

Examen de Fundamentos de Bases de Datos

Febrero 2015

Indicaciones Generales:

- La duración 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 lápiz y de un solo lado de las hojas.
- Numerar todas las hojas. Incluir en cada hoja la cédula y el nombre. En la primer hoja, incluir la cantidad de hojas que se entregan.

Ejercicio 1. (25 puntos)

Una agencia de viajes desea modelar la planificación de distintos paquetes turísticos realizados en barcos. De los paquetes se conoce su identificador, su nombre, la fecha de partida, la fecha de llegada, el puerto de origen y el puerto de destino. A cada paquete se le asignan barcos, de los cuales se conoce su matrícula, que lo identifica, y su nombre. También se conoce el conjunto de camarotes que hay en cada barco, y cada camarote tiene un nombre, un tipo (suite, con balcón o interior) y un número que lo identifica dentro del barco.

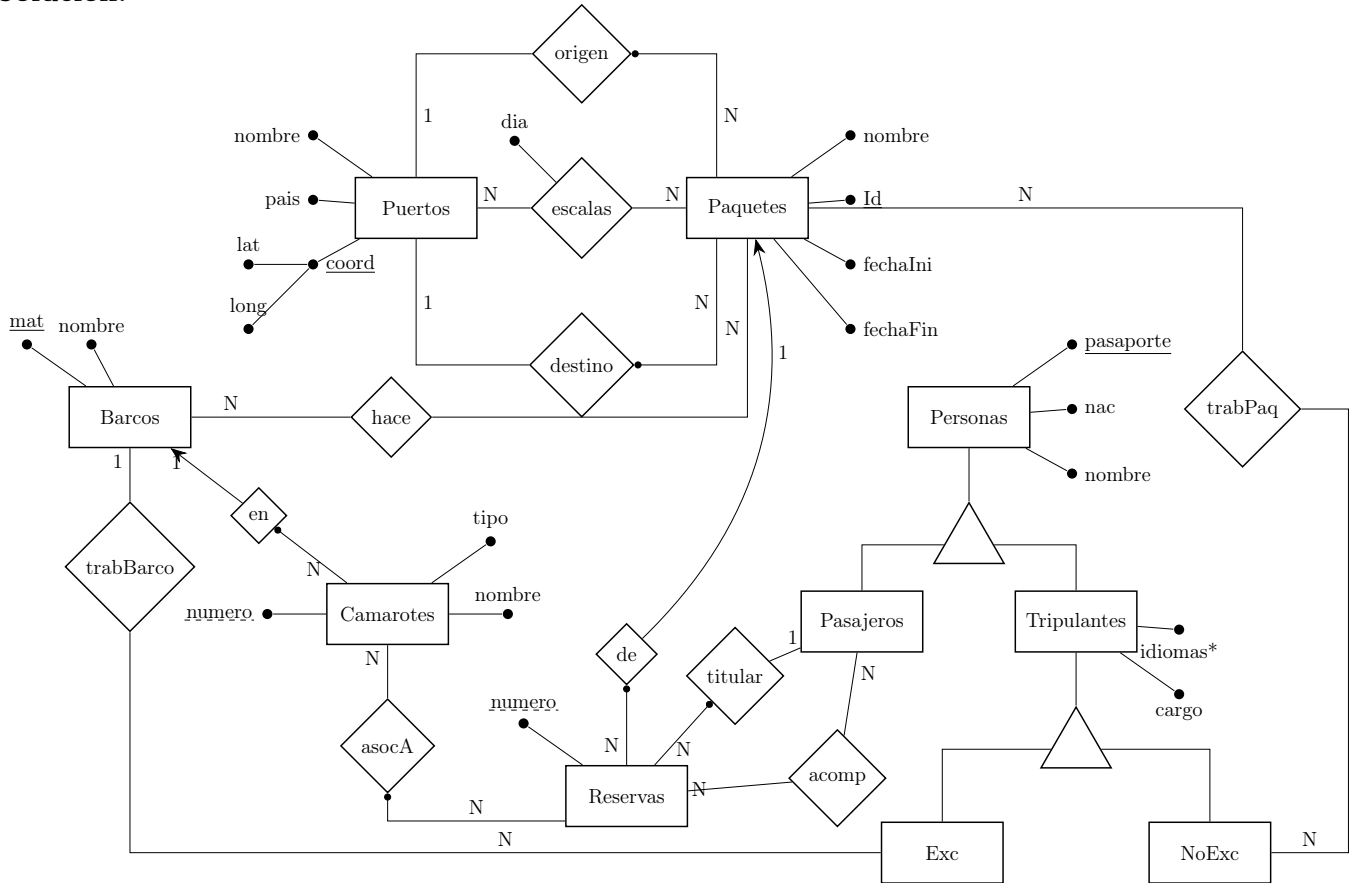
De los puertos se conoce su nombre, el país al cual pertenece y sus coordenadas (latitud, longitud), estas últimas lo identifican. De cada paquete se conoce un conjunto de puertos donde se realizan escalas. Interesa saber en qué día relativo al paquete se realiza cada escala. Por ejemplo, en el paquete *Paseo por México* el día 1 se hace escala en el puerto *Isla Blanca*, el día 4 se hace escala en el puerto *Isla Mujeres* y el día 7 se hace escala en el puerto *Cozumel*.

