

## **FUNDAMENTOS DE BASES DE DATOS**

### **Examen Diciembre 2024**

**La duración del examen es de 3 horas.**

**Presentar la resolución del examen:**

- Con las hojas numeradas y escritas de un solo lado.
- Con la cantidad de hojas entregadas en la primer hoja.
- Con cédula de identidad y nombre en cada hoja.
- Escrita a lápiz y en forma prolija.
- Comenzando cada ejercicio en una nueva hoja

### **Ejercicio 1 (25 puntos)**

Se desea modelar la realidad correspondiente a las obras que se presentan en un teatro. De las obras se registra la fecha de estreno junto a la fecha de creación, ambas son necesarias para identificar a cada obra. Además, se registra el tipo de obra, el cual puede ser uno o varios (por ejemplo infantil, musical y/o drama, etc.) y el nombre.

Por otro lado, se lleva registro de los objetos que se usan en cada obra (también llamados utilería, como por ejemplo vestuario, instrumentos, muebles, etc.). Cada objeto tiene un identificador, una descripción y una fecha de compra. A su vez, interesa saber en qué obra se usa cada objeto, así como también todos los períodos (cada período implica una fecha de inicio y una de fin), en los que se usa cada objeto en cada obra. Tener en cuenta que hay obras que no usan objetos, pero todos los objetos han sido usados al menos una vez.

También interesa información sobre todas las personas vinculadas al teatro: integrantes y espectadores. Los integrantes del teatro pueden ser espectadores. Para todos ellos se registra e-mail, que los identifica, nombre y todos sus teléfonos (al menos uno). En particular, para los integrantes del teatro también se registra cédula de identidad y fecha de ingreso. Entre los integrantes del teatro hay artistas, utileros y otros. Los artistas no son utileros.

Interesa registrar en qué obras actúa cada artista y se sabe que en toda obra actúan artistas. Los artistas pueden ser directores de obras. En todas las obras solo hay un director y un artista puede ser director de varias obras.

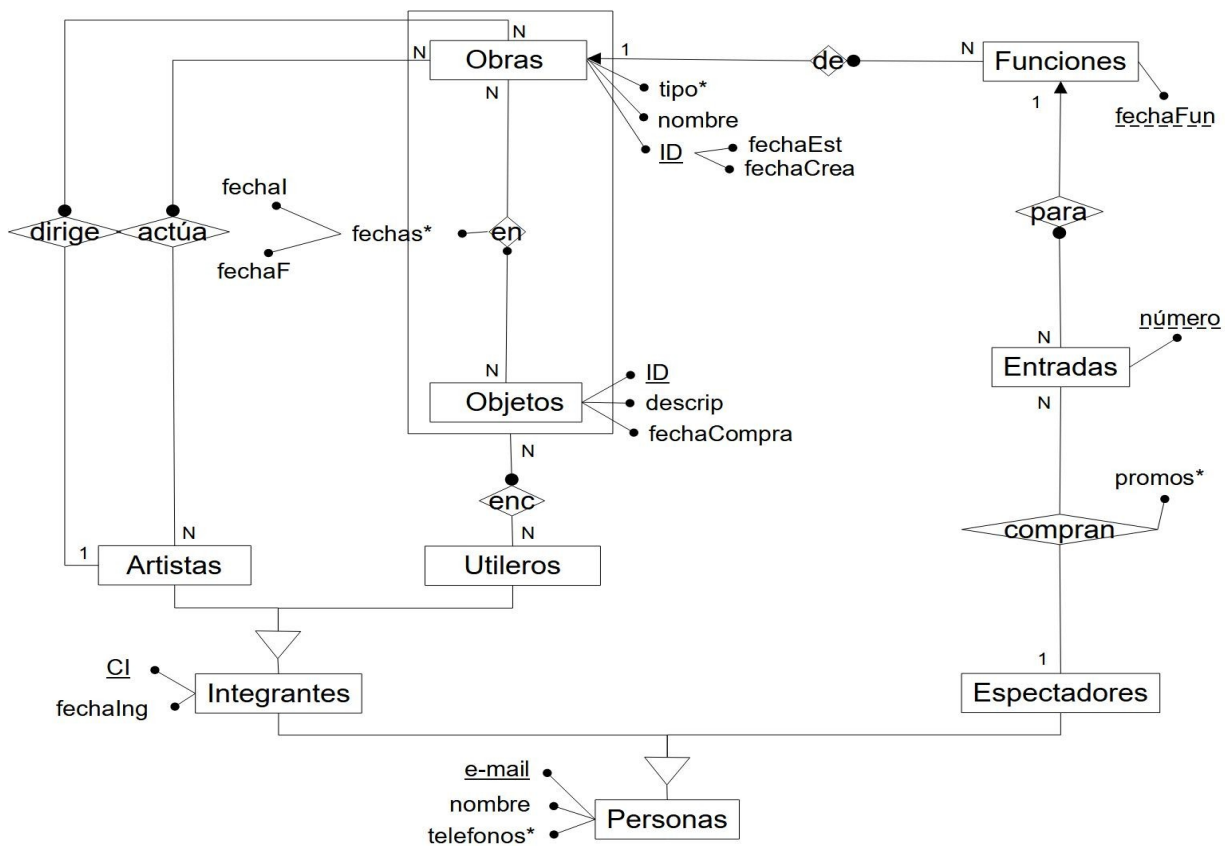
Por otro lado, los utileros se encargan de los objetos usados en cada obra, interesa saber quiénes son los utileros que se encargan de cada objeto en cada obra. Es posible que cada utilero se encargue de varios objetos en varias obras y cuando el objeto es muy importante puede haber más de un utilero encargado. Cada objeto en una obra tiene al menos un utilero como encargado.

