

# Design DB model for Guvi Zen class:

## Users:

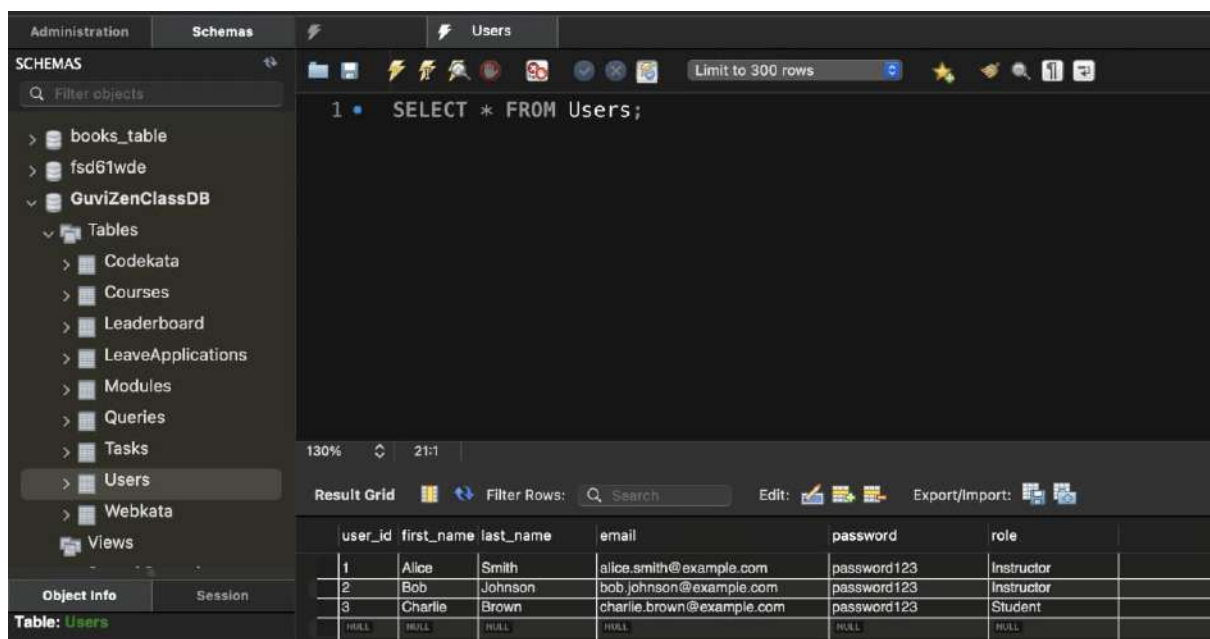
---

```
create table users (  
  user_id int not null auto_increment,  
  first_name varchar(50) not null,  
  last_name varchar (50) not null,  
  email varchar (100) not null unique,  
  password varchar(255) not null,  
  role varchar(20) not null,  
  primary key (user_id)  
);
```

```
insert into users (first_name, last_name, email, password, role)  
values  
( 'Alice', 'Smith', 'alice.smith@example.com', 'password123', 'Instructor'),  
( 'Bob', 'Johnson', 'bob.johnson@example.com', 'password123', 'Instructor'),  
( 'Charlie', 'Brown', 'charlie.brown@example.com', 'password123', 'Student');
```

