

EVALUACIÓN	Obligatorio	GRUPOS	todos	FECHA	Agosto 2024
MATERIA	Bases de Datos 2				
CARRERA	Analista en Tecnologías de Información / Analista Programador				
CONDICIONES	<p>- Puntaje máximo: 40 puntos - Puntaje mínimo: 1 punto - Fecha de entrega: 25/11/2024 hasta las 21:00 horas en gestion.ort.edu.uy (max. 40Mb en formato zip, rar o pdf)</p> <p>Uso de material de apoyo y/o consulta</p> <p><u>Inteligencia Artificial Generativa</u></p> <ul style="list-style-type: none"> - Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso. - Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó. - Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla. - Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de “alucinaciones”, los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso - En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema <p>IMPORTANTE:</p> <ol style="list-style-type: none"> 1) Inscribirse 2) Formar grupos de hasta 2 personas del mismo dictado 3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: “RECORDATORIO”) <p>Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails alamon@ort.edu.uy y fernandez_ma@ort.edu.uy, o telefónicamente al 29021505 - int 1156 u 1138</p>				

La realidad de este obligatorio hace referencia a un sistema de gestión para un Hotel de Gatos.



A continuación, se lista un modelo de base de datos simplificado (se adjunta el script completo en un anexo).

Propietario (*propietarioDocumento*, propietarioNombre, propietarioTelefono, propietarioEmail)

Los propietarios de los gatos están identificados con un documento y siempre se conoce su nombre. El teléfono y el correo electrónico no pueden ser nulos simultáneamente

Gato (*gatoID*, gatoNombre, gatoRaza, gatoEdad, gatoPeso, propietarioDocumento)

Cada gato tiene un identificador autonumérico y siempre se conoce tanto su nombre como su propietario. Se almacena también raza, edad, y peso.

Habitacion (*habitacionNombre*, habitacionCapacidad, habitacionPrecio, habitacionEstado)

Las habitaciones del hotel están identificadas por un nombre y tienen una capacidad máxima de cantidad de gatos, el precio standard por estancia diaria (en dólares) y un estado que indica si la habitación está disponible, llena, o en proceso de limpieza.

Reserva (*reservaID*, gatoID, habitacionNombre, reservaFechaInicio, reservaFechaFin, reservaMonto)

Las reservas realizadas por los propietarios para sus gatos en el hotel tienen un identificador autonumérico y siempre se conoce estos datos: la identificación del gato hospedado, la habitación asignada, las fechas de inicio y fin de la reserva, y el monto en dólares (el cual no incluye los servicios adicionales que eventualmente puede contratar el dueño)

Servicio (*servicioNombre*, servicioPrecio)

Son los servicios adicionales que ofrece el hotel y pueden ser contratados dentro de una reserva, ejemplos: alimentación especial, peluquería, juguetes, etc. Cada servicio está identificado por un nombre y siempre se conoce su precio en dólares.

Reserva_Servicio (reservaID, servicioID, cantidad)

Relaciona las reservas con los servicios adicionales solicitados durante la estancia para un gato en el hotel. Almacena la cantidad de veces que un servicio específico ha sido solicitado en una reserva en particular.

Se Pide:

1. Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso).
2. Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos que la cantidad).
3. Utilizando SQL implementar las siguientes consultas:
 - a. Mostrar el nombre del gato, el nombre del propietario, la habitación y el monto de la reserva más reciente en la(s) habitación con la capacidad más alta
 - b. Mostrar los 3 servicios más solicitados, con su nombre, precio y cantidad total solicitada en el año anterior. Solo listar el servicio si cumple que tiene una cantidad total solicitada mayor o igual que 5
 - c. Listar nombre de gato y nombre de habitación para las reservas que tienen asociados todos los servicios adicionales disponibles
 - d. Listar monto total de reserva por año y por gato (nombre) para los gatos que tienen más de 10 años de edad, son de raza "Persa" y que en el año tuvieron montos total de reserva superior a 500 dólares.
 - e. Mostrar el ranking de reservas más caras, tomando como monto total de una reserva el monto propio de la reserva más los servicios adicionales contratados en la reserva
 - f. Calcular el promedio de duración en días de las reservas realizadas durante el año en curso. Deben ser consideradas solo aquellas reservas en las que se contrató el servicio "CONTROL_PARASITOS" pero no se contrató el servicio "REVISION_VETERINARIA"
 - g. Para cada habitación, listar su nombre, la cantidad de días que ha estado ocupada y la cantidad de días transcurridos desde la fecha de inicio de la primera reserva en el hotel. Además, incluir una columna adicional que indique la categoría de rentabilidad, asignando el valor "REDITUABLE" si la habitación estuvo ocupada más del 60% de los días, "MAGRO" si estuvo ocupada entre el 40% y el 60%, y "NOESNEGOCIO" si estuvo ocupada menos del 40%.
4. Utilizando T-SQL realizar los siguientes ejercicios:
 - a. Escribir un procedimiento almacenado para reservar una habitación.
Se debe actualizar el estado de DISPONIBLE a LLENA si se alcanzó la capacidad de la habitación con la reserva en cuestión
No permitir realizar la reserva si el estado de la habitación es LLENA o LIMPIANDO.
Se debe retornar el número de reserva asignado (cero sino se logró reservar)
 - b. Mediante una función que reciba un nombre de servicio, devolver un booleano indicando si este año el servicio fue contratado más veces que el año pasado

5. Escribir los siguientes disparadores (por supuesto: considerando modificaciones múltiples)
 - a. Cada vez que se crea una nueva reserva se debe crear un registro de auditoria con todos los datos ingresados en una tabla ReservaLog (definir su estructura libremente). Y adicionalmente cada vez que se modifica el campo monto de una reserva: debe registrar monto previo y nuevo monto en la tabla ReservaLog.
En todos los casos se debe grabar fecha-hora de registro, usuario(login), nombre de equipo desde el que se realizó la modificación.
 - b. Antes de insertar una nueva reserva, se debe controlar posibles solapamientos de reservas (un gato no podría estar alojado simultáneamente 2 veces en el hotel).
Se debe dar de alta las reservas válidas y simplemente ignorar las reservas solapadas
6. Crear una vista que liste el monto total a facturar por propietario por las reservas y servicios del mes pasado. Se debe listar el nombre del propietario, el monto total de sus reservas, el monto total de servicios adicionales que contrató y la suma de ambos montos (monto a facturar)
7. Se quiere utilizar las capacidades de MongoDB para registrar la misma información del Hotel que en el esquema relacional, con la diferencia que se debe soportar almacenar atributos de los gatos de forma dinámica en el sentido que no se conoce de antemano los atributos(Para un gato se podría querer almacenar el Color y el Carácter, para otro gato la Lista de vacunas y sus Fechas de vencimiento, para otro guardar una Foto del gato en una estadía, etc.)
 - A. Cree la(s) colecciones de documentos JSON para almacenar una reserva y que luego se pueda consultar de forma óptima todos los datos de una reserva (habitación, servicios adicionales, gato, dueño, etc.). Oriente su diseño a performance de lectura como objetivo principal por encima de los criterios de integridad de datos que nos tiene acostumbrados el modelo relacional
 - B. Listar reservas del propietario con documento "12345678"
 - C. Listar las reservas que incluyen el servicio "PELUQUERIA "
 - D. Actualizar el estado de la habitación "Suite1" asegurándose que está en estado "DISPONIBLE" y pasándolo a estado "LLENA"
 - E. Listar nombre de propietario y cantidad de reservas con fecha de inicio en julio 2024, para los propietarios que tengan más de una reserva en ese mes

Se debe entregar:

- a. Script con índices, y el ingreso de datos de prueba.
- b. Script con la resolución de las consultas.
- c. Script con la resolución de los procedimientos y funciones
- d. Script con los disparadores
- e. Script con la creación de la vista
- f. Script de MongoDB con la creación de la colección y las consultas

Consideraciones generales:

1. Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.
2. La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. **Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.**
3. Durante la última semana los docentes **no** contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.

