

# Examen de Fundamentos de Bases de Datos

Diciembre 2014

## Indicaciones Generales:

- La duracin del examen es de **tres (3)** horas.
- En la prueba **NO** se permite consultar material alguno.
- Empezar cada ejercicio en una hoja nueva.
- Escribir con lpiz y de un solo lado de las hojas.
- Numerar todas las hojas. Incluir en cada hoja la cdula y el nombre. En la primer hoja, incluir la cantidad de hojas que se entregan.

## Ejercicio 1. (27 puntos)

Se est desarrollando un sistema de informacin para la ONASEV (Organizacin Nacional de Seguridad Vial).

Cuando ocurre un accidente, se genera un reporte identificado por la fecha, la hora y el lugar en donde ocurri. Sobre este reporte se asocian posteriormente, la mayor parte de los datos de intere para la ONASEV.

De los lugares, se concen sus coordenadas (latitud y longitud) que lo identifican y un conjunto de referencias. Las referencias son bsicamente direcciones o nombres de parajes por lo que son strings y posiblemente varios.

De los vehculos, se conoce la matrula que lo identifica, la marca, modelo y ao y sus responsables<sup>1</sup>. Todos los vehculos deben tener responsables.

De las personas, se conoce su Cdula de Identidad, nombre, edad, nacionalidad, el lugar donde vive y un conjunto de lugares que frecuenta. Se asume que todas las personas viven en algn lugar. Tambin se asume que slo se registra un lugar como vivienda de una persona.

En cada reporte de un accidente, puede haber un conjunto de vehculos involucrados y en cada vehculo, un conjunto de personas involucradas (al menos una) que no son necesariamente los responsables de los vehculos. Algunas de esas personas estn en alguno de los vehculos y otras pueden no estar en ninguno de los vehculos. Tenga en cuenta que en un accidente debe haber algn vehculo involucrado.

Para cualquiera de las personas involucrada en un accidente, se conoce el estado de esa persona (“ile-so”, “lesiones leves”, etc.)

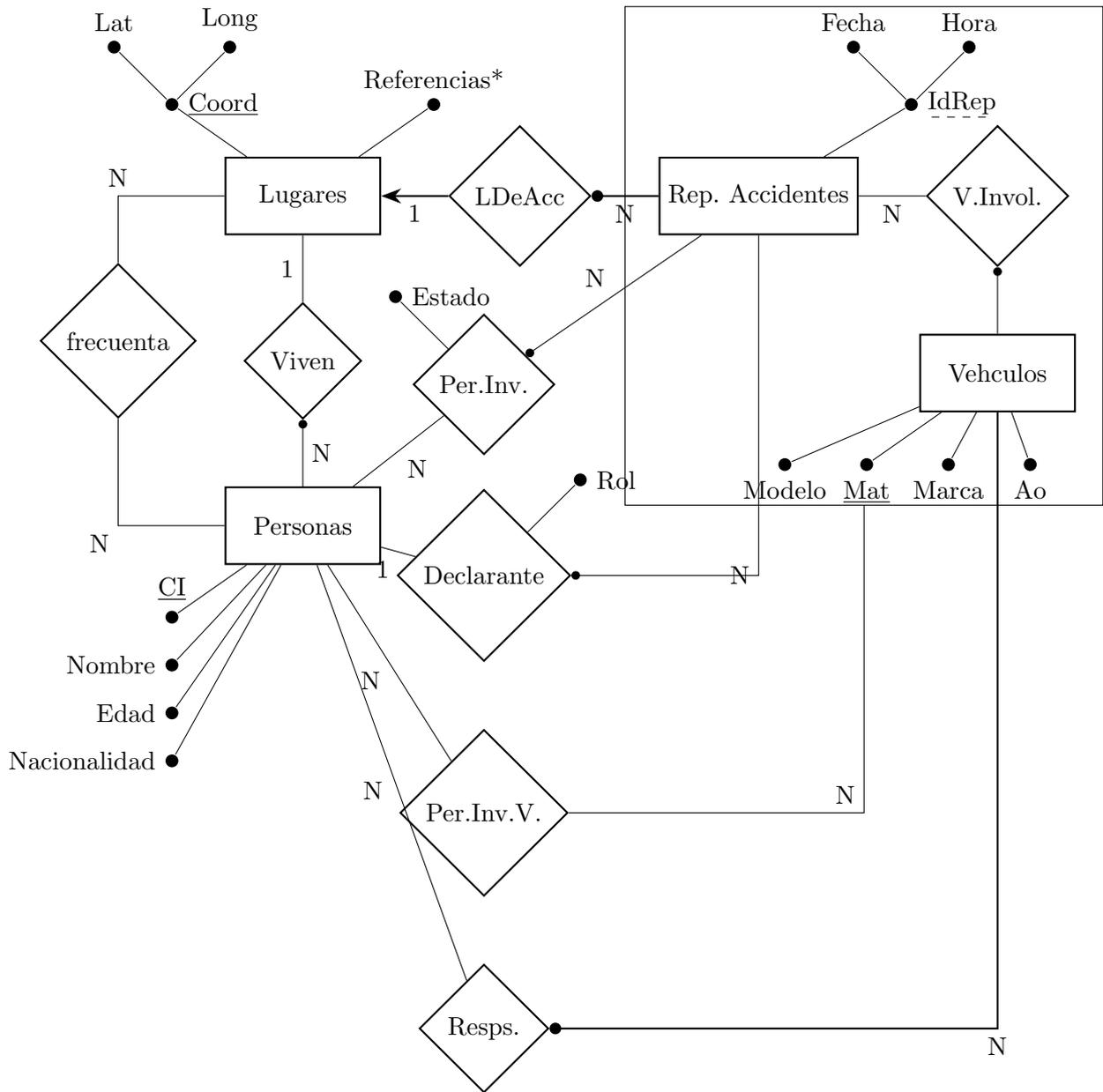
De cada reporte, tambn se conoce un nico declarante del cual se conoce su cdula, y en qu rol hace la declaracin (“seguro”, “polica”, “testigo”, etc.) Un persona que acta como declarante en un accidente, slo lo puede hacer en un solo rol. Por otra parte, no puede ser a la vez declarante e involucrado o responsable de un vehculo involucrado en el mismo accidente.