## Output:

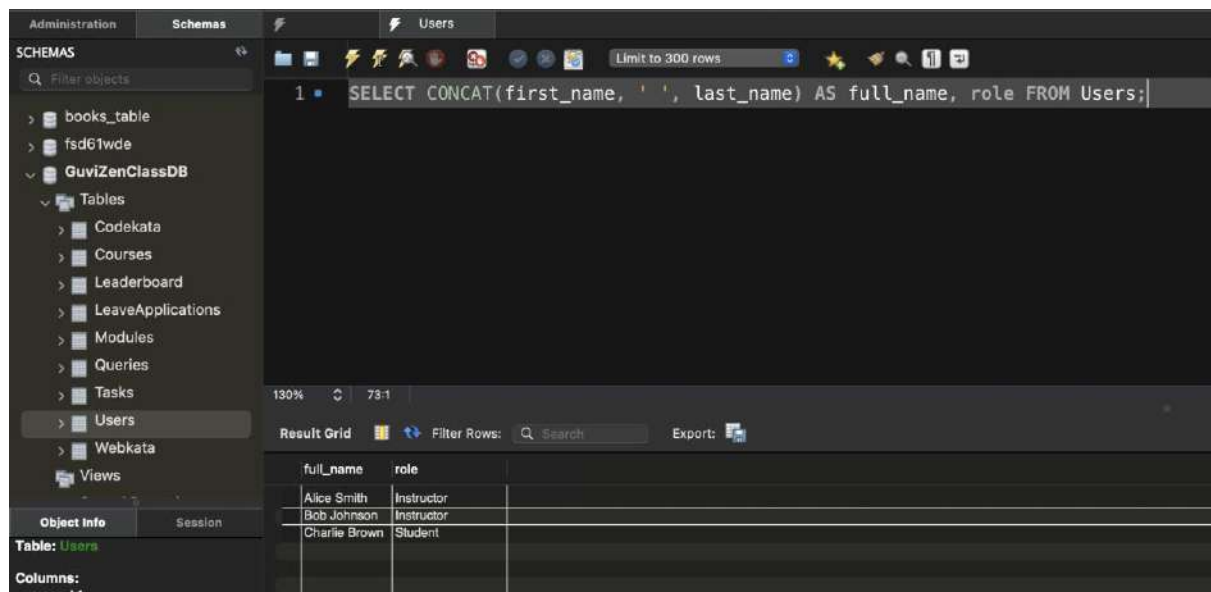
---



user_id	first_name	last_name	email	password	role
1	Alice	Smith	alice.smith@example.com	password123	Instructor
2	Bob	Johnson	bob.johnson@example.com	password123	Instructor
3	Charlie	Brown	charlie.brown@example.com	password123	Student
NULL	NULL	NULL	NULL	NULL	NULL

*List all users, showing their full name and role from users table ?*

---



## Courses:

---

```
create table courses (  
  course_id int not null auto_increment,  
  course_name varchar(100) not null,  
  course_description text,  
  start_date date not null,  
  end_date date not null,  
  instructor_id int not null,  
  primary key (course_id),  
  foreign key (instructor_id) references users(user_id)  
);
```

```
insert into courses (course_name, course_description, start_date, end_date,  
instructor_id)
```

values

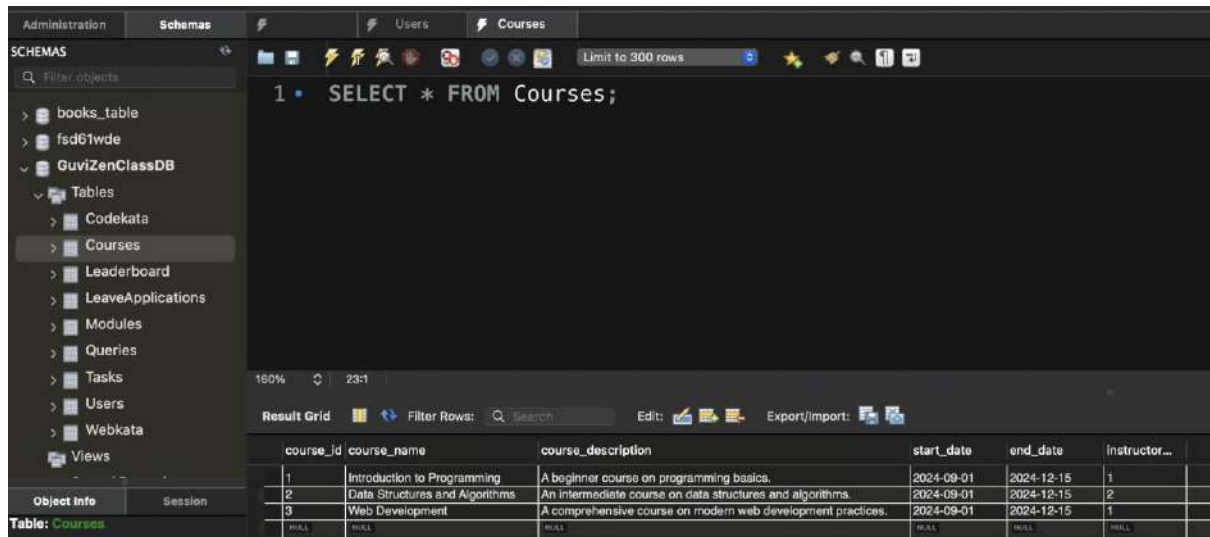
```
('Introduction to Programming', 'A beginner course on programming basics.',  
'2024-09-01', '2024-12-15', 1),
```

```
('Data Structures and Algorithms', 'An intermediate course on data structures  
and algorithms.', '2024-09-01', '2024-12-15', 2),
```

```
('Web Development', 'A comprehensive course on modern web development  
practices.', '2024-09-01', '2024-12-15', 1);
```

