

[Área personal](#) [Cursos](#) [Exámenes](#) [BD-Tecn-Finales](#) [Finales Julio-Septiembre 2025](#)

[Final del 4 de Septiembre de 2025 Sede Tandil](#)

Comenzado el jueves, 4 de septiembre de 2025, 09:54

Estado Finalizado

Finalizado en jueves, 4 de septiembre de 2025, 12:08

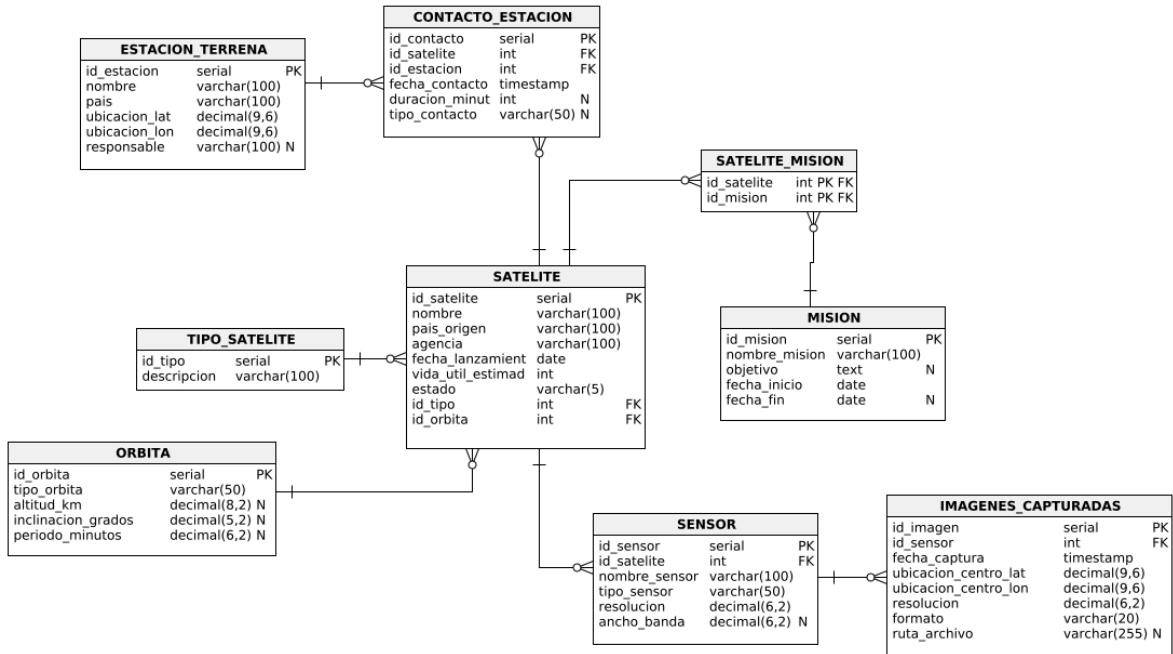
Tiempo 2 horas 13 minutos

empleado

Calificación Sin calificar aún

Información

Dado el siguiente esquema que será utilizado en éste examen: ([SQL](#))



Vertabelo

**Pregunta 1**

Parcialmente correcta

Se puntuó 0,50 sobre 1,00

1a) En el esquema dado se requiere incorporar la siguiente restricción según SQL estándar utilizando el recurso declarativo más restrictivo posible (a nivel de atributo, de tupla, de tabla o general) y utilizando sólo las tablas/atributos necesarios.

