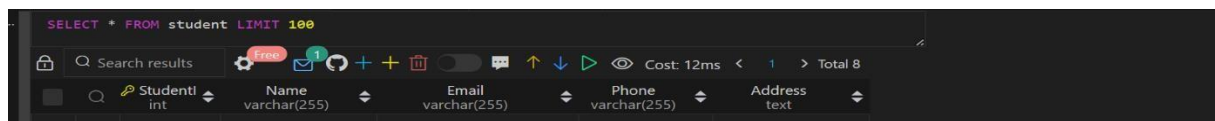


Assignment 3: Create tables

Table 1: Student Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_1="""
create table if not exists Student(
    StudentID int PRIMARY KEY,
    Name varchar(255),
    Email varchar(255),
    Phone varchar(255),
    Address text
) """
mycursor.execute(Table_1)
conn.commit()
conn.close()
```

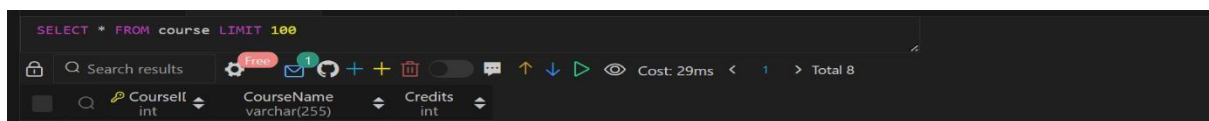


The screenshot shows a database query tool interface. At the top, the query is "SELECT * FROM student LIMIT 100". Below the query bar, there's a toolbar with various icons. The main area displays the table structure for 'student':

StudentID	Name	Email	Phone	Address
int	varchar(255)	varchar(255)	varchar(255)	text

Table 2: Course Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_2="""
create table if not exists Course(
    CourseID int PRIMARY KEY,
    CourseName varchar(255),
    Credits int
) """
mycursor.execute(Table_2)
conn.commit()
conn.close()
```

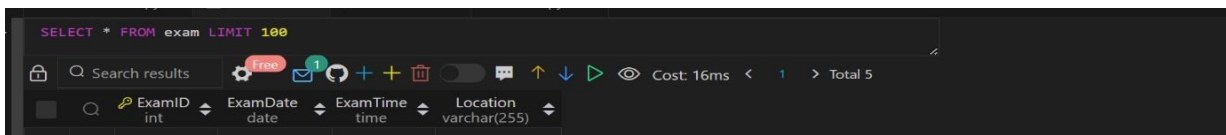


The screenshot shows a database query tool interface. At the top, the query is "SELECT * FROM course LIMIT 100". Below the query bar, there's a toolbar with various icons. The main area displays the table structure for 'course':

CourseID	CourseName	Credits
int	varchar(255)	int

Table 3: Exam Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_3="""
create table if not exists Exam(
    ExamID int PRIMARY KEY,
    ExamDate Date,
    ExamTime Time,
    Location varchar(255)
)
"""
mycursor.execute(Table_3)
conn.commit()
conn.close()
```

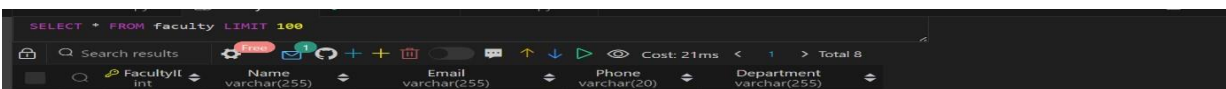


The screenshot shows a database query interface. At the top, the query is `SELECT * FROM exam LIMIT 100`. Below the query bar, there is a toolbar with various icons. The main area displays the table structure for 'exam' with columns: ExamID (int), ExamDate (date), ExamTime (time), and Location (varchar(255)). The interface also shows a 'Cost: 16ms' and 'Total 5' results.

ExamID	ExamDate	ExamTime	Location
--------	----------	----------	----------

Table 4: Faculty Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_4="""
create table if not exists Faculty(
    FacultyID int PRIMARY KEY,
    Name varchar(255),
    Email varchar(255),
    Phone varchar(20),
    Department varchar(255)
)
"""
mycursor.execute(Table_4)
conn.commit()
conn.close()
```



The screenshot shows a database query interface. At the top, the query is `SELECT * FROM faculty LIMIT 100`. Below the query bar, there is a toolbar with various icons. The main area displays the table structure for 'faculty' with columns: FacultyID (int), Name (varchar(255)), Email (varchar(255)), Phone (varchar(20)), and Department (varchar(255)). The interface also shows a 'Cost: 21ms' and 'Total 8' results.

