

Practica de PL-SQL 1 modelo facultad.

1. Crear una función que dado un legajo y una carrera retorne un 1 si el alumno es regular y un 0 en caso que no lo sea.
2. Crear una función que dado un legajo y una carrera retorne el promedio de notas obtenidas en finales rendidos en el año actual. En caso de no haber rendido ningún final para este año retornar el promedio de las notas de los parciales rendidos en este año.
3. Crear una función que dada una materia retorne el nombre y apellido del último alumno que haya obtenido la mejor nota en un final de dicha materia.
4. Crear una función que dado un docente retorne cual es el promedio de duración de los cursos que ha dictado en el corriente año (se debe considerar los cursos dictados en el corriente año que hayan finalizado al momento que se ejecute la función). El resultado debe ser expresado en meses.
SQLSERVER: DATEDIFF (datepart , startdate , enddate)
ORACLE: MONTHS_BETWEEN(d1, d2)
5. Crear un procedimiento que dado un legajo elimine al alumno y retorne un mensaje en un parámetro de salida indicando si la operación se realizó correctamente o en caso de no poder realizarse indique claramente por que.
6. Crear un procedimiento que dados un curso y una nota promocie a los alumnos de dicho curso cuyo promedio de notas obtenidos en los parciales sea mayor o igual a la nota ingresada como parámetro.
El procedimiento deberá generar las filas necesarias de la tabla finales con los datos correspondientes y retornar en un parámetro de salida cuantos alumnos de dicho curso se promocionaron.
7. Crear un procedimiento que dados un curso, una fecha de inicio y una de fin, genere un nuevo curso para las fechas ingresadas (para la misma materia y con el mismo docente del curso ingresado como parámetro) e inscriba en dicho curso a los alumnos que no aprobaron el curso ingresado como parámetro.
El procedimiento deberá retornar dos parámetros que indiquen la cantidad de alumnos inscriptos en el nuevo curso y el código del nuevo curso. El código del curso deberá ser el siguiente al último curso existente.
8. Crear un procedimiento que dado un año inserte para todas las materias en la tabla ranking_materias el nombre y apellido del alumno que mejor nota obtuvo en dicha materia para el año ingresado como parámetro (si hubiera mas de uno se debe seleccionar el ultimo que rindió). En caso que nadie haya rendido la materia durante ese año insertar en el campo NOMBRE la leyenda "Materia no rendida durante el año XXXX". Las materias deben ser insertadas por orden alfabético.
RANKING_MATERIAS

NOTA: LA TABLA ESTA CREADA Y VACIA

9. Crear un procedimiento que dada una materia y una cantidad n muestre por pantalla los n primeros nombres y apellidos de los docentes posibles de asignar a un curso de dicha materia. Se deben mostrar aquellos docentes que ya hayan dictado esa materia al menos 2 veces. Y que durante el corriente año hayan dictado como máximo 5 cursos de cualquier materia. Los docentes deben ser procesados por legajo ascendente. En caso de no haber n docentes que cumplan los requisitos se completaran los n mensajes con la leyenda "No hay docentes con los requisitos ingresados"
10. Crear un procedimiento que dados un curso y una nota controle que los alumnos de dicho curso hayan aprobado el curso solo si en todos los parciales obtuvieron una nota mayor o igual a la ingresada como parámetro.
En el caso que un alumno haya sido aprobado y no cumpla con lo antes mencionado se deberá modificar la tabla `alumno_curso` y en la columna `alcu_aprobado` colocar el valor 0, además se modificará la columna `observaciones` donde se colocara el mensaje "Alumno aprobado incorrectamente numero n " siendo n un contador de alumnos procesados por dicho procedimiento hasta el momento.
En caso que el alumno haya sido aprobado correctamente se agregara a las observaciones existentes la frase "Aprobación verificada" (se deberá controlar que la frase completa no supere el máximo de la columna en dicho caso las observaciones no se modificaran).
El procedimiento deberá retornar dos parámetros, uno indicando la cantidad total de alumnos verificados y otro la cantidad de alumnos aprobados incorrectamente.
Los alumnos se deberán procesar por orden alfabético según su apellido.