## Output:

---

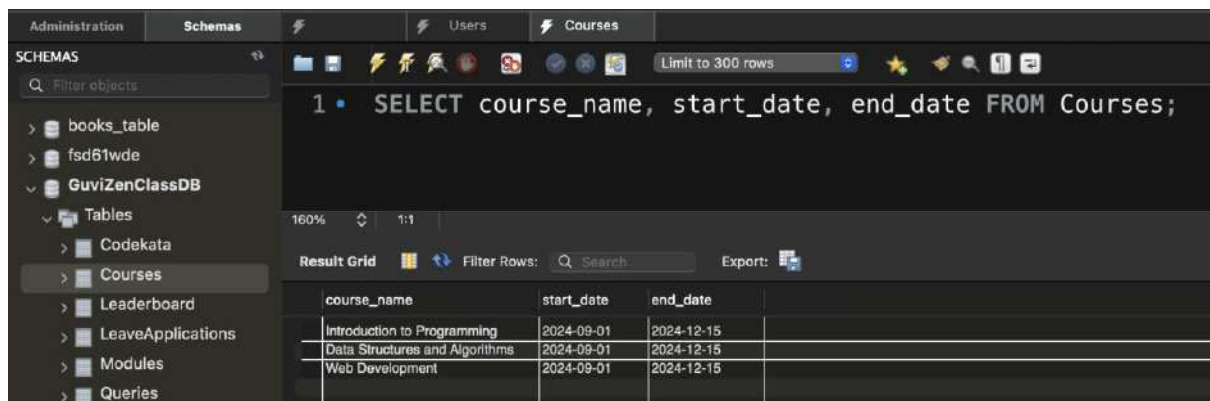


The screenshot shows a database management interface with a sidebar on the left listing schemas and tables. The main area displays a query: `1 • SELECT * FROM Courses;`. Below the query, a 'Result Grid' shows the data for the 'Courses' table. The table has columns: `course_id`, `course_name`, `course_description`, `start_date`, `end_date`, and `instructor...`. The data is as follows:

course_id	course_name	course_description	start_date	end_date	instructor...
1	Introduction to Programming	A beginner course on programming basics.	2024-09-01	2024-12-15	1
2	Data Structures and Algorithms	An intermediate course on data structures and algorithms.	2024-09-01	2024-12-15	2
3	Web Development	A comprehensive course on modern web development practices.	2024-09-01	2024-12-15	1
NULL	NULL	NULL	NULL	NULL	NULL

*List all courses along with start and end date ?*

---



The screenshot shows the same database management interface, but the query is now: `1 • SELECT course_name, start_date, end_date FROM Courses;`. The 'Result Grid' shows the data for the 'Courses' table, filtered to show only the columns `course_name`, `start_date`, and `end_date`. The data is as follows:

course_name	start_date	end_date
Introduction to Programming	2024-09-01	2024-12-15
Data Structures and Algorithms	2024-09-01	2024-12-15
Web Development	2024-09-01	2024-12-15

