

NEO4J

Para cada cuestión, será necesario copiar el comando que se ha ejecutado en el editor de Neo4J para responder lo que se pide en cada cuestión. Además, se deberá adjuntar una captura de la tabla (formato *table* o *text*) o grafo resultante con las columnas que se piden.

Habrá alguna cuestión en la que además de adjuntar las capturas, se pide responder a alguna pregunta concreta analizando los datos obtenidos como resultado de la consulta.

CASO 1: identificar el número de personas contagiadas y de personas sanas en la muestra de 40 personas. Devolver el resultado en formato *table* o *text*. Los campos a devolver se muestran a continuación:

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with three tabs: 'Table' (selected), 'Text' (disabled), and 'Code' (disabled). The main area contains a query and its results. The query is:

```
1 MATCH (p:Persona)
2 RETURN COUNT(CASE WHEN p.estado = 'Contagiado' THEN 1 END) AS Total_Contagiados,
3     COUNT(CASE WHEN p.estado = 'Sano' THEN 1 END) AS Total_Sanos
```

The results are displayed in a table:

"Total_Contagiados"	"Total_Sanos"
10	30

MATCH (p:Persona)

RETURN COUNT(CASE WHEN p.estado = 'Contagiado' THEN 1 END) AS Total_Contagiados,
COUNT(CASE WHEN p.estado = 'Sano' THEN 1 END) AS Total_Sanos

CASO 2: Encontrar las personas sanas que han estado en contacto con una persona que ha dado positivo.

Debido a que nos encontramos en las primeras fases de estudio de la enfermedad, se da por supuesto que la COVID puede infectar a personas sanas que hayan estado en un mismo lugar en el que ha estado una persona contagiada, aunque sea en diferentes días.

Identificar todas las personas sanas que han estado en el mismo lugar (sin importar fecha ni hora) en el que también ha estado una persona contagiada (lógicamente el comienzo de la visita de la persona sana tiene que ser posterior al de la persona enferma, ya que es imposible contagiarse si la persona sana ha ido antes que la contagiada).

Devolver el resultado en formato table o text.

```
1 MATCH (p_sano:Persona {estado: 'Sano'}) - [:REALIZA_VISITA] → (v_sano:Visita)
  -[:A_ESTABLECIMIENTO] → (u:Ubicacion),
2 (p_contagiado:Persona {estado: 'Contagiado'}) - [:REALIZA_VISITA] →
  (v_contagiado:Visita) - [:A_ESTABLECIMIENTO] → (u)
3 WHERE v_sano.inicio_visita > v_contagiado.inicio_visita
4 RETURN DISTINCT p_sano.nombre AS Persona_Sana, u.nombre AS Establecimiento
```

The screenshot shows a Neo4j browser window with a query results table. The table has two columns: 'Persona_Sana' and 'Establecimiento'. The data is as follows:

Persona_Sana	Establecimiento
"Daniel Benitez"	"Hospital Rio Hortega"
"Lara Fernandez"	"Teatro Calderon"
"Alba Hoyos"	"Teatro Calderon"
"Daniel Benitez"	"Teatro Calderon"
"Ramon Cuadrado"	"Teatro Calderon"
"Francisco Pardo"	"Teatro Calderon"
"Jorge Cuenca"	"Teatro Calderon"
"Alba Hoyos"	"Rio Shopping"
"Ramon Cuadrado"	"Rio Shopping"

```
MATCH (p_sano:Persona {estado: 'Sano'}) - [:REALIZA_VISITA] -> (v_sano:Visita) -
  -[:A_ESTABLECIMIENTO] -> (u:Ubicacion),
  (p_contagiado:Persona {estado: 'Contagiado'}) - [:REALIZA_VISITA] -> (v_contagiado:Visita) -
  -[:A_ESTABLECIMIENTO] -> (u)

WHERE v_sano.inicio_visita > v_contagiado.inicio_visita

RETURN DISTINCT p_sano.nombre AS Persona_Sana, u.nombre AS Establecimiento
```

Según se deduce del enunciado del caso 2, la idea es identificar sólo a aquellas personas sanas que han iniciado su visita a una ubicación de manera posterior a la visita a esa misma ubicación de al menos una persona contagiada. De ahí que el enunciado diga “el comienzo de la visita de la persona sana tiene que ser posterior al de la persona enferma”, y el código contenga una cláusula WHERE en la que se comparan los inicios de las visitas a una ubicación de una persona sana y una contagiada, siendo el inicio de la persona sana mayor que (posterior) el inicio de la visita de la persona contagiada.

Una alternativa podría ser que, independientemente de la hora de inicio de la visita de la persona contagiada, que la hora de fin de la visita de la persona contagiada fuese anterior a la hora de inicio de la persona sana. De esta forma, estaríamos asegurándonos de que la persona sana comienza su visita a cierta ubicación con posterioridad a que la persona contagiada la abandonara.

En cualquier caso, se considera que el código resultante respeta lo sugerido por el enunciado y ya recoge a las personas sanas que comienzan su visita a una ubicación con posterioridad a que lo haga una persona contagiada.

Primero se buscan las personas sanas y las contagiadas que hayan realizado una visita a un establecimiento y establecemos la condición de que el inicio de la visita de la persona sana sea mayor (posterior) a la del inicio de la persona contagiada.

En el RETURN se emplea un DISTINCT para asegurarnos de que cada combinación Persona_Sana – Establecimiento aparece una única vez, puesto que si obtenemos diferentes combinaciones de persona sana en una ubicación con una persona contagiada diferente, nos aparecería el registro de esta persona sana y la ubicación en cuestión tantas veces como personas contagiadas existan en la combinación. Por ejemplo, si Alba Hoyos apareciese combinada con 3 personas contagiadas en “Teatro Calderon”, sólo aparecería una vez en nuestra tabla.

Por último, reseñar que, si se deseara ver en qué Establecimientos ha estado cada persona sana con posterioridad a una persona contagiada, bastaría con añadir una línea extra al final del código:

```
ORDER BY Persona_Sana
```

De esta manera, se podría visualizar rápidamente las veces que aparece cada persona sana y los establecimientos en los que ha estado con posterioridad a una persona contagiada.

CASO 3: mostrar grafo con las personas sanas que han coincidido con una persona contagiada, concretamente con “Marcelino Rodriguez”. Hay que mostrar en un grafo con este nodo persona, junto con todos los lugares que ha visitado, y con los nodos de etiqueta persona sana que han visitado también ese lugar posteriormente.