FacultyID	Name	Email	Phone	Department
-----------	------	-------	-------	------------

Table 5: Enrolment Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_5="""
create table if not exists Faculty(
    FacultyID int PRIMARY KEY,
    Name varchar(255),
    Email varchar(255),
    Phone varchar(20),
    Department varchar(255)
)
"""
mycursor.execute(Table_5)
conn.commit()
conn.close()
```

Table 6: Teaching Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_6="""
create table if not exists Teaching(
    TeachingID int PRIMARY KEY,
    FacultyID int,
    CourseID int,
    Foreign key(FacultyID) references Faculty(FacultyID),
    Foreign key(CourseID) references Course(CourseID)
) """
mycursor.execute(Table_4)
conn.commit()
conn.close()
```

SELECT * FROM teaching LIMIT 100

Search results

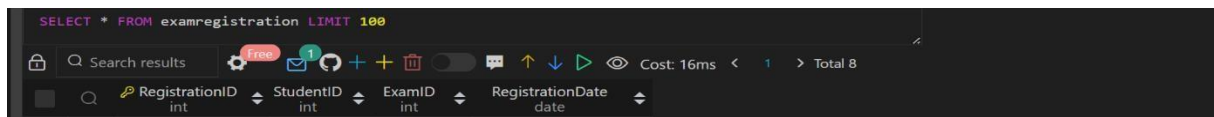
Free 1

Cost: 11ms < 1 > Total 8

TeachingID int FacultyID int CourseID int

Table 7: Exam Registration Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_7="""
create table if not exists ExamRegistration(
    RegistrationID int PRIMARY KEY,
    StudentID int,
    ExamID int,
    RegistrationDate Date,
    Foreign key(StudentID) references Student(StudentID),
    Foreign key(ExamID) references Exam(ExamID)
) """
mycursor.execute(Table_4)
conn.commit()
conn.close()
```

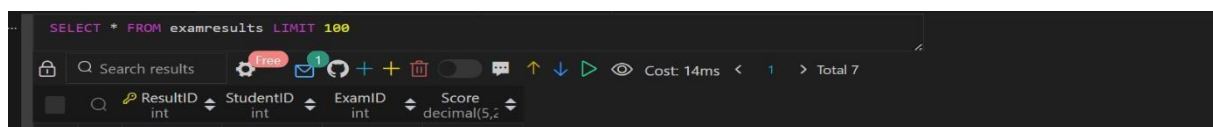


The screenshot shows a database query result for the 'examregistration' table. The query is 'SELECT * FROM examregistration LIMIT 100'. The result is displayed in a table with columns: RegistrationID (int), StudentID (int), ExamID (int), and RegistrationDate (date). The table has 8 rows in total, with the first row highlighted. The interface includes a search bar, a 'Free' button, and a 'Total 8' indicator.

RegistrationID	StudentID	ExamID	RegistrationDate
1	1	1	2023-10-27
2	1	2	2023-10-27
3	1	3	2023-10-27
4	1	4	2023-10-27
5	1	5	2023-10-27
6	1	6	2023-10-27
7	1	7	2023-10-27
8	1	8	2023-10-27

Table 8: Exam Results Table

```
import mysql.connector conn =
mysql.connector.connect(host='localhost', password='123456789',
user='root', database = "db") mycursor=conn.cursor()
Table_8="""
create table if not exists ExamResults(
    ResultID int PRIMARY KEY,
    StudentID int,
    ExamID int,
    Score Decimal(5,2),
    Foreign key(StudentID) references Student(StudentID),
    Foreign key(ExamID) references Exam(ExamID)
) """
mycursor.execute(Table_4)
conn.commit()
conn.close()
```



The screenshot shows a database query result for the 'examresults' table. The query is 'SELECT * FROM examresults LIMIT 100'. The result is displayed in a table with columns: ResultID (int), StudentID (int), ExamID (int), and Score (decimal(5,2)). The table has 7 rows in total, with the first row highlighted. The interface includes a search bar, a 'Free' button, and a 'Total 7' indicator.

ResultID	StudentID	ExamID	Score
1	1	1	85.50
2	1	2	78.25
3	1	3	92.10
4	1	4	65.75
5	1	5	88.90
6	1	6	72.30
7	1	7	81.45