De los pasajeros se conoce el pasaporte, el nombre y su nacionalidad. Estos realizan reservas de paquetes turísticos. En ese momento, la agencia le entrega un documento (voucher) de reserva que tiene un número de reserva y el resto de los datos correspondientes. Estos datos incluyen: datos de un único titular de la reserva, datos de los acompañantes, datos del paquete que reserva, y un conjunto de camarotes asociados a la reserva. Tenga en cuenta que: titulares de reservas pueden tener reservas con igual número en paquetes turísticos diferentes pero no en el mismo paquete, y que todas las reservas tienen al menos un camarote asociado.

De los tripulantes de los barcos, se conoce el pasaporte, el nombre, su nacionalidad, el cargo (por ejemplo, capitán), y los idiomas que habla. Se sabe que algunos miembros de la tripulación trabajan exclusivamente en un barco (por ejemplo, el capitán, los técnicos, el ingeniero, etc.), mientras que otros son asignados a paquetes turísticos (por ejemplo, los animadores). Por otra parte, tenga en cuenta que los propios tripulantes pueden ser pasajeros en diversos cruceros.

Se Pide: Mer completo incluyendo restricciones no estructurales.

Solución.



RNE's:

- $Exc \cup NoExc = Tripulantes$
- $Exc \cap NoExc = \emptyset$
- $Pasajeros \cup Tripulantes = Personas$
- Una *Reserva* sólo puede tener asociados *Camarotes* en *Barcos* que hacen el *Paquete* de la *Reserva*.

$\forall r \in Reservas.$

$\forall c \in Camarotes.$

$\forall b \in Barcos.$

$\forall p \in Paquetes.$

$\langle r, c \rangle \in asocA \wedge \langle c, b \rangle \in en \wedge \langle r, p \rangle \in de \rightarrow \langle b, p \rangle \in hace$

- Un *Tripulante* no puede ser *Pasajero* en un *Paquete* en el que es *Tripulante*.

$\forall t \in Tripulantes. \forall p \in Paquete.$

$\neg \exists r \text{ in } Reserva. (\langle t, r \rangle \in titular \vee \langle t, r \rangle \in acomp) \wedge$

$(\langle t, p \rangle \in trabPaq \vee \exists b \in Barcos. (\langle t, b \rangle \in trabBarco \wedge$

$\langle b, p \rangle \in hacen))$

Ejercicio 2. (30 puntos)

Una institución deportiva almacena los información de sus socios, profesores y actividades en una base de datos con el siguiente esquema relacional:

PERSONAS (ci, nombre, domicilio, teléfono)

De cada persona (socio o profesor) se conoce su cédula que la identifica, su nombre completo, domicilio y teléfono.

SOCIOS (ci, fechaIngreso, fechaNacimiento, bonificación)

De cada socio, también se conocen sus fechas de ingreso y de nacimiento, junto ccon el porcentaje de bonificación de la cuota mensual.

PROFESORES (ci, especialidad)

De cada profesor, también se conoce la disciplina de su especialidad (ej: natación, fútbol, etc).