Las obras tienen funciones y cada función tiene una fecha que la identifica. Tener en cuenta que en una misma fecha puede haber varias funciones de diferentes obras. Además, para cada función se habilita un conjunto de entradas, y cada una de ellas se identifica con un número que es único en cada función. Es decir, puede haber dos entradas con igual número en distintas funciones.

Finalmente, de los espectadores interesa registrar todas las entradas que compra para cada obra. Interesa saber si cada entrada comprada por un espectador es realizada usando una o varias promociones. Ejemplos de promociones pueden ser “mayores de 60 con 50% menos”, “menores de 12 con 80% menos”, “oca 20% menos”, etc.

**Se pide Modelo Entidad Relación completo.**

**Solución**



RNE :

$Integrantes \cup Espectadores = Personas$

$Artistas \cap Utileros = \emptyset$

$tipo \neq \emptyset$

$telefonos \neq \emptyset$

$fechas \neq \emptyset$

$(\forall (o, f) \in de)(o.ID . fechaEst \leq f . fechaFun)$

$(\forall (o, f) \in de)(o.ID . fechacrea < f . fechaFun)$

## Ejercicio 2 (25 puntos)

Una librería almacena en una base de datos información sobre los libros que vende y las ventas realizadas. Se cuenta con las siguientes tablas:

**Libros**(ISBN, título, autorId, generoId, añoPub, precio, stock, provId)

Contiene información de los libros que vende la librería, su precio, la cantidad de ejemplares en stock e información sobre el proveedor.

**Autores**(autorId, nombre, bio, fechaNac)

Contiene información sobre los autores de los libros disponibles en la librería.

**Generos**(generoId, nombre)

Contiene información sobre los géneros de los libros disponibles en la librería.

**Cientes**(clienteId, nombre, email, teléfono, dirección)

Almacena datos de los clientes que hicieron alguna compra.

**Compras**(compraId, clienteId, isbn, fecha)

Almacena información sobre las compras, el cliente que la realizó, qué libro compró y la fecha en la que se realizó la compra.

**Proveedores**(provId, nombre, email, teléfono, dirección)

Contiene los datos de los proveedores de libros.

