1. Retrieve the names of members who enjoy thriller books.

Query:

MATCH (r:Reader)-[:LIKES]->(:Book)-[:IS\_GENRE]->(:Genre {name: 'thriller'})

RETURN r.name

Result:

| **r.name** |
| --- |
| **1** | "Anne HatsAway" |
| **2** | "Robert Bertoli" |
| **3** | "Wilfried Lemahieu" |
| **4** | "Bart Baesens" |
| **5** | "Wilfried Lemahieu" |
| **6** | "Elvis Presley" |
| **7** | "Seppe vanden Broucke" |

**(2)** Determine which book has been liked by the youngest reader.

Query:

MATCH (r:Reader)

WITH MIN(r.age) AS minAge

MATCH (r:Reader)-[:LIKES]->(b:Book)

WHERE r.age = minAge

RETURN b.title

| **b.title** |
| --- |
| **1** | "Where are my Keys?" |
| **2** | "Database Management" |
| **3** | "Laughing and Learning" |
| **4** | "A Wizardly Story" |

**(3)** Determine the genre most beloved by the club members.

Query:

MATCH (r:Reader)-[:LIKES]->(:Book)-[:IS\_GENRE]->(g:Genre)

RETURN g.name, COUNT(r) AS readerCount

ORDER BY readerCount DESC

LIMIT 1

Result:

| **g.name** | **readerCount** |
| --- | --- |
| **1** | "humor" | 19 |

**(4)** Identify the most popular book based on the number of likes.

Query:

MATCH (:Reader)-[l:LIKES]->(b:Book)

WITH b, COUNT(l) AS likeCount

ORDER BY likeCount DESC

LIMIT 1

RETURN b.title AS mostPopularBook, likeCount

Result:

| **mostPopularBook** | **likeCount** |
| --- | --- |
| **1** | **"Database Management"** | **5** |

**(5)** Find the common liked books between ‘Wilfried Lemahieu’ and his friends.

Query:

MATCH (wilfried:Reader {name: 'Wilfried Lemahieu'})-[:FRIEND\_OF]->(friend:Reader)-[:LIKES]->(book:Book)

WITH DISTINCT book AS commonLikedBook

MATCH (wilfried)-[:LIKES]->(commonLikedBook)

RETURN commonLikedBook.title AS CommonLikedBook

Result:

| **CommonLikedBook** |
| --- |
| **1** | "Where are my Keys?" |
| **2** | "Where are my Keys?" |
| **3** | "Where are my Keys?" |
| **4** | "Where are my Keys?" |
| **5** | "A Wizardly Story" |
| **6** | "A Wizardly Story" |
| **7** | "A Wizardly Story" |
| **8** | "Who Says Love is Outdated?" |
| **9** | "Who Says Love is Outdated?" |
| **10** | "Without you I am Nothing" |
| **11** | "Without you I am Nothing" |
| **12** | "Mathematics for the Rest of Us" |
| **13** | "Mathematics for the Rest of Us" |
| **14** | "A Chainsaw Massacre" |
| **15** | "An Unsolved Problem for Detective Whiskers" |
| **16** | "An Unsolved Problem for Detective Whiskers" |
| **17** | "An Unsolved Problem for Detective Whiskers" |

**6)** Calculate the average age of readers who like ‘Detective’ genre books.

Query:

MATCH (r:Reader)-[:LIKES]->(:Book)-[:IS\_GENRE]->(:Genre {name: 'detective'})

WITH r.age AS age

RETURN AVG(age) AS averageAge

Result:

| **averageAge** |
| --- |
| **1** | 35.0 |

**(7)** Find friends of 'Bart Baesens' who liked thriller books, ensuring no name is repeated in your results.

Query:

MATCH (bart:Reader {name: 'Bart Baesens'})-[:FRIEND\_OF]->(friend:Reader)-[:LIKES]->(:Book)-[:IS\_GENRE]->(:Genre {name: 'thriller'})

RETURN DISTINCT friend.name AS FriendName

Result:

| **FriendName** |
| --- |
| **1** | "Wilfried Lemahieu" |
| **2** | "Seppe vanden Broucke" |

**(8)** List the books that are liked by friends of 'Wilfried Lemahieu' but not by Wilfried himself.

Query:

MATCH (wilfried:Reader {name: 'Wilfried Lemahieu'})-[:FRIEND\_OF]->(friend:Reader)-[:LIKES]->(book:Book)

WHERE NOT (wilfried)-[:LIKES]->(book)

RETURN DISTINCT book.title AS LikedByFriendOnly

Result:

| **LikedByFriendOnly** |
| --- |
| **1** | "A Wizardly Story" |
| **2** | "Who Says Love is Outdated?" |
| **3** | "Without you I am Nothing" |
| **4** | "Mathematics for the Rest of Us" |
| **5** | "A Chainsaw Massacre" |
| **6** | "An Unsolved Problem for Detective Whiskers" |

**(9)** Determine which reader has the broadest taste in books, meaning they like the widest variety of genres.

Query:

MATCH (r:Reader)-[:LIKES]->(:Book)-[:IS\_GENRE]->(g:Genre)

WITH r, COUNT(DISTINCT g) AS genreCount

ORDER BY genreCount DESC

LIMIT 1

RETURN r.name AS ReaderWithBroadestTaste, genreCount AS NumberOfGenresLiked

Result:

| **ReaderWithBroadestTaste** | **NumberOfGenresLiked** |
| --- | --- |
| **1** | **"Robert Bertoli"** | 6 |

**(10)** Query for books which haven’t caught anyone’s interest yet. You may get no result for this query.

Query:

MATCH (b:Book)

WHERE NOT (:Reader)-[:LIKES]->(b)

RETURN b.title AS UnlikedBooks

Result:

(no changes, no records)