

[Área personal](#)[Mis cursos](#)[BD-TUDAI-tand-IC-2025](#)[Exámenes](#)[Parcial Demo Bases de Datos](#)

Comenzado el sábado, 7 de junio de 2025, 12:20

Estado Finalizado

Finalizado en sábado, 7 de junio de 2025, 13:23

**Tiempo
empleado** 1 hora 3 minutos

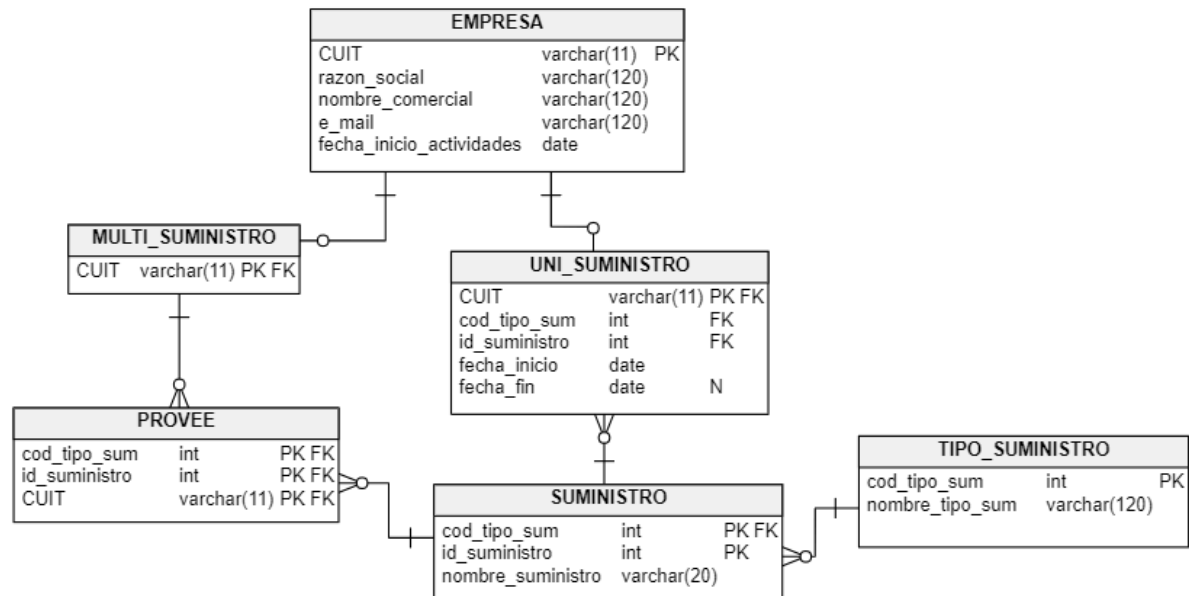
Pregunta 1

Finalizado

Se puntúa como 0 sobre 1,00

Tiempo estimado 15 min

Dado el esquema de la figura:



Si las foreign key fueran las que se indican debajo y se encuentran cargados los siguientes datos:

TIPO_SUMINISTRO		
cod_tipo_sum	nombre_tipo_sum
A	nombre1	
B	nombre2	
C	nombre3	
D	nombre4	
E	nombre5	

SUMINISTRO		
cod_tipo_sum	id_suministro	nombre_suminis
A	2	suministro1
B	3	suministro2
A	1	suministro3
D	1	suministro4
C	4	suministro5

PROVEE		
cod_tipo_sum	id_suministro	CUIT
A	1	1-26356-0
B	3	1-39568-2
C	4	2-39564-5
A	2	1-40592-3
C	4	1-39568-2

FK_SUMINISTRO_TIPO_SUMINISTRO
 ON UPDATE CASCADE
 ON DELETE CASCADE
 FK_PROVEE_SUMINISTRO
 ON UPDATE RESTRICT
 ON DELETE RESTRICT

¿Cuál/es de las siguientes operaciones proceden y cuál/es fallan?

