



[Área personal](#)

Mis cursos

[BD-TUDAI-Tandil-1C-2023](#)

Exámenes

[2do Parcial](#)



Comenzado el sábado, 24 de junio de 2023, 09:11

Estado Finalizado

Finalizado en sábado, 24 de junio de 2023, 11:45

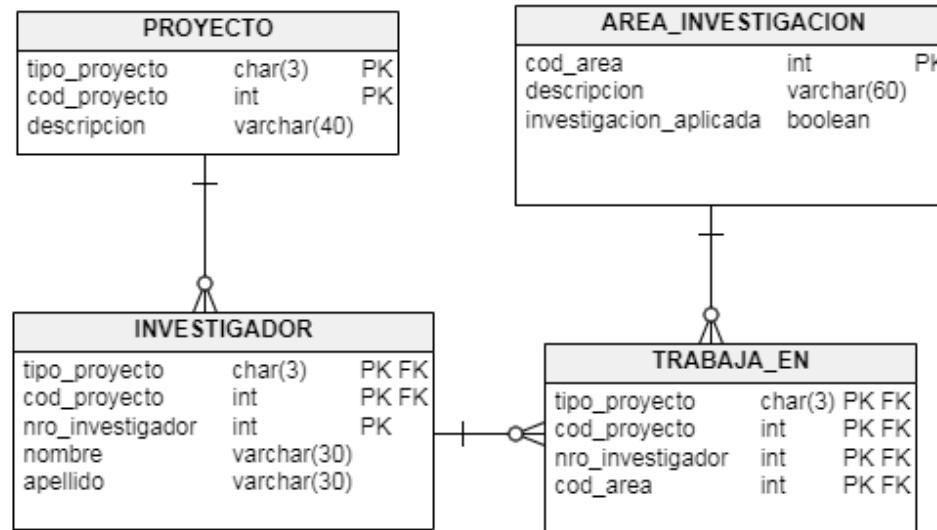
Tiempo 2 horas 33 minutos
empleado

Pregunta 1

Incorrecta

Puntúa como
2,00

Para el esquema de la figura



Se debe controlar que "los investigadores con apellido que comience con la letra A hasta la J pueden trabajar a lo sumo en 5 proyectos" cuya restricción declarativa es:

```
CREATE ASSERTION ck_medico_sala
CHECK ( NOT EXISTS (
    SELECT 1
    FROM trabaja_en t JOIN investigador i on
(t.tipo_proyecto=i.tipo_proyecto
    AND t.cod_proyecto=i.cod_proyecto
    AND i.nro_investigador=t.nro_investigador)
    WHERE SUBSTRING (apellido, 1, 1) BETWEEN 'A' and 'J'
    GROUP BY i.tipo_proyecto, i.cod_proyecto,
i.nro_investigador
    HAVING count(*) > 5 ));
```

¿Cuáles son las tablas, los eventos y para qué columnas se deben despertar los triggers?

Nota: las opciones señaladas de manera incorrecta restan del puntaje total



a. INSERT trabaja_en ✓



b. UPDATE OF cod_proyecto ON investigador ✗



c. UPDATE OF cod_area ON trabaja_en ✗



d. UPDATE OF tipo_proyecto ON investigador ✗



e. UPDATE OF nro_investigador ON trabaja_en ✓



f. UPDATE investigador



g. UPDATE OF nombre ON investigador



h. UPDATE OF apellido ON investigador ✓



i. UPDATE OF cod_proyecto ON trabaja_en



j. INSERT investigador ✗



k. UPDATE OF nro_investigador ON investigador ✗



l. UPDATE OF tipo_proyecto ON trabaja_en ✓



m. DELETE investigador ✗



n. UPDATE trabaja_en ✗



o. DELETE trabaja_en ✗

Respuesta incorrecta.

Las respuestas correctas son:

INSERT trabaja_en,

UPDATE OF tipo_proyecto ON trabaja_en,

UPDATE OF cod_proyecto ON trabaja_en,

UPDATE OF nro_investigador ON trabaja_en,

UPDATE OF apellido ON investigador



Información





Debe resolver el siguiente ejercicio teniendo en cuenta lo siguiente:

