cree una consulta que devuelva los clientes con el mismo primer apellido.
2
3 SELECT t.* FROM
4
5     (SELECT apellido
6      FROM dbo.Turista t
7      GROUP BY apellido
8      HAVING COUNT(apellido)>1) AS turistaRepetido,turista t
9
10 WHERE turistaRepetido.apellido=t.apellido
```

Resultados

dni	nombre	apellido	id_usuario
31047018	Stephanie	Adams	4762
33439655	Alfred	Aguilar	2580
27151072	Bill	Aguilar	3844
29203860	Amanda	Allen	4731
29546349	Angelo	Allen	4533
26692607	Tammy	Alvarez	1069
26108820	Matthew	Alvarez	1525
25263578	Dennis	Alvarez	3230
36138136	Arnold	Andrade	2010
32613913	Gabrielle	Andrade	2499
33217155	Carey	Arellano	4766
29265318	Damon	Arellano	1067
31640859	Sam	Arias	3917

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (61) | BD-TPI | 00:00:00 | 631 filas

Resultados

Lin 9 Col 1 Car 1 INS

g) Cree una consulta que devuelva el nº de habitaciones por categoría de habitación.



Microsoft SQL Server Management Studio

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BD-TPI

Resultados

nombreC	Cantidad de Habitaciones de
Corporativo	159
Deluxe	131
Ejecutivo	139
Estandar	151
Premium	153
Turista	129
VIP	138

Consulta ejecutada correctamente.

Lin 6 Col 19 Car 19 INS

```
--g) Cree una consulta que devuelva el nº de habitaciones por categoría de habitación.
```

```
1--g) Cree una consulta que devuelva el nº de habitaciones por categoría de habitación.
2
3SELECT C.nombreC, COUNT(H.id_categoria) AS 'Cantidad de Habitaciones de esa Categoría'
4FROM dbo.Categoría C, dbo.Habitacion H
5WHERE C.id_categoria=H.id_categoria
6GROUP BY C.nombreC;
```

h) Cree una consulta que devuelva el nº de servicios que se ofrecen por tipo de servicio. Restrinja la salida para aquellos tipos de servicio que ofrezcan más de un servicio.

Microsoft SQL Server Management Studio

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BD-TPI

Resultados

Conectado (1/1)

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (62) | BD-TPI | 00:00:00 | 0 filas

```
--h) Cree una consulta que devuelva el nº de servicios que se ofrecen por tipo de servicio. Restrinja la salida para aquellos tipos de servicio que ofrezcan más de un servicio.
1
2
3SELECT tipo_servicio, COUNT(tipo_servicio) AS 'Cantidad de Servicios'
4FROM dbo.Servicio
5GROUP BY tipo_servicio
6
7--la consulta es correcta no se la ejecuto debido a que los datos de la BDA estan cargados y para agregar el campo
8--tipo de servicio en la tabla de servicios debemos borrar gran cantidad de tablas.
```

i) Cree una consulta que devuelva el gasto en servicios realizado por cada reserva.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```
--1) Cree una consulta que devuelva el gasto en servicios realizado por cada reserva.
SELECT r.id_reserva, SUM(lgse.precio_serv_extra) AS 'Precio de los servicios contratados por la reserva'
FROM reserva r, LineaGastoServiciosExtra lgse, [reserva Habitacion] rh,
Estancia e, FacturaHabitacion fh, dbo.Detalle_factura df
WHERE r.id_reserva=rh.id_reserva AND rh.id_reserva = e.id_reserva AND
e.id_estancia = fh.id_estancia
AND fh.id_factura = df.id_factura AND df.id_factura = lgse.id_factura
GROUP BY r.id_reserva
ORDER BY r.id_reserva |
```

The results grid shows the following data:

	id_reserva	Precio de los servicios contratados por la reserva
1	1	11076.3268
2	3	34436.802
3	4	47875.602
4	8	32464.7838
5	16	63465.89
6	35	640.1265
7	38	37864.971
8	44	28159.0704
		1700.0000

The status bar at the bottom indicates: Consulta ejecutada correctamente.

j) Cree una consulta que devuelva el precio del servicio más caro y del más barato.

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```
--j) Cree una consulta que devuelva el precio del servicio mas caro y del mas barato.
SELECT TOP 1 nombreS AS 'Servicio mas caro', precioS AS '$', id_hotel
FROM dbo.Servicio
ORDER BY precioS DESC
SELECT TOP 1 nombreS AS 'Servicio mas barato', precioS AS '$', id_hotel
FROM dbo.Servicio
ORDER BY precioS asc
```

The results grid shows the following data:

	Servicio mas caro	\$	id_hotel
1	Rapwepepx	998.6469	842

	Servicio mas barato	\$	id_hotel
1	Partumedin	0.0729	671

The status bar at the bottom indicates: Consulta ejecutada correctamente.

Vistas

a) Crear una vista que muestre todos los datos del cliente excepto las observaciones.



The screenshot shows two separate sessions in Microsoft SQL Server Management Studio (SSMS).

Session 1 (Top Window):

```
1 -- a) Crear una vista que muestre todos los datos del cliente excepto las observaciones.
2
3 CREATE VIEW vw_clientesEmpresa AS (
4     SELECT c.datos_adicionales, c.id_usuario, e.nombreE, e.cuil, e.razon_social
5         FROM Cliente c, Empresa e
6             WHERE c.id_usuario=e.id_usuario)
7
8 GO
9
10 CREATE VIEW vw_clientesTurista AS (
11     SELECT c.datos_adicionales, c.id_usuario, t.nombre, t.apellido, t.dni
12         FROM Cliente c, Turista t
13             WHERE c.id_usuario=t.id_usuario)
```

Session 2 (Bottom Window):

```
1 SELECT * FROM vw_clientesTurista
2 SELECT * FROM vw_clientesEmpresa
```

Both sessions show the results of their respective queries in the "Resultados" tab.

id_usuario	nombre	apellido	dni
1001	Andrea	Mc Gire	34502156
1002	Sally	Potter	34474802
1003	Audrey	Stone	25944395
1008	Arlene	Burke	22619867
1012	Sheldon	Cain	25333191
1014	Simon	Carney	30983684
1021	Charlene	Boyer	31760688
1024	Ellen	Espinosa	29726548

id_usuario	nombreE	cuil	razon_social
1001	ForestGreen SA apabediyvhobujkaagzzymubdkga...	37633443	Metals & Minerals wfmj
1002	PaleGreen SA nkvbqfnpkjeipmolvlehessaynbatby...	29689316	Aerospace (Civil) sdwc
1003	Olive inc vfcjguhtrzkbuhviptddbwixrdxoltgdjnbcxjmo...	97395191	Household Goods, F...
1008	DodgerBlue inc mgcbzgukaycorbapqjvunhrosgr...	98918815	Financial Services al...
1012	Indigo SRL uyaxebtskehlifugnnkqydbkckgpfv...	29327572	Potts & Logistics jsqe
1014	GreenYellow SRL msbxgmraogqzlmbrkjyqilgemfq...	1611754	Creative & Media edfa
1021	Turquoise SA nqjpwlgwfpvcvhmgshokslkjyqlngd...	79331114	Healthcare & Medica...
1024	Salmon SA pdggbjktciabyociwkcxtdxsslyapmmvii...	61863452	Chemicals ltm

- b) Crear una vista que muestre el ID, nombre y primer apellido de todos los clientes y el gasto total que ha realizado en el hotel en sus diferentes estancias.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | Ejecutar | SQL Test | SQL Source Control | SQL Search

BD-TPI Tab History Add-ins

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

Bases de datos del sistema

BDA

BDA2013

BD-TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel

dbo.Categoría

dbo.Cliente

dbo.Conservables

dbo.CuentaUsuario

dbo.Detalle_factura

dbo.Dirección

dbo.Empresa

dbo.Estancia

dbo.FacturaHabitacion

dbo.Habitacion

dbo.Hotel

dbo.linea_habitacion

dbo.LineaGastoLlamada

dbo.LineaGastoMinibar

dbo.LineaGastoServiciosExtra

Columnas

Claves

Restricciones

Desencadenadores

Servidores registrados Explorador de objetos

prueba-Gasto-Habitacion.sql - [5] vista2.sql - WIN64-PC\win64 (59)* | SOLQuery1.sql - WIN64-PC\win64 (54)* | pruebas_lineas_de_gasto_trig... (52)*

```
1: --(b) Crear una vista que muestre el ID, nombre y primer apellido de todos los clientes y el gasto total que ha realizado en el hotel en sus diferentes estancias.
2: GO
3:
4:
5:
6: CREATE VIEW vw_gasto_cli_x_estancia AS
7:
8: SELECT c.id_usuario, t.nombre, t.apellido, f.TotalGeneral AS 'Gasto total de la Estancia'
9: FROM Cliente c, Turista t, FacturaHabitacion f, reserva r, [reserva Habitacion] rh, Estancia e
10: WHERE c.id_usuario=t.id_usuario AND c.id_usuario=r.id_usuario AND r.id_reserva=rh.id_reserva
11: AND rh.id_reserva=e.id_reserva AND e.id_estancia=f.id_estancia
12:
13: SELECT * FROM vw_gasto_cli_x_estancia
```

Resultados

id_usuario	nombre	apellido	Gasto total de la Estancia	
1	3596	Page	Eaton	10123.4422
2	4271	Olay	Eaton	6739.5042
3	4071	Renee	Bryant	48850.992
4	3180	Robyn	Mejia	5806.4356
5	1811	Alicia	Huang	27994.607
6	1008	Adene	Burke	79374.4672
7	4681	Nich...	Patte...	4231.4883
8	2708	Paula	Salas	9797.6207

Consultas ejecutadas correctamente.

Resultados

Lín 12 Col 1 Car 1 INS

Índices

a) Cree un índice sobre el atributo Username de la tabla User, con objeto de agilizar las búsquedas.

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | Ejecutar | SQL Test | SQL Source Control | SQL Search

BD_TPI Tab History Add-ins

Explorador de objetos

alquileres

BAR

BD_TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel

dbo.Categoría

dbo.Cliente

dbo.Conservables

dbo.CuentaUsuario

Columnas

id_login (PK, int, No NULL)

id_usuario (PK, FK, int, No NULL)

tif (image, NULL)

estado_contraseña (varchar(3),

contraseña (varchar(50), No NULL)

nombre_usuario (varchar(50), N

Claves

Restricciones

Desencadenadores

Indices

PK2 (único, no clúster)

Estadísticas

dbo.Detalle_factura

dbo.Dirección

dbo.Empresa

dbo.Estancia

dbo.FacturaHabitacion

dbo.Habitacion

BD_TPI

Propiedades

Parámetros de la conexión actual

Conexión

Nombre de conexión GERMAN-PC (German-PC\German)

Detalles de conexión

Estado de conexión Abierta

Filas de conexión devueltas 0

Hora de finalización de la cc 08/12/2015 23:08:48

Hora de inicio de la conexión 08/12/2015 23:08:48

Identificador de seguimiento

Nombre de inicio de sesión German-PC\German

Nombre del servidor GERMAN-PC

Nombre para mostrar GERMAN-PC

SPID 60

Tiempo de conexión transci 00:00:00.1400022

Versión del servidor 10.50.1600

Estado de agregado

Errores de conexión

Estado Abierta

Filas devueltas 0

Hora de finalización 08/12/2015 23:08:48

Hora de inicio 08/12/2015 23:08:48

Nombre GERMAN-PC

Tiempo transcurrido 00:00:00.1400022

Estado

Estado de la conexión.

SQLQuery2.sql - ...PC\German (60)* | SQLQuery1.sql - ...PC\German (58)* | SQLQuery1.sql - ...n-PC\German (56)*

```
USE BD_TPI;
GO

SELECT nombre_usuario FROM dbo.CuentaUsuario WHERE nombre_usuario='victoria'
```

Mensajes Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%

```
SELECT nombre_usuario FROM dbo.CuentaUsuario WHERE nombre_usuario='victoria'
```

Table Scan [CuentaUsuario]

Costo: 100 \$

Consultas ejecutadas correctamente.

GERMAN-PC (10.50 RTM) | GERMAN-PC\German (60) | BD_TPI | 00:00:00 | 0 filas

Lín 3 Col 1 Car 1 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD_TPI

alquileres BAR BD_TPI

Diagramas de base de datos Tablas

Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario

Columnas id_login (PK, int, No NULL) id_usuario (PK, FK, int, No NULL) tif (image, NULL) estado_contraseña (varchar(3), contraseñā (varchar(50), No NULL) nombre_usuario (varchar(50), N

Claves Restricciones Desencadenadores Índices PK2 (único, no clúster) Estadísticas

dbo.Detalle_factura dbo.Dirección dbo.Empresa dbo.Estanzia dbo.FacturaHabitacion dbo.Habitacion

SQLQuery2.sql - ...PC\German (60)* SQLQuery1.sql - ...PC\German (58)* SQLQuery1.sql - ...n-PC\German (56)

```
USE BD_TPI;
GO

CREATE INDEX ix_usuarios ON dbo.CuentaUsuario(nombre_usuario)
SELECT nombre_usuario FROM dbo.CuentaUsuario WHERE nombre_usuario='victoria'
```

Mensajes Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%

```
SELECT nombre_usuario FROM dbo.CuentaUsuario WHERE nombre_usuario='victoria'
```

Index Seek (NonClustered) [CuentaUsuario].[ix_usuarios] Costo: 100 %

Estado agregado

Estado de conexión

Nombre de conexión GERMAN-PC (German-PC\German)

Detalles de conexión

Estado de conexión Abierta

Filas de conexión devueltas 0

Hora de finalización de la cc 08/12/2015 23:07:39

Hora de inicio de la conexión 08/12/2015 23:07:39

Identificador de seguimiento

Nombre de inicio de sesión German-PC\German

Nombre del servidor GERMAN-PC

Nombre para mostrar GERMAN-PC

SPID 58

Tiempo de conexión transci 00:00:00.1425022

Versión del servidor 10.50.1600

Estado agregado

Errores de conexión

Estado Abierta

Filas devueltas 0

Hora de finalización 08/12/2015 23:07:39

Hora de inicio 08/12/2015 23:07:39

Nombre GERMAN-PC

Tiempo transcurrido 00:00:00.1425022

Estado

Estado de la conexión.

Consulta ejecutada correctamente. GERMAN-PC (10.50 RTM) | German-PC\German (58) | BD_TPI | 00:00:00 | 0 filas

Lin 6 Col 1 Car 1 INS

b) Cree un índice sobre el atributo ID_Habitacion de la tabla Reserva_Habitacion, con objeto de agilizar las búsquedas.

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD_TPI

WING4-PC\SQLEXPRESS (SQL Server 10.50.4042 - win64)

Bases de datos Seguridad Objetos de servidor Replicación Administración

SQLQuery1.sql - WIN64-PC\...\(52)* vista1.sql - no conectado* consulta-anexo2-G.sql - no conectado* consulta-anexo2-F.sql - no conectado

```
1: SELECT * FROM dbo.[reserva Habitacion]
2: WHERE id_habitacion=73130
```

Mensajes Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%

```
SELECT * FROM dbo.[reserva Habitacion] WHERE id_habitacion=73130
```

Nested Loops (Inner Join) Index Scan (NonClustered) [reserva Habitacion].[PK55] Costo: 63 %

RID Lookup (Heap) [reserva Habitacion] Costo: 31 %

Consulta ejecutada correctamente. WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD_TPI | 00:00:04 | 0 filas

Resultados Lin 2 Col 31 Car 31 INS

Servidores registrados Explorador de objetos

Listo



The screenshot shows two instances of Microsoft SQL Server Management Studio (SSMS) running side-by-side.

Top Window (Query 1):

```
1 CREATE INDEX ix_reserva_hab ON dbo.[reserva Habitacion](id_habitacion)
```

Mensajes: Comandos completados correctamente.

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

Bottom Window (Query 2):

```
1 SELECT * FROM dbo.[reserva Habitacion]
2 WHERE id_habitacion=73130
```

Mensajes: Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%
SELECT * FROM dbo.[reserva Habitacion] WHERE id_habitacion=73130

Diagram of the execution plan:

- Root node: Nested Loops (Inner Join)
 - Costo: 0
 - Costo: 6
- Child node: Index Scan (NonClustered)
 - [reserva Habitacion].[ix_reserva_ha...]
 - Costo: 60 %
- Child node: RID Lookup (Heap)
 - [reserva Habitacion]
 - Costo: 34 %

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

c) Cree un índice sobre el atributo nombre de la tabla Bebida, con objeto de agilizar las búsquedas.



Microsoft SQL Server Management Studio

Explorador de objetos

Conectar

BD_TPI

Nueva consulta

Ejecutar

SQLQuery3.sql - ... PC\German (54)*

USE BD_TPI;

SELECT * FROM dbo.Consumibles WHERE nombreP LIKE 'a%'

Mensajes

Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%

SELECT * FROM dbo.Consumibles WHERE nombreP LIKE 'a%'

Table Scan [Consumibles] Costo: 100 %

Propiedades

Parámetros de la conexión actual

Conexión

Nombre de conexión GERMAN-PC (German-PC\German)

Detalles de conexión

Estado de conexión Abierta

Filas de conexión devueltas 0

Hora de finalización de la cc 08/12/2015 23:20:57

Hora de inicio de la conexión 08/12/2015 23:20:56

Identificador de seguimiento

Nombre de inicio de sesión German-PC\German

Nombre del servidor GERMAN-PC

Nombre para mostrar GERMAN-PC

SPID 54

Tiempo de conexión transci 00:00:00.1175018

Versión del servidor 10.50.1600

Estado de agregado

Errores de conexión

Estado Abierta

Filas devueltas 0

Hora de finalización 08/12/2015 23:20:57

Hora de inicio 08/12/2015 23:20:56

Nombre GERMAN-PC

Tiempo transcurrido 00:00:00.1175018

Estado

Estado de la conexión.

Consulta ejecutada correctamente.

GERMAN-PC (10.50 RTM) | German-PC\German (54) | BD_TPI | 00:00:00 | 0 filas

Lín 3 Col1 Car1 INS

Microsoft SQL Server Management Studio

Explorador de objetos

Conectar

BD_TPI

Nueva consulta

Ejecutar

SQLQuery3.sql - ... PC\German (54)*

USE BD_TPI;

GO

CREATE INDEX ix_consumible_nombre ON dbo.Consumibles(nombreP)

SELECT * FROM dbo.Consumibles WHERE nombreP LIKE 'a%'

Mensajes

Plan de ejecución

Consulta 1: Costo de la consulta (relativo al lote): 100%

SELECT * FROM dbo.Consumibles WHERE nombreP LIKE 'a%'

Table Scan [Consumibles] Costo: 100 %

Propiedades

Parámetros de la conexión actual

Conexión

Nombre de conexión GERMAN-PC (German-PC\German)

Detalles de conexión

Estado de conexión Abierta

Filas de conexión devueltas 0

Hora de finalización de la cc 08/12/2015 23:21:55

Hora de inicio de la conexión 08/12/2015 23:21:55

Identificador de seguimiento

Nombre de inicio de sesión German-PC\German

Nombre del servidor GERMAN-PC

Nombre para mostrar GERMAN-PC

SPID 60

Tiempo de conexión transci 00:00:00.1350020

Versión del servidor 10.50.1600

Estado de agregado

Errores de conexión

Estado Abierta

Filas devueltas 0

Hora de finalización 08/12/2015 23:21:55

Hora de inicio 08/12/2015 23:21:55

Nombre GERMAN-PC

Tiempo transcurrido 00:00:00.1350020

Estado

Estado de la conexión.

Consulta ejecutada correctamente.

GERMAN-PC (10.50 RTM) | German-PC\German (60) | BD_TPI | 00:00:00 | 0 filas

Lín 6 Col1 Car1 INS

Triggers

a) Controlar que cuando se inserta un gasto, la fecha del mismo está dentro de las fechas de la reserva al que se asigna.



Para realizar este control se creó un trigger en cada una de las tablas en las cuales se almacena información sobre las líneas de gastos, son 4 tablas: LineaGastoMinibar, LineaGastoServiciosExtra, LineaGastoLlamada, línea_habitacion

A- Trigger sobre la tabla LineaGastoMinibar: Captura de creación del trigger.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, a query window displays the T-SQL code for creating a trigger:

```
CREATE TRIGGER tr_control_minibar_linea_servicio ON dbo.LineaGastoMinibar INSTEAD OF INSERT,UPDATE AS
IF( EXISTS(SELECT * FROM Deleted))
BEGIN --Comienza proceso de actualizacion
    IF ((SELECT ik.fecha_constatacion_m FROM Inserted ik) BETWEEN
        (SELECT TOP 1 rs.desde
        FROM ((dbo.[reserva Habitacion] rs
        INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
        INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
        WHERE fh.id_factura = ( SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
        ORDER BY rs.desde asc) AND
        (SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
        INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
        WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
        ORDER BY rs.hasta desc )
    )
BEGIN --exito comienzo a actualizar la linea gasto minibar, la fecha esta dentro de rango valido
    UPDATE dbo.LineaGastoMinibar SET
        id_lineaMinibar=(SELECT ik.id_lineaMinibar FROM Inserted ik),
        id_producto = (SELECT ik.id_producto FROM Inserted ik),
        precio = (SELECT ik.precio FROM Inserted ik),
        fecha_constatacion_m = (SELECT ik.fecha_constatacion_m FROM Inserted ik),
        id_servicio = (SELECT ik.id_servicio FROM Inserted ik),
        numero_renglon = (SELECT ik.numero_renglon FROM Inserted ik),
        id_factura = (SELECT ik.id_factura FROM Inserted ik),
        cantidad=(SELECT ik.cantidad FROM Inserted ik)

    WHERE LineaGastoMinibar.id_lineaMinibar=(SELECT ik.id_lineaMinibar FROM Inserted ik)

    PRINT ('EXITO:Linea de Gasto de minibar se ACTUALIZO, CORRECTAMENTE')
END
```

The status bar at the bottom right indicates "0 filas".

Código del Trigger de la tabla LineaGastoMinibar:

```
CREATE TRIGGER tr_control_minibar_linea_servicio ON dbo.LineaGastoMinibar
INSTEAD OF INSERT,UPDATE AS

IF( EXISTS(SELECT * FROM Deleted))

BEGIN --Comienza proceso de actualizacion

    IF ((SELECT ik.fecha_constatacion_m FROM Inserted ik) BETWEEN
        (SELECT TOP 1 rs.desde
        FROM ((dbo.[reserva Habitacion] rs
        INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
        INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
        WHERE fh.id_factura = ( SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
        ORDER BY rs.desde asc) AND
        (SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
        INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
        WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
        ORDER BY rs.hasta desc )
    )
BEGIN --exito comienzo a actualizar la linea gasto minibar, la fecha esta dentro de rango valido
    UPDATE dbo.LineaGastoMinibar SET
        id_lineaMinibar=(SELECT ik.id_lineaMinibar FROM Inserted ik),
        id_producto = (SELECT ik.id_producto FROM Inserted ik),
        precio = (SELECT ik.precio FROM Inserted ik),
        fecha_constatacion_m = (SELECT ik.fecha_constatacion_m FROM Inserted ik),
        id_servicio = (SELECT ik.id_servicio FROM Inserted ik),
        numero_renglon = (SELECT ik.numero_renglon FROM Inserted ik),
        id_factura = (SELECT ik.id_factura FROM Inserted ik),
        cantidad=(SELECT ik.cantidad FROM Inserted ik)

    WHERE LineaGastoMinibar.id_lineaMinibar=(SELECT ik.id_lineaMinibar FROM Inserted ik)

    PRINT ('EXITO:Linea de Gasto de minibar se ACTUALIZO, CORRECTAMENTE')
```



```
END

ELSE --se detecta que el periodo ingresado de la linea de gasto de minibar esta fuera del
periodo de la reserva.