UPDATE SUMINISTRO SET id_suministro = '2' WHERE cod_tipo_sum = 'B';

Falla

DELETE FROM SUMINISTRO WHERE id_suministro = 3;

Falla

DELETE FROM TIPO_SUMINISTRO WHERE cod_tipo_sum = 'B';

Procede

UPDATE TIPO_SUMINISTRO SET cod_tipo_sum ='F' WHERE cod_tipo_sum ='B';

Falla

DELETE FROM PROVEE WHERE CUIT = '1-26356-0';

Falla

UPDATE PROVEE SET id_suministro = 1 WHERE CUIT = '1-40592-3';

Falla

DELETE FROM TIPO_SUMINISTRO WHERE cod_tipo_sum ='D';

Procede

Respuesta parcialmente correcta.

La respuesta correcta es:

UPDATE SUMINISTRO SET id_suministro ='2' WHERE cod_tipo_sum ='B'; → Falla,

DELETE FROM SUMINISTRO WHERE id_suministro = 3; → Falla,

DELETE FROM TIPO_SUMINISTRO WHERE cod_tipo_sum ='B'; → Falla,

UPDATE TIPO_SUMINISTRO SET cod_tipo_sum ='F' WHERE cod_tipo_sum ='B';

→ Falla,

DELETE FROM PROVEE WHERE CUIT = '1-26356-0'; → Procede,

UPDATE PROVEE SET id_suministro = 1 WHERE CUIT = '1-40592-3';

→ Procede,

DELETE FROM TIPO_SUMINISTRO WHERE cod_tipo_sum ='D';

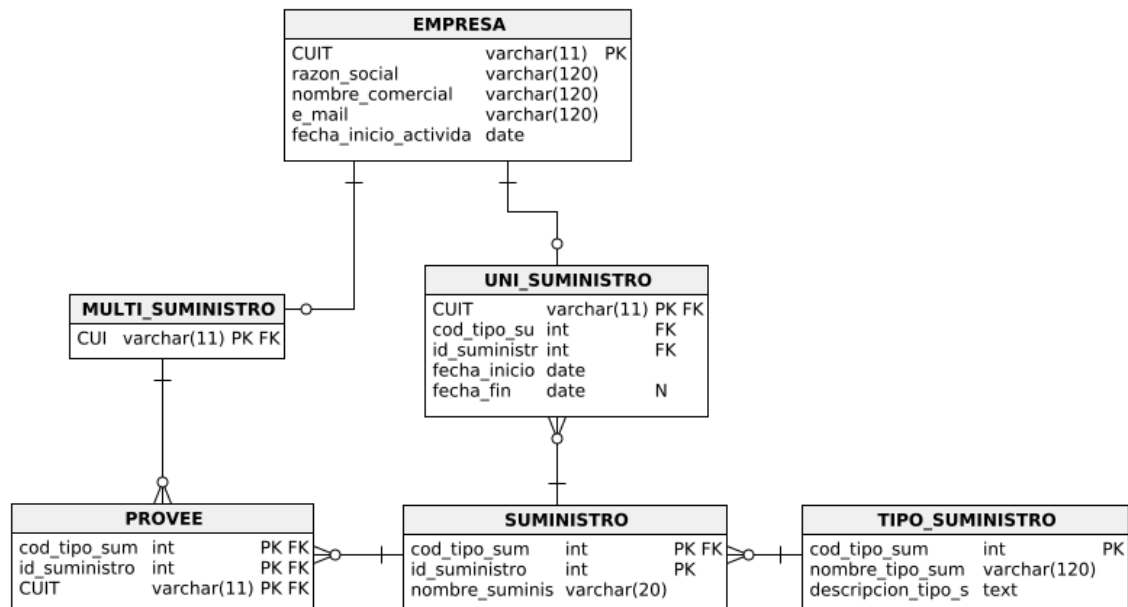
→ Procede

Pregunta 2

Finalizado

Se puntúa como 0 sobre 1,00

Tiempo estimado 45 min

Dado el siguiente esquema - [Script de creación de Tablas](#)

