

**PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU
FACULTAD DE CIENCIAS E INGENIERÍA**

**BASES DE DATOS
2do Examen
(2024-2)**

Indicaciones generales:

- Duración: 2 horas 50 minutos.
- Está permitido el uso de material de clase (diapositivas), pero no compartirlo. Solo hasta las 10:30 a.m.
- Para el caso de las consultas no pueden usar vistas.
- Si la respuesta tiene errores de sintaxis, no se le considera puntaje en la pregunta.
- Si la consulta muestra filas no solicitadas o repeticiones, no se le considera puntaje en la pregunta.
- El archivo que contenga sus scripts o modelos no deberá ser comprimido. Debe cumplir con el formato y tipo de extensión solicitado sino la pregunta no será calificada.
- Los archivos indicados se subirán a **PAIDEIA**, en el espacio donde se ha encontrado este documento. Se destinarán los últimos 10 minutos exclusivamente para subir los trabajos a **PAIDEIA**.
- Guarde cada uno de sus archivos con el nombre que se le indica. Es importante seguir el estándar del nombre de archivo indicado.
- La presentación, el cumplimiento del estándar del nombre del archivo, la ortografía y la gramática influyen en la calificación.

Puntaje total: 20 puntos

Pregunta 1. (5 puntos)

Es un cuestionario que se realiza en PAIDEIA del curso (2024-2) BASE DE DATOS (1INF33). Debe de realizarse en el horario de 10:30 hasta 11:00 horas y tiene una duración de 10 minutos.

Escoja la opción adecuada de las preguntas. Cada respuesta correcta vale un (1) punto. Cada pregunta mal contestada tendrá un puntaje negativo de -0.25 puntos.

Pregunta 2. (5 puntos)

Se tiene los siguientes datos para la tabla PRODUCTO:

```
CREATE TABLE PRODUCTO (  
  CODIGO INT PRIMARY KEY,  
  NOMBRE VARCHAR(50),  
  PRECIO DECIMAL(10, 2),  
  STOCK INT  
);
```

```
INSERT INTO PRODUCTO VALUES (101, 'Laptop', 1200.00, 10);  
COMMIT;
```

Tres personas (U1, U2 y U3) intentan modificar estos datos ejecutando las sentencias que se indican en el respectivo tiempo T (Ver Tabla 1). En un archivo Excel como se indica en la Tabla 2 colocar la explicación de lo que sucede en cada tiempo T en todas

las sesiones. En las columnas siguientes colocar aquellos valores de la variable Precio que aparecen con signos interrogación en la celda correspondiente.

Tabla 1: Ejercicio: Manejo de Transacciones

T	U1	U2	U3
1	UPDATE PRODUCTO SET PRECIO = PRECIO * 1.10 WHERE CODIGO = 101;		
2		SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;	
3			UPDATE PRODUCTO SET PRECIO = PRECIO * 0.95 WHERE CODIGO = 101;
4	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;		
5	COMMIT;		
6		SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;
7			ROLLBACK;
8	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;
9		UPDATE PRODUCTO SET PRECIO = PRECIO * 0.93 WHERE CODIGO = 101;	
10		COMMIT;	
11	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;		
12			UPDATE PRODUCTO SET PRECIO = PRECIO * 1.15 WHERE CODIGO = 101;
13		SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;
14	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;		
15			COMMIT;
16	SELECT PRECIO FROM PRODUCTO WHERE CODIGO = 101;		

Se pide elaborar:

- a) (5 puntos) Una tabla donde se indique lo que sucede en cada tiempo T y los valores que se obtienen en cada sesión.

Debe desarrollar su respuesta en la Tabla 2.

Tabla 2: Formato para Respuesta en Excel

<u>Tiempo</u>	<u>Explicación</u>	<u>Precio U1</u>	<u>Precio U2</u>	<u>Precio U3</u>
<u>1</u>		<u>?</u>		
<u>2</u>			<u>?</u>	
<u>3</u>				
<u>4</u>		<u>?</u>		
<u>5</u>		<u>?</u>		
<u>6</u>			<u>?</u>	<u>?</u>
<u>7</u>				
<u>8</u>		<u>?</u>	<u>?</u>	<u>?</u>
<u>9</u>				
<u>10</u>				
<u>11</u>		<u>?</u>		
<u>12</u>				
<u>13</u>			<u>?</u>	<u>?</u>
<u>14</u>		<u>?</u>		
<u>15</u>				
<u>16</u>		<u>?</u>		

Grabar su archivo que contiene su solución con el nombre de:
P2_<código_alumno>.xls

Pregunta 3. (6 puntos)

Se tienen las siguientes tablas de una base de datos que servirán para almacenar los datos de las elecciones presidenciales en el Perú:

REGION_PERU(idregion, nombre)

idregion: código del Perú

nombre: nombre de la región

PARTIDO_POLITICO(idpartido, nombre)

idpartido: código del partido político

nombre: nombre del partido político

VOTACION(idregion, idpartido, votos)

idregion: código de la región

idpartido: código del partido político

votos: cantidad de votos obtenidos por el partido en esa región

VOTOS_PARTIDO(nombre_region, votos)

