



I PARCIAL

- 1) (1,25 puntos) ¿Qué características añade el modelo de datos relacional orientado a objetos con respecto al modelo relacional?

Solución (10 min):

Son características derivadas de la POO, como:

- La posibilidad de trabajar con tipos de datos más elaborados; en concreto, con tipos de datos estructurados.
 - Encapsula los tipos de datos con sus operaciones mediante métodos y propiedades.
 - Herencia de tipos y de tablas.
- 2) (1,25 puntos) Dados los esquemas $a(u)$, $b(v)$, $c(u)$, $d(u)$, traducir a Datalog la siguiente consulta SQL:
- ```

SELECT u, v
FROM c, d
WHERE c.u = d.u
UNION
SELECT u, v
FROM a, b
WHERE NOT (u IN SELECT u FROM c);

```

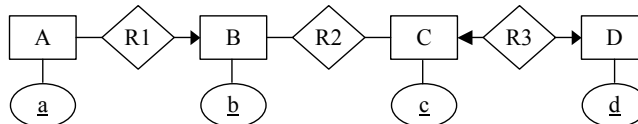
Solución (10 min):

```

r(U,V) :- c(U), d(U).
r(U,V) :- a(U), b(V), not c(U).

```

- 3) (1,9 puntos) Dado el siguiente esquema entidad-relación se pide:
- Traducirlo a una única relación del modelo relacional y expresar las restricciones de cardinalidad con dependencias funcionales.
  - Determinar las claves candidatas de la relación obtenida.



Solución (15 min):

a)

Relación:

$r(a,b,c,d)$

Dependencias funcionales:

$a \rightarrow b$

$c \rightarrow d$

$d \rightarrow c$

b)

Las claves candidatas deben contener al atributo a porque éste no se encuentra en ningún consecuente de las dependencias funcionales:

$\{a\} \neq \{a,b\}$  Sólo con a es insuficiente, hay que agregar c o d:

$\{ac\} \neq \{a,b,c,d\}$  Clave candidata

$\{ad\} \neq \{a,b,c,d\}$  Clave candidata

- 4) (5,6 puntos) Se ha encargado a una empresa de servicios informáticos que realice la facturación de los consumos que se producen en una red eléctrica compartida por diferentes compañías. Se desea tener información sobre los clientes (con su NIF, nombre y Dirección) que consumen la electricidad, las compañías (con su CIF y nombre) y los apuntes que representan el consumo (en kW/h) de un cliente en una fecha en concreto. Los clientes eligen la compañía que les presta el servicio entre dos fechas (pueden cambiar en cualquier momento de compañía). Las compañías aplican tarifas válidas entre dos fechas que indican el coste por kW/h. Una compañía sólo puede aplicar una tarifa cada día, independientemente del cliente. Un cliente se entiende como tal si ha elegido una compañía. Se pide:
- Usar el modelo entidad-relación para construir el esquema conceptual sin redundancias y sin considerar operaciones. No se deben introducir atributos artificiales que actúen como identificadores.

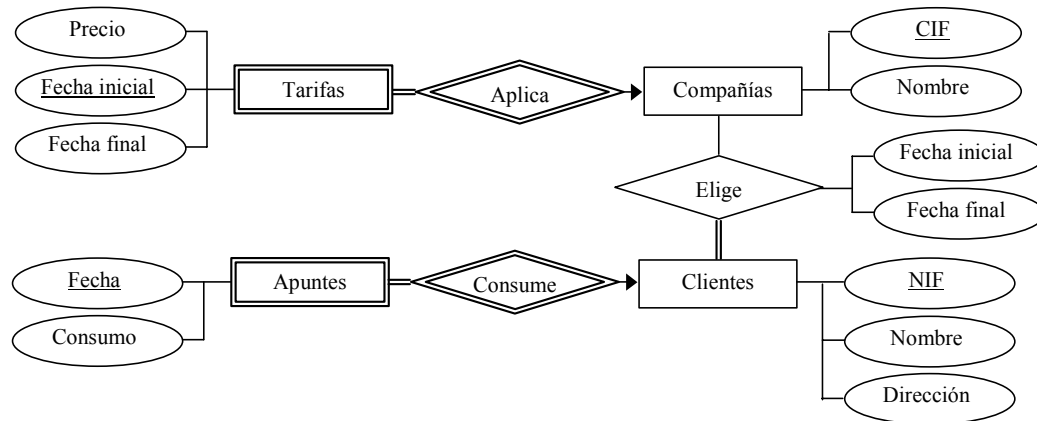


Explicar las claves, las restricciones de cardinalidad una a una, una a varias o varias a varias, y las que no se puedan expresar que aparezcan en el esquema propuesto.