HORARIOS (idHorario, disciplina, ci, diaSemana, hora)

De cada horario que ofrece la institución deportiva se conoce un código que lo identifica, la cédula del profesor encargado, el día de la semana, la hora del día, y la disciplina (ej: natación, fútbol, etc). **NOTA:** esta relación tiene una clave candidata compuesta por los atributos **ci**, **diaSemana** y **hora**. Esto último impide que un profesor pueda tener más de un horario asignado al mismo tiempo.

ASISTENCIAS (ciSocio, idHorario, fecha)

La institución lleva un registro de los horarios a los que asiste cada socio en cada fecha.

En estos esquemas se cumplen las siguientes dependencias de inclusión:

$$\Pi_{ci}(SOCIOS) \subseteq \Pi_{ci}(PERSONAS)$$

$$\Pi_{ci}(PROFESORES) \subseteq \Pi_{ci}(PERSONAS)$$

$$\Pi_{ci}(HORARIOS) \subseteq \Pi_{ci}(PROFESORES)$$

$$\Pi_{idHorario}(ASISTENCIAS) \subseteq \Pi_{idHorario}(HORARIOS)$$

$$\Pi_{ciSocio}(ASISTENCIAS) \subseteq \Pi_{ci}(SOCIOS)$$

Se Pide: Resolver las siguientes consultas en los lenguajes indicados.

(a) **En Álgebra Relacional:**

Obtener la identificación del horario, disciplina que se dicta y el nombre completo del profesor que lo dicta, de los horarios en los que nunca fue nadie y todos los profesores tienen un horario en que se dicta esa disciplina.

Solución.

Disciplinas que tiene horarios asignados para todos los profesores.

$$A = \Pi_{disciplina,ci}(Horarios) \div \Pi_{ci}(Profesores)$$

Horarios de las disciplinas de A que no tienen asistencia.

$$B = \Pi_{idHorario}(Horarios * A) - \Pi_{idHorario}(Asistencias)$$

Horarios, nombres de profesores y disciplinas (de las anteriores) en los que nunca hubo asistencia pero los podría dictar cualquier otro (SOLUCION).

$$SOL = \Pi_{idHorario,nombre,disciplina}(B * Horarios * Profesores * Personas)$$

(b) **En Cálculo Relacional:**

- I. Obtener la cédula de los profesores que únicamente tienen horarios en la disciplina que es su especialidad

Solución.

$$\{p.ci / PROFESORES(p) \wedge (\exists h_1)(HORARIOS(h_1) \wedge h_1.ci = p.ci \wedge h_1.disciplina = p.especialidad \wedge \neg(\exists h_2)(HORARIOS(h_2) \wedge h_2.ci = p.ci \wedge h_2.disciplina \neq p.especialidad))\}$$

- II. Obtener la cédula y la bonificación que se le otorga a los socios con mayor edad dentro de los que tienen mayor antigüedad.

Solución.

$$\{s.ci, s.bonificacion / SOCIOS(s) \wedge (\forall s_1)(SOCIOS(s_1) \rightarrow s_1.fechaIngreso \geq s.fechaIngreso) \wedge (\forall s_2)(SOCIOS(s_2) \wedge s.fechaIngreso = s_1.fechaIngreso \rightarrow s_2.fechaNacimiento \geq s.fechaNacimiento)\}$$

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

- I. Nombre completo y especialidad de los profesores que poseen la mayor cantidad de horarios dentro de su especialidad

Solución.

La consulta admite 3 soluciones.

```

1.      SELECT Per.nombre, Prof.especialidad
        FROM Personas Per JOIN
              Profesores Prof ON Prof.ci = Per.ci JOIN
              Horarios Hor ON Hor.ci = Prof.ci
        WHERE Prof.especialidad = hor.disciplina
        GROUP BY Per.ci, Per.nombre, Prof.especialidad
        HAVING COUNT(*) >= ALL (
              SELECT COUNT(*)
              FROM Profesores P JOIN
                    Horarios H ON H.ci = P.ci
              WHERE P.especialidad = h.disciplina
              GROUP BY P.ci, P.especialidad
        )