Observando los resultados del grafo, comentar seis personas de las que han estado en alguna ubicación en la que ha estado Marcelino después de haber estado él, tienen menos riesgo que el resto de haberse contagiado que el resto:

```

1 MATCH (marcelino:Persona {nombre: 'Marcelino Rodriguez'})-[:REALIZA_VISITA]→
  (v_marc:Visita)-[:A_ESTABLECIMIENTO]→(u:Ubicacion),
2 (persona_sana:Persona {estado: 'Sano'})-[:REALIZA_VISITA]→(v_sana:Visita)-
  [:A_ESTABLECIMIENTO]→(u)
3 WHERE v_marc.inicio_visita < v_sana.inicio_visita
4 RETURN marcelino, u, persona_sana
  
```



```
MATCH (marcelino:Persona {nombre: 'Marcelino Rodriguez'})-[:REALIZA_VISITA]->(v_marc:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion),  
(persona_sana:Persona {estado: 'Sano'})-[:REALIZA_VISITA]->(v_sana:Visita)-[:A_ESTABLECIMIENTO]->(u)  
WHERE v_marc.inicio_visita < v_sana.inicio_visita  
RETURN marcelino, u, persona_sana
```

Se busca a Marcelino Rodríguez y sus visitas a los diferentes establecimientos, así como todas aquellas personas sanas que también han visitado los mismos establecimientos. Establecemos la condición de que el inicio de la visita de Marcelino debe ser inferior (anterior) al inicio de la visita de las personas coincidentes, de modo que tenemos en consideración tanto a coincidentes temporales como personas que visitaron la ubicación con posterioridad a que Marcelino la hubiera abandonado. Y finalmente, devolvemos a Marcelino, las ubicaciones y las personas sanas que cumplen esa condición en un grafo.

Dado que en el grafo no aparece información como el tiempo que pasó cada uno de los coincidentes o el propio Marcelino en cada ubicación, por lo que no se puede establecer basándonos en tiempos de exposición quiénes tienen mayor riesgo, ni información sobre la distancia temporal entre el final de la visita de Marcelino y el inicio de la visita de los coincidentes, o la coincidencia física entre éstos y el propio Marcelino, la única información en la que podemos basarnos para establecer quiénes podrían tener un menor riesgo es el número de establecimientos visitados con posterioridad a que lo hiciera Marcelino así como el número de veces, de forma que aquellas personas que sólo visitaron una única ubicación coincidente con Marcelino tienen menos probabilidad de ser contagiados por éste, y de entre estas personas, aquellas que menos visitaron el establecimiento en cuestión tienen menos probabilidad de contagiarse que aquellas que visitaron el establecimiento en varias ocasiones.

En conclusión, vamos a seleccionar a Francisco Pardo (1) y Jorge Cuenca (2) que visitaron una única vez “Teatro Calderón”, Enrique Caminero (3) que visitó una única vez “Rio Shopping”, y María Soto (4), Sheila Diez (5) y Fátima Camino (6) que visitaron una única vez “Bar comandante” (descartamos a Lara Fernández que, aunque sólo coincide en una única ubicación con Marcelino, realiza 2 visitas a “Teatro Calderón” después de que lo hiciera Marcelino, por lo que consideramos que se expone más que quienes realizan una única visita). El resto quedan descartados por haber coincidido en al menos 2 ubicaciones distintas después de la visita de Marcelino.

CASO 4: construir la misma consulta anterior, pero mostrando el resultado como una tabla y no como un grafo (formato *table* o *text*) mostrando los campos:

- Esparcidor_virus.
- Comienzo_esparcimiento_virus.
- Establecimiento.
- Persona_en_riesgo.
- Inicio_visita_persona_en_riesgo.

Table	Esparcidor_virus	Comienzo_esparcimiento_virus	Ubicacion	Inicio_visita_persona_en_riesgo	Persona_en_riesgo

```

1 MATCH (esparcidor:Persona {nombre: 'Marcelino Rodriguez'})-[:REALIZA_VISITA]→
2   (v:Visita)-[:A_ESTABLECIMIENTO]→(e:Ubicacion),
3     (persona_riesgo:Persona {estado: 'Sano'})-[:REALIZA_VISITA]→
4   (v_riesgo:Visita)-[:A_ESTABLECIMIENTO]→(e)
5 WHERE v_riesgo.inicio_visita > v.inicio_visita
6 RETURN
7   esparcidor.nombre AS Esparcidor_virus,
8   v.inicio_visita AS Comienzo_esparcimiento_virus,
9   e.nombre AS Establecimiento,
10  persona_riesgo.nombre AS Persona_en_riesgo,
11  v_riesgo.inicio_visita AS Inicio_visita_persona_en_riesgo
12 ORDER BY Comienzo_esparcimiento_virus

```

\$ MATCH (esparcidor:Persona {nombre: 'Marcelino Rodriguez'})-[:REALIZA_VISITA]→(v:Visita)-[:A_ESTABLECIMIENTO]→(e:Ub:

Table	"Esparcidor_virus"	"Comienzo_esparcimiento_virus"	"Establecimiento"	"Persona_en_riesgo"	"Inicio_visita_persona_en_riesgo"
A	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Francisco Pardo"	"2020-09-02T17:59:17Z"
Text	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Jorge Cuenca"	"2020-09-02T17:56:22Z"
Code	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Ramon Cuadrado"	"2020-09-02T19:31:35Z"
	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Daniel Benitez"	"2020-09-03T17:53:52Z"
	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Lara Fernandez"	"2020-09-03T09:24:18Z"
	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Lara Fernandez"	"2020-09-04T20:23:18Z"
	"Marcelino Rodriguez"	"2020-08-30T18:39:54Z"	"Teatro Calderon"	"Alba Hoyos"	"2020-09-02T17:57:35Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Fatima Camino"	"2020-09-02T08:30:25Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Sergio Barbero"	"2020-09-01T17:21:13Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Maria Soto"	"2020-09-01T17:21:23Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Sheila Diez"	"2020-09-02T18:11:21Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Daniel Benitez"	"2020-08-31T16:52:36Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Ramon Cuadrado"	"2020-09-01T17:41:42Z"
	"Marcelino Rodriguez"	"2020-08-31T11:39:54Z"	"Bar Comandante"	"Alba Hoyos"	"2020-09-01T16:12:30Z"
	"Marcelino Rodriguez"	"2020-09-01T18:04:37Z"	"Rio Shopping"	"Sergio Barbero"	"2020-09-01T19:41:39Z"
	"Marcelino Rodriguez"	"2020-09-01T18:04:37Z"	"Rio Shopping"	"Enrique Caminero"	"2020-09-03T18:51:03Z"
	"Marcelino Rodriguez"	"2020-09-01T18:04:37Z"	"Rio Shopping"	"Ramon Cuadrado"	"2020-09-02T16:51:35Z"
	"Marcelino Rodriguez"	"2020-09-01T18:04:37Z"	"Rio Shopping"	"Alba Hoyos"	"2020-09-03T15:52:30Z"

```

MATCH (esparcidor:Persona {nombre: 'Marcelino Rodriguez'})-[:REALIZA_VISITA]->(v:Visita)-
[:A_ESTABLECIMIENTO]->(e:Ubicacion),
(persona_riesgo:Persona {estado: 'Sano'})-[:REALIZA_VISITA]->(v_riesgo:Visita)-
[:A_ESTABLECIMIENTO]->(e)
WHERE v_riesgo.inicio_visita > v.inicio_visita
RETURN
    esparcidor.nombre AS Esparcidor_virus,
    v.inicio_visita AS Comienzo_esparcimientovirus,
    e.nombre AS Establecimiento,
    persona_riesgo.nombre AS Persona_en_riesgo,
    v_riesgo.inicio_visita AS Inicio_visita_persona_en_riesgo
ORDER BY Comienzo_esparcimientovirus