- Que debe reemplazar en TODO el script donde dice `unc_xxxxxx` por su usuario (`unc_123456` por ejemplo).
Consejo: baje el script, ábralo con un editor de texto (notepad+ + o equivalente) y haga un reemplazar todo de `unc_xxxxxx` por `unc_123456` (siendo 123456 su número de legajo)
- Después de hacer el reemplazo, debe ejecutar el script completo.
- La creación de las tablas y sus datos DEBE formar parte de su solución
- Que la cátedra realizará un Copy & Paste del texto de su respuesta y si Ud. deja una aclaración que no esté como comentario de SQL dentro del texto de la misma, ésto dará error.
- Que los objetos que Ud. declare se denominen en el caso de los trigger `tg_[leg]_[nombre]`, las funciones `fn_[leg]_[nombre]` y los procedimientos `pr_[leg]_[nombre]`;
 - por ejemplo si hiciera una función, un posible nombre sería `fn_123456_manejo_punto_1`
 - `[leg]` es su número de legajo
 - `[nombre]` es el nombre que Ud. crea conveniente para el trigger o función.
- Que el código que Ud. entregue NO debe hacer referencia a un esquema determinado, ni tampoco en la declaración de los objetos (triggers o funciones) suponga que al principio de su código hay un `set search_path =`

Criterios por los cuales el ejercicio no será evaluado (y por lo tanto tendrá 0 punto):

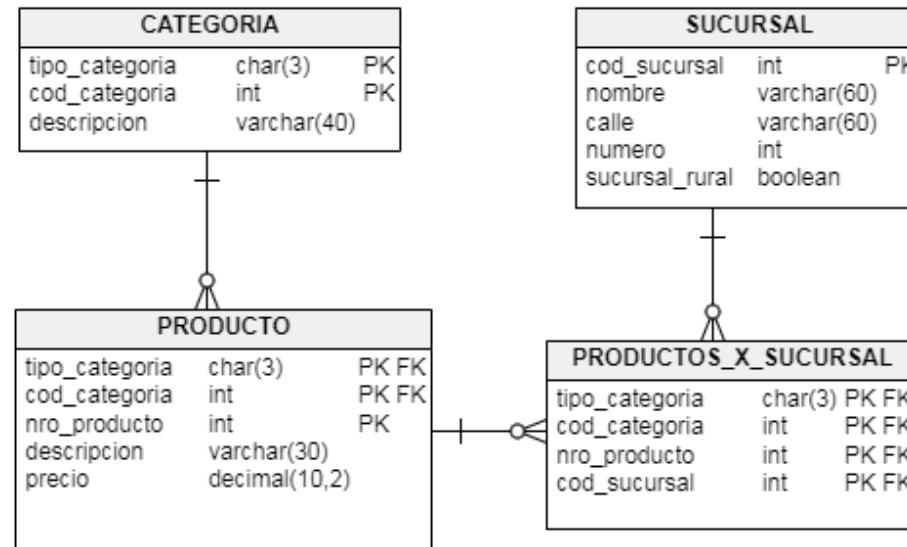
- Si el trigger o la función no compila (no importa qué tipo de error sea ni donde de el error).
- Si el SQL para modificar el esquema es incorrecto (sin importar el error)
- Si llegara a dar error en la ejecución (no compilación)

Pregunta 2

Finalizado

Puntúa como
3,75

Considerando el siguiente esquema - (recuerde que las tablas serán unc_xxxxxx_tabla siendo tabla en este caso, CATEGORIA, PRODUCTO, etc.)



Se necesita mantener actualizado la cantidad de productos por cada categoría y la cantidad de sucursales por cada categoría; los nuevos atributos se deben llamar **cant_prod**, **cant_suc**.

Las tablas y datos se encuentran con [éste link](#) (recuerde que todo lo que Ud. dé como respuesta será ejecutado, realice lo pedido en el orden indicado)

1. Modifique y ejecute el script dado (creación de tablas e inserción de datos)
2. De la/s sentencia/s SQL para modificar la o las tablas que crea adecuado junto con el tipo de dato que corresponde (agregado de los atributos **cant_prod** y **cant_suc**).

