

TAREA BASES DE DATOS II

PRÁCTICA DE CONSULTAS SQL: UNIONES ENTRE RELACIONES

Instituto Tecnológico de Costa Rica

IC4302 Bases de Datos II - GR 20

Prof. Alberto Shum Chan

Integrantes:

Jonathan Quesada Salas

Alberto Zumbado Abarca



Escuela de Ingeniería en Computación
Bachillerato en Ingeniería en Computación
Sede Interuniversitaria de Alajuela

Prof. Alberto Shum Chan
Base de Datos II
Semestre II, 2021

Práctica de consultas SQL: uniones entre relaciones

Objetivo: poner en práctica el conocimiento del concepto de JOIN de tablas utilizando SQL estándar para generar informes a partir de datos en una base de datos.

Entrega en el TecDigital:

- Deben presentar las sentencias SQL de creación de tablas, los inserts y consultas.
- Forma de trabajo: en grupo máximo de dos personas
- Pueden utilizar el RDBMS de su elección (ej. <https://livesql.oracle.com>).

1. Sean las relaciones de un sistema simplificado de manejo de datos de hospitales, médicos y especialidades que se muestran en la figura #1.

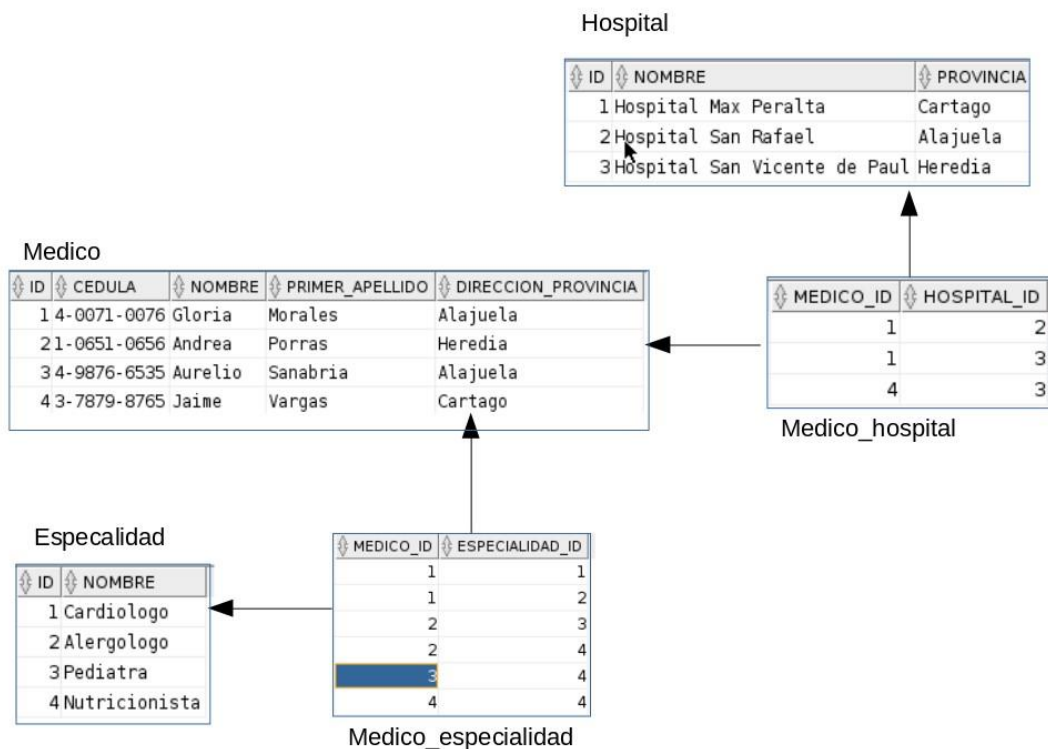


Figura 1. Modelo de datos simplificado de un sistema de manejo de información de hospitales, médicos y especialidades. Las flechas entre dos relaciones representan la restricción llave foránea.

Escriba las siguientes consultas en SQL estándar.

