

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

Enlace del diagrama: <https://drive.google.com/file/d/1lpmPoitU7QDQNFY0Oevqf8vHZGOI3BKZ/view?usp=sharing>

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

Enlace del diagrama: <https://drive.google.com/file/d/1lpmPoitU7QDQNFY0Oevqf8vHZGOI3BKZ/view?usp=sharing>

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

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

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

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

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

(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.

¡Muchos éxitos!

--CREAR OPERARIO:

```
CREATE TABLE operario ( id_operario SERIAL PRIMARY KEY, fecha_visita  
VARCHAR(100) NOT NULL, novedad VARCHAR(100) NOT NULL, );
```

--CREAR ESTACION:

```
CREATE TABLE estacion ( id_estacion SERIAL PRIMARY KEY, modelo VARCHAR(100)  
NOT NULL, fecha_instalacion VARCHAR(100) NOT NULL, fecha_compra VARCHAR(100)  
NOT NULL );
```

--CREAR UBICACION DE ESTACION:

```
CREATE TABLE ubicacion_estacion ( BARRIO_UBICACION VARCHAR(50), latitud  
VARCHAR(50), longitud VARCHAR(50), id_estacion INT NOT NULL, CONSTRAINT  
fk_direccion_estacion FOREIGN KEY(id_estacion) REFERENCES estacion (id_estacion) );
```

--CREAR MEDICION:

```
CREATE TABLE medicion ( id_medicion SERIAL PRIMARY KEY, contaminante  
VARCHAR(100) NOT NULL, variable_meteorologica VARCHAR(100) NOT NULL, );
```

-CREAR TABLA REGISTRA:

```
CREATE TABLE registra ( id SERIAL PRIMARY KEY, codigo_modelo INT NOT NULL,  
codigo_medicion INT NOT NULL, hora TIMESTAMP NOT NULL DEFAULT  
CURRENT_TIMESTAMP, fecha DATE NOT NULL, CONSTRAINT fk_modelo FOREIGN  
KEY (codigo_modelo) REFERENCES estacion (codigo_modelo), CONSTRAINT  
fk_medicion FOREIGN KEY (codigo_medicion ) REFERENCES medicion (codigo_medicion  
));
```

-- Insertar 5 registros para la entidad operarios:

```
INSERT INTO operario (fecha_visita, novedad)
VALUES ('2024-05-01', 'Revisión completa, sin novedades'), ('2024-05-05', 'Cambio de filtro
de aire'), ('2024-05-10', 'Reparación de tubería dañada'), ('2024-05-15', 'Inspección de
seguridad, todo en orden'), ('2024-05-20', 'Sustitución de piezas desgastadas');
```

-- Insertar 5 registros para la entidad estacion

```
INSERT INTO estacion (modelo, fecha_instalacion, fecha_compra)
VALUES ('Modelo A', '2023-01-10', '2022-12-15'), ('Modelo B', '2023-03-20', '2023-02-25'),
('Modelo C', '2023-05-05', '2023-04-10'), ('Modelo D', '2023-07-15', '2023-06-20'), ('Modelo
E', '2023-09-01', '2023-08-05');
```

PARA CORRER LOS CONTENEDORES DE PG ADMIN Y POSTGRES:

CORRER POSTGRES:

```
docker run -e POSTGRES_USER=EDINSON -e POSTGRES_PASSWORD=ED -e POSTGRES_DB=text -p
5432:5432 -d postgres → Sirve para ejecutar la imagen de docker, inicializar la contraseña, el
usuario y asignarle nombre a la base de datos
```

```
> docker run -e POSTGRES_USER=EDINSON -e POSTGRES_PASSWORD=ED -e POSTGRES_DB=test -p 5432:5432 -d postgres
e088638c4d045c6c7b5a8ff277fb8228540e6163aa4c0160f03d8fdfecf72709
```

CORRER PGADMIN:

Nos permite usar el pgadmin, asignarle un correo y una contraseña y mostrar el puerto que dejaremos abe

```
docker run -p 80:80 -e PGADMIN_DEFAULT_EMAIL=user@domain.com -e PGADMIN_DEFAULT_PASSWORD=Super -d dpage/pgadmin4
```