**Se Pide:** Construya un Modelo Entidad-Relacin completo para representar la realidad descrita en los items anteriores. Debe incluir el diagrama, las restricciones no estructurales y los comentarios pertinentes para la correcta interpretacin de su trabajo.

---

<sup>1</sup>Pueden ser los propietarios o en algunos casos personas designadas por la empresa propietaria

**Solucin.**



RNEs:

- Las personas involucradas en un accidente porque estaban en un vehculo, tambin estn en la relacin *Per.Inv.V.*.  
 $\forall p \in \text{Personas.}$   
 $\forall r \in \text{Rep.Accidentes.}$   
 $\forall v \in \text{Vehiculos.}$   
 $(\langle \langle v, r \rangle, p \rangle \in \text{Per.Inv.V} \rightarrow \langle p, r \rangle \in \text{Per.Inv.})$
- Las personas que funcionan como *declarantes* no puede ser involucradas en el mismo accidente.  
 $\forall p \in \text{Personas.} \forall r \in \text{Rep.Accidentes.} ($   
 $\langle p, r \rangle \in \text{Declarante} \rightarrow$   
 $\neg(\langle p, r \rangle \in \text{Per.Inv.}) \wedge \neg \exists v \in \text{Vehiculos.} (\langle p, v \rangle \in \text{Resps.} \wedge \langle v, r \rangle \in \text{V.Invol.})$

Observar que la primer restriccin, garantiza que se conoce el estado de todas las personas involucradas al incluir a todas las personas involucradas en la relacin *Per.Inv.* . Por otra parte, la segunda utiliza el hecho de que se registran a todas las personas involucradas estn registradas en esa relacin.

**Ejercicio 2.** (24 puntos)

Considrese el esquema relacin  $R(A, B, C, D, E, G)$ , el conjunto de dependencias funcionales  $F = \{A \rightarrow BC, C \rightarrow DG, BD \rightarrow E, AB \rightarrow D, BC \rightarrow G\}$  sobre  $R$ , y la descomposicin  $\rho = \{(ABDE), (BCG), (CDG)\}$

- (a) Determinar si la descomposicin  $\rho$  tiene join sin prdida.