## Modules:

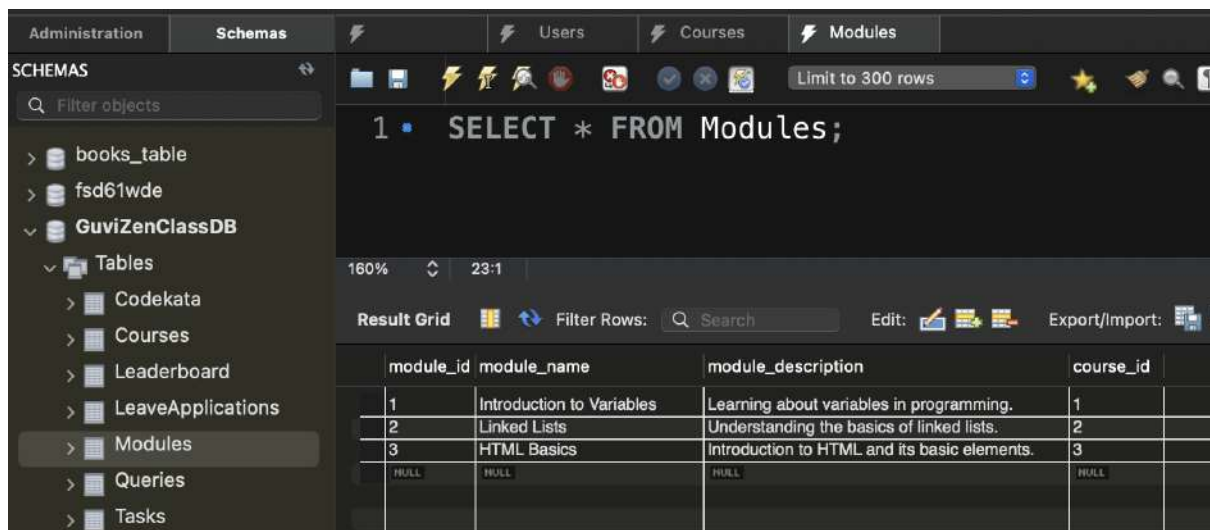
---

create table modules (  
  module\_id int not null auto\_increment,  
  module\_name varchar(100) not null,  
  module\_description text,  
  course\_id int not null,  
  primary key (module\_id),  
  foreign key (course\_id) references courses(course\_id)  
);

insert into modules (module\_name, module\_description, course\_id)  
values  
( 'Introduction to Variables', 'Learning about variables in programming.', 1),  
( 'Linked Lists', 'Understanding the basics of linked lists.', 2),  
( 'HTML Basics', 'Introduction to HTML and its basic elements.', 3);

*Output:*

---

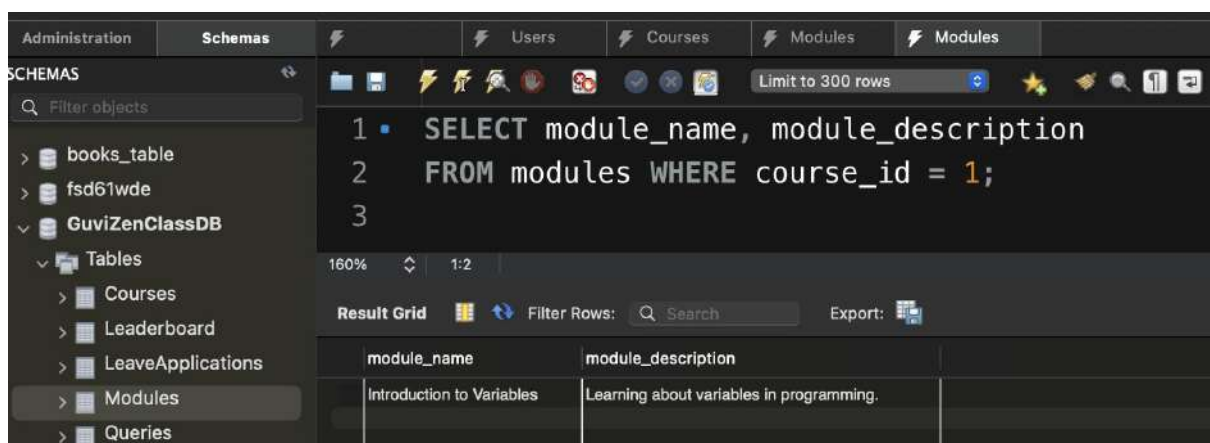


The screenshot shows a database management interface with a sidebar on the left containing a tree view of schemas. The 'Schemas' tab is active, and the 'GuviZenClassDB' schema is selected. The main area displays a SQL query: `1 SELECT * FROM Modules;`. Below the query, the 'Result Grid' shows the results of the query. The grid has four columns: `module_id`, `module_name`, `module_description`, and `course_id`. The results are as follows:

module_id	module_name	module_description	course_id
1	Introduction to Variables	Learning about variables in programming.	1
2	Linked Lists	Understanding the basics of linked lists.	2
3	HTML Basics	Introduction to HTML and its basic elements.	3
HULL	HULL	HULL	HULL

*Module along with description for specific course identify by the course\_id:*

---



The screenshot shows the same database management interface. The SQL query is now: `1 SELECT module_name, module_description  
2 FROM modules WHERE course_id = 1;  
3`. The 'Result Grid' shows the results of this query. The grid has two columns: `module_name` and `module_description`. The results are as follows:

module_name	module_description
Introduction to Variables	Learning about variables in programming.