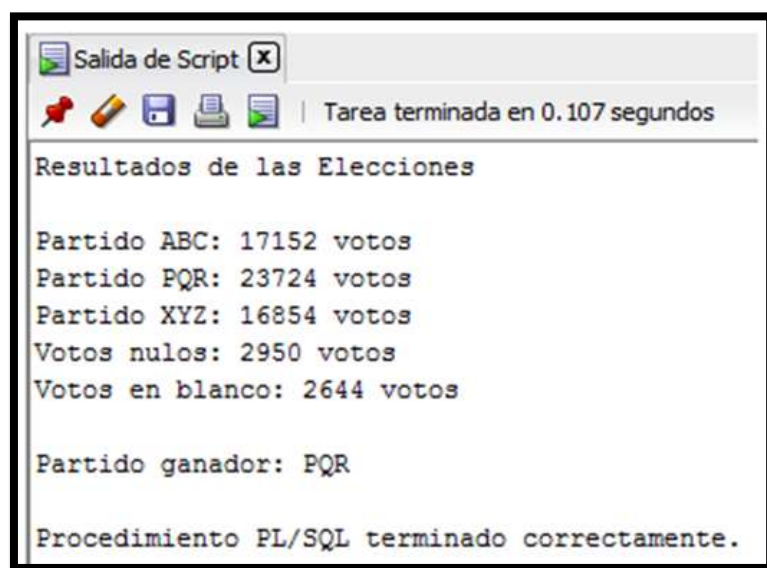
nombre_region: nombre de la región

votos: cantidad de votos obtenidos por un partido en esa región

Archivos para utilizar: ***DROSPreg3.txt, DDLpreg3.txt y DMLpreg3.txt***

Observe que en las dos primeras filas de la tabla PARTIDO_POLITICO, se han almacenado Blanco y Nulo, que obviamente no son nombres de partidos políticos, por lo que tienen un tratamiento especial. Por el momento, solo se han almacenado tres partidos (ABC, PQR, XYZ), sin embargo, las soluciones que desarrollará deben funcionar independientemente de la cantidad de partidos que haya.

- a) (3.0 puntos) Escriba un procedimiento llamado **pr_resultados_elecciones** que permita mostrar los resultados de las elecciones tal y como se muestra en la siguiente imagen:



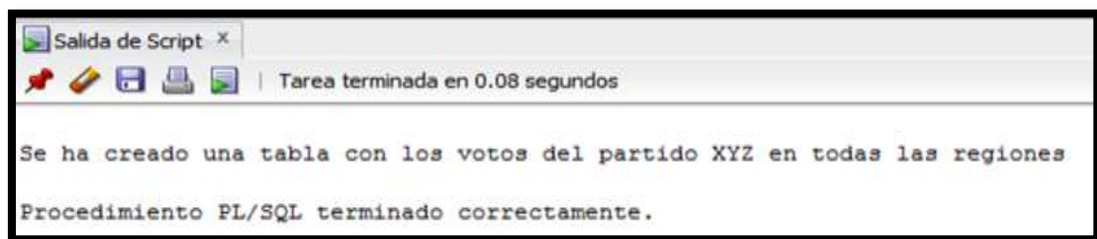
Observe los siguientes detalles:

- los nombres de los partidos están ordenados alfabéticamente
- la cantidad de votos nulos y en blanco se muestran después de los nombres de los partidos
- al final se muestra el nombre del partido ganador

b) (3 puntos) Escriba un procedimiento llamado **pr_almacenar_votos_partido** que reciba como parámetro el nombre de un partido político y que almacene en la tabla **VOTOS_PARTIDO** los votos obtenidos por dicho partido en cada una de las regiones del Perú. En el caso de que el partido no exista se debe activar una excepción que muestre el mensaje indicando tal situación. Esta excepción debe ser creada por el programador (no se permite el uso de excepciones predefinidas en PL/SQL). Los nombres de las regiones deben ser insertadas en orden alfabético. En caso la tabla VOTOS_PARTIDO ya tenga contenido, elimine dicho contenido antes de realizar las inserciones.

Por ejemplo:

exec pr_almacenar_votos_partido('XYZ');



Si el procedimiento se ha implementado correctamente y se consulta el contenido de la tabla VOTOS_PARTIDO, se debería obtener el siguiente resultado:

Resultado de la Consulta	
Todas las Filas Recuperadas: 4 en 0.004 segundos	
NOMBRE_REGION	VOTOS
1 CUZCO	2356
2 HUANUCO	2574
3 PIURA	7125
4 TACNA	4799

Grabar su archivo conteniendo los procedimientos con el nombre:
P3_<código_alumno>.txt

Pregunta 4. (4 puntos)

Las columnas "WHO" (también llamadas "*audit columns*" o "*tracking columns*") son utilizadas en bases de datos para realizar un seguimiento de las modificaciones realizadas a las filas de una tabla. Estas columnas almacenan información sobre quién creó o modificó un registro, cuándo lo hizo, y en algunos casos cómo o desde dónde. Son particularmente útiles en sistemas que requieren un seguimiento de auditoría, como sistemas ERP, CRM, o cualquier aplicación empresarial que necesite mantener un historial de cambios en los datos.

