

PAUTA DE CORRECCION

CERTAMEN #2

ICI-343

Wenceslao Palma <wenceslao.palma@ucv.cl>

1. Responda cada una de las siguientes preguntas.

(a) (8 pts) Qué es una clave foránea? Cómo se implementa en PostgreSQL? Ejemplifique.

R.: Una clave foránea es una clave heredada desde una entidad fuerte que permite asegurar la integridad referencial. En PostgreSQL se implementa usando REFERENCES al momento de definir el atributo que se hereda como clave desde una entidad fuerte. Por ejemplo, supongamos que en una empresa un ingeniero puede ser jefe de muchos proyectos y un proyecto puede tener sólo un jefe. En PostgreSQL se define de la sgte manera:

```
create table ingeniero ( clave char(9) PRIMARY KEY,
.....
);
create table proyecto ( jefe char(9) REFERENCES ingeniero(clave),
.....
);
```

(b) (7 pts) Explique cómo se implementa una relación n:m en PostgreSQL.

R.: Una relación n:m no es implementable en el modelo relacional, para resolver dicho problema se debe crear una entidad de asociación la cual debe contener como claves foráneas las claves de las dos entidades que tienen una relación con cardinalidad n:m. Por ejemplo para el caso de “alumnos que inscriben asignaturas” tenemos la entidad de asociación INSCRIPCION la cual se define en PostgreSQL de la sgte manera:

```
create table inscripcion(clave char(8) REFERENCES asignatura,
                        rol char(8) REFERENCES alumno,
                        .....
);
```

(c) (7 pts) Cuando un índice tipo grid puede tener una gran cantidad de buckets vacíos?

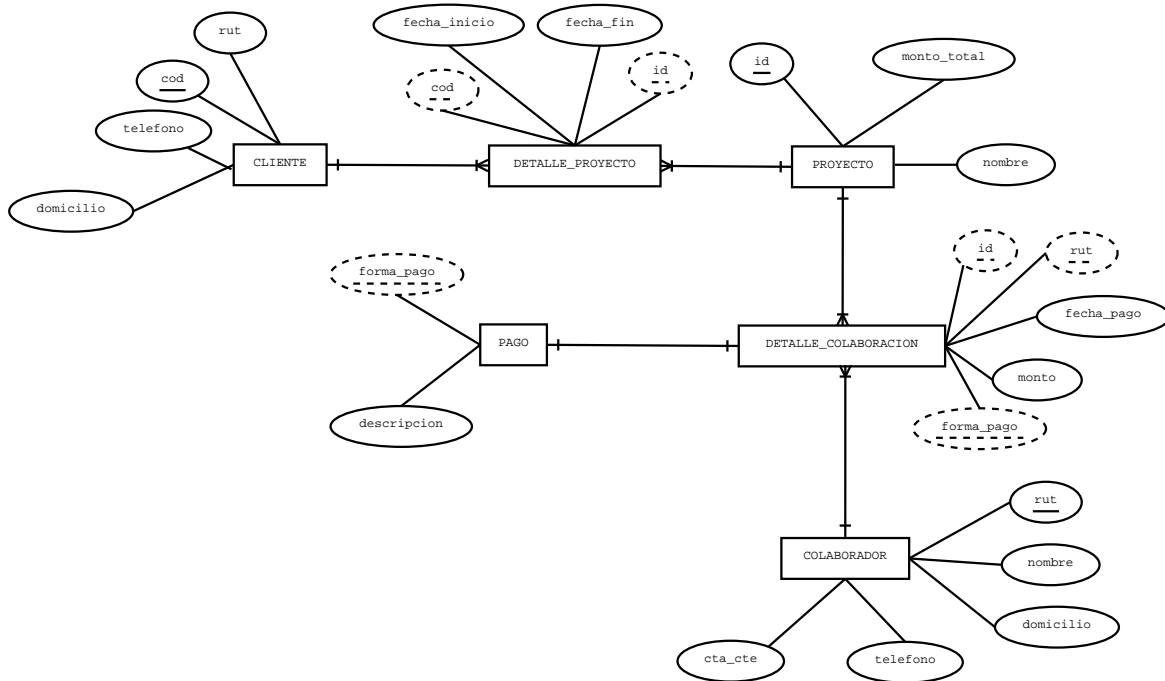
R.: cuando existe una correlación entre los datos a indexar. Considerando un índice basado en dos dimensiones, la correlación entre los valores de los datos a indexar puede provocar que una gran cantidad de éstos se almacenen en buckets correspondientes a la diagonal.

(d) (8 pts) Es posible dar respuesta a una consulta basada en rangos sobre dos atributos x e y usando índices basados en árboles B+? Ejemplifique.

R.: Sí, cuando la consulta requiere recuperar registros que cumplan con $a \leq x \leq b$ y $c \leq y \leq d$ se puede usar un árbol B+ para indexar sobre el atributo x y otro árbol B+ para el atributo y . Luego se recorre el primer árbol y se almacenan todos los punteros correspondiente a registros que cumplen con $a \leq x \leq b$, posteriormente se recorre el segundo árbol para almacenar todos los punteros correspondiente a registros que cumplen con $c \leq y \leq d$. Finalmente se realiza una intersección a nivel de punteros y se recuperan los registros desde el disco.

2. Una empresa consultora desea almacenar la información de los proyectos que realiza. Para cada uno de los proyectos realizados interesa almacenar el código, descripción, costo del proyecto, fecha de inicio y fecha de fin. Los proyectos son realizados a clientes de los que se desea guardar el código, teléfono, domicilio y RUT. Un cliente puede registrar varios proyectos, pero un proyecto es realizado a un único cliente. En los proyectos participan colaboradores de los que se dispone la siguiente información: RUT, nombre, domicilio, teléfono, banco y número de cuenta. Un colaborador puede participar en varios proyectos. Los proyectos son realizados por uno o más colaboradores. Los colaboradores de los proyectos reciben pagos. De los pagos realizados se quiere guardar el número de pago, concepto, cantidad y fecha de pago. También interesa almacenar los diferentes tipos de pagos que puede realizar la empresa. De cada uno de los tipos de pagos se desea guardar el código y descripción.

(a) (8 pts) Construya el diagrama ER que modele la presente situación.



(b) (8 pts) Tome lo propuesto en (a) y escriba las sentencias para la creación del Modelo Relacional usando PostgreSQL.

R.:

```

create table cliente(
    cod serial PRIMARY KEY,
    rut varchar(12),
    domicilio varchar(30),
    telefono varchar(12)
);

create table proyecto(
    id varchar(5) PRIMARY KEY,
    monto_total integer,
    nombre varchar(20)
);

create table detalle_proyecto(
    cod serial REFERENCES cliente,
    id varchar(5) REFERENCES proyecto,
    fecha_inicio date,
    fecha_fin date
);

create table detalle_colaboracion(
    id serial PRIMARY KEY,
    rut varchar(12),
    fecha_pago date,
    monto integer,
    forma_pago varchar(20)
);

create table colaborador(
    rut varchar(12),
    nombre varchar(20),
    domicilio varchar(30),
    cta_cte varchar(12),
    telefono varchar(12)
);

create table pago(
    descripcion varchar(30),
    forma_pago varchar(20)
);

```

```

    fecha_inicio DATE,
    fecha_fin DATE
);

create table colaborador(
    rut varchar(12) PRIMARY KEY,
    nombre varchar(40),
    domicilio varchar(30),
    telefono varchar(12),
    cta_cta varchar(10)

);

create table detalle_colaboracion(
    id varchar(5) REFERENCES proyecto,
    rut varchar(12) REFERENCES colaborador,
    forma_pago varchar(10) REFERENCES pago,
    monto integer,
);

create table pago(
    forma_pago varchar(10) PRIMARY KEY,
    descripcion varchar(20)
);

```

(c) Escriba las sentencias SQL para responder las sgtes consultas:

– (7 ptos) Monto de lo pagado a un determinado colaborador durante los últimos 6 meses.

R.:

```

SELECT sum(monto)
FROM detalle_colaboracion
WHERE rut='xx.xxx.xxx-x' AND
      fecha_pago BETWEEN '2015-30-6' AND '2015-01-1'

```

– (7 ptos) Los proyectos que ha realizado la empresa a un determinado cliente durante el presente año.

R.:

```

SELECT proyecto.nombre, proyecto.monto_total
FROM detalle_proyecto,proyecto
WHERE detalle_proyecto.cod = xxxx AND
      detalle_proyecto.id = proyecto.id

```