En la base de datos no hay tablas vacías y se cumplen las siguientes dependencias de inclusión:

$$\Pi_{\text{provId}} \text{Libros} \subseteq \Pi_{\text{provId}} \text{Proveedores}$$

$$\Pi_{\text{generoId}} \text{Libros} \subseteq \Pi_{\text{generoId}} \text{Generos}$$

$$\Pi_{\text{autorId}} \text{Libros} \subseteq \Pi_{\text{autorId}} \text{Autores}$$

$$\Pi_{\text{clienteId}} \text{Compras} \subseteq \Pi_{\text{clienteId}} \text{Cientes}$$

$$\Pi_{\text{isbn}} \text{Compras} \subseteq \Pi_{\text{isbn}} \text{Libros}$$

Se pide:

### 1. Resolver en Álgebra Relacional las siguientes consultas

- a) Devolver los nombres de los autores que publicaron al menos 3 libros distintos después del año 2020.

$$L1 = \rho_{\text{isbn, autorId, añoPub} \rightarrow \text{isbn1, autorId1, añoPub1}}(\text{Libros})$$

$$L2 = \rho_{\text{isbn, autorId, añoPub} \rightarrow \text{isbn2, autorId2, añoPub2}}(\text{Libros})$$

$$A = \Pi_{\text{autorId}} \left( \sigma_{\text{añoPub} > 2020}(\text{Libros}) \bowtie_{\text{isbn} \neq \text{isbn1} \wedge \text{autorId} = \text{autorId1} \wedge \text{añoPub1} > 2020} L1 \bowtie_{\text{isbn} \neq \text{isbn2} \wedge \text{isbn1} \neq \text{isbn2} \wedge \text{autorId} = \text{autorId2} \wedge \text{añoPub2} > 2020} L2 \right)$$

$$RES = \Pi_{\text{nombre}} \text{Autores} * A$$

- b) Devolver los nombres de los clientes que compraron solo libros del género "Terror".

$$CLIENTES\_NO\_TERROR = \prod_{clienteId} (\sigma_{nombre \neq Terror}(Género) * Libros * Compras)$$

$$CLIENTES\_TERROR = \prod_{clienteId} (\sigma_{nombre = Terror}(Género) * Libros * Compras)$$

$$RES = \prod_{nombre} (\prod_{clienteId} (CLIENTES\_TERROR) - CLIENTES\_NO\_TERROR) * Clientes$$

## 2. Resolver en Cálculo Relacional la siguiente consulta

- a) Devolver el nombre de los autores tales que todos sus libros fueron vendidos en el 2024.

$$\{ \langle a.nombre \rangle / AUTORES(a) \wedge (\forall l)(LIBROS(l) \wedge l. autorId = a. autorId \rightarrow (\exists c)(COMPRAS(c) \wedge c.isbn = l.isbn \wedge c.fecha \leq '01-01-2024' \wedge c.fecha \geq '31-12-2024')) \}$$

## 3. Resolver en SQL las siguientes consultas sin usar vistas ni subconsultas en el FROM

- a) Devolver el identificador de los proveedores que proveen libros de al menos 2 géneros distintos.

```
SELECT provId
FROM Proveedores prov, LIBROS l1
WHERE prov.provId = l1.provId and EXISTS (
    SELECT 1
    FROM LIBROS l2
    WHERE l2.provId = prov.provId and l2.genreId != l1.genreId
)
```

- b) Devolver el isbn, el título y la cantidad de veces que se vendió un libro, para aquellos libros que se hayan vendido por lo menos 3 veces en el 2024. El resultado debe estar ordenado de forma decreciente según la cantidad de ventas.

```
SELECT l.isbn, l.titulo, COUNT(*)
FROM Libros l NATURAL JOIN Compras c
WHERE c.fecha >= '1/1/2024' AND c.fecha <= '31/12/2024'
GROUP BY l.isbn
HAVING COUNT(*) >= 3

ORDER BY COUNT(*) DESC
```

### Ejercicio 3 (25 puntos)

Dado el esquema relación **R (A, B, C, D, E, G, H)** y los siguientes conjuntos de dependencias funcionales:

$$\mathbf{F} = \{DG \rightarrow B, DG \rightarrow E, C \rightarrow B, AB \rightarrow H, G \rightarrow AH, DH \rightarrow C, DH \rightarrow G\}$$

$$\mathbf{J} = \{DG \rightarrow B, C \rightarrow DH, G \rightarrow AE, DH \rightarrow BG, BAG \rightarrow H\}$$

1. Determinar si **F** y **J** son equivalentes.

Para que sean equivalentes se debe cumplir que  $F^+ \subseteq J^+$  (J cubre a F) y  $J^+ \subseteq F^+$  (F cubre a J). Esto se cumple si todas las dfs de F se pueden deducir de J y todas las dfs de J se pueden deducir de F.

