

Trabajo 2. PL/SQL

Alquiler de vehículos

1. Descripción de la práctica

Se presenta el caso de uso de una empresa que gestiona reservas de vehículos. Para este caso de uso trabajaremos con las siguientes tablas:

1. **Precio_Combustible** (*Tipo_Combustible*, *precio_por_litro*)
Es una tabla de dos filas una con el precio actual de la gasolina y otra con el del diésel.
2. **Modelos** (*id_modelo* (PK), *nombre*, *precio_cada_dia*, *capacidad_deposito*, *tipo_combustible* (FK → *Precio_Combustible*)
3. **Vehículos** (*matricula* (PK), *id_modelo* (FK → *Modelos*), *color*, *alquiladoA* (FK → *Clientes*)
El campo alquilado puede estar a nulo si el vehículo está disponible.
4. **Clientes** (*NIF* (PK), *nombre*, *ape1*, *ape2*, *dirección*)
5. **Reservas** (*idReserva* (PK), *cliente* (FK → *Clientes*), *matricula* (FK → *Vehículos*), *fecha_ini*, *fecha_fin*)
6. **Facturas** (*nroFactura* (PK), *importe*, *cliente* (FK → *Clientes*)).
7. **Lineas_Factura** (*NroFactura* (FK → *Facturas*), *concepto*, *importe*)
[PK compuesta *NroFactura* + *Concepto*]

Además, existen las siguientes secuencias:

Nombre de la secuencia	Alimenta los valores de
seq_modelos	La clave primaria de la tabla modelos: <i>id_modelo</i>
seq_num_fact	La clave primaria de la tabla de facturas: <i>nroFactura</i>
seq_reservas	La clave primaria de la tabla de reservas: <i>idReserva</i>

Para facilitar la tarea se facilita el script **alquilerCoches2022_enun.sql** junto a este enunciado, que además de crear las tablas previamente descritas contiene:

1. El procedimiento almacenado **reset_seq(p_seq_name varchar)** que sirve para resetear las secuencias, similar al utilizado en prácticas anteriores.
2. Un procedimiento almacenado **inicializa_test** que reinicia el contenido de la base de datos y la provee de unas filas de prueba (véase que *inicializa_test* llama a *reset_seq*).
3. Un procedimiento almacenado con los tests automáticos **test_alquila_coches** que llama a *inicializa_test* cada vez que prueba un caso.

En este trabajo se pide la implementación de una transacción que permita alquilar un coche.

```
create or replace procedure alquilar( arg_NIF_cliente varchar,  
                                     arg_matricula varchar, arg_fecha_ini date, arg_fecha_fin date) is
```

Para proceder a alquilar un coche la transacción debe:

1. Comprobar si la fecha de inicio pasada como argumento no es posterior a la fecha fin pasada como argumento. En caso contrario devolverá el **error -20001 con el mensaje 'El numero de dias sera mayor que cero.'**
2. Si se cumple la comprobación anterior, deberá utilizar una SELECT con un par de *joins* para saber el valor del modelo del vehículo pasado como argumento, el precio que cuesta alquilarlo diariamente, la capacidad de su depósito de combustible, el tipo de combustible que utiliza y el precio por litro del mismo.

Del resultado de esta SELECT deberías ser capaz de deducir si el vehículo existe. Si no existiese has de devolver el **error -20002 con el mensaje 'Vehiculo inexistente.'**

Se recomienda utilizar esta SELECT para bloquear el vehículo y que nadie pueda insertar reservas que se solapen con la que estoy insertando antes de que finalice mi transacción. Se recomienda la lectura de la siguiente consulta en asktom (Ask The Oracle Mentor) , especialmente la parte final

https://asktom.oracle.com/pls/asktom/f?p=100:11:::P11_QUESTION_ID:42171194352295

Asegúrate que lo que estás bloqueando en ese *join* es el vehículo y no el modelo o el tipo de combustible. Aquí tienes alguna idea: <https://asktom.oracle.com/pls/apex/asktom.search?tag=for-update-with-a-join>

3. Siguiendo la misma estrategia del artículo anteriormente referenciado utilizar una SELECT para saber si en el intervalo entre **arg_fecha_ini** y **arg_fecha_fin** no existe ya alguna reserva solapada en la tabla de reservas.