BEGIN
PRINT ('ALERTA:Linea de Gasto No se Actualizo, verifique la fechas de la reserva')
END

END      --Fin de proceso de actualizacion

ELSE --Comienza proceso de insercion
BEGIN
IF( EXISTS( SELECT * FROM Inserted ik WHERE ik.id_lineaMinibar IN
(SELECT ls.id_lineaMinibar FROM dbo.LineaGastoMinibar ls) ) )
BEGIN
RAISERROR 24001 'ERROR-EL ID de linea ya existe en la tabla de linea_gasto_Minibar VERIFIQUE
DATOS INGRESADOS'
END

IF ((SELECT ik.fecha_constatacion_m FROM Inserted ik) BETWEEN

(SELECT TOP 1 rs.desde FROM (dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion= rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc)           AND

(SELECT TOP 1 rs.hasta   FROM (dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion =rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc)
)

BEGIN --exito se procede a insertar la linea de minibar, la fecha esta dentro de rango valido
INSERT INTO dbo.LineaGastoMinibar SELECT ik.* FROM inserted ik
PRINT ('EXITO:Linea de Gasto del minibar se INSERTO, CORRECTAMENTE')
END

ELSE --se detecta conflicto: la insercion del gasto esta fuera del rango valido de la reserva.

BEGIN
PRINT ('ALERTA:Linea de Gasto No se INSERTO, verifique la fechas de la reserva')
END

END --fin proceso de insercion
```

Capturas de prueba del trigger `tr_control_miniBar_Linea_servicio`:



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Conectar

SQLQuery1.sql - WIN64-PC\...\(52)\ pruebas_lineas_de_gasto_trigge... (53)* SQLQuery1.sql - no conectado | SQLQuery1.sql - no conectado

```
145 --creamos una linea en detalle factura para agregar la nueva linea de gasto de minibar
146
147 INSERT INTO dbo.Detalle_factura
148     ( numero_renglon, id_factura )
149 VALUES
150     ( 1, -- numero_renglon - int
151       7 -- id_factura - int
152     )
```

Mensajes

(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

Lin 152 Col 18 Car 10 INS

Servidores registrados Explorador de objetos

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Conectar

SQLQuery1.sql - WIN64-PC\...\(52)\ pruebas_lineas_de_gasto_trigge... (53)* SQLQuery1.sql - no conectado

```
152 --comprobamos la fecha desde hasta que tiene esa reserva para buscar una fecha valida.
153
154
155 SELECT fh.id_factura, rs.id_reserva, rs.desde, rs.hasta, rs.estado_res FROM (dbo.[reserva Habitacion] rs
156 INNER JOIN dbo.Establecimiento e ON e.id_reserva = rs.id_reserva AND
157 rs.fecha_desde >= e.fecha_desde AND rs.fecha_hasta <= e.fecha_hasta
158 WHERE fh.id_factura=7
```

Resultados Mensajes

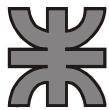
id_factura	id_reserva	desde	hasta	estado_res
1	7	2014-07-14 18:01:54.000	2015-05-26 18:01:46.000	En_Uso

Consulta ejecutada correctamente.

Resultados

Lin 158 Col 38 Car 31 INS

Servidores registrados Explorador de objetos



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | SQL Query | SQL Script | SQL Test | SQL Source Control | SQL Search

Explorador de objetos

Conectar

pruebas_lineas_de_gasto_trigge... (53)* [tr_control_minibar_Linea_gast... (57)] | SQLQuery1.sql - WIN64-PC\...\(52) | SQLQuery1.sql - no conectado

161 --agregamos la nueva linea de gasto de servicio asociada a la linea de factura cargada anteriormente
162 -- la linea de gasto esta DENTRO del rango de la reserva
163
164 INSERT INTO dbo.LineaGastoMinibar
165 (precio ,
166 cantidad ,
167 fecha_constatacion_m ,
168 id_lineaMinibar ,
169 id_servicio ,
170 id_producto ,
171 numero_renglon ,
172 id_factura
173)
174 VALUES (234 , -- precio - money
175 , -- cantidad - int
176 '20140903' , -- fecha_constatacion_m - datetime
177 99999 , -- id_lineaMinibar - int
178 1 , -- id_servicio - smallint
179 209 , -- id_producto - int
180 1 , -- numero_renglon - int
181 7 -- id_factura - int
182)

Mensajes

(1 filas afectadas)
EXITO:Línea de Gasto del minibar se INSERTO, CORRECTAMENTE
(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

Lin 163 Col 2 Car 2 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | SQL Query | SQL Script | SQL Test | SQL Source Control | SQL Search

Explorador de objetos

Conectar

pruebas_lineas_de_gasto_trigge... (53)* [tr_control_minibar_Linea_gast... (57)] | SQLQuery1.sql - WIN64-PC\...\(52) | SQLQuery1.sql - no conectado

183
184
185 --vemos que se agrego la linea de gasto
186 SELECT * FROM dbo.LineaGastoMinibar df
187 WHERE df.id_factura=7
188
189

Resultados

precio	cantidad	fecha_constatacion_m	id_lineaMinibar	id_servicio	id_producto	numero_renglon	id_factura
234.00	2	2014-09-03 00:00:00.000	99999	1	209	1	7

Consulta ejecutada correctamente.

Resultados

Lin 189 Col 9 Car 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | Explorador de objetos | Tab History | Add-ins | SQL Test | SQL Source Control | SQL Search |

Explorador de objetos | Conectar |

pruebas_lineas_de_gasto_trigge... (53)* | tr_control_miniBar_Linea_gast... (57) | SQLQuery1.sql - WIN64-PC\win64(52) | SQLQuery1.sql - no conectado

```
161 --agregamos la nueva linea de gasto de servicio asociada a la linea de factura cargada anteriormente
162 -- la linea de gasto esta fuera del rango de la reserva
163
164 INSERT INTO dbo.LineaGastoMinibar
165     ( precio ,
166       cantidad ,
167       fecha_constatacion_m ,
168       id_lineaMinibar ,
169       id_servicio ,
170       id_producto ,
171       numero_renglon ,
172       id_factura
173   )
174   VALUES ( 234 , -- precio - money
175             2 , -- cantidad - int
176             '20140103' , -- fecha_constatacion_m - datetime
177             99999 , -- id_lineaMinibar - int
178             209 , -- id_servicio - smallint
179             350 , -- id_producto - int
180             1 , -- numero_renglon - int
181             7 -- id_factura - int
182         )
183 
```

Mensajes

ALERTA:Línea de Gasto No se INSERTO, verifique la fechas de la reserva
(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

Lin 162 Col 57 Car 57 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | Explorador de objetos | Tab History | Add-ins | SQL Test | SQL Source Control | SQL Search |

Explorador de objetos | Conectar |

pruebas_lineas_de_gasto_trigge... (53)* | tr_control_miniBar_Linea_gast... (57) | SQLQuery1.sql - WIN64-PC\win64(52) | SQLQuery1.sql - no conectado

```
183
184 --vemos que solo esta la linea agregada anteriormente
185
186 SELECT * FROM dbo.LineaGastoMinibar df
187 WHERE df.id_factura=7
188
189
190 
```

Resultados

precio	cantidad	fecha_constatacion_m	id_lineaMinibar	id_servicio	id_producto	numero_renglon	id_factura
234.00	2	2014-09-03 00:00:00.000	99999	1	209	1	7

Consulta ejecutada correctamente.

Resultados

Lin 190 Col 9 Car 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar Explorador de objetos Conectar BD-TPI Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

pruebas líneas de gasto trigger... (53)* tr_control_minibar_Linea_gast... (57) SQLQuery1.sql - WIN64-PC\win64 (52) SQLQuery1.sql - no conectado

```
190 UPDATE dbo.LineaGastoMinibar SET fecha_constatacion_m='20100501' WHERE id_lineaMinibar='99999'
191
192 --tratamos de actualizar una linea de gasto de minibar en un periodo invalido de la reserva.
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
```

Mensajes

ALERTA:Línea de Gasto No se Actualizo, verifique las fechas de la reserva

(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

Lin 211 Col 4 Car 4 INS

Servidores registrados Explorador de objetos

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar Explorador de objetos Conectar BD-TPI Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

pruebas líneas de gasto trigger... (53)* tr_control_minibar_Linea_gast... (57) SQLQuery1.sql - WIN64-PC\win64 (52) SQLQuery1.sql - no conectado

```
182 )
183
184
185
186 --vemos que la linea no se actualizo
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
```

Resultados

precio	cantidad	fecha_constatacion_m	id_lineaMinibar	id_servicio	id_producto	numero_renglon	id_factura
234.00	2	2014-09-03 00:00:00.000	99999	1	209	1	7

Consulta ejecutada correctamente.

Resultados

Lin 192 Col 10 Car 4 INS

Servidores registrados Explorador de objetos



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a code editor window titled "pruebas líneas de gasto trigger... (53)". The code contains an UPDATE statement:

```
UPDATE dbo.LineaGastoMinibar SET fecha_constatacion_m='20140801' WHERE id_lineaMinibar='99999'
```

Below the code editor, the "Mensajes" (Messages) pane displays the results of the execution:

```
(1 filas afectadas)
EXITO:Línea de Gasto de minibar se ACTUALIZO, CORRECTAMENTE
(1 filas afectadas)
```

The status bar at the bottom right indicates "Lín 232 Col19 Car 3 INS".

B- Trigger de la tabla LineaGastoServicioExtra: Captura de creación del trigger.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a code editor window titled "trigger tr_control_gasto_Linea_... (54)". The code contains the creation of a trigger:

```
--drop trigger tr_control_gasto_Linea_servicio
CREATE TRIGGER tr_control_gasto_Linea_servicio ON dbo.LineaGastoServiciosExtra INSTEAD OF INSERT,UPDATE AS
IF( EXISTS(SELECT * FROM Deleted)
BEGIN --Comienza proceso de actualizacion
    IF ((SELECT ik.fecha_uso_s FROM Inserted ik) BETWEEN
        (SELECT TOP 1 ms.desde
        FROM
        (SELECT ms.fecha_desde, ms.fecha_hasta, ms.descripcion
        FROM LineaGastoMinibar ms
        WHERE ms.id_lineaMinibar = ik.id_lineaMinibar) AS ms
        ORDER BY ms.fecha_desde)) AND
        (SELECT TOP 1 ms.hasta
        FROM
        (SELECT ms.fecha_desde, ms.fecha_hasta, ms.descripcion
        FROM LineaGastoMinibar ms
        WHERE ms.id_lineaMinibar = ik.id_lineaMinibar) AS ms
        ORDER BY ms.fecha_hasta))
```

The status bar at the bottom right indicates "Lín 130 Col10 Car 4 INS".

Código del Trigger de la tabla LineaGastoServicioExtra:

```
CREATE TRIGGER tr_control_gasto_Linea_servicio ON dbo.LineaGastoServiciosExtra
INSTEAD OF INSERT, UPDATE AS
IF( EXISTS(SELECT * FROM Deleted))
```



```
BEGIN --Comienza proceso de actualizacion
IF ((SELECT ik.fecha_uso_s FROM Inserted ik) BETWEEN

(SELECT TOP 1 rs.desde
FROM ((dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
WHERE fh.id_factura = (SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc ) AND

(SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc )

)

BEGIN --exito comienzo a actualizar el el servicio- la fecha esta dentro de rango valido

UPDATE dbo.LineaGastoServiciosExtra SET

id_LineaSevicio=(SELECT ik.id_LineaSevicio FROM Inserted ik),
cantidad_serv = (SELECT ik.cantidad_serv FROM Inserted ik),
precio_serv_extra = (SELECT ik.precio_serv_extra FROM Inserted ik),
fecha_uso_s = (SELECT ik.fecha_uso_s FROM Inserted ik),
id_servicio = (SELECT ik.id_servicio FROM Inserted ik),
numero_renglon = (SELECT ik.numero_renglon FROM Inserted ik),
id_factura = (SELECT ik.id_factura FROM Inserted ik)

WHERE LineaGastoServiciosExtra.id_LineaSevicio=(SELECT ik.id_LineaSevicio FROM Inserted ik)

PRINT ('EXITO:Linea de Gasto de servicio se ACTUALIZO, CORRECTAMENTE')
END

ELSE --se detecta que el periodo ingresado de la linea de gasto de servicio esta fuera del periodo de
la reserva.
BEGIN
PRINT ('ALERTA:Linea de Gasto No se Actualizo, verifique la fechas de la reserva')
END
END --Fin de proceso de actualizacion

ELSE --Comienza proceso de insercion

BEGIN
IF( EXISTS( SELECT * FROM Inserted ik
WHERE ik.id_LineaSevicio IN (SELECT ls.id_LineaSevicio FROM dbo.LineaGastoServiciosExtra ls))
)

BEGIN
RAISERROR 24001 'ERROR-EL ID de linea ya existe en la tabla de linea_gasto_extra VERIFIQUE DATOS
INGRESADOS'
END

IF ((SELECT ik.fecha_uso_s FROM Inserted ik) BETWEEN

(SELECT TOP 1 rs.desde FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc ) AND

(SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc )

)

BEGIN --exito se procede a insertar el servicio la fecha esta dentro de rango valido

INSERT INTO dbo.LineaGastoServiciosExtra SELECT ik.* FROM inserted ik
PRINT ('EXITO:Linea de Gasto de servicio se INSERTO, CORRECTAMENTE')

```



```
END

ELSE    --se detecta conflicto: la insercion del gasto esta fuera del rango valido de la reserva.

BEGIN
PRINT ('ALERTA:Línea de Gasto No se INSERTO, verifique las fechas de la reserva')
END

END  --fin proceso de insercion
```

Capturas de prueba del trigger `tr_control_gasto_Linea_servicio`:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database structure. The central pane contains the following SQL code:

```
--creamos una linea en detalle factura para agregar la nueva linea de gasto de servicio
INSERT INTO dbo.Detalle_factura
( numero_renglon, id_factura )
VALUES ( 1, -- numero_renglon - int
        103 -- id_factura - int
)
```

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "1 filas".

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database structure. The central pane contains the following SQL code:

```
--comprobamos la fecha desde hasta que tiene esa reserva para buscar una fecha valida.
SELECT fh.id_factura, rs.id_reserva, rs.desde, rs.hasta
FROM (reserva Habitacion) rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion
INNER JOIN dbo.FacturarHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=1
```

The results pane shows a single row of data:

id_factura	id_reserva	desde	hasta
1	10	2014-05-29 15:01:27.000	2015-07-22 03:02:27.000

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "1 filas".



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | Ejecutar | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar | BD-TPI | SQL Query1.sql - WIN64-PC\win64 (54) | pruebas_lineas_de_gasto_trigge... (52) | trigger_tr_control_gasto_Linea... (58)

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos | Bases de datos del sistema | BDA | BDA2013 | BD-TPI | Diagramas de base de datos | Tablas | Tablas del sistema | dbo.CadenaHotel | dbo.Categoría | dbo.Cliente | dbo.Consumibles | dbo.CuentaUsuario | dbo.Detalle_factura | dbo.Direccion | dbo.Empresa | dbo.Estantia | dbo.FacturaHabitacion | dbo.Habitacion | dbo.Hotel | dbo.linea_habitacion | dbo.LineaGastoLlamada | dbo.LineaGastoMinibar | dbo.LineaGastoServiciosExtra | Claves | Restricciones | Desencadenadores | + control punto Linea

prueba-Gasto-Habitacion.sql - (53) | SQLQuery1.sql - WIN64-PC\win64 (54) | pruebas_lineas_de_gasto_trigge... (52) | trigger_tr_control_gasto_Linea... (58)

```
22 --agregamos la nueva linea de gasto de servicio asociada a la linea de factura cargada anteriormente
23
24 INSERT INTO dbo.LineaGastoServiciosExtra
25     ( id_LineaSevicio ,
26      cantidad_serv ,
27      precio_serv_extra ,
28      fecha_uso_s ,
29      id_servicio ,
30      numero_renglon ,
31      id_factura
32 )
33 VALUES ( '25458745' , -- id_LineaSevicio - char(10)
34          4 , -- cantidad_serv - int
35          25 , -- precio_serv_extra - money
36          '20141001' , -- fecha_uso_s - datetime --> la fecha es VALIDA esta dentro de el periodo de la reserva
37          54 , -- id_servicio - smallint
38          1 , -- numero_renglon - int
39          103 -- id_factura - int -->Esta factura debe ser de una reserva que tenga estado = 'En_Uso'
40      )
41
42
43
```

Mensajes

(1 filas afectadas)
EXITO:Línea de Gasto de servicio se INSERTO, CORRECTAMENTE

(1 filas afectadas)

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 22 Col 102 Car 102 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BDA BDA2013 BD-TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

Resultados Mensajes

prueba-Gasto-Habitacion.sql - ... (53) SQLQuery1.sql - WIN64-PC\...\(54) pruebas líneas de gasto trigger... (52) trigger tr_control_gasto_Linea... (58)

```
41
42
43 --vemos si se agrego el gasto exitosamente
44 SELECT * FROM dbo.LineaGastoServiciosExtra df
45 WHERE df.id_factura=103
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
```

Resultados

id_LineaServicio	cantidad_serv	precio_serv_extra	fecha_uso_s	id_servicio	numero_renglon	id_factura	
1	25458745	4	25.00	2014-10-01 00:00:00.000	54	1	103

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 1 filas

Resultados

Lin 61 Col 10 Car 10 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos

BDA BDA2013 BD-TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

Mensajes

prueba-Gasto-Habitacion.sql - ... (53) SQLQuery1.sql - WIN64-PC\...\(54) pruebas líneas de gasto trigger... (52) trigger tr_control_gasto_Linea... (58)

```
86
87
88 --tratamos de agregar una nueva linea para tratar de insertar un nuevo gasto.
89
90 INSERT INTO dbo.Detalle_factura
91     ( numero_renglon, id_factura )
92 VALUES ( 2, -- numero_renglon - int
93           103 -- id_factura - int
94         )
95
96
97
98
99
100
101
102
103
104
```

Mensajes

(1 filas afectadas)

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 97 Col 3 Car 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos BDA BDA2013 BD-TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

tr_control_gasto_Linea

Servidores registrados Explorador de objetos

Lista

prueba-Gasto-Habitacion.sql - (53) SQLQuery1.sql - WIN64-PC\...\...(54) pruebas líneas de gasto trigger... (52)* trigger tr_control_gasto_Linea... (58)