## Tasks:

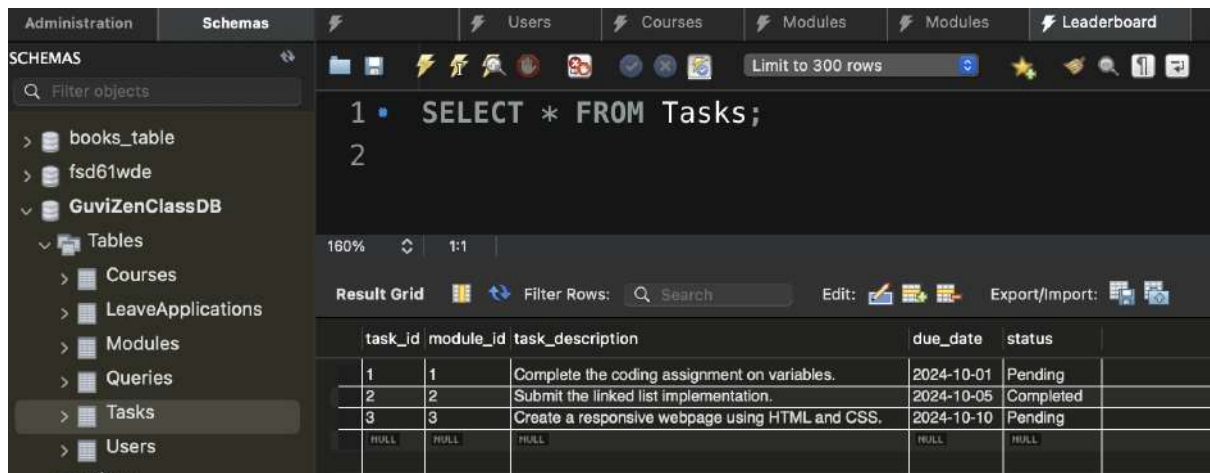
---

```
create table tasks (  
  task_id int not null auto_increment,  
  module_id int not null,  
  task_description text,  
  due_date date,  
  status varchar(20),  
  primary key (task_id),  
  foreign key (module_id) references modules(module_id)  
);
```

```
insert into tasks (module_id, task_description, due_date, status) values  
(1, 'complete the coding assignment on variables.', '2024-10-01', 'pending'),  
(2, 'submit the linked list implementation.', '2024-10-05', 'completed'),  
(3, 'create a responsive webpage using html and css.', '2024-10-10', 'pending');
```

## Output:

---

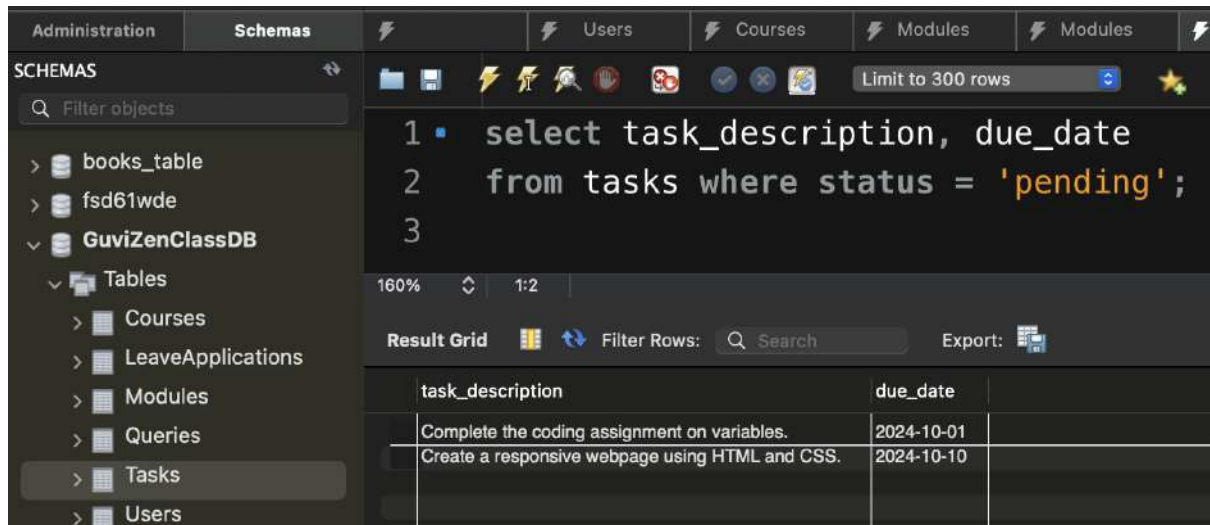


The screenshot shows a database management interface with a sidebar on the left displaying a schema tree. The 'Tasks' table is selected under the 'GuviZenClassDB' database. The main area shows the SQL query 'SELECT \* FROM Tasks;' and the resulting data grid.