**Solucin.**

Dado que son ms de dos esquemas, se debe aplicar el test tabular de JSP:

$S_1$	A	B	C	D	E	G	$S_2$	A	B	C	D	E	G
(ABDE)	$a_1$	$a_2$	$b_{13}$	$a_4$	$a_5$	$b_{16}$	(ABDE)	$a_1$	$a_2$	$b_{13}$	$a_4$	$a_5$	$b_{16}$
(BCG)	$b_{21}$	$a_2$	$a_3$	$b_{24}$	$b_{25}$	$a_6$	(BCG)	$b_{21}$	$a_2$	$a_3$	$a_4$	$b_{25}$	$a_6$
(CDG)	$b_{31}$	$b_{32}$	$a_3$	$a_4$	$b_{35}$	$a_6$	(CDG)	$b_{31}$	$b_{32}$	$a_3$	$a_4$	$b_{35}$	$a_6$
$S_3$	A	B	C	D	E	G	$S_4$	A	B	C	D	E	G
(ABDE)	$a_1$	$a_2$	$b_{13}$	$a_4$	$a_5$	$b_{16}$	(ABDE)	$a_1$	$a_2$	$b_{13}$	$a_4$	$a_5$	$b_{16}$
(BCG)	$b_{21}$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	(BCG)	$b_{21}$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
(CDG)	$b_{31}$	$b_{32}$	$a_3$	$a_4$	$b_{35}$	$a_6$	(CDG)	$b_{31}$	$b_{32}$	$a_3$	$a_4$	$b_{35}$	$a_6$

Dado que no existe ninguna fila que tenga todos smbolos  $a_j$ , entonces **la descomposicin  $\rho$  no tiene JSP.**

- (b) Determinar que dependencias se proyectan en cada esquema de  $\rho$ .

**Solucin.**

El mecanismo de proyeccin puede parecer oscuro. Hay algunas formas de realizarlo correctamente. La siguiente es una de ellas. Dado un esquema de relacin  $R$

1. Para cada dependencia  $X \rightarrow A$  tal que  $X \subseteq R$  busco encadenamientos hacia adelante va transitiva y pseudo transitiva obteniendo dependencias  $X \rightarrow A'$ . Para cada  $A' \in R$ , agrego  $X \rightarrow A'$  en la proyeccin sobre  $R$ . No es necesario agregar dependencias que se deduzcan de las que ya tenemos.
2. Para cada dependencia  $X \rightarrow A$  tal que  $A \in R$ , busco encadenamientos **hacia atrs** va transitiva y pseudo transitiva obteniendo dependencias  $X' \rightarrow A$ . Para cada  $X' \subseteq R$ , agrego  $X' \rightarrow A$  en la proyeccin sobre  $R$ . No es necesario agregar dependencias que se deduzcan de las que ya tenemos.

De esta forma, en este caso particular que tenemos, para  $R_1(ABDE)$ , se proyectan las dependencias  $A \rightarrow BD, BD \rightarrow E$  que surgen por la regla 1. Por la regla 2, en particular, no se agregan dependencias que no se deduzcan de las dependencias que ya tenemos de esta forma:

$$\prod_{R_1}(F) = \{A \rightarrow BD, BD \rightarrow E\}$$

Aplicando las mismas ideas sobre  $R_2$  y  $R_3$  obtenemos.

$$\prod_{R_2}(F) = \{C \rightarrow G, BC \rightarrow G\}$$

$$\prod_{R_3}(F) = \{C \rightarrow DG\}$$

- (c) Determinar si  $\rho$  preserva dependencias funcionales.