```
102 --tratamos de insertar una nueva linea de gasto de servicio, pero colocamos un periodo fuera del rango de la reserva
103
104
105 INSERT INTO dbo.LineaGastoServiciosExtra
106     ( id_LineaServicio ,
107     cantidad_serv ,
108     precio_serv_extra ,
109     fecha_uso_s ,
110     id_servicio ,
111     numero_renglon ,
112     id_factura
113 )
114 VALUES ( '25458745' , -- id_LineaServicio - char(10)
115           4 , -- cantidad_serv - int
116           12 , -- precio_serv_extra - money
117           '20111001' , -- fecha_uso_s - datetime    la fecha NO esta dentro de el periodo de la reserva
118           99 , -- id_servicio - smallint
119           729 , -- numero_renglon - int
120           105 -- id_factura - int
121
122
123
```

Mensajes

Mens 24001, Nivel 16, Estado 1, Procedimiento tr_control_gasto_Linea_servicio, Línea 73
ERROR-EL ID de linea ya existe en la tabla de linea_gasto_extra VERIFIQUE DATOS INGRESADOS
ALERTA:Línea de Gasto No se INSERTO, verifique las fechas de la reserva

(1 filas afectadas)

Consulta completada con errores.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin121 Col13 Car12 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

VIN64-PC\SQLEXPRESS (SQL Server 10.50.40)

Bases de datos BDA BDA2013 BD-TPI

Diagramas de base de datos

Tablas

Tablas del sistema

dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estancia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra

Columnas Claves Restricciones Desencadenadores

tr_control_gasto_Linea

Servidores registrados Explorador de objetos

Lista

prueba-Gasto-Habitacion.sql - (53) SQLQuery1.sql - WIN64-PC\...\...(54) pruebas líneas de gasto trigger... (52)* trigger tr_control_gasto_Linea... (58)

```
133 --tratamos de actualizar una linea de gasto de servicio en un periodo valido de la reserva.
134
135
136
137
138
139 UPDATE dbo.LineaGastoServiciosExtra SET fecha_uso_s='20150501' WHERE id_LineaServicio='25458745'
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
```

Mensajes

(1 filas afectadas)
EXITO:Línea de Gasto de servicio se ACTUALIZO, CORRECTAMENTE

(1 filas afectadas)

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin146 Col4 Car4 INS



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The menu bar includes Archivo, Editar, Ver, Consulta, Depurar, SQL Prompt, Herramientas, Ventana, Comunidad, Ayuda. The toolbar has icons for New Query, Execute, Save, etc. The Object Explorer on the left shows the database structure for "BD-TPI". The main pane displays a query window with the following code:

```
133 --tratamos de actualizar una linea de gasto de servicio en un periodo invalido de la reserva.
134
135
136
137
138 UPDATE dbo.LineaGastoServiciosExtra SET fecha_uso_s='20100501' WHERE id_LineaServicio='25458745'
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
```

The status bar at the bottom indicates "Lin 139 Col1 Car1 INS".

C- Trigger de la tabla LineaGastoLlamada: Captura de creación del trigger.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The menu bar includes Archivo, Editar, Ver, Consulta, Depurar, SQL Prompt, Herramientas, Ventana, Comunidad, Ayuda. The toolbar has icons for New Query, Execute, Save, etc. The Object Explorer on the left shows the database structure for "BD-TPI". The main pane displays a query window with the following code:

```
1 --drop trigger tr_control_llamada_Linea
2
3 CREATE TRIGGER tr_control_llamada_Linea ON dbo.LineaGastoLlamada INSTEAD OF INSERT,UPDATE AS
4
5 IF( EXISTS(SELECT * FROM Deleted)
6
7 BEGIN --Comienza proceso de actualizacion
8
9 IF ((SELECT l.fechaLlamada FROM Inserted ik,dbo.Llamada l
10 WHERE ik.id_llamada=l.id_llamada) BETWEEN
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
```

The status bar at the bottom indicates "Lin 3 Col1 Car1 INS".

Código del Trigger de la tabla LineaGastoLlamada:

```
CREATE TRIGGER tr_control_llamada_Linea ON dbo.LineaGastoLlamada
INSTEAD OF INSERT,UPDATE AS
IF( EXISTS(SELECT * FROM Deleted))
BEGIN --Comienza proceso de actualizacion
IF ((SELECT l.fechaLlamada FROM Inserted ik,dbo.Llamada l
WHERE ik.id_llamada=l.id_llamada) BETWEEN
(SELECT TOP 1 rs.desde
FROM
(SELECT
    MAX(rs.fechaReserva) AS desde,
    MIN(rs.hasta) AS hasta
FROM
    LineaGastoServiciosExtra
    ) rs)
```



```
WHERE ik.id_llamada=l.id_llamada) BETWEEN

(SELECT TOP 1 rs.desde FROM ((dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
WHERE fh.id_factura = ( SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc ) AND

(SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc )
)

BEGIN --exito comienzo a actualizar la linea gasto llamada, la fecha esta dentro de rango valido

UPDATE dbo.LineaGastoLlamada SET

idLineaLLamada=(SELECT ik.idLineaLLamada FROM Inserted ik),
id_llamada = (SELECT ik.id_llamada FROM Inserted ik),
id_factura = (SELECT ik.id_factura FROM Inserted ik),
numero_renglon = (SELECT ik.numero_renglon FROM Inserted ik),
precio_llamada = (SELECT ik.precio_llamada FROM Inserted ik)

WHERE LineaGastoLlamada.idLineaLLamada=(SELECT ik.idLineaLLamada FROM Inserted ik)

PRINT ('EXITO:Linea de Gasto de llamada se ACTUALIZO, CORRECTAMENTE')
END

ELSE --se detecta que el periodo ingresado de la linea de gasto de llamada esta fuera del periodo de la
reserva.

BEGIN
PRINT ('ALERTA:Linea de Gasto No se Actualizo, verifique la fechas de la reserva')
END
END --Fin de proceso de actualizacion

ELSE --Comienza proceso de insercion

BEGIN
IF( EXISTS( SELECT * FROM Inserted ik
WHERE ik.idLineaLLamada IN (SELECT ls.idLineaLLamada FROM dbo.LineaGastoLlamada ls)))
BEGIN
RAISERROR 24001 'ERROR-EL ID de linea ya existe en la tabla de linea_gasto_LLAMADA VERIFIQUE DATOS
INGRESADOS'
END

IF ((SELECT l.fechaLlamada FROM Inserted ik,dbo.Llamada l
WHERE ik.id_llamada=l.id_llamada) BETWEEN

(SELECT TOP 1 rs.desde FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik)AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc ) AND

(SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc )
)

BEGIN --exito se procede a insertar la linea de minibar, la fecha esta dentro de rango valido

INSERT INTO dbo.LineaGastoLlamada SELECT ik.* FROM inserted ik
PRINT ('EXITO:Linea de Gasto del LLAMADA se INSERTO, CORRECTAMENTE')
END

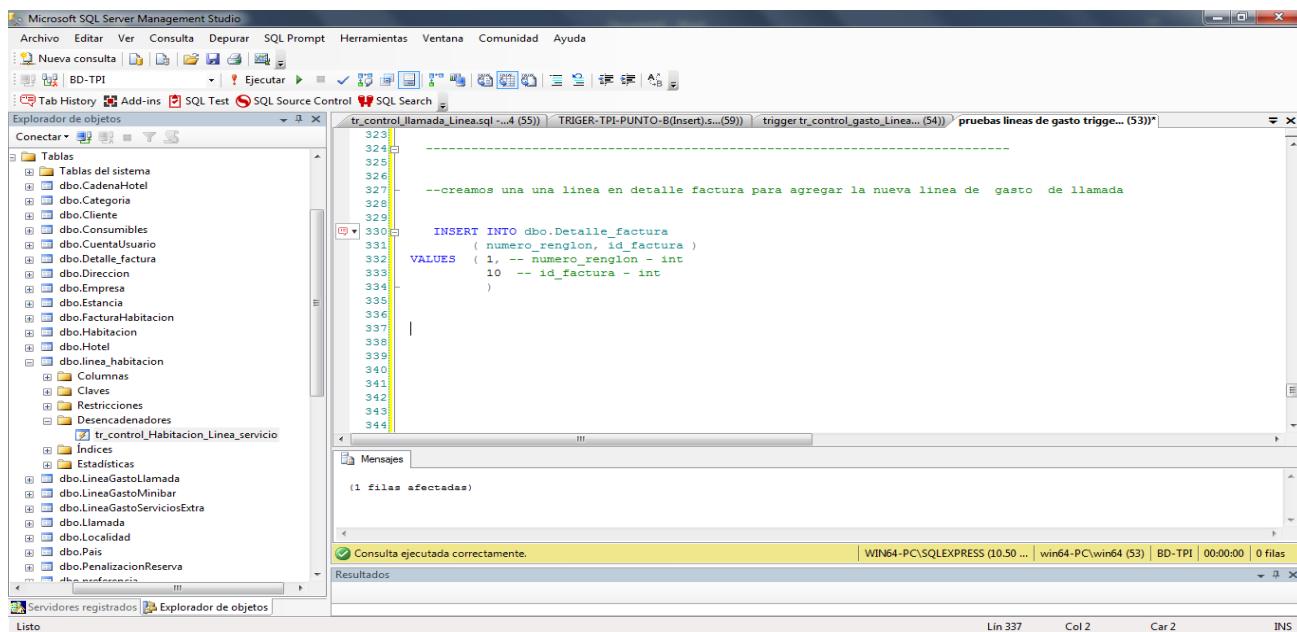
ELSE --se detecta conflicto: la insercion del gasto esta fuera del rango valido de la reserva.

BEGIN
PRINT ('ALERTA:Linea de Gasto No se INSERTO, verifique la fechas de la reserva')
```



```
END  
  
END --fin proceso de insercion
```

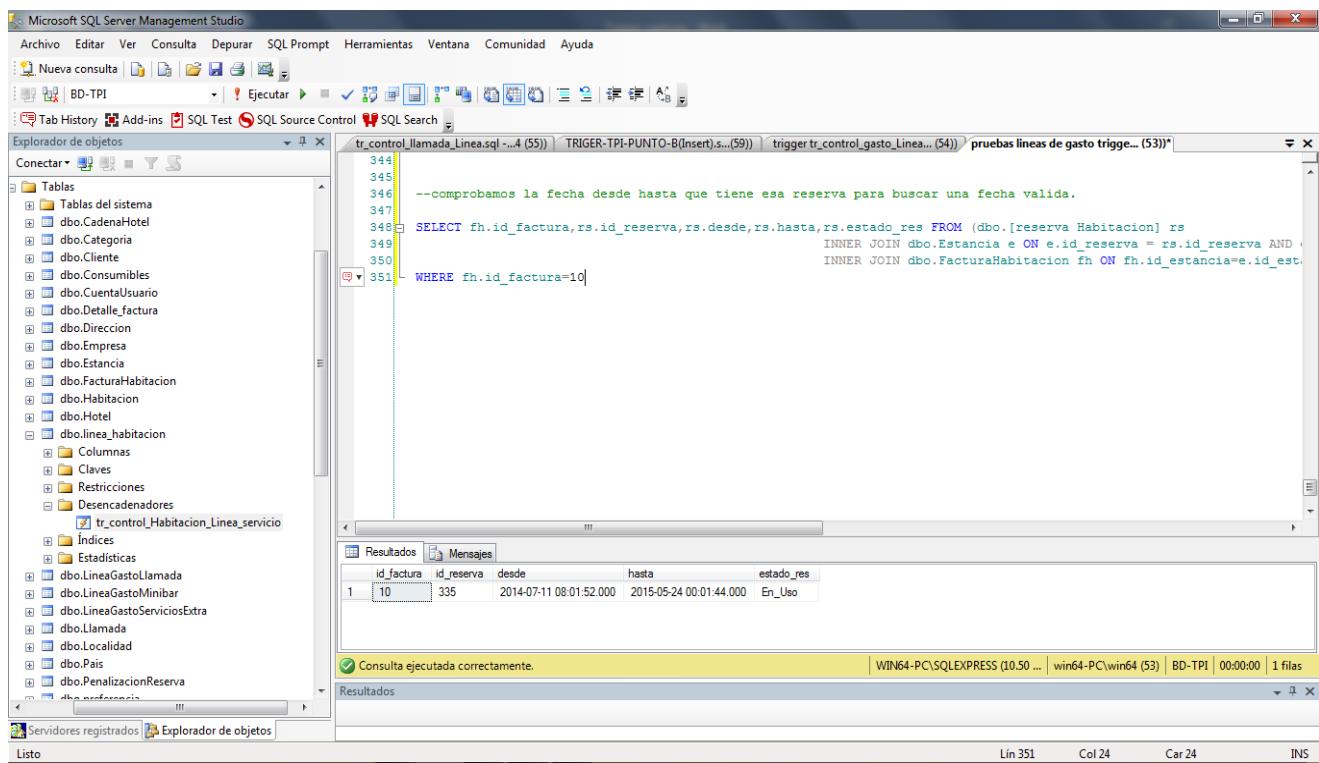
Capturas de prueba del trigger `tr_control_llamada_Linea:`



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The menu bar includes Archivo, Editar, Ver, Consulta, Depurar, SQL Prompt, Herramientas, Ventanas, Comunidad, Ayuda. The toolbar has icons for New Query, Save, Print, etc. The Object Explorer on the left shows the database structure under "Tablas" and "Desencadenadores". The central pane displays the script for creating the trigger:

```
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
  
--creamos una linea en detalle factura para agregar la nueva linea de gasto de llamada  
  
INSERT INTO dbo.Detalle_factura  
( numero_renglon, id_factura )  
VALUES ( 1, -- Numero_renglon - int  
        10 -- id_factura - int  
       )
```

The status bar at the bottom right indicates "Lin 337 Col 2 Car 2 INS".



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Microsoft SQL Server Management Studio". The menu bar includes Archivo, Editar, Ver, Consulta, Depurar, SQL Prompt, Herramientas, Ventana, Comunidad, Ayuda. The toolbar has icons for New Query, Save, Print, etc. The Object Explorer on the left shows the database structure under "Tablas" and "Desencadenadores". The central pane displays the script for verifying the trigger's logic:

```
345  
346  
347  
348  
349  
350  
351  
  
--comprobamos la fecha desde hasta que tiene esa reserva para buscar una fecha valida.  
  
SELECT fh.id_factura, rs.id_reserva, rs.desde, rs.hasta, rs.estado_res FROM (dbo.[reserva_Habitacion] rs  
INNER JOIN dbo.Establecimiento e ON e.id_reserva = rs.id_reserva AND e.id_establecimiento = fh.id_establecimiento  
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_establecimiento=e.id_establecimiento  
WHERE fh.id_factura=10)
```

The results pane shows a single row of data:

id_factura	id_reserva	desde	hasta	estado_res
1	10	2014-07-11 08:01:52.000	2015-05-24 00:01:44.000	En_Uso

The status bar at the bottom right indicates "Lin 351 Col 24 Car 24 INS".



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | SQL Test | SQL Source Control | SQL Search

Explorador de objetos

Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanzia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - Columnas
 - Claves
 - Restricciones
- Desencadenadores
 - tr_control_Habitacion_Linea_servicio
- Índices
- Estadísticas
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizacionReserva

dbo.preferences

Servidores registrados Explorador de objetos

Listo

354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375

--buscamos una llamada

```
SELECT * FROM dbo.Llamada
```

Resultados Mensajes

id_llamada	característica	fechaLlamada	duracion	Cod_interno	id_tipoLlamada
1	1751060903	5468	2014-01-04 00:00:03.000	01:38:08.8075450	004
2	634715584	0026	2014-01-05 00:00:04.000	02:23:56.3869253	5
3	226343054	6449	2014-01-06 00:00:05.000	01:43:24.9158915	167
4	1134956479	6077	2014-01-07 00:00:06.000	02:26:18.0336369	922

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 1000 filas

Resultados

Lin 373 Col 2 Car 2 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | SQL Test | SQL Source Control | SQL Search

Explorador de objetos

Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanzia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - Columnas
 - Claves
 - Restricciones
- Desencadenadores
 - tr_control_Habitacion_Linea_servicio
- Índices
- Estadísticas
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizacionReserva

dbo.preferences

Servidores registrados Explorador de objetos

Listo

375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393

--agregamos la nueva linea de gasto de llamada asociada a la llamada anterior
-- la linea de llamada esta FUERA del rango de la reserva

```
INSERT INTO dbo.LineaGastoLlamada
(
    idLineaLlamada ,
    precio_llamada ,
    id_llamada ,
    numero_renglon ,
    id_factura
)
VALUES (
    '175123' , -- idLineaLlamada - char(10)
    123, -- precio_llamada - money
    1751060903, -- id_llamada - int
    1 , -- numero_renglon - int
    10 -- id_factura - int
)
```

Mensajes

ALERTA:Línea de Gasto No se INSERTO, verifique la fechas de la reserva

(1 filas afectadas)

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 393 Col 9 Car 3 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanzia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - COLUMNAS
 - Claves
 - Restricciones
 - Desencadenadores
 - tr_control_Habitacion_Linea_servicio
- Índices
- Estadísticas

dbo.LineaGastoLlamada

dbo.LineaGastoMinibar

dbo.LineaGastoServiciosExtra

dbo.Llamada

dbo.Localidad

dbo.Pais

dbo.PenalizacionReserva

dbo.preferences

Servidores registrados Explorador de objetos

Lista

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanzia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - COLUMNAS
 - Claves
 - Restricciones
 - Desencadenadores
 - tr_control_Habitacion_Linea_servicio
- Índices
- Estadísticas

dbo.LineaGastoLlamada

dbo.LineaGastoMinibar

dbo.LineaGastoServiciosExtra

dbo.Llamada

dbo.Localidad

dbo.Pais

dbo.PenalizacionReserva

dbo.preferences

Explorador de objetos

tr_control_llamada_Linea.sql -... (45) TRIGER-TPI-PUNTO-B(Insert).s...(59) trigger tr_control_gasto_Linea... (54) pruebas líneas de gasto trigge... (53)*

```
390 )
391
392
393
394
395
396 --buscamos una llamada la 2 por ejemplo que esta dentro del rango de la reserva
397
398 SELECT * FROM dbo.Llamada
399
```

Resultados Mensajes

	id_llamada	característica	fechaLlamada	duración	Cod_interno	id_tipoLlamada
1	1751060903	5468	2014-01-04 00:00:03.000	01:38:08.8075450	004	00949
2	634715584	0026	2014-08-25 00:00:00.000	02:23:56.3869253	5	53711
3	226343054	6449	2014-01-06 00:00:05.000	01:43:24.9158915	167	18270
4	1124966178	6007	2014-01-07 00:00:00.000	02:26:18.0236269	922	02095

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 1000 filas

Resultados

Lín.2 Col.3 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estanzia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - COLUMNAS
 - Claves
 - Restricciones
 - Desencadenadores
 - tr_control_Habitacion_Linea_servicio
- Índices
- Estadísticas

dbo.LineaGastoLlamada

dbo.LineaGastoMinibar

dbo.LineaGastoServiciosExtra

dbo.Llamada

dbo.Localidad

dbo.Pais

dbo.PenalizacionReserva

dbo.preferences

Explorador de objetos

tr_control_llamada_Linea.sql -... (55)* TRIGER-TPI-PUNTO-B(Insert).s...(59) trigger tr_control_gasto_Linea... (54) pruebas líneas de gasto trigge... (53)*

```
405
406 --agregamos la nueva linea de gasto de llamada asociada a la llamada anterior
407 -- la linea de llamada esta DENTRO del rango de la reserva
408 INSERT INTO dbo.LineaGastoLlamada
409     ( idLineaLlamada ,
410       precio_llamada ,
411       id_llamada ,
412       numero_renglon ,
413       id_factura
414   )
415 VALUES ( '25467456' , -- idLineaLlamada - char(10)
416           123 , -- precio_llamada - money
417           634715584 , -- id_llamada - int
418           1 , -- numero_renglon - int
419           10 -- id_factura - int
420   )
421
```

Mensajes

(1 filas afectadas)
EXITO:Línea de Gasto del LLAMADA se INSERTO, CORRECTAMENTE

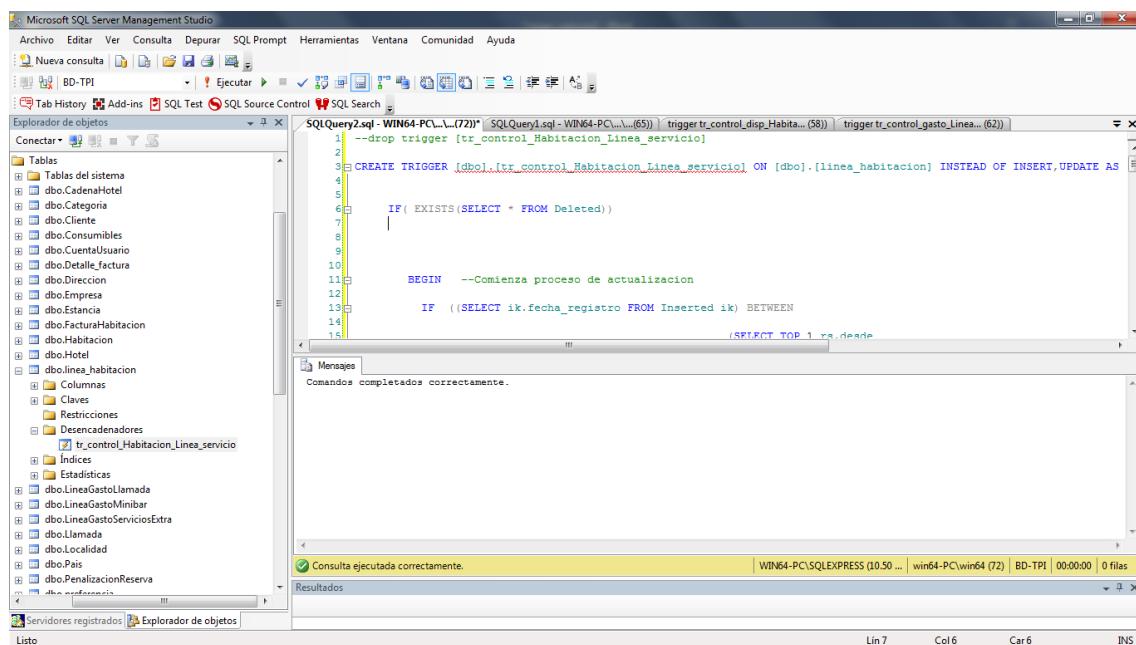
(1 filas afectadas)

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lín 421 Col 9 Car 5 INS

D- Trigger linea_habitacion: Captura de creación del trigger.



Código del Trigger de la tabla línea_habitacion:

```
CREATE TRIGGER [dbo].[tr_control_Habitacion_Linea_servicio] ON [dbo].[linea_habitacion]
INSTEAD OF INSERT,UPDATE AS

IF( EXISTS(SELECT * FROM Deleted))

BEGIN --Comienza proceso de actualizacion

IF ((SELECT ik.fecha_registro FROM Inserted ik) BETWEEN
(SELECT TOP 1 rs.desde
FROM ((dbo.[reserva Habitacion] rs INNER JOIN dbo.Estancia e
ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia )
INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura
WHERE fh.id_factura = ( SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc) AND

(SELECT TOP 1 rs.hasta FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc )

BEGIN --exito comienzo a actualizar la linea gasto minibar, la fecha esta dentro de rango valido

UPDATE dbo.linea_habitacion SET

id_reserva=(SELECT ik.id_reserva FROM Inserted ik),
cantidad_dias = (SELECT ik.cantidad_dias FROM Inserted ik),
precio_unitario= (SELECT ik.precio_unitario FROM Inserted ik),
fecha_registro = (SELECT ik.fecha_registro FROM Inserted ik),
penalizacion = (SELECT ik.penalizacion FROM Inserted ik),
id_habitacion = (SELECT ik.id_habitacion FROM Inserted ik),
numero_renglon = (SELECT ik.numero_renglon FROM Inserted ik),
id_factura=(SELECT ik.id_factura FROM Inserted ik),
id_PenalizacionDeTemp=(SELECT ik.id_PenalizacionDeTemp FROM Inserted ik)

WHERE linea_habitacion.id_reserva=(SELECT ik.id_reserva FROM Inserted ik) AND
linea_habitacion.numero_renglon=(SELECT ik.numero_renglon FROM Inserted ik) AND
linea_habitacion.id_habitacion=(SELECT ik.id_habitacion FROM Inserted ik)

PRINT ('EXITO:Linea de Gasto de habitacion se ACTUALIZO, CORRECTAMENTE')
END
```



ELSE --se detecta que el periodo ingresado de la linea de gasto de habitacion que esta fuera del periodo de la reserva.

```
BEGIN
PRINT ('ALERTA:Linea de Gasto No se Actualizo, verifique la fechas de la reserva')
END
--Fin de proceso de actualizacion

ELSE --Comienza proceso de insercion

BEGIN
IF ((SELECT ik.fecha_registro FROM Inserted ik) BETWEEN

(SELECT TOP 1 rs.desde FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.desde asc ) AND

(SELECT TOP 1 rs.hasta   FROM ( dbo.[reserva Habitacion] rs
INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva AND e.id_habitacion = rs.id_habitacion)
INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia
WHERE fh.id_factura=(SELECT ik.id_factura FROM Inserted ik) AND rs.estado_res='En_Uso'
ORDER BY rs.hasta desc ) )

BEGIN --exito se procede a insertar la linea de habitacion, la fecha esta dentro de rango valido

INSERT INTO dbo.linea_habitacion SELECT ik.* FROM inserted ik

PRINT ('EXITO:Linea de Gasto de habitacion se INSERTO, CORRECTAMENTE')

END

ELSE --se detecta conflicto: la insercion del gasto esta fuera del rango valido de la reserva.

BEGIN
PRINT ('ALERTA:Linea de Gasto No se INSERTO, verifique la fechas de la reserva')
END

END --fin proceso de insercion
```

Capturas de prueba del trigger `tr_control_Habitacion_Linea_servicio`:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with the database 'BD-TPI' selected. The right pane shows a query window with the following content:

```
TRIGER-TPL-PUNTO-8(Insert)...(59) trigger tr_control_gasto_linea...(54) pruebas lineas de gasto trigge... (53)*| tr_control_minibar_linea_gast... (57))
--creamos una linea en detalle factura para agregar la nueva linea de gasto de habitacion
234-
235-
236- INSERT INTO dbo.Detalle_factura
237-     ( numero_renglon, id_factura )
238- VALUES
239-     ( 2, -- numero_renglon - int
240-      9 -- id_factura - int
241-
242-
243-
244-
245-
246-
247-
248-
249-
250-
251-
252-
253-
254-
255-
```

The status bar at the bottom indicates: Lin 251 Col 5 Car 3 INS.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizaciónReserva
- dbo.preferencias

Desencadenadores

- tr_control_Habitacion_Linea_servicio

Índices

Estadísticas

Servidores registrados Explorador de objetos

TRIGGER-TPI-PUNTO-B(Insert).s... (59) trigger tr_control_gasto_Linea... (54) pruebas linea de gasto trigge... (53)* tr_control_minibar_Linea_gast... (57)

```
254
255
256    --comprobamos la fecha desde hasta que tiene esa reserva para buscar una fecha valida.
257
258    SELECT fh.id_factura,rs.id_reserva,rs.desde,rs.hasta,rs.estado_res FROM [dbo].[reserva_Habitacion] rs
259        INNER JOIN dbo.Estancia e ON e.id_reserva = rs.id_reserva
260        AND e.id_habitacion = rs.id_habitacion
261        INNER JOIN dbo.FacturaHabitacion fh
262            ON fh.id_estancia=e.id_estancia
263    WHERE fh.id_factura=9
264
265
```

Resultados Mensajes

	id_factura	id_reserva	desde	hasta	estado_res
1	9	747	2014-07-11 08:01:52.000	2015-08-25 12:02:52.000	En_Uso

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 1 filas

Resultados

Lin 264 Col 5 Car 2 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizaciónReserva
- dbo.preferencias

Desencadenadores

- tr_control_Habitacion_Linea_servicio

Índices

Estadísticas

Servidores registrados Explorador de objetos

TRIGGER-TPI-PUNTO-B(Insert).s... (59) trigger tr_control_gasto_Linea... (54) pruebas linea de gasto trigge... (53)* tr_control_minibar_Linea_gast... (57)

```
266    --agregamos la nueva linea de gasto de habitacion asociada a la linea de factura cargada anteriormente
267    -- la linea de gasto esta DENTRO del rango de la reserva
268    INSERT INTO dbo.linea_habitacion
269        ( id_reserva ,
270         cantidad_dias ,
271         precio_unitario ,
272         penalizacion ,
273         id_habitacion ,
274         numero_renglon ,
275         id_factura ,
276         id_PenalizacionDeTemp ,
277         fecha_registro
278     )
279     VALUES ( 747 , -- id_reserva - int
280               60 , -- cantidad_dias - int
281               2345 , -- precio_unitario - money
282               1 , -- penalizacion - money
283               '75481' , -- id_habitacion - char(10)
284               2 , -- numero_renglon - int
285               9 , -- id_factura - int
286               4323 , -- id_PenalizacionDeTemp - smallint
287               '20140815' -- fecha_registro - datetime
288   )
```

Mensajes

(1 filas afectadas)

EXITO:Línea de Gasto de habitacion se INSERTO, CORRECTAMENTE

(1 filas afectadas)

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 284 Col 40 Car 40 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

Tablas Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estanzia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion Columnas Claves Restricciones Desencadenadores tr_control_Habitacion_Linea_servicio Índices Estadísticas dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra dbo.Llamada dbo.Localidad dbo.Pais dbo.PenalizacionReserva dbo.perfomancia

TRIGER-TPI-PUNTO-B(Insert).s... (59) trigger tr_control_gasto_Linea... (54) pruebas linea de gasto trigge... (53)* tr_control_miniBar_Linea_gast... (57)

```
265
266 --agregamos la nueva linea de gasto de habitacion asociada a la linea de factura cargada anteriormente
267 -- la linea de gasto esta FUERA del rango de la reserva
268 INSERT INTO dbo.linea_habitacion
269     ( id_reserva ,
270      cantidad_dias ,
271      precio_unitario ,
272      penalizacion ,
273      id_habitacion ,
274      numero_renglon ,
275      id_factura ,
276      id_PenalizacionDeTemp ,
277      fecha_registro
278 )
279     VALUES
280     ( 747 , -- id_reserva - int
281       60 , -- cantidad_dias - int
282       2345 , -- precio_unitario - money
283       1 , -- penalizacion - money
284       '75481' , -- id_habitacion - char(10)
285       2 , -- numero_renglon - int
286       9 , -- id_factura - int
287       4323 , -- id_PenalizacionDeTemp - smallint
288       ''
289 )
290
291 --vemos que lineas se insertaron
292
293 SELECT * FROM dbo.linea_habitacion df
294 WHERE df.id_factura=9
```

Mensajes

ALERTA:Línea de Gasto No se INSERTO, verifique las fechas de la reserva
(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 0 filas

Lin 280 Col 40 Car 40 INS

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar BD-TPI

Tablas Tablas del sistema dbo.CadenaHotel dbo.Categoría dbo.Cliente dbo.Consumibles dbo.CuentaUsuario dbo.Detalle_factura dbo.Direccion dbo.Empresa dbo.Estanzia dbo.FacturaHabitacion dbo.Habitacion dbo.Hotel dbo.linea_habitacion Columnas Claves Restricciones Desencadenadores tr_control_Habitacion_Linea_servicio Índices Estadísticas dbo.LineaGastoLlamada dbo.LineaGastoMinibar dbo.LineaGastoServiciosExtra dbo.Llamada dbo.Localidad dbo.Pais dbo.PenalizacionReserva dbo.perfomancia

TRIGER-TPI-PUNTO-B(Insert).s... (59) trigger tr_control_gasto_Linea... (54) pruebas linea de gasto trigge... (53)* tr_control_miniBar_Linea_gast... (57)

```
289
290
291 --vemos que lineas se insertaron
292
293 SELECT * FROM dbo.linea_habitacion df
294 WHERE df.id_factura=9
```

Resultados

id_reserva	cantidad_dias	precio_unitario	penalizacion	id_habitacion	numero_renglon	id_factura	id_PenalizacionDeTemp	fecha_registro	
1	747	60	2345.00	1.00	75481	2	9	4323	2014-08-15 00:00:00.000

Mensajes

Consulta ejecutada correctamente.

Resultados

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (53) | BD-TPI | 00:00:00 | 1 filas

Lin 294 Col 31 Car 25 INS



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Conectar Explorador de objetos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - Columnas
 - Claves
 - Restricciones
 - Desencadenadores
 - tr_control_Habitacion_Linea_servicio
 - Indices
 - Estadísticas
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizacionReserva
- dbo.profesional

Servidores registrados Explorador de objetos

Mensajes

```
(1 filas afectadas)
EXITO:Linea de Gasto de habitacion se ACTUALIZO, CORRECTAMENTE
(1 filas afectadas)
```

Consulta ejecutada correctamente.

Resultados

Lin 313 Col 4 Car 4 INS

```
--tratamos de actualizar una linea de gasto de habitacion en un periodo valido de la reserva.

UPDATE dbo.linea_habitacion SET fecha_registro='20140901' WHERE id_habitacion='75481' AND id_reserva=747
```

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI Ejecutar Explorador de objetos Tab History Add-ins SQL Test SQL Source Control SQL Search

Conectar Explorador de objetos

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitacion
 - Columnas
 - Claves
 - Restricciones
 - Desencadenadores
 - tr_control_Habitacion_Linea_servicio
 - Indices
 - Estadísticas
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.PenalizacionReserva
- dbo.profesional

Servidores registrados Explorador de objetos

Mensajes

```
ALERTA:Linea de Gasto No se Actualizo, verifique las fechas de la reserva
(1 filas afectadas)
```

Consulta ejecutada correctamente.

Resultados

Lin 313 Col 56 Car 56 INS

```
--tratamos de actualizar una linea de gasto de habitacion en un periodo invalido de la reserva.

UPDATE dbo.linea_habitacion SET fecha_registro='20110901' WHERE id_habitacion='75481' AND id_reserva=747
```



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, a query window displays the following T-SQL code:

```
--vemos que cambios se guardaron
SELECT * FROM dbo.linea_habitacion WHERE id_habitacion='75481' AND id_reserva=747
```

The results pane shows a single row of data:

id_reserva	cantidad_dias	precio_unitario	penalizacion	id_habitacion	numero_renglon	id_factura	id_PenalizacionDeTemp	fecha_registro
747	60	2345.00	1.00	75481	2	9	4323	2014-09-01 00:00:00.000

Below the results, a message indicates the query was executed successfully.

b) Controlar que cuando se inserta o modifica una reserva que la habitación está disponible en esas fechas.

Para realizar este control se crea un trigger sobre la tabla [reserva Habitación].

Captura de creación del Trigger:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, a query window displays the creation of a trigger:

```
--drop trigger tr_control_disp_HabitacionReservada
CREATE TRIGGER tr_control_disp_HabitacionReservada ON dbo.[reserva Habitacion] INSTEAD OF INSERT,UPDATE AS
IF(NOT EXISTS(SELECT * FROM Deleted))
BEGIN
    IF ( NOT EXISTS ( SELECT * FROM
        (SELECT * FROM ( SELECT * FROM dbo.[reserva Habitacion] rs WHERE rs.id_habitacion=(SELECT ik.id_habitacion f
        where (( habitacionRes.desde BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM
        OR ( habitacionRes.hasta BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM
        (( SELECT ik.desde FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta)
        OR ( (SELECT ik.hasta FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta)) )
        WHERE reservas_En_Conflicto.estado_res='pendiente' OR reservas_En_Conflicto.estado_res='En_Uso' ))
```

The results pane shows a message indicating the command was completed correctly.

Código del Trigger `tr_control_disp_HabitacionReservada`:



```
CREATE TRIGGER tr_control_disp_HabitacionReservada ON dbo.[reserva Habitacion]
INSTEAD OF INSERT,UPDATE AS

IF(NOT EXISTS(SELECT * FROM Deleted))

BEGIN
IF ( NOT EXISTS
(SELECT * FROM (SELECT * FROM (SELECT * FROM dbo.[reserva Habitacion] rs
WHERE rs.id_habitacion= (
SELECT ik.id_habitacion
from Inserted ik)) habitacionRes
where (

(habitacionRes.desde BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM Inserted ik))OR
(habitacionRes.hasta BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM Inserted ik)))OR

(((SELECT ik.desde FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta)OR
((SELECT ik.hasta FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta)))AS
reservas_En_Conflicto
WHERE reservas_En_Conflicto.estado_res='pendiente' OR reservas_En_Conflicto.estado_res='En_Uso' ))


begin

INSERT INTO dbo.[reserva Habitacion] (id_reserva,estado_res,desde,hasta,id_habitacion)
SELECT ik.* FROM Inserted ik

DECLARE @IdRes INT = (SELECT ik.id_reserva FROM Inserted ik)
DECLARE @IdHabita CHAR(10)
SELECT @idHabita = (SELECT ik.id_habitacion FROM inserted ik)

PRINT '*****'
PRINT'Reserva Exitosa:Numero de Reserva Valido:' + CONVERT(VARCHAR(10),@IdRes )
PRINT '*****'

UPDATE dbo.Habitacion SET estado='RES' WHERE id_habitacion=@IdHabita
END
ELSE
DECLARE @IdHab CHAR(10) = (SELECT ik.id_habitacion FROM Inserted ik)

PRINT'ERROR:Reserva no efectuada:La habitacion numero: ' + CONVERT(VARCHAR(5),@IdHab ) + ' esta ocupada
durante el periodo ingresado'

END
ELSE
BEGIN

IF (EXISTS
(SELECT * FROM (SELECT * FROM (SELECT * FROM dbo.[reserva Habitacion] rs
WHERE rs.id_habitacion=
(SELECT ik.id_habitacion from Inserted ik)) habitacionRes
where (

(habitacionRes.desde BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM Inserted ik)) OR
(habitacionRes.hasta BETWEEN (SELECT ik.desde FROM Inserted ik) AND (SELECT ik.hasta FROM Inserted ik))) OR

(((SELECT ik.desde FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta) OR
((SELECT ik.hasta FROM Inserted ik) BETWEEN habitacionRes.desde AND habitacionRes.hasta)) ) AS
reservas_En_Conflicto

WHERE reservas_En_Conflicto.estado_res='pendiente' OR reservas_En_Conflicto.estado_res='En_Uso' ))


BEGIN

DECLARE @IdHabi CHAR(10) = (SELECT ik.id_habitacion FROM Inserted ik)

PRINT'ERROR:ACTUALIZACION EN CONFLICTO CON RESERVA EXISTENTE-->La habitacion numero: ' +
CONVERT(VARCHAR(5),@IdHabi ) + ' interfiere con el periodo ingresado'
end
```



```
ELSE
BEGIN
    UPDATE dbo.[reserva Habitacion] SET
        id_reserva=(SELECT ik.id_reserva FROM Inserted ik),
        id_habitacion=(SELECT ik.id_habitacion FROM Inserted ik),
        estado_res=(SELECT ik.estado_res FROM Inserted ik),
        hasta=(SELECT ik.hasta FROM Inserted ik),
        desde=(SELECT ik.desde FROM Inserted ik)

    WHERE [reserva Habitacion].id_habitacion=(SELECT ik.id_habitacion FROM Inserted ik)
        AND [reserva Habitacion].id_reserva=(SELECT ik.id_reserva FROM Inserted ik)

    DECLARE @IdRese INT = (SELECT ik.id_reserva FROM Inserted ik)

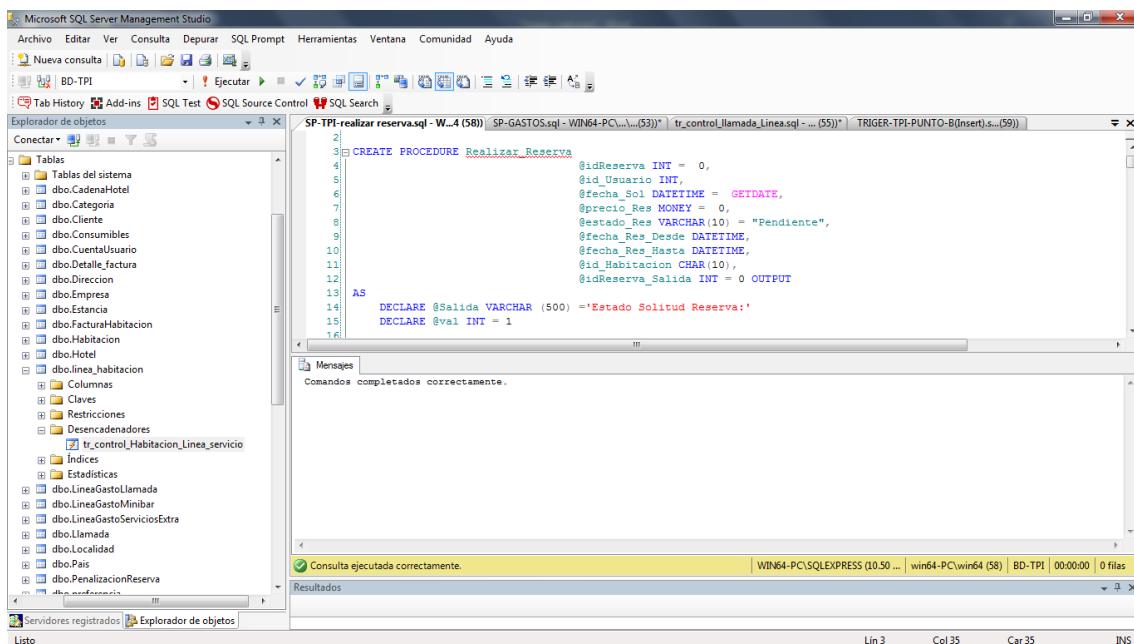
    PRINT '*****'
    PRINT 'Reserva ACTUALIZADA Exitosamente:Numero de Reserva :'+ CONVERT(VARCHAR(10),@IdRese )
    PRINT '*****'