task_id	module_id	task_description	due_date	status
1	1	Complete the coding assignment on variables.	2024-10-01	Pending
2	2	Submit the linked list implementation.	2024-10-05	Completed
3	3	Create a responsive webpage using HTML and CSS.	2024-10-10	Pending

*List all tasks that are currently pending along with their due dates:*

---



The screenshot shows a database management interface with a sidebar on the left containing a tree view of schemas. The 'Schemas' tab is active, and the 'GuvizZenClassDB' schema is selected. Under 'Tables', the 'Tasks' table is highlighted. The main area displays a SQL query: `select task_description, due_date from tasks where status = 'pending';`. Below the query, the 'Result Grid' shows two rows of data. The first row has a task description 'Complete the coding assignment on variables.' and a due date '2024-10-01'. The second row has a task description 'Create a responsive webpage using HTML and CSS.' and a due date '2024-10-10'.

task_description	due_date
Complete the coding assignment on variables.	2024-10-01
Create a responsive webpage using HTML and CSS.	2024-10-10

## Queries:

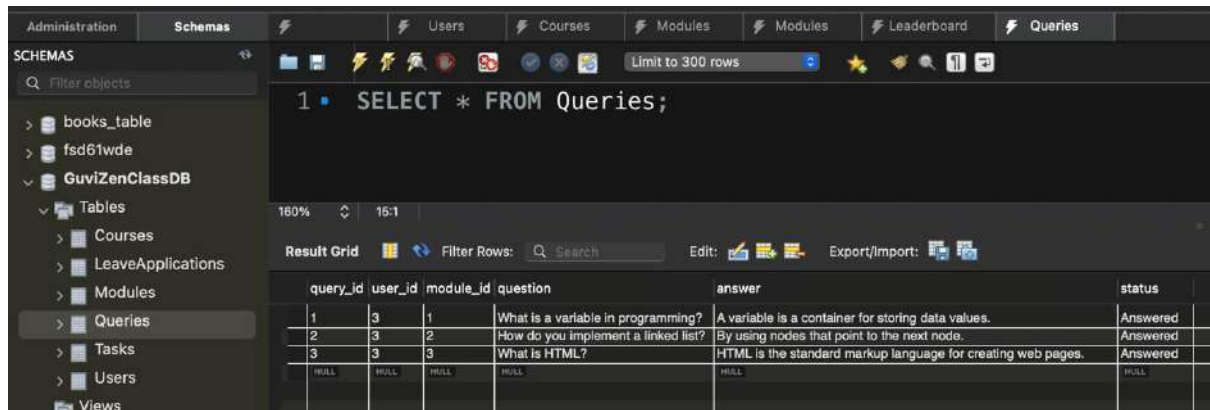
---

```
create table queries (  
  query_id int not null auto_increment,  
  user_id int not null,  
  module_id int not null,  
  question text,  
  answer text,  
  status varchar(20),  
  primary key (query_id),  
  foreign key (user_id) references users(user_id),  
  foreign key (module_id) references modules(module_id)  
);
```

```
insert into queries (user_id, module_id, question, answer, status) values  
(3, 1, 'what is a variable in programming?', 'a variable is a container for storing  
data values.', 'answered'),  
(3, 2, 'how do you implement a linked list?', 'by using nodes that point to the  
next node.', 'answered'),  
(3, 3, 'what is html?', 'html is the standard markup language for creating web  
pages.', 'answered');
```