- Verificar que cada satélite tenga al menos una misión

Seleccione la/las opción/es que considera correcta/s, de acuerdo a lo solicitado y justifique claramente porqué la/s considera correcta/s

- a. CREATE ASSERTION check\_satelite\_mision CHECK ( NOT EXISTS (  
    SELECT 1 FROM satelite s LEFT JOIN satelite\_mision m USING (id\_satelite)  
    GROUP BY s.id\_satelite HAVING COUNT(id\_mision) < 1 ) ); ✓
- b. ALTER TABLE satelite\_mision ADD CONSTRAINT check\_satelite\_mision CHECK (NOT EXISTS (  
    SELECT 1 FROM satelite\_mision s  
    GROUP BY s.id\_satelite HAVING COUNT(\*) < 1));
- c. CREATE ASSERTION check\_satelite\_mision CHECK (NOT EXISTS (  
    SELECT 1 FROM satelite s LEFT JOIN satelite\_mision m USING (id\_satelite) LEFT JOIN mision (id\_mision)  
    GROUP BY s.id\_satelite HAVING COUNT(id\_mision) < 1));
- d. CREATE ASSERTION check\_satelite\_mision CHECK (EXISTS (   
    SELECT 1 FROM satelite s JOIN satelite\_mision m USING (id\_satelite)  
    GROUP BY s.id\_satelite HAVING COUNT(id\_mision) < 1 ));

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son:

CREATE ASSERTION check\_satelite\_mision CHECK ( NOT EXISTS (   
    SELECT 1 FROM satelite s LEFT JOIN satelite\_mision m USING (id\_satelite)  
    GROUP BY s.id\_satelite HAVING COUNT(id\_mision) < 1 ) );  
CREATE ASSERTION check\_satelite\_mision CHECK (NOT EXISTS (   
    SELECT 1 FROM satelite s LEFT JOIN satelite\_mision m USING (id\_satelite) LEFT JOIN mision (id\_mision)  
    GROUP BY s.id\_satelite HAVING COUNT(id\_mision) < 1));

**Pregunta 2**

Finalizado

Se puntúa como 0 sobre 1,00

**Justifique su/sus elección/es en el punto 1a**

La opcion d no es la correcta porque al pensar en una restricciòn declarativa buscamos lo que no existe.

La c involucra una tabla de mas que no es necesaria que es la de mision, ya que el id\_mision ya se encuentra en la tabla satelite\_mision.

Si bien la opcion b parece la mas adecuada ya que como es una restricciòn de multiples atributos (id\_satelite, id\_mision) que se encuentran en 1 unica tabla (satelite\_mision), no lo es porque el esquema muestra la relacion de que un satelite puede tener asignada 0,N misiones, por ende no se contemplarian aquellos satelites que no tienen misiones asignadas aun.

Por ende la opcion correcta es la opcion a. Chequea agrupando por id\_satelite en la tabla sat que no exista ningun satelite que tenga menos de una mision.

**Comentario:**

Regular

- La c tambi n controlar a ya que es cierto que la tabla est  de mas, pero justamente est  de mas. Lo importante son los LEFT join
- No queda claro que sabe dos cosas importantes, que es el LEFT Join y el count(...) ya que al contar nulos que pasa ?

**Pregunta 3**

Finalizado

Se puntuá como 0 sobre 1,00

1b) En el esquema dado se requiere incorporar la siguiente restricción según SQL estándar utilizando el recurso declarativo más restrictivo posible (a nivel de atributo, de tupla, de tabla o general) y utilizando solo las tablas/atributos necesarios.

**Las fechas de contacto de cada estación deben ser posteriores a la fecha de lanzamiento de cada satélite.**

Resuelva según lo solicitado y justifique el tipo de chequeo utilizado.

```
CREATE ASSERTION chk_fechas
CHECK (NOT EXISTS (
SELECT 1
FROM satelite s
JOIN contacto_estacion ce ON s.id_satelite = ce.id_satelite
WHERE ce.fecha_contacto > s.fecha_lanzamiento
));
```

Comentario:

Bien

**Pregunta 4**

Finalizado

Se puntuá como 0 sobre 1,00

Justifique su elección en el punto anterior (1b)