END
end
```

Procedimientos Almacenados

a) Cree un procedimiento “Realizar_Reserva” que como dato de entradas tenga los necesarios para realizar una reserva y como dato de salida devuelva si la reserva pudo realizarse correctamente y su nro o, caso contrario, el motivo del rechazo.

Captura de creación del procedimiento almacenado “Realizar Reserva”:



Código del procedimiento almacenado “Realizar Reserva”:

```
CREATE PROCEDURE Realizar_Reserva
    @idReserva INT = 0,
```



```
@id_Usuario INT,
@fecha_Sol DATETIME = GETDATE,
@precio_Res MONEY = 0,
@estado_Res VARCHAR(10) = "Pendiente",
@fecha_Res_Desde DATETIME,
@fecha_Res_Hasta DATETIME,
@id_Habitacion CHAR(10),
@idReserva_Salida INT = 0 OUTPUT
AS

DECLARE @Salida VARCHAR (500) ='Estado Solitud Reserva:'
DECLARE @val INT = 1

IF NOT EXISTS (SELECT c.id_usuario FROM Cliente c WHERE c.id_usuario=@id_Usuario)

BEGIN
    SELECT @Salida='---ERROR- ID DE USUARIO NO EXISTE ->DEBE crearse ---'
    PRINT @Salida
    SELECT @val=0
    end

IF NOT EXISTS ( SELECT h.id_habitacion FROM dbo.Habitacion h WHERE h.id_habitacion=@id_Habitacion)

begin
    SELECT @Salida= @Salida+'---ERROR- ID DE HABITACION INEXISTENTE VERIFIQUE ---'
    PRINT @Salida
    SELECT @val=0
END

IF NOT EXISTS ( SELECT r.id_reserva FROM dbo.reserva r WHERE r.id_reserva=@idReserva)
BEGIN
    select @Salida='---- Nuevo numero Reserva asignado --Valido-- se verificará Disponibilidad---'
    PRINT @Salida
    SELECT @val=1
END

BEGIN TRANSACTION

INSERT INTO dbo.reserva( id_reserva ,fecha_solicitud_r ,precio_venta_reserva ,id_usuario)
VALUES  ( @idReserva , -- id_reserva - int
          @fecha_Sol , -- fecha_solicitud_r - datetime
          @precio_Res , -- precio_venta_reserva - money
          @id_Usuario -- id_usuario - int
        )
if @@error <> 0
    BEGIN
        PRINT @Salida
        rollback transaction
        raiserror 24001 '*****ERROR*****'
        return 0
    END
END

COMMIT TRANSACTION

End

BEGIN TRANSACTION
    INSERT INTO dbo.[reserva Habitacion]( id_reserva ,estado_res ,desde ,hasta ,id_habitacion)
    VALUES  ( @idReserva , -- id_reserva - int
              'pendiente' , -- estado_res - varchar(10)
              @fecha_Res_Desde , -- desde - datetime
              @fecha_Res_Hasta , -- hasta - datetime
              @id_Habitacion -- id_habitacion - char(10)
            )
    --los mensajes de Error/inserción real ocurren en los triggers ....si ocurren
    IF(@val=1)
        BEGIN
            select @idReserva_Salida = @idReserva
        end

```



COMMIT TRANSACTION

Capturas de las Pruebas:

En este ejemplo se mostrará un intento de inserción de una reserva de una habitación válida pero en un período en que dicha habitación se encuentra ocupada.

En las siguientes pruebas los Triggers creados en el punto anterior ya se encuentran implementados.

A continuación se muestran las capturas:

1º Captura del Mensaje de Error al intentar insertar una reserva de la habitación para un período no válido, es decir un período en el cual la habitación ya se encuentra reservada.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a SQL query window with the following code:

```
120
121
122 --Se trata de insertar una habitacion a una reserva valida pero en un periodo en el cual dicha habitacion esta ocupada
123
124 DECLARE @Numero_reserva_valido INT
125 EXEC dbo.Realizar_Reserva
126     967, -- int Reserva
127     1957, --int usuario
128     '20140126', -- datetime solicitud
129     0, -- money
130     'pendiente', -- varchar(10)
131     '20140222', -- datetime desde
132     '20150405', -- datetime hasta
133     '64234', -- char(10) numeroHabitacion
134     @Numero_reserva_valido OUTPUT
135     select @Numero_reserva_valido AS NumeroReservaAnalizado
136
```

Below the code, the results pane shows the error message:

ERROR:Reserva no efectuada:La habitacion numero: 64234 esta ocupada durante el periodo ingresado

(1 filas afectadas)

(1 filas afectadas)

At the bottom, a status bar indicates: Consulta ejecutada correctamente.

2º Captura del Resultado: en este caso se muestra el número de reserva que se quiso insertar.



The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database structure for 'WIN64-PC\SQLEXPRESS'. In the center is the 'SQLQuery1.sql' query editor window, containing the following T-SQL code:

```
--Se trata de insertar una habitacion a una reserva valida pero en un periodo en el cual dicha habitacion esta ocupada
DECLARE @Numero_reserva_valido INT
EXEC dbo.Realizar_Reserva
    967, -- int reserva
    1957, --int usuario
    '20140126', -- datetime solicitud
    0, -- money
    'pendiente', -- varchar(10)
    '20140222', -- datetime desde
    '20150405', -- datetime hasta
    '64234', -- char(10) numeroHabitacion
    @Numero_reserva_valido OUTPUT
select @Numero_reserva_valido AS NumeroReservaAnalizado
```

Below the code, the 'Resultados' tab shows the output of the query:

NumeroReservaAnalizado
967

A status bar at the bottom indicates: Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 1 filas.

3º Se realiza una consulta para corroborar si la reserva que se quiso insertar se efectivizó.

Recordemos los datos: id_reserva=967, id_habitacion='64234'.

En la captura se observa que efectivamente NO se insertó la reserva de esa habitación, pero existen otras habitaciones reservadas para ese número de reserva (recordemos que en el planteo se consideró que en una misma reserva se pueden reservar 1 o más habitaciones).

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'SQLQuery1.sql' query editor window contains the following T-SQL code:

```
--consultamos si la reserva que quisimos insertar se agrego
SELECT * FROM dbo.[reserva Habitacion] rs
WHERE rs.id_reserva=967
```

Below the code, the 'Resultados' tab shows the output of the query:

id_reserva	estado_res	desde	hasta	id_habitacion
967	pendiente	2014-02-22 00:00:00.000	2015-04-05 00:00:00.000	44332
967	incumplida	2010-03-01 00:00:00.000	2017-05-03 00:00:00.000	96555

A status bar at the bottom indicates: Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 2 filas.

4º A continuación observamos las reservas existentes para la habitación '96555', la cual cuenta únicamente con una sola reserva por lo que no debe presentar conflicto a la hora de actualizar desde/hasta:



The screenshot shows the Microsoft SQL Server Management Studio interface. The left sidebar displays the database structure for 'WIN64-PC\SQLEXPRESS'. The 'Explorador de objetos' pane shows the 'BD-TPI' database with its tables, views, and stored procedures. The main window contains a query editor with the following SQL code:

```
147 --revisamos el uso de la habitacion 96555
148
149
150
151 SELECT * FROM dbo.[reserva Habitacion] rs
152 WHERE rs.id_habitacion=96555
153
154
155
156
157
158
159
160
161
162
163
```

The results pane shows a single row from the 'reserva' table:

id_reserva	estado_res	desde	hasta	id_habitacion
1	incumplida	2010-03-01 00:00:00.000	2017-05-03 00:00:00.000	96555

The status bar at the bottom indicates: Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 1 filas.

5° Se realiza la Actualización de las fechas desde/hasta para la habitación '96555' de la reserva 967 y se observa el mensaje de Éxito en la actualización:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left sidebar displays the database structure for 'WIN64-PC\SQLEXPRESS'. The main window contains a query editor with the following SQL code:

```
156
157
158
159
160
161
162
163 --vamos a actualizar la fecha desde,hasta del uso de la habitacion como no presenta conflicto se deberia permitir la actualización
164
165 UPDATE dbo.[reserva Habitacion] SET desde='20140301', hasta='20150503' WHERE id_habitacion=96555 AND id_reserva=967
166
167
168
169
170
171
172
```

The results pane shows the message: Reserva ACTUALIZADA Exitosamente:Número de Reserva :967

The status bar at the bottom indicates: Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (52) | BD-TPI | 00:00:00 | 0 filas.

6° Se intenta realizar una inserción para una nueva reserva 72 para la habitación '64234' en un período en que la habitación se encuentra reservada y se observa el mensaje de error:



```
147 --seleccionamos la reserva 72 y tratamos de asignarle otra habitacion, en este caso la habitacion seleccionada anteriormente
148 --en un periodo que sabemos que va a causar conflicto
149
150     DECLARE @Numero_reserva_valido INT
151 EXEC dbo.Realizar_Reserva
152     72, -- int reserva
153     1957, --int usuario
154     '20140126', -- datetime solicitud
155     0, -- money
156     'pendiente', -- varchar(10)
157     '20140101', -- datetime desde
158     '20150405', -- datetime hasta
159     '64234', -- char(10) numeroHabitacion
160     @Numero_reserva_valido OUTPUT
161
162     select @Numero_reserva_valido AS NumeroReservaAnalizado
163
```

Resultados

```
ERROR:Reserva no efectuada:La habitacion numero: 64234 esta ocupada durante el periodo ingresado
(1 filas afectadas)

(1 filas afectadas)
```

Consulta ejecutada correctamente.

7º Se inserta la reserva 72 para la habitación ‘64234’ para un período que se sabe que NO va a generar conflicto y se observa el mensaje de éxito.

```
147 --seleccionamos la reserva 72 y tratamos de asignarle otra habitacion, en este caso la habitacion seleccionada anteriormente
148 --en un periodo que sabemos que NO va a causar conflicto
149
150     DECLARE @Numero_reserva_valido INT
151 EXEC dbo.Realizar_Reserva
152     72, -- int reserva
153     1957, --int usuario
154     '20140126', -- datetime solicitud
155     0, -- money
156     'pendiente', -- varchar(10)
157     '20150930', -- datetime desde
158     '20151205', -- datetime hasta
159     '64234', -- char(10) numeroHabitacion
160     @Numero_reserva_valido OUTPUT
161
162     select @Numero_reserva_valido AS NumeroReservaAnalizado
163
```

Resultados

```
(1 filas afectadas)
*****
Reserva Exitosa:Numero de Reserva Valido:72
*****

(1 filas afectadas)

(1 filas afectadas)
```

Consulta ejecutada correctamente.

8º Se corrobora la inserción de la reserva 72 para la habitación ‘64234’ y se observan las demás reservas existentes para la habitación.

Seguidamente se intentará extender el tiempo hasta de la reserva 624 para la habitación ‘64234’ en una fecha que genere conflicto con la reserva 72 para la misma habitación.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

WIN64-PC\SQLEXPRESS (BD-TPI)

Bases de datos Bases de datos del BDA BDA2013 Diagramas de l Tablas Vistas Síntomas Programación Service Broker Almacenamiento Seguridad cooperativa TestBda Seguridad Objetos de servidor Replicación Administración

SQLQuery1.sql - WIN64-PC\...\(56) | SQLQuery1.sql - no conectado | TRIGGER-TPI-PUNTO...I - no conectado | SP-TPLsql - WIN64-PC\...\win64 (52) | TriggerTPLsql - no conectado

```
164 --vemos que
165
166
167 SELECT * FROM dbo.[reserva Habitacion] rs
168 WHERE rs.id_habitacion=64234
169
170
171
172 --Vamos a tratamos de extender el periodo hasta de la reserva anterior,
173 -- de manera que quede en conflicto con la reserva cargada antes.
174
175 UPDATE dbo.[reserva Habitacion] SET hasta='20161230' WHERE id_habitacion=64234 AND id_reserva=624
176
177
178
179
180
```

Resultados Mensajes

id_reserva	estado_res	desde	hasta	id_habitacion
1	pendiente	2015-09-30 00:00:00.000	2015-12-05 00:00:00.000	64234
2	pendiente	2014-07-28 10:02:02.000	2015-03-16 06:00:54.000	64234

Consulta ejecutada correctamente.

Resultados

Lin 170 Col 6 Car 6 INS

9º Se corrobora que No se efectivizó la modificación.

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

WIN64-PC\SQLEXPRESS (BD-TPI)

Bases de datos Bases de datos del BDA BDA2013 Diagramas de l Tablas Vistas Síntomas Programación Service Broker Almacenamiento Seguridad cooperativa TestBda Seguridad Objetos de servidor Replicación Administración

SQLQuery1.sql - WIN64-PC\...\(56) | SQLQuery1.sql - no conectado | TRIGGER-TPI-PUNTO...I - no conectado | SP-TPLsql - WIN64-PC\...\win64 (52) | TriggerTPLsql - no conectado

```
170
171
172 --Vamos a tratamos de extender el periodo hasta de la reserva anterior,
173 -- de manera que quede en conflicto con la reserva cargada antes.
174
175 UPDATE dbo.[reserva Habitacion] SET hasta='20161230' WHERE id_habitacion=64234 AND id_reserva=624
176
177
178 --se produce un error y no se permite la actualización.
179
180
181 --como vemos los update y el SP de realizar_reserva son los que disparan a los triggers hechos anteriormente.
182 --los update e inserts reales se producen dentro de los triggers y son estos lo que controlan las
183 --dichas transacciones se hagan correctamente.
184
185
186
```

Mensajes

ERROR:ACTUALIZACION EN CONFLICTO CON RESERVA EXISTENTE-->La habitacion numero: 64234 interfiere con el periodo ingresado
(1 filas afectadas)

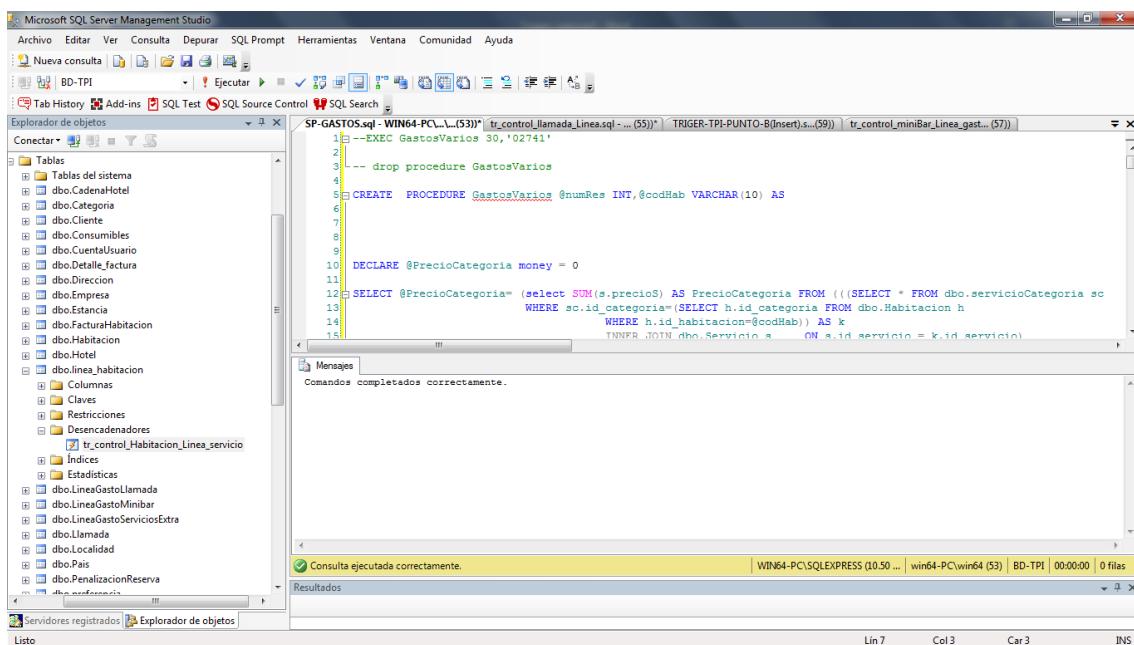
Consulta ejecutada correctamente.

Resultados

Lin 178 Col 26 Car 26 INS

b) Cree un procedimiento llamado "GastosVarios" que devuelva el gasto por uso de habitación de una reserva. Como entrada debe tener el nro de reserva y como salida un detalle de los gastos realizados en el minibar y en llamadas realizadas.

Captura de creación del procedimiento almacenado:



Código del procedimiento almacenado “Gastos Varios”:

```
CREATE PROCEDURE GastosVarios @numRes INT, @codHab VARCHAR(10) AS
DECLARE @PrecioCategoria money = 0
SELECT @PrecioCategoria= (select SUM(s.precioS) AS PrecioCategoria
FROM ((SELECT *
      FROM dbo.servicioCategoria sc
      WHERE sc.id_categoria=(SELECT h.id_categoria
                                FROM dbo.Habitacion h
                                WHERE h.id_habitacion=@codHab)) AS k
      INNER JOIN dbo.Servicio s ON s.id_servicio = k.id_servicio)
      INNER JOIN dbo.Hotel ho ON s.id_hotel = ho.id_hotel) )

DECLARE @Precio_Tipo_Habitacion MONEY = 0
SELECT @Precio_Tipo_Habitacion = th.PrecioTh
FROM dbo.Habitacion h INNER JOIN dbo.TipoHabitacion th ON th.id_TipoH = h.id_TipoH
WHERE h.id_habitacion=@codHab

DECLARE @Recargo MONEY = 0
select @Recargo = ( select TOP 1 t.porcentajeRebajaRecargo
      FROM ((dbo.temporada t INNER JOIN dbo.Hotel h ON h.id_hotel = t.id_hotel)
            INNER JOIN dbo.Habitacion ha ON h.id_hotel = t.id_hotel)
            INNER JOIN dbo.[reserva Habitacion] rs ON ha.id_habitacion=rs.id_habitacion
      WHERE rs.id_reserva=@numRes AND rs.id_habitacion=@codHab
            AND rs.desde BETWEEN t.inicio_temporada AND t.fin_temporada)

SELECT @Recargo = (ISNULL ( @Recargo , 1 ))
DECLARE @penalizacion MONEY = 0
IF(EXISTS ( SELECT * FROM dbo.[reserva Habitacion] rs
      WHERE rs.id_habitacion=@codHab AND rs.id_reserva=@numRes AND rs.estado_res='incumplida'))
begin
SELECT @penalizacion =pr.porcentaje FROM dbo.linea_habitacion lh
      INNER JOIN dbo.[reserva Habitacion] rs ON
      rs.id_reserva = lh.id_reserva AND rs.id_habitacion = lh.id_habitacion
      INNER JOIN dbo.PenalizacionReserva pr ON
      pr.id_PenalizacionDeTemp = lh.id_PenalizacionDeTemp
      WHERE lh.id_reserva=@numRes AND lh.id_habitacion=@codHab
end
```



```
DECLARE @diasReservados INT

select @diasReservados =( SELECT DATEDIFF(day,
(SELECT rs.desde FROM dbo.[reserva Habitacion] rs
WHERE rs.id_habitacion=@codHab AND rs.id_reserva=@numRes)
,
(SELECT rs.hasta hasta FROM dbo.[reserva Habitacion] rs
WHERE rs.id_habitacion=@codHab AND rs.id_reserva=@numRes) ) )

DECLARE @Total_Reserva MONEY = 0

IF (@penalizacion <> 0 )
BEGIN
    SELECT @Total_Reserva =
((@diasReservados*(@Precio_Tipo_Habitacion+@PrecioCategoria))*@Recargo)*@penalizacion
END

ELSE
BEGIN
    SELECT @Total_Reserva = ((@diasReservados*(@Precio_Tipo_Habitacion+@PrecioCategoria))*@Recargo)
END

--como los datos no son consistentes la reservas canceladas tendran datos cargados.
--Esto se podria solucionar con un trigger que no permita ninguna linea de gasto hasta que la reserva
pase a En_uso

SELECT fh.id_factura,rs.id_habitacion,@PrecioCategoria AS Categoria_Habitacion,@Precio_Tipo_Habitacion AS
TipoHabitacion,rs.id_reserva,rs.estado_res,@diasReservados AS diasReservados,@penalizacion AS
Penalizacion,@Recargo AS Recargo

FROM (dbo.Estancia e INNER JOIN dbo.[reserva Habitacion] rs ON
        rs.id_reserva = e.id_reserva AND rs.id_habitacion = e.id_habitacion)
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia = e.id_estancia
WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab

SELECT @Total_Reserva = (ISNULL ( @Total_Reserva , 0 ))

SELECT @Total_Reserva AS TOTAL_GASTO_EN_CONCEPTO_DE_ALQUILER_DE_HABITACION_DURANTE_RESERVA

DECLARE @costoTotalLlamada MONEY = 0

SELECT @costoTotalLlamada = (SELECT SUM(lg.precio_llamada)
                            FROM (((dbo.Estancia e INNER JOIN dbo.FacturaHabitacion fh ON
                                    fh.id_estancia=e.id_estancia)
                                INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
                                INNER JOIN dbo.LineaGastoLlamada lg ON lg.id_factura = df.id_factura
                                AND df.numero_renglon=lg.numero_renglon)
                            WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab)