Si existiese habría que devolver la **excepción -20003, con el mensaje 'El vehiculo no esta disponible.'**

4. Insertamos una fila en la tabla de reservas para el cliente, vehículo e intervalo de fechas pasado como argumento. En esta operación deberíamos ser capaces de detectar si el cliente no existe, en cuyo caso lanzaremos la **excepción -20004, con el mensaje 'Cliente inexistente'**.

Contesta a las siguientes preguntas como comentarios dentro de tu script referenciando la pregunta cómo P4a, P4b, P4c.

- a) El resultado de la SELECT del paso anterior ¿sigue siendo fiable en este paso?:
 - b) En este paso, la ejecución concurrente del mismo procedimiento ALQUILA con, quizás otros o los mismos argumentos, ¿podría habernos añadido una reserva no recogida en esa SELECT que fuese incompatible con nuestra reserva?, ¿por qué?.
 - c) En este paso otra transacción concurrente cualquiera ¿podría hacer INSERT o UPDATE sobre reservas y habernos añadido una reserva no recogida en esa SELECT que fuese incompatible con nuestra reserva?, ¿por qué?.
5. Se crea una factura correspondiente a alquilar al cliente con ese NIF el vehículo con esa matrícula durante los días transcurridos: $n_dias = fecha_fin - fecha_ini$.

El campo importe de la factura se rellena con la suma de los importes de las líneas de factura, que se crearán como se indica más adelante. Conviene que tengas esa suma hallada en este paso para no tener que hacer un UPDATE innecesario sobre la factura.

Piensa por qué en este paso ninguna transacción concurrente podría habernos borrado el cliente o haber cambiado su NIF. Incluye la explicación en un comentario sobre el script con la referencia P5.

Si `fecha_fin` es `null`, se entiende que la fecha final no está cerrada.

La transacción creará además dos líneas de factura:

- Una correspondiente a lo que cuesta alquilar el vehículo $n_días$, para ello en la primera SELECT tiene el precio del alquiler diario del vehículo que le multiplicará por $n_días$. El concepto de la línea será “*tantos días de alquiler vehículo modelo el-que-sea*”. Donde *tantos* es lo que valga $n_días$ y *el-que-sea* será el modelo del vehículo, que se puede saber también con esa primera SELECT.

Si la fecha final no está cerrada se le cobra 3 días.

- Otra línea de factura correspondiente a un depósito lleno de gasolina de ese vehículo. Para ello, la primera SELECT ya nos da la capacidad en litros de ese modelo, y el precio del combustible de ese modelo, los cuales multiplicaremos. En el concepto figurará “Deposito lleno (X litros de Y)”, donde X es el número de litros e Y el tipo de combustible).

(Utiliza CURRVAL para rellenar el campo NroFactura de la tabla de *Lineas de Factura*).

2. Normas de Entrega

La práctica se realizará por grupos de **DOS o TRES personas**.

Formato de entrega:

- Se enviará un fichero `.zip` a través de la plataforma **UBUVirtual** completando la tarea:

[ABD] Entrega de Trabajo 2 - PLSQL-1C

El fichero comprimido (`.zip` o `.rar`) seguirá el siguiente formato de nombre, **sin utilizar tildes**:

Nombre_1erApellido-Nombre_1erApellido.zip

- El fichero comprimido contendrá todos los scripts PL/SQL (formato `.sql`) generados para resolver el enunciado y que incluya todas las cuestiones planteadas. Las respuestas a las preguntas se incluirán como comentarios en el propio script.
Además, en caso de haber considerado un enfoque de programación frente a otros posibles, se añadirá su explicación (ej: elección de estilo de programación y cómo afecta en el número de peticiones SQL; elección en el tratamiento de excepciones; utilización de estrategia pesimista o no, etc)
- Aunque el trabajo se haga en grupos de DOS o TRES, **será enviado por todos los integrantes del grupo**.
- **NO se admiten envíos posteriores a la fecha y hora límite**, ni a través de otro medio que no sea la entrega de la tarea en UBUVirtual. Si no se respetan las normas de envío la calificación es cero.
- **Fecha límite: 20 de Abril, 23:59**
- En caso de **alguna irregularidad** podría solicitarse una defensa del trabajo en horario de tutorías, a fijar con los profesores responsables de la asignatura. Una exagerada diferencia entre el resultado de este trabajo y el del examen del tema 2, suponen razón suficiente para solicitar esa defensa. En caso de no ser capaz de defender el trabajo, la calificación será modificada acorde.