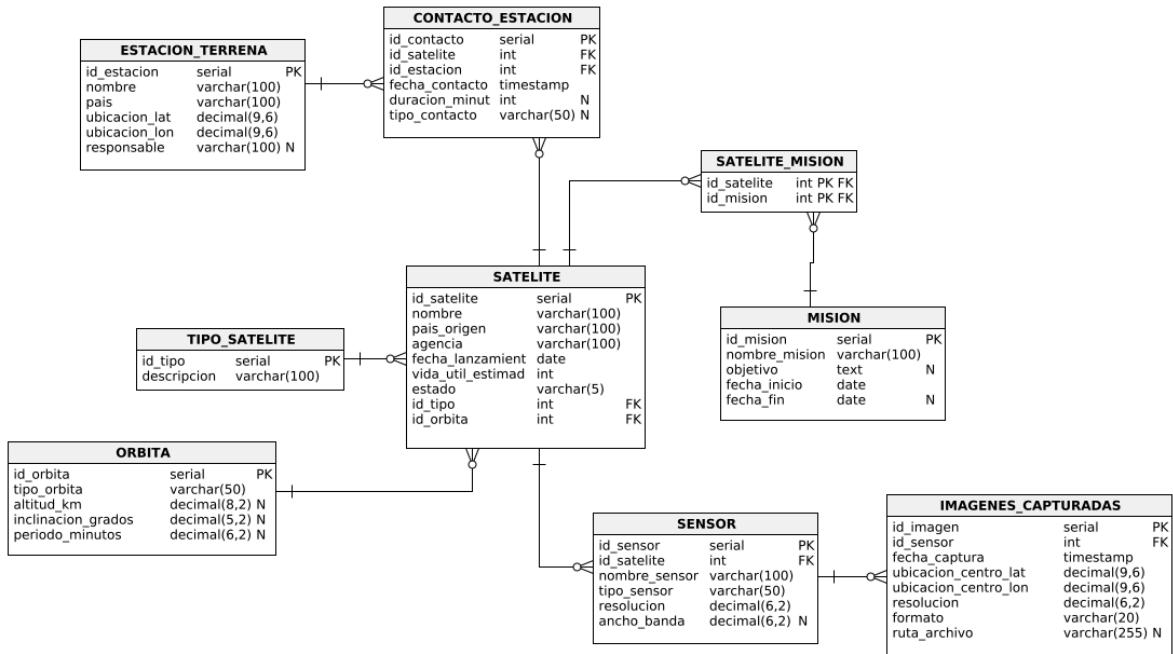
El tipo de chequeo es general de base de datos ya que los atributos que se deben controlar (fecha\_contacto y fecha\_lanzamiento) pertenecen a dos tablas distintas del esquema (contacto\_estacion, satelite)

Comentario:

Bien

Información

Dado el siguiente esquema que será utilizado en éste examen: ([SQL](#))



### Pregunta 5

Finalizado

Se puntúa como 0 sobre 1,00

2a) Sobre el esquema dado se requiere definir la siguiente vista, de manera que resulte automáticamente actualizable en PostgreSQL, siempre que sea posible:

- VI: que contenga todos los datos de los satélites que aún no han hecho contacto con ninguna estación terrena

```

CREATE V1 AS
SELECT *
FROM satelite s
WHERE s.id_satelite NOT IN (SELECT id_satelite
                            FROM contacto_estacion);
  
```

Comentario:

Bien

**Pregunta 6**

Correcta

Se puntuó 1,00 sobre 1,00

2b) Considerando la siguiente definición para V1, seleccione la/s afirmación/es que considere correcta/s respecto de esta vista y justifíquela/s claramente.

```
CREATE VIEW V1 AS
```

```
SELECT * FROM satélite s
```

```
WHERE NOT EXISTS (SELECT 1 FROM tipo_satélite t WHERE s.id_tipo = t.id_tipo AND upper(descripción) like '%RECONOCIMIENTO%')
```

- a. Ninguna de las opciones es correcta
- b. No es posible reformular la consulta para seleccionar todos los satélites que son del tipo reconocimiento (y sea automáticamente actualizable)
- c. No resulta automáticamente actualizable en PostgreSQL
- d. Contiene todos los datos de los satélites de reconocimiento
- e. Es automáticamente actualizable en PostgreSQL ✓
- f. Para seleccionar todos los satélites que son del tipo reconocimiento hay que reformular la consulta cambiando el NOT EXISTS por EXISTS ✓
- g. No está correctamente correlacionada la consulta con la subconsulta
- h. Está correctamente correlacionada la consulta con la subconsulta ✓