SELECT lg.idLineaLlamada,lg.precio_llamada
FROM (((dbo.Estancia e
        INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia)
        INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
        INNER JOIN dbo.LineaGastoLlamada lg ON lg.id_factura = df.id_factura
        AND df.numero_renglon=lg.numero_renglon)
WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab

SELECT @costoTotalLlamada = (ISNULL ( @costoTotalLlamada , 0 ))

SELECT @costoTotalLlamada AS TOTAL_GASTO_EN_LLAMADA_DURANTE_RESERVA_DE_HABITACION

DECLARE @costoTotalMinibar MONEY = 0

SELECT @costoTotalMinibar = (SELECT SUM(lg.precio)
                            FROM ((dbo.Estancia e
                                INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia)
                                INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
                                INNER JOIN dbo.LineaGastoMinibar lg ON lg.id_factura = df.id_factura)
```



```
        AND df.numero_renglon=lg.numero_renglon)
WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab)

SELECT lg.id_producto AS ID_producto,c.nombreP AS nombre_Producto,lg.cantidad,lg.precio AS Precio_Unitario
FROM (((dbo.Estancia e
    INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia)
    INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
    INNER JOIN dbo.LineaGastoMinibar lg ON lg.id_factura = df.id_factura
    AND df.numero_renglon=lg.numero_renglon)
    INNER JOIN dbo.Consumibles c ON c.id_producto = lg.id_producto

WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab

SELECT @costoTotalMinibar = (ISNULL ( @costoTotalMinibar , 0 ))

SELECT @costoTotalMinibar AS TOTAL_GASTO_EN_MINIBAR_DURANTE_RESERVA_DE_HABITACION

DECLARE @costoTotalServicios MONEY = 0

SELECT @costoTotalServicios = (SELECT SUM(lg.precio_serv_extra*lg.cantidad_serv)
from (((dbo.Estancia e
    INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia)
    INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
    INNER JOIN dbo.LineaGastoServiciosExtra lg ON lg.id_factura = df.id_factura
    AND df.numero_renglon=lg.numero_renglon)
    WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab)

SELECT lg.id_servicio AS ID_servicio,s.nombreS AS nombre_Servicio ,s.precioS AS Precio_Unitario,
lg.cantidad_serv AS Cantidad_Usada

FROM (((dbo.Estancia e
    INNER JOIN dbo.FacturaHabitacion fh ON fh.id_estancia=e.id_estancia)
    INNER JOIN dbo.Detalle_factura df ON df.id_factura=fh.id_factura)
    INNER JOIN dbo.LineaGastoServiciosExtra lg ON lg.id_factura = df.id_factura
    AND df.numero_renglon=lg.numero_renglon)
    INNER JOIN servicio s ON s.id_servicio=lg.id_servicio
WHERE e.id_reserva=@numRes AND e.id_habitacion=@codHab

SELECT @costoTotalServicios = (ISNULL ( @costoTotalServicios , 0 ))

SELECT @costoTotalServicios AS TOTAL_GASTO_EN_SERVICIOS_EXTRA_EN_RESERVA_DE_HABITACION

DECLARE @totalGeneral MONEY = 0

SELECT @totalGeneral = @Total_Reserva + @costoTotalMinibar +@costoTotalLlamada + @costoTotalServicios

SELECT @totalGeneral = (ISNULL ( @totalGeneral , 0 ))

SELECT @totalGeneral AS GASTOS_TOTALES_EN_LA_HABITACION_RESERVADA
```

Captura de Ejecución del procedimiento almacenado Gastos Varios



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar | BD-TPI | SQLQuery1.sql - no conectado | TPi-TP2.sql - WIN64-PC...\win... (52)* | SP-GASTOS.sql - WIN64-PC...\win... (57)* | TRIGER-TPI-PUNT...I - no conectado

1 | EXEC GastosVarios 30, '02741'

Resultados | Mensajes

Numero_Factura	Numero_Habitacion	numeroReserva	cantidad_Dias	estado_res	Penalización	Recargo_Temporada	Precio_Tipo_Habitacion	Precio_Categoría
1	273	02741	30	360	En_Uso	0.00	1.00	650.686
								13883.4845
TOTAL_GASTO_EN_CONCEPTO_DE_ALQUILER_DE_HABITACION_DURANTE_RESERVA								
1 5232301.38								
idLineaLlamada precio_llamada								
1 77738 27.7059								
TOTAL_GASTO_EN_LLAMADA_DURANTE_RESERVA_DE_HABITACION								
1 27.7059								
ID_producto	nombre_Producto	cantidad	Precio_Unitario					
1	653	alfajor de BACABABBBBACCCBABA...	1	232.7051				
2	657	sandwich de LFMMCUTZEYKIPTTKBLUFD	4	189.4292				
3	605	whisky de ACBCABAACABABCAABB	4	168.1254				
4	495	papas Fritas de ALEKOQEIOJANGOBILR...	4	234.0924				
5	668	alfajor de BCABACCCACCBCACABA...	10	57.0227				
6	978	ensalada de fruta conCBBBCBACACCAB...	3	134.5663				
7	872	chips de EGHHIEHC	10	72.8195				
8	717	chips de GEKIDEAGB	7	55.654				
TOTAL_GASTO_EN_MINIBAR_DURANTE_RESERVA_DE_HABITACION								
1 51942.899								
ID_servicio	nombre_Servicio	Precio_Unitario	Cantidad_Usada					
1	31	Zeebanackex	168.8288	3				
2	33	Suptanplower	327.2014	4				
3	34	Gropelab	527.0805	9				
4	37	Addimar	798.3011	4				
5	35	Winnipantor	545.1757	3				
6	38	Partumistor	85.6613	1				
7	35	Winnipantor	545.1757	7				
8	30	Frodimax	326.2691	8				
TOTAL_GASTO_EN_SERVICIOS_EXTRA_EN_RESERVA_DE_HABITACION								
1 22774.4494								
GASTOS_TOTALES_EN_LA_HABITACION_RESERVADA								
1 5307046.4343								
Consulta ejecutada correctamente.								

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | BD-TPI | Ejecutar | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar | BD-TPI | SQLQuery1.sql - no conectado | TPi-TP2.sql - WIN64-PC...\win... (52)* | SP-GASTOS.sql - WIN64-PC...\win... (57)* | TRIGER-TPI-PUNT...I - no conectado

1 | EXEC GastosVarios 30, '02741'

Resultados | Mensajes

2	657	sandwich de LFMMCUTZEYKIPTTKBLUFD	4	189.4292				
3	605	whisky de ACBCABAACABABCAABB	4	168.1254				
4	495	papas Fritas de ALEKOQEIOJANGOBILR...	4	234.0924				
5	668	alfajor de BCABACCCACCBCACABA...	10	57.0227				
6	978	ensalada de fruta conCBBBCBACACCAB...	3	134.5663				
7	872	chips de EGHHIEHC	10	72.8195				
8	717	chips de GEKIDEAGB	7	55.654				
TOTAL_GASTO_EN_MINIBAR_DURANTE_RESERVA_DE_HABITACION								
1 51942.899								
ID_servicio	nombre_Servicio	Precio_Unitario	Cantidad_Usada					
1	31	Zeebanackex	168.8288	3				
2	33	Suptanplower	327.2014	4				
3	34	Gropelab	527.0805	9				
4	37	Addimar	798.3011	4				
5	35	Winnipantor	545.1757	3				
6	38	Partumistor	85.6613	1				
7	35	Winnipantor	545.1757	7				
8	30	Frodimax	326.2691	8				
TOTAL_GASTO_EN_SERVICIOS_EXTRA_EN_RESERVA_DE_HABITACION								
1 22774.4494								
GASTOS_TOTALES_EN_LA_HABITACION_RESERVADA								
1 5307046.4343								
Consulta ejecutada correctamente.								

XML

Ejercicio 1. Adicione un campo datos_adicionales del tipo XML a la tabla de clientes de manera de agregar en ellas datos semiestructurados de los mismos. Por ejemplo:



```
<?xml version="1.0" ?>
<datos adicionales>
<profesion>Doctor</profesion>
<TelPart>123-45678</TelPart>
<TelCel>15-4567890</TelCel>
<emailtrab>richal@ejemplo.com</emailtrab>
<emailpart>richal2010@hotmail.com</emailpart>
<Hobby>Futbol</Hobby>
<Hobby2>Voley</Hobby2>
</datos adicionales>
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with the 'Tablas' (Tables) node expanded, showing various database tables like 'dbo.Cliente', 'dbo.Categoria', etc. The right pane contains a query window titled 'SQLQuery1.sql - WIN64-PC\... (69)*'. The query is:

```
--Definimos un campo xml sin tipo para cliente
ALTER TABLE cliente ADD datos_adicionales xml
```

The 'Mensajes' (Messages) pane at the bottom shows the message: 'Comandos completados correctamente.' (Commands completed successfully.). The 'Resultados' (Results) pane shows a successful execution message: 'Consulta ejecutada correctamente.' (Query executed successfully.) and '0 filas' (0 rows). The status bar at the bottom indicates the command was run on 'WIN64-PC\SQLEXPRESS (10.50 ...)' at 'win64-PC\win64 (69)' with a duration of '00:00:00'.



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with a tree view of database objects, including tables like 'dbo.Cliente' and its columns. The right pane contains a SQL query window with the following code:

```
--Definimos un campo XML sin tipo para cliente
ALTER TABLE cliente ADD datos_adicionales XML

--Actualizamos un cliente con un dato XML
UPDATE cliente SET datos_adicionales='<?xml version="1.0" ?>
<datos adicionales>
<profesion>Tenista</profesion>
<TelPart>4332-4238</TelPart>
<TelCel>1423-4890</TelCel>
<emailtrab>CarlosFerro@ejemplo.com</emailtrab>
<emailpart>FershiC@hotmail.com</emailpart>
<Hobby>Tenis</Hobby>
<Hobby>Natación</Hobby2>
</datos adicionales>'
WHERE id_usuario=4271'
```

The status bar at the bottom indicates: Mensajes (1 filas afectadas), Consulta ejecutada correctamente., WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas.

Ejercicio 2. Defina un esquema XML para validar el campo anterior.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with a tree view of database objects. The right pane contains a SQL query window with the following code:

```
--Definimos un esquema XML para asociarlo al campo XML sin tipo.
CREATE XML SCHEMA COLLECTION datosAdicionales AS
N'<?xml version="1.0" encoding="UTF-16"?>
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema">
<xss:element name="datosAdicionales">
<xss:complexType>
<xss:sequence>
<xss:element name="profesion" type="xs:string"/>
<xss:element name="TelPart" type="xs:string"/>
<xss:element name="TelCel" type="xs:string"/>
<xss:element name="emailtrab" type="xs:string"/>
<xss:element name="emailpart" type="xs:string"/>
<xss:element name="Hobby" type="xs:string"/>
<xss:element name="Hobby2" type="xs:string"/>
</xss:sequence>
</xss:complexType>
</xss:element>
</xss:schema>'
```

The status bar at the bottom indicates: Mensajes (Comandos completados correctamente.), Consulta ejecutada correctamente., WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas.



```
--Actualizamos la columna xml, asignandole un esquema xsd. Esto valida el campo cargado anteriormente  
--si el xml cargado fuera 'NO VALIDO' se produciría un error al ejecutar la consulta.  
  
ALTER TABLE dbo.Cliente ALTER COLUMN datos_adicionales XML(datosAdicionales)
```

Mensajes
Comandos completados correctamente.

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas

Resultados

Ejercicio 3. Inserte para algunos clientes distintos atributos que validen el esquema del punto 2.

```
--Declaramos una variable XML que contiene un documento XML "Valido" según el esquema asociado al campo  
--definido anteriormente y actualizamos los datos adicionales de un cliente.  
  
DECLARE @varXml XML  
SELECT @varXml = '<?xml version="1.0" ?>  
    <datos adicionales>  
        <profesion>Doctor</profesion>  
        <TelPart>123-45678</TelPart>  
        <Telcel>15-4567890</Telcel>  
        <emailPart>richal@templo.com</emailPart>  
        <emailPart>richal2010@hotmail.com</emailPart>  
        <Hobby>Futbol</Hobby>  
        <Hobby>Voley</Hobby>  
    </datos adicionales>'  
  
UPDATE cliente SET datos_adicionales=@varXml WHERE id_usuario=3896
```

Mensajes
(1 filas afectadas)

Consulta ejecutada correctamente. | WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas

Resultados



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
 - Colas
 - observaciones (varchar(300), NULL)
 - id_usuario (PK, FK, int, No NULL)
 - datos_adicionales (XML(), NULL)
- Claves
- Restricciones
- Desencadenadores
- Indices
- Estatísticas
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitación
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada

Servidores registrados Explorador de objetos

Mensajes (1 filas afectadas)

SQLQuery1.sql - WIN64-PC\...\(69)* | SQLQuery1.sql - WIN64-PC\...\(56) | SQLQuery1.sql - WIN64-PC\...\(52) | SQLQuery3.sql - WIN64-PC\...\(59)*

```
--Actualizamos algunos clientes que contengan el campo (xml(datosAdicionales))

UPDATE cliente SET datos_adicionales='<?xml version="1.0" ?>
<datosadicionales>
<profesion>Tenista</profesion>
<TelPart>4332-4238</TelPart>
<TelCel>1423-4890</TelCel>
<emailtrab>CarlosFers@ejemplo.com</emailtrab>
<emailpart>Fershi@hotmail.com</emailpart>
<Hobby>Tenis</Hobby>
<Hobby2>Natación</Hobby2>
</datosadicionales>'
WHERE id_usuario=4271'
```

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 84 Col 38 Car 26 INS

Lista

13:24

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
 - Colas
 - observaciones (varchar(300), NULL)
 - id_usuario (PK, FK, int, No NULL)
 - datos_adicionales (XML(), NULL)
- Claves
- Restricciones
- Desencadenadores
- Indices
- Estatísticas
- dbo.Consumibles
- dbo.CuentaUsuario
- dbo.Detalle_factura
- dbo.Dirección
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabitacion
- dbo.Habitación
- dbo.Hotel
- dbo.linea_habitacion
- dbo.LineaGastoLlamada
- dbo.LineaGastoMinibar
- dbo.LineaGastoServiciosExtra
- dbo.Llamada

Servidores registrados Explorador de objetos

Mensajes (1 filas afectadas)

SQLQuery1.sql - WIN64-PC\...\(69)* | SQLQuery1.sql - WIN64-PC\...\(56) | SQLQuery1.sql - WIN64-PC\...\(52) | SQLQuery3.sql - WIN64-PC\...\(59)*

```
UPDATE cliente SET datos_adicionales='<?xml version="1.0" ?>
<datosadicionales>
<profesion>Futbolista</profesion>
<TelPart></TelPart>
<TelCel>4332-4238</TelCel>
<emailtrab>CarlosFers@ejemplo.com</emailtrab>
<emailpart></emailpart>
<Hobby>Futbol</Hobby>
<Hobby2></Hobby2>
</datosadicionales>'
WHERE id_usuario=4069'
```

Consulta ejecutada correctamente.

WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (69) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 105 Col 45 Car 28 INS

Lista

13:24



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is an open query window titled "SQLQuery1.sql - WIN64-PC\...\(69)*". The query is:

```
UPDATE cliente SET datos_adicionales='<?xml version="1.0" ?>
<datos adicionales>
<profesion>Empresario</profesion>
<TelPart>4332-4238</TelPart>
<TelCel></TelCel>
<emailtrab>CarlosFerss@ejemplo.com</emailtrab>
<emailpart></emailpart>
<Hobby>golf</Hobby>
<Hobby2></Hobby2>
</datos adicionales>' WHERE id_usuario=3172
```

Below the query window, the "Messages" pane shows "(1 filas afectadas)". The status bar at the bottom indicates "Consulta ejecutada correctamente." and "00:00:00 | 0 filas".

Ejercicio 4. Escriba tres instrucciones XML, que utilicen, a su elección, alguna de las siguientes funciones (una distinta para cada instrucción):

a. query()

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is an open query window titled "SQLQuery1.sql - WIN64-PC\...\(69)*". The query is:

```
select datos_adicionales.query('/datos adicionales/profesion') AS ConsularProfesiones
From Cliente
```

Below the query window, the "Resultados" pane shows the output of the query:

ConsularProfesiones
1 <profesion>Doctor</profesion>
2 <profesion>Tenista</profesion>
3 <profesion>Futbolista</profesion>
4 <profesion>Empresario</profesion>

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "00:00:00 | 1000 filas".

b. exists()



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

SQLQuery1.sql - WIN64-PC\...\(69)* SQLQuery1.sql - WIN64-PC\...\(56) SQLQuery1.sql - WIN64-PC\...\(52) SQLQuery3.sql - WIN64-PC\...\(59)*

```
SELECT datos_adicionales.exist ('/datosadicionales/TelPart[3]') AS DioTelefonoParticular
FROM Cliente
```

Resultados Mensajes

DioTelefonoParticular
0

Consulta ejecutada correctamente.

Resultados

Lin 140 Col 1 Car 1 INS

Servidores registrados Explorador de objetos

Listo

13:36

c. nodes()

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

SQLQuery1.sql - WIN64-PC\...\(69)* SQLQuery1.sql - WIN64-PC\...\(56) SQLQuery1.sql - WIN64-PC\...\(52) SQLQuery3.sql - WIN64-PC\...\(59)*

```
Select X.TelCel.value ('.', 'varchar(40)' ) as ListatelefonoCel
from Cliente CROSS APPLY
-datos_adicionales.nodes('/datosadicionales/TelCel') as X(TelCel)
```

Resultados Mensajes

ListatelefonoCel
1 154567890
2 14234890
3 43324230
4

Consulta ejecutada correctamente.

Resultados

Lin 159 Col 1 Car 1 INS

Servidores registrados Explorador de objetos

Listo

13:52

d. value()



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

SQLQuery1.sql - WIN64-PC\win64(69)* | SQLQuery1.sql - WIN64-PC\win64(56) | SQLQuery1.sql - WIN64-PC\win64(52) | SQLQuery3.sql - WIN64-PC\win64(59)*

```
SELECT datos_adicionales.value('/datosadicionales/Hobby[1]', 'varchar(30)' ) as ListaHobbiesPrincipales
from Cliente
```

Resultados Mensajes

ListaHobbiesPrincipales

1	Fútbol
2	Tenis
3	Fútbol
4	golf

Consulta ejecutada correctamente.

Resultados

Lin 153 Col 1 Car 1 INS 13:51

e. modify()

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD-TPI Ejecutar SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Bases de datos

XML-TPI.sql - WIN64-PC\win64(70)* | SQLQuery99.sql - WIN64-PC\win64(58) | SQLQuery123.sql - WIN64-PC\win64(55) | SQLQuery1111.sql - WIN64-PC\win64(51)

```
UPDATE cliente SET
    datos_adicionales.modify('replace value of (/datosadicionales/profesión)[1] with "enfermero"')
WHERE id_usuario=3896
```

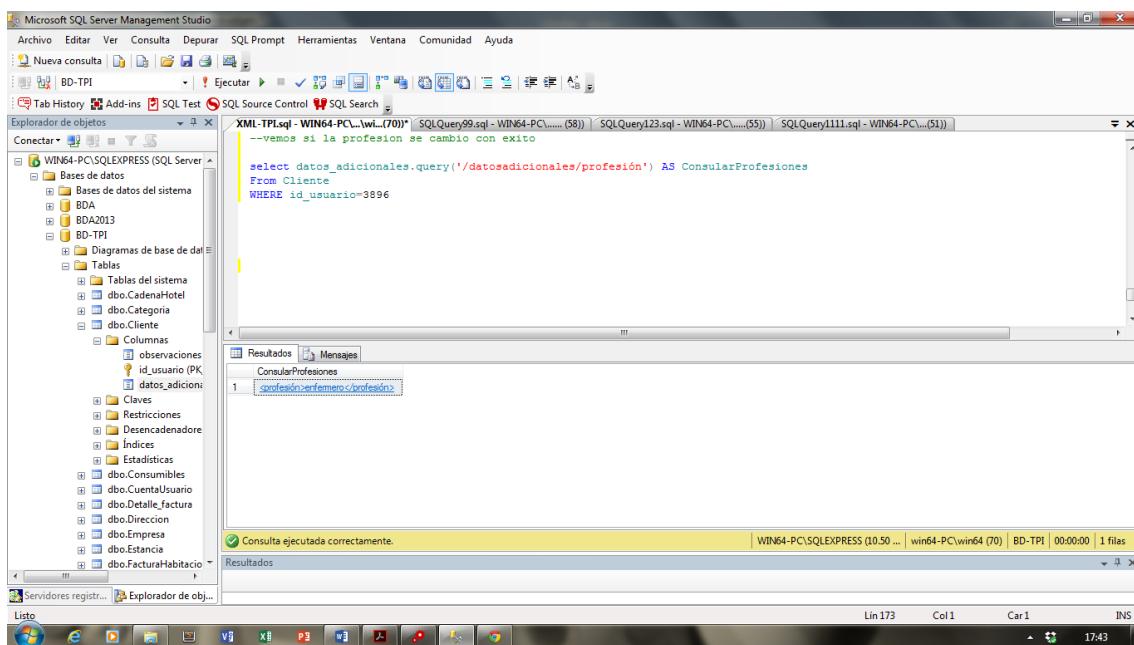
Mensajes

(1 filas afectadas)

Consulta ejecutada correctamente.