Comenzamos verificando si J cubre a F:

$$(DG)^+_J = \{D, G, B, A, E, H\}, \text{ se cumple } DG \rightarrow B \text{ y } DG \rightarrow E \text{ en } J^+$$

$$(C)^+_J = \{C, D, H, B, G, A, E\}, \text{ se cumple } C \rightarrow B \text{ en } J^+$$

$$(AB)^+_J = \{A, B\}, \text{ NO se cumple } AB \rightarrow H \text{ en } J^+$$

entonces J no cubre a F, por lo tanto **F y J no son equivalentes**.

Otro camino de solución:

Si se hubiera comenzado verificando si F cubre a J se hubiera encontrado que  $C \rightarrow DH$  no se deduce de F, por lo tanto F no cubre a J.

2. Hallar todas las claves de R **según J**.

$$\mathbf{J} = \{DG \rightarrow B, C \rightarrow DH, G \rightarrow AE, DH \rightarrow BG, BAG \rightarrow H\}$$

C nunca se encuentra a la derecha de una df, por lo tanto C pertenece a todas las claves.

E nunca se encuentra a la izquierda de una df, por lo tanto E no pertenece a ninguna clave.

Comenzamos probando con la clausura de C:

$$C_{J^+} = \{C, D, H, B, G, A, E\} = R$$

C es clave y es la única clave.

Es la única porque debe estar en todas las claves y si agregamos algún atributo, sería una superclave pero no una clave.

3. Hallar una descomposición  $\rho$  de R en 3NF con JSP y preservación de dependencias **según F**.

Comenzamos hallando el cubrimiento minimal de F:

$$F = \{DG \rightarrow B, DG \rightarrow E, C \rightarrow B, AB \rightarrow H, G \rightarrow AH, DH \rightarrow C, DH \rightarrow G\}$$

**Paso 1: Separar lados derechos**

$$F_{\min} = \{DG \rightarrow B, DG \rightarrow E, C \rightarrow B, AB \rightarrow H, G \rightarrow A, G \rightarrow H, DH \rightarrow C, DH \rightarrow G\}$$

**Paso 2: Eliminar Atrib. redundantes**

Verifico  $DG \rightarrow B$ .  $(D)^+ = \{D\}$  y  $(G)^+ = \{G, A, H\}$ , ninguno es redundante

Verifico  $DG \rightarrow E$ . Por las clausuras anteriores vemos que tampoco es redundante ninguno.

Verifico  $AB \rightarrow H$ . Vemos que en las dfs A solo nunca está a la izquierda y B solo tampoco. Entonces ni A ni B son redundantes.

Verifico  $DH \rightarrow C$  y  $DH \rightarrow G$ . Sabemos que H no es redundante porque hicimos antes  $(D)^+$ . Vemos que H nunca está solo a la izquierda, por lo tanto D tampoco es redundante.

No hay atributos redundantes.

**Paso 3: Eliminar Dependencias redundantes**

Verifico si  $DG \rightarrow B$  es redundante.

$(DG)^+_{F_{\min} - \{DG \rightarrow B\}} = \{D, G, E, A, H, C, B\}$ . Llegamos a B, por lo tanto  **$DG \rightarrow B$  es redundante.**

$$F_{\min} = \{DG \rightarrow E, C \rightarrow B, AB \rightarrow H, G \rightarrow A, G \rightarrow H, DH \rightarrow C, DH \rightarrow G\}$$

Sólo resta verificar  $AB \rightarrow H$  y  $G \rightarrow H$ , porque en el resto de las dfs el lado izquierdo está solo en ellas por lo tanto no se podrían deducir sin ellas.

Verifico  $AB \rightarrow H$ :

$(DG)^+_{F_{\min} - \{AB \rightarrow H\}} = \{A, B\}$ , no es redundante.

Verifico  $G \rightarrow H$ :

$(G)^+_{F_{\min} - \{G \rightarrow H\}} = \{G, A\}$ , no es redundante.

