

Bases de Datos

Clase 11: Recursión

Recursión

Motivación

`Vuelos_Directos(Ciudad_Origen, Ciudad_Destino)`

Ciudad Origen	Ciudad Destino
Santiago	Buenos Aires
Buenos Aires	Lima
Buenos Aires	Doha
Lima	Los Ángeles
...	...

Recursión

Motivación

¿Hasta que ciudad podemos llegar partiendo desde Santiago, sin importar el número de escalas?

Limitaciones de SQL

- ¿Puedo expresar la consulta anterior en SQL?
- ¿Qué cosas **no** son expresables en SQL?

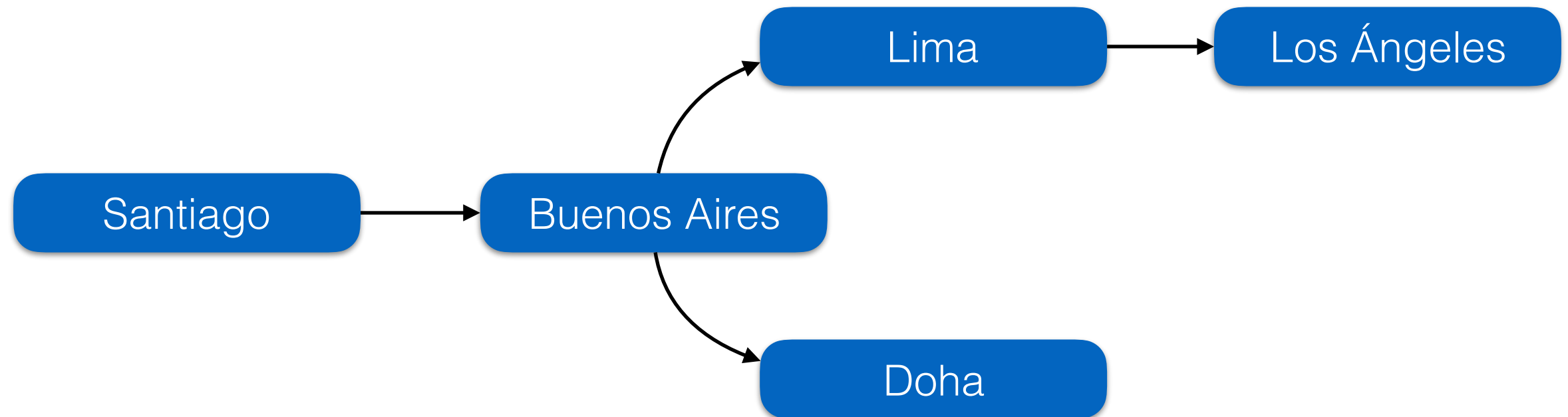
Recursión

Motivación

Ciudad Origen	Ciudad Destino
Santiago	Buenos Aires
Buenos Aires	Lima
Buenos Aires	Doha
Lima	Los Ángeles
...	...

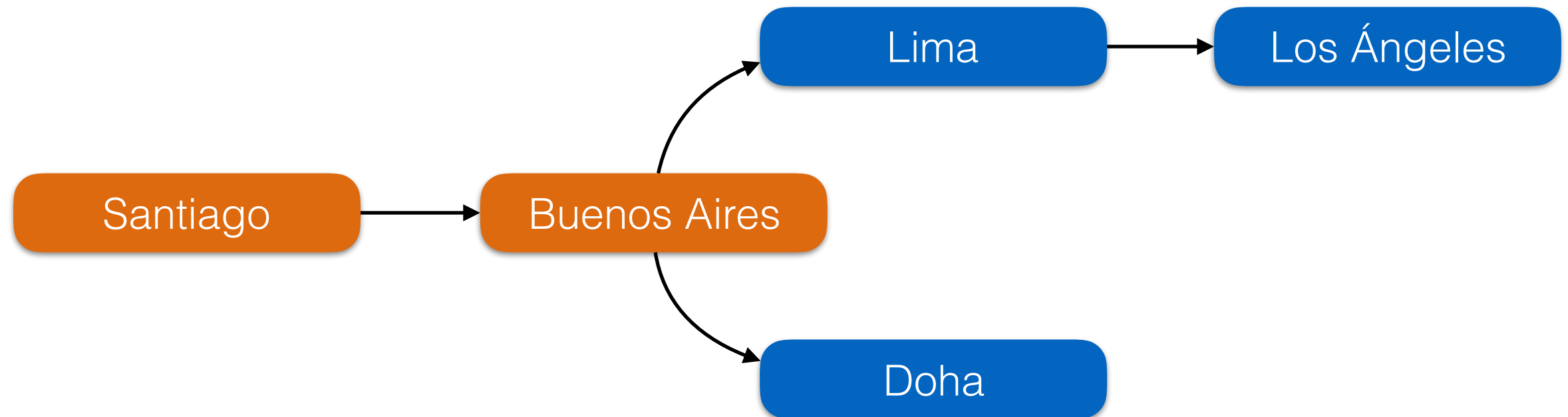
- ¿Cómo obtengo los vuelos directos?
- ¿Cómo obtengo los vuelos a una escala?
- ¿Cómo obtengo los vuelos a dos escalas?
- ...

Recursión



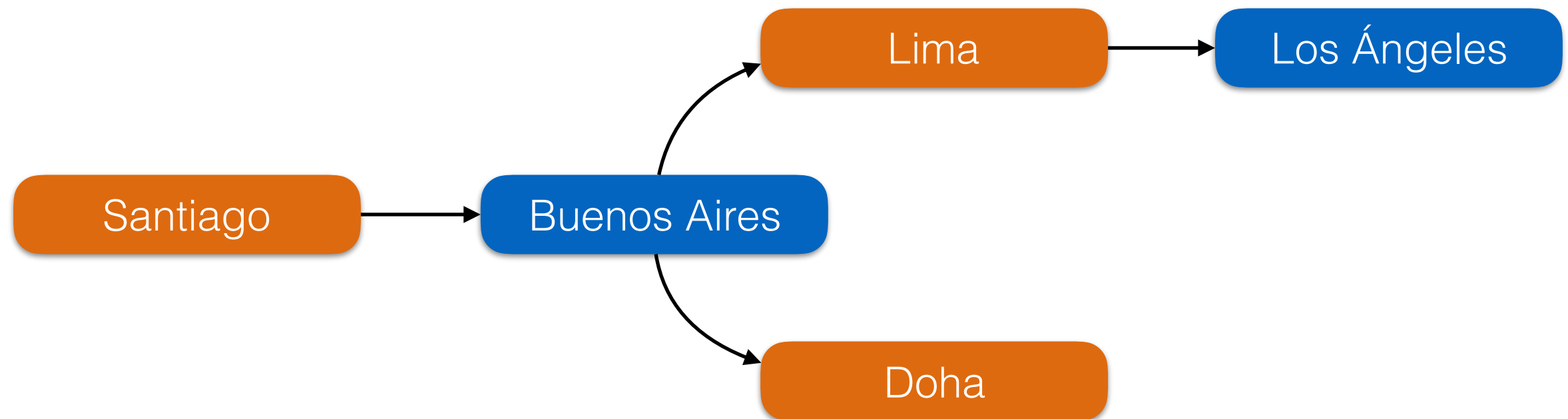
Recursión

Vuelos directos: 0 joins



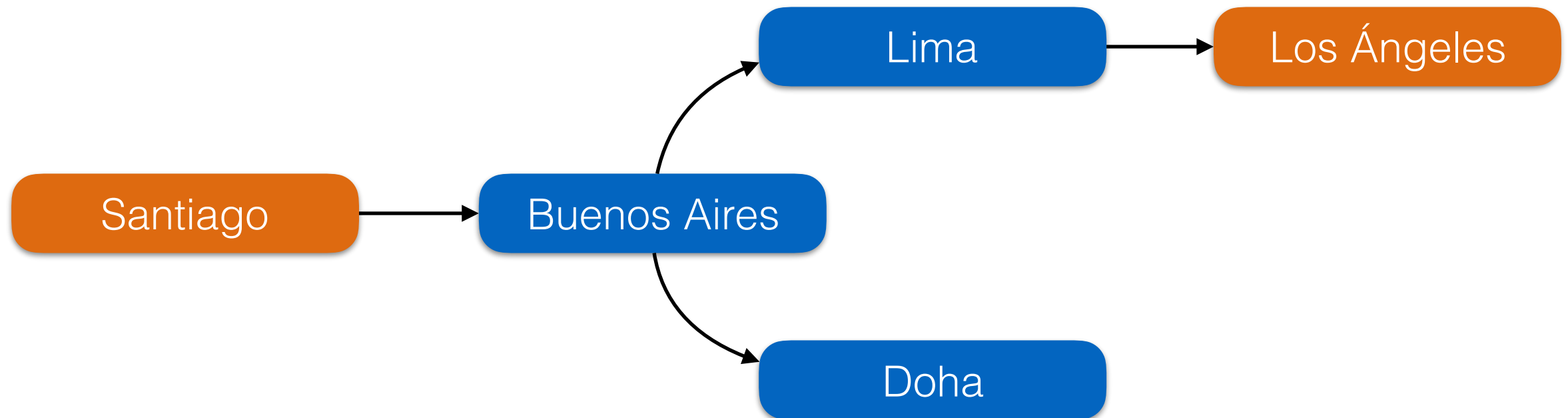
Recursión

Vuelos a una escala: 1 join



Recursión

Vuelos a dos escalas: 2 joins



Consultas Recursivas

- Puedo expresar la consulta “vuelos con **n** escalas”
- La consulta “vuelos con **n** o menos escalas” es la unión de los vuelos con 1, 2, ..., **n** escalas
- Pero con lo que sabemos de SQL **no** podemos expresar la consulta vuelos con cualquier número de escalas!

Consultas Recursivas

- Puedo expresar la consulta “vuelos con **n** escalas”
- La consulta “vuelos con **n** o menos escalas” es la unión de los vuelos con 1, 2, ..., **n** escalas
- Pero con lo que sabemos de SQL **no** podemos expresar la consulta vuelos con cualquier número de escalas!

¿Demostración?

Consultas Recursivas

- La comunidad advirtió este problema a los encargados del estándar
- Se agregó a SQL3 la capacidad de hacer consultas recursivas (SQL3 es la última gran revisión de SQL)
- Consultas recursivas son engorrosas y no son soportadas por todos los sistemas!

Consultas Recursivas

Evaluación

Para comprenderlas, expresaremos las reglas de la recursión para el ejemplo:

- Se genera una tabla auxiliar `alcanzo(co, cd)`
- Inicialmente `alcanzo(co, cd)` está vacía

Consultas Recursivas

Evaluación

Para comprenderlas, expresaremos las reglas de la recursión:

- En la primera iteración `alcanzo(co, cd)` será igual a `Vuelos_Directos(co, cd)`
- En las siguientes iteraciones `alcanzo(co, cd)` será igual al valor actual, unido con el resultado del join entre:

`alcanzo(co, cd), Vuelos_Directos(co, cd)`

Consultas Recursivas

Evaluación

Para comprenderlas, expresaremos las reglas de la recursión:

- La evaluación para cuando `alcanzo(co, cd)` sea igual entre una iteración y otra

Consultas Recursivas

Evaluación - Idea General

$$\textit{alcanzo}_0(\textit{co}, \textit{cd}) = \emptyset$$

$$\textit{alcanzo}_1(\textit{co}, \textit{cd}) = \textit{Vuelos_Directos}$$

$$\textit{alcanzo}_{i+1}(\textit{co}, \textit{cd}) = \textit{alcanzo}_i \cup \textit{alcanzo}_i \bowtie \textit{Vuelos_Directos}$$

Common Table Expressions (CTE)

Forma de hacer subconsultas:

```
WITH <Tabla Auxiliar>(<Atributos>) AS  
(  
    <Consulta a la Base de Datos>  
)  
<Consulta que utiliza a <Tabla Auxiliar>>
```

La consulta a la base de datos se almacena en la tabla auxiliar, pero se elimina después de realizar la consulta que la utiliza

Recursión

Es una CTE especial

```
WITH RECURSIVE<Tabla Auxiliar>(<Atributos>) AS  
(  
    <Consulta a la Base de Datos y a <Tabla Auxiliar>>  
)  
<Consulta que utiliza a <Tabla Auxiliar>>
```

Recursión

Es una CTE especial

```
WITH RECURSIVE<Tabla Auxiliar>(<Atributos>) AS  
(  
    <Consulta a la Base de Datos y a <Tabla Auxiliar>>  
)  
<Consulta que utiliza a <Tabla Auxiliar>>
```

La consulta anidada se computa y actualiza el valor de la tabla auxiliar

Recursión

Es una CTE especial

```
WITH RECURSIVE<Tabla Auxiliar>(<Atributos>) AS  
(  
    <Consulta a la Base de Datos y a <Tabla Auxiliar>>  
)  
<Consulta que utiliza a <Tabla Auxiliar>>
```

La consulta anidada se computa y actualiza el valor de la tabla auxiliar

Cuando la consulta anidada tiene las mismas tuplas que en la iteración anterior, la recursión termina

Recursión

Estructura de nuestra recursión:

```
WITH RECURSIVE<Tabla Auxiliar>(<Atributos>) AS  
(  
    <Inner Query (Consulta Base)>  
    UNION  
    <Outer Query (Fragmento Recursivo)>  
)  
<Consulta que utiliza a <Tabla Auxiliar>>
```

La Outer Query por lo general es un Join entre la **Tabla Auxiliar** y la **Inner Query**

Recursión

Ejemplo con Vuelos

```
WITH RECURSIVE Alcanzo(origen, destino) AS
(
    SELECT * FROM Vuelos

    UNION

    SELECT V.origen, A.destino
    FROM Vuelos V, Alcanzo A
    WHERE V.destino = A.origen
)
SELECT * FROM Alcanzo
```

Recursión

Ejemplo con Vuelos

- En la primera iteración la tabla recursiva está vacía
- Luego la tabla recursiva es igual a la relación base
- Luego vienen los caminos de largo 2, 3, 4...

Recursión

¡Cuidado!

- Tener cuidado cuando usar negación en la recursión
- Ojo con los nulos en la recursión
- Ojo con el uso de el operador **UNION ALL** (¿Por qué?)

Otras limitaciones de SQL

`Vuelos_Directos(Ciudad_Origen, Ciudad_Destino)`

¿Puedo pasar por todas las ciudades una vez,
partiendo y llegando a Santiago?

Otras limitaciones de SQL

- Podemos demostrar que la consulta anterior no es expresable en SQL, ni si quiera con recursión
- Para más detalles, tomar Lógica para Ciencias de la Computación