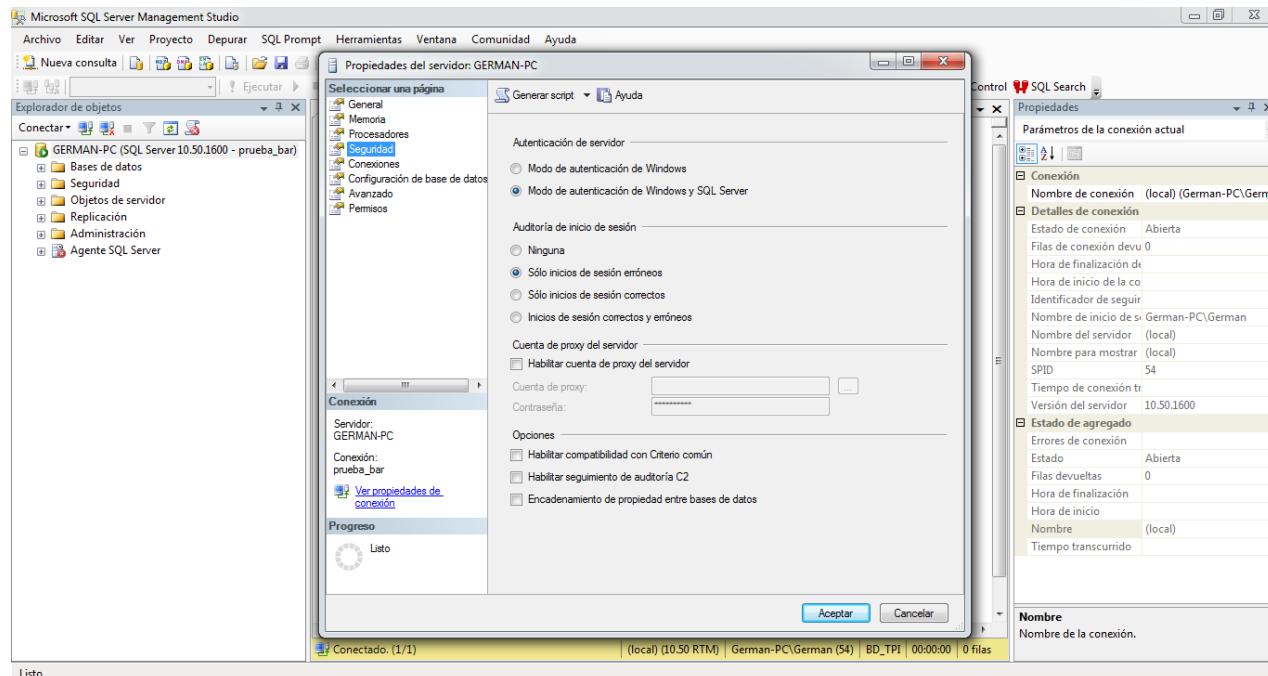
Resultados

Lin 166 Col 1 Car 1 INS 17:40



Seguridad

Primero hay que habilitar la opción de Modo de autenticación de Windows y SQL Server (clik derecho en el servidor: Propiedades -> Seguridad) debido a que la configuración inicial era solamente Modo autenticación de Windows:



Luego se deben reiniciar los servicios de SQL Server (clik derecho en el servidor: Reiniciar)

Implemente las siguientes premisas relacionadas con la seguridad en su Base de Datos:



1. Crear un inicio de sesión con nombre “pepe” y contraseña “pepe1234”, base de datos por defecto “hotel” e idioma “Spanish”. No exigir cambio de contraseña en la siguiente conexión.
No asignar ningún permiso.

```
USE BD_TPI;
GO
CREATE LOGIN pepe WITH
    PASSWORD = 'pepe1234',
    DEFAULT_DATABASE = BD_TPI,
    DEFAULT_LANGUAGE = Spanish,
    CHECK_POLICY = OFF
```

Mensajes

Comandos completados correctamente.

Consulta ejecutada correctamente. | (local) (10.50 RTM) | German-PC\German (53) | BD_TPI | 00:00:00 | 0 filas

Propiedades

Parámetros de la conexión actual

Conexión

Nombre de conexión (local) (German-PC\German)

Detalles de conexión

Estado de conexión Abierta

Filas de conexión devu. 0

Hora de finalización 07/12/2015 18:24:54

Hora de inicio de la co 07/12/2015 18:24:54

Identificador de seguir

Nombre de inicio s. German-PC\German

Nombre del servidor (local)

Nombre para mostrar (local)

SPID 53

Tiempo de conexión t 00:00:00.202804

Versión del servidor 10.50.1600

Estado de agregado

Errores de conexión

Estado Abierta

Filas devueltas 0

Hora de finalización 07/12/2015 18:24:54

Hora de inicio 07/12/2015 18:24:54

Nombre (local)

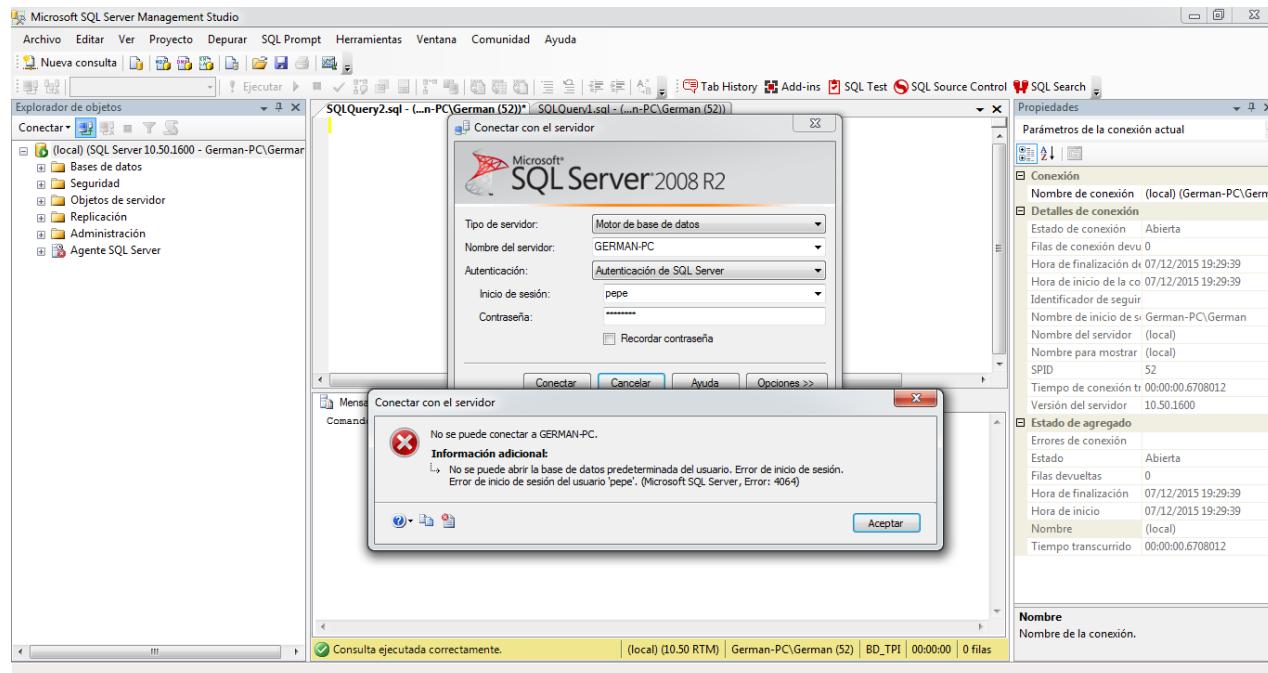
Tiempo transcurrido 00:00:00.202804

Nombre

Nombre de la conexión.

2. Conectarse con este usuario ¿qué puede hacer? ¿por qué?

No se puede realizar una conexión ya que no se creó un usuario sino que se creó un login al cual no se le asignó ningún usuario.



3. Cree el usuario de base de datos “pepe” en la base de datos “hotel” asociado al inicio de sesión “pepe”. ¿Qué puede hacer? ¿se la ha asignado algún rol a “pepe”?



Creación del usuario pepe para el login pepe:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there are two tabs: 'SQLQuery2.sql' and 'SQLQuery1.sql'. The 'SQLQuery1.sql' tab contains the following SQL code:

```
USE BD_TPI;
GO
CREATE USER pepe FOR LOGIN pepe;
```

The 'Messages' pane below the tabs displays the message: "Comandos completados correctamente." (Commands completed successfully.). The status bar at the bottom right shows: Lin 3 Col 32 Car 32 INS.

El rol asignado por defecto al usuario “pepe” es public.

Inicio de Sesión con el usuario pepe:

No permite el acceso a las Bases de Datos que no fueron asignadas por defecto al login, al intentar ingresar a una Base de Datos no asignada se genera un error:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows a connection to 'GERMAN-PC (SQL Server 10.50.1600 - pepe)'. The 'Tables' node under 'BD_TPI' is expanded, showing various tables like 'ALMACENES', 'alquileres', 'BAR', etc. The 'Messages' pane shows the message: "La base de datos DB_XML no es accesible. (ObjectExplorer)" (The database DB_XML is not accessible. (ObjectExplorer)). An 'Aceptar' (Accept) button is visible in the dialog box. The status bar at the bottom right shows: Conectado: 1/1 GERMAN-PC (10.50 RTM) | pepe (53) | BD_TPI | 00:00:00 | 0 filas.

No se pueden observar las tablas de la Base de Datos asignada por defecto, en este caso BD_TPI:



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with a tree view of the database structure under 'BD_TPI'. The right pane shows the 'Propiedades' (Properties) window for the current connection, which is named 'GERMAN-PC (pepe)'. The 'Conexión' section shows the connection is open ('Abierta'). The bottom status bar indicates 'Conectado. (1/1)'.

4. Asignarle el rol de base de datos db_datareader para la base de datos "hotel".

The screenshot shows the Microsoft SQL Server Management Studio interface. The central pane contains a query window with the following T-SQL code:

```
USE BD_TPI;
GO
EXEC sp_addrolemember 'db_datareader', 'pepe';
```

The 'Messages' pane below the query window shows the message: 'Comandos completados correctamente.' (Commands completed successfully.). The bottom status bar indicates 'Consulta ejecutada correctamente.'

5. Conectarse de nuevo con el usuario "pepe" ¿puede insertar datos en reservas? ¿y leerlos?

No puede insertar los datos en la tabla de Reservas ya que solo tiene asignado permisos de datareader, es decir solo puede leer los datos mediante consultas SELECT:



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a SQL query window with the following code:

```
1 USE [BD-TPI];
2 GO
3 INSERT INTO reserva VALUES (9999999, '20140305', 658, 3341);
```

In the 'Mensajes' (Messages) pane, an error message is displayed:

Mens. 229, Nivel 14, Estado 5, Línea 1
Se denegó el permiso INSERT en el objeto 'reserva', base de datos 'BD-TPI', esquema 'dbo'.

The status bar at the bottom right indicates: Consulta completada con errores. (Query completed with errors.)

Puede leer datos de las tablas:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a SQL query window with the following code:

```
1 USE [BD-TPI];
2 GO
3
4 SELECT TOP 50 * FROM Reserva;
```

The 'Resultados' (Results) pane displays the output of the query, which is a table of 50 rows from the 'Reserva' table. The columns are: id_reserva, fecha_solicitud_r, precio_venta_reserva, and id_usuario.

	id_reserva	fecha_solicitud_r	precio_venta_reserva	id_usuario
1	1	2012-12-19 02:02:10.500	104,5401	3896
2	2	2013-07-27 15:14:20.820	170,2256	4271
3	3	2013-01-04 05:52:22.770	283,1675	4071
4	4	2013-04-19 02:37:22.330	874,7819	3180
5	5	2012-10-24 18:31:37.600	871,659	1811
6	6	2013-06-18 14:07:27.530	643,9987	1008
7	7	2012-09-24 10:18:16.000	831,2155	4681
8	8	2012-05-01 08:51:29.270	699,5834	2708
9	9	2013-01-04 13:29:55.640	665,3891	4926
10	10	2012-07-13 23:09:22.700	670,6945	2080
11	11	2013-05-13 14:06:11.100	476,3575	2165
12	12	2013-06-18 18:20:36.950	560,4433	2826
13	13	2012-07-15 03:33:05.610	958,6626	3520
14	14	2013-02-12 21:42:41.530	171,3199	2941

The status bar at the bottom right indicates: Consulta ejecutada correctamente. (Query executed successfully.)

6. Definir un rol de bases de datos “usuariosHotel” en la BD hotel. El propietario del rol será el esquema “dbo”. Otorgarle la posibilidad de manipular los datos de las tablas y ejecutar las funciones programadas.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, a query window titled 'SQLQuery6.sql - ...PC(German (52))' contains the following T-SQL code:

```
USE BD_TPI;
GO
CREATE ROLE usuariosHotel AUTHORIZATION dbo
EXEC sp_addrolemember 'db_datawriter','usuariosHotel'
EXEC sp_addrolemember 'db_ddladmin','usuariosHotel'
```

The 'Messages' pane at the bottom displays the message: 'Comandos completados correctamente.' (Commands completed successfully.). The status bar at the bottom right indicates: 'Consulta ejecutada correctamente. | GERMAN-PC (10.50 RTM) | German-PC\German (52) | BD_TPI | 00:00:00 | 0 filas'.

7. Asignar al usuario “pepe” el rol (función de seguridad) “usuariosHotel”. Comprobar qué se le permite hacer ahora.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, a query window titled 'SQLQuery6.sql - ...PC(German (52))' contains the following T-SQL code:

```
USE BD_TPI;
GO
EXEC sp_addrolemember 'db_securityadmin','pepe'
EXEC sp_addrolemember 'usuariosHotel','pepe'
```

The 'Messages' pane at the bottom displays the message: 'Comandos completados correctamente.' (Commands completed successfully.). The status bar at the bottom right indicates: 'Consulta ejecutada correctamente. | GERMAN-PC (10.50 RTM) | German-PC\German (52) | BD_TPI | 00:00:00 | 0 filas'.

El usuario “pepe” ahora tiene los permisos del rol usuariosHotel:

db_datawriter: puede agregar, eliminar o cambiar datos en todas las tablas.

db_ddladmin: puede ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.

Y también el rol individual asignado:

db_securityadmin: puede modificar la pertenencia a roles y administrar permisos.



Microsoft SQL Server Management Studio

Explorador de objetos

SQLQuery1.sql - W...D-TPI (pepe (57)) [TPi-TP2.sql - WIN64-PC...\win... (51)] [SP-GASTOS.sql - W...D-TPI (pepe (55))] [TRIGER-TPI-PUNTO-B(Insert).s... (58)]

```
20
21| INSERT INTO reserva values (9999999, '20140305', 658, 341)
```

Mensajes

(1 filas afectadas)

Consulta ejecutada correctamente.

Resultados

Lin 21 Col 53 Car 53 INS

Microsoft SQL Server Management Studio

Explorador de objetos

SQLQuery1.sql - W...D-TPI (pepe (57)) [TPi-TP2.sql - WIN64-PC...\win... (51)] [SP-GASTOS.sql - W...D-TPI (pepe (55))] [TRIGER-TPI-PUNTO-B(Insert).s... (58)]

```
19
20
21| CREATE TABLE tablaPrueba ( id_prueba INT
, nombrePrueba VARCHAR (50) )
```

Mensajes

Comandos completados correctamente.

Consulta ejecutada correctamente.

Resultados

8. Crear el usuario “juan” y asignarle el rol “db_datawriter” y “db_datareader”. ¿Puede ejecutar funciones?

db_datawriter: puede agregar, eliminar o cambiar datos en todas las tablas.

db_datareader: puede leer todos los datos de todas las tablas.



Microsoft SQL Server Management Studio

Archivo Editor Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar SQL Source Control SQL Search

BD-TPI Tab History Add-ins SQL Test

Explorador de objetos Conectar BD-TPI

SQLQuery1.sql - WIN64-PC\...\(57)* TPI-TP2.sql - WIN64-PC\...\win...\(51)* SP-GASTOS.sql - W...D-TPI(pepe (55)) TRIGER-TPI-PUNTO-B(Insert).s... (58))

```
25 USE [BD-TPI];
26 GO
27 CREATE LOGIN juan WITH PASSWORD='juan1234',
28 DEFAULT_DATABASE=[BD-TPI],
29 DEFAULT_LANGUAGE=spanish,
30 CHECK_POLICY=OFF
31
32 CREATE USER juan FOR LOGIN juan
33
34 EXEC sp_addrolemember 'db_datareader','juan'
35 EXEC sp_addrolemember 'db_datawriter','juan'
```

Mensajes Comandos completados correctamente.

Consulta ejecutada correctamente. WIN64-PC\SQLEXPRESS (10.50 ... | win64-PC\win64 (57) | BD-TPI | 00:00:00 | 0 filas

Resultados

Lin 36 Col 46 Car 46 INS

Listo

Explorador de objetos

Servidores registrados Explorador de objetos

Puede ejecutar algunas funciones de sistema:

Microsoft SQL Server Management Studio

Archivo Editor Ver Consulta Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar SQL Source Control SQL Search

BD-TPI Tab History Add-ins SQL Test

Explorador de objetos Conectar BD-TPI

SQLQuery1.sql - W...D-TPI(juan (53)) TPI-TP2.sql - WIN64-PC\...\win...\(51)* SP-GASTOS.sql - W...D-TPI(pepe (55)) TRIGER-TPI-PUNTO-B(Insert).s... (58))

```
37
38
39 SELECT GETDATE()
40
41
```

Resultados Mensajes (Sin nombre de columna)

1	2015-12-07 21:10:15.740
---	-------------------------

Consulta ejecutada correctamente. WIN64-PC\SQLEXPRESS (10.50 ... | juan (53) | BD-TPI | 00:00:00 | 1 filas

Resultados

Lin 41 Col 1 Car 1 INS

Listo

Explorador de objetos

No puede ejecutar procedimientos almacenados:



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query window with the following code:

```
40
41 EXEC sp_calcular_precio 30, '02741'
```

Below the code, the 'Mensajes' (Messages) pane displays an error message:

Mens 229, Nivel 14, Estado 5, Procedimiento 'sp_calcular_precio', Línea 1
Se denegó el permiso EXECUTE en el objeto 'sp_calcular_precio', base de datos 'BD-TPI', esquema 'dbo'.

The status bar at the bottom indicates: Consulta completada con errores. WIN64-PC\SQLEXPRESS (10.50 ... | juan (53) | BD-TPI | 00:00:00 | 0 filas

9. Modificar el rol “usuariosHotel” dejándole únicamente la posibilidad de gestionar las reservas y clientes, y consultar el resto de información.

The screenshot shows the Microsoft SQL Server Management Studio interface. On the right, a script window contains the following T-SQL code:

```
USE BD_TPI;
GO
REVOKE INSERT,DELETE TO usuariosHotel

GRANT SELECT,INSERT,DELETE ON dbo.Cliente TO usuariosHotel
GRANT SELECT,INSERT,DELETE ON dbo.Empresa TO usuariosHotel
GRANT SELECT,INSERT,DELETE ON dbo.Turista TO usuariosHotel
GRANT SELECT,INSERT,DELETE ON dbo.reserva TO usuariosHotel
GRANT SELECT,INSERT,DELETE ON dbo.[reserva Habitacion] TO usuariosHotel
```

The 'Mensajes' (Messages) pane shows the output: Comandos completados correctamente.

The 'Propiedades' (Properties) pane on the right shows the current connection details:

- Nombre de conexión: GERMAN-PC (German-PC)
- Estado de conexión: Abierta
- Filas devueltas: 0
- Hora de finalización: 07/12/2015 21:26:17
- Hora de inicio: 07/12/2015 21:26:16
- Identificador de seguir:
- Nombre de inicio de s: German-PC\German
- Nombre del servidor: GERMAN-PC
- Nombre para mostrar: GERMAN-PC
- SPID: 52
- Tiempo de conexión tr: 00:00:00.3120006
- Versión del servidor: 10.50.1600

10. Conectarse como “pepe” y tratar de insertar un nuevo servicio. ¿Se puede?

El usuario “pepe” puede insertar un nuevo Servicio ya que la alteración de los permisos al rol usuariosHotel no se hizo en cascada, esto quiere decir que los nuevos usuarios que se creen con el rol usuariosHotel contaran con los privilegios nuevos del rol, pero los que ya fueron creados con el rol usuariosHotel mantienen los permisos originales. En caso de



querer modificar los permisos de todos los usuarios que tengan el rol usuariosHotel, se debería haber realizado un REVOKE INSERT, DELETE TO usuariosHotel CASCADE.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a query editor window titled "SQLQuery1.sql - W...D-TPI (pepe (57))". The query is:

```
53 INSERT INTO Servicio (nombreS,descripcionS,precioS,id_hotel) VALUES ('masajes','lindo masaje',876,1)
54
55
56
57
```

Below the query editor, the "Mensajes" (Messages) pane shows the result: "(1 filas afectadas)". The status bar at the bottom right indicates "Lin 57 Col 1 Car 1 INS".

11. Hacerlo con la cuenta de “administrador”. Borrar el rol “usuariosHotel” ¿Se puede?

No se puede eliminar el rol debido a que el rol tiene miembros y debe estar vacío antes de quitarlo.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center pane, there is a query editor window titled "SQLQuery8.sql - ...PC\German (52)". The query is:

```
USE BD_TPI;
GO
DROP ROLE usuariosHotel;
```

The "Mensajes" (Messages) pane shows the error message: "Mens. 15144, Nivel 16, Estado 1, Línea 2 El rol tiene miembros. Debe estar vacío antes de quitarlo." (Role has members. It must be empty before it can be dropped.)

The status bar at the bottom right indicates "Lin 4 Col 24 Car 24 INS".



Backup

Ejercicio 1: Estrategia de Backup

Semana 1	12 hs	1	1	1	1	2	2	2
	20 hs	3	3	3	3	4	4	4
	22 hs				5			

■ Diferencial
■ Full semanal
■ Full mensual

Semana 2	12 hs	6	6	6	6	7	7	7
	20 hs	8	8	8	8	9	9	9
	22 hs				10			

Semana 3	12 hs	11	11	11	11	12	12	12
	20 hs	13	13	13	13	14	14	14
	22 hs				15			

Semana 4	12 hs	16	16	16	16	17	17	17
	20 hs	18	18	18	18	19	19	19
	22 hs				20			21

Se diseña una estrategia de backup que consiste en realizar:

- Dos backups de log diarios, uno al mediodía (12 hs) y el otro a las 20 hs.
- Un backup full semanal los días viernes a las 22 hs.
- Un backup full mensual los días domingos correspondientes a la 4ta semana, este backup es el destinado a ser llevado lejos del sector de sistemas.

Se considera además un total de 21 cintas para realizar la planificación mensual, por lo cual para el tiempo de 3 meses se necesitarían 63 cintas.

Además al implementar la estrategia de usar 2 cintas por día (por ejemplo el primer lunes se usa la cinta 1 a las 12hs y la cinta 3 a las 20hs), nos brinda la posibilidad de tener un día en 2 cintas diferentes (por supuesto que la cinta de las 12 solo tendrá la mitad de la actividad del día y la cinta de las 20 la actividad casi completa del día)

Se opta por hacer backup diferenciales todos los días, ya que de esta manera tenemos en cada hora detallada siempre una copia completa desde el último backup full, por lo que si falla algún medio, en el otro medio tenemos la copia inmediata anterior. Es decir que si fallase la cinta 3 el día martes de la primera semana, podríamos restaurar la base de datos hasta ese día al mediodía, ya que ese respaldo se encuentra en la cinta 1.

Todos los viernes se realiza la copia full a las 22. Se elige este método por 2 motivos: uno para que este respaldo sea independiente (no como el diferencial que depende de un backup full) y el segundo motivo es para ir reduciendo el incremento del tamaño de los backups diferenciales que ocurren en el transcurso de la semana.

Cabe aclarar que antes de ejecutar el primer backup diferencial se debe realizar un backup full para que este pueda funcionar.



Ejercicio 2:

- A) 1) Como se aclara en el punto anterior, para poder empezar con los backups diferenciales, es necesario primero un backup full. Por lo cual se crea el primer backup full.

The screenshot shows the SQL Server Management Studio interface with two query panes. The top pane contains T-SQL code to create a full backup of the 'BD_TPI' database to a disk file named 'BDTPIFull'. The bottom pane shows the results of the execution, indicating that 312 pages were processed for the data file and 2 pages for the log file, totaling 314 pages processed in 0.319 seconds.

```
USE BD_TPI
alter database BD_TPI
set recovery full;

declare @nombreFull nvarchar(200)
set @nombreFull=N'D:\Backup\BDTPIFull'+ CONVERT (nvarchar(20),SUBSTRING(convert(varchar(20),GETDATE(), 120),0,14))+N'.bak'

BACKUP DATABASE BD_TPI TO DISK =@nombreFull
```

100 % < >

Resultados

Se han procesado 312 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
BACKUP DATABASE procesó correctamente 314 páginas en 0.319 segundos (7.667 MB/s).

100 % < >

Consulta ejecutada correctamente. | ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | BD_TPI | 00:00:00 | 0 filas

- 2) La creación de los backups diferenciales se crean de la siguiente manera.

The screenshot shows the SQL Server Management Studio interface with two query panes. The top pane contains T-SQL code to create a differential backup of the 'BD_TPI' database to a disk file named 'BDTPIDif'. The bottom pane shows the results of the execution, indicating that 40 pages were processed for the data file and 1 page for the log file, totaling 41 pages processed in 0.233 seconds.

```
USE BD_TPI
declare @nombreDif nvarchar(200)
set @nombreDif=N'D:\Backup\BDTPIDif'+ CONVERT (nvarchar(20),SUBSTRING(convert(varchar(20),GETDATE(), 120),0,14))+N'.bak'

BACKUP DATABASE BD_TPI TO DISK=@nombreDif WITH DIFFERENTIAL
```

100 % < >

Resultados

Se han procesado 40 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 1 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
BACKUP DATABASE WITH DIFFERENTIAL procesó correctamente 41 páginas en 0.233 segundos (1.374 MB/s).

100 % < >

Consulta ejecutada correctamente. | ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | BD_TPI | 00:00:00 | 0 filas

- 3) El backup full de los días viernes se realiza de la siguiente manera

The screenshot shows the SQL Server Management Studio interface with two query panes. The top pane contains T-SQL code to create a full backup of the 'BD_TPI' database to a disk file named 'BDTPIFull'. The bottom pane shows the results of the execution, indicating that 312 pages were processed for the data file and 2 pages for the log file, totaling 314 pages processed in 0.597 seconds.

```
USE BD_TPI
declare @nombreFull nvarchar(200)
set @nombreFull=N'D:\Backup\BDTPIFull'+ CONVERT (nvarchar(20),SUBSTRING(convert(varchar(20),GETDATE(), 120),0,14))+N'.bak'

BACKUP DATABASE BD_TPI TO DISK =@nombreFull
```

100 % < >

Resultados

Se han procesado 312 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
BACKUP DATABASE procesó correctamente 314 páginas en 0.597 segundos (4.096 MB/s).

100 % < >

Consulta ejecutada correctamente. | ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | BD_TPI | 00:00:01 | 0 filas



Ejercicio 3

- a) El procedimiento para la restauración en el dia jueves a las 11 consiste en:
 - 1) Al estar en modo full la base de datos, primero se debe realizar un backup del log en el caso de ser posible

```
USE BD_TPI
CHECKPOINT

USE master
BACKUP LOG BD_TPI
TO DISK= 'D:\Backup\BDTPILog.bak'
WITH NORECOVERY
```

Resultados

Se han procesado 5 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 4.
BACKUP LOG procesó correctamente 5 páginas en 0.077 segundos (0.469 MB/s).

Consulta ejecutada correctamente.

ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | master | 00:00:00 | 0 filas

- 2) Luego habría que restaurar el backup full correspondiente al viernes anterior

```
USE master
RESTORE DATABASE BD_TPI
FROM DISK='D:\Backup\BDTPIFullViernesAnterior.bak'
WITH NORECOVERY
```

Resultados

Se han procesado 312 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
RESTORE DATABASE procesó correctamente 314 páginas en 0.369 segundos (6.628 MB/s).

Consulta ejecutada correctamente.

ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | master | 00:00:03 | 0 filas

- 3) Luego hay que restaurar el diferencial del día miércoles a las 20 hs

```
USE master
RESTORE DATABASE BD_TPI
FROM DISK='D:\Backup\BDTPIDiffMiércole20hs.bak'
WITH NORECOVERY
```

Resultados

Se han procesado 40 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 1 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
RESTORE DATABASE procesó correctamente 41 páginas en 0.137 segundos (2.338 MB/s).

Consulta ejecutada correctamente.

ERIC_NET\SQLEXPRESS (11.0 RTM) | sa (52) | master | 00:00:01 | 0 filas



- 4) Por ultimo al estar en modo full y al haber backupeado el log antes de iniciar el proceso de recuperación, se debe restaurar el mismo

The screenshot shows a SQL query window titled "SQLQuery1.sql - ERI...ESS.master (sa (52))". The query is:

```
USE master
RESTORE LOG BD_TPI
FROM DISK='D:\Backup\BDTPILog.bak'
WITH RECOVERY
```

The results pane below the query shows the output of the restore command:

```
Se han procesado 0 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 26 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
RESTORE LOG procesó correctamente 26 páginas en 0.157 segundos (1.290 MB/s).
```

At the bottom of the results pane, there is a green checkmark icon and the message "Consulta ejecutada correctamente." (Query executed successfully).

Con este procedimiento podríamos recuperar la base hasta el último momento de fallo ya que, en el caso de no perderse el log, podríamos rescatarlo y llevar la base hasta el momento anterior al fallo. En el caso de que no se disponga de este log, solo se podría llegar hasta el miércoles a las 20hs.

- b) Si en lugar de 3 meses se debieran conservar los datos por 6 años se deberá aplicar iterativamente la estrategia adoptada para los 3 meses. Esto implica una inversión adicional en Hardware debido a que se debe realizar la compra de los dispositivos de almacenamiento por cada iteración de 3 meses.

Ejercicio 4:

- a) Copiar los ficheros de base de datos utilizando la opción detach (Separar...) que permiten desvincular los ficheros de la BD del servidor. Copiar los ficheros (*.mdf, *.ldf) que por defecto residen en C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA y guardarlos en un disco de almacenamiento seguro. Luego vincular la BD de nuevo con la opción attach (Adjuntar...)



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar | Propiedades | Parámetros de la conexión |

master | Ejecutar | Tab History | Add-ins | SQL Test | SQL Source Control | SQL Search

SQLQuery6.sql - G...n-PC\German (63) | SQLQuery5.sql - G...n-PC\German (59) | SQLQuery4.sql - G...n-PC\German (60) | SQLQuery3.sql - G...n-PC\German (55)

GERMAN-PC (SQL Server 10.50.1600) | Bases de datos | Instantáneas de bases de datos | ALMACENES | alquileres | BAR | BD_TPI | DB_XML | DBA | DBA_Practica | Jurasik_Park | PRUEBA | VIDEOCLUB | Seguridad | Objetos de servidor | Replicación | Administración | Agente SQL Server

```
USE [master]
GO
ALTER DATABASE [BD_TPI] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO
USE [master]
GO
EXEC master.dbo.sp_detach_db @dbname = N'BD_TPI', @skipchecks = 'false'
GO
```

Mensajes

Actualizando [dbo].[Cliente]
[PK6], no es necesaria su actualización...
Se actualizaron 0 índices o estadísticas; no fue necesario actualizar 1.

Actualizando [dbo].[Consumibles]
Se actualizó [PK67]...
[_WA_Sys_00000003_0317E3D], no es necesaria su actualización...
[ix_consumible_nombre], no es necesaria su actualización...
Se actualizaron 1 índices o estadísticas; no fue necesario actualizar 2.

Actualizando [dbo].[CuentaUsuario]
[PK2], no es necesaria su actualización...
[_WA_Sys_00000006_0519CEAF], no es necesaria su actualización...
Se actualizaron 0 índices o estadísticas; no fue necesario actualizar 2.

Actualizando [dbo].[Detalle_factura]
[PK75], no es necesaria su actualización...
[_WA_Sys_00000002_07020F21], no es necesaria su actualización...
Se actualizaron 0 índices o estadísticas; no fue necesario actualizar 2.

Actualizando [dbo].[Direccion]
[PK3], no es necesaria su actualización...
Se actualizaron 0 índices o estadísticas; no fue necesario actualizar 1.

Consulta ejecutada correctamente.

GERMAN-PC (10.50 RTM) | German-PC\German (63) | master | 00:00:00 | 0 filas

Línea 9 Columna 1 Columna 1 Índice

Estado | Estado de la conexión.

Equipo > Disco local (C:) > Program Files > Microsoft SQL Server > MSSQL10_50.MSSQLSERVER > MSSQL > DATA > Lugar Seguro

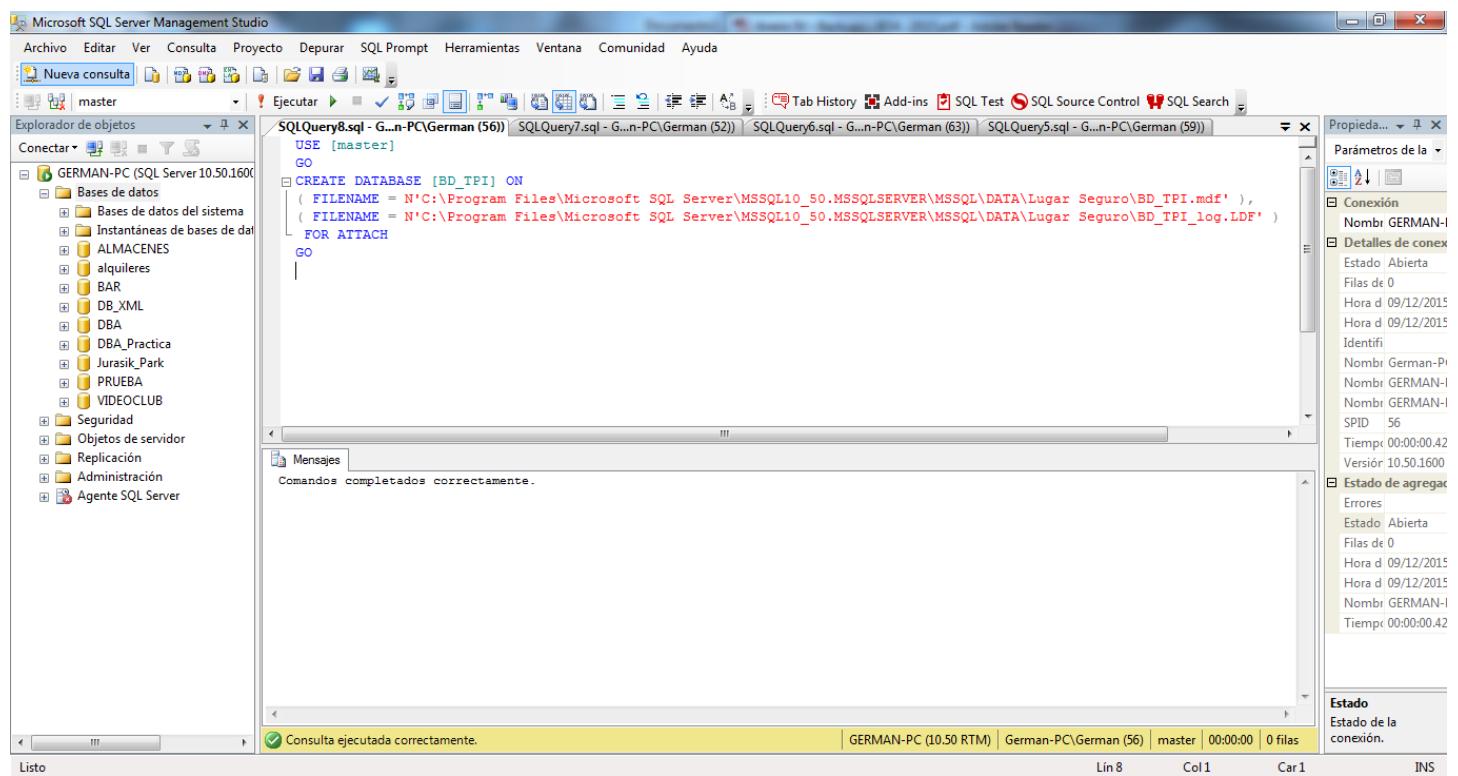
Organizar | Abrir con... | Grabar | Nueva carpeta | Buscar Lugar Seguro

Favoritos | Descargas | Escritorio | Sitios recientes | Bibliotecas | Documentos | Imágenes | Música | Videos | Equipo | Disco local (C:) | Nuevo vol (E:) | Red

Nombre | Fecha de modifica... | Tipo | Tamaño

BD_TPI	09/12/2015 16:31	SQL Server Database	10.496 KB
BD_TPI_log	09/12/2015 16:31	SQL Server Log	9.408 KB

2 elementos seleccionados Fecha de modifica... 09/12/2015 16:31 Fecha de creación: 09/12/2015 16:33 Tamaño: 19,4 MB



- b) Hacer una copia de seguridad completa con la opción Copia de seguridad que ofrece el MS SQLServer Management Studio. Crear una tabla cualquiera. A continuación, borrar la base de datos y restaurarla del backup. La tabla creada entre la copia y la restauración no debe aparecer.



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta BD_TPI Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

GERMAN-PC (SQL Server 10.50.1600)

- Bases de datos
 - Bases de datos del sistema
 - Instantáneas de bases de datos
 - ALMACENES
 - alquileres
 - BAR
 - BD_TPI
 - DB.XML
 - DBA
 - DBA_Practica
 - Jurasik_Park
 - PRUEBA
 - VIDEOCLUB
- Seguridad
- Objetos de servidor
- Replicación
- Administración
- Agente SQL Server

SQLQuery9.sql - ...PC(German (62)) SQLQuery8.sql - G...n-PC(German (56)) SQLQuery7.sql - G...n-PC(German (52)) SQLQuery6.sql - G...n-PC(German (63))

```
USE BD_TPI;
GO
BACKUP DATABASE BD_TPI TO DISK ='C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\Backups\TPI.bak'
```

Mensajes

Se han procesado 1312 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
BACKUP DATABASE procesó correctamente 1314 páginas en 1.017 segundos (10.088 MB/s).

Consulta ejecutada correctamente.

GERMAN-PC (10.50 RTM) | German-PC\German (62) | BD_TPI | 00:00:01 | 0 filas

Lin 3 Col 116 Car 116

Microsoft SQL Server Management Studio

Archivo Editar Ver Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta Ejecutar Tab History Add-ins SQL Test SQL Source Control SQL Search

Explorador de objetos Conectar

Tablas

- Tablas del sistema
- dbo.CadenaHotel
- dbo.Categoría
- dbo.Cliente
- dbo.Consumibles
- dbo.CuentaUser
- dbo.Detalle_factu
- dbo.Direccion
- dbo.Empresa
- dbo.Estancia
- dbo.FacturaHabit
- dbo.Habitacion
- dbo.Hotel
- dbo.linea_habitac
- dbo.LineaGastoLl
- dbo.LineaGastoM
- dbo.LineaGastoSe
- dbo.Llamada
- dbo.Localidad
- dbo.Pais
- dbo.Penalizacionf
- dbo.preferencia
- dbo.prueba
- dbo.reserva
- dbo.reserva Habit
- dbo.Servicio
- dbo.servicioCateg
- dbo.servicioMinit
- dbo.TelefonoHot

SQLQuery9.sql - ...PC(German (62))* SQLQuery8.sql - G...n-PC(German (56)) SQLQuery7.sql - G...n-PC(German (52)) SQLQuery6.sql - G...n-PC(German (63))

```
USE BD_TPI;
GO
CREATE TABLE prueba(id_prueba int )
```

Mensajes

Comandos completados correctamente.

Consulta ejecutada correctamente.

GERMAN-PC (10.50 RTM) | German-PC\German (62) | BD_TPI | 00:00:00 | 0 filas



Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | Ejecutar | Tab History | Add-ins | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar |

SQLQuery9.sql - ...PC\German (62)) | SQLQuery8.sql - G...n-PC\German (56) | SQLQuery7.sql - G...n-PC\German (52) | SQLQuery6.sql - G...n-PC\German (63)

```
USE master;
GO
DROP DATABASE BD_TPI
```

Mensajes

Comandos completados correctamente.

Consulta ejecutada correctamente. | GERMAN-PC (10.50 RTM) | German-PC\German (62) | master | 00:00:00 | 0 filas

Lín 1 Col 11 Car 11

Microsoft SQL Server Management Studio

Archivo Editar Ver Consulta Proyecto Depurar SQL Prompt Herramientas Ventana Comunidad Ayuda

Nueva consulta | Ejecutar | Tab History | Add-ins | SQL Test | SQL Source Control | SQL Search

Explorador de objetos | Conectar |

SQLQuery9.sql - ...PC\German (62)) | SQLQuery8.sql - G...n-PC\German (56) | SQLQuery7.sql - G...n-PC\German (52) | SQLQuery6.sql - G...n-PC\German (63)

```
USE master;
GO
RESTORE DATABASE BD_TPI FROM DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\Backups\TPI.bak'
```

Mensajes

Se han procesado 1312 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'BD_TPI', archivo 'BD_TPI_log' en el archivo 1.
RESTORE DATABASE procesó correctamente 1314 páginas en 0.548 segundos (18.723 MB/s).

Consulta ejecutada correctamente. | GERMAN-PC (10.50 RTM) | German-PC\German (62) | master | 00:00:03 | 0 filas

Lín 3 Col 125 Car 125