

## BD - Examen de la Segunda Evaluación

Para la realización de este examen dispones de **tres horas**.

Para la entrega del examen se entregarán los archivos generados separados y comprimidos en un único archivo .ZIP cuyo nombre debe ser **Apellidos\_Nombre.zip** del alumno.

**Presentación:** Se valorará positivamente la presentación del script, la claridad de las explicaciones de los comentarios y que esté correctamente estructurado.

Si dudas en algún elemento de los enunciados, pregunta.

Los criterios de evaluación son los siguientes:

CRITERIOS DE EVALUACIÓN		VALORACIÓN MÁXIMA
Ejercicio 1 (2.25 puntos)	Apdo. A	0.75 puntos
	Apdo. B	0.50 puntos
	Apdo. C	0.50 puntos
	Apdo. D	0.50 puntos
Ejercicio 2 (3.75 puntos)	Apdo. A.	2.00 puntos
	Apdo. B.	1.75 puntos
Ejercicio 3 (4.00 puntos)	Apdo. A.	0.75 puntos
	Apdo. B.	2.00 puntos
	Apdo. C.	1.25 puntos

### Recursos:

Se suministra al alumno equipo con una máquina virtual Windows 10, OracleDB 18c Express Edition y Sql Developer.

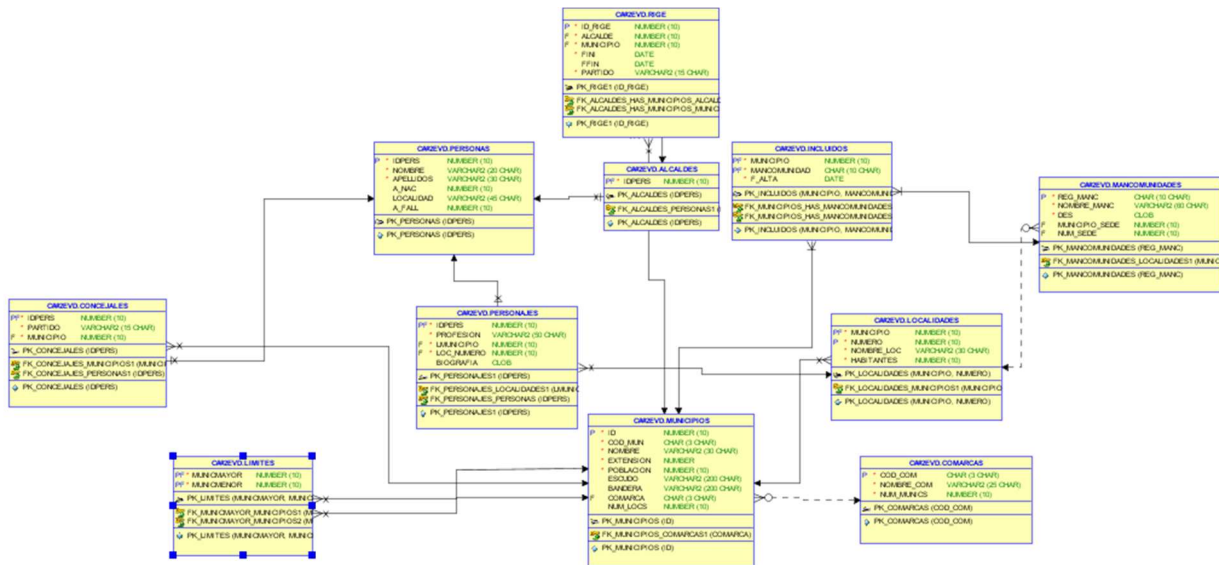
La contraseña del usuario administrador de Oracle es 123456.

Se suministra el archivo MunicipiosPLSQL.sql con las sentencias para crear las tablas necesarias en OracleDB para la realización de la prueba. Si tienes alguna dificultad para cargar las tablas en OracleBD pregunta.

### Archivos a entregar en el archivo .zip:

- **Archivo .sql** con lo solicitado en los tres ejercicios. El código deberá estar correctamente comentado y formateado.

### Ejercicio 1 DML-Actualizaciones (2.5 puntos)



Teniendo en cuenta el diagrama relacional mostrado de la base de datos facilitada de municipios, realiza las siguientes inserciones/actualizaciones/borrados:

- A. Inserta las dos localidades del municipio de Limpias en la tabla LOCALIDADES. Se supone que no hay actualmente ninguna localidad en esa tabla LOCALIDADES para el municipio de Limpias. Por ello, esas localidades estarán numeradas con 1 y 2. Se debe realizar las instrucciones usando el nombre del municipio. Utiliza las sentencias sql que consideres necesarias.

