

Opcional No. 1

Descripción general

A continuación se presentan los lineamientos para la presentación de un examen tipo parcial para el curso de base de datos

Introducción

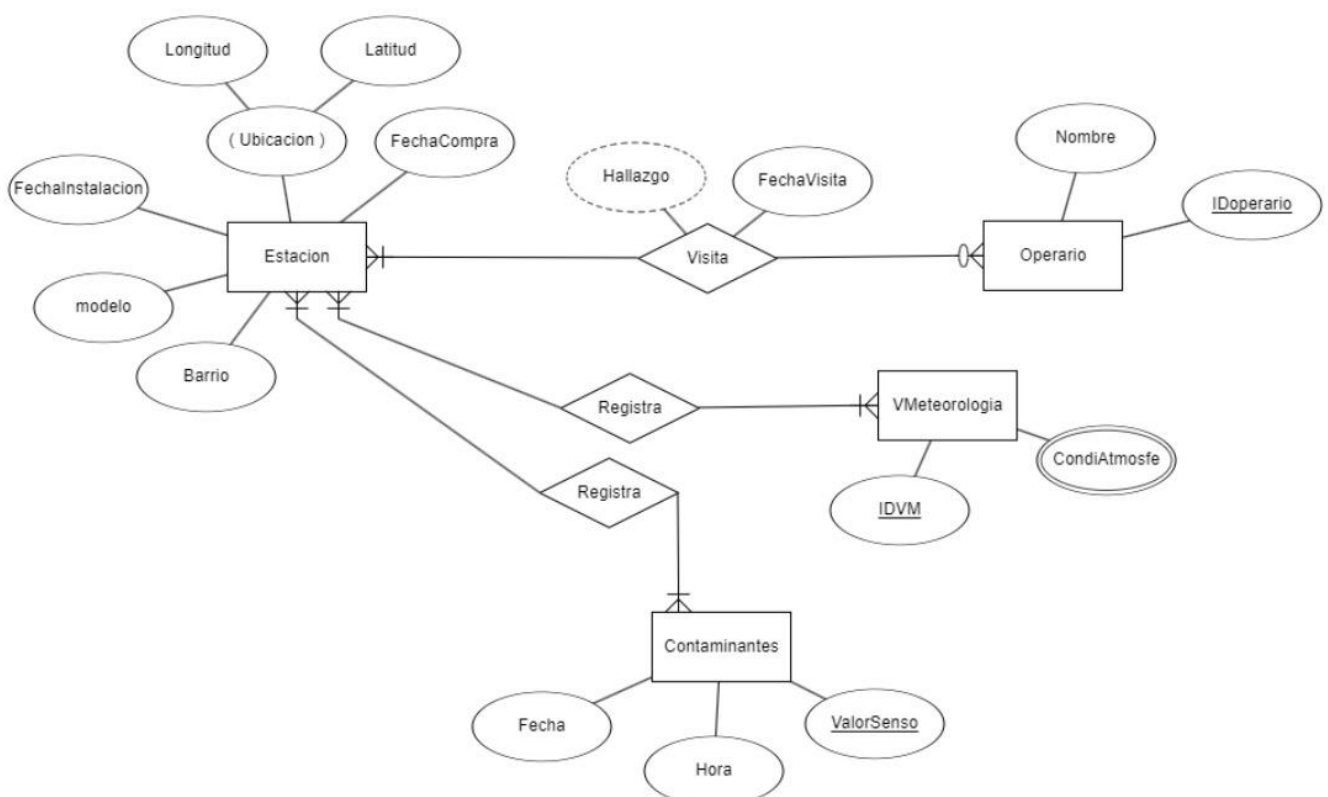
En la ciudad de Cali, el Sistema de Vigilancia de la Calidad del Aire de Santiago de Cali (SVCASC) opera bajo la coordinación y administración del Grupo de Calidad del Aire del DAGMA. El SVCASC tiene como objetivo medir continuamente los contaminantes y la meteorología en diferentes puntos de la ciudad.

