

Examen Final de Asignatura

EXAMEN FINAL

ASIGNATURA:

Programación de Base de Datos

APRENDIZAJES ESPERADOS

- En esta actividad lograrás construir componentes PL/SQL con flujos y resultados controlados por excepciones de acuerdo con los requerimientos de la organización.
- Utilizar cursores explícitos con y sin parámetros que permitan procesar datos masivamente para solucionar los requerimientos de información planteados.
- Crear triggers asociados a eventos específicos que efectúen operaciones automáticas sobre los datos para solucionar los requerimientos de información planteados.
- Codificar la estructura de creación, parámetros necesarios y lógica de procedimientos almacenados de acuerdo con los requerimientos de información requeridos.
- Codificar la estructura de creación, parámetros y lógica de funciones de acuerdo con los requerimientos de información requeridos.
- Codificar el retorno de la función de acuerdo con los requerimientos de información requeridos.

INSTRUCCIONES GENERALES

Estimado(a) estudiante:

- A partir de los siguientes requerimientos, se solicita desarrolle las sentencias PL/SQL que permitan dar solución a estos.
- Se pide leer atentamente los requerimientos y desarrollar las sentencias PL/SQL para dar solución a estos. Se han agregado imágenes para que pueda establecer una base de lo que se espera como resultado, una vez haya ejecutado su sentencia. Las imágenes son referenciales, puesto en ocasiones solo muestran una parcialidad de los registros (por temas de espacio dentro del documento).
- Para el desarrollo del examen final de la asignatura, debe utilizar el software Oracle Developer asociado a su base datos Oracle Cloud con la conexión del usuario creado. Guarde la evaluación con el formato: NOMBRE_APELLIDO. sql o .txt.
- Como respaldo, también se requiere que envíe un Word con imágenes (pantallazos o screenshots) de lo desarrollado por usted, además de los pasos que describan el proceso de desarrollo de su evaluación.

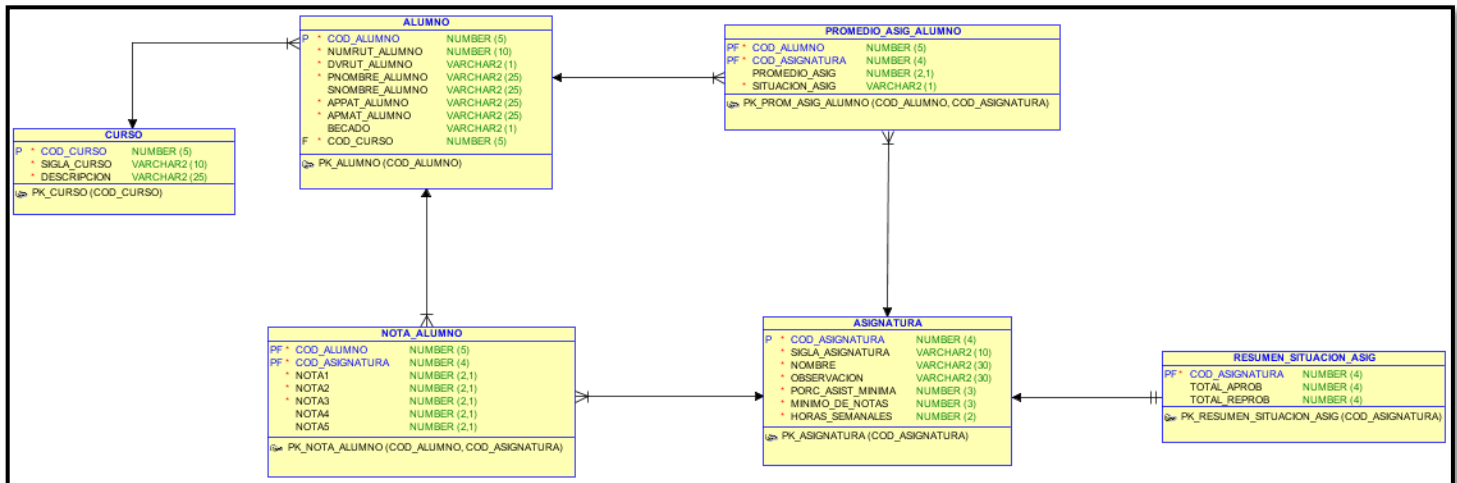


Esta actividad se complementa con una pauta de autoevaluación al finalizar, para que verifiques el logro de tus aprendizajes

Contexto

La dirección del colegio NUESTRO MUNDO lo ha contactado a Ud. para que pueda implementar una solución que automatice su proceso de cálculo de promedios de notas para cada una de las asignaturas de los alumnos. Si bien ya se encuentra implementado el proceso de registro online de las notas de los alumnos, falta complementarlo con la solución que usted implemente.

La información base se encuentra almacenada en una base de Datos cuyo modelo es el que muestra a continuación:



Cree un usuario en su base de datos Oracle Cloud, luego conéctese con el usuario y ejecute el archivo script_creacion_tablas_examen.sql para crear y poblar las tablas del Modelo con los siguientes datos:

TABLA CURSO

COD_CURSO	SIGLA_CURSO	DESCRIPCION
211	PMA	PRIMERO MEDIO A
212	PMB	PRIMERO MEDIO B
221	SMA	SEGUNDO MEDIO A
222	SMB	SEGUNDO MEDIO B

TABLA ASIGNATURA

COD_ASIGNATURA	SIGLA_ASIGNATURA	NOMBRE	OBSERVACION	PORC_ASIST_MINIMA	MINIMO_DE_NOTAS	HORAS_SEMANALES
1211	MATEM	MATEMÁTICAS	ASIG. PRIMERO MEDIO	90	5	10
1212	LENG	LENGUAJE	ASIG. PRIMERO MEDIO	90	5	10
1213	ING	INGLES	ASIG. PRIMERO MEDIO	70	4	8
1214	ED.FIS	EDUCACIÓN FÍSICA	ASIG. PRIMERO MEDIO	70	4	7
1215	FIS	FÍSICA	ASIG. PRIMERO MEDIO	66	4	5
2211	MATEM2	MATEMÁTICAS 2	ASIG. SEGUNDO MEDIO	90	5	10
2212	LENG2	LENGUAJE 2	ASIG. SEGUNDO MEDIO	90	5	10
2213	ING2	INGLES 2	ASIG. SEGUNDO MEDIO	70	4	8
2214	ED.FIS2	EDUCACIÓN FÍSICA 2	ASIG. SEGUNDO MEDIO	70	4	7
2215	FIS2	FÍSICA 2	ASIG. SEGUNDO MEDIO	66	4	5

TABLA ALUMNO

COD_ALUMNO	NUMRUT_ALUMNO	DVRUT_ALUMNO	PNOMBRE_ALUMNO	SNOMBRE_ALUMNO	APPAT_ALUMNO	APMAT_ALUMNO	BECADO	COD_CURSO
2111	18012987	1	JUAN	PABLO	RAMOS	PLAZA	S	211
2112	19983767	K	MARIA	ISABEL	ALVAREZ	FLORES	S	211
2113	19243861	8	ALEJANDRA	ISABEL	GUAJARDO	SAN MARTIN	S	211
2114	20997232	7	SERGIO	RAFAEL	ARAVENA	ARAYA	(null)	211
2211	18777666	6	CRISTIAN	HERNAN	BELTRAN	FLORES	(null)	221
2221	18998111	1	FRANCISCO	JAVIER	CASTRO	DONOSO	S	222

TABLAS NOTA_ALUMNO

COD_ALUMNO	COD_ASIGNATURA	NOTA1	NOTA2	NOTA3	NOTA4	NOTA5
2111	1211	5,5	5	4,5	3,6	3,7
2111	1212	4,6	6,6	3,4	5	4,4
2111	1213	4	4,1	4,2	4	3,8
2111	1214	5	4,5	4,4	4,2	4,3
2111	1215	6,3	2,8	2,7	2,5	3,7
2112	1211	6,7	6,8	7	7	6,8
2112	1212	6,8	6,9	7	6,4	6,2
2112	1213	7	7	7	7	4,5
2112	1214	6,7	6,9	5,8	6,9	5
2112	1215	6,9	7	6,8	6,6	6,6
2113	1211	2,5	3,7	4,8	4,4	4,8
2113	1212	3,5	3,6	3,3	2	2,2
2113	1213	2,4	2,4	2,2	2,1	2,2
2113	1214	3,3	3,4	3,5	3,2	3,7
2113	1215	2,2	2,1	2,3	2,2	1
2114	1211	4,5	4,4	4,3	4,2	4,9
2114	1212	4	4	4,2	3,9	3,9
2114	1213	4,4	4,3	4,1	4	4,4
2114	1214	5,4	5,2	4,4	4,3	5,4
2114	1215	3,4	3,5	4,4	4,4	4,8
2211	1211	6,5	6,6	6,7	6,8	7
2211	1212	5,3	4,2	4,4	3,9	4
2211	1213	3,5	2,9	3,5	4,5	3,5
2211	1214	2,4	3,3	4,5	5,6	5,5
2211	1215	4,1	4,2	4	4,2	4,9
2221	1211	6,7	6,8	6,9	7	7
2221	1212	6,9	6,8	6,6	6,4	5,5
2221	1213	5,6	5,5	5,7	6,8	7
2221	1214	7	7	7	7	7
2221	1215	4,6	5,5	5,6	6,6	2

TABLA RESUMEN_SITUACION_ASIG

COD_ASIGNATURA	TOTAL_APROB	TOTAL_REPROB
1211	0	0
1212	0	0
1213	0	0
1214	0	0
1215	0	0
2211	0	0
2212	0	0
2213	0	0
2214	0	0
2215	0	0

Requerimiento 1

Se le solicita construir procedimientos, funciones y triggers que consideren los siguientes requerimientos mínimos en términos de diseño:

- Crear una función almacenada que obtenga el promedio de notas de un alumno y la asignatura que cursa, ingresados como parámetros.
- Un procedimiento almacenado principal que procese a todos los alumnos. En este procedimiento debe utilizar la función almacenada para obtener el promedio de notas por cada alumno procesado. Los alumnos aprobados son aquellos cuyo promedio es mayor o igual a 4,0.
- Al momento de insertar el promedio de la asignatura para cada alumno en tabla PROMEDIO_ASIG_ALUMNO, se deberá actualizar la tabla RESUMEN_SITUACION_ASIG sumándolo a la columna total_aprob si el alumno aprobó la asignatura o sumándolo a la columna total_reprob si el alumno reprobó la asignatura.
- Todos los programas creados deben controlar cualquier error que ocurra durante el procesamiento de la información garantizando así que el proceso no se interrumpa. Se debe almacenar los errores producidos en la tabla con los siguientes datos: id error, nombre subprograma en el que se produjo el error y el mensaje de error.
- Al finalizar la ejecución del proceso, las tablas deberían tener los siguientes datos:

COD_ALUMNO	COD_ASIGNATURA	PROMEDIO_ASIG	SITUACION_ASIG
2111	1211	4,5 A	
2111	1212	4,8 A	
2111	1213	4 A	
2111	1214	4,5 A	
2111	1215	3,6 R	
2112	1211	6,9 A	
2112	1212	6,7 A	
2112	1213	6,5 A	
2112	1214	6,3 A	
2112	1215	6,8 A	
2113	1211	4 A	
2113	1212	2,9 R	
2113	1213	2,3 R	
2113	1214	3,4 R	
2113	1215	2 R	
2114	1211	4,5 A	
2114	1212	4 A	
2114	1213	4,2 A	
2114	1214	4,9 A	
2114	1215	4,1 A	
2211	1211	6,7 A	
2211	1212	4,4 A	
2211	1213	3,6 R	
2211	1214	4,3 A	
2211	1215	4,3 A	
2221	1211	6,9 A	
2221	1212	6,4 A	
2221	1213	6,1 A	
2221	1214	7 A	
2221	1215	4,9 A	

TABLA RESUMEN_SITUACION_ASIG

COD_ASIGNATURA	TOTAL_APROB	TOTAL_REPROB
1211	6	0
1212	5	1
1213	4	2
1214	5	1
1215	4	2
2211	0	0
2212	0	0
2213	0	0
2214	0	0
2215	0	0

Inicio de desarrollo de script, para determinar valores para tabla PROMEDIO_ASIG_ALUMNO y modificación valores a tabla RESUMEN_SITUACION_ASIG, según lo solicitado en Req 1.

```

--REQUERIMIENTO 1:

-- Habilitación de mensajes
SET SERVEROUTPUT ON;

-- Creación de secuencia para los errores
CREATE SEQUENCE ERRORES_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

-- Función FN_PROMEDIO_ALUMNO
CREATE OR REPLACE FUNCTION FN_PROMEDIO_ALUMNO (
    p_cod_alumno IN NUMBER,
    p_cod_asignatura IN NUMBER
) RETURN NUMBER IS
    v_promedio NUMBER;
BEGIN
    -- Calcular promedio de notas
    SELECT ROUND((NVL(NOTA1, 0) + NVL(NOTA2, 0) + NVL(NOTA3, 0) + NVL(NOTA4, 0) + NVL(NOTA5, 0)) / 5, 2)
    INTO v_promedio
    FROM NOTA_ALUMNO
    WHERE COD_ALUMNO = p_cod_alumno AND COD_ASIGNATURA = p_cod_asignatura;

    RETURN v_promedio;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No se encontraron notas para el alumno ' || p_cod_alumno || ' y asignatura ' || p_cod_asignatura);
        RETURN NULL;
    WHEN OTHERS THEN
        RAISE;
END FN_PROMEDIO_ALUMNO;
/

```

```
-- Procedimiento SP_PROCESAR_ALUMNOS
CREATE OR REPLACE PROCEDURE SP_PROCESAR_ALUMNOS IS
-- Declaraciones
CURSOR cur_alumnos IS
    SELECT COD_ALUMNO, COD_ASIGNATURA FROM NOTA_ALUMNO;

v_promedio NUMBER;
v_situacion_asig VARCHAR2(10);
BEGIN
-- Recorremos cada alumno y asignatura
FOR reg IN cur_alumnos LOOP
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Procesando alumno: ' || reg.COD_ALUMNO || ', asignatura: ' || reg.COD_ASIGNATURA);

        -- Calcular promedio usando la función
        v_promedio := FN_PROMEDIO_ALUMNO(reg.COD_ALUMNO, reg.COD_ASIGNATURA);

        -- Verificar si el promedio es válido
        IF v_promedio IS NOT NULL THEN
            -- Determinar situación del alumno
            IF v_promedio >= 4.0 THEN
                v_situacion_asig := 'A';
            ELSE
                v_situacion_asig := 'R';
            END IF;

            -- Insertar en la tabla PROMEDIO_ASIG_ALUMNO
            INSERT INTO PROMEDIO_ASIG_ALUMNO (COD_ALUMNO, COD_ASIGNATURA, PROMEDIO_ASIG, SITUACION_ASIG)
            VALUES (reg.COD_ALUMNO, reg.COD_ASIGNATURA, v_promedio, v_situacion_asig);

            DBMS_OUTPUT.PUT_LINE('Promedio insertado: ' || v_promedio || ', situación: ' || v_situacion_asig);

            -- Actualizar la tabla RESUMEN_SITUACION_ASIG
            IF v_promedio >= 4.0 THEN
                UPDATE RESUMEN_SITUACION_ASIG
                SET TOTAL_APROB = TOTAL_APROB + 1
                WHERE COD_ASIGNATURA = reg.COD_ASIGNATURA;

                DBMS_OUTPUT.PUT_LINE('Asignatura aprobada actualizada para: ' || reg.COD_ASIGNATURA);
            ELSE
                UPDATE RESUMEN_SITUACION_ASIG
                SET TOTAL_REPROB = TOTAL_REPROB + 1
                WHERE COD_ASIGNATURA = reg.COD_ASIGNATURA;

                DBMS_OUTPUT.PUT_LINE('Asignatura reprobada actualizada para: ' || reg.COD_ASIGNATURA);
            END IF;
        ELSE
            DBMS_OUTPUT.PUT_LINE('Promedio no calculado para alumno ' || reg.COD_ALUMNO || ' y asignatura ' || reg.COD_ASIGNATURA);
        END IF;
    EXCEPTION
    WHEN OTHERS THEN
        -- Registrar error
        DECLARE
            v_error_msg VARCHAR2(4000);
        BEGIN
            v_error_msg := SQLERRM;
            INSERT INTO ERRORES (ID_ERROR, SUBPROGRAMA_ERROR, DESCRIPCION_ERROR)
            VALUES (ERRORES_SEQ.NEXTVAL, 'SP_PROCESAR_ALUMNOS', v_error_msg);

            DBMS_OUTPUT.PUT_LINE('Error procesando alumno ' || reg.COD_ALUMNO || ': ' || v_error_msg);
        END;
    END;
END LOOP;
```

```
-- Confirmar cambios
COMMIT;
DBMS_OUTPUT.PUT_LINE('Proceso completado.');
```

END SP_PROCESAR_ALUMNOS;

/

-- Trigger TRG_MANEJO_ERRORES

CREATE OR REPLACE TRIGGER TRG_MANEJO_ERRORES

AFTER INSERT OR UPDATE OR DELETE ON PROMEDIO_ASIG_ALUMNO

FOR EACH ROW

DECLARE

v_error_msg VARCHAR2(4000);

BEGIN

IF INSERTING THEN

DBMS_OUTPUT.PUT_LINE('Se insertó un promedio para el alumno ' || :NEW.COD_ALUMNO);

ELSIF UPDATING THEN

DBMS_OUTPUT.PUT_LINE('Se actualizó un promedio para el alumno ' || :OLD.COD_ALUMNO);

ELSIF DELETING THEN

DBMS_OUTPUT.PUT_LINE('Se eliminó un promedio para el alumno ' || :OLD.COD_ALUMNO);

END IF;

EXCEPTION

WHEN OTHERS THEN

v_error_msg := SQLERRM;

INSERT INTO ERRORES (ID_ERROR, SUBPROGRAMA_ERROR, DESCRIPCION_ERROR)

VALUES (ERRORES_SEQ.NEXTVAL, 'TRG_MANEJO_ERRORES', v_error_msg);

DBMS_OUTPUT.PUT_LINE('Error en el trigger: ' || v_error_msg);

END;

/

-- Ejecutar el procedimiento

BEGIN

SP_PROCESAR_ALUMNOS;

END;

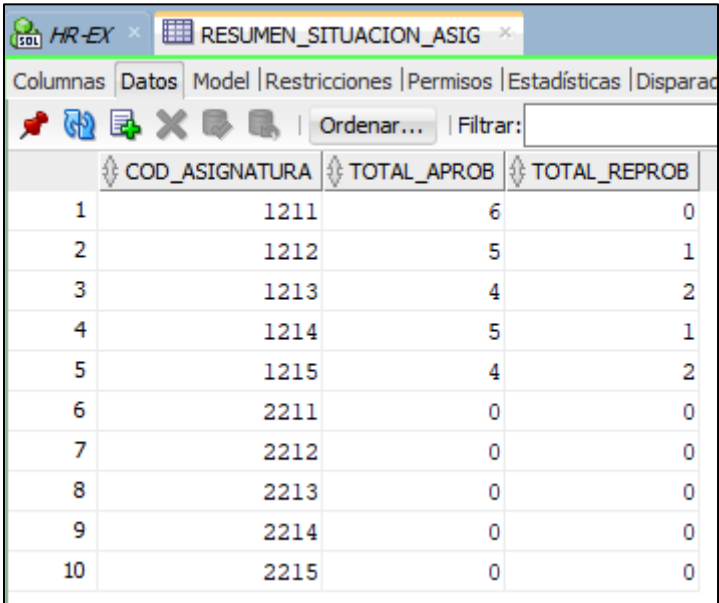
/

COMMIT;

Ejecución de script para la revisión por tabla grafica de incorporación datos en tabla PROMEDIO_ASIG_ALUMNO, según lo solicitado en Req 1

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias				
Ordenar... Filtrar:				
	COD_ALUMNO	COD_ASIGNATURA	PROMEDIO_ASIG	SITUACION_ASIG
1	2111	1211	4,5 A	
2	2111	1212	4,8 A	
3	2111	1213	4 A	
4	2111	1214	4,5 A	
5	2111	1215	3,6 R	
6	2112	1211	6,9 A	
7	2112	1212	6,7 A	
8	2112	1213	6,5 A	
9	2112	1214	6,3 A	
10	2112	1215	6,8 A	
11	2113	1211	4 A	
12	2113	1212	2,9 R	
13	2113	1213	2,3 R	
14	2113	1214	3,4 R	
15	2113	1215	2 R	
16	2114	1211	4,5 A	
17	2114	1212	4 A	
18	2114	1213	4,2 A	
19	2114	1214	4,9 A	
20	2114	1215	4,1 A	
21	2211	1211	6,7 A	
22	2211	1212	4,4 A	
23	2211	1213	3,6 R	
24	2211	1214	4,3 A	
25	2211	1215	4,3 A	
26	2221	1211	6,9 A	
27	2221	1212	6,4 A	
28	2221	1213	6,1 A	
29	2221	1214	7 A	
30	2221	1215	4,9 A	

Ejecución de script para la revisión por tabla grafica de modificación de datos en tabla RESUMEN_SITAUACION_ASIG, según lo solicitado en Req 1



	COD_ASIGNATURA	TOTAL_APROB	TOTAL_REPROB
1	1211	6	0
2	1212	5	1
3	1213	4	2
4	1214	5	1
5	1215	4	2
6	2211	0	0
7	2212	0	0
8	2213	0	0
9	2214	0	0
10	2215	0	0

Requerimiento 2

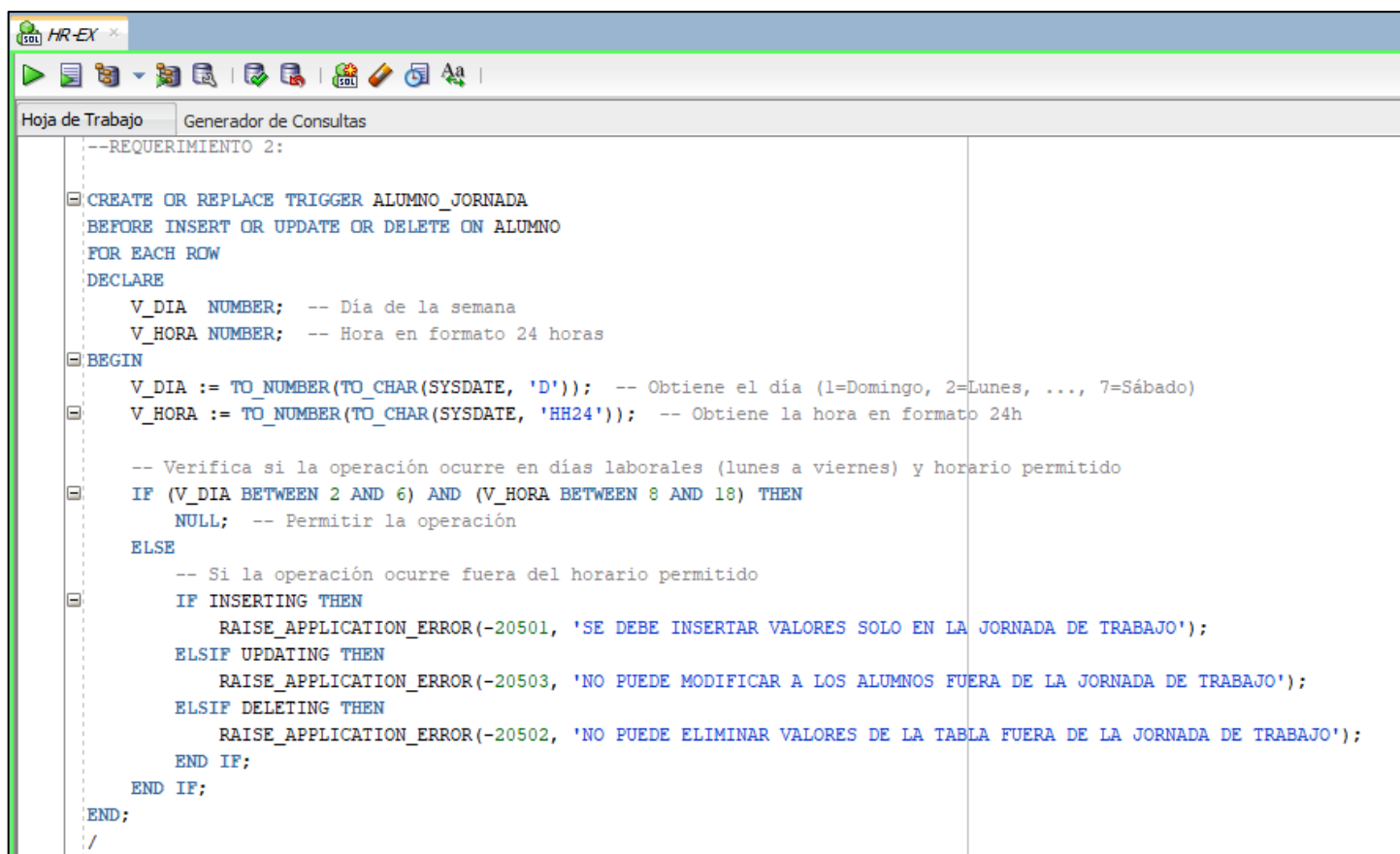
El colegio es uno de los más importantes del país y tiene políticas súper claras de bienestar familiar, por lo mismo quiere que el sistema de ingresos de nuevos alumnos, solo se ocupe en las jornadas de trabajo y que sea de lunes a viernes de 08:00 a 18:00. Para esto le solicitan a usted crear algo que impida el ingreso de nuevos alumnos, eliminación de alumnos y actualización de alumnos ya registrados. Estos errores deben ser controlados como si fueran errores del sistema de Oracle y en los siguientes números de errores:

- Si están insertando, el error es: -20501 y el mensaje debe ser 'SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO'.
- Si están actualizando los datos del alumno, el error es: -20503 y el mensaje debe ser 'NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO'.
- Si están eliminando, el error es: -20502 y el mensaje debe ser 'NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO'.

Para comprobar el buen funcionamiento de esta sección se le solicita hacer lo siguiente:

- Ingresar un nuevo alumno.
- Eliminar un alumno.
- Actualizar el nombre a un alumno.

Desarrollo requerimiento 2:

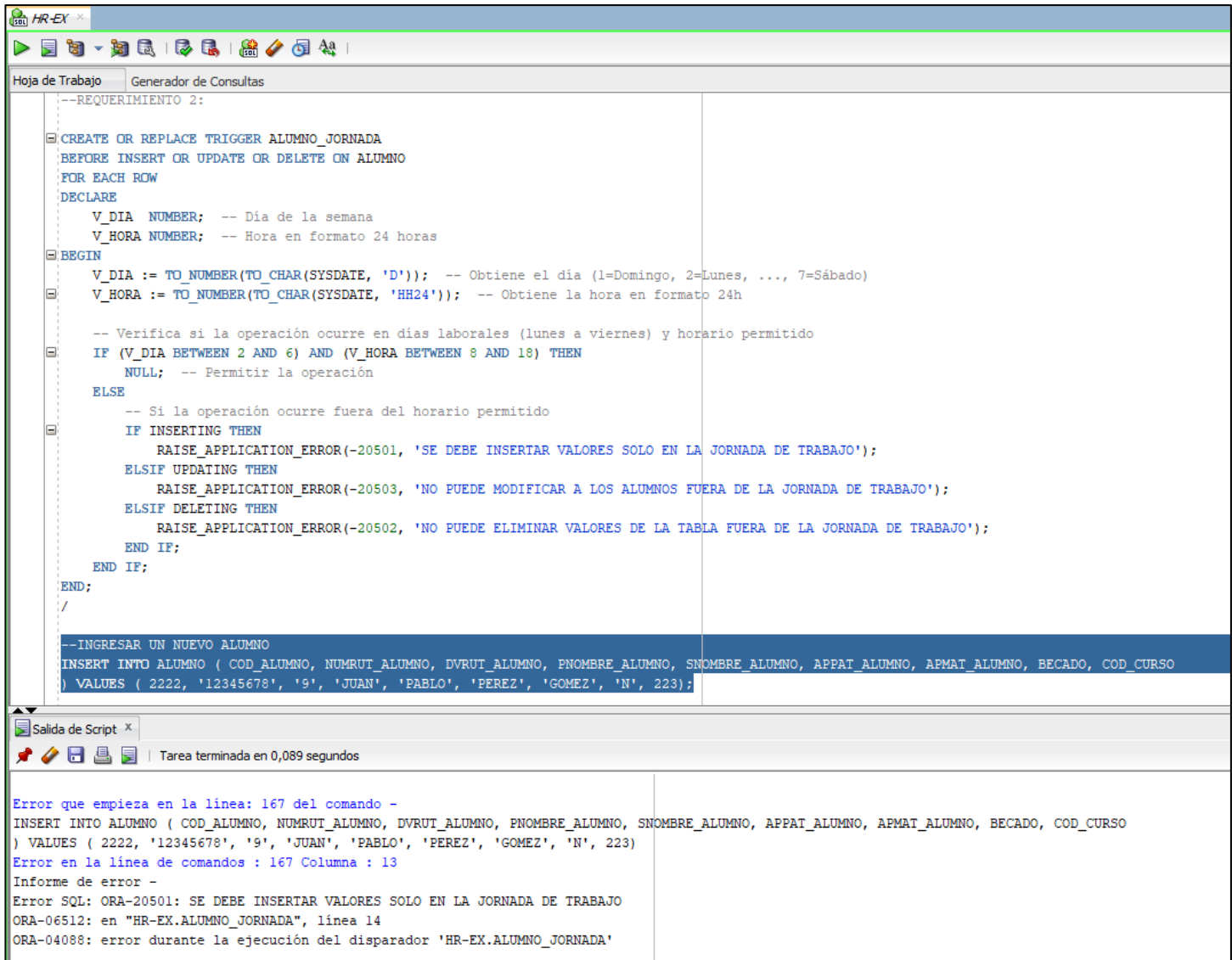


```
--REQUERIMIENTO 2:

CREATE OR REPLACE TRIGGER ALUMNO_JORNADA
BEFORE INSERT OR UPDATE OR DELETE ON ALUMNO
FOR EACH ROW
DECLARE
    V_DIA    NUMBER; -- Día de la semana
    V_HORA   NUMBER; -- Hora en formato 24 horas
BEGIN
    V_DIA := TO_NUMBER(TO_CHAR(SYSDATE, 'D')); -- Obtiene el día (1=Domingo, 2=Lunes, ..., 7=Sábado)
    V_HORA := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')); -- Obtiene la hora en formato 24h

    -- Verifica si la operación ocurre en días laborales (lunes a viernes) y horario permitido
    IF (V_DIA BETWEEN 2 AND 6) AND (V_HORA BETWEEN 8 AND 18) THEN
        NULL; -- Permitir la operación
    ELSE
        -- Si la operación ocurre fuera del horario permitido
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20501, 'SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20503, 'NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20502, 'NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO');
        END IF;
    END IF;
END;
/
```

Ejecución de Ingreso un nuevo alumno, en donde, se genera mensaje de error al no estar en horarios correspondientes de trabajo.



The screenshot shows the HR-EX SQL Developer application. The main window displays a SQL script for a trigger named ALUMNO_JORNADA. The script is as follows:

```
--REQUERIMIENTO 2:
CREATE OR REPLACE TRIGGER ALUMNO_JORNADA
BEFORE INSERT OR UPDATE OR DELETE ON ALUMNO
FOR EACH ROW
DECLARE
    V_DIA NUMBER; -- Día de la semana
    V_HORA NUMBER; -- Hora en formato 24 horas
BEGIN
    V_DIA := TO_NUMBER(TO_CHAR(SYSDATE, 'D')); -- Obtiene el día (1=Domingo, 2=Lunes, ..., 7=Sábado)
    V_HORA := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')); -- Obtiene la hora en formato 24h

    -- Verifica si la operación ocurre en días laborales (lunes a viernes) y horario permitido
    IF (V_DIA BETWEEN 2 AND 6) AND (V_HORA BETWEEN 8 AND 18) THEN
        NULL; -- Permitir la operación
    ELSE
        -- Si la operación ocurre fuera del horario permitido
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20501, 'SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20503, 'NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20502, 'NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO');
        END IF;
    END IF;
END;
/
```

Below the script, there is a comment and an SQL statement:

```
--INGRESAR UN NUEVO ALUMNO
INSERT INTO ALUMNO ( COD_ALUMNO, NUMRUT_ALUMNO, DVRUT_ALUMNO, PNOMBRE_ALUMNO, SNOMBRE_ALUMNO, APPAT_ALUMNO, APMAT_ALUMNO, BECADO, COD_CURSO
) VALUES ( 2222, '12345678', '9', 'JUAN', 'PABLO', 'PEREZ', 'GOMEZ', 'N', 223);
```

The bottom panel shows the output of the script execution, which includes an error message:

```
Salida de Script x
Tarea terminada en 0,089 segundos

Error que empieza en la línea: 167 del comando -
INSERT INTO ALUMNO ( COD_ALUMNO, NUMRUT_ALUMNO, DVRUT_ALUMNO, PNOMBRE_ALUMNO, SNOMBRE_ALUMNO, APPAT_ALUMNO, APMAT_ALUMNO, BECADO, COD_CURSO
) VALUES ( 2222, '12345678', '9', 'JUAN', 'PABLO', 'PEREZ', 'GOMEZ', 'N', 223)
Error en la línea de comandos : 167 Columna : 13
Informe de error -
Error SQL: ORA-20501: SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO
ORA-06512: en "HR-EX.ALUMNO_JORNADA", línea 14
ORA-04088: error durante la ejecución del disparador 'HR-EX.ALUMNO_JORNADA'
```

Ejecución de Eliminación de un alumno, en donde, se genera mensaje de error al no estar en horarios correspondientes de trabajo.

The screenshot shows the HR-EX SQL Developer application. The main window displays a PL/SQL script with the following content:

```
--REQUERIMIENTO 2:

CREATE OR REPLACE TRIGGER ALUMNO_JORNADA
BEFORE INSERT OR UPDATE OR DELETE ON ALUMNO
FOR EACH ROW
DECLARE
    V_DIA NUMBER; -- Día de la semana
    V_HORA NUMBER; -- Hora en formato 24 horas
BEGIN
    V_DIA := TO_NUMBER(TO_CHAR(SYSDATE, 'D')); -- Obtiene el día (1=Domingo, 2=Lunes, ..., 7=Sábado)
    V_HORA := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')); -- Obtiene la hora en formato 24h

    -- Verifica si la operación ocurre en días laborales (lunes a viernes) y horario permitido
    IF (V_DIA BETWEEN 2 AND 6) AND (V_HORA BETWEEN 8 AND 18) THEN
        NULL; -- Permitir la operación
    ELSE
        -- Si la operación ocurre fuera del horario permitido
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20501, 'SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20503, 'NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20502, 'NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO');
        END IF;
    END IF;
END;
/

--INGRESAR UN NUEVO ALUMNO
INSERT INTO ALUMNO ( COD_ALUMNO, NUMRUT_ALUMNO, DVRUT_ALUMNO, PNOMBRE_ALUMNO, SNOMBRE_ALUMNO, APPAT_ALUMNO, APMAT_ALUMNO, BECADO, COD_CURSO
) VALUES ( 2222, '12345678', '9', 'JUAN', 'PABLO', 'PEREZ', 'GOMEZ', 'N', 223);

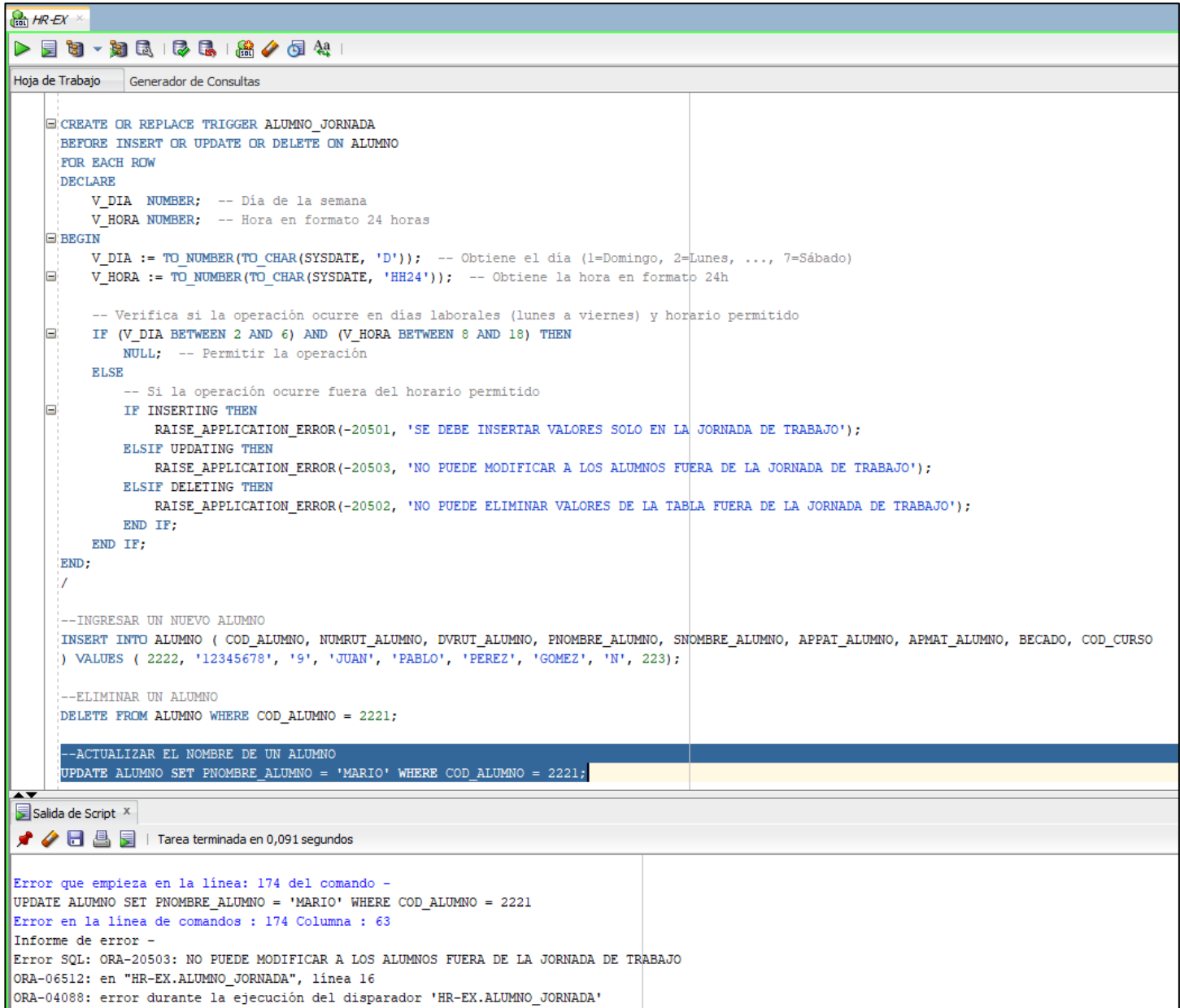
--ELIMINAR UN ALUMNO
DELETE FROM ALUMNO WHERE COD_ALUMNO = 2221;
```

The bottom panel shows the execution output:

```
Salida de Script x
Tarea terminada en 0,095 segundos

Error que empieza en la línea: 171 del comando -
DELETE FROM ALUMNO WHERE COD_ALUMNO = 2221
Error en la línea de comandos : 171 Columna : 39
Informe de error -
Error SQL: ORA-20502: NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO
ORA-06512: en "HR-EX.ALUMNO_JORNADA", línea 18
ORA-04088: error durante la ejecución del disparador 'HR-EX.ALUMNO_JORNADA'
```

Ejecución de Actualización de un alumno, en donde, se genera mensaje de error al no estar en horarios correspondientes de trabajo.



```

CREATE OR REPLACE TRIGGER ALUMNO_JORNADA
BEFORE INSERT OR UPDATE OR DELETE ON ALUMNO
FOR EACH ROW
DECLARE
    V_DIA NUMBER; -- Día de la semana
    V_HORA NUMBER; -- Hora en formato 24 horas
BEGIN
    V_DIA := TO_NUMBER(TO_CHAR(SYSDATE, 'D')); -- Obtiene el día (1=Domingo, 2=Lunes, ..., 7=Sábado)
    V_HORA := TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')); -- Obtiene la hora en formato 24h

    -- Verifica si la operación ocurre en días laborales (lunes a viernes) y horario permitido
    IF (V_DIA BETWEEN 2 AND 6) AND (V_HORA BETWEEN 8 AND 18) THEN
        NULL; -- Permitir la operación
    ELSE
        -- Si la operación ocurre fuera del horario permitido
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20501, 'SE DEBE INSERTAR VALORES SOLO EN LA JORNADA DE TRABAJO');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20503, 'NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO');
        ELSIF DELETING THEN
            RAISE_APPLICATION_ERROR(-20502, 'NO PUEDE ELIMINAR VALORES DE LA TABLA FUERA DE LA JORNADA DE TRABAJO');
        END IF;
    END IF;
END;
/

--INGRESAR UN NUEVO ALUMNO
INSERT INTO ALUMNO ( COD_ALUMNO, NUMRUT_ALUMNO, DVRUT_ALUMNO, PNOMBRE_ALUMNO, SNOMBRE_ALUMNO, APPAT_ALUMNO, APMAT_ALUMNO, BECADO, COD_CURSO
) VALUES ( 2222, '12345678', '9', 'JUAN', 'PABLO', 'PEREZ', 'GOMEZ', 'N', 223);

--ELIMINAR UN ALUMNO
DELETE FROM ALUMNO WHERE COD_ALUMNO = 2221;

--ACTUALIZAR EL NOMBRE DE UN ALUMNO
UPDATE ALUMNO SET PNOMBRE_ALUMNO = 'MARIO' WHERE COD_ALUMNO = 2221;
    
```

Salida de Script x

Tarea terminada en 0,091 segundos

Error que empieza en la línea: 174 del comando -
 UPDATE ALUMNO SET PNOMBRE_ALUMNO = 'MARIO' WHERE COD_ALUMNO = 2221
 Error en la línea de comandos : 174 Columna : 63
 Informe de error -
 Error SQL: ORA-20503: NO PUEDE MODIFICAR A LOS ALUMNOS FUERA DE LA JORNADA DE TRABAJO
 ORA-06512: en "HR-EX.ALUMNO_JORNADA", línea 16
 ORA-04088: error durante la ejecución del disparador 'HR-EX.ALUMNO_JORNADA'

PAUTA DE EVALUACIÓN

Criterios de Evaluación	Indicadores de Logro	Puntaje
<p>1) Los requerimientos planteados se resuelven utilizando de manera correcta los procedimientos y funciones almacenadas necesarias.</p> <p>2) La resolución de los requerimientos planteados se realiza controlando de manera correcta los eventuales errores que puedan aparecer durante el proceso.</p> <p>3) La resolución de los requerimientos planteados se desarrolla mediante la correcta realización del trigger correspondiente.</p>	Crea de forma correcta el procedimiento almacenado, necesario para resolver los requerimientos solicitados.	20
	Crea de forma correcta la función almacenada, necesaria para resolver los requerimientos solicitados.	20
	Controla de forma correcta los posibles errores que se produzcan durante la ejecución del procedimiento.	20
	Realiza correctamente el trigger, necesario para resolver el requerimiento solicitado.	20
	Imprime los resultados con la estructura de ejemplo en los requerimientos.	20
Puntaje Total		100

PAUTA DE AUTOEVALUACIÓN:

Estimado estudiante:

La autoevaluación es la valoración y/o reflexión que usted realiza sobre su proceso de aprendizaje. Este ejercicio es fundamental para identificar tanto debilidades como fortalezas de la evaluación realizada.

Para realizar la autoevaluación usted deberá contestar las siguientes preguntas, las cuales se encontrarán en la pestaña AUTOEVALUACIÓN, donde deberá completar cada una de ellas.

Conteste con toda la sinceridad posible, ya que será un insumo para la retroalimentación que realizará posteriormente el docente.

- ¿Qué hice bien? – Generar el desarrollo de lo solicitado.
- ¿En qué necesito ayuda? – En el desarrollo de la lógica
- ¿Sobre qué quiero saber más? – En la estructura adecuada de las BD
- ¿Qué aprendí en esta evaluación? – Aplicación resumen de los contenidos aprendidos hasta la fecha, pasando por las tres unidades del ramo.
- ¿Qué debo mejorar? – En el ordenamiento de la estructura de la BD
- ¿Qué acciones realizaré para mejorar? – Lectura y material relacionado a elaboración de bases.



4 INSTITUCION
ACREDITADA
NIVEL AVANZADO
AÑOS Hasta octubre 2025



GESTIÓN INSTITUCIONAL Y DOCENCIA DE PREGRADO