

THE SRC CONSULTATION PRESENTATION TEAM 5



Andrew Rochford
Rita Sakha

NARRATIVE

- SR Consulting (SRC) is a firm that creates custom software for its clients. When the client goes searching for a job, the manager at SRC will decide the staffing needs, responsibilities, and required skills for the consultant. Eventually, they will find a consultant who is available with the required skills and assign them to a project. When a consultant is assigned to become a project manager. When the project is done, the project manager will evaluate the work of the consultants and the consultants will evaluate the project manager. Projects can be subdivided into many projects.



BUSINESS RULES:

THE SRC CONSULTATION
PRESENTATION

- Consultants may receive more than one evaluation.
- Sometimes a consultant does not receive an evaluation, only because they may have not been assigned to a project.
- A consultant may be assigned to one or more projects, or they can be assigned to no projects.
- A project may be assigned to one or more consultants.
- Only one consultant can be assigned as a manager.
- Only one manager can manage each project.
- A project can be subdivided into sub-projects.
- One or more projects can belong to one client
- A project can require more than one skill.
- A manager may have one or more skills for management.

• + COMPONENTS OF ERD • +

Components of the ERD

Entity	Relationship	Connectivity	Entity
Assignments	Tasked to	M:1	Project
Certification	Gains	M:1	Skills
Consultant	Assists with	1:M	Assignments
Consultant	Recieves	1:M	Certification
Evaluation	Performed by	M:1	Consultant
Manager	Appraises	1:M	Evaluation
Manager	Determines	1:M	Assignments
Manager	Handles	1:M	Project
Project	Contains	1:M	SubProject
Project	Associates	M:1	Client
Project Requirement	Belongs to	M:1	Project

Components of ERD

• + ENTITY DESCRIPTIONS • +

ENTITY: CONSULTANT

+

○

THE SRC CONSULTATION
PRESENTATION

- This entity consists of all the information about each consultant employed at the SR Consulting Firm. These attributes include the consultant's ID as a primary key with their attributes of full name, address, city, state, and contact information.

ENTITY: CLIENT

- The client entity holds the relevant information relating to each client. This includes the Client ID which is a primary key and the attributes for client will be the contact name and contact phone.

ENTITY: PROJECT

- The project entity consists of each project the SRC firm has worked on or is currently working on. The primary key that is in use right now is the project ID number. This entity few foreign keys. The foreign keys founded in project will be the project manager ID, Parent project ID, and the Client ID. The client ID is also from the client entity. This entity also maintains the project name.

ENTITY: SKILLS

- For skills there is only one primary key and the primary key is the Skill ID. The attribute for skill would be the skill contact name. In this firm every client will show us their skills.

ENTITY: CONSULTANT PROJECT HISTORY



THE SRC CONSULTATION
PRESENTATION

- For the consultant project history, you will find three primary keys, consultant ID, project ID, and evaluator ID. The consultant history in the SRC firm will have projects, and which each project will have a start and end date, with the total hours worked.

ENTITY: EVALUATION

+

○

THE SRC CONSULTATION
PRESENTATION

- For the evaluation, the primary key is an evaluator ID. At SRC after completing a project we will have an evaluation for each client, including management. In the evaluation there will be a date and comment for everyone to fill out.

ENTITY: MANAGEMENT

+

○

THE SRC CONSULTATION
PRESENTATION

- For the management, the primary key is the Project Manager ID. At SRC we have the parent project ID which contains a subprojects member, user groups, and roles with their corresponding permissions which is also derived. Also, a subproject can receive project members and their roles from the parent project.



DATA DICTIONARY

Data Dictionary

Data Dictionary of ERD

Table Name	Attributes	Contents	Type	Format	PK or FK
Evaluation	Evaluator_ID	Evaluation ID	CHAR(3)	123	PK
	Eval_Date	Date of Evaluation	DATE	DD-MM-YY	
	Eval_Comment	Comments about Evaluation	VARCHAR(50)	Lorem Ipsum	
Management	Project_Manager_ID	Project Manager ID	VARCHAR(3)	123	PK
	Parent_Project_ID	Parent Project ID	VARCHAR(3)	123	PK
Consultant	Consultant_ID	Consultant ID	VARCHAR(10)	ABCDE12345	PK
	Project_ID	Project ID	CHAR(3)	123	FK
	Skill_ID	Skill ID	CHAR(1)	1	FK
	Consultant_FName	First Name of Consultant	VARCHAR(15)	John Doe	
	Consultant_LName	Last Name of Consultant	VARCHAR(15)	John Doe	
	Consultant_Address	Address of Consultant	VARCHAR(35)	Lorem Ipsum	
	Consultant_City	City of Consultant	VARCHAR(15)	Lorem	
	Consultant_State	State of Consultant	VARCHAR(2)	AB	
	Consultant_PhoneNumber	Consultant's Phone Number	CHAR(12)	999-999-9999	
	Consultant_Email	Consultant's Email	VARCHAR(25)	JohnDoe@email.com	
Project	Project ID	Project ID	CHAR(3)	123	PK
	Project_Manager_ID	Project Manager ID	CHAR(3)	123	FK
	Parent_Project_ID	Parent Project ID	CHAR(3)	123	FK
	Client_ID	Client ID	VARCHAR(3)	123	FK
	Project_Name	Project Name	VARCHAR(15)	Lorem Ipsum	
Client	Client_ID	Client ID	VARCHAR(3)	123	PK
	Client_Name	Name of Client	VARCHAR(25)	John Doe	
	Client_ContactName	Client's Contact Name	VARCHAR(15)	John Doe	
	Client_Phone	Client's phone number	VARCHAR(12)	999-999-9999	
Skills	Skill_ID	Skill ID	CHAR(1)	1	PK
	Skill_Name	Skill Name	VARCHAR(25)	Lorem Ipsum	
CONSULTANT_PROJECT_HISTORY	Consultant_ID	Consultant ID	VARCHAR(10)	ABCDE12345	BOTH
	Project_ID	Project ID	CHAR(3)	123	BOTH
	Evaluator_ID	Evaluation ID	CHAR(3)	123	BOTH
	Project_StartDate	Start Date of Project	DATE	DD-MM-YY	
	Project_EndDate	End Date of Project	DATE	DD-MM-YY	
	Total_HoursWorked	Total Hours Worked on Project	INT	23	

ERD

+

•