1) De la sentencia declarativa mas restrictiva que controle lo siguiente:

- Un mismo suministro, no puede ser ofrecido por una empresa en sus dos modalidades

2) En caso de que no pueda ser implementado en PostgreSQL declarativamente, de la solución procedural mas eficiente.

```

/* 1) De la sentencia declarativa mas restrictiva que controle lo siguiente:
- Un mismo suministro, no puede ser ofrecido por una empresa en sus dos modalidades */
-- CHECK Global
/*
CHECK ASSERTION suministro_modalidad
CHECK (NOT EXISTS(
    SELECT 1
    FROM PROVEE p
    JOIN UNI_SUMINISTRO us ON (p.cod_tipo_sum = us.cod.cod_tipo_sum AND p.id_suministro = us
)); */

-- 2) En caso de que no pueda ser implementado en PostgreSQL declarativamente, de la solución
CREATE FUNCTION fn_suministro_no_duplicado()
RETURNS TRIGGER AS $$
BEGIN
    IF (TG_TABLE_NAME = 'PROVEE') THEN
        IF EXISTS (
            SELECT 1 FROM UNI_SUMINISTRO
            WHERE cod_tipo_sum = NEW.cod_tipo_sum
            AND id_suministro = NEW.id_suministro)
            THEN RAISE EXCEPTION 'ERROR! Ya existe el Suministro % de tipo % contratado en moda
        END IF;
    ELSEIF (TG_TABLE_NAME = 'UNI_SUMINISTRO') THEN
        IF EXISTS (
            SELECT 1 FROM PROVEE
            WHERE cod_tipo_sum = NEW.cod_tipo_sum
            AND id_suministro = NEW.id_suministro)
            THEN RAISE EXCEPTION 'ERROR! Ya existe el Suministro % de tipo % contratado en moda
        END IF;
    END IF;
END IF;

```

```

1)
---- Si se interpreta que un mismo suministro no puede ser ofrecido por 2 modalidades diferentes:
CHECK ( NOT EXISTS( SELECT 1
    FROM provee
    WHERE (cod_tipo_sum, id_suministro)
    IN (SELECT cod_tipo_sum, id_suministro
        FROM uni_suministro)));
--- Si se interpreta que una misma empresa no puede ofrecer el mismo suministro en 2 modalidades diferentes:
CHECK ( NOT EXISTS( SELECT 1
    FROM provee
    WHERE (CUIT, cod_tipo_sum, id_suministro)
    IN (SELECT CUIT, cod_tipo_sum, id_suministro
        FROM uni_suministro)));

2)
create function fn_uni_sum() returns trigger as $$
begin
    if (exists (select 1
        from uni_suministro u
        where u.cuit = new.cuit
            and u.id_suministro = new.id_suministro
            and u.cod_tipo_sum = new.cod_tipo_sum)) then
        raise exception 'Error';
    end if;
    return new;
end; $$ language plpgsql;

```

```

create trigger tr_biu_provee
before insert or update on provee
for each row
execute function fn_uni_sum();

```

```
create function fn_provee() returns trigger as $$
begin
    if (exists (select 1
                from provee p
                where p.cuit = new.cuit
                    and p.id_suministro = new.id_suministro
                    and p.cod_tipo_sum = new.cod_tipo_sum)) then
        raise exception 'Error';
    end if;
    return new;
end; $$ language plpgsql;
```

```
create trigger tr_biu_unisum
```

```
before insert or update of cuit, cod_tipo_sum, id_suministro on uni_suministro
for each row
execute function fn_provee();
```

Pregunta 3

Finalizado

Se puntúa como 0 sobre 1,00

Utilizando el esquema `unc_esq_peliculas`. Contar todos los departamentos que no pertenezcan a ninguna ciudad y que la calle empiece con la letra C.

