Query 1: Find all students who prefer living in clean places and their block information.

Natural Language Phrasing: Retrieve all students who prefer living in clean places, along with the blocks they live in.

```
SELECT S.name, B.letter AS BlockLetter, B.is_clean
FROM Students S
JOIN Prefers P ON S.student_id = P.student_id
JOIN Preferences PR ON P.preference_id = PR.preference_id
JOIN Lives_in L ON S.student_id = L.student_id
JOIN Blocks B ON L.block_id = B.block_id
WHERE PR.is_clean = TRUE;
```

Explanation:

- This query joins `Students`, `Prefers`, `Preferences`, `Lives_in`, and `Blocks` tables.
- It selects students who prefer clean places (`PR.is_clean = TRUE`) and retrieves the block they live in (`B.letter`).

Query 2: Find the number of students in each block, grouped by the cleanliness of the block.

Natural Language Phrasing: Count how many students live in each block, grouped by whether the block is clean or not.

```
SELECT B.letter AS BlockLetter, B.is_clean, COUNT(L.student_id) AS TotalStudents
FROM Blocks B
JOIN Lives_in L ON B.block_id = L.block_id
GROUP BY B.letter, B.is_clean;
```

Explanation:

- The query joins the `Blocks` and `Lives_in` tables to count how many students live in each block.
- The results are grouped by the block (`B.letter`) and the cleanliness status of the block (`B.is_clean`).

Query 3: Find the average distance students prefer for their social life preferences.

Natural Language Phrasing: Get the average preferred distance for students who have a social life preference.

```
SELECT AVG(PR.distance) AS AverageDistance
FROM Preferences PR
WHERE PR.social_life = TRUE;
```

Explanation:

- This query selects the average distance from the `Preferences` table for students who prefer social life (`PR.social_life = TRUE`).

Query 4: List blocks where the average friendliness level of blockmates is above 7.

Natural Language Phrasing: Show the blocks where the average friendliness level of the blockmates is greater than 7.

```
SELECT B.letter AS BlockLetter, AVG(BM.friendliness_level) AS AvgFriendliness
FROM Blocks B
JOIN Blockmate BM ON B.block_id = BM.block_id
GROUP BY B.letter
HAVING AVG(BM.friendliness_level) > 7;
```

Explanation:

- This query joins `Blocks` and `Blockmate`, then calculates the average friendliness level for each block (`AVG(BM.friendliness_level)`).
- The `HAVING` clause filters blocks where the average friendliness level is greater than 7.

Query 5: Find the total number of blocks managed by strict managers.

Natural Language Phrasing: Count how many blocks have a strict manager.

```
SELECT COUNT(DISTINCT W.college_id) AS TotalManagedBlocks
FROM Manager M
JOIN Works_for W ON M.manager_id = W.manager_id
WHERE M.is_strict = TRUE;
```

Explanation:

- This query counts the total number of blocks (`COUNT(DISTINCT W.college_id)`) that have strict managers (`M.is_strict = TRUE`).
- The query joins the `Manager` and `Works_for` tables to match managers with the colleges they manage.

Query 6: Find the most active block (highest activity level).

Natural Language Phrasing: Retrieve the block with the highest activity level.

```
SELECT B.letter AS BlockLetter, B.activity_level

FROM Blocks B

ORDER BY B.activity_level DESC

LIMIT 1;
```

- **Explanation:**
  - This query retrieves the block with the highest activity level by ordering blocks in descending order (`ORDER BY B.activity_level DESC`) and limiting the result to one row (`LIMIT 1`).

Query 7: List all colleges with more than 2 blocks and the total number of blocks they have.

Natural Language Phrasing: Find all colleges that have more than 2 blocks, along with the total number of blocks for each college.

```
SELECT C.name AS CollegeName, COUNT(B.block_id) AS TotalBlocks

FROM Colleges C

JOIN Blocks B ON C.college_id = B.college_id

GROUP BY C.name

HAVING COUNT(B.block_id) > 2;
```

Explanation:

- The query joins the `Colleges` and `Blocks` tables, counts the number of blocks per college (`COUNT(B.block_id)`), and filters colleges with more than 2 blocks using the `HAVING` clause.

Query 8: Find all managers who manage a college and also live in the same college block.

Natural Language Phrasing: Retrieve managers who both manage a college and live in a block within that college.

```
SELECT M.manager_u, S.name AS ManagerName, C.name AS CollegeName

FROM Manager M

JOIN Works_for WF ON M.manager_id = WF.manager_id

JOIN Colleges C ON WF.college_id = C.college_id

JOIN Lives_in L ON M.student_id = L.student_id

JOIN Blocks B ON L.block_id = B.block_id

WHERE B.college_id = WF.college_id;
```

Explanation:

- This query joins `Manager`, `Works_for`, `Colleges`, `Lives_in`, and `Blocks` to find managers who both manage a college and live in a block within that college (`B.college_id = WF.college_id`).

Query 9: Find the total number of students living in blocks that are not clean.

Natural Language Phrasing: Count how many students live in blocks that are not clean.

```sql
SELECT COUNT(L.student_id) AS TotalStudents

FROM Lives_in L

JOIN Blocks B ON L.block_id = B.block_id

WHERE B.is_clean = FALSE;
```

Explanation:

- This query counts the total number of students (`COUNT(L.student_id)`) who live in blocks that are not clean (`B.is_clean = FALSE`).