Respuesta correcta

Las respuestas correctas son:

Está correctamente correlacionada la consulta con la subconsulta,

Es automáticamente actualizable en PostgreSQL,

Para seleccionar todos los satélites que son del tipo reconocimiento hay que reformular la consulta cambiando el NOT EXISTS por EXISTS

## Pregunta 7

Finalizado

Se puntuó como 0 sobre 1,00

2c) Sobre el esquema dado se requiere definir la siguiente vista, de manera que resulte automáticamente actualizable en PostgreSQL, siempre que sea posible, y que se verifique que no haya migración de tuplas de la vista. Resuelva según lo solicitado y justifique su solución.

- V2: Listar los satélites lanzados después del año 2024 que no poseen sensores.

```
CREATE V2 AS
SELECT *
FROM satelite s
WHERE EXTRACT(YEAR FROM fecha_lanzamiento) > 2024
AND s.id_satelite NOT IN (SELECT id_satelite FROM sensor)
WITH CHECK OPTION;
```

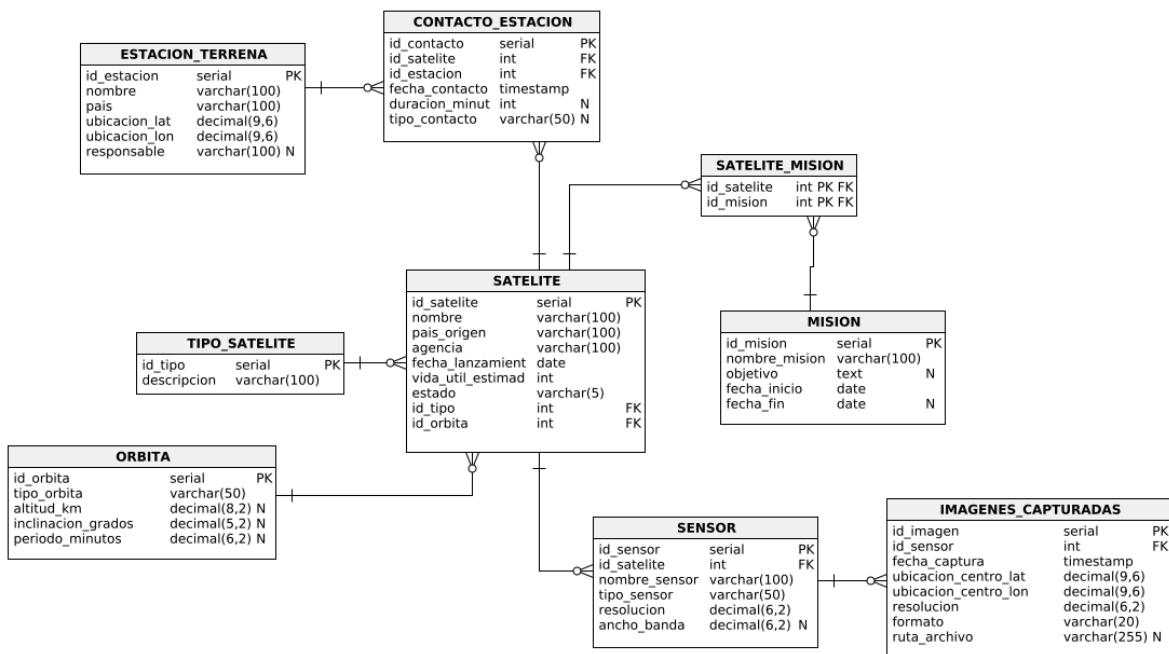
Comentario:

Bien

- Pero sería create VIEW no create

## Información

Dado el siguiente esquema que será utilizado en éste examen: ([SQL](#))



## Pregunta 8

Finalizado

Se puntuó 0,00 sobre 1,00

- 3) Para el esquema dado, es necesario consultar los contactos de un satélite realizados después de una determinada fecha, mostrando: estación, fecha del contacto y duración; el satélite y la fecha son datos que se aportan. Resuelva con el recurso que considere más conveniente.

```
CREATE V3 AS
SELECT id_estacion, fecha_contacto, duracion_minutos
FROM contacto_estacion
WHERE fecha_contacto > fecha_aportada
AND id_estacion = id_estacion_aportado;
```

