

-- Create a table for patients

```
CREATE TABLE Patients (  
  pid INTEGER PRIMARY KEY,  
  name TEXT NOT NULL,  
  gender TEXT NOT NULL,  
  Phno INTEGER NOT NULL,  
  Email VARCHAR NOT NULL,  
  AppointmentID VARCHAR NOT NULL,  
  Prescription TEXT NOT NULL  
);
```

-- Insert some values into the Patients table

```
INSERT INTO Patients VALUES (1, 'Patchari', 'F', 124, '023@gmail.com', 'A123', 'ABCD');  
INSERT INTO Patients VALUES (2, 'Talluri', 'M', 1234, '123@gmail.com', 'B123', 'ABCDE');  
INSERT INTO Patients VALUES (3, 'Tarak', 'M', 12345, '12345@gmail.com', 'B1234',  
'ABCDEF');
```

-- Select patients with gender 'M'

```
--SELECT * FROM Patients WHERE gender = 'M';
```

-- Create a table for records with a foreign key constraint referencing the Patients table

```
CREATE TABLE Records (  
  Rid INTEGER PRIMARY KEY,  
  Pid INTEGER,  
  CONSTRAINT Records_FK FOREIGN KEY (Pid) REFERENCES Patients(pid)  
);
```

-- Insert some values into the Records table

```
INSERT INTO Records VALUES (1, 1);
```

```
INSERT INTO Records VALUES (2, 2);
```

```
-- Select records for a specific patient (e.g., pid=2)
```

```
--SELECT * FROM Records WHERE pid = 2;
```

```
-- Create table for diagnoses
```

```
CREATE TABLE diagnoses (
```

```
  Did INTEGER PRIMARY KEY,
```

```
  Medicine TEXT NOT NULL,
```

```
  Type TEXT NOT NULL,
```

```
  Pid INTEGER,
```

```
  CONSTRAINT diagnoses_Fk FOREIGN KEY (Pid) REFERENCES Patients (Pid)
```

```
);
```

```
-- Insert values into the diagnoses table
```

```
INSERT INTO diagnoses VALUES (1, 'dolo', 'generic', 1);
```

```
INSERT INTO diagnoses VALUES (2, 'synergic', 'fungal', 2);
```

```
INSERT INTO diagnoses VALUES (3, 'typhoid', 'chicken', 3);
```

```
-- Select from the diagnoses table where Did=2
```

```
--SELECT * FROM diagnoses WHERE Did = 2;
```

```
-- Update table diagnoses
```

```
UPDATE diagnoses
```

```
SET Type = 'antibiotic'
```

```
WHERE Did = 1;
```

```
--select * from diagnoses where Did=1;
```

```
-- Delete the Email column from the Patients table
```

```
ALTER TABLE Patients
```

```
DROP COLUMN Email;
```

```
--SELECT * FROM Patients WHERE gender = 'M';
```

```
--SELECT patients.pid, diagnoses.Did, patients.name, diagnoses.Medicine
```

```
--FROM patients
```

```
--INNER JOIN diagnoses
```

```
--ON patients.Pid = diagnoses.Pid;
```

```
CREATE INDEX idx_patients_gender ON Patients (gender);
```

myCompiler

mycode

SQL

```
9 Prescription TEXT NOT NULL
10 );
11
12 -- Insert some values into the Patients table
13 INSERT INTO Patients VALUES (1, 'Patchari', 'F', 124, '023@gmail.com', 'A123', 'ABCD');
14 INSERT INTO Patients VALUES (2, 'Talluri', 'M', 1234, '123@gmail.com', 'B123', 'ABCDE');
15 INSERT INTO Patients VALUES (3, 'Tarak', 'M', 12345, '12345@gmail.com', 'B1234', 'ABCDEF');
16
17 -- Select patients with gender 'M'
18 --SELECT * FROM Patients WHERE gender = 'M';
19
20
21 -- Create a table for records with a foreign key constraint referencing the Patients table
22 CREATE TABLE Records (
23   Rid INTEGER PRIMARY KEY,
24   Pid INTEGER,
25   CONSTRAINT Records_FK FOREIGN KEY (Pid) REFERENCES Patients(pid)
26 );
27
28 -- Insert some values into the Records table
29 INSERT INTO Records VALUES (1, 1);
30 INSERT INTO Records VALUES (2, 2);
31
32 -- Select records for a specific patient (e.g., pid=2)
33 SELECT * FROM Records WHERE pid = 2;
34
```

Output

```
2|2
[Execution complete with exit code 0]
```

Adobe Stock Get 10 Free Images From Adobe Stock. Start Now.