Este sistema posee una red de seis (6) estaciones de monitoreo de las que se conoce el modelo, la fecha de compra, la fecha de instalación y un nombre asociado al barrio donde ha sido ubicada. Además, se tiene información de su ubicación en términos de latitud y longitud. Para que todo funcione correctamente, se ha dispuesto a tres (3) operarios encargados del mantenimiento. Ellos pueden visitar una o más estaciones y se deja un registro de la fecha de visita y si hubo o no un hallazgo o novedad.

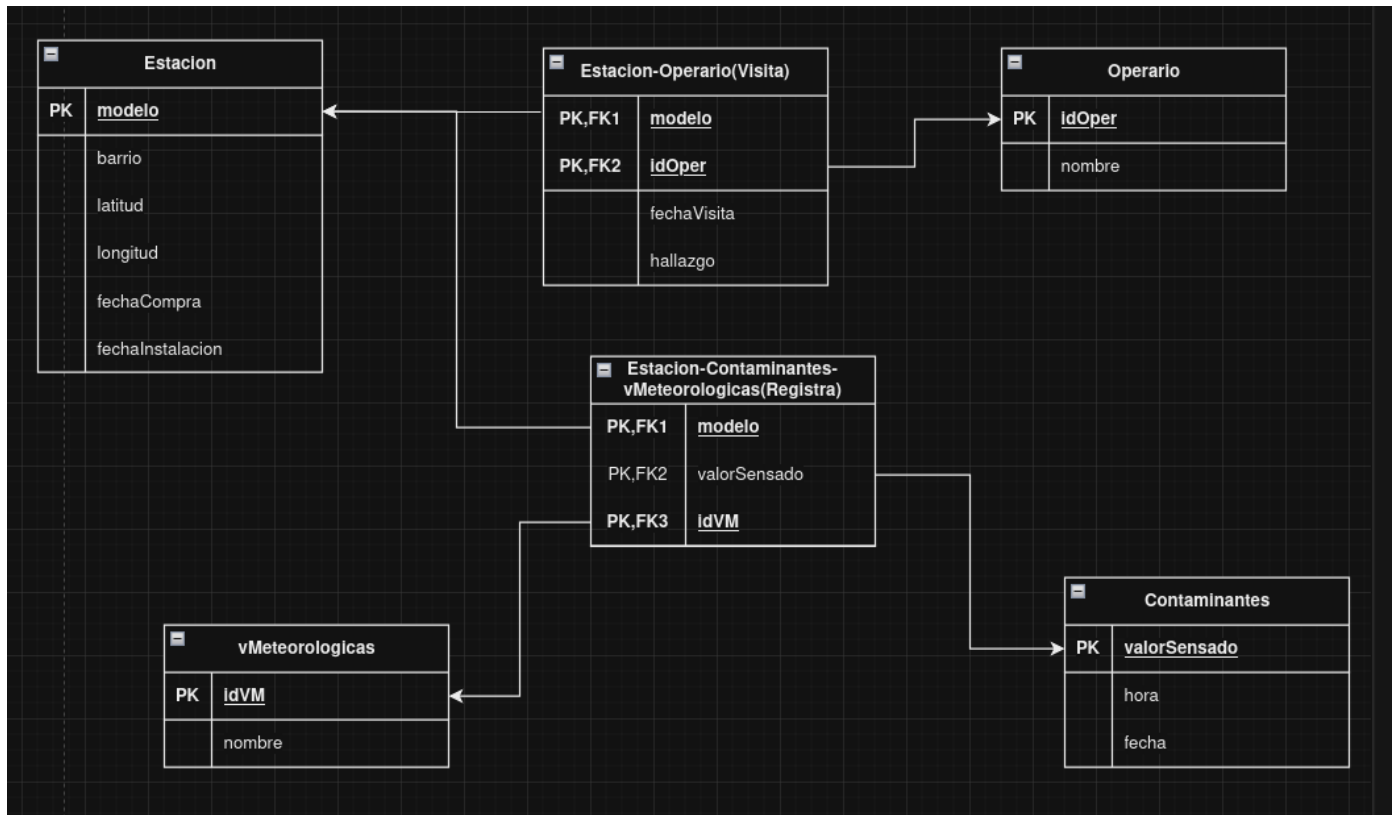
Una estación de monitoreo puede registrar el comportamiento de uno varios contaminantes entre los que se encuentran el PM2.5, PM10, Ozono (O3), Ácido sulfúrico (H2s) y una o varias variables meteorológicas como la temperatura, velocidad y dirección del viento, lluvia, radiación solar entre otras. Cada vez que una estación mide uno de los contaminantes registra el valor sensado con la fecha y la hora.