```

La búsqueda de datos es igual que en el punto anterior, siendo el RETURN lo que cambia: ahora vamos a devolver los campos para esparcidor.nombre (que es Marcelino), v.inicio_visita (el inicio de la visita de Marcelino a cada establecimiento), e.nombre (el nombre de los establecimientos visitados por Marcelino), persona_riesgo.nombre (el nombre de las personas coincidentes con Marcelino) y v_riesgo.inicio_visita (la fecha en que visitan estas personas los establecimientos que fueron visitados por Marcelino anteriormente).

De esta forma podemos ver una tabla con la información requerida que amplía la información del grafo anterior.

CASO 5.1: construir una tabla (formato *text* y *table*) que identifique para cada persona contagiada (columna uno), las personas sanas con las que ha coincidido en un establecimiento en el mismo tiempo. Construir como segunda columna un *array* de elementos JSON llamado “Contactos” con claves:

- Persona_en_contacto.
- Establecimiento.
- Fecha_comienzo_solapamiento.
- Fecha_fin_solapamiento.

Se muestra a continuación un ejemplo del formato de cada documento JSON y de los dos campos a obtener:

```
{  
    "Fecha_fin_solapamiento":  
        "2020-09-02T18:44:54Z",  
    "Establecimiento": "Rio  
Shopping",  
    "Fecha_comienzo_solapamiento":  
        "2020-09-02T17:18:16Z",  
    "Persona_en_contacto": "Ramon  
Cuadrado"  
}
```

 Table	Persona_Contagiada	Contactos

```
MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]->(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion),  
(p_sano:Persona {estado: 'Sano'})-[:REALIZA_VISITA]->(v_sano:Visita)-[:A_ESTABLECIMIENTO]->(u)  
  
WHERE v_sano.inicio_visita < v_contagiado.fin_visita AND v_contagiado.inicio_visita <  
v_sano.fin_visita  
  
WITH p_contagiado, p_sano, u,  
  
CASE WHEN datetime(v_contagiado.inicio_visita) > datetime(v_sano.inicio_visita) THEN  
datetime(v_contagiado.inicio_visita) ELSE datetime(v_sano.inicio_visita) END AS  
fecha_comienzo_solapamiento,
```

```

CASE WHEN datetime(v_contagiado.fin_visita) < datetime(v_sano.fin_visita) THEN
datetime(v_contagiado.fin_visita) ELSE datetime(v_sano.fin_visita) END AS
fecha_fin_solapamiento

RETURN

p_contagiado.nombre AS Persona_Contagiada,
collect(
{
    Persona_en_contacto: p_sano.nombre,
    Establecimiento: u.nombre,
    Fecha_comienzo_solapamiento: apoc.date.format(timestamp(fecha_comienzo_solapamiento),
"ms", "yyyy-MM-dd'T'HH:mm:ss'Z'"),
    Fecha_fin_solapamiento: apoc.date.format(timestamp(fecha_fin_solapamiento), "ms", "yyyy-
MM-dd'T'HH:mm:ss'Z'")
}
) AS Contactos

1 MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]→(u:Ubicacion),
2 (p_sano:Persona {estado: 'Sano'})-[:REALIZA_VISITA]→(v_sano:Visita)-[:A_ESTABLECIMIENTO]→(u)
3 WHERE v_sano.inicio_visita < v_contagiado.fin_visita AND v_contagiado.inicio_visita < v_sano.fin_visita
4 WITH p_contagiado, p_sano, u,
5 CASE WHEN datetime(v_contagiado.inicio_visita) > datetime(v_sano.inicio_visita) THEN
    datetime(v_contagiado.inicio_visita) ELSE datetime(v_sano.inicio_visita) END AS fecha_comienzo_solapamiento.
6
7 Persona_en_contacto: p_sano.nombre,
8 Establecimiento: u.nombre,
9 Fecha_comienzo_solapamiento: apoc.date.format(timestamp(fecha_comienzo_solapamiento), "ms", "yyyy-MM-
dd'T'HH:mm:ss'Z'"),
10 Fecha_fin_solapamiento: apoc.date.format(timestamp(fecha_fin_solapamiento), "ms", "yyyy-MM-
dd'T'HH:mm:ss'Z'")
11 }
12 ) AS Contactos

```

```
$ MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→(v_contagiado:Visita)-[:A_E
```

Table	Persona_Contagiada	Contactos
Text	"Saturnino Castaño"	[<pre>{ "Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T19:17:29Z", "Persona_en_contacto": "Celia Cuenca" }</pre>
Code		[<pre>{ "Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento":</pre>

Started streaming 8 records after 48 ms and completed after 48 ms.

```
$ MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→(v_contagiado:Visit
```

	"Persona_Contagiada"	"Contactos"
A Text	"Saturnino Castaño"	[{"Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T19:17:29Z", "Persona_en_contacto": "Celia Cuenca"}, {"Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T19:18:16Z", "Persona_en_contacto": "Shelia Diez"}, {"Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T19:51:03Z", "Persona_en_contacto": "Enrique Caminero"}, {"Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T18:24:19Z", "Persona_en_contacto": "Noelia Valentin"}, {"Fecha_fin_solapamiento": "2020-09-02T20:59:21Z", "Establecimiento": "Bar La Chica de Ayer", "Fecha_comienzo_solapamiento": "2020-09-02T19:11:30Z", "Persona_en_contacto": "Sergio Barbero"}]
	"Berta Romero"	[{"Fecha_fin_solapamiento": "2020-08-31T22:16:03Z", "Establecimiento": "Rio Shopping", "Fecha_comienzo_solapamiento": "2020-08-31T19:13:03Z", "Persona_en_contacto": "Alvaro Fernandez"}, {"Fecha_fin_solapamiento": "2020-08-31T14:33:03Z", "Establecimiento": "Colegio San José", "Fecha_comienzo_solapamiento": "2020-08-31T09:36:03Z", "Persona_en_contacto": "Alvaro Fernandez"}, {"Fecha_fin_solapamiento": "2020-08-31T14:23:18Z", "Establecimiento": "Colegio San José", "Fecha_comienzo_solapamiento": "2020-08-31T09:38:23Z", "Persona_en_contacto": "Maria Soto"}, {"Fecha_fin_solapamiento": "2020-08-31T14:19:06Z", "Establecimiento": "Colegio San José", "Fecha_comienzo_solapamiento": "2020-08-31T09:36:03Z", "Persona_en_contacto": "Esther Liebana"}, {"Fecha_fin_solapamiento": "2020-08-31T22:16:03Z", "Establecimiento": "Rio Shopping", "Fecha_comienzo_solapamiento": "2020-08-31T19:11:30Z"}]