ADS VIA CARBON

30°C 22:04 21-04-2024

myCompiler

mycode

SQL

```
25 CONSTRAINT Records_FK FOREIGN KEY (Pid) REFERENCES Patients(pid)
26 );
27
28 -- Insert some values into the Records table
29 INSERT INTO Records VALUES (1, 1);
30 INSERT INTO Records VALUES (2, 2);
31
32 -- Select records for a specific patient (e.g., pid=2)
33 --SELECT * FROM Records WHERE pid = 2;
34
35 -- Create table for diagnoses
36 CREATE TABLE diagnoses (
37   Did INTEGER PRIMARY KEY,
38   Medicine TEXT NOT NULL,
39   Type TEXT NOT NULL,
40   Pid INTEGER,
41   CONSTRAINT diagnoses_Fk FOREIGN KEY (Pid) REFERENCES Patients (Pid)
42 );
43
44 -- Insert values into the diagnoses table
45 INSERT INTO diagnoses VALUES (1, 'dolo', 'generic', 1);
46 INSERT INTO diagnoses VALUES (2, 'synergic', 'fungal', 2);
47
48 -- Select from the diagnoses table where Did=2
```

Output

```
2|synergic|fungal|2
[Execution complete with exit code 0]
```

Adobe Stock Get 10 Free Images From Adobe Stock. Start Now.

Delete

myCompiler

English Recent Login Sig

mycode

SQL ⓘ

Run

Sa

```
37 Did INTEGER PRIMARY KEY,
38 Medicine TEXT NOT NULL,
39 Type TEXT NOT NULL,
40 Pid INTEGER,
41 CONSTRAINT diagnoses_Fk FOREIGN KEY (Pid) REFERENCES Patients (Pid)
42 );
43
44 -- Insert values into the diagnoses table
45 INSERT INTO diagnoses VALUES (1, 'dolo', 'generic', 1);
46 INSERT INTO diagnoses VALUES (2, 'synergic', 'fungal', 2);
47
48 -- Select from the diagnoses table where Did=2
49 --SELECT * FROM diagnoses WHERE Did = 2;
50 -- Update table diagnoses
51
52 UPDATE diagnoses
53 SET Type = 'antibiotic'
54 WHERE Did = 1;
55
56 --select * from diagnoses where Did=1;
57
58 -- Delete the Email column from the Patients table
59
60 ALTER TABLE Patients
61 DROP COLUMN Email;
62 SELECT * FROM Patients WHERE gender = 'M';
```

Output

2|Talluri|M|1234|8123|ABCDE
3|Tarak|M|12345|81234|ABCDEF

[Execution complete with exit code 0]

St Adobe Stock

Get 10 Free Images From Adobe Stock. Start Now.

ADS VIA CARBON

Type here to search

30°C

22:36

21-04-2024

Update

myCompiler

English Recent Login Sign up

mycode

SQL ⓘ

Run

Save

```
31
32 -- Select records for a specific patient (e.g., pid=2)
33 --SELECT * FROM Records WHERE pid = 2;
34
35 -- Create table for diagnoses
36 CREATE TABLE diagnoses (
37 Did INTEGER PRIMARY KEY,
38 Medicine TEXT NOT NULL,
39 Type TEXT NOT NULL,
40 Pid INTEGER,
41 CONSTRAINT diagnoses_Fk FOREIGN KEY (Pid) REFERENCES Patients (Pid)
42 );
43
44 -- Insert values into the diagnoses table
45 INSERT INTO diagnoses VALUES (1, 'dolo', 'generic', 1);
46 INSERT INTO diagnoses VALUES (2, 'synergic', 'fungal', 2);
47
48 -- Select from the diagnoses table where Did=2
49 --SELECT * FROM diagnoses WHERE Did = 2;
50 -- Update table diagnoses
51
52 UPDATE diagnoses
53 SET Type = 'antibiotic'
54 WHERE Did = 1;
55
56 select * from diagnoses where Did=1;
```

Output

1|dolo|antibiotic|1

[Execution complete with exit code 0]

St Adobe Stock

Get 10 Free Images From Adobe Stock. Start Now.