```

2.

```
SELECT Per.nombre, Prof.especialidad
FROM Personas Per JOIN
     Profesores Prof ON Prof.ci = Per.ci JOIN
     Horarios Hor ON Hor.ci = Prof.ci
WHERE Prof.especialidad = hor.disciplina
GROUP BY Per.ci, Per.nombre, Prof.especialidad
HAVING COUNT(*) >= ALL (
     SELECT COUNT(*)
     FROM Horarios H
     WHERE Prof.especialidad = H.disciplina
     GROUP BY H.ci
)
)
```
3.

```
SELECT Per.nombre, Prof.especialidad
FROM Personas Per JOIN
     Profesores Prof ON Prof.ci = Per.ci JOIN
     Horarios Hor ON Hor.ci = Prof.ci
WHERE Prof.especialidad = Hor.disciplina
GROUP BY Per.ci, Per.nombre, Prof.especialidad
HAVING COUNT(*) >= ALL (
     SELECT COUNT(*)
     FROM Profesores P JOIN
           Horarios H ON H.ci = P.ci
     WHERE Prof.especialidad = H.disciplina and
           Prof.especialidad = P.especialidad
     GROUP BY P.ci
)
)
```

- II. Para cada disciplina, devolver disciplina y cantidad de horarios en que es dictada por profesores cuya especialidad es la disciplina.

Solución.

```
SELECT disciplina, COUNT(*)
FROM Horarios
WHERE CI in ( Select ci
              from Profesores
              where especialidad=disciplina )
GROUP BY disciplina
```

Ejercicio 3. (25 puntos)

Considere el esquema $R(A, B, C, D, E, G, H)$ y las siguientes dependencias funcionales:

$$F = \{A \rightarrow B, B \rightarrow CD, D \rightarrow G, CG \rightarrow E\}$$

- (a) Considere la siguiente descomposición: $\rho_1 = \{R_1(A, C, D), R_2(B, G, E), R_3(A, H)\}$
- I. Calcule las proyecciones de F^+ sobre cada uno de los subesquemas de ρ_1 . Justifique su respuesta.

Solución.

Una de las formas correctas de hacer las proyecciones es hacer los siguientes 2 pasos:

1. Tomar partes izquierdas de las dependencias que aparezcan en el esquema y hacer encadenamientos “hacia adelante” usando transitiva y pseudo transitiva.
2. Tomar los lados atributos individuales de las partes derechas de las dependencias que aparezcan en el esquema y hacer encadenamientos “hacia atrás” usando transitiva.

Las dependencias encontradas de por una de estas formas y que cumplen que todos los atributos están en el esquema, quedan en la proyección.

De esta forma

$$\Pi_{R_1}(F) = \{A \rightarrow CD\}$$

$$\Pi_{R_2}(F) = \{B \rightarrow GE\}$$

$$\Pi_{R_3}(F) = \{\}$$

En este caso, todas las dependencias salieron directamente por el primer paso, pero puede haber dependencias que salgan por el segundo.

- II. Determine si se pierden dependencias. Justifique su respuesta.

Solución.

Para ver si ρ_1 pierde dependencias, debemos chequear si las dependencias de F se deducen de $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) \cup \Pi_{R_3}(F)$.

Al hacer esto nos encontramos que $A \rightarrow B$ y $B \rightarrow C$ (y todas las que se deducen de ellas) se perdieron.

- (b) La descomposición $\rho_2 = \{R_1(A, C, D), R_2(A, B, G, E, H)\}$ tiene join sin pérdida?. Justifique su respuesta.

Solución.

Dado que ρ_2 tiene sólo dos esquemas, se puede aplicar el teorema del join sin pérdida, por lo que basta con verificar si en F^+ está alguna de las siguientes dependencias:

$$A \rightarrow CD$$

$$A \rightarrow BGEH$$

Dado que la primera dependencia está directamente en la proyección de R_1 , se puede afirmar que ρ_2 tiene join sin pérdida.

- (c) Un programador carga una base con el esquema ρ_2 , usando las siguientes vistas sobre R :

$$R_1 = \Pi_{A,C,D}(R)$$

$$R_2 = \Pi_{B,G,E}(R) * \Pi_{A,H}(R)$$

- I. Justifique claramente (con ejemplo si fuera necesario) por qué el trabajo del programador está errado.

Solución.