3. Provea la sentencia de inicialización de dichos atributos (**cant_prod** y **cant_suc**)
4. Provea el/los trigger/s junto con su/s función/es utilizando los parámetros adecuados para que se comporten lo más eficientemente para que éstos nuevos atributos sigan actualizados (**cant_prod** y **cant_suc**)

```
--Ejercicio 2
ALTER TABLE unc_250605_categoria
ADD cant_prod INT DEFAULT 0,
ADD cant_suc INT DEFAULT 0;

ALTER TABLE unc_250605_producto
ADD cant_prod INT DEFAULT 0;

--Ejercicio 3

UPDATE unc_250605_categoria cat SET cant_prod =
(SELECT COUNT(*)
FROM unc_250605_producto prod
WHERE prod.tipo_categoria = cat.tipo_categoria
AND prod.cod_categoria = cat.cod_categoria);

UPDATE unc_250605_categoria cat SET cant_suc =
(SELECT COUNT(*)
FROM unc_250605_sucursal s
WHERE s.cod_sucursal IN (
SELECT ps.cod_sucursal
FROM unc_250605_productos_x_sucursal ps
WHERE cat.cod_categoria = ps.cod_categoria
));
--ejercicio 4
CREATE OR REPLACE FUNCTION fn_250605_manejo_ejercicio_
RETURNS TRIGGER AS $BODY$


BEGIN

    UPDATE unc_250605_categoria cat
```



```
SET cant_prod = (
    SELECT COUNT(*)
    FROM unc_250605_producto prod
    WHERE prod.tipo_categoria = NEW.tipo_categoria
        AND prod.cod_categoria = NEW.cod_categoria
)
WHERE cat.tipo_categoria = NEW.tipo_categoria
    AND cat.cod_categoria = NEW.cod_categoria;
```



Comentario:

Innecesario agregar el atributo cant_prod en la tabla producto.

Incorrecta la actualización de cant_suc porque se debe actualizar este campo en la tabla categoria.

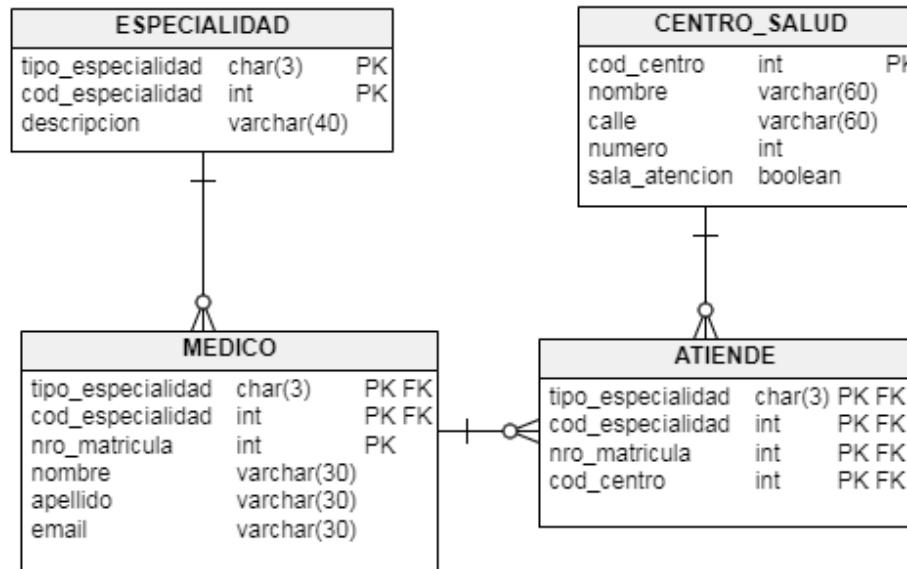
Solo se especifica un trigger. En el evento update no se especifica por que atributos. El trigger da errores cuando se ejecuta el evento delete debido a que no existe el registro new.

Pregunta 3

Correcta

Puntúa como
1,75





Para el esquema de la figura y para las siguientes definiciones de vistas:

```
CREATE OR REPLACE VIEW v_sala_atencion
AS SELECT cod_centro, nombre, calle, numero, sala_atencion
FROM centro_salud
WHERE sala_atencion = TRUE;
```

```
CREATE OR REPLACE VIEW atencion_primaria
AS SELECT * FROM v_sala_atencion
WHERE nombre like '%primaria%'
WITH LOCAL CHECK OPTION;
```

Para las siguientes sentencias ejecutadas de manera independiente señalar las opciones que son verdaderas:

Nota: las opciones señaladas de manera incorrecta restan del puntaje total

- a. La sentencia



```
INSERT INTO atencion_primaria (cod_centro,  
nombre, calle, numero,sala_atencion)
```

```
VALUES (12, 'Atención primaria La Movediza', 'Las  
Azucenas', 1235, FALSE);
```

Procede, inserta datos en la tabla centro_salud
pero no se muestran en la vista v_sala_atencion

- b. La sentencia

```
INSERT INTO v_sala_atencion (cod_centro, nombre,  
calle, numero,sala_atencion)
```

```
VALUES (12, 'Atención primaria La Movediza', 'Las  
Azucenas', 1235, FALSE);
```