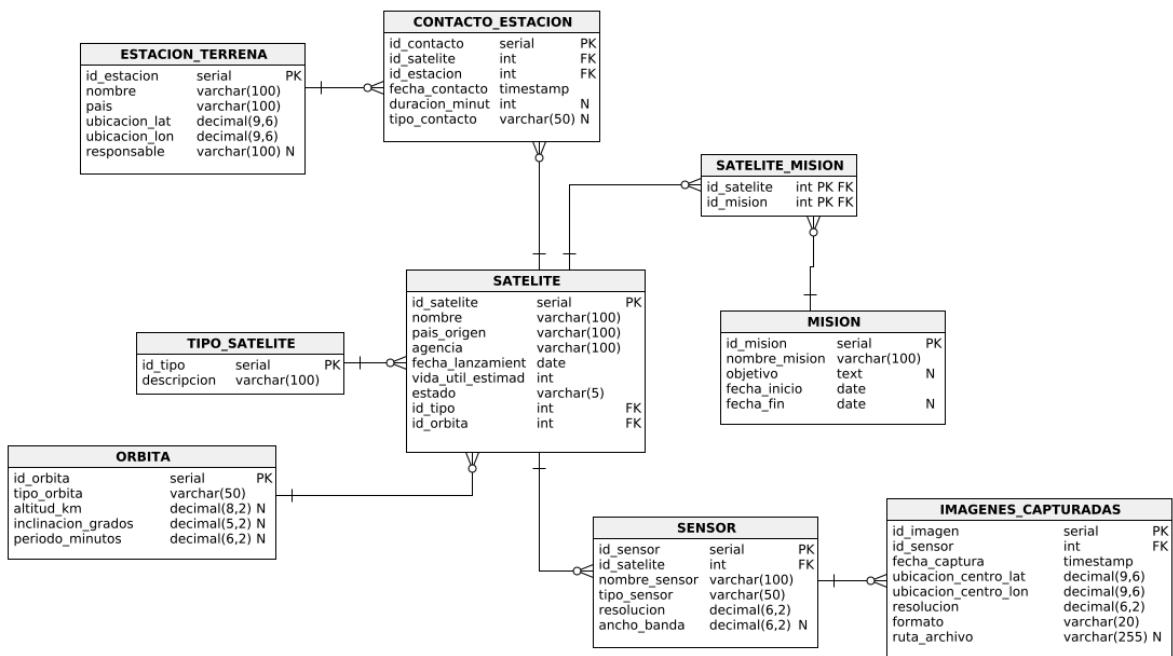
Comentario:

Mal

- Debería haber realizado un stored procedure / función que devuelva la consulta
- Una si quiero otro satélite u otra fecha tengo que cambiar la vista

## Información

Dado el siguiente esquema que será utilizado en éste examen: ([SQL](#))