Al realizar la carga de esa forma, el programador no está utilizando realmente la descomposición ρ_2 sino ρ_1 . Esto es porque para reconstruir R , la operación que debe realizar sería equivalente a: $\Pi_{A,C,D}(R) * \Pi_{B,G,E}(R) * \Pi_{A,H}(R)$

Esto corresponde al join de todas las tablas de ρ_1 por lo que la pregunta que debió hacerse el programador era: ρ_1 tiene join sin pérdida?

La primera sospecha de que esto no se cumple debería surgir del hecho de que se pierden dependencias. La pérdida de dependencias no siempre obliga a un join **con** pérdida pero, si

las dependencias que se pierden participan del test, entonces pueden aparecer pérdidas de información en el join.

Una forma de ver que esto es así, es realizar el test tabular de JSP sobre ρ_1 .

La inicialización, deja la siguiente tabla.

	A	B	C	D	E	G	H
R_1	a_1	b_{12}	a_3	a_4	b_{15}	b_{16}	b_{17}
R_2	b_{21}	a_2	b_{23}	b_{24}	a_5	a_6	b_{27}
R_3	a_1	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}	a_7

Luego de aplicar la dependencia $A \rightarrow CD$, se obtiene la tabla final dado que no hay dos tuplas que tengan el mismo valor en B:

	A	B	C	D	E	G	H
R_1	a_1	b_{12}	a_3	a_4	b_{15}	b_{16}	b_{17}
R_2	b_{21}	a_2	b_{23}	b_{24}	a_5	a_6	b_{27}
R_3	a_1	b_{32}	a_3	a_4	b_{35}	b_{36}	a_7

Dado que ninguna fila tiene sólo a's, entonces no hay JSP.

Solución.

(OTRA) Para construir R_2 , el programador hace la siguiente operación:

$$R_2 = \Pi_{B,G,E}(R) * \Pi_{A,H}(R)$$

Esta operación es join natural, pero en realidad es un producto cartesiano dado que no hay atributos en común. Por esto, es fácil poner un ejemplo en donde se hace evidente el error. Una posible instancia de R es la siguiente, en donde asumimos que todos los valores son distintos.

A	B	C	D	E	G	H
a_1	b_1	c_1	d_1	e_1	g_1	h_1
a_2	b_2	c_2	d_2	e_2	g_2	h_2

Si de esta tabla, calculamos las proyecciones anteriores obtenemos las siguientes instancias:

Para $\Pi_{B,E,G}(R)$:

B	E	G
b_1	e_1	g_1
b_2	e_2	g_2

Para $\Pi_{A,H}(R)$:

A	H
a_1	h_1
a_2	h_2

Al hacer el producto cartesiano se van a obtener las siguientes tuplas que no están en la tabla original:

A	B	C	D	E	G	H
a_2	b_1	c_1	d_1	e_1	g_1	h_2
a_1	b_2	c_2	d_2	e_2	g_2	h_1

Esto muestra que hay casos en donde no es posible reconstruir la información original.

II. Sugiera una corrección. Justifique su respuesta.

Solución.

La forma correcta de cargar la base, es precisamente proyectando R según ρ_2 :

$$R_1 = \Pi_{A,C,D}(R)$$

$$R_2 = \Pi_{A,B,G,E,H}(R)$$

Esto es así, dada la definición se JSP.

Ejercicio 4. (20 puntos)

Dada la siguiente consulta en SQL donde C_1, C_2 y C_3 son condiciones simples¹ sobre atributos:

```
SELECT A1, A2, A3
FROM T1, T2, T3
WHERE C1 and C2 and C3
```

- (a) Describa cómo construir el árbol canónico a partir de la consulta SQL especificada anteriormente. Construya el árbol canónico para la consulta de ejemplo.

Solución.

Si comienzo por las hojas del árbol, por cada tabla especificada en la cláusula FROM agregó una hoja. Comenzando desde la izquierda y respetando el orden en el que aparecen, agregó las primeras dos tablas (en caso de haber más de una), relacionándolas con el operador de producto cartesiano. En caso de que hayan más de dos tablas, las demás se irán agregando a la derecha, relacionándolas con el resultado anterior utilizando el operador de producto cartesiano.

Para el caso del ejemplo, primero se agregan las tablas T_1 e T_2 , relacionándolas con el operador de producto cartesiano, y luego al resultado de esta operación se le aplica el producto cartesiano con la tabla T_3 .

