StudentsData Project - SQL Questions and Answers

Q1. Write an SQL query to add a primary key to the `student_id` column in the `students` table.

```
ALTER TABLE students ADD CONSTRAINT pk_student_id PRIMARY KEY (student_id);
```

Q2. Add a new column `email` to the `students` table.

```
ALTER TABLE students ADD email VARCHAR(100);
```

Q3. Insert a new student into the `students` table.

```
INSERT INTO students (student_id, age, gender, major) VALUES (100010, 23, 'Female',
'Engineering');
```

Q4. Update the major of the student with `student_id = 100002` to `'Psychology'`.

```
UPDATE students SET major = 'Psychology' WHERE student_id = 100002;
```

Q5. Delete all students older than 30.

```
DELETE FROM students WHERE age > 30;
```

Q6. Select all students majoring in `'Computer Science'`.

```
SELECT * FROM students WHERE major = 'Computer Science';
```

Q7. Find the average age of students grouped by gender.

```
SELECT gender, AVG(age) AS avg_age FROM students GROUP BY gender;
```

Q8. List all distinct majors in the student data.

```
SELECT DISTINCT major FROM students;
```

Q9. Join students and performance tables to display student_id, gender, and exam_score.

```
SELECT s.student_id, s.gender, p.exam_score FROM students s JOIN performance p ON
s.student_id = p.student_id;
```

Q10. Perform a LEFT JOIN to show all students and their habits (if available).

```
SELECT s.student_id, s.major, h.study_hours_per_day FROM students s LEFT JOIN habits
h ON s.student_id = h.student_id;
```

Q11. Count the number of students in each major.

```
SELECT major, COUNT(*) AS student_count FROM students GROUP BY major;
```

Q12. Find the maximum screen time for each study environment.

```
SELECT study_environment, MAX(screen_time) AS max_screen_time FROM habits GROUP BY
study_environment;
```

StudentsData Project - SQL Questions and Answers

Q13. Find students who are older than the average student age.

```
SELECT * FROM students WHERE age > (     SELECT AVG(age) FROM students );
```

Q14. Find the student(s) with the highest exam score.

```
SELECT * FROM performance WHERE exam_score = (       SELECT MAX(exam_score) FROM performance );
```

Q15. Add a UNIQUE constraint to the `email` column.

```
ALTER TABLE students ADD CONSTRAINT unique_email UNIQUE (email);
```

Q16. Add a FOREIGN KEY from `performance.student_id` to `students.student_id`.

```
ALTER TABLE performance ADD CONSTRAINT fk_student_id FOREIGN KEY (student_id) REFERENCES students(student_id);
```

Q17. Create a view for female students majoring in Business.

```
CREATE VIEW business_females AS SELECT * FROM students WHERE gender = 'Female' AND major = 'Business';
```

Q18. Insert a student and a performance record in a single transaction.

```
BEGIN;  INSERT INTO students (student_id, age, gender, major) VALUES (100011, 24, 'Male', 'Arts');  INSERT INTO performance (student_id, exam_score) VALUES (100011, 85);  COMMIT;
```

Q19. List all students from the `performance` table ordered by `exam_score` in descending order.

```
SELECT * FROM performance ORDER BY exam_score DESC;
```

Q20. Show all records from the `habits` table ordered by `study_hours_per_day` in ascending order.

```
SELECT * FROM habits ORDER BY study_hours_per_day ASC;
```

Q21. Display student IDs and screen time from `habits`, ordered by screen time (highest first), then by student ID.

```
SELECT student_id, screen_time FROM habits ORDER BY screen_time DESC, student_id;
```

Q22. Select student IDs and uppercase study environments from `habits`.

```
SELECT student_id, UPPER(study_environment) AS upper_env FROM habits;
```

Q23. Find students from the `habits` table whose study environment contains the word `'quiet'` (case-insensitive).

```
SELECT * FROM habits WHERE LOWER(study_environment) LIKE '%quiet%';
```

Q24. Show the first 3 characters of the study environment for each student in the `habits` table.

```
SELECT student_id, SUBSTRING(study_environment FROM 1 FOR 3) AS env_prefix FROM habits;
```

Q25. Concatenate student ID with exam score in the `performance` table (as a string).

```
SELECT student_id || ' scored ' || exam_score || ' marks' AS result_summary FROM performance;
```

Q26. Replace the word `'noisy'` with `'quiet'` in the study environment column of `habits`.

```
SELECT student_id, REPLACE(study_environment, 'noisy', 'quiet') AS improved_env FROM habits;
```