Tiempo estimado 20 min

- ☐ a. Ninguna de las opciones es correcta
- ☐ b. 1054
- ☐ c. No hay registros que cumplan con este requerimiento
- ☐ d. 236
- ☒ e. 0

Respuesta correcta

El SQL que la resuelve es:

```
SELECT count(*)
FROM unc_esq_peliculas.departamento
WHERE id_ciudad IS NULL
and calle like 'C%';
```

La respuesta correcta es:

0

Pregunta 4

Finalizado

Se puntúa como 0 sobre 1,00

Utilizando el esquema unc_esq_voluntario. Cual/Cuales son los coordinadores(nombre) que han tenido a cargo la mayor cantidad de voluntarios que hayan realizado cualquier tarea terminada en CLERK

Tiempo estimado 20 min

- ☒ a. Ninguna de las opciones es correcta
- ☐ b. Luisa
- ☐ c. Den, Luisa
- ☐ d. Laura, Guy
- ☐ e. Payam, Matthew, Adam, Kevin, Shanta

Respuesta incorrecta.

SQL que lo resuelve:

```
select coor.nro_voluntario, coor.nombre as "Coordinador",count(V.nombre)
from voluntario V
join voluntario coor on V.id_coordinador=coor.nro_voluntario
join tarea t on V.id_tarea = t.id_tarea
where t.id_tarea like '%CLERK'
group by coor.nro_voluntario, coor.nombre
order by 3 desc
```

La respuesta correcta es:

Payam, Matthew, Adam, Kevin, Shanta

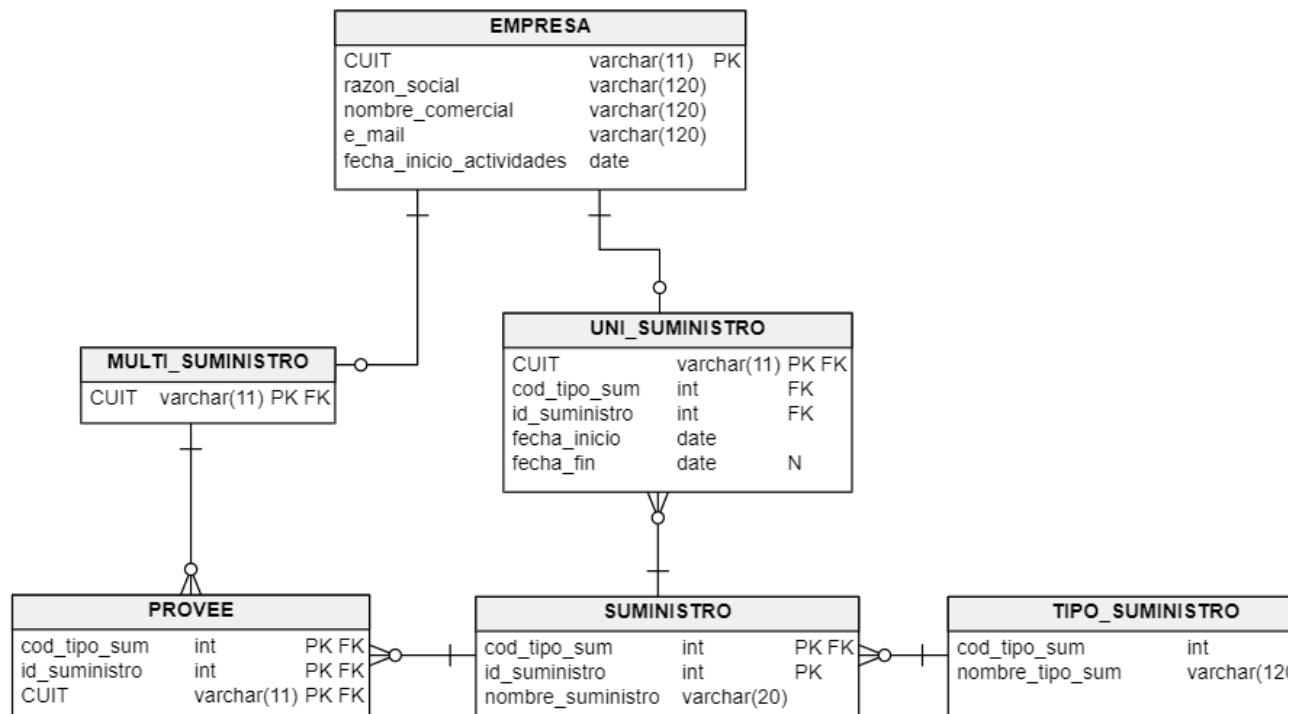
Pregunta 5

Finalizado

Se puntúa como 0 sobre 1,00

Tiempo estimado 25 min

Considere el siguiente esquema (script creación solo de tablas) correspondiente a un sistema de contratación de servicios, cuyo diagrama en Vertabelo es el siguiente:



Considere las siguientes sentencias SQL de declaración de claves primarias (PK) y claves extranjeras (FK) sobre el esquema dado:

```

-- Primary keys-- Primary keys
ALTER TABLE EMPRESA ADD CONSTRAINT PK_EMPRESA PRIMARY KEY (CUIT);
ALTER TABLE MULTI_SUMINISTRO ADD CONSTRAINT PK_MULTI_SUMINISTRO PRIMARY KEY (CUIT);
ALTER TABLE TIPO_SUMINISTRO ADD CONSTRAINT PK_TIPO_SUMINISTRO PRIMARY KEY (cod_tipo_sum);
ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT PK_UNI_SIMINISTRO PRIMARY KEY (CUIT);

-- foreign keys
ALTER TABLE PROVEE ADD CONSTRAINT FK_PROVEE_MULTI_SUM
    FOREIGN KEY (CUIT)
    REFERENCES MULTI_SUMINISTRO (CUIT);

ALTER TABLE SUMINISTRO ADD CONSTRAINT FK_SUMINISTRO_TIPO_SUMINISTRO
    FOREIGN KEY (cod_tipo_sum)
    REFERENCES TIPO_SUMINISTRO (cod_tipo_sum);

ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT FK_UNI_SUM_SERVICIO
    FOREIGN KEY (id_suministro, cod_tipo_sum)
    REFERENCES SUMINISTRO (id_suministro, cod_tipo_sum);
  
```

Escriba las sentencias SQL de **declaración de claves primarias (PK)**, **claves alterativas (AK)** y **claves extranjeras (FK)** que faltan teniendo en cuenta los siguiente:

- Deben definirse las acciones referenciales para que cada vez que se elimine o modifique (PK) de empresa debe hacerse también de las tablas subtipo de la jerarquía.
- Los nombres de los tipos de suministro deben ser únicos.

/* Pregunta 5

Escriba las sentencias SQL de declaración de claves primarias (PK), claves alternativas (AK) y claves extranjeras (FK) que faltan teniendo en cuenta los siguiente:

a) Deben definirse las acciones referenciales para que cada vez que se elimine o modifique (PK) de empresa debe hacerse también de las tablas subtipo de la jerarquía.

b) Los nombres de los tipos de suministro deben ser únicos.

En TIPO_SUMINISTRO esta línea CONSTRAINT AK_TIPO_SUMINISTRO UNIQUE (nombre_tipo_sum) NOT DEFERRABLE INITIALLY IMMEDIATE

cumple ya esta condición. */

-- Primary keys

ALTER TABLE EMPRESA ADD CONSTRAINT PK_EMPRESA PRIMARY KEY (CUIT);

ALTER TABLE MULTI_SUMINISTRO ADD CONSTRAINT PK_MULTI_SUMINISTRO PRIMARY KEY (CUIT);

ALTER TABLE TIPO_SUMINISTRO ADD CONSTRAINT PK_TIPO_SUMINISTRO PRIMARY KEY (cod_tipo_sum);

ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT PK_UNI_SUMINISTRO PRIMARY KEY (CUIT);

-- Resuelto

ALTER TABLE PROVEE ADD CONSTRAINT PK_PROVEE PRIMARY KEY (id_suministro, cod_tipo_sum, CUIT);

ALTER TABLE SUMINISTRO ADD CONSTRAINT PK_SUMINISTRO PRIMARY KEY (id_suministro, cod_tipo_sum);