## Output:

---

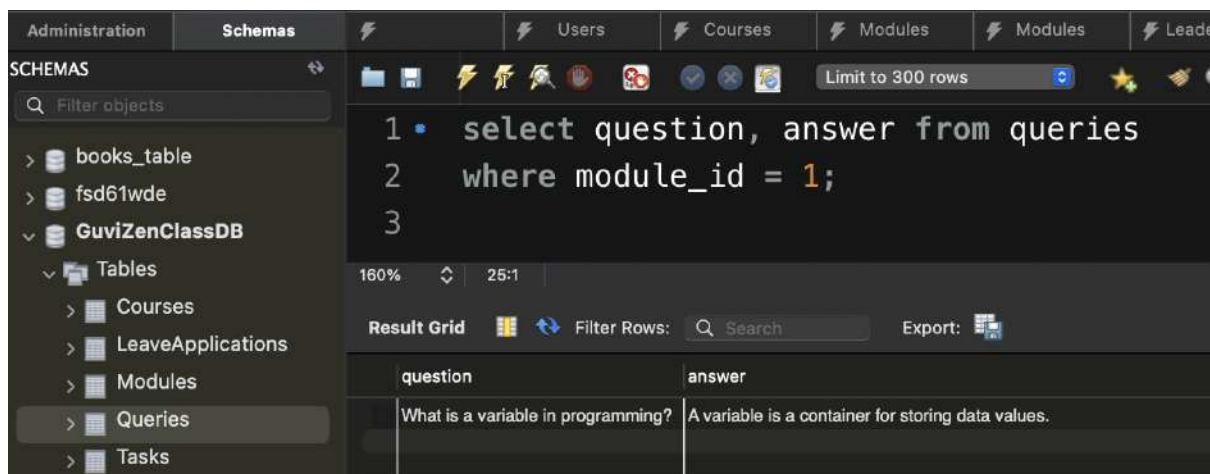


The screenshot shows a database management interface with a sidebar on the left containing a tree view of schemas. The main area displays a query editor with the text "1 • SELECT \* FROM Queries;". Below the editor, a "Result Grid" shows the results of the query. The grid has columns for query\_id, user\_id, module\_id, question, answer, and status. The first three rows of data are visible, showing questions about programming variables, linked lists, and HTML.

query_id	user_id	module_id	question	answer	status
1	3	1	What is a variable in programming?	A variable is a container for storing data values.	Answered
2	3	2	How do you implement a linked list?	By using nodes that point to the next node.	Answered
3	3	3	What is HTML?	HTML is the standard markup language for creating web pages.	Answered

*Retrieve all the questions and their corresponding answers for a specific module:*

---



The screenshot shows the same database management interface. The query editor now contains the text "1 • select question, answer from queries where module\_id = 1;". The "Result Grid" shows the results of this filtered query, displaying only the first row of data where module\_id is 1.

question	answer
What is a variable in programming?	A variable is a container for storing data values.

## LeaveApplication:

---

```
create table leaveapplications (  
  leave_id int not null auto_increment,  
  user_id int not null,  
  start_date date not null,  
  end_date date not null,  
  reason text,  
  status varchar(20) default 'pending',  
  primary key (leave_id),
```



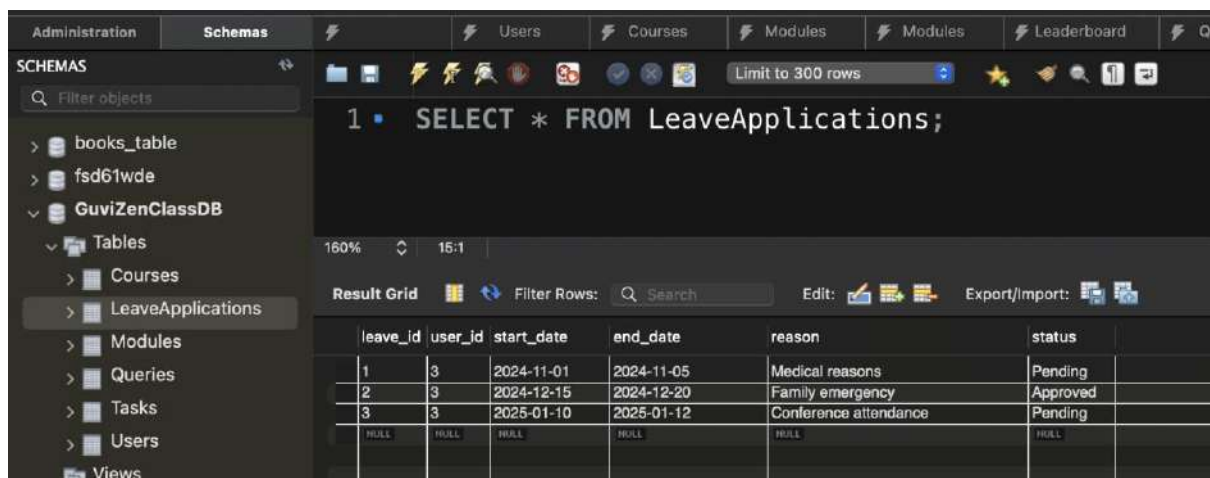
foreign key (user\_id) references users(user\_id)  
);

insert into leaveapplications (user\_id, start\_date, end\_date, reason, status)  
values