Localidad	Habitantes
Limpias	1329
Seña	347

- B. Se supone que se acaban de insertar los datos de cuatro personas en la tabla PERSONAS y que los identificadores de esas cuatro personas son números comprendidos entre 943 y 946. Insertar cuatro filas en la tabla CONCEJALES para indicar que esas cuatro personas son concejales del municipio de Arredondo por el partido PRC.
- C. Modifica los habitantes de todas las localidades del municipio de Alfoz de Lloredo de más de 200 habitantes para incrementar esos habitantes en un 2%.
- D. Elimina la localidad Cabrojo perteneciente al municipio de Cabezón de la Sal.

### Ejercicio 2 PL/SQL Lenguaje Procedimental

- A. En la base de datos municipios desarrolla un procedimiento PL/SQL que cargue, *cada vez que se ejecute*, ciertos datos en una tabla llamada Informe que ya existe en las tablas facilitadas en Municipios.sql. El procedimiento deberá insertar en la tabla informe los nombres de los municipios ordenados alfabéticamente junto al nombre y habitantes de las cuatro localidades con más habitantes de cada municipio. Hay que tener en cuenta que hay municipios que pueden tener menos de 4 localidades e incluso ninguna. En los casos de municipios que no tengan ninguna localidad, en la tabla informe deberá aparecer el nombre del municipio, la localidad vacía y la población que tenga el municipio. Debes utilizar al menos un cursor. Escribe la sentencia para ejecutar el procedimiento creado. Puedes ver un ejemplo de cómo debería quedar la información en la tabla informe tras ejecutar el procedimiento en la siguiente imagen:

NOMBREMUNI	NOMBRELOC	HABITANTES
Alfoz de Lloredo	Oreña	860
Alfoz de Lloredo	Cóbreces	613
Alfoz de Lloredo	Novales	466
Alfoz de Lloredo	Rudagüera	379
Ampuero	(null)	0
Anievas	(null)	0
Arenas de Iguña	(null)	0
Argoños	(null)	0

- B. En la base de datos municipios, crea un TRIGGER actúe cuando se inserte o actualice una localidad. El trigger deberá asegurar que se actualicen adecuadamente los habitantes del municipio al que pertenece la localidad insertada o modificada en función de los habitantes que tenga la localidad insertada/modificada.

### Ejercicio 3 PL/SQL Objetos

- A. Crear un tipo objeto llamado PELDANO con tres atributos de tipo número con dos decimales, ancho, alto y fondo. Crear un tipo de objeto llamado MODELO, subtipo de peldano, con dos atributos, código de tipo char(1), descmodelo texto(10). Este tipo de objeto debe tener un método que permita ordenar por él basado en el código del modelo. Crea un tipo llamado ListaPeldanos que sea una colección de objetos de tipo peldano.
- B. Crea un tipo de objeto llamado Escalera con los atributos: numserie varchar(20), numpeldanos de tipo entero, modeloEsc de tipo Modelo, listaPeldanosEsc de tipo listaPeldanos y altura total.

- Este tipo de objeto debe tener una **función** llamada *calculaNumPeldanos* que devuelva el número de peldaños que necesita tener la escalera ( $\text{altura} / (\text{distanciaentre peldaños} + \text{alturapeldaño})$ ).
- Este tipo de objeto debe tener un **constructor** que reciba 3 parámetros, el número de serie, la altura total de la escalera y la distancia entre peldaños. El constructor del tipo de objeto escalera debe:

Inicializar los valores necesarios.

Definir el modelo de la escalera en función de la altura total de esta:

- Si la escalera mide menos de 100 cm la escalera será del modelo pequeño: P - Pequeño - Peldaño de 25 ancho, 5 grosor y 10 fondo
- Si la escalera mide entre 100 y 200 la escalera será del modelo mediano: M - Mediano - Peldaño de 35 ancho, 8 grosor y 20 fondo
- Si la escalera mide más de 200 la escalera será del modelo grande: G - Grande - Peldaño de 50 ancho, 10 grosor y 25 fondo

Calcular el número de peldaños necesarios (usando función ya creada) y establecer el valor del atributo numpeldanos.

Llamar al método *construyeEscalera*.

- El método *construyeEscalera* debe:

Añadir a la lista de peldaños tantos PELDANOS como indique el atributo numpeldanos.

Añadir un peldaño más al final de la lista de peldaños que será el tope superior de la escalera de iguales características que los anteriores.

Nota: Para construir cada peldaño deberá usarse la información que tenemos de medidas del peldaño en el modelo que tenga asociado a la escalera.

- C. Crear una tabla llamada Escaleras para almacenar objetos de tipo Escalera. Insertar en la tabla Escaleras varios *objetos* de tipo escalera:

```
numserie:'00000000000001',altura total:250, distancia entre peldaños:25  
numserie:'00000000000002',altura total:150, distancia entre peldaños:15  
numserie:'00000000000003',altura total:50, distancia entre peldaños:20
```

Crear una consulta que devuelva la información suficiente para verificar que el modelo que cada escalera creada tiene asignado en la tabla Escaleras es correcto según su altura total.

¿Es posible realizar una consulta de las escaleras almacenadas ordenadas por la columna modelo1? Explica el motivo.