-- Foreign keys

-- Inicio resuelto

ALTER TABLE PROVEE ADD CONSTRAINT FK_PROVEE_SUMINISTRO

FOREIGN KEY (id_suministro, cod_tipo_sum)

REFERENCES SUMINISTRO (id_suministro, cod_tipo_sum);

-- Fin resuelto

ALTER TABLE PROVEE ADD CONSTRAINT FK_PROVEE_MULTI_SUM

FOREIGN KEY (CUIT)

REFERENCES MULTI_SUMINISTRO (CUIT);

ALTER TABLE SUMINISTRO ADD CONSTRAINT FK_SUMINISTRO_TIPO_SUMINISTRO

FOREIGN KEY (cod_tipo_sum)

REFERENCES TIPO_SUMINISTRO (cod_tipo_sum);

-- Inicio resuelto

ALTER TABLE MULTI_SUMINISTRO ADD CONSTRAINT FK_MULTI_SUM_EMPRESA

FOREIGN KEY (CUIT)

REFERENCES EMPRESA (CUIT)

ON UPDATE CASCADE

ON DELETE CASCADE;

ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT FK_UNI_SUM_EMPRESA

FOREIGN KEY (CUIT)

REFERENCES EMPRESA (CUIT)

ON UPDATE CASCADE

ON DELETE CASCADE;

-- Fin resuelto

ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT FK_UNI_SUM_SUMINISTRO

FOREIGN KEY (id_suministro, cod_tipo_sum)

REFERENCES SUMINISTRO (id_suministro, cod_tipo_sum);

-- Primary keys

ALTER TABLE PROVEE ADD CONSTRAINT PK_PROVEE PRIMARY KEY (id_suministro,cod_tipo_sum,CUIT);

ALTER TABLE SUMINISTRO ADD CONSTRAINT PK_SUMINISTRO PRIMARY KEY (id_suministro,cod_tipo_sum);

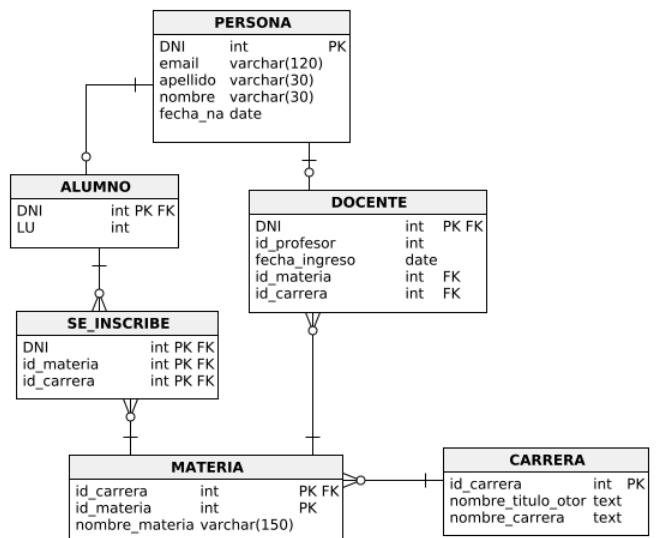
```
-- foreign keys
ALTER TABLE MULTI_SUMINISTRO ADD CONSTRAINT FK_MULTI_SUM_EMPRESA
    FOREIGN KEY (CUIT)
    REFERENCES EMPRESA (CUIT)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
ALTER TABLE UNI_SUMINISTRO ADD CONSTRAINT FK_UNI_SUM_EMPRESA
    FOREIGN KEY (CUIT)
    REFERENCES EMPRESA (CUIT)
    ON DELETE CASCADE
    ON UPDATE CASCADE;
ALTER TABLE PROVEE ADD CONSTRAINT FK_PROVEE_SUMINISTRO
    FOREIGN KEY (id_suministro, cod_tipo_sum)
    REFERENCES SUMINISTRO (id_suministro, cod_tipo_sum);
```

```
ALTER TABLE TIPO_SUMINISTRO ADD CONSTRAINT AK_TIPO_SUMINISTRO UNIQUE (nombre_tipo_sum);
```