**Solucin.**

Sea  $F' = \left(\bigcup_{i=1}^3 \prod_{R_i}(F)\right)$ . De acuerdo con la definicin se preservarn las dependencias funcionales si la unin de las proyecciones ( $F'$ ) es equivalente a  $F$ . En otras palabras, si se cumple:

$$(F')^+ = F^+$$

Dado que las dependencias de  $F'$  se derivaron de las de  $F$ , se puede afirmar que  $F'$  cubre a  $F$ . Luego, para determinar si  $F'$  cubre a  $F$ , para cada dependencia funcional  $X \rightarrow Y \in F$  se debe calcular  $X_{F'}^+$  y verificar que incluya todos los atributos en  $Y$ . Para simplificar esta tarea, pueden omitirse las dependencias funcionales que se encuentren en  $F' \cap F$  ya que pertenecerán a ambas clausuras.

Sea

$$F' = \left( \bigcup_{i=1}^3 \prod_{R_i}(F) \right) = \{A \rightarrow BD, BD \rightarrow E, C \rightarrow G, BC \rightarrow G, C \rightarrow DG\}$$

Considerando la dependencia funcional  $A \rightarrow C \in F^+$ , se tiene:

$$(A)_{F'}^+ = \{A, B, D, E\} \not\supseteq \{C\}$$

Por tanto,  $A \rightarrow C$  **no se cumple en  $F'^+$** . En consecuencia  $F'$  no cubre a  $F$ , por lo que los conjuntos de dependencias no son equivalentes. Por tanto, **no se preservan las dependencias funcionales**.

- (d) Determinar en que forma normal se encuentra la descomposición  $\rho$  y cada uno de sus esquemas de relacin.

**Solucin.**

-  $R_1(ABDE)$

$$\prod_{R_1}(F) = \{A \rightarrow BDE, BD \rightarrow E, AB \rightarrow DE\}$$

$$(BD)_F^+ = \{B, D, E\}, BD \text{ no es clave de } R_1$$

$$(A)_F^+ = \{A, B, D, E\}, A \text{ es la única clave de } R_1$$

$BD \rightarrow E$  no cumple BCNF ya que  $BD$  no es superclave de  $R_1$ . Tampoco cumple 3NF ya que  $E$  no es un atributo primo. Si cumple con 2NF ya que  $BD$  no es un subconjunto de ninguna clave (dependencia parcial de clave).

-  $R_2(BCG)$

$$\prod_{R_2}(F) = \{C \rightarrow G, BC \rightarrow G\}$$

$$(C)_F^+ = \{C, G\}, C \text{ no es clave de } R_2$$

$$(BC)_F^+ = \{B, C, G\}, BC \text{ es la única clave de } R_2$$

$C \rightarrow G$  no cumple BCNF ya que  $C$  no es superclave de  $R_2$ . Tampoco cumple 3NF ya que  $G$  no es un atributo primo. De la misma forma, no cumple 2NF ya que  $C$  forma parte de una clave de  $R_2$  (dependencia parcial de clave).

-  $R_3(CDG)$

$$\prod_{R_3}(F) = \{C \rightarrow DG\}$$

$$(C)_F^+ = \{C, D, G\}, C \text{ es la única clave de } R_3$$

$C \rightarrow DG$  cumple BCNF ya que  $C$  es clave de  $R_3$ .

Dado que el esquema con menor forma normal es  $R_2$  con 1NF, la descomposición  $\rho_3$  **está en 1NF**.

**Ejercicio 3.** (22 puntos)

- (a) Para la gestión de la concurrencia el DBMS puede elegir, entre otros, los métodos basados en bloqueos (Locks) y el protocolo 2PL. Para estos casos, algunos manejadores consideran varios niveles

de granularidad y autorizan la posibilidad de elegir el nivel más adecuado por parte del usuario. En los casos más sofisticados, para resolver la concurrencia, el manejador elige dinámicamente el mejor nivel de granularidad para cada transacción e inclusive puede cambiar durante la transacción. En base a esto:

- I. Decir cuáles son los distintos niveles de granularidad considerados por un DBMS y explicar brevemente cuál es la motivación de su uso.

**Solucin.**

Los niveles de granularidad definen el nivel de agregación (en algún sentido, el tamaño) de ítem sobre el cual va a estar trabajando el manejador. De esta forma, los niveles de granularidad habituales en una base relacional son: *atributo, tupla, tabla, base o página de disco (bloque)*.

- II. Explicar brevemente cuál es la motivación del uso de diferentes niveles de granularidad.

**Solucin.**

No es lo mismo trabajar a cualquier nivel de granularidad.