ADS VIA CARBON

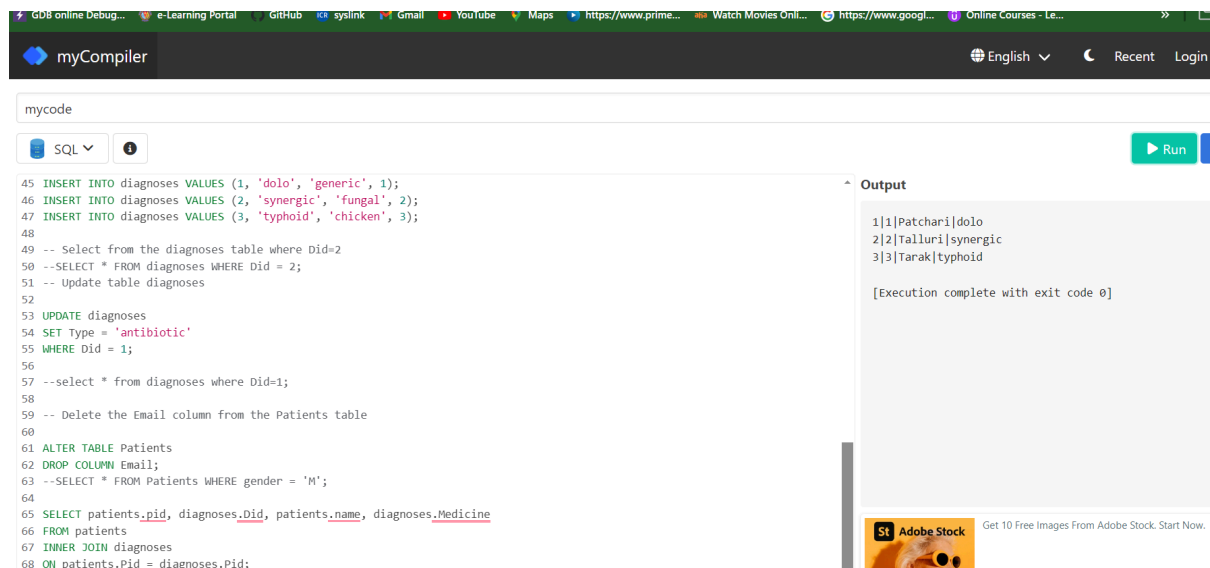
Type here to search

30°C

22:36

21-04-2024

Complex



The screenshot shows the myCompiler web IDE interface. The top navigation bar includes links to GitHub, syslink, Gmail, YouTube, Maps, and various search engines. The main editor area contains the following SQL code:

```
45 INSERT INTO diagnoses VALUES (1, 'dolo', 'generic', 1);
46 INSERT INTO diagnoses VALUES (2, 'synergic', 'fungal', 2);
47 INSERT INTO diagnoses VALUES (3, 'typhoid', 'chicken', 3);
48
49 -- Select from the diagnoses table where Did=2
50 --SELECT * FROM diagnoses WHERE Did = 2;
51 -- Update table diagnoses
52
53 UPDATE diagnoses
54 SET Type = 'antibiotic'
55 WHERE Did = 1;
56
57 --select * from diagnoses where Did=1;
58
59 -- Delete the Email column from the Patients table
60
61 ALTER TABLE Patients
62 DROP COLUMN Email;
63 --SELECT * FROM Patients WHERE gender = 'M';
64
65 SELECT patients.pid, diagnoses.Did, patients.name, diagnoses.Medicine
66 FROM patients
67 INNER JOIN diagnoses
68 ON patients.Pid = diagnoses.Pid;
```

The output panel on the right shows the results of the queries:

```
1|1|Patchari|dolo
2|2|Talluri|synergic
3|3|Tarak|typhoid

[Execution complete with exit code 0]
```

At the bottom of the output panel, there is an Adobe Stock advertisement.

Q5

Patients Table: Since the primary key column (`pid`) is already indexed by default due to its primary key constraint, there's no need to create an additional index for it. However, if you frequently query based on the `gender` column, you might consider indexing it.

```
CREATE INDEX idx_patients_gender ON Patients (gender);
```

Records Table: Since the `Pid` column in the `Records` table is a foreign key referencing the `pid` column in the `Patients` table, it's often automatically indexed for referential integrity purposes. However, if you frequently join or filter based on the `Pid` column, you may create an index explicitly.

```
CREATE INDEX idx_records_pid ON Records (Pid);
```

Diagnoses Table: Similar to the `Records` table, the `Pid` column in the `Diagnoses` table is a foreign key referencing the `pid` column in the `Patients` table. Therefore, it might already be indexed for referential integrity. If you often filter or join based on `Pid`, consider creating an index.

```
CREATE INDEX idx_diagnoses_pid ON diagnoses (Pid);
```

1. Other Indices: Depending on your query patterns and performance requirements, you may need to create additional indices for other columns that are frequently accessed or involved in join and filtering conditions.