- 1.1. Liste los datos de todos los hospitales que cuentan con al menos un cardiólogo asociado. La siguiente figura (figura 2) muestra el resultado de su consulta SQL, utilizando los datos de la figura 1.



Query Result x

SQL | All Rows Fetched: 2 in 0.149 seconds

	NOMBRE	CEDULA	NOMBRE_1	PRIMER_APELLIDO	DIRECCION_PROVINCIA
1	Hospital San Rafael	4-0071-0076	Gloria	Morales	Alajuela
2	Hospital San Vicente de Paul	4-0071-0076	Gloria	Morales	Alajuela

Figura 2

- 1.2. Liste todos los médicos con todos sus datos disponibles (si no hay datos para alguno, las columnas correspondientes deben quedar vacías). El informe debe incluir las siguientes columnas:

De la relación Medico: cedula, nombre, primer_apellido, direccion_provincia.

De la relación hospital: nombre (ie. los nombres de los hospitales a los que el médico está asociado)

De la relación especialidad: nombre (ie. los nombres de las especialidades médicas adquiridas por el médico)

La figura 3 muestra el resultado de su consulta utilizando los datos de la figura 1.



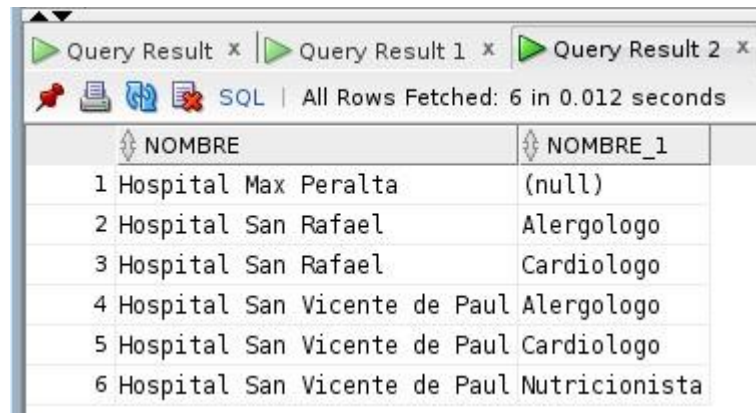
Query Result x Query Result 1 x Query Result 2 x

SQL | All Rows Fetched: 8 in 0.004 seconds

	CEDULA	NOMBRE	PRIMER_APELLIDO	DIRECCION_PROVINCIA	NOMBRE_1	NOMBRE_2
1	1-0651-0656	Andrea	Porras	Heredia	(null)	Nutricionista
2	1-0651-0656	Andrea	Porras	Heredia	(null)	Pediatra
3	3-7879-8765	Jaime	Vargas	Cartago	Hospital San Vicente de Paul	Nutricionista
4	4-0071-0076	Gloria	Morales	Alajuela	Hospital San Rafael	Alergologo
5	4-0071-0076	Gloria	Morales	Alajuela	Hospital San Rafael	Cardiologo
6	4-0071-0076	Gloria	Morales	Alajuela	Hospital San Vicente de Paul	Alergologo
7	4-0071-0076	Gloria	Morales	Alajuela	Hospital San Vicente de Paul	Cardiologo
8	4-9876-6535	Aurelio	Sanabria	Alajuela	(null)	Nutricionista

Figura 3

- 1.3. Liste todos los hospitales y la especialidad médica con las que cuenta el personal médico asociado. Todos los hospitales deben aparecer en la lista. La figura 4 muestra el resultado de su consulta.



	NOMBRE	NOMBRE_1
1	Hospital Max Peralta	(null)
2	Hospital San Rafael	Alergologo
3	Hospital San Rafael	Cardiologo
4	Hospital San Vicente de Paul	Alergologo
5	Hospital San Vicente de Paul	Cardiologo
6	Hospital San Vicente de Paul	Nutricionista

Figura 4

2. Realice los siguientes ejercicios de URI Online Judge

Ingresa al sitio URI Online Judge y realice los siguientes ejercicios. Al finalizar cada ejercicio debe copiar y pegar la sentencia SQL exitosa. El sitio permite bajar los datos e incluirlos en su base de datos o utilizar el servicio en línea para probar sus respuestas, para esto es aconsejable crear un usuario.