Cuanto mayor es el tamaño del “grnulo” (mayor granularidad), menor es el nivel de concurrencia permitido, ya que de esta forma se restringe el acceso a las partes de los objetos.

Cuanto menor es el tamaño del “grnulo” (menor granularidad), mayor es el overhead que se impone en el sistema, porque hay que llevar el control de una mayor cantidad de timestamps o locks.

- III. Qué consideraciones tendrá respecto al nivel de granularidad si sabe que cada transacción accede a pocos registros?

Y para el caso en que cada transacción accediera a muchos registros?

Justifique sus respuestas.

**Solucin.**

Si las transacciones acceden a pocos registros, es mejor manejar un nivel de granularidad baja. De esta forma, se fomenta una alta concurrencia, permitiendo que una mayor cantidad de transacciones trabajen simultáneamente sobre los ítems más “gruesos” (base, tabla, etc). En este caso, se espera que la cantidad de conflictos sea muy baja. De lo contrario, si se acceden a muchos registros de una misma tabla, es mejor manejar un nivel de granularidad alta.

- (b) Considere las definiciones de recuperabilidad y seriabilidad.

- I. Explicar brevemente que significa que dos historias son equivalentes por conflictos.

**Solucin.**

Dos historias son equivalentes por conflicto cuando cada pareja de operaciones en conflicto estén en el mismo orden en las dos historias.

- II. Explicar brevemente que significa que una historia es serializable por conflictos.

**Solucin.**

Una historia es serializable por conflictos cuando es equivalente por conflictos a una historia serial sobre las mismas transacciones.

- (c) Existe otro mecanismo para garantizar la seriabilidad de las historias conocido como Basado en Timestamps

- I. Describa en no más de dos líneas que es un Timestamp.

**Solucin.**

Un Timestamp (TS) es un identificador que se asocia a cada transacción tal que si  $TS(T_i) < TS(T_j)$  entonces  $T_i$  comenzó antes que  $T_j$ .

- II. Describa en no más de cuatro líneas cuál es la idea fundamental de la seriabilidad basada en Timestamps.

**Solucin.**

La idea detrás de la seriabilidad basada en Timestamps es que las lecturas y escrituras se hagan siempre en el orden de timestamps. Si este orden se viola en algún momento, la transacción que lo está violando, debe abortar y reiniciarse.

**Ejercicio 4.** (27 puntos)

Una empresa dedicada a la realización de mudanzas maneja su información en una base de datos con el siguiente esquema relacional:

**MUDANZAS** (codMud, ciCli, direcciónOrigen, direcciónDestino, nroCajas, nroMuebles, fechaInicio)

Para cada una de las mudanzas que realiza se conoce un código que la identifica, la cédula del cliente que contrata el servicio, las direcciones de origen y de destino, la cantidad de cajas y muebles a mover, y la fecha de inicio de la mudanza. Las mudanzas donde coinciden la dirección de origen y de destino se llaman “mudanzas internas”.

**EMPLEADOS** (ci, nombre, fechaIngreso, especialidad)

De los empleados de la empresa se conoce un nombre que lo identifica, la fecha de ingreso y la especialidad de los mismos.

**ASIGNACION** (ci, codMud)

En esta tabla se mantiene la asignación de empleados a las mudanzas.

**CAMIONES** (mat, ciChofer, costoDiario)

La empresa no posee vehículos propios. En caso de necesitarlos los contrata por fuera. Cada vehículo contratado tiene un chofer propio que no es empleado de la empresa. En esta tabla se mantiene información de los camiones con que trabaja la empresa. Para cada uno de ellos se conoce su matrícula, la cédula del chofer y cuánto paga la empresa por día que contrata el camión.

**CONTRATACIONES** (codMud, mat)

La empresa no posee camiones propios para realizar las mudanzas, en caso de necesitarlos los contrata puntualmente.

