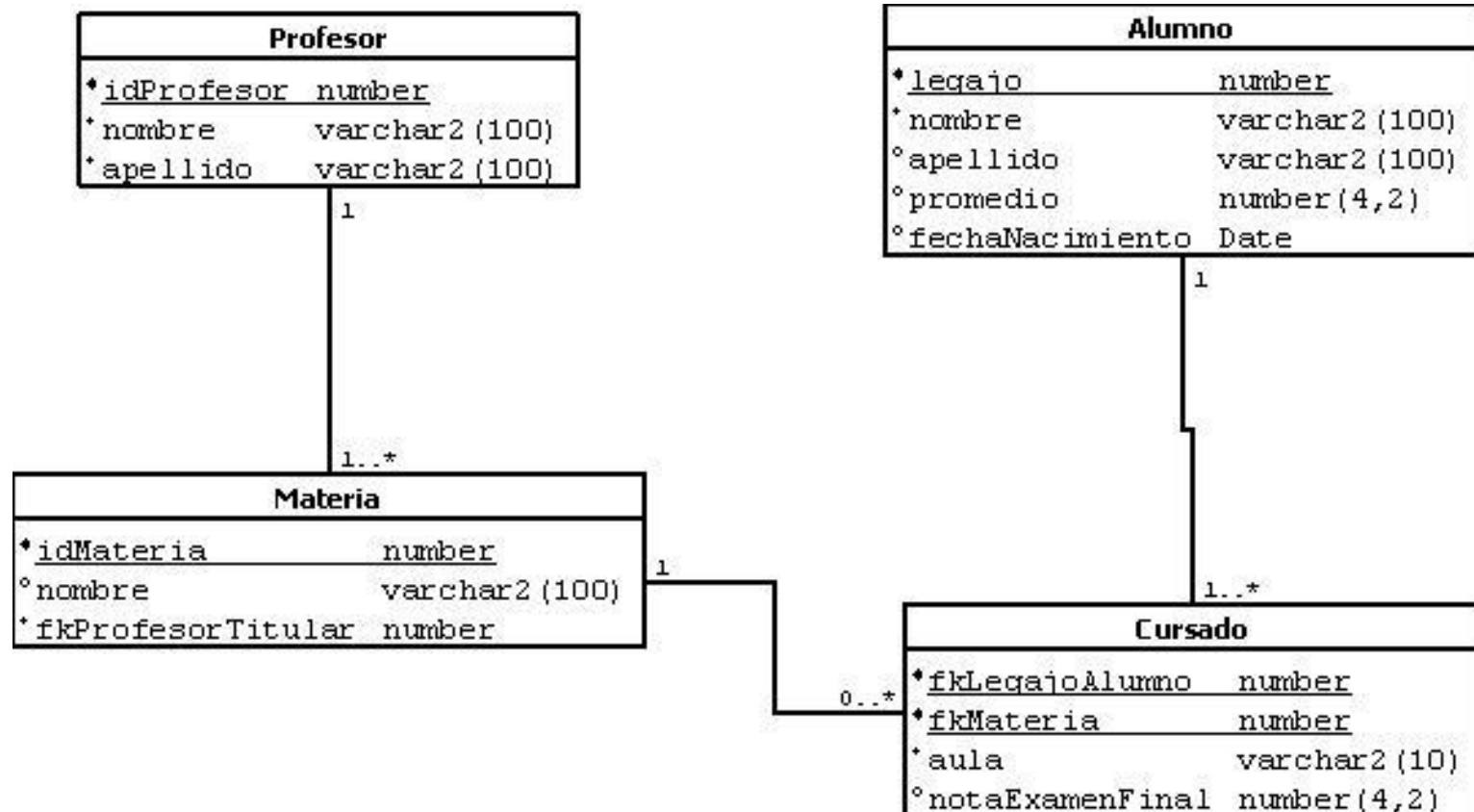


Ejercicios PL/SQL- Clase 1 y 2

1. Cree una base de datos con el siguiente esquema tablas:



2. Carga de tabla Profesor

Armar un bloque pl que utilizando un bucle FOR inserte 10 Profesores.

Las características de los registros son:

- El idProfesor será un numero generado que comience en 10 y se vaya incrementado de 20 en 20
- El nombre del profesor se formara de la siguiente manera= Nombre_idProfesor (*Nombre_10,...*)
- El apellido del profesor se formara de la siguiente manera= Apellido_idProfesor (*Apellido_10,...*)

3. Carga de tabla Materia

Armar un bloque pl que utilizando alguna estructura de bucle inserte datos en la tabla materia.

Se deben insertar por cada profesor de la tabla Profesor una materia con las siguientes características:

- El idMateria será un numero generado que comience en 1 y se vaya incrementado de 1 en 1
- El nombre de la materia se formará de la siguiente manera = Materia_idMateria (*Materia_1,...*)
- El fkProfesorTitular de la materia se sacara de la tabla Profesor

4. Carga de tabla Alumno

Armar un bloque pl que utilizando alguna estructura de bucle inserte datos en la tabla alumnos.