Práctica [50%]

(10 puntos) Cree y pegue a continuación un diagrama entidad-relación (ER) para los requerimientos de datos del SVCASC



(10 puntos) Realice la transformación al modelo relacional (REL) siguiendo las reglas vistas en clase y pegue a continuación el diagrama lógico que obtuvo



(5 puntos) Escriba y pegue a continuación las instrucciones del DDL que permitan crear las tablas/relaciones según su diseño lógico

-- Crear tabla Estacion

```
CREATE TABLE Estacion (  
    modelo VARCHAR(50) PRIMARY KEY,  
    barrio VARCHAR(255),  
    latitud DECIMAL(10,6),  
    longitud DECIMAL(10,6),  
    fechaCompra DATE,  
    fechaInstalacion DATE  
);
```

-- Crear tabla Operario

```
CREATE TABLE Operario(  
    IdOper SERIAL PRIMARY KEY,  
    nombre VARCHAR(255)  
);
```

-- Crear tabla Estacion-Operario(Visita)

```
CREATE TABLE Estacion_Operario_Visita(  
    modelo VARCHAR(50),  
    IdOper SERIAL,  
    fechaVisita DATE,  
    hallazgo VARCHAR(255),  
    FOREIGN KEY(modelo) REFERENCES Estacion(modelo),  
    FOREIGN KEY(idOper) REFERENCES Operario(idOper),  
    PRIMARY KEY(modelo, idOper)  
);
```

-- Crear tabla vMeteorologicas

```
CREATE TABLE vMeteorologicas(  
    IdVM SERIAL PRIMARY KEY,  
    nombre VARCHAR(255)  
);
```

-- Crear tabla Contaminantes

```
CREATE TABLE Contaminantes(  
    valorSensado VARCHAR(200) PRIMARY KEY,  
    hora TIME,  
    fecha DATE  
);
```

-- Crear tabla Estacion-Contaminantes-vMeteorologicas(Registra)

```
CREATE TABLE Estacion_Contaminantes_vMeteorologicas_Registra(  
    modelo VARCHAR(50),  
    valorSensado VARCHAR(200),  
    IdVM SERIAL,  
    FOREIGN KEY(modelo) REFERENCES Estacion(modelo),  
    FOREIGN KEY(valorSensado) REFERENCES Contaminantes(valorSensado),  
    FOREIGN KEY(idVM) REFERENCES vMeteorologicas(idVM),  
    PRIMARY KEY(modelo, valorSensado, idVM)  
);
```

(5 puntos) Escriba y pegue las instrucciones para insertar los operarios (inventa los datos)

-- Insertar operarios I

```
INSERT INTO Operario (pk, nombre) VALUES
```

```
(1, 'Lucía Sánchez'),  
(2, 'Miguel Suárez'),  
(3, 'Sofía Guzman'),  
(4, 'David Ramírez'),  
(5, 'Natalia Jiménez');
```

(2 puntos) Escriba y pegue las instrucciones para insertar las estaciones ([estaciones](#))