No procede, da error



c. La sentencia

```
INSERT INTO atencion_primaria (cod_centro, nombre,  
calle, numero,sala_atencion)
```

```
VALUES (12, 'Atención primaria La Movediza', 'Las  
Azucenas', 1235, FALSE);
```

Procede, no inserta datos en la tabla centro_salud ni
tampoco en la vista v_sala_atencion

d. La sentencia



```
INSERT INTO atencion_primaria (cod_centro,  
nombre, calle, numero,sala_atencion)
```

```
VALUES (12, 'Atención primaria La Movediza', 'Las  
Azucenas', 1235, FALSE);
```

Procede, inserta datos en la tabla centro_salud
pero no se muestran en la vista atencion_primaria

Respuesta correcta

Las respuestas correctas son:

La sentencia

INSERT INTO atencion_primaria (cod_centro, nombre, calle,
numero,sala_atencion)

VALUES (12, 'Atención primaria La Movediza', 'Las Azucenas',
1235, FALSE);

Procede, inserta datos en la tabla centro_salud pero no se
muestran en la vista v_sala_atencion,



La sentencia



```
INSERT INTO atencion_primaria (cod_centro, nombre, calle,  
numero,sala_atencion)
```



VALUES (12, 'Atención primaria La Movediza', 'Las Azucenas',
1235, FALSE);



Procede, inserta datos en la tabla centro_salud pero no se
muestran en la vista atencion_primaria



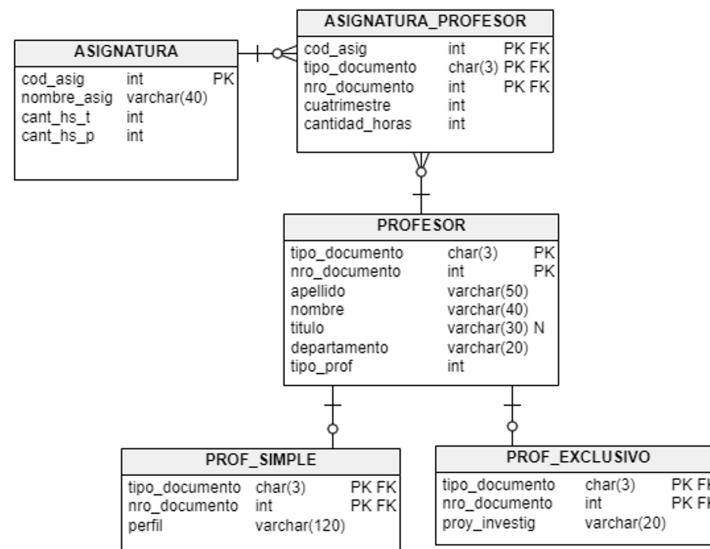
Pregunta 4

Parcialmente
correcta

Puntúa como
2,50

Dado el siguiente esquema y las consultas A y B que podrían o no responder el siguiente requerimiento "Seleccionar el identificador de asignatura de aquellas asignaturas con nombre Base de Datos y que pertenezcan a profesores simples con título de alguna ingeniería ordenados por código de asignatura" marque cuál de las siguientes afirmaciones son correctas.

Notas las opciones incorrectas restan puntos.



A

```
SELECT DISTINCT a.cod_asig
FROM asignatura a JOIN asignatura_profesor ap ON
(a.cod_asig = ap.cod_asig)
JOIN profesor p ON (ap.nro_documento =
p.nro_documento AND ap.tipo_documento =
p.tipo_documento)
JOIN prof_simple ps ON (ps.nro_documento =
p.nro_documento AND ps.tipo_documento =
p.tipo_documento)
WHERE a.nombre_asig = 'Base de Datos'
AND p.tipo_prof = 'S'
AND p.titulo LIKE 'Ing%'
ORDER BY 1;
```

B



```
SELECT a.cod_asig
FROM asignatura a
WHERE a.cod_asig IN (
    SELECT ap.cod_asig
    FROM asignatura_profesor ap
    JOIN (
        SELECT p.nro_documento,
        p.tipo_documento
        FROM profesor p
        WHERE p.titulo LIKE 'Ing%'
        AND p.tipo_prof = 'S'
    ) as t
    ON (ap.nro_documento = t.nro_documento
    AND ap.tipo_documento = t.tipo_documento)
)
AND a.nombre_asig = 'Base de Datos'
ORDER BY 1;
```

- La consulta B se encuentra optimizada bajo reglas de equivalencia.
- La consulta A aplica la regla de equivalencia de proyectar solo los atributos necesario.
- La consulta A NO aplica la regla de equivalencia de ✓



seleccionar antes de ensamblar.