Las columnas típicas que se pueden usar en el esquema "WHO" en una tabla de base de datos:

1. **CREATED_BY**: Esta columna registra el identificador del usuario que creó el registro.
2. **CREATION_DATE**: La fecha y hora exacta en que se creó el registro.
3. **LAST_UPDATED_BY**: El identificador del usuario que modificó por última vez el registro.
4. **LAST_UPDATE_DATE**: La fecha y hora de la última modificación del registro.

Se le pide crear un mecanismo que automatice del relleno de estas columnas.

Realice la creación de la tabla usando el siguiente script.

```
CREATE TABLE EX2_EMPLEADOS (  
    EMPLEADO_ID NUMBER PRIMARY KEY,  
    NOMBRE VARCHAR2(100),  
    APELLIDO VARCHAR2(100),  
    SALARIO NUMBER,  
    CREATED_BY VARCHAR2(50),  
    CREATION_DATE DATE,  
    LAST_UPDATED_BY VARCHAR2(50),  
    LAST_UPDATE_DATE DATE  
);
```

- a) (4.0 puntos) Implemente un **TRIGGER** que se dispare inmediatamente después de que se inserte una nueva fila en la tabla **EX2_EMPLEADO**. Este **TRIGGER** también debe actualizar en la tabla **EX2_EMPLEADO** las columnas `last_updated_by` y `last_update_date` cada vez que realice una actualización de una fila.

Antes de iniciar las sentencias debe ejecutar:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'dd/MM/YYYY HH24:MI:SS';
```

Si su *trigger* ha sido correctamente implementado, debería obtener el siguiente resultado:

Primero inserte los datos en la tabla **EX2_EMPLEADOS** que se proporcionan, mantenga la secuencia de ejecución:

- INSERT INTO EX2_EMPLEADOS (EMPLEADO_ID, NOMBRE, APELLIDO, SALARIO) VALUES(100, 'JUAN', 'PEREZ', 1500);
- INSERT INTO EX2_EMPLEADOS (EMPLEADO_ID, NOMBRE, APELLIDO, SALARIO) VALUES(101, 'ANA', 'PEREZ', 1600);

Luego de realizados los *INSERTS* indicados, haga la siguiente consulta:

```
SELECT * FROM EX2_EMPLEADOS;
```

EMPLEADO_ID	NOMBRE	APELLIDO	SALARIO	CREATED_BY	CREATION_DATE	LAST_UPDATED_BY	LAST_UPDATE_DATE
1	100	JUAN PEREZ	1500	PROFESOR	09/12/2024 07:42:25	PROFESOR	09/12/2024 07:42:25
2	101	ANA PEREZ	1600	PROFESOR	09/12/2024 07:42:30	PROFESOR	09/12/2024 07:42:30

- UPDATE EX2_EMPLEADOS SET salario=1800 WHERE empleado_id= 100;

Luego del primer UPDATE haga la consulta:

SELECT * FROM EX2_EMPLEADOS;

EMPLEADO_ID	NOMBRE	APELLIDO	SALARIO	CREATED_BY	CREATION_DATE	LAST_UPDATED_BY	LAST_UPDATE_DATE
1	100	JUAN PEREZ	1800	PROFESOR	09/12/2024 07:42:25	PROFESOR	09/12/2024 07:43:27
2	101	ANA PEREZ	1600	PROFESOR	09/12/2024 07:42:30	PROFESOR	09/12/2024 07:42:30

- UPDATE EX2_EMPLEADOS SET salario=1925;

Luego del segundo UPDATE haga la consulta:

SELECT * FROM EX2_EMPLEADOS;

EMPLEADO_ID	NOMBRE	APELLIDO	SALARIO	CREATED_BY	CREATION_DATE	LAST_UPDATED_BY	LAST_UPDATE_DATE
1	100	JUAN PEREZ	1925	PROFESOR	09/12/2024 07:42:25	PROFESOR	09/12/2024 07:51:37
2	101	ANA PEREZ	1925	PROFESOR	09/12/2024 07:42:30	PROFESOR	09/12/2024 07:51:37

Grabar su archivo de la solución con el nombre de archivo: **P4_<código_alumno>.txt**

Puede usar las funciones:

USER

devuelve el identificador de autorización o el nombre del usuario actual

SYSDATE

retorna la fecha y hora actuales en el servidor de Oracle.

Profesores del curso: César Aguilera Serpa
Hilmar Hinojosa Lazo
Isaac Yrigoyen Montestruque

San Miguel, 10 de diciembre del 2024