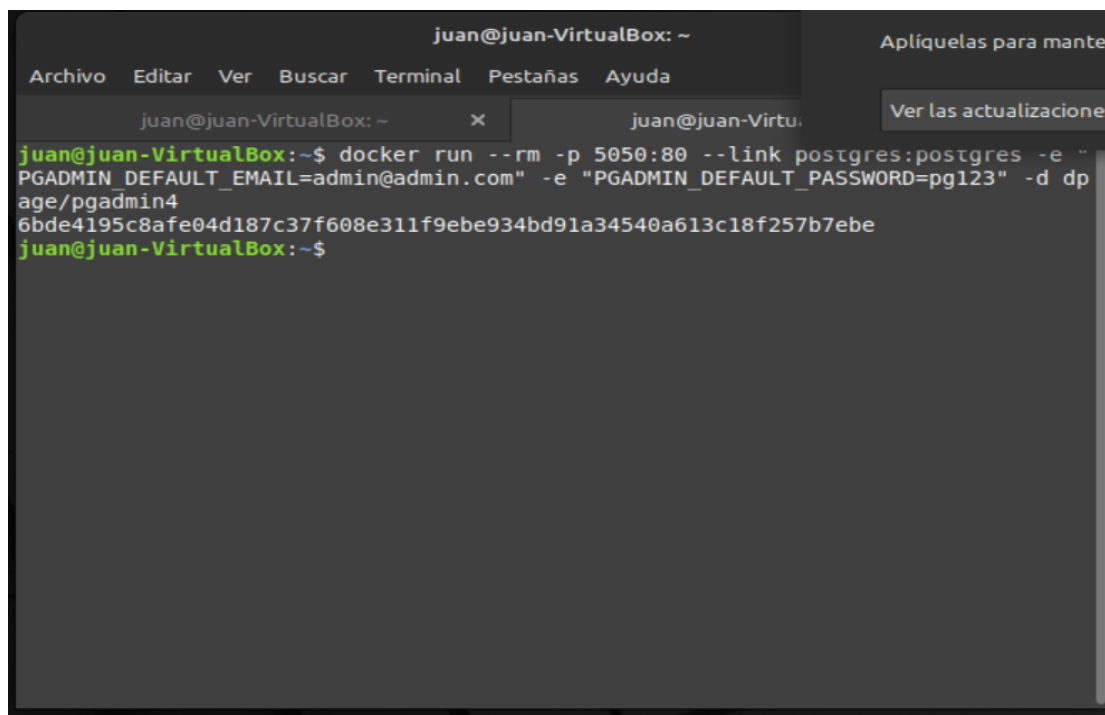
- Insertar datos de estaciones

```
INSERT INTO Estacion (pk, barrio, latitud, longitud, fechaCompra, fechaInstalacion) VALUES
('EST006', 'Chamberí', 40.433186, -3.706643, '2022-06-30', '2022-08-15'),
('EST007', 'Arganzuela', 40.396472, -3.703718, '2021-11-20', '2021-12-01'),
('EST008', 'Moncloa-Aravaca', 40.446941, -3.728365, '2023-02-28', '2023-04-01'),
('EST009', 'Latina', 40.391778, -3.751803, '2022-09-05', '2022-10-15'),
('EST010', 'Usera', 40.389205, -3.710621, '2020-04-10', '2020-06-01'),
('EST011', 'Carabanchel', 40.385235, -3.731715, '2021-05-01', '2021-07-01'),
('EST012', 'Fuencarral-El Pardo', 40.489175, -3.693619, '2022-12-15', '2023-01-15'),
('EST013', 'Tetuán', 40.454315, -3.703416, '2023-05-01', '2023-06-15'),
('EST014', 'Centro', 40.415728, -3.703286, '2021-08-20', '2021-10-01'),
('EST015', 'Hortaleza', 40.470417, -3.648903, '2022-03-01', '2022-05-01');
```