## Pregunta 9

Finalizado

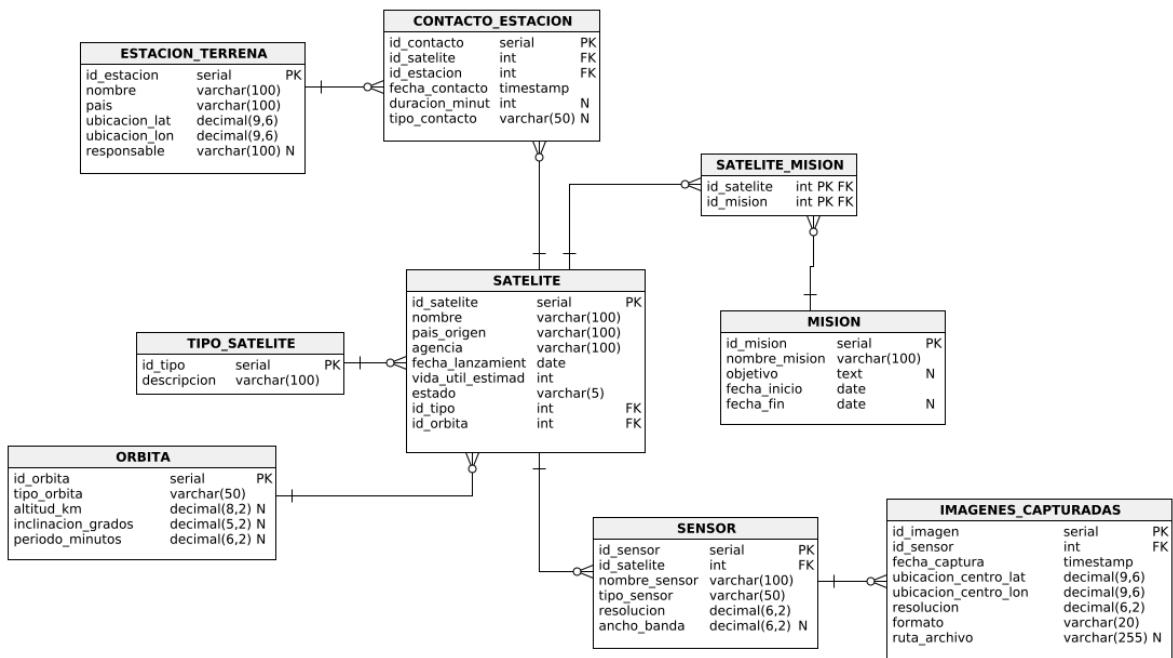
Se puntuó como 0 sobre 1,00

- 4) Listar todas las imágenes capturadas mostrando el id\_imagen, fecha\_captura, id\_sensor y un número de fila ordenado por fecha de captura dentro de cada sensor además de la fecha de la primera imagen tomada por ese sensor.

```
CREATE V1 AS
SELECT id_imagen, fecha_captura, id_sensor,
RANK() OVER (PARTITION BY fecha_captura DESC) AS ranking,
MIN(fecha_captura) AS fecha_primer_imagen
FROM imagenes_capturadas
GROUP BY id_imagen, id_sensor;
```

## Información

Dado el siguiente esquema que será utilizado en éste examen: ([SQL](#))



**Pregunta 10**

Finalizado

Se puntuá como 0 sobre 1,00

**5a)** Es posible plantear con una sentencia declarativa un control que no permita agregar imágenes a satélites que están activos? Si su respuesta es positiva plantee el control; caso contrario justifique porque.

Si se puede plantear una sentencia declarativa de tipo general a la base de datos ya que el control involucra atributos que se encuentran en distintas tablas del esquema que son sensor e imagenes\_capturadas.

Se deben unir estas tablas con JOIN's y chequear que cada vez que se inserta en imagenes\_cap una nueva fila, que esa imagen no pertenezca a un satelite que tiene como estado "activo".

Y tambien se debe chequear que si se modifica el id\_sensor de imagenes\_capturadas, que ese id\_sensor no pertenezca ahora a un id\_satelite que este activo.

```
CREATE ASSERTION chk_img_sat_activos
CHECK (NOT EXISTS (
```

Comentario:

Solución correcta.

**Pregunta 11**

Finalizado

Se puntuá como 0 sobre 1,00

**5b)** En caso de que sea posible plantee de manera procedural lo requerido en el punto 5.a)

```
CREATE OR REPLACE FUNCTION fn_img_sat_activos()
RETURNS TRIGGER AS
$$
DECLARE
v_estado satelite.estado%type;
BEGIN

SELECT s.estado INTO v_estado
FROM satelite s
JOIN sensor sen ON sen.id_satelite = s.id_satelite
```

Comentario:

Debería haber un trigger en cada tabla especificada en la declarativa.

**Actividad previa**[◀ Avisos](#)[Ir a...](#)

## 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](tel:+5402494385650)

✉️ [moodle@exa.unicen.edu.ar](mailto:moodle@exa.unicen.edu.ar)



📱 Descargar la app para dispositivos móviles

[Facultad de Ciencias Exactas – UNICEN](#)

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