Se deben insertar 100 alumnos con las siguientes características:

- El legajo será un numero generado que comience en 10 y se vaya incrementado de 2 en 2
- El nombre del alumno se formará de la siguiente manera = Nombre_Legajo (*Nombre_10,...*)
- El apellido del alumno se formará de la siguiente manera = Apellido_Legajo (*Apellido_10,...*)
- El promedio del alumno en cero
- Fecha de nacimiento será igual a:
 $\text{fechaNacimiento} = \text{Fecha_Actual} - \text{XCantidad}$
 donde $\text{XCantidad} = (\text{legajo} * 2)$

De la fecha actual se deben restar "Años", por ej:

$\text{XCantidad} = 2$

$\text{fechaNacimiento} = 30/03/2012 - \text{XCantidad} = 30/03/2010$

No se debe almacenar la hora, minutos y segundos en la fecha de nacimiento.

Se recomienda analizar la siguiente consulta e investigar sobre las funciones que aparecen en la misma:

```
select trunc((sysdate + numtoyminterval(2, 'YEAR')) from dual;
```

5. Carga de tabla Cursado:

Armar un bloque pl que utilizando alguna estructura de bucle inserte datos en la tabla cursado.

- Se deben asociar a cada alumno todas las materias, por ejemplo si hay solo 2 alumnos Alumno_1 y Alumno_2 y existen 3 materias Materia_1, Materia_2 y Materia_3, el total de registros será 6:

fkLegajoAlumno	fkMateria
1	1
1	2
1	3
2	1
2	2
2	3

- El aula se formara de la siguiente manera: A_<<contador>> (A_1, A_2...)
Tenga en cuenta que para toda materia siempre se mantiene la misma Aula.
(No puede existir un Alumno_1 que curse la Materia_1 en el aula A_1 y un Alumno_2 que curse la misma materia en el aula A_2)

- Las notas de los exámenes finales se deben obtener de manera aleatoria en el rango 1-10. Se recomienda utilizar la siguiente consulta:

```
SELECT (1+ABS(MOD(dbms_random.random,10)))FROM dual;
```

Analice cada una de las funciones de la consulta.

6. Crear los sub-bloques que crea necesario para realizar un buen tratamiento de errores mediante excepciones.
7. Investigar sobre la excepción TOO_MANY_ROWS y agregar la misma en los bloques que sea factible que se lance.
8. Investigar sobre la excepción NO_DATA_FOUND y agregar la misma en los bloques que sea factible que se lance.
9. Investigar sobre la excepción DUP_VAL_ON_INDEX y agregar la misma en los bloques que sea factible que se lance.
10. Crear un bloque pl que calcule el promedio para todos los alumnos y haga el update pertinente en la tabla Alumno.
11. Crear un bloque pl que para un alumno en base al promedio general del mismo lo califique como:
promedio entre 1 y 4 => Alumno Nivel 1
promedio entre 5 y 8 => Alumno Nivel 2
promedio entre 9 y 10 => Alumno Nivel 3
12. Cree un paquete con el nombre PkgUtilColegio que contenga un procedimiento por cada bloque pl que ha desarrollado antes.

Todo procedimiento deberá contener sus parámetros de entrada (en el caso de que lo requiera para su ejecución) y 3 parámetros de salida que informen si el procedimiento se ejecuto exitosamente o no:

p_appcod (number): código propio del procedimiento (establecido por el desarrollador). Si es cero => OK. En otro caso se considera que el procedimiento no se ejecuto exitosamente.

p_sqlcode (number): código SQLCODE

p_sql_error (varchar2(500)): descripción SQLERRM

Cada procedimiento debe tener el control pertinente para efectuar el commit o rollback y dejar consistente la BD.

13. Crear un paquete con el nombre PkgLogger que implemente un procedimiento llamado registerError que servirá para dejar un registro de los errores que se pueden producir en la ejecución de los procedimientos del paquete PkgUtilColegio.

Este procedimiento deberá recibir por parámetro el nombre del proceso que genero el error (por ej.

PkgUtilColegio.cargarProfesores), el código de la aplicación (p_appcod), el SQLCODE y el SQLERRM

Esos valores se deberán registrar en una tabla llamada logger_transaction que contenga un campo por cada uno de los parámetros y un campo mas con el nombre add_date (date) que registrara la fecha en que ingreso el error a la tabla (se debe tomar la fecha actual que posee el motor de la BD).

Tenga en cuenta que este procedimiento debe hacer commit o rollback en una transacción autónoma, de tal manera que si se efectúa un rollback desde la transacción de donde se invoco el procedimiento registerError, no vuelva atrás las instrucciones que se efectuaron dentro de este ultimo (registerError).

14. Dentro del paquete PkgLogger cree un procedimiento llamado summarize_errors que lleve a cabo un informe de los errores.

Recibirá un parametro p_tipo_informe de entrada del tipo number que indicara el tipo de informe que se quiere generar y otro parámetro p_fecha_desde que indicara a partir de qué fecha se quiere sumarizar los datos (add_date >= p_fecha_resumen).

- Si p_tipo_informe=1, generar informe que resuma por proceso la cantidad de errores asociado al mismo (por ej. Proceso: PkgUtilColegio.cargarProfesores Cantidad Total errores:100)

- Si p_tipo_informe=2, generar informe que resuma por proceso la cantidad de errores distintos asociados al mismo
(por ej. Proceso: PkgUtilColegio.cargarProfesores Cantidad Total errores distintos:50)

Para mostrar los informes use put_line del paquete dbms_output.