Pregunta 6

Finalizado

Se puntúa como 0 sobre 1,00



Tiempo estimado 15 min

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

```

CREATE OR REPLACE VIEW v_gmail
AS
SELECT DNI, email, apellido, nombre, fecha_nac
FROM persona
WHERE email like '%gmail%';

CREATE OR REPLACE VIEW v_gmail_mayor
AS
SELECT *
FROM v_gmail
WHERE DNI > 23456789
WITH LOCAL CHECK OPTION;

CREATE OR REPLACE VIEW v_gmail_parcial
AS
SELECT *
FROM v_gmail_mayor
WHERE apellido like 'Bet%'
WITH CASCADED CHECK OPTION;
  
```

Para las siguientes sentencias ejecutadas de manera independiente señalar las opciones que son **FALSAS**. Nota: Las respuestas incorrectas restan del puntaje total. Tenga cuidado al cortar y pegar las sentencias con las comillas simples ''.

☒ a.

```

INSERT INTO v_gmail_mayor
(DNI, email, apellido, nombre,
fecha_nac)
VALUES (33456789,
'cc@hotmail.com', 'Beta', 'J',
to_date('20170103', 'YYYYMMDD'))
NO Procede, da error.
  
```

Es falsa, ya que la operación procede insertando en la tabla base si error alguno al cumplir con el WHERE de v_gmail_mayor (LOCAL CHECK OPTION) y no evaluar el WHERE de v_gmail por no tener opción de chequeo en su definición.

- ☒ b. INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))
NO Procede, porque no cumple con la condición de la vista v_gmail
Es falsa ya que la vista v_gmail_mayor no tiene opción de chequeo y por lo tanto no se comprueba la condición de su WHERE. La operación procede y la tupla se inserta en la tabla base.
- ☐ c. INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))
Procede, inserta los datos en la tabla persona pero no se muestran en la vista v_gmail
- ☒ d. INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@gmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))
Procede, inserta los datos en la tabla persona y se muestran en la vista v_gmail y en la vista v_gmail_mayor
Es Verdadera porque la operación procede e inserta los datos en la tabla persona al cumplir con el WHERE de v_gmail_mayor (LOCAL CHECK OPTION) y no evaluar el WHERE de v_gmail por no tener opción de chequeo en su definición. Luego, la tupla se muestra en la vista v_gmail porque email contiene "gmail", y luego se muestra en la vista v_gmail_mayor porque el dni 33 millones es mayor a 23456789
- ☐ e. INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))
Procede, inserta los datos en la tabla persona pero no se muestran en la vista v_gmail_mayor
- ☒ f. INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))
Procede, inserta los datos en la tabla persona y se muestran en todas las vistas
Es Falsa, ya que la operación procede e inserta los datos en la tabla persona** pero la tupla no se muestra en la vista v_gmail porque email contiene "hotmail" y no "gmail", por ende no se muestra en ninguna de las vistas que dependen de ella.
** La tupla se inserta en la tabla base porque cumple con el WHERE de v_gmail_mayor (LOCAL CHECK OPTION) y no evalúa el WHERE de v_gmail al no tener opción de chequeo en su definición.

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son:

INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)

VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))

Procede, inserta los datos en la tabla persona y se muestran en todas las vistas

```
INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))

NO Procede, porque no cumple con la condición de la vista v_gmail
```

```
INSERT INTO v_gmail_mayor (DNI, email, apellido, nombre, fecha_nac)
VALUES (33456789, 'cc@hotmail.com', 'Beta', 'J', to_date('20170103','YYYYMMDD'))

NO Procede, da error.
```

Actividad previa

◀ [Práctico 9 - SQL Avanzado](#)

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](tel:+5402494385650) Conmutador: int. 2000

✉ 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 – Tel. [+54 0249 4385650](tel:+5402494385650) int. 2098