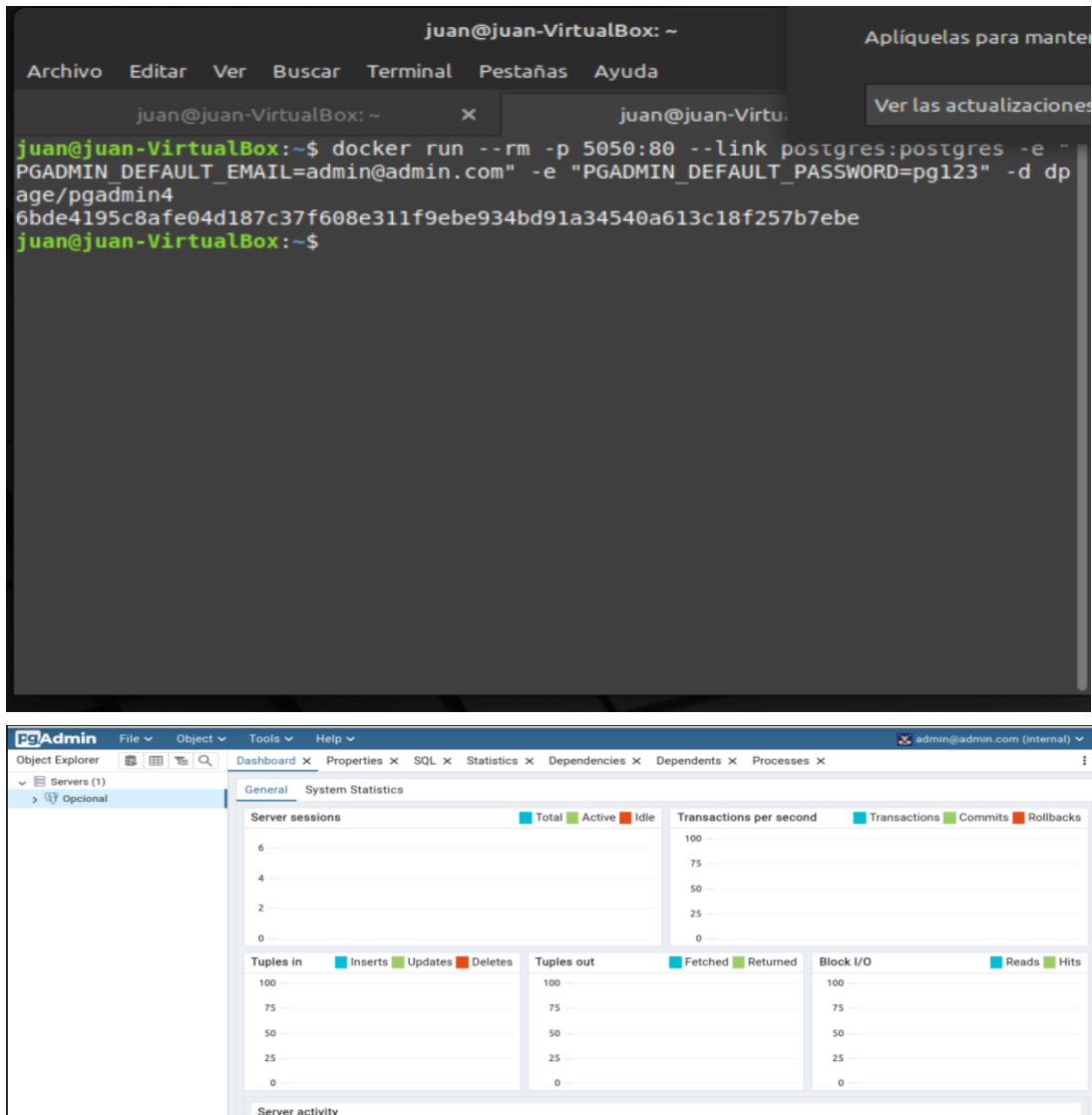
(3 puntos) Escriba y pegue las instrucciones para insertar las [variables y/o contaminantes](#)

```
INSERT INTO CONTAMINANTES (valorSensado, hora, fecha)
VALUES ('Ozono','2:00:00','2024-04-06')
```

(2 puntos) Escriba y pegue las instrucciones para levantar un contenedor con postgres y un contenedor con pgadmin