- 2.1. Categories: Cuando los datos de la empresa en la que usted trabaja se migraron a la base de datos, hubo un pequeño malentendido del DBA. Su jefe necesita que seleccione el ID y el nombre de los productos, cuyo nombre de categoría comienza con 'super'. Detalle del ejercicio en: <https://www.urionlinejudge.com.br/judge/es/problems/view/2606>
- 2.2. September rentals. Una tienda de vídeos está haciendo su informe semestral y necesita su ayuda. Todo lo que tiene que hacer es seleccionar el nombre de los clientes y la fecha de alquiler, de los alquileres realizados en septiembre de 2016. Detalle del ejercicio en: <https://www.urionlinejudge.com.br/judge/es/problems/view/2614>
- 2.3. Categories with Various Products. La industria de ventas necesita un informe para saber qué productos quedan en stock. <https://www.urionlinejudge.com.br/judge/es/problems/view/2623>

Parte 1

-- Borrar las tablas por aquello

```
drop table hospital;
```

```
drop table medico_hospital;
```

```
drop table medico;
```

```
drop table especialidad;
```

```
drop table medico_especialidad;
```

-- Creacion de tablas

```
CREATE TABLE hospital (  
    hospital_id NUMBER,  
    nombre    VARCHAR2(40),  
    provincia VARCHAR2(20),  
    PRIMARY KEY ( hospital_id )  
);
```

```
CREATE TABLE medico (  
    medico_id    NUMBER,  
    cedula       VARCHAR2(15),  
    nombre       VARCHAR2(25),  
    primer_apellido VARCHAR2(25),  
    direccion_provincia VARCHAR2(15),  
    PRIMARY KEY ( medico_id )  
);
```

```
CREATE TABLE medico_hospital  
    (medico_id number,  
    hospital_id number,  
    PRIMARY KEY (hospital_id, medico_id)  
);
```

```
CREATE TABLE especialidad (  
    especialidad_id NUMBER,  
    nombre VARCHAR2(20),  
    PRIMARY KEY ( especialidad_id )  
);
```

```
CREATE TABLE medico_especialidad (  
    medico_id    NUMBER,  
    especialidad_id NUMBER,  
    PRIMARY KEY (medico_id, especialidad_id)  
);
```

-- Insertar datos

```
insert into hospital values(1,'Hospital Max Peralta', 'Cartago');
```

```
insert into hospital values(2,'Hospital San Rafael', 'Alajuela');
```

```
insert into hospital values(3,'Hospital San Vicente de Paul', 'Heredia');
```

```
select * from hospital;
```

```
insert into medico_hospital values(1, 2);
```

```
insert into medico_hospital values(1, 3);
insert into medico_hospital values(4, 3);
```

```
insert into medico values(1,'4-0071-0076', 'Gloria', 'Morales', 'Alajuela');
insert into medico values(2,'1-0071-0656', 'Andrea', 'Porrás', 'Heredia');
insert into medico values(3,'4-0071-6535', 'Aurelio', 'Sanabria', 'Alajuela');
insert into medico values(4,'3-7879-8765', 'Jaime', 'Vargas', 'Cartago');
```

```
insert into medico_especialidad values (1,1);
insert into medico_especialidad values (1,2);
insert into medico_especialidad values (2,3);
insert into medico_especialidad values (2,4);
insert into medico_especialidad values (3,4);
insert into medico_especialidad values (4,4);
```

```
insert into especialidad values (1, 'Cardiologo');
insert into especialidad values (2, 'Alergologo');
insert into especialidad values (3, 'Pediatra');
insert into especialidad values (4, 'Nutricionista');
```

-- 1.1 Liste los datos de todos los hospitales que cuentan con al menos un cardiólogo asociado. La siguiente figura (figura 2) muestra el resultado de su consulta SQL, -- utilizando los datos de la figura 1.

```
SELECT
  h.nombre,
  m.cedula,
  m.nombre,
  m.primer_apellido,
  m.direccion_provincia
FROM
  hospital h
  INNER JOIN medico_hospital mh ON h.hospital_id = mh.hospital_id
  INNER JOIN medico m ON m.medico_id = mh.medico_id
  INNER JOIN medico_especialidad me ON m.medico_id = me.medico_id
  INNER JOIN especialidad e ON me.especialidad_id = e.especialidad_id where
e.especialidad_id = 1;
```