En estos esquemas se cumplen las siguientes dependencias de inclusión:

$$\Pi_{ci}(ASIGNACION) \subseteq \Pi_{ci}(EMPLEADOS)$$

$$\Pi_{codMud}(ASIGNACION) \subseteq \Pi_{codMud}(MUDANZAS)$$

$$\Pi_{codMud}(CONTRATACIONES) \subseteq \Pi_{codMud}(MUDANZAS)$$

$$\Pi_{mat}(CONTRATACIONES) \subseteq \Pi_{mat}(CAMIONES)$$

**(a) En Álgebra Relacional:**

- I. Obtener las cédulas de identidad de los choferes que solo fueron contratados en mudanzas con más de 10 cajas.

**Solucin.**

A son los choferes que fueron contratados en alguna mudanza con más de 10 cajas

$$A = \Pi_{ciChofer}(\sigma_{nroCajas > 10}(MUDANZAS) * CONTRATACIONES * CAMIONES)$$

B son los choferes que fueron contratados en alguna mudanza con 10 o menos cajas

$$B = \Pi_{ciChofer}(\sigma_{nroCajas \leq 10}(MUDANZAS) * CONTRATACIONES * CAMIONES)$$

$SOL$  es la diferencia entre  $A$  y  $B$ .

$$SOL = A - B$$

- II. Cédula y nombre de los empleados que fueron asignados a las mudanzas que se hicieron el primer día registrado en el sistema en el que se realizaron mudanzas.

**Solucin.**

$A$  contiene las fechas de inicio de las mudanzas.

$$A = \Pi_{fechaInicio}(MUDANZAS)$$

$B$  contiene la fecha mínima del inicio de las mudanzas.

$$B = (A \bowtie_{fechaInicio \leq fechaMayor} \rho_{fechaInicio \rightarrow fechaMayor}(A)) \% A$$

$SOL$  contiene la Cédula y nombre de los empleados que fueron asignados a las mudanzas realizadas en esa fecha.

$$SOL = \Pi_{ci,nombre}(EMPLEADOS * ASIGNACION * B)$$

(b) **En Cálculo Relacional:**

- I. Cédula de los choferes contratados para realizar mudanzas donde no se asignaron empleados de especialidad ARMADOR.

**Solucin.**

$$\{c.ciChofer / CAMIONES(c) \wedge (\exists u)(CONTRATACIONES(u) \wedge u.mat = c.mat \wedge \neg(\exists e)(EMPLEADOS(e) \wedge e.especialidad = ARMADOR \wedge (\exists a)(ASIGNACION(a) \wedge a.ci = e.ci \wedge a.codMud = u.codMud)))\}$$

- II. Especialidad de los empleados que fueron asignados solo a mudanzas internas.

**Solucin.**

$$\{e.especialidad / EMPLEADOS(e) \wedge (\exists a)(ASIGNACION(a) \wedge (\exists l)(MUDANZAS(l) \wedge a.codMud = l.codMud \wedge a.ci = e.ci \wedge l.direccionOrigen = l.direccionDestino)) \wedge \neg(\exists a1)(ASIGNACION(a1) \wedge (\exists l1)(MUDANZAS(l1) \wedge a1.ci = e.ci \wedge a1.codMud = l1.codMud \wedge l1.direccionOrigen \neq l1.direccionDestino))\}$$

(c) **En SQL (sin usar vistas ni subconsultas en el FROM):**

- I. Fecha de inicio y dirección de origen de las mudanzas para las cuales se contrataron más de 20 camiones pero se asignaron menos de 10 empleados.

**Solucin.**

```
SELECT L.fechaInicio, L.direccionOrigen
FROM MUDANZAS L
WHERE L.codMud IN
    (SELECT C.codMud
     FROM CONTRATACIONES C
     GROUP BY C.codMud
     HAVING count(*) >= 20)
```

INTERSECT

```
SELECT A.codMud
FROM ASIGNACION A
GROUP BY A.codMud
HAVING count(*) < 10
)
```

- II. Para cada mudanza obtener: c3digo de la mudanza, especialidad y la cantidad de empleados asignados de esa especialidad, considerando 3nicamente las especialidades donde fueron asignados a la mudanza por lo menos 3 empleados.

**Solucin.**

```
SELECT A.codMud, E.especialidad, count(*)
FROM ASIGNADOS A natural join EMPLEADOS E
GROUP BY A.codMud, E.especialidad
HAVING count(*) >3
```