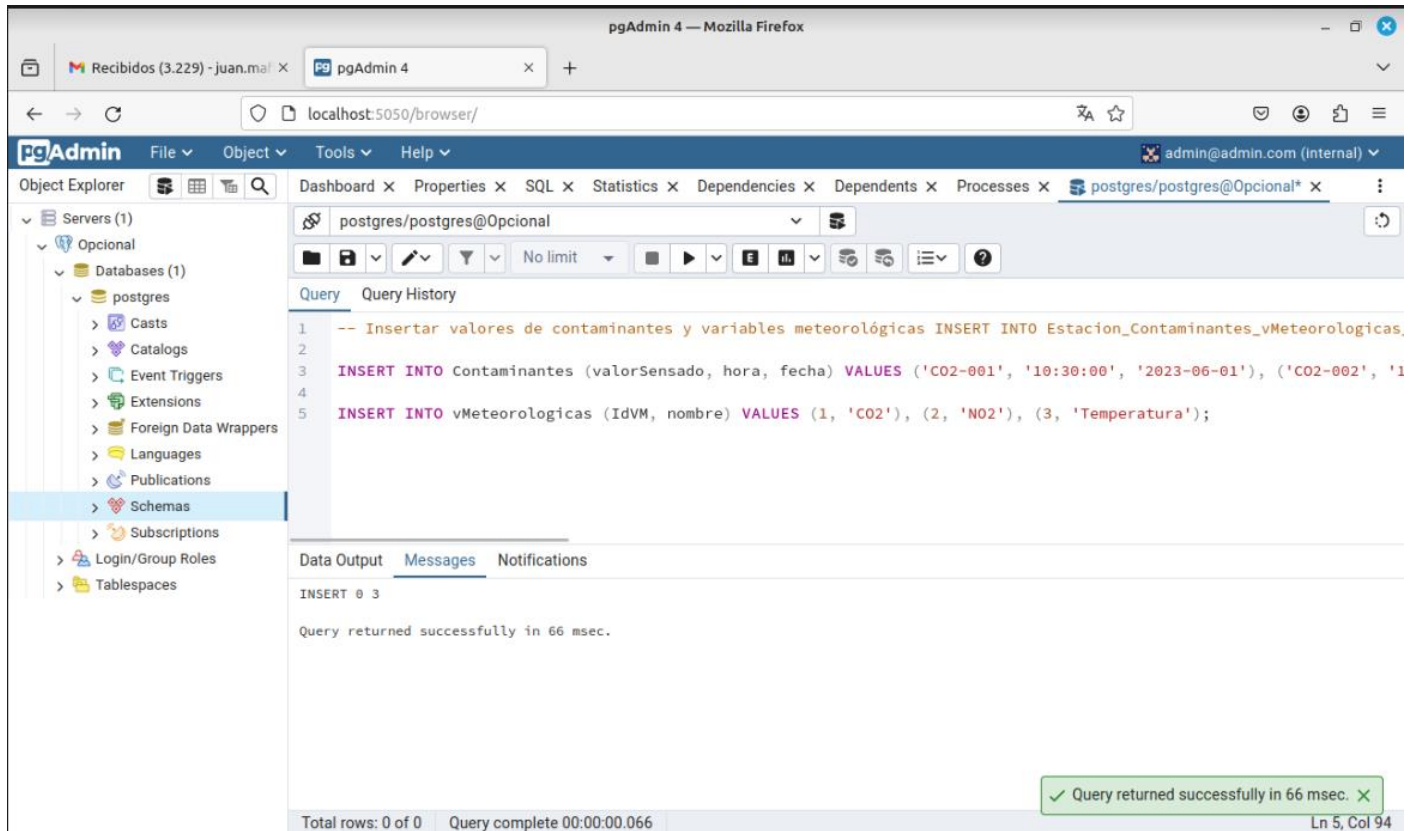


```
juan@juan-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juan@juan-VirtualBox: ~ x juan@juan-Virtu: Ver las actualizaciones
juan@juan-VirtualBox:~$ docker run --rm -p 5050:80 --link postgres:postgres -e "
PGADMIN_DEFAULT_EMAIL=admin@admin.com" -e "PGADMIN_DEFAULT_PASSWORD=pg123" -d dp
age/pgadmin4
6bde4195c8afe04d187c37f608e311f9ebe934bd91a34540a613c18f257b7ebe
juan@juan-VirtualBox:~$
```