-- 1.2 . Liste todos los médicos con todos sus datos disponibles (si no hay datos para alguno, -- las columnas correspondientes deben quedar vacías). El informe debe incluir las -- siguientes columnas:
-- De la relación Medico: cedula, nombre, primer_apellido, direccion_provincia.
-- De la relación hospital: nombre (ie. los nombres de los hospitales a los que el médico -- está asociado)
-- De la relación especialidad: nombre (ie. los nombres de las especialidades médicas -- adquiridas por el médico)

```
SELECT
  m.cedula,
  m.nombre,
  m.primer_apellido,
  m.direccion_provincia,
```

```

    h.nombre,
    e.nombre
FROM
    medico          m
    LEFT JOIN medico_especialidad me ON me.medico_id = m.medico_id
    INNER JOIN especialidad e ON me.especialidad_id = e.especialidad_id
    LEFT JOIN medico_hospital    mh ON m.medico_id = mh.medico_id
    LEFT JOIN hospital          h ON h.hospital_id = mh.hospital_id;

```

-- 1.3. . Liste todos los hospitales y la especialidad médica con las que cuenta el personal

-- médico asociado. Todos los hospitales deben aparecer en la lista. La figura 4 muestra

-- el resultado de su consulta.

-- médico asociado. Todos los hospitales deben aparecer en la lista. La figura 4 muestra

-- el resultado de su consulta.

```

SELECT
    h.nombre,
    e.nombre
FROM
    hospital h
    LEFT JOIN medico_hospital    mh ON h.hospital_id = mh.hospital_id
    LEFT JOIN medico          m ON m.medico_id = mh.medico_id
    LEFT JOIN medico_especialidad me ON m.medico_id = me.medico_id
    LEFT JOIN especialidad      e ON me.especialidad_id = e.especialidad_id;

```

Parte 2:

2.1 Ejercicio Categories

```
SELECT p.id, p.name FROM products p LEFT JOIN categories c ON p.id_categories = c.id  
WHERE c.name LIKE 'super%';
```



The screenshot shows the URI Online Judge interface. The user is logged in as 'Hi, Alberto Zumbado'. The page displays the submission details for problem 2606 - Categories. The submission status is 'Accepted'. The user's solution code is shown in the 'CÓDIGO FUENTE' section.

URI ONLINE JUDGE PROBLEMS & CONTESTS

EN VIVO
Mira los envíos en tiempo real.

LISTAR
Listar todos tus envíos.

CÓDIGO FUENTE
MIRA EL CÓDIGO FUENTE DE TU ENVÍO, HÁS ALGUNOS DETALLES ADICIONALES.

ENVÍO # 23886962

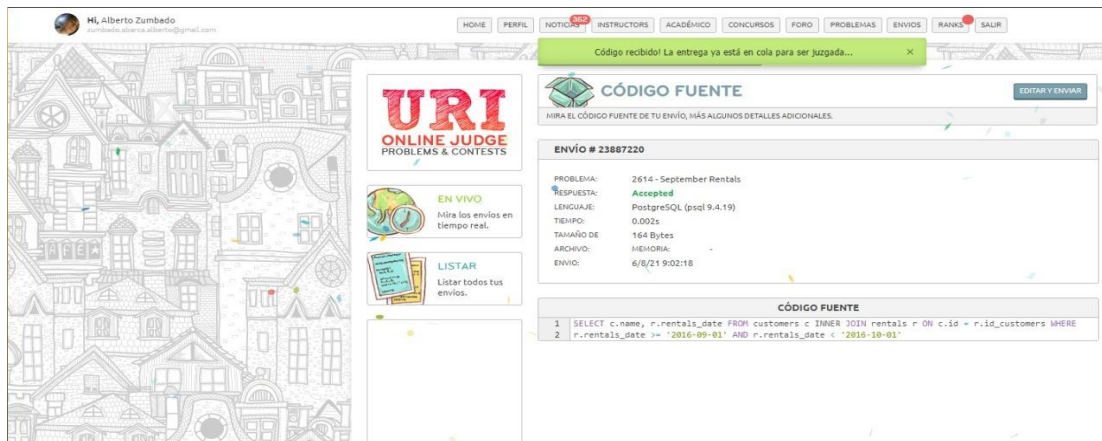
PROBLEMA: 2606 - Categories
RESPUESTA: Accepted
LENGUAJE: PostgreSQL (psql 9.4.19)
TIEMPO: 0.002s
TAMAÑO DE ARCHIVO: 114 Bytes
MEMORIA: -
ENVÍO: 6/8/21 8:40:00