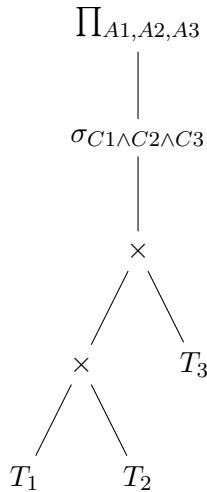
Al resultado anterior, se le aplica una selección con las condiciones de JOIN, y las especificadas en la cláusula WHERE.

Para el caso del ejemplo, al resultado de los productos cartesianos de las tablas T_1 , T_2 y T_3 , se le aplica una selección con la condición $C_1 \wedge C_2 \wedge C_3$.

Como raíz del árbol, se tiene la proyección por los atributos especificados en la cláusula SELECT. Para el caso del ejemplo, al resultado de la selección se le aplica una proyección por los atributos A_1, A_2 y A_3 .

El árbol canónico se especifica en la siguiente figura:

¹sin subconsultas



(b) Enumere las reglas heurísticas vistas en el curso para obtener un plan lógico. Dé un ejemplo práctico de dos de ellas sobre la consulta anterior, explicando por qué es conveniente aplicar esa regla.

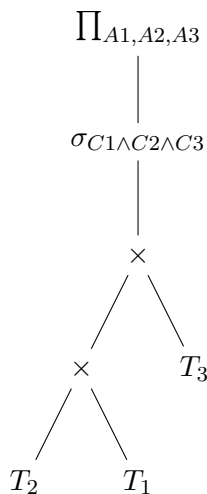
Solución.

Estas reglas heurísticas se basan en aplicar equivalencias de los operadores del álgebra:

1. Cambiar las selecciones conjuntivas por una “cascada” de selecciones simples.
2. Mover las selecciones lo más abajo que se pueda en el árbol.
3. Poner a la izquierda de los productos cartesianos las hojas que generen menos tuplas.
4. Cambiar secuencias de selecciones y productos cartesianos por JOINS.
5. Mover las proyecciones lo más abajo posible en el árbol, agregando las proyecciones que sean necesarias.

Regla 3.

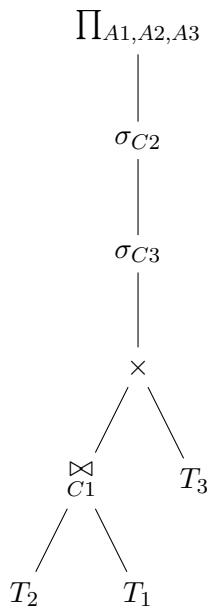
Tomando como ejemplo el árbol canónico construido en la parte a) de este ejercicio, supongamos que se consulta el catálogo y la cantidad de tuplas de T_1 es 100 y la cantidad de tuplas de T_2 es 10. Aplicando esta regla, intercambiaríamos las tablas para que la tabla T_2 me quede a la izquierda, quedando el árbol de esta forma:



Es conveniente aplicarla para reducir el costo del algoritmo utilizado para el JOIN, ya que en todas las implementaciones se recorre íntegramente la tabla izquierda.

Regla 4.

Tomando como ejemplo el árbol canónico construido en la parte a) de este ejercicio, supongamos que la condición $C1$ es una condición de JOIN entre las tablas T_2 y T_1 . Aplicando esta regla, se cambia el producto cartesiano entre estas tablas por un JOIN con la condición $C1$, quedando el árbol de esta forma:



Es conveniente aplicarla para reducir el costo de implementación, ya que los algoritmos de JOIN pueden ser mucho más eficientes que un producto cartesiano.

- (c) Elija una de las implementaciones del operador de selección considerando una condición de igualdad de una clave primaria con un valor y describa sus características (restricciones, descripción de cómo se implementa y costos).

Solución.

Como ejemplo, se propone el algoritmo de búsqueda lineal. Hay que tener en cuenta que la condición es de igualdad por una clave por lo que sólo se recupera un registro de la tabla.

- Restricciones de uso: ninguna.
- Descripción: leer cada registro y si cumple la condición se pone en el resultado.
- Costos de lectura:
 - Peor caso: b_R (cantidad de bloques de la relación R). Esto se debe a que en el peor caso, el registro no está en la tabla.
 - Promedio: $b_R/2$.