- Usar el esquema anterior para construir el esquema lógico usando el modelo relacional y explicando todas las restricciones que se impongan. Indicar las tablas que se podrían omitir por contener información redundante. Indicar todas las restricciones de integridad (incluidas las dependencias funcionales, si surgen) y expresar las de integridad referencial y de participación total en notación algebraica.
- Plantear una consulta SQL que, dado el nombre de un cliente y un mes, permita emitir la factura de este cliente en el mes indicado. El resultado de esta consulta debe listar: el consumo en kW/h, la fecha en la que se ha realizado este consumo, su importe, la compañía que ha facturado y la tarifa que ha aplicado.

Solución (45 min):

a)



Claves:

- Entidades fuertes:
  - Compañías: El CIF identifica unívocamente a una compañía.
  - Clientes: El NIF identifica unívocamente a un cliente.
- Entidades débiles:
  - Tarifas: Una tarifa está identificada unívocamente por la fecha en que se comienza a aplicar y la compañía a la que corresponde (identificada unívocamente por su CIF). No es necesario incluir Fecha final en la clave porque nunca se da el caso de que la misma compañía aplique dos tarifas diferentes en la misma fecha.
  - Apuntes: Un apunte está identificado unívocamente por la fecha en que se produce el apunte y el cliente al que corresponde.

Restricciones de cardinalidad:

- Tarifas – Aplica -> Compañías: Dada una tarifa, ésta corresponde a una única compañía (También se podría asumir que una tarifa pueda corresponder a varias compañías). Dada una compañía, en general le corresponden varias tarifas.
- Apuntes – Consumo -> Clientes: Dado un apunte, éste corresponde a un solo cliente. Dado un cliente, en general le corresponden varios apuntes.
- Clientes – Elige – Compañías: Un cliente puede elegir varias compañías (en diferentes momentos). Una compañía puede ser elegida por varios clientes.

Restricciones de participación total:

- Un cliente se entiende como tal si ha elegido una compañía.

Restricciones no expresables en este esquema:

- Un cliente no puede elegir diferentes compañías en el mismo momento.
- Si un cliente elige una compañía en un intervalo de tiempo, la compañía debe ofrecer tarifas para ese intervalo.

b)

No es necesaria la fase de reestructuración porque no hay atributos multivalorados, generalizaciones ni mezclas o divisiones de entidades o relaciones debidas a operaciones.

Entidades fuertes:

- Compañías(CIF, Nombre)
- Clientes(NIF, Nombre, Dirección)

Entidades débiles:

- Tarifas(Fecha inicial, CIF, Precio, Fecha final)



- Apuntes(Fecha, NIF, Consumo)

*Relaciones:*

- Aplica(Fecha inicial, CIF). Se puede omitir porque muestra la misma información que Tarifas.
- Consume(Fecha, NIF). Se puede omitir porque muestra la misma información que Apuntes.
- Elige(NIF, CIF, Fecha inicial, Fecha final). Hay que ampliar la clave con Fecha inicial para admitir que el mismo cliente pueda elegir la misma compañía en diferentes momentos.

*Restricciones de integridad:*

- Cardinalidad: Expresadas como claves.
- No expresables en el esquema conceptual:
  - Un cliente no puede elegir diferentes compañías en el mismo momento: Se añade la dependencia funcional {Fecha, NIF} → {CIF} a la tabla Elige.
  - Si un cliente elige una compañía en un intervalo de tiempo, la compañía debe ofrecer tarifas para ese intervalo.

*Restricciones de integridad referencial:*

Entidades débiles:

Con respecto a Tarifas:

$$\Pi_{CIF}(Tarifas) \subseteq \Pi_{CIF}(Compañías)$$

Con respecto a Apuntes:

$$\Pi_{NIF}(Apuntes) \subseteq \Pi_{NIF}(Clientes)$$

*Relaciones:*

Con respecto a Aplica:

$$\Pi_{Fecha\ inicial, CIF}(Aplica) \subseteq \Pi_{Fecha\ inicial, CIF}(Tarifas)$$

$$\Pi_{CIF}(Aplica) \subseteq \Pi_{CIF}(Clientes)$$

Con respecto a Consume:

$$\Pi_{Fecha, NIF}(Consume) \subseteq \Pi_{Fecha, NIF}(Apuntes)$$

$$\Pi_{NIF}(Consume) \subseteq \Pi_{NIF}(Compañías)$$

Con respecto a Elige:

$$\Pi_{NIF}(Elige) \subseteq \Pi_{NIF}(Compañías)$$

$$\Pi_{CIF}(Elige) \subseteq \Pi_{CIF}(Clientes)$$

*Restricciones de participación total:*

Con respecto a Aplica:

$$\Pi_{Fecha\ inicial, CIF}(Aplica) = \Pi_{Fecha\ inicial, CIF}(Tarifas)$$

Con respecto a Consume:

$$\Pi_{Fecha, NIF}(Consume) = \Pi_{Fecha, NIF}(Apuntes)$$

Con respecto a Elige:

$$\Pi_{CIF}(Elige) = \Pi_{CIF}(Clientes)$$

c)

```
CREATE VIEW Factura AS
PARAMETERS NombreCliente, Mes
SELECT Apuntes.Consumo, Apuntes.Fecha, Apuntes.Consumo * Tarifas.Precio AS Importe,
Compañías.Nombre, Tarifas.Precio
FROM Apuntes, Clientes, Elige, Compañías, Tarifas
WHERE Apuntes.CIF=Elige.CIF AND
Elige.CIF=Clientes.CIF AND
Elige.NIF=Tarifas.NIF AND
Elige.NIF=Compañías.NIF AND
Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final AND
Clientes.Nombre = NombreCliente AND
MONTH(Fecha) = Mes;
```

**Nota:** Los apartados siguientes no se han incluido en el examen.



d)

Dada la consulta anterior, plantear otra que indique los mismos resultados pero que agregue otra fila en la que se indique en la columna del importe el importe total de la factura.

En el modelo conceptual se añade un atributo a la entidad débil Apuntes. El modelo lógico sólo se modifica en la tabla Apuntes, que pasa a tener el siguiente esquema:

Apuntes(Fecha, NIF, Consumo, Importe)

Facturas

UNION

SELECT NULL, NULL, SUM(Importe), NULL, NULL FROM Facturas;

e)

Expresar explícitamente el importe de cada apunte en los esquemas conceptual y lógico. Escribir un aserto que asegure la consistencia del valor del importe con respecto a la tarifa aplicada, entendiendo que la tarifa se refiere al precio por unidad de consumo. El cálculo del importe se debe expresar como una consulta SQL.

El aserto sería:

```
CREATE ASSERTION ComprobarImporte CHECK
```

```
Apuntes.Importe =
```

```
SELECT Tarifas.Precio * Apuntes.Consumo
```

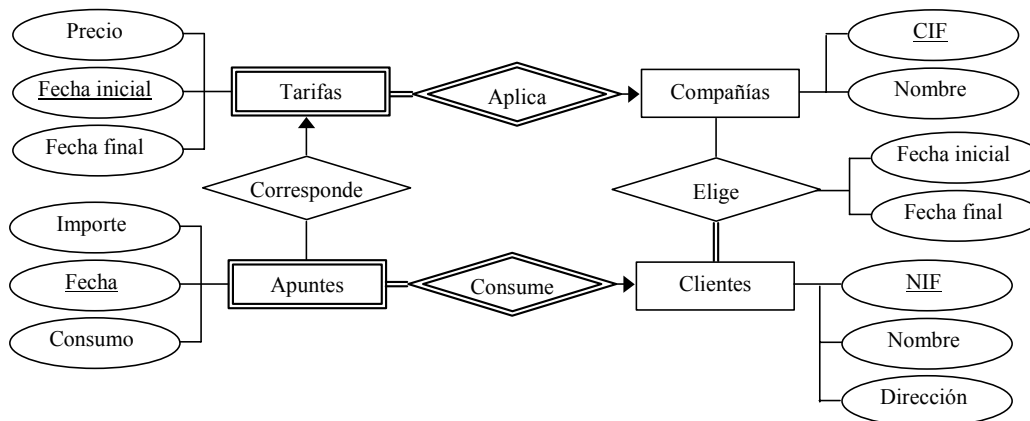
```
FROM Apuntes, Elige, Tarifas
```

```
WHERE Apuntes.CIF=Elige.CIF AND
```

```
Elige.NIF=Tarifas.NIF AND
```

```
Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final;
```

Se podría pensar en simplificar el cálculo del importe añadiendo la relación *Corresponde*, como se muestra a continuación:



La traducción al modelo relacional daría una nueva tabla:

Corresponde(CIF, Fecha, NIF, Fecha inicial)

El aserto quedaría:

```
CREATE ASSERTION ComprobarImporte CHECK
```

```
Apuntes.Importe =
```

```
SELECT Tarifas.Precio * Apuntes.Consumo
```

```
FROM Tarifas, Apuntes, Corresponde
```

```
WHERE Corresponde.CIF=Apuntes.CIF AND
```

```
Corresponde.Fecha=Apuntes.Fecha AND
```

```
Corresponde.NIF=Tarifas.NIF AND
```

```
Corresponde.Fecha inicial=Tarifas.Fecha inicial AND
```

```
Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final;
```

Sin embargo, vemos que se sigue haciendo referencia a tres tablas y la condición se ha complicado.



### I PARCIAL

- 1) (1,25 puntos) ¿Qué características añade el modelo de datos relacional orientado a objetos con respecto al modelo relacional?

Solución (10 min):

Son características derivadas de la POO, como:

- La posibilidad de trabajar con tipos de datos más elaborados; en concreto, con tipos de datos estructurados.
- Encapsula los tipos de datos con sus operaciones mediante métodos y propiedades.
- Herencia de tipos y de tablas.

- 2) (1,25 puntos) Dados los esquemas  $a(u)$ ,  $b(v)$ ,  $c(u)$ ,  $d(u)$ , traducir a Datalog la siguiente consulta SQL:

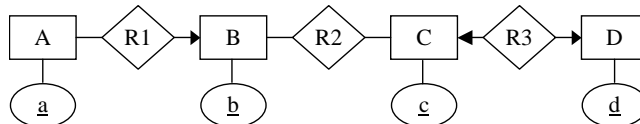
```
SELECT u, v
FROM c, d
WHERE c.u = d.u
UNION
SELECT u, v
FROM a, b
WHERE NOT (u IN SELECT u FROM c);
```

Solución (10 min):

```
r(U,V) :- c(U), d(U).
r(U,V) :- a(U), b(V), not c(U).
```

- 3) (1,9 puntos) Dado el siguiente esquema entidad-relación se pide:

- Traducirlo a una única relación del modelo relacional y expresar las restricciones de cardinalidad con dependencias funcionales.
- Determinar las claves candidatas de la relación obtenida.



Solución (15 min):

a)

Relación:

$r(a,b,c,d)$

Dependencias funcionales:

$a \rightarrow b$

$c \rightarrow d$

$d \rightarrow c$

b)

Las claves candidatas deben contener al atributo a porque éste no se encuentra en ningún consecuente de las dependencias funcionales:

$\{a\}^+ = \{a,b\}$  Sólo con a es insuficiente, hay que agregar c o d:

$\{ac\}^+ = \{a,b,c,d\}$  Clave candidata

$\{ad\}^+ = \{a,b,c,d\}$  Clave candidata

- 4) (5,6 puntos) Se ha encargado a una empresa de servicios informáticos que realice la facturación de los consumos que se producen en una red eléctrica compartida por diferentes compañías. Se desea tener información sobre los clientes (con su NIF, nombre y Dirección) que consumen la electricidad, las compañías (con su CIF y nombre) y los apuntes que representan el consumo (en kW/h) de un cliente en una fecha en concreto. Los clientes eligen la compañía que les presta el servicio entre dos fechas (pueden cambiar en cualquier momento de compañía). Las compañías aplican tarifas válidas entre dos fechas que indican el coste por kW/h. Una compañía sólo puede aplicar una tarifa cada día, independientemente del cliente. Un cliente se entiende como tal si ha elegido una compañía. Se pide:

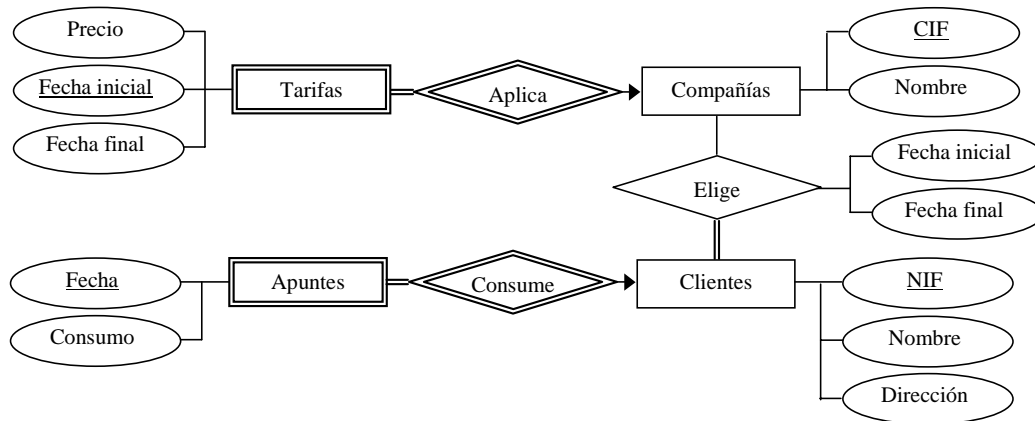
- Usar el modelo entidad-relación para construir el esquema conceptual sin redundancias y sin considerar operaciones. No se deben introducir atributos artificiales que actúen como identificadores.

Explicar las claves, las restricciones de cardinalidad una a una, una a varias o varias a varias, y las que no se puedan expresar que aparezcan en el esquema propuesto.

- Usar el esquema anterior para construir el esquema lógico usando el modelo relacional y explicando todas las restricciones que se impongan. Indicar las tablas que se podrían omitir por contener información redundante. Indicar todas las restricciones de integridad (incluidas las dependencias funcionales, si surgen) y expresar las de integridad referencial y de participación total en notación algebraica.
- Plantear una consulta SQL que, dado el nombre de un cliente y un mes, permita emitir la factura de este cliente en el mes indicado. El resultado de esta consulta debe listar: el consumo en kW/h, la fecha en la que se ha realizado este consumo, su importe, la compañía que ha facturado y la tarifa que ha aplicado.

Solución (45 min):

a)



Claves:

- Entidades fuertes:
  - Compañías: El CIF identifica unívocamente a una compañía.
  - Clientes: El NIF identifica unívocamente a un cliente.
- Entidades débiles:
  - Tarifas: Una tarifa está identificada unívocamente por la fecha en que se comienza a aplicar y la compañía a la que corresponde (identificada unívocamente por su CIF). No es necesario incluir Fecha final en la clave porque nunca se da el caso de que la misma compañía aplique dos tarifas diferentes en la misma fecha.
  - Apuntes: Un apunte está identificado unívocamente por la fecha en que se produce el apunte y el cliente al que corresponde.

Restricciones de cardinalidad:

- Tarifas – Aplica -> Compañías: Dada una tarifa, ésta corresponde a una única compañía (También se podría asumir que una tarifa pueda corresponder a varias compañías). Dada una compañía, en general le corresponden varias tarifas.
- Apuntes – Consume -> Clientes: Dado un apunte, éste corresponde a un solo cliente. Dado un cliente, en general le corresponden varios apuntes.
- Clientes – Elige – Compañías: Un cliente puede elegir varias compañías (en diferentes momentos). Una compañía puede ser elegida por varios clientes.

Restricciones de participación total:

- Un cliente se entiende como tal si ha elegido una compañía.

Restricciones no expresables en este esquema:

- Un cliente no puede elegir diferentes compañías en el mismo momento.
- Si un cliente elige una compañía en un intervalo de tiempo, la compañía debe ofrecer tarifas para ese intervalo.

b)

No es necesaria la fase de reestructuración porque no hay atributos multivalorados, generalizaciones ni mezclas o divisiones de entidades o relaciones debidas a operaciones.

Entidades fuertes:

- Compañías(CIF, Nombre)
- Clientes(NIF, Nombre, Dirección)

Entidades débiles:

- Tarifas(Fecha inicial, CIF, Precio, Fecha final)



- Apuntes(Fecha, NIF, Consumo)

*Relaciones:*

- Aplica(Fecha inicial, CIF). Se puede omitir porque muestra la misma información que Tarifas.
- Consume(Fecha, NIF). Se puede omitir porque muestra la misma información que Apuntes.
- Elige(NIF, CIF, Fecha inicial, Fecha final). Hay que ampliar la clave con Fecha inicial para admitir que el mismo cliente pueda elegir la misma compañía en diferentes momentos.

*Restricciones de integridad:*

- Cardinalidad: Expresadas como claves.
- No expresables en el esquema conceptual:
  - Un cliente no puede elegir diferentes compañías en el mismo momento: Se añade la dependencia funcional {Fecha, NIF} → {CIF} a la tabla Elige.
  - Si un cliente elige una compañía en un intervalo de tiempo, la compañía debe ofrecer tarifas para ese intervalo.

*Restricciones de integridad referencial:*

Entidades débiles:

Con respecto a Tarifas:

$$\Pi_{CIF}(Tarifas) \subseteq \Pi_{CIF}(Compañías)$$

Con respecto a Apuntes:

$$\Pi_{NIF}(Apuntes) \subseteq \Pi_{NIF}(Clientes)$$

Relaciones:

Con respecto a Aplica:

$$\Pi_{Fecha\ inicial, CIF}(Aplica) \subseteq \Pi_{Fecha\ inicial, CIF}(Tarifas)$$

$$\Pi_{CIF}(Aplica) \subseteq \Pi_{CIF}(Clientes)$$

Con respecto a Consume:

$$\Pi_{Fecha, NIF}(Consume) \subseteq \Pi_{Fecha, NIF}(Apuntes)$$

$$\Pi_{NIF}(Consume) \subseteq \Pi_{NIF}(Compañías)$$

Con respecto a Elige:

$$\Pi_{NIF}(Elige) \subseteq \Pi_{NIF}(Compañías)$$

$$\Pi_{CIF}(Elige) \subseteq \Pi_{CIF}(Clientes)$$

*Restricciones de participación total:*

Con respecto a Aplica:

$$\Pi_{Fecha\ inicial, CIF}(Aplica) = \Pi_{Fecha\ inicial, CIF}(Tarifas)$$

Con respecto a Consume:

$$\Pi_{Fecha, NIF}(Consume) = \Pi_{Fecha, NIF}(Apuntes)$$

Con respecto a Elige:

$$\Pi_{CIF}(Elige) = \Pi_{CIF}(Clientes)$$

c)

CREATE VIEW Factura AS

PARAMETERS NombreCliente, Mes

SELECT Apuntes.Consumo, Apuntes.Fecha, Apuntes.Consumo \* Tarifas.Precio AS Importe, Compañías.Nombre, Tarifas.Precio

FROM Apuntes, Clientes, Elige, Compañías, Tarifas

WHERE Apuntes.CIF=Elige.CIF AND

Elige.CIF=Clientes.CIF AND

Elige.NIF=Tarifas.NIF AND

Elige.NIF=Compañías.NIF AND

Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final AND

Clientes.Nombre = NombreCliente AND

MONTH(Fecha) = Mes;

**Nota:** Los apartados siguientes no se han incluido en el examen.



d)

Dada la consulta anterior, plantear otra que indique los mismos resultados pero que agregue otra fila en la que se indique en la columna del importe el importe total de la factura.

En el modelo conceptual se añade un atributo a la entidad débil Apuntes. El modelo lógico sólo se modifica en la tabla Apuntes, que pasa a tener el siguiente esquema:

Apuntes(Fecha, NIF, Consumo, Importe)

Facturas

UNION

SELECT NULL, NULL, SUM(Importe), NULL, NULL FROM Facturas;

e)

Expresar explícitamente el importe de cada apunte en los esquemas conceptual y lógico. Escribir un aserto que asegure la consistencia del valor del importe con respecto a la tarifa aplicada, entendiendo que la tarifa se refiere al precio por unidad de consumo. El cálculo del importe se debe expresar como una consulta SQL.

El aserto sería:

```
CREATE ASSERTION ComprobarImporte CHECK
```

```
Apuntes.Importe =
```

```
SELECT Tarifas.Precio * Apuntes.Consumo
```

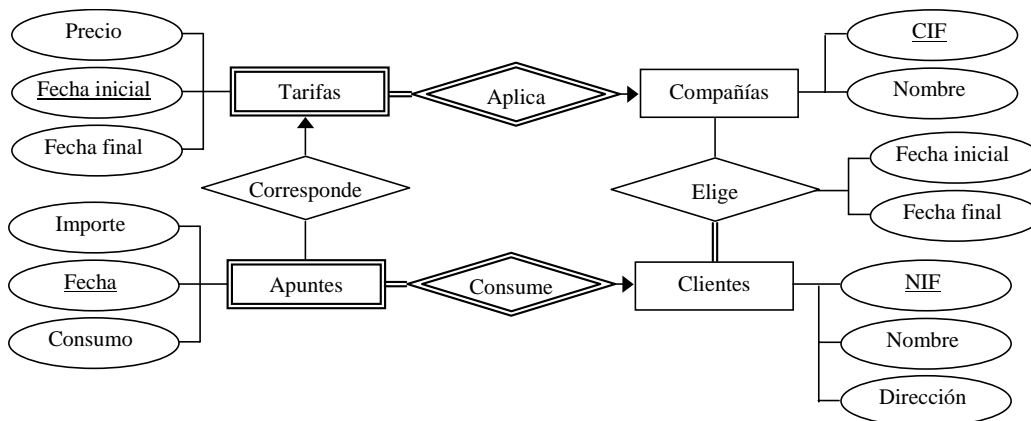
```
FROM Apuntes, Elige, Tarifas
```

```
WHERE Apuntes.CIF=Elige.CIF AND
```

```
Elige.NIF=Tarifas.NIF AND
```

```
Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final;
```

Se podría pensar en simplificar el cálculo del importe añadiendo la relación *Corresponde*, como se muestra a continuación:



La traducción al modelo relacional daría una nueva tabla:

Corresponde(CIF, Fecha, NIF, Fecha inicial)

El aserto quedaría:

```
CREATE ASSERTION ComprobarImporte CHECK
```

```
Apuntes.Importe =
```

```
SELECT Tarifas.Precio * Apuntes.Consumo
```

```
FROM Tarifas, Apuntes, Corresponde
```

```
WHERE Corresponde.CIF=Apuntes.CIF AND
```

```
Corresponde.Fecha=Apuntes.Fecha AND
```

```
Corresponde.NIF=Tarifas.NIF AND
```

```
Corresponde.Fecha inicial=Tarifas.Fecha inicial AND
```

```
Apuntes.Fecha BETWEEN Tarifas.Fecha inicial AND Tarifas.Fecha final;
```

Sin embargo, vemos que se sigue haciendo referencia a tres tablas y la condición se ha complicado.





## II PARCIAL

1) (3 puntos) Para el siguiente conjunto de valores clave:

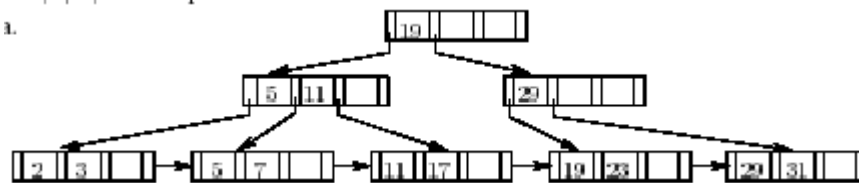
(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

Se pide:

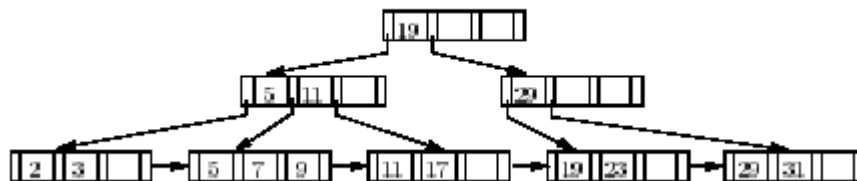
- Construir un árbol B+ con cuatro punteros por nodo.
- Insertar 9.
- Insertar 10.
- Insertar 8.
- Borrar 23.

Solución (25 min):

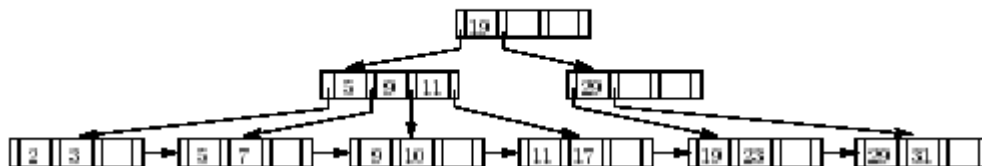
a.



Insert 9:



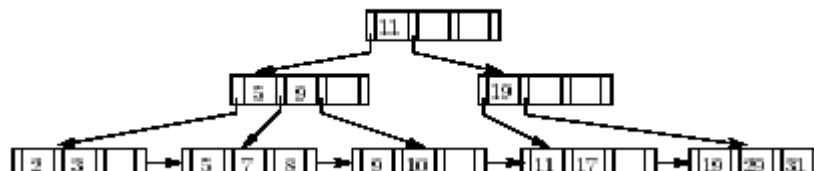
Insert 10:



Insert 8:



Delete 23:



2) (1,15 puntos) Dada la siguiente consulta SQL:

```
SELECT T.Nombre
FROM Sucursal T, Sucursal S
WHERE T.Activo > S.Activo AND S.Ciudad = "Arganzuela";
```



Escríbase una expresión equivalente del álgebra relacional que sea más eficiente, considerando que el número de filas de la tabla Sucursal es mucho mayor que el de columnas y que CS(Ciudad, Sucursal) es un número pequeño con distribución uniforme de valores. Justifíquese su elección.

Solución (10 min):

$$\prod_{T.Nombre} \left( \left( \prod_{Nombre.Activo} (\rho_T(Sucursal)) \right) \times \prod_{T.Activo > S.Activo} \left( \prod_{Activo} (\sigma_{(Ciudad='Arganzuela')}( \rho_S(Sucursal) )) \right) \right)$$

Esta expresión realiza la combinación con la menor cantidad posible de datos. Para ello, en el lado derecho de la reunión se seleccionan sólo las filas que contienen el valor Arganzuela (pocas, debido a la baja cardinalidad de la selección) y de ambas ramas elimina todos los atributos innecesarios.

- 3) (3,5 puntos) Dado un fichero secuencial de registros de dos campos A y B ordenado por el campo A, y los triples (0,8,1), (3,2,2), (7,8,3), (11,2,4), (23,4,5), (25,6,6), (27,2,7), (35,8,8) que indican (Valor del campo A, Valor del campo B, Dirección de bloque) para un campo clave A, y un factor de bloqueo 1 para los ficheros de índices y de 0,05 para el fichero de datos, se pide:
- Crear un índice secundario denso de lista invertida sobre el campo B.

Solución (30 min):

| Valores de B | Nº de registros | Direcciones |   |   |
|--------------|-----------------|-------------|---|---|
| 2            | 3               | 2           | 4 | 7 |
| 4            | 1               | 5           |   |   |
| 6            | 1               | 6           |   |   |
| 8            | 3               | 1           | 3 | 8 |

- Crear un índice primario escaso en el que aparezcan los valores de la clave tales que no hay ningún valor de A menor que los valores de la decena a la que pertenece (es decir, menor que  $\lceil A \text{ mod } 10 \rceil * 10$  en el intervalo  $[(A \text{ div } 10) * 10, \lceil (A+10) \text{ div } 10 \rceil * 10$ )). Por ejemplo, en la decena 20 de los datos del problema sólo aparecería en el índice el valor 23.

(0,1) -> (11,4) -> (23,5) -> (35,8)

- Calcular el coste de la consulta “Determinar el número de valores diferentes del campo indexado” usando los índices de los casos (a) y (b), con 10.000 registros, factor de repetición de valores en media de 10 y valores comprendidos entre 1 y 1.000.

Con 10.000 registros y 10 registros por valor hay  $10.000/10=1.000$  valores diferentes de la clave (sea A o B, según el apartado).

- Lista invertida sobre el campo B:

Sólo hay que acceder a la lista invertida para hacer el recuento de registros de esta lista, que proporciona el número de valores de registros. Por lo tanto, el coste es 1.000 asumiendo una operación de lectura de bloque por cada registro de la lista invertida.

- Índice primario sobre el campo A:

A diferencia de la lista invertida, en este caso el índice es escaso y no nos puede aportar esta información. La única forma es explorar secuencialmente el fichero de datos. Por lo tanto, el coste es  $10.000/0,05=200.000$ .

- Calcular el coste de insertar un registro cuyo valor de la clave no aparezca en el índice en los casos (a) y (b) con los mismos supuestos del apartado anterior.

- Lista invertida:

En media, hay que acceder a 500 bloques para localizar la entrada con el menor valor mayor o igual que el valor de A del registro que se va a insertar. Después hay que solicitar un bloque y encajarlo en la lista enlazada. Por tanto, el coste es 501 lecturas (para localizar el lugar en el que insertar), 3 escrituras en el índice (bloques anterior y posterior para enlazar adecuadamente y bloque nuevo) y 20 ( $1/0,05=20$  bloques por registro) escrituras en el fichero de datos.

Total=501 lecturas + 23 escrituras

- Índice primario:



Hay que recorrer el índice para localizar la entrada con el menor valor mayor o igual que el valor de A del registro que se va a insertar. Habrá  $1.000/10=100$  entradas en el índice, por lo que habrá que recorrer 50 registros en media. Por tanto, el coste es de 50 lecturas (para localizar el lugar en el que insertar), 1 escritura (considerando el peor caso en que se modifica el valor del índice) y 20 escrituras en el fichero de datos.

Total=50 lecturas + 21 escrituras

**Nota:** Los costes se piden en términos de accesos a bloque en lectura y escritura. Asumir asignación enlazada en los índices.

- 4) (2,35 puntos) Dadas las tablas Seno(Ángulo,Valor), Coseno(Ángulo,Valor) y Ángulos(Ángulo), se pide:
- Crear una tabla Operaciones(Ángulo,Seno,Coseno) que relacione cada ángulo de la tabla Ángulos con los valores de su seno y su coseno.
  - Crear un procedimiento PL/SQL que, a partir de la tabla Operaciones, genere las filas de otra tabla Tangentes(Ángulo,Valor) (ya creada) que incluya los valores de la tangente de cada ángulo calculados como el cociente entre su seno y coseno almacenados en la tabla Operaciones. El procedimiento debe implementar una transacción con un bucle que recorra todas las filas de Operaciones en el que, en primer lugar, se consulte y borre una fila de esta tabla, y, en segundo lugar, se inserte una nueva fila con el resultado de la tangente en Tangentes. Si hay una división por cero se debe deshacer la transacción sólo para esa fila.

Solución (20 min):

```
a)
CREATE TABLE Ángulos(Ángulo NUMBER(6,3));
CREATE TABLE Seno(Ángulo NUMBER(6,3), Valor NUMBER(7,6));
CREATE TABLE Coseno(Ángulo NUMBER(6,3), Valor NUMBER(7,6));
```

```
DECLARE
 i INTEGER;
BEGIN
 FOR i IN 0..359 LOOP
 INSERT INTO Seno VALUES (i, SIN(i*3.1416/180));
 INSERT INTO Coseno VALUES (i, COS(i*3.1416/180));
 END LOOP;
END;
/
```

```
DECLARE
 i INTEGER;
BEGIN
 FOR i IN 85..95 LOOP
 INSERT INTO Ángulos VALUES (i);
 END LOOP;
END;
/
```

```
SELECT Ángulo, Seno.Valor, Coseno.Valor
INTO Operaciones
FROM (Ángulos NATURAL INNER JOIN Seno) INNER JOIN Coseno USING (Ángulo);
```

```
b)
CREATE OR REPLACE PROCEDURE Txn_Tangente () AS
 CURSOR C_Operaciones IS SELECT Ángulo, Seno, Coseno FROM Operaciones;
 V_Ángulo Operaciones.Ángulo%TYPE;
 V_Seno Operaciones.Seno%TYPE;
 V_Coseno Operaciones.Coseno%TYPE;
 V_Tangente NUMBER;
BEGIN
```



```
OPEN C_Operaciones;
LOOP
 FETCH C_Operaciones INTO V_Ángulo, V_Seno, V_Coseno;
 EXIT WHEN C_Operaciones%NOT_FOUND;
 DELETE FROM Operaciones WHERE Ángulo = V_Ángulo;
 V_Tangente = V_Seno / V_Coseno;
 INSERT INTO Tangentes VALUES (V_Ángulo, V_Tangente);
 COMMIT;
END LOOP;
CLOSE C_Operaciones;
EXCEPTION
 WHEN ZERO_DIVIDE THEN
 ROLLBACK;
END;
```