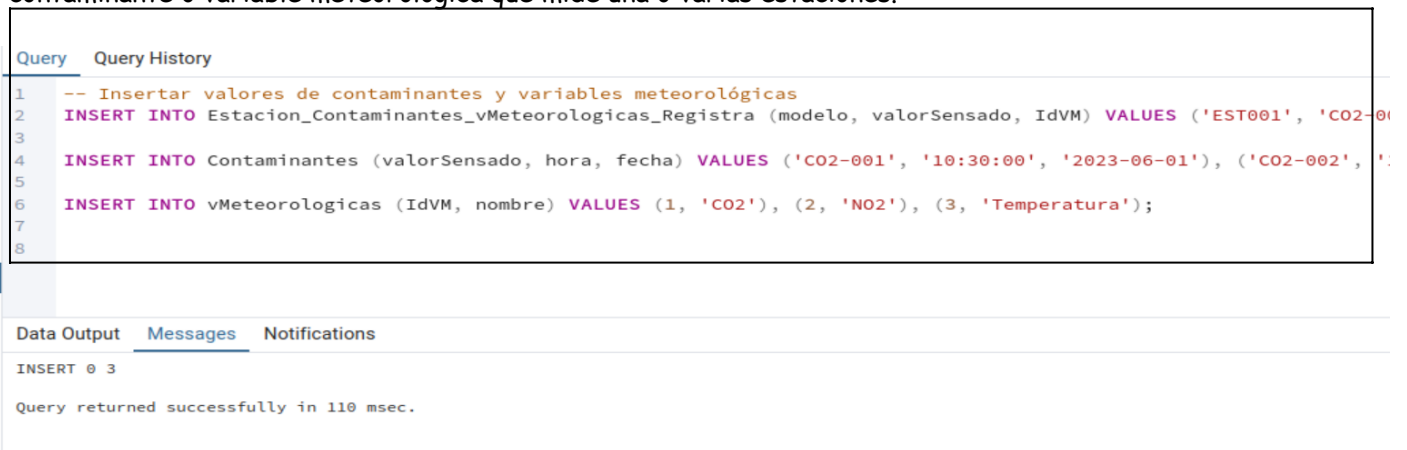


The image shows a terminal window and the PgAdmin interface. The terminal window, titled "juan@juan-VirtualBox: ~", displays the command to run a Docker container for PostgreSQL with pgAdmin4. The command is: `docker run --rm -p 5050:80 --link postgres:postgres -e "PGADMIN_DEFAULT_EMAIL=admin@admin.com" -e "PGADMIN_DEFAULT_PASSWORD=pg123" -d postgres/pgadmin4`. The output shows the container ID: `6bde4195c8afe04d187c37f608e311f9ebe934bd91a34540a613c18f257b7ebe`. The PgAdmin interface, titled "PgAdmin", shows the "System Statistics" tab. It displays various metrics including Server sessions, Transactions per second, Tuples in, Tuples out, and Block I/O. The "Server sessions" chart shows a total of 6 sessions, with 4 active and 2 idle. The "Transactions per second" chart shows a peak of 100 transactions. The "Tuples in" chart shows a peak of 100 tuples, with 100 inserts, 0 updates, and 0 deletes. The "Tuples out" chart shows a peak of 100 tuples, with 100 fetched and 0 returned. The "Block I/O" chart shows a peak of 100 blocks, with 100 reads and 0 hits.

3 puntos) Copie y pegue una captura de pantalla del momento en que se insertan (5) visitas de dos de los operarios a una de las estaciones



(3 puntos) Copie y pegue una varias capturas de pantalla donde se inserten (3 o más) [valores asociados](#) a cada contaminante o variable meteorológica que mide una o varias estaciones.



(2 puntos) Cree un repositorio en github de acceso público llamado database-course-midterm donde almacene capturas, archivos sql y todo lo que se ha creado para este examen.

(2 puntos) Copie y pegue a continuación el enlace a su repositorio.

(3 puntos) Realice una copia de este [documento](#), genere una versión en pdf y envíela a través del campus virtual.

Nota: Siga las instrucciones de este documento y cumpla con todos los requisitos. No se tendrán en cuenta las entregas realizadas en otro formato.

iMuchos éxitos!