**Segundo trabajo de bases de datos (25%)**

**Fecha de entrega**: Jueves 16 de diciembre de 2021 hasta las **6pm**. Trabajos recibidos después de esa hora no se califican. Los trabajos deben ser enviados a este correo:

*[fjmoreno@unal.edu.co](mailto:fjmoreno@unal.edu.co)*

- Trabajos enviados “por accidente” a otros correos no se calificarán.

- Envíe **un solo** archivo comprimido con todo el trabajo (informe y programas). Si envía varios archivos, el trabajo se le calificará como máximo con una nota de 3.0.

- Para facilitar la identificación, el nombre del archivo comprimido debe tener el primer apellido de cada integrante del grupo. Ejemplo: **Lipa\_Electra\_Jolie.zip**

- No se reciben ni se califican versiones “mejoradas”. **No envíe varias versiones**. No envíe varios programas (soluciones). **Solo un programa por cada punto**.

- Grupos máximo de **tres** estudiantes.

**Nota**: Los progamas deben estar comentados, explicando cada línea de código o cada grupo de líneas (cada 2, 3, 4, o 5 líneas de código) que se está haciendo. **Trabajos copiados parcial o totalmente**, se calificarán con cero para todos los integrantes involucrados sin excepción. **Programas no comentados (explicados) se calificarán con cero, así hagan lo solicitado**.

Se debe entregar un informe de **máximo 3 páginas** donde se explique brevemente cómo se solucionó cada punto. No incluya en dicho informe el código de los programas, el código de los programas se presenta en archivos por aparte.

Cualquier duda consultarla con el profesor. El monitor les puede ayudar con recomendaciones pero su labor no es hacerles el trabajo **ni está autorizado para cambiar las condiciones del trabajo**.

Realice los programas en el lenguaje que desee, puede ser Java, Python u otro, pero todos los programas deben estar hechos en el mismo lenguaje (es decir, o todos en Java, o todos en Python o todos en C++, etc.)

**Punto 1 (40%)**

Realice un programa que haga lo siguiente. El programa recibe como parámetros dos conjuntos de dependencias funcionales **S1** y **S2**. El programa debe indicar si los conjuntos **S1** y **S2** son equivalentes.

**Nota**: **S1** y **S2** son equivalentes si **todas** las dependencias funcionales de **S1** son implicadas por **S2** y si **todas** las dependencias funcionales de **S2** son implicadas por **S1**.

Usted define el formato de ingreso de **S1** y de **S2**.

Por simplicidad cada dependencia funcional tendrá como máximo 4 atributos a la izquierda y como máximo 4 atributos a la derecha.

Cada atributo se simboliza con una letra mayúscula.

Tanto **S1** como **S2** pueden tener cualquier número de dependencias funcionales.

**Ejemplo**

Sea **S1** = {AP 🡪 BHW, BHW 🡪 C}

Sea **S2** = {AP 🡪 BHW, BHW 🡪 C, AP 🡪 C}

En este ejemplo **S1** y **S2** son equivalentes.

**Nota**: en el anterior ejemplo, AP representa **dos** atributos, es decir, los atributos A y P. BHW representa tres atributos: B, H y W.

**Punto 2**

Este punto tiene parte **a)** y parte **b)**.

El programa recibe como entrada una muestra de datos (una relación) que tendrá **como máximo** **4** atributos. Se **garantiza** que la muestra de datos es una relación (es decir, no tiene, por ejemplo, tuplas repetidas). La muestra de datos puede tener **cualquier** número de tuplas. Usted decide como se debe ingresar la muestra de datos en su programa: debe recibir los nombres de los atributos y las tuplas.

**Parte a) (35%)**

El programa debe analizar la relación ingresada e indicar si se encuentra en BCNF. El programa debe imprimir además todos los determinantes y todas las claves candidatas (Nota: puede haber determinantes y claves candidatas compuestas).

Nota: Su programa debe funcionar para **cualquier** muestra de datos que tenga hasta 4 atributos (inclusive), es decir, no solamente para los siguientes ejemplos.

**Ejemplo.**

Supongamos que se ingresa la siguiente muestra de datos:

**Cédula Nombre**

12 Juan

23 Juan

76 Lisa

El programa debe imprimir lo siguiente:

La relación se encuentra en BCNF.

Claves candidatas: {Cédula}

Determinantes: {Cédula}

**Otro ejemplo:**

Supongamos que se ingresa la siguiente muestra de datos:

**Código Usuario Tipodesangre**

12 5 A

35 6 B

76 5 A

34 5 A

78 6 B

99 4 A

El programa debe imprimir lo siguiente:

La relación NO se encuentra en BCNF.

Claves candidatas: {Código}

Determinantes: {Código} y {Usuario}

**Parte b) (25%)**

El usuario ingresa una dependencia funcional y el programa le debe indicar si esa dependencia funcional se cumple en la muestra de datos ingresada. El programa debe indicar si la dependencia funcional ingresada es completa o incompleta.

Por simplicidad cada dependencia funcional ingresada tendrá como máximo 4 atributos a la izquierda y como máximo 4 atributos a la derecha.

**Ejemplo.**

En la muestra de datos:

**Código Usuario Tipodesangre**

12 5 A

35 6 B

76 5 A

34 5 A

78 6 B

99 4 A

Si el usuario ingresa la dependencia:

{Código 🡪 Tipodesangre}

El programa debe imprimir:

La dependencia que acaba de ingresar se cumple en la muestra y es completa.

Si ahora el usuario ingresa la dependencia:

{Código, Usuario 🡪 Tipodesangre}

El programa debe imprimir:

La dependencia que acaba de ingresar se cumple en la muestra y es incompleta.

Si ahora el usuario ingresa la dependencia:

{Tipodesangre 🡪 código}

El programa debe imprimir:

La dependencia que acaba de ingresar no se cumple en la muestra.