

EVALUACION	Obligatorio 2	GRUPO	TODOS	FECHA	
MATERIA	Base de datos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la Información				
CONDICIONES	<p>Puntos: Máximo: 25 Mínimo: 0</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF.</p> <p>IMPORTANTE:</p> <ul style="list-style-type: none"> - Inscribirse - Formar grupos de hasta dos personas del mismo dictado. - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO" - Defensa Obligatoria: La fecha será informada oportunamente por el docente. 				

El Segundo obligatorio tiene como punto de partida la misma realidad planteada para el Obligatorio 1, y en base al siguiente modelo relacional provisto por los docentes:

```
create table Pais(
    IdPais decimal not null Primary Key,
    NombrePais varchar(50));

create table Pasajero (
    IdPasajero decimal not null Primary Key,
    NombrePasajero varchar(60),
    APaternoPasajero varchar(60),
    AMaternoPasajero varchar(60),
    TipoDocumentoPasajero char(3),
    NumeroDocumentoPasajero decimal,
    FechaNacimientoPasajero date,
    EmailPasajero varchar(60),
    PasswordPasajero character(40),
    IdPais decimal not null foreign key references Pais(IdPais));

create table PasajeroTelefono (
    IdPasajero decimal not null foreign key references Pasajero (IdPasajero),
    TelefonoPasajero varchar(60),
    Primary Key (IdPasajero, TelefonoPasajero));

create table Terminal (
    IdTerminal decimal not null Primary Key,
    NombreTerminal varchar(60),
    IdPais decimal not null foreign key references Pais(IdPais));
```

```
create table bus (  
    Idbus decimal not null Primary Key,  
    Tipobus varchar(60),  
    Capacidadbus integer,  
    Marcabus varchar(80));  
  
create table Asiento (  
    Idbus decimal not null foreign key references bus (Idbus),  
    FilaAsiento integer,  
    LetraAsiento char(2),  
    Primary Key (Idbus,FilaAsiento,LetraAsiento));  
  
create table Viaje (  
    IdTerminalOrigenViaje decimal not null foreign key references Terminal  
(IdTerminal),  
    IdTerminalDestinoViaje decimal not null foreign key references Terminal  
(IdTerminal),  
    FechaHoraViaje datetime,  
    Importe float,  
    Idbus decimal foreign key references bus (Idbus),  
    IdViaje decimal not null Primary Key);  
  
create table Boleto (  
    IdPasajero decimal not null foreign key references Pasajero (IdPasajero),  
    Idbus decimal not null foreign key references bus (Idbus),  
    FilaAsiento integer,  
    LetraAsiento char(2),  
    FechaCompraBoleto date,  
    IdViaje decimal not null foreign key references Viaje (IdViaje),  
    Foreign key (Idbus,FilaAsiento,LetraAsiento) references  
Asiento(Idbus,FilaAsiento,LetraAsiento));  
  
create table PasajeroMercoSur (  
    IdPasajero decimal not null foreign key references Pasajero (IdPasajero),  
    PorcentajePasajeroMercoSur float,  
    Primary Key (IdPasajero));  
  
create table PasajeroMercoSurBeneficio (  
    IdPasajero decimal not null foreign key references Pasajero (IdPasajero),  
    BeneficioMercoSur varchar(80),  
    Primary Key (IdPasajero,BeneficioMercoSur));
```

En base al modelo relacional provisto por los docentes (DDL publicado en la pestaña Evaluaciones) se solicita la implementación de las siguientes consultas en SQL:

1. Listar el o los nombres de los pasajeros con la mayor cantidad de pasajes comprados a su nombre.
2. Listar todos los datos de los buses con más de 35 asientos que no tengan asignado ningún destino que parta el día de mañana.
3. Listar todos los datos de los pasajeros para los cuales haya registrados en el sistema más de 5 pasajes comprados.
4. Listar idpasajero, nombre, apellidos y asiento (idasiento y fila) que correspondan a pasajes comprados para el destino cuyo idviaje es 255.
5. Listar todos los iddestino y cantidad de pasajes comprados durante el mes de Setiembre de este año para c/u de los destinos del pasajero cuyo correo es soyturista@gmail.com.

La lista debe estar ordenada por iddestino ascendente.

Se pide:

- Sentencias de Lenguaje de Manipulación de Datos en SQL para responder las consultas enumeradas previamente con las correspondientes sentencias SQL que permitan cargar el juego de datos con el que se realizaron las pruebas previas a la entrega.
- Evidencia de ejecución de las consultas sobre la base de datos brindada por el cuerpo docente. (script será publicado en aulas en la pestaña Evaluaciones).

NOTA: Las consultas deben ser compatibles con MSSQL 2012 o superior.
La versión utilizada en el curso es MSSQL 2019

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

➤ Obligatorios

La entrega de los obligatorios será en formato digital online.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde gestion.ort.edu.uy
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. **Sugerimos realizarlo con anticipación.**
3. **Uno de los integrantes del grupo de obligatorio será el administrador del mismo** y es quien formará el equipo y subirá la entrega
4. Cada equipo (2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener **un tamaño máximo de 40mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'**.
7. **La hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.