Entonces el cubrimiento minimal es:

$$F_{\min} = \{DG \rightarrow E, C \rightarrow B, AB \rightarrow H, G \rightarrow A, G \rightarrow H, DH \rightarrow C, DH \rightarrow G\}$$

A continuación construimos los subesquemas siguiendo el algoritmo visto en el curso.

R1 (D, G, E)

R2 (C, B)

R3 (A, B, H)

R4 (G, A, H)

R5 (D, H, C, G)

Resta verificar si alguno de los subesquemas contiene a una clave de R según F.

Anteriormente vimos que  $(DG)^+ = \{ D, G, E, A, H, C, B \}$ , por lo tanto  $(DG)$  es superclave.

Además sabemos que  $(D)^+ = \{D\}$  y  $(G)^+ = \{G, A, H\}$ . Por lo tanto  $(DG)$  es una clave.

Como  $(DG)$  está incluido en  $R_5$ , no es necesario agregar ningún subesquema más.

También verificamos que ningún subesquema  $R_i$  está incluido en otro.

Por lo tanto la descomposición obtenida por el algoritmo de 3NF con JSP y preservación de dependencias es la siguiente:

$\rho = \{ R_1 (D, G, E); R_2 (C, B); R_3 (A, B, H); R_4 (G, A, H); R_5 (D, H, C, G) \}$

#### 4. Determinar si la descomposición anterior ( $\rho$ ) está en BCNF.

Para verificar la forma normal, debemos proyectar las dependencias en cada subesquema.

$R_1 (D, G, E)$

$F_{R_1} = \{ DG \rightarrow E \}$ , está en BCNF.

$R_2 (C, B)$

$F_{R_2} = \{ C \rightarrow B \}$ , está en BCNF

$R_3 (A, B, H)$

$F_{R_3} = \{ AB \rightarrow H \}$ , está en BCNF

$R_4 (G, A, H)$

$F_{R_4} = \{ G \rightarrow AH \}$ , está en BCNF

$R_5 (D, H, C, G)$

$F_{R_5} = \{ DH \rightarrow CG; G \rightarrow H \}$ , NO está en BCNF, porque  $G$  no es superclave en  $R_5$ .

La descomposición  $\rho$  no está en BCNF.

#### 5. Obtener una descomposición de $R$ en 4NF con JSP, sabiendo que además de las dependencias de $F$ se cumple la dependencia multivaluada embebida $D \twoheadrightarrow C \mid HG$ .

Primero llevamos a  $\rho$  a BCNF. Para eso debemos llevar a  $R_5$  a BCNF.

Aplicamos el algoritmo, y tomando la  $df$  que viola BCNF,  $G \rightarrow H$ , nos queda:

**$R_{51} (G, H)$** , donde  $F_{R_{51}} = \{ G \twoheadrightarrow H \}$ , que cumple BCNF

**$R_{52} (G, D, C)$** , donde  $F_{R_{52}} = \{ \}$ , que cumple BCNF

Luego verificamos si la  $dmv$   $D \twoheadrightarrow C \mid HG$  nos afecta en alguno de los subesquemas. Nos preguntamos si afecta a  $R_{52}$ .

Según la definición de  $mvds$ , para un mismo valor de  $D$ , todos los valores de  $C$  deben combinarse con todos los valores de  $HG$ . Además sabemos que  $G \twoheadrightarrow H$ . Por lo tanto, si no estuviera  $H$ , se sigue cumpliendo que todos los valores de  $C$  deben combinarse con todos los de  $G$ . Es decir que se cumple  $D \twoheadrightarrow C \mid G$ .

Por lo tanto, R52 está en BCNF, pero no en 4NF. Aplicamos el algoritmo de 4NF y generamos:

**R521 (D, C)**, cumple 4NF

**R522 (D, G)**, cumple 4NF

Finalmente, la descomposición final que cumple 4NF es la siguiente:

R1 (D, G, E)

R2 (C, B)

R3 (A, B, H)

R4 (G, A, H)

R51 (G, H)

R521 (D, C)

R522 (D, G)



#### Ejercicio 4 (25 puntos)

Dadas las siguientes transacciones:

**T1: w1(X), r1(Y), r1(Z), w1(Z), c1**

**T2: r2(X), r2(Y), w2(Y), c2**

a) Dar una historia **entrelazada** de T1 y T2 que evite abortos en cascada, justificando.