(3, '2024-11-01', '2024-11-05', 'medical reasons', 'pending'),  
(3, '2024-12-15', '2024-12-20', 'family emergency', 'approved'),  
(3, '2025-01-10', '2025-01-12', 'conference attendance', 'pending');

*Output:*

---

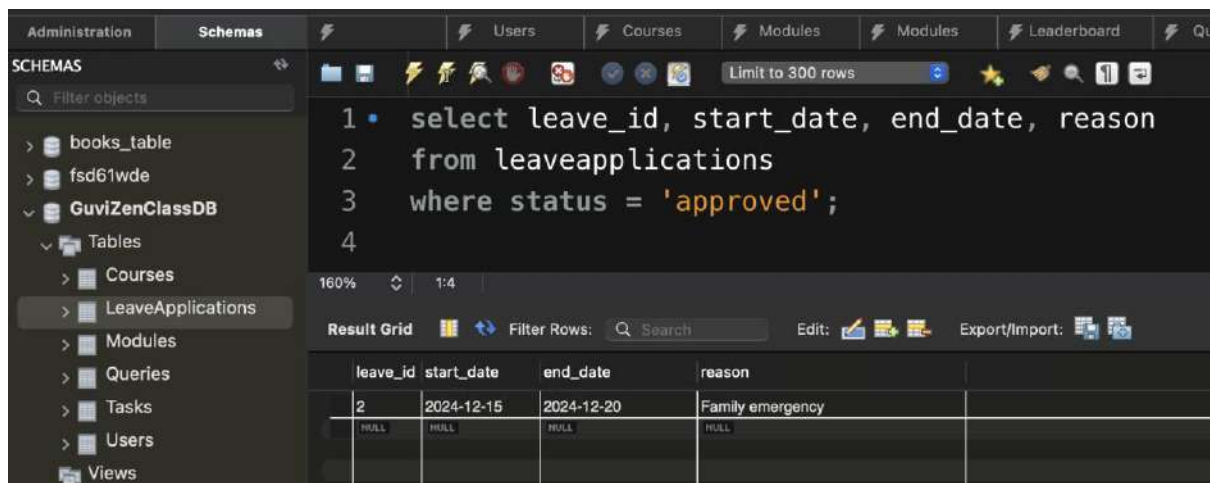


1 • `SELECT * FROM LeaveApplications;`

leave_id	user_id	start_date	end_date	reason	status
1	3	2024-11-01	2024-11-05	Medical reasons	Pending
2	3	2024-12-15	2024-12-20	Family emergency	Approved
3	3	2025-01-10	2025-01-12	Conference attendance	Pending

*List all leave applications that have been approved:*

---



1 • `select leave_id, start_date, end_date, reason`  
2 `from leaveapplications`  
3 `where status = 'approved';`  
4

leave_id	start_date	end_date	reason
2	2024-12-15	2024-12-20	Family emergency



## // Join Clause

List all approved leave applications along with the full name of the user who applied for the leave?

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' panel displays a tree view of the database structure, including tables like 'Users' and 'LeaveApplications'. The main editor displays a SQL query:

```
1 use GuviZenClassDB;
2
3 select
4     u.first_name,
5     u.last_name,
6     l.start_date,
7     l.end_date
8 from
9     leaveapplications l
10 join
11     users u
12 on
13     l.user_id = u.user_id
14 where
15     l.status = 'approved';
16
```

Below the query editor, the 'Result Grid' shows the results of the query. The columns are 'first\_name', 'last\_name', 'start\_date', and 'end\_date'. The results are:

first_name	last_name	start_date	end_date
Charlie	Brown	2024-12-15	2024-12-20

List all modules for each course, showing the course name and module name ?

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' panel displays a tree view of the database structure, including tables like 'Courses' and 'Modules'. The main editor displays a SQL query:

```
1 use GuviZenClassDB;
2
3 select
4     c.course_name,
5     m.module_name
6 from
7     courses c
8 join
9     modules m
10 on
11     c.course_id = m.course_id;
12
```

Below the query editor, the 'Result Grid' shows the results of the query. The columns are 'course\_name' and 'module\_name'. The results are:

course_name	module_name
Introduction to Programming	Introduction to Variables
Data Structures and Algorithms	Linked Lists
Web Development	HTML Basics