

Apuntes de Neo4j – Ampliación de Bases de Datos

...[contenido original conservado]...

9. Ejercicios adicionales estilo examen (ampliación)

9.1 Operaciones de creación y modificación

```
// Crear 3 personas con sus edades
CREATE (:Persona {nombre:'Ana', edad:30}),
      (:Persona {nombre:'Luis', edad:25}),
      (:Persona {nombre:'Marta', edad:40});

// Añadir campo 'activo' a todas las personas
MATCH (p:Persona)
SET p.activo = true;

// Eliminar campo 'activo'
MATCH (p:Persona)
REMOVE p.activo;
```

9.2 Relaciones y patrones

```
// Relación AMIGO entre personas mayores de 18
MATCH (a:Persona), (b:Persona)
WHERE a <> b AND a.edad > 18 AND b.edad > 18
MERGE (a)-[:AMIGO]->(b);

// Crear una relación bidireccional de colaboración
MATCH (a:Persona {nombre:'Ana'}), (b:Persona {nombre:'Luis'})
MERGE (a)-[:COLABORA]->(b);

// Crear relación con propiedades
MATCH (a:Persona {nombre:'Marta'}), (b:Persona {nombre:'Luis'})
CREATE (a)-[:CONOCIDO_DESDE {año:2010, evento:'Universidad'}]->(b);
```

9.3 Consultas con filtros y funciones

```
// Personas mayores de 30
MATCH (p:Persona)
WHERE p.edad > 30
RETURN p.nombre;
```

```
// Personas cuyo nombre empieza por 'A'
MATCH (p:Persona)
WHERE p.nombre STARTS WITH 'A'
RETURN p;

// Número de relaciones por persona
MATCH (p:Persona)-[r]->()
RETURN p.nombre, COUNT(r) AS numRelaciones;
```

9.4 Consultas con path y agregaciones

```
// Caminos hasta 3 saltos
MATCH path=(a:Ciudad {nombre:'Madrid'})-[:CARRETERA*1..3]->(b:Ciudad)
RETURN b.nombre, length(path) AS hops;

// Ruta más corta y coste total
MATCH p=shortestPath((a:Ciudad {nombre:'Madrid'})-[:CARRETERA*]->(b:Ciudad {nombre:'Barcelona'}))
RETURN p, reduce(c=0, rel IN relationships(p) | c + rel.coste) AS costeTotal;
```

9.5 Agregaciones avanzadas

```
// Media de edad por tipo de persona
MATCH (p:Persona)
RETURN labels(p)[0] AS tipo, AVG(p.edad) AS mediaEdad;

// Nodos más conectados
MATCH (n)-[r]-()
RETURN n.nombre AS entidad, COUNT(r) AS conexiones
ORDER BY conexiones DESC LIMIT 5;
```

9.6 Actualizaciones condicionales

```
// Si edad > 30 → añadir etiqueta 'Senior'
MATCH (p:Persona)
WHERE p.edad > 30
SET p:Senior;

// Eliminar nodos sin relaciones
MATCH (n)
WHERE NOT (n)--()
DELETE n;
```

9.7 Importaciones y APOC

```
// Cargar nodos desde CSV
LOAD CSV WITH HEADERS FROM 'file:///datos.csv' AS row
CREATE (:Ciudad {nombre: row.nombre, poblacion: toInteger(row.poblacion)});

// Crear relaciones desde CSV
LOAD CSV WITH HEADERS FROM 'file:///rutas.csv' AS row
MATCH (a:Ciudad {nombre: row.origen}), (b:Ciudad {nombre: row.destino})
CREATE (a)-[:CARRETERA {km: toInteger(row.km)}]->(b);

// Usar APOC para iterar y modificar
CALL apoc.periodic.iterate(
    'MATCH (n:Persona)',
    'SET n.hash = apoc.util.md5(n.nombre)',
    {batchSize:1000}
);
```

Estos ejemplos amplían aún más el espectro de preguntas que podrías encontrar en el examen: desde consultas con filtros y relaciones hasta importación de datos, uso de APOC y caminos con funciones de agregación. Prácticalos y adáptalos a tus casos reales o a los ejercicios propuestos por tu profesor.