#### Solución:

Se recuerda que una historia Evita Abortos en Cascada (EAC) si ninguna transacción lee de transacciones no confirmadas.

Una historia entrelazada de T1 y T2 posible que cumpla EAC:

**H1a: r2(X), w1(X), r1(Y), r1(Z), w1(Z), c1, r2(Y), w2(Y), c2**

Ni T2 lee de T1 ni T1 lee de T2, por lo tanto H1a cumple EAC.

Otra posible donde una transacción lea de la otra y se cumpla EAC:

**H1b: r2(X), w1(X), r2(Y), w2(Y), c2, r1(Y), r1(Z), w1(Z), c1**

En este caso T1 lee Y de T2 pero T2 hace el commit antes del r1(Y), por lo tanto H1b cumple EAC.

b) Dar una historia entrelazada de T1 y T2 que no sea recuperable, justificando. Utilizando esta historia mostrar un caso en el que por culpa de no ser recuperable, la base de datos podría quedar inconsistente.

#### Solución:

Se recuerda que una historia es Recuperable (RC) si ninguna transacción confirma hasta que confirmaron todas las transacciones desde las cuales leyó ítems.

Una posible historia que NO cumple RC es la siguiente:

**H2a: w1(X), r1(Y), r1(Z), w1(Z), r2(X), r2(Y), w2(Y), c2, c1**

T2 lee de T1 (w1(X) está antes que r2(X) y no hay un a1 en el medio ni otro wj(X)) y T2 confirma antes que T1. No cumple con la definición de RC.

A continuación se presenta un ejemplo donde la base de datos podría quedar inconsistente debido a que T2 lee de T1 y T1 aborta luego que T2 hizo commit.

**H2b: w1(X), r1(Y), r1(Z), r2(X), w1(Z), r2(Y), w2(Y), c2, a1**

No cumple RC ya que T2 lee X de T1 y no hay commit de T1 antes del commit de T2. El a1 provoca que deshaga el w1(x). Sin embargo, T2 ya leyó el X escrito por ese write. Si el resto de las operaciones de T2 depende del valor de X puede generarse una inconsistencia en los datos.

c) Dada la siguiente historia de T1 y T2:

**H1: w1(X), r1(Y), r2(X), r2(Y), r1(Z), w1(Z), w2(Y), c2, c1**

Decir si es serializable y en caso de que lo sea, dar la historia serial equivalente.

**Solución:**

Se recuerda que una historia H es serializable si su grafo de seriabilidad es acíclico.

T1 → T2 por w1(X), r2(X)

T1 → T2 por r1(Y), w2(Y)

Es serializable porque su grafo de seriabilidad es acíclico (solo hay un arco de T1 a T2).

La historia serial equivalente es T1; T2:

**H3: w1(X), r1(Y), r1(Z), w1(Z), c1, r2(X), r2(Y), w2(Y), c2**

d) Escribir T1 y T2 siguiendo el protocolo 2PL (utilizando locks de lectura y de escritura).

**Solución:**

A continuación se presenta una posible solución agregando los locks y unlocks a T1 y T2 utilizando el protocolo 2PL básico:

**T1': w1(X), w1(X), r1(Y), r1(Y), w1(Z), r1(Z), w1(Z), u1(X), u1(Y), u1(Z), c1**

**T2': r1(Z), r2(X), w2(Y), r2(Y), w2(Y), u2(X), u2(Y), c2**

e) Dar una historia entrelazada de las T1 y T2 de la parte d). Decir si la historia es serializable, justificando.

**Solución:**

**H4: w1(X), w1(X), r1(Y), w1(Z), u1(X), r1(Z), r2(X), r1(Y), u1(Y), w2(Y), r2(Y), w2(Y), r1(Z), w1(Z), u1(Z), c1, u2(X), u2(Y), c2**

La historia se genera a partir de transacciones T1' y T2' que siguen el protocolo 2PL (2PL básico, en particular) Cualquier historia en estas condiciones será serializable por resultado visto en el curso.