Last login: Fri Sep 26 13:02:54 on console

vasavimopidevi@Mac ~ % sqlite3 student.db

SQLite version 3.51.0 2025-06-12 13:14:41

Enter ".help" for usage hints.

sqlite> PRAGMA foreign\_keys = ON;

sqlite>

sqlite> CREATE TABLE students (

(x1...> id INTEGER PRIMARY KEY AUTOINCREMENT,

(x1...> name TEXT NOT NULL,

(x1...> age INTEGER,

(x1...> gender TEXT

(x1...> );

sqlite>

sqlite> CREATE TABLE courses (

(x1...> id INTEGER PRIMARY KEY AUTOINCREMENT,

(x1...> name TEXT NOT NULL

(x1...> );

sqlite>

sqlite> CREATE TABLE enrollments (

(x1...> id INTEGER PRIMARY KEY AUTOINCREMENT,

(x1...> student\_id INTEGER NOT NULL,

(x1...> course\_id INTEGER NOT NULL,

(x1...> marks INTEGER NOT NULL,

(x1...> FOREIGN KEY(student\_id) REFERENCES students(id),

(x1...> FOREIGN KEY(course\_id) REFERENCES courses(id)

(x1...> );

sqlite>

sqlite> .tables

courses enrollments students

sqlite> -- Insert Students

sqlite> INSERT INTO students (name, age, gender) VALUES

...> ('Alice',20,'F'),

...> ('Bob',21,'M'),

...> ('Carol',19,'F'),

...> ('David',22,'M'),

...> ('Eva',20,'F'),

...> ('Frank',23,'M'),

...> ('Grace',21,'F'),

...> ('Henry',20,'M'),

...> ('Irene',22,'F'),

...> ('Jack',19,'M');

sqlite>

sqlite> -- Insert Courses

sqlite> INSERT INTO courses (name) VALUES ('Math'),('Physics'),('English');

sqlite>

sqlite> -- Insert Enrollments

sqlite> INSERT INTO enrollments (student\_id, course\_id, marks) VALUES

...> (1,1,95),(1,2,88),(1,3,76),

...> (2,1,78),(2,2,82),(2,3,69),

...> (3,1,85),(3,2,90),(3,3,92),

...> (4,1,60),(4,2,55),(4,3,70),

...> (5,1,91),(5,2,89),(5,3,94),

...> (6,1,45),(6,2,50),(6,3,48),

...> (7,1,88),(7,2,84),(7,3,82),

...> (8,1,73),(8,2,77),(8,3,68),

...> (9,1,96),(9,2,94),(9,3,90),

...> (10,1,55),(10,2,60),(10,3,58);

sqlite>

sqlite> SELECT \* FROM students;

1|Alice|20|F

2|Bob|21|M

3|Carol|19|F

4|David|22|M

5|Eva|20|F

6|Frank|23|M

7|Grace|21|F

8|Henry|20|M

9|Irene|22|F

10|Jack|19|M

sqlite> SELECT \* FROM students;

1|Alice|20|F

2|Bob|21|M

3|Carol|19|F

4|David|22|M

5|Eva|20|F

6|Frank|23|M

7|Grace|21|F

8|Henry|20|M

9|Irene|22|F

10|Jack|19|M

sqlite> SELECT s.name AS student, c.name AS course, e.marks

...> FROM enrollments e

...> JOIN students s ON e.student\_id = s.id

...> JOIN courses c ON e.course\_id = c.id

...> ORDER BY s.id, c.id;

Alice|Math|95

Alice|Physics|88

Alice|English|76

Bob|Math|78

Bob|Physics|82

Bob|English|69

Carol|Math|85

Carol|Physics|90

Carol|English|92

David|Math|60

David|Physics|55

David|English|70

Eva|Math|91

Eva|Physics|89

Eva|English|94

Frank|Math|45

Frank|Physics|50

Frank|English|48

Grace|Math|88

Grace|Physics|84

Grace|English|82

Henry|Math|73

Henry|Physics|77

Henry|English|68

Irene|Math|96

Irene|Physics|94

Irene|English|90

Jack|Math|55

Jack|Physics|60

Jack|English|58

sqlite> SELECT s.name, ROUND(AVG(e.marks),2) AS avg\_marks

...> FROM students s

...> JOIN enrollments e ON s.id = e.student\_id

...> GROUP BY s.id

...> ORDER BY avg\_marks DESC

...> LIMIT 3;

Irene|93.33

Eva|91.33

Carol|89.0

sqlite> SELECT c.name AS course, ROUND(AVG(e.marks),2) AS avg\_marks

...> FROM courses c

...> JOIN enrollments e ON c.id = e.course\_id

...> GROUP BY c.id;

Math|76.6

Physics|76.9

English|74.7

sqlite> SELECT s.name, ROUND(AVG(e.marks),2) AS avg\_marks

...> FROM students s

...> JOIN enrollments e ON s.id = e.student\_id

...> GROUP BY s.id

...> HAVING AVG(e.marks) > 80;

Alice|86.33

Carol|89.0

Eva|91.33

Grace|84.67

Irene|93.33

sqlite> SELECT s.name, e.marks

...> FROM students s

...> JOIN enrollments e ON s.id = e.student\_id

...> WHERE e.course\_id = (SELECT id FROM courses WHERE name = 'Math')

...> AND e.marks > 80;

Alice|95

Carol|85

Eva|91

Grace|88

Irene|96

sqlite> BEGIN TRANSACTION;

sqlite>

sqlite> UPDATE enrollments

...> SET marks = 97

...> WHERE student\_id = 1 AND course\_id = 1;

sqlite>

sqlite> -- Check result

sqlite> SELECT \* FROM enrollments WHERE student\_id = 1;

1|1|1|97

2|1|2|88

3|1|3|76

sqlite>

sqlite> -- If correct:

sqlite> COMMIT;

sqlite>

sqlite> -- If not correct, you could instead run:

sqlite> -- ROLLBACK;

sqlite> SELECT s.name, c.name, e.marks

...> FROM enrollments e

...> JOIN students s ON e.student\_id = s.id

...> JOIN courses c ON e.course\_id = c.id

...> WHERE s.name = 'Alice' AND c.name = 'Math';

Alice|Math|97

sqlite> CREATE VIEW top\_students AS

...> SELECT s.id, s.name, ROUND(AVG(e.marks),2) AS avg\_marks

...> FROM students s

...> JOIN enrollments e ON s.id = e.student\_id

...> GROUP BY s.id

...> HAVING AVG(e.marks) >= 85;

sqlite> SELECT \* FROM top\_students ORDER BY avg\_marks DESC;

9|Irene|93.33

5|Eva|91.33

3|Carol|89.0

1|Alice|87.0

sqlite> CREATE INDEX idx\_students\_name ON students(name);

sqlite> EXPLAIN QUERY PLAN

...> SELECT \* FROM students WHERE name = 'Alice';

QUERY PLAN

`--SEARCH students USING INDEX idx\_students\_name (name=?)

sqlite>