En primer lugar se seleccionan las personas contagiadas y sanas y los lugares que ha visitado cada uno, con las condiciones de que el inicio de la visita de la persona sana sea menor que (anterior) al final de la visita de la persona contagiada y que el inicio de la visita de la persona contagiada sea inferior (anterior) al final de la visita de la persona sana. De esta forma nos aseguramos de que, independientemente de quién llegara antes y quién se fuera antes del establecimiento, ambas personas, sana y contagiada, han estado físicamente en el mismo local al mismo tiempo.

Encadenamos la consulta inicial mediante el WITH a otra consulta en la que estudiaremos las fechas de inicio y fin de visita para contagiados y sanos, seleccionando como fecha de comienzo del solapamiento la fecha más tardía de inicio de las visitas y como fecha de fin de solapamiento, la fecha más temprana de final de las visitas. De esta forma se obtiene la ventana temporal en la que contagiado y sano han estado en el mismo establecimiento en el mismo momento de tiempo. Ante la imposibilidad de usar `.epochMillis` (`Neo.ClientError.Procedure.ProcedureCallFailed`) se han convertido las fechas mediante `datetime()`.

Finalmente, se retorna tanto el nombre de los contagiados como una lista en una segunda columna de documentos JSON con el formato y la información requerida.

Debido a que el Query Input de Neo4J es bastante estrecho, se han sacado varias capturas de

pantalla del código empleado. También se ha incluido captura de pantalla del resultado en formato *text* ya que en formato *table* apenas se veían un par de documentos JSON de la primera persona contagiada. Con esta opción, se pueden visualizar al menos dos personas contagiadas y el JSON de ambas.

CASO 5.2: una vez obtenida la consulta anterior, conseguir, añadiendo tres sentencias a esta, una tabla que tenga el nombre de la persona contagiada y otra columna con el número de personas sanas con las que ha tenido contacto, obteniendo ese número a partir de los elementos del array de elementos JSON. Ordenar los resultados por “Numero_de_contactos_sanos” descendente:

 Table	Persona_Contagiada	Numero_de_contactos_sanos
---	--------------------	---------------------------

```
MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]->(v_contagiado:Visita)-[:ESTABLECIMIENTO]->(u:Ubicacion),
(p_sano:Persona {estado: 'Sano'})-[:REALIZA_VISITA]->(v_sano:Visita)-[:ESTABLECIMIENTO]->(u)

WHERE v_sano.inicio_visita < v_contagiado.fin_visita AND v_contagiado.inicio_visita <
v_sano.fin_visita

WITH p_contagiado, p_sano, u,
CASE WHEN datetime(v_contagiado.inicio_visita) > datetime(v_sano.inicio_visita) THEN
datetime(v_contagiado.inicio_visita) ELSE datetime(v_sano.inicio_visita) END AS
fecha_comienzo_solapamiento,
CASE WHEN datetime(v_contagiado.fin_visita) < datetime(v_sano.fin_visita) THEN
datetime(v_contagiado.fin_visita) ELSE datetime(v_sano.fin_visita) END AS
fecha_fin_solapamiento

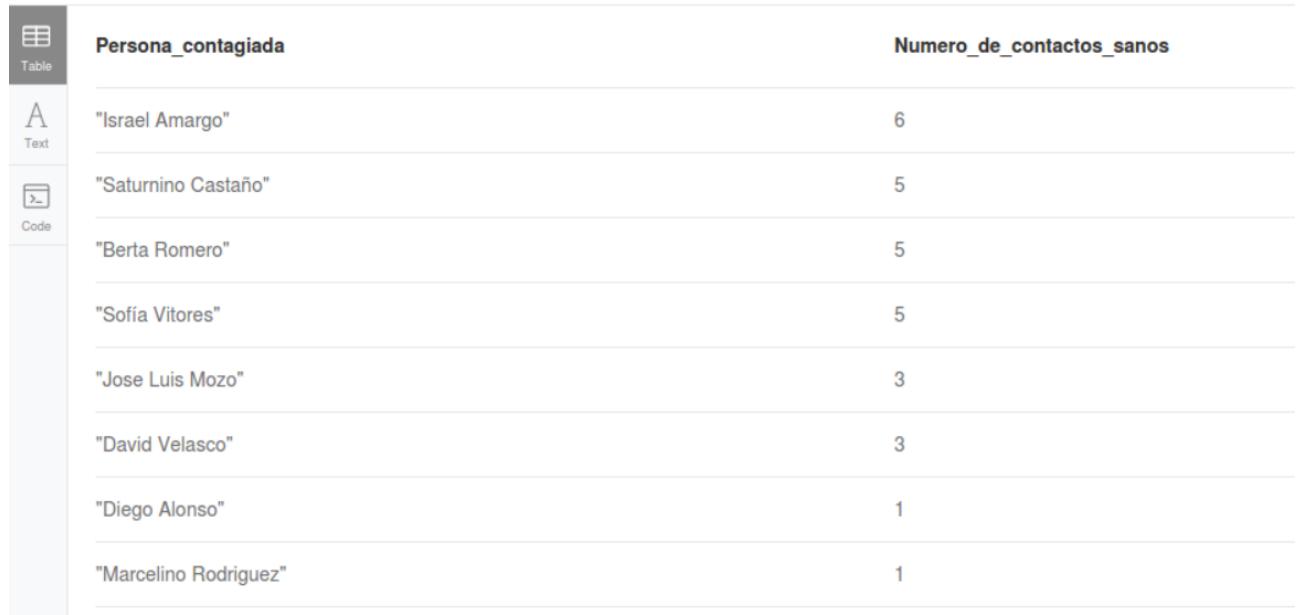
RETURN
p_contagiado.nombre AS Persona_Contagiada,
size(collect(
{
    Persona_en_contacto: p_sano.nombre,
    Establecimiento: u.nombre,
    Fecha_comienzo_solapamiento: apoc.date.format(timestamp(fecha_comienzo_solapamiento),
"ms", "yyyy-MM-dd'T'HH:mm:ss'Z'"),
    Fecha_fin_solapamiento: apoc.date.format(timestamp(fecha_fin_solapamiento), "ms", "yyyy-
MM-dd'T'HH:mm:ss'Z'")
})
```

) AS Numero_de_contactos_sanos

ORDER BY Numero_de_contactos_sanos DESC

```
1 MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]→(u:Ubicacion),
2 (p_sano:Persona {estado: 'Sano'})-[:REALIZA_VISITA]→(v_sano:Visita)-[:A_ESTABLECIMIENTO]→(u)
3 WHERE v_sano.inicio_visita < v_contagiado.fin_visita AND v_contagiado.inicio_visita < v_sano.fin_visita
4 WITH p_contagiado, p_sano, u,
5 CASE WHEN datetime(v_contagiado.inicio_visita) > datetime(v_sano.inicio_visita) THEN
       datetime(v_contagiado.inicio_visita) ELSE datetime(v_sano.inicio_visita) END AS fecha_comienzo_solapamiento,
6 CASE WHEN datetime(v_contagiado.fin_visita) < datetime(v_sano.fin_visita) THEN
       datetime(v_contagiado.fin_visita) ELSE datetime(v_sano.fin_visita) END AS fecha_fin_solapamiento
7 RETURN
8   p_contagiado.nombre AS Persona_Contagiada,
9   size(collect(
10    {
11      Persona_en_contacto: p_sano.nombre,
12      Establecimiento: u.nombre,
13      Fecha_comienzo_solapamiento: apoc.date.format(timestamp(fecha_comienzo_solapamiento), "ms", "yyyy-MM-
14 dd'T'HH:mm:ss'Z'"),
15      Fecha_fin_solapamiento: apoc.date.format(timestamp(fecha_fin_solapamiento), "ms", "yyyy-MM-
16 dd'T'HH:mm:ss'Z'")
16    }) AS Numero_de_contactos_sanos
16  )) AS Numero_de_contactos_sanos
17 ORDER BY Numero_de_contactos_sanos DESC
```

\$ MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→(v_contagiado:Visita)-[:A_



The screenshot shows the Neo4j browser interface with a table results panel. On the left, there are three tabs: 'Table' (selected), 'Text', and 'Code'. The table has two columns: 'Persona_contagiada' and 'Numero_de_contactos_sanos'. The data is as follows:

Persona_contagiada	Numero_de_contactos_sanos
"Israel Amargo"	6
"Saturnino Castaño"	5
"Berta Romero"	5
"Sofía Vidores"	5
"Jose Luis Mozo"	3
"David Velasco"	3
"Diego Alonso"	1
"Marcelino Rodríguez"	1

Para conseguir el conteo de contactos sanos de cada persona contagiada a partir de la consulta anterior, empleando para ello el *array* de elementos JSON, basta con envolver el método *collect()*

que lo genera con un método `size()`, el cuál calcula el número de elementos del `array`. De esta forma se convierte la anterior columna de `array` de elementos JSON en una columna de conteo de casos.

Basta con añadir al final un `ORDER BY` que logre la ordenación descendente de las observaciones por número de personas sanas contactadas.

Una forma alternativa que hubiera necesitado de las 3 líneas de comandos que sugiere el enunciado hubiera sido transformar el `RETURN` en un `WITH` y, manteniendo el `collect()` tal como aparece en el ejercicio anterior (es decir, sin pasarlo directamente a un método `size()`), crear un nuevo `RETURN` en el que se hiciera el conteo empleando, ahora sí, el método `size()`. El final del código vendría a ser algo así:

```
WITH p_contagiado.nombre AS Persona_contagiada, collect(...) AS Contactos  
RETURN Persona_contagiada, size(Contactos) AS Numero_de_contactos_sanos  
ORDER BY Numero_de_contactos_sanos DESC
```

donde “`WITH`” sustituye al `RETURN` del código del ejercicio anterior y `collect(...)` mantiene toda su estructura. Todo el resto del código anterior a estas 3 últimas líneas queda exactamente igual.

Estas dos formas parecidas (realmente iguales, aunque con un paso desglosado en la segunda opción y abreviado en la primera, que realiza el `collect()` y el `size()` en el mismo paso) logran crear una columna de conteo del número de personas sanas contactadas por cada persona contagiada a partir del `array` de elementos JSON.

CASO 6: encontrar a aquellas personas (si es que hay alguna) que visitaron un establecimiento incluso después de saber que habían dado positivo en el test. Los campos de salida a mostrar son:

Contagiado	Fecha_inicio_visita	Fecha_Confirmacion_Contagio
------------	---------------------	-----------------------------

```
MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]->(v:Visita)
```

```
WHERE v.inicio_visita > p_contagiado.hora_result_test
```

```
RETURN p_contagiado.nombre AS Contagiado, v.inicio_visita AS Fecha_inicio_visita,  
p_contagiado.hora_result_test AS Fecha_confirmacion_contagio
```

```
1 MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]->(v:Visita)  
2 WHERE v.inicio_visita > p_contagiado.hora_result_test  
3 RETURN p_contagiado.nombre AS Contagiado, v.inicio_visita AS  
Fecha_inicio_visita, p_contagiado.hora_result_test AS  
Fecha_confirmacion_contagio
```

\$ MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]->(v:Visita... | ↴ | ⌂ | ↵

Contagiado	Fecha_inicio_visita	Fecha_confirmacion_contagio
"Marta Valle"	"2020-09-03T09:23:21Z"	"2020-09-01T19:07:22Z"

Started streaming 1 records after 1 ms and completed after 5 ms.

Primero buscamos las personas con estado “Contagiado” y las visitas que realizan, le aplicamos la condición de que la hora y fecha de inicio de la visita sea mayor que (posterior) la hora y fecha de confirmación del resultado del test, y finalmente retornamos los tres datos en una tabla.

Se puede apreciar que la única persona que realizó una visita a un establecimiento público con posterioridad a conocer el resultado positivo del test es Marta Valle, quien conoció el resultado del test el día 01/09/20 y realizó una visita a un establecimiento el 03/09/20.

CASO 7: ahora que se han obtenido todas las personas sanas que coincidieron en algún establecimiento con alguna contagiada, se quiere averiguar el tiempo exacto (duración) que coincidió cada persona sana con la persona contagiada.

Expresar la duración en horas y redondeada a cuatro decimales. Devolver el resultado en formato *table* o *text*.

```
MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]-  
>(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion)<-[A_ESTABLECIMIENTO]-  
(v_sano:Visita)<-[REALIZA_VISITA]-(p_sano:Persona {estado:"Sano"})  
  
WHERE v_sano.inicio_visita < v_contagiado.fin_visita  
  
AND v_sano.fin_visita > v_contagiado.inicio_visita  
  
WITH p_contagiado.nombre AS Contagiado, p_sano.nombre AS Persona_Sana, u.nombre AS  
Establecimiento,  
    max(v_sano.inicio_visita, v_contagiado.inicio_visita) AS Fecha_comienzo_solapamiento,  
    min(v_sano.fin_visita, v_contagiado.fin_visita) AS Fecha_fin_solapamiento  
  
WITH Contagiado, Persona_Sana, Establecimiento,  
    (toFloat(datetime(Fecha_fin_solapamiento).epochMillis) -  
    toFloat(datetime(Fecha_comienzo_solapamiento).epochMillis)) / (1000*60*60) AS  
Duracion_en_horas  
  
RETURN Contagiado, Persona_Sana, Establecimiento,  
    round(Duracion_en_horas*10000)/10000 AS Duracion_en_horas  
  
ORDER BY Duracion_en_horas DESC  
  
1 MATCH (p_contagiado:Persona {estado: "Contagiado"})-[:REALIZA_VISITA]->  
    (v_contagiado:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion)  
    <-[A_ESTABLECIMIENTO]-(v_sano:Visita)<-[REALIZA_VISITA]-(p_sano:Persona  
    {estado: "Sano"})  
2 WHERE v_sano.inicio_visita < v_contagiado.fin_visita  
3 AND v_sano.fin_visita > v_contagiado.inicio_visita  
4 WITH p_contagiado.nombre AS Contagiado, p_sano.nombre AS Persona_Sana, u.nombre  
  
    AS Establecimiento,  
5     max(v_sano.inicio_visita, v_contagiado.inicio_visita) AS  
    Fecha_comienzo_solapamiento,  
6     min(v_sano.fin_visita, v_contagiado.fin_visita) AS Fecha_fin_solapamiento  
7 WITH Contagiado, Persona_Sana, Establecimiento,  
8     (toFloat(datetime(Fecha_fin_solapamiento).epochMillis) -
```

```

8   (toFloat(datetime(Fecha_fin_solapamiento).epochMillis) -
9    toFloat(datetime(Fecha_comienzo_solapamiento).epochMillis)) / (1000*60*60) AS
10   Duracion_en_horas
11  RETURN Contagiado, Persona_Sana, Establecimiento,
12      round(Duracion_en_horas*10000)/10000 AS Duracion_en_horas
13  ORDER BY Duracion_en_horas DESC

```

\$ MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]-(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]-(u:Ubicacion)

Contagiado	Persona_Sana	Establecimiento	Duracion_en_horas
"Berta Romero"	"Esther Liebana"	"Colegio San José"	5.3
"Sofía Vidores"	"Esther Liebana"	"Colegio San José"	5.3
"David Velasco"	"Sergio Barbero"	"Colegio Amor de Dios"	5.2475
"David Velasco"	"Irene Montes"	"Colegio Amor de Dios"	5.1667
"Berta Romero"	"Alvaro Fernandez"	"Colegio San José"	5.1194
"Sofía Vidores"	"Alvaro Fernandez"	"Colegio San José"	5.1194
"David Velasco"	"Noelia Valentin"	"Colegio Amor de Dios"	5.055
"Berta Romero"	"Alvaro Fernandez"	"Rio Shopping"	4.75
"Berta Romero"	"Maria Soto"	"Colegio San José"	4.7486
"Sofía Vidores"	"Maria Soto"	"Colegio San José"	4.7486
"Berta Romero"	"Juan Robles"	"Rio Shopping"	4.5806
"Saturnino Castaño"	"Noelia Valentin"	"Bar La Chica de Aver"	4.4892

Para calcular el tiempo que han pasado en un mismo establecimiento y al mismo tiempo cada persona contagiada con cada persona sana, primero tenemos que seleccionarlos a ambos junto a las respectivas visitas a un establecimiento en común. Establecemos la condición de que ambos inicien su visita antes (menor que) de que la finalice la otra persona, de modo que exista un solapamiento.

Agregamos con WITH el nombre de ambas personas y del establecimiento, y establecemos el inicio y el final del solapamiento mediante el máximo de las fechas de inicio de visita (el solapamiento comienza cuando llega el segundo de ellos, es decir, el más tardío) y el mínimo de las fechas de final de visita (el solapamiento termina cuando se va el primero de ellos, pues dejan de estar juntos en el establecimiento).

A continuación, se calcula la duración en horas de este solapamiento, para lo cual se pasan las fechas a milisegundos y se establecen como *float*, se restan, y se pasan a horas dividiendo entre 1000*60*60 (se pasa de milisegundos a segundos, luego a minutos y finalmente a horas).

Al retornar los resultados, las horas obtenidas en el paso anterior se redondean, para lo cual hay que usar un pequeño engaño, ya que la versión de Neo4j de la máquina virtual no admite más de un parámetro en el método *round()*, el número a redondear. Multiplicando el número de horas por 10000 antes de redondear y dividiendo el resultado entre 10000 se consigue el efecto.

CASO 8: una persona ha estado en dos sitios diferentes con personas contagiadas, en uno estuvo una hora y media en contacto y en el otro dos. El total de exposición de esa persona habrá sido de tres horas y media.

La duración de cada contacto entre persona sana y contagiada será el resultado que se obtenga en la cuestión anterior. Por tanto, se puede utilizar la consulta anterior como base y será necesario añadirle algo más para conseguir el resultado esperado.

Si una persona sana coincidió con dos contagiadas el mismo día en el mismo establecimiento, también se sumará el tiempo que estuvo en contacto con cada contagiado, entendiendo que el haber estado rodeado de más contagiados supone que esa persona tenga un mayor riesgo de contraer la enfermedad.

Solamente se mostrarán en la tabla (formato *table* o *text*) las cinco personas sanas con más tiempo de exposición. A esas cinco personas se les realizará inmediatamente una llamada para que comiencen a guardar cuarentena. El tiempo total se mostrará en horas con redondeo a cuatro decimales (por ejemplo: 9,4972 horas, que serán nueve horas y 30 minutos).

Table	Persona_Sana	Tiempo_total_exposicion

```
MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]->(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion)<-[A_ESTABLECIMIENTO]-(v_sano:Visita)<[:REALIZA_VISITA]-(p_sano:Persona {estado:"Sano"})

WHERE v_sano.inicio_visita < v_contagiado.fin_visita
AND v_sano.fin_visita > v_contagiado.inicio_visita
WITH p_sano, p_contagiado,
      max(v_sano.inicio_visita, v_contagiado.inicio_visita) AS Fecha_comienzo_solapamiento,
      min(v_sano.fin_visita, v_contagiado.fin_visita) AS Fecha_fin_solapamiento
WITH p_sano,
      (toFloat(datetime(Fecha_fin_solapamiento).epochMillis) -
      toFloat(datetime(Fecha_comienzo_solapamiento).epochMillis)) / (1000*60*60) AS Duracion_en_horas
WITH p_sano,
      sum(Duracion_en_horas) AS Total_Exposicion_Horas
```

```

WITH p_sano,
    round(toInteger(Total_Exposicion_Horas * 10000)) / 10000 AS Total_Exposicion
ORDER BY Total_Exposicion DESC
LIMIT 5

RETURN p_sano.nombre AS Persona_Sana,
       Total_Exposicion AS Tiempo_Exposicion_Horas

```

```

1 MATCH (p_contagiado:Persona {estado:"Contagiado"})-
  [:REALIZA_VISITA]→(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]→
  (u:Ubicacion)←[:A_ESTABLECIMIENTO]-(v_sano:Visita)
  ←[:REALIZA_VISITA]-(p_sano:Persona {estado:"Sano"})
2 WHERE v_sano.inicio_visita < v_contagiado.fin_visita
3 AND v_sano.fin_visita > v_contagiado.inicio_visita
4 WITH p_sano, p_contagiado,
      max(v_sano.inicio_visita, v_contagiado.inicio_visita) AS
      Fecha_comienzo_solapamiento,
      min(v_sano.fin_visita, v_contagiado.fin_visita) AS
      Fecha_fin_solapamiento
7 WITH p_sano,
8     (toFloat(datetime(Fecha_fin_solapamiento).epochMillis) -
      toFloat(datetime(Fecha_comienzo_solapamiento).epochMillis)) /
      (1000*60*60) AS Duracion_en_horas
9 WITH p_sano,
10    sum(Duracion en horas) AS Total_Exposicion_Horas
13 ORDER BY Total_Exposicion DESC
14 LIMIT 5
15 RETURN p_sano.nombre AS Persona_Sana,
16       Total_Exposicion AS Tiempo_Exposicion_Horas

```

```
$ MATCH (p_contagiado:Persona {estado:"Contagiado"})-[:REALIZA_VISITA]→(v_co
```

Table	Persona_Sana	Tiempo_Exposicion_Horas
A Text	"Sergio Barbero"	15.7733
Code	"Esther Liebana"	10.6
	"Noelia Valentin"	9.5441
	"Maria Soto"	9.4972
	"Ramon Cuadrado"	5.8458

Started streaming 5 records after 52 ms and completed after 52 ms.

Para lograr el resultado de la consulta se han aplicado a la consulta anterior los siguientes cambios:

- Una vez obtenidas las duraciones de cada una de los solapamientos entre persona contagiada y persona sana, se emplea *sum()* para obtener el sumatorio de horas totales de exposición o solapamiento para cada persona sana.
- Se ordena por el total de horas de exposición de manera descendente y se limita el resultado a las 5 primeras observaciones, es decir, las 5 personas con mayor número de horas de exposición.

Si nos fijamos en la tabla del ejercicio anterior, Esther Liebana (sana) aparece dos veces como coincidente con personas contagiadas, siendo el mismo establecimiento y el mismo tiempo (5.3 horas) para ambas coincidencias. En esta tabla, aparece con un sumatorio de 10.6 horas, lo que nos indica que se están contabilizando bien las coincidencias con distintas personas en el mismo establecimiento y al mismo tiempo.

CASO 9: se pretende tratar de reducir la afluencia e implementar aún más medidas de precaución en aquellos establecimientos en los que hayan estado más tiempo personas contagiadas.

Se pide devolver una tabla que contenga cada establecimiento, que ha sido visitado por al menos una persona contagiada, el total de visitas de contagiados en cada establecimiento, el total de visitas en cada establecimiento, el porcentaje de visitas de contagiados respecto al total de visitas de cada establecimiento, y la ciudad a la que pertenece el establecimiento. Expresar el porcentaje redondeado a dos decimales.

Comentar cuáles son los dos establecimientos con mayor y los dos con menor porcentaje de visitas de contagiados respecto del total de cada establecimiento.

Table	Establecimiento	Num_visitas_Contagiados	Num_total_visitas	Porcentaje_visitas_de_personas_contagiadas	Ciudad
-------	-----------------	-------------------------	-------------------	--	--------

```
MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]->(v_contagiado:Visita)-[:A_ESTABLECIMIENTO]->(u:Ubicacion)-[:PARTE_DE]->(c:Ciudad)
WITH u, c, count(v_contagiado) AS Num_visitas_contagiados
MATCH (u)<-[A_ESTABLECIMIENTO]-(v:Visita)
WITH u, c, Num_visitas_contagiados, count(v) AS Num_total_visitas
WITH u.nombre AS Establecimiento,
    Num_visitas_contagiados,
    Num_total_visitas,
   toFloat(Num_visitas_contagiados) / Num_total_visitas * 100 AS Porcentaje_contagiados,
    c.nombre AS Ciudad
RETURN Establecimiento,
    Num_visitas_contagiados,
    Num_total_visitas,
    toFloat(toInteger(Porcentaje_contagiados * 100)) / 100 AS
    Porcentaje_visitas_personas_contagiadas,
    Ciudad
ORDER BY Porcentaje_visitas_personas_contagiadas DESC
```

```
1 MATCH (p_contagiado:Persona {estado: 'Contagiado'})-[:REALIZA_VISITA]→  
  (v_contagiado:Visita)-[:ESTABLECIMIENTO]→(u:Ubicacion)-[:PARTE_DE]→(c:Ciudad)  
2 WITH u, c, count(v_contagiado) AS Num_visitas_contagiados  
3 MATCH (u)←[:ESTABLECIMIENTO]-(v:Visita)  
4 WITH u, c, Num_visitas_contagiados, count(v) AS Num_total_visitas  
5 WITH u.nombre AS Establecimiento,  
6   Num_visitas_contagiados,  
/  
    Num_total_visitas,  
8  toFloat(Num_visitas_contagiados) / Num_total_visitas * 100 AS  
  Porcentaje_contagiados,  
9   c.nombre AS Ciudad  
10 RETURN Establecimiento,  
11   Num_visitas_contagiados,  
12   Num_total_visitas,  
12   Num_total_visitas,  
13   toFloat(toInteger(Porcentaje_contagiados * 100)) / 100 AS  
  Porcentaje_visitas_personas_contagiadas,  
14   Ciudad  
15 ORDER BY Porcentaje_visitas_personas_contagiadas DESC
```

Establecimiento	Num_visitas_contagiados	Num_total_visitas	Porcentaje_visitas_personas_contagiadas	Ciudad
"Hospital Rio Hortega"	6	11	54.54	"Valladolid"
"Teatro Calderon"	6	13	46.15	"Valladolid"
"Rio Shopping"	5	13	38.46	"Valladolid"
"Colegio San José"	2	7	28.57	"Valladolid"
"Colegio Amor de Dios"	2	10	20.0	"Salamanca"
"Bar Comandante"	1	8	12.5	"Valladolid"
"Hospital Santisima Trinidad"	1	8	12.5	"Salamanca"
"Bar La Chica de Ayer"	1	11	9.09	"Salamanca"

En primer lugar, se seleccionan las personas contagiadas junto a las visitas que han realizado a los distintos establecimientos y las ciudades de éstos, y se hace un conteo del número de visitas de los contagiados a cada establecimiento. A continuación, se seleccionan todas las visitas a los distintos establecimientos y se realiza un conteo de éstas.

Teniendo el número de visitas de los contagiados y el número total de visitas por cada establecimiento, se puede calcular el porcentaje que las primeras representan sobre las segundas para cada establecimiento. Finalmente, retornamos toda la información previo redondeo (empleando el engaño comentado en un ejercicio anterior a causa de la imposibilidad de usar adecuadamente el `count()` en esta versión) del porcentaje calculado y ordenamos por porcentaje de visitas de personas contagiadas para poder comentar a continuación los establecimientos con

mayor y menor porcentaje.

Los dos establecimientos con mayor porcentaje de visitas de personas contagiadas respecto del total de visitas recibidas son Hospital Rio Ortega y Teatro Calderon, con 6 visitas de personas contagiadas cada uno sobre un total de 11 y 13 visitas respectivamente, lo cual deja unos porcentajes de 54.54% y 46.15%.

En contraposición, los dos establecimientos con menor porcentaje de visitas de personas contagiadas son Hospital Santisima Trinidad, que fue visitado por una única persona contagiada sobre un total de 8 visitantes totales (lo que deja un 12.5%), y Bar la Chica de Ayer, que, a pesar de recibir un total de hasta 11 visitas, tan sólo una estaba contagiada, lo que lo convierte con un 9.09% de visitas de personas contagiadas sobre el total de visitas como el establecimiento con menor porcentaje entre aquellos que recibieron al menos 1 visitante contagiado.

CASO 10: mostrar las distancias entre los domicilios de los contagiados de Valladolid que hayan ido a un mismo establecimiento aunque haya sido en diferentes fechas.

Hay que obtener el resultado mediante una sola consulta en el editor de Neo4J. Se quiere devolver solo los tres registros Persona1 – Persona2 CONTAGIADAS de Valladolid que viven a más distancia.

Importante: evitar obtener dos registros con la misma distancia y nodos intercambiados como este:

Persona_Sana_1	Persona_Sana_2	Ciudad	Distancia_domicilios_km
"Berta Romero"	"Sofia Vidores"	"Valladolid"	
"Sofia Vidores"	"Berta Romero"	"Valladolid"	

Nota: no es necesario que los contagiados hayan coincidido en tiempo en el mismo momento en el mismo establecimiento.

Devolver el nombre de ambas personas, la ciudad a la que pertenecen (que tendrá que ser Valladolid) y la distancia entre los domicilios de ambas personas contagiadas como se muestra en la imagen anterior.

Tener en cuenta que aproximadamente Valladolid tiene una distancia de unos 7,5 km de punta a punta, por lo que si alguna de las distancias es mayor de este valor, será indicativo de que hay algún tipo de error.

```
MATCH (p1:Persona {estado: "Contagiado"})-[:VISITA_EMPLAZAMIENTO]->(u:Ubicacion)-[:PARTE_DE]->(c:Ciudad {nombre: "Valladolid"})
```

```
MATCH (u)<-[[:VISITA_EMPLAZAMIENTO]]-(p2:Persona {estado: "Contagiado"})
```

```
WHERE p1.nombre < p2.nombre
```

```
WITH distinct p1, p2, c,
```

```
    round(distance(p1.ubicacion_domicilio, p2.ubicacion_domicilio) / 1000.0 * 100) / 100 AS
```

```
Distancia_domicilios_km
```

```
ORDER BY Distancia_domicilios_km DESC
```

```
LIMIT 3
```

```
RETURN p1.nombre AS Persona_Contagiada_1,
```

```
    p2.nombre AS Persona_Contagiada_2,
```

```
    c.nombre AS Ciudad,
```

```
    Distancia_domicilios_km
```

```

1 MATCH (p1:Persona {estado: "Contagiado"})-[:VISITA_EMPLAZAMIENTO]→(u:Ubicacion)-
  [:PARTE_DE]→(c:Ciudad {nombre: "Valladolid"})
2 MATCH (u)←[:VISITA_EMPLAZAMIENTO]-(p2:Persona {estado: "Contagiado"})
3 WHERE p1.nombre < p2.nombre
4 WITH distinct p1, p2, c,
5     round(distance(p1.ubicacion_domicilio, p2.ubicacion_domicilio) / 1000.0 * 100)
6 
7     / 100 AS Distancia_domicilios_km
8 ORDER BY Distancia_domicilios_km DESC
9 LIMIT 3
10
11 RETURN p1.nombre AS Persona_Contagiada_1,
12     p2.nombre AS Persona_Contagiada_2,
13     c.nombre AS Ciudad,
14     Distancia_domicilios_km

```

\$ MATCH (p1:Persona {estado: "Contagiado"})-[:VISITA_EMPLAZAMIENTO]→(u:Ubicacion)-[...]

Persona_Contagiada_1	Persona_Contagiada_2	Ciudad	Distancia_domicilios_km
"Berta Romero"	"Sofia Vidores"	"Valladolid"	2.22
"Berta Romero"	"Fidel Figueroa"	"Valladolid"	2.02
"Berta Romero"	"Marcelino Rodriguez"	"Valladolid"	1.67

En primer lugar se seleccionan las personas contagiadas que han visitado emplazamientos ubicados en Valladolid, y se vuelve a hacer una segunda vez para crear dos entes diferenciados. Establecemos una condición para que no se repita una combinación de nombres intercambiando las posiciones de columnas (es decir, que si ya existe la combinación A-B, no aparezca la combinación B-A de personas contagiadas).

Con un *distinct* evitamos además que aparezca una misma combinación repetida tantas veces como coincidencias entre dos personas haya en uno o distintos establecimientos (por lo que, añadido a la condición anterior, cada distancia entre contagiados coincidentes sólo se calculará una vez).

A continuación, calculamos la distancia entre los domicilios empleando *distance()*, que devuelve la distancia en metros entre dos puntos geográficos dadas sus latitudes y longitudes, dividimos entre 1000 para pasar a km y redondeamos manualmente a 2 decimales.

Finalmente, ordenamos por distancia de mayor a menor, limitamos la salida a 3 registros y retornamos los campos solicitados. Éstas serían las 3 mayores distancias entre domicilios de personas contagiadas que han visitado al menos un mismo establecimiento en Valladolid.