Anexo 1 - Script de creación de tablas (puede descargar el archivo aparte)


```
CREATE TABLE Propietario (  
    propietarioDocumento CHAR(30) PRIMARY KEY,  
    propietarioNombre VARCHAR(100) NOT NULL,  
    propietarioTelefono VARCHAR(20) NULL,  
    propietarioEmail VARCHAR(100) NULL,  
    CONSTRAINT CHK_Propietario_TelefonoEmail CHECK (propietarioTelefono IS NOT NULL OR propietarioEmail  
IS NOT NULL) );  
  
CREATE TABLE Gato (  
    gatoID INT IDENTITY(1,1) PRIMARY KEY,  
    gatoNombre VARCHAR(50) NOT NULL,  
    gatoRaza VARCHAR(50),  
    gatoEdad INT,  
    gatoPeso DECIMAL(5,2),  
    propietarioDocumento CHAR(30) NOT NULL,  
    CONSTRAINT CHK_Gato_Edad CHECK (gatoEdad >= 0),  
    CONSTRAINT CHK_Gato_Peso CHECK (gatoPeso > 0),  
    CONSTRAINT FK_Gato_Propietario FOREIGN KEY (propietarioDocumento) REFERENCES  
Propietario(propietarioDocumento) );  
  
CREATE TABLE Habitacion (  
    habitacionNombre CHAR(30) PRIMARY KEY,  
    habitacionCapacidad INT,  
    habitacionPrecio DECIMAL(6,2),  
    habitacionEstado VARCHAR(20),  
    CONSTRAINT CHK_Habitacion_Capacidad CHECK (habitacionCapacidad > 0),  
    CONSTRAINT CHK_Habitacion_Precio CHECK (habitacionPrecio > 0),  
    CONSTRAINT CHK_Habitacion_Estado CHECK (habitacionEstado IN ('DISPONIBLE', 'LLENA', 'LIMPIANDO')) );  
  
CREATE TABLE Reserva (  
    reservaID INT IDENTITY(1,1) PRIMARY KEY,  
    gatoID INT NOT NULL,  
    habitacionNombre CHAR(30) NOT NULL,  
    reservaFechaInicio DATE NOT NULL,  
    reservaFechaFin DATE NOT NULL,  
    reservaMonto DECIMAL(7,2) NOT NULL,  
    CONSTRAINT FK_Reserva_Gato FOREIGN KEY (gatoID) REFERENCES Gato(gatoID),  
    CONSTRAINT FK_Reserva_Habitacion FOREIGN KEY (habitacionNombre) REFERENCES  
Habitacion(habitacionNombre),  
    CONSTRAINT CHK_Reserva_Fecha CHECK (reservaFechaFin > reservaFechaInicio) );  
  
CREATE TABLE Servicio (  
    servicioNombre CHAR(30) NOT NULL PRIMARY KEY,  
    servicioPrecio DECIMAL(7,2),  
    CONSTRAINT CHK_Servicio_Precio CHECK (servicioPrecio >= 0) );  
  
CREATE TABLE Reserva_Servicio (  
    reservaID INT NOT NULL,  
    servicioNombre CHAR(30) NOT NULL,  
    cantidad INT DEFAULT 1,  
    PRIMARY KEY (reservaID, servicioNombre),  
    CONSTRAINT CHK_ReservaServicio_Cantidad CHECK (cantidad > 0),  
    CONSTRAINT FK_ReservaServicio_Reserva FOREIGN KEY (reservaID) REFERENCES Reserva(reservaID),  
    CONSTRAINT FK_ReservaServicio_Servicio FOREIGN KEY (servicioNombre) REFERENCES  
Servicio(servicioNombre) );
```

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

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

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

Cualquier integrante podrá:

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 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). El archivo a subir debe tener **un tamaño máximo de 40mb**

Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. **La hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con la Coordinadora o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails, alamon@ort.edu.uy y fernandez_ma@ort.edu.uy, o telefónicamente al 29021505 - int 1156 u 1138.