CÓDIGO FUENTE

```
1 SELECT p.id, p.name FROM products p LEFT JOIN categories c ON p.id_categories = c.id WHERE  
2 c.name LIKE 'super%';
```

2.2 Ejercicio Septiembre rentals

```
SELECT c.name, r.rentals_date FROM customers c INNER JOIN rentals r ON c.id =  
r.id_customers WHERE r.rentals_date >= '2016-09-01' AND r.rentals_date < '2016-10-01';
```



The screenshot shows the URI Online Judge interface. The user is logged in as 'Hi, Alberto Zumbado'. The page displays the submission details for problem 2614 - September Rentals. The submission status is 'Accepted'. The user's solution code is shown in the 'CÓDIGO FUENTE' section.

URI ONLINE JUDGE PROBLEMS & CONTESTS

EN VIVO
Mira los envíos en tiempo real.

LISTAR
Listar todos tus envíos.

CÓDIGO FUENTE
MIRA EL CÓDIGO FUENTE DE TU ENVÍO, HÁS ALGUNOS DETALLES ADICIONALES.

ENVÍO # 23887220

PROBLEMA: 2614 - September Rentals
RESPUESTA: Accepted
LENGUAJE: PostgreSQL (psql 9.4.19)
TIEMPO: 0.002s
TAMAÑO DE ARCHIVO: 164 Bytes
MEMORIA: -
ENVÍO: 6/8/21 9:02:18

CÓDIGO FUENTE

```
1 SELECT c.name, r.rentals_date FROM customers c INNER JOIN rentals r ON c.id = r.id_customers WHERE  
2 r.rentals_date >= '2016-09-01' AND r.rentals_date < '2016-10-01';
```

3.1 Ejercicio Categories with Various Products

```
SELECT p.name, c.name FROM products p INNER JOIN categories c ON c.id =  
p.id_categories WHERE p.id_categories IN(1,2,3,6,9) AND p.amount > 100;
```




Hi, Alberto Zumbado
zumbado.abarca.alberto@gmail.com

[HOME](#) [PERFIL](#) [NOTICIAS](#) [INSTRUCTORES](#) [ACADÉMICO](#) [CONCURSOS](#) [FORO](#) [PROBLEMAS](#) [ENVÍOS](#) [RANKS](#) [SALIR](#)

Código recibido! La entrega ya está en cola para ser juzgada...



URI
ONLINE JUDGE
PROBLEMS & CONTESTS



EN VIVO
Mira los envíos en tiempo real.



LISTAR
Listar todos tus envíos.



CÓDIGO FUENTE
MIRA EL CÓDIGO FUENTE DE TU ENVÍO, MÁS ALGUNOS DETALLES ADICIONALES.

ENVÍO # 23887384

PROBLEMA: 2623 - Categories with Various Products

RESPUESTA: **Accepted**

LENGUAJE: PostgreSQL (psql 9.4.19)

TIEMPO: 0.003s

TAMAIÑO DE ARCHIVO: 145 Bytes

MEMORIA: MENOR

ENVÍO: 6/8/21 9:19:28

CÓDIGO FUENTE

```
1 SELECT p.name, c.name FROM products p INNER JOIN categories c ON c.id = p.id_categories WHERE
2 p.id_categories IN(1,2,3,6,9) AND p.amount > 100
```