- Utilizando la función EXPLAIN ANALYSE de PostgreSQL ambas consultas A y B brindan el mismo plan de ejecución.
- La consulta A contiene errores léxicos, sintácticos y no está validada.
- La consulta A responde al requerimiento. ✓
- En la consulta A es necesario el operador DISTINCT.
- La consulta A NO contiene errores léxicos, sintácticos y no está validada. ✓
- La consulta B NO aplica la regla de equivalencia de proyectar solo los atributos necesarios.
- La consulta B aplica la regla de equivalencia de proyectar solo los atributos necesarios.
- La consulta B contiene errores léxicos, sintácticos y no está validada.
- Las consultas A y B son equivalentes.
- La consulta B NO se encuentra optimizada bajo reglas de equivalencia.
- La consulta A NO se encuentra optimizada bajo reglas de equivalencia.
- En la consulta B falta el ensamble con la tabla PROF_SIMPLE.
- La consulta B aplica la regla de equivalencia de seleccionar antes de ensamblar. ✓



- Las consultas A y B NO son equivalentes.
- En la consulta A NO es necesario el operador DISTINCT.
- La consulta A aplica la regla de equivalencia de seleccionar antes de ensamblar. ✗
- La consulta A NO responde al requerimiento. ✗
- La consulta B NO responde al requerimiento.
- La consulta A se encuentra optimizada bajo reglas de equivalencia. ✗
- La consulta B responde al requerimiento.
- Utilizando la función EXPLAIN ANALYSE de PostgreSQL ambas consultas A y B NO brindan el mismo plan de ejecución.
- En la consulta B NO es necesario el operador DISTINCT.
- En la consulta B es necesario el operador DISTINCT.
- La consulta A NO aplica la regla de equivalencia de proyectar solo los atributos necesarios. ✓
- La consulta B NO aplica la regla de equivalencia de seleccionar antes de ensamblar. ✗
- En la consulta A es necesario el ensamblaje con la tabla PROF_SIMPLE.
- En la consulta B NO falta el ensamblaje con la tabla PROF_SIMPLE. ✓
- La consulta B NO contiene errores léxicos, sintácticos y ✓



no está validada.

- En la consulta A NO es necesario el ensamblaje con la tabla PROF_SIMPLE.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 7.

Las respuestas correctas son:

Las consultas A y B son equivalentes.,

La consulta A responde al requerimiento.,

La consulta B responde al requerimiento.,

La consulta A NO contiene errores léxicos, sintácticos y no está validada.,

La consulta B NO contiene errores léxicos, sintácticos y no está validada.,

Utilizando la función EXPLAIN ANALYSE de PostgreSQL ambas consultas A y B brindan el mismo plan de ejecución.,

La consulta A NO se encuentra optimizada bajo reglas de equivalencia.,

La consulta B se encuentra optimizada bajo reglas de equivalencia.,

La consulta A NO aplica la regla de equivalencia de seleccionar antes de ensamblar.,

La consulta B aplica la regla de equivalencia de seleccionar antes de ensamblar.,

La consulta A NO aplica la regla de equivalencia de proyectar



solo los atributos necesarios.,

La consulta B aplica la regla de equivalencia de proyectar solo los atributos necesarios.,

En la consulta A es necesario el operador DISTINCT.,

En la consulta B NO es necesario el operador DISTINCT.,

En la consulta A NO es necesario el ensamblaje con la tabla PROF_SIMPLE.,

En la consulta B NO falta el ensamblaje con la tabla PROF_SIMPLE.

Actividad previa

◀ Recuperatorio del 2do
Parcial

Ir a...

Siguiente actividad

Ter Parcial ►

Mantente en contacto

Facultad, Pabellón Central Paraje Arroyo Seco.
Campus Universitario. (B7001BBO) Tandil. Buenos Aires, Argentina

🌐 <https://exa.unicen.edu.ar/>

📞 (+54) (0249) 438-5650 Conmutador: int. 2000



⬇ Descargar la app para dispositivos móviles

✉ moodle@exa.unicen.edu.ar



[Facultad de Ciencias Exactas – UNICEN](#)



Contacto administradores plataforma: E-mail moodle@exa.unicen.edu.ar – Tel. [+54 0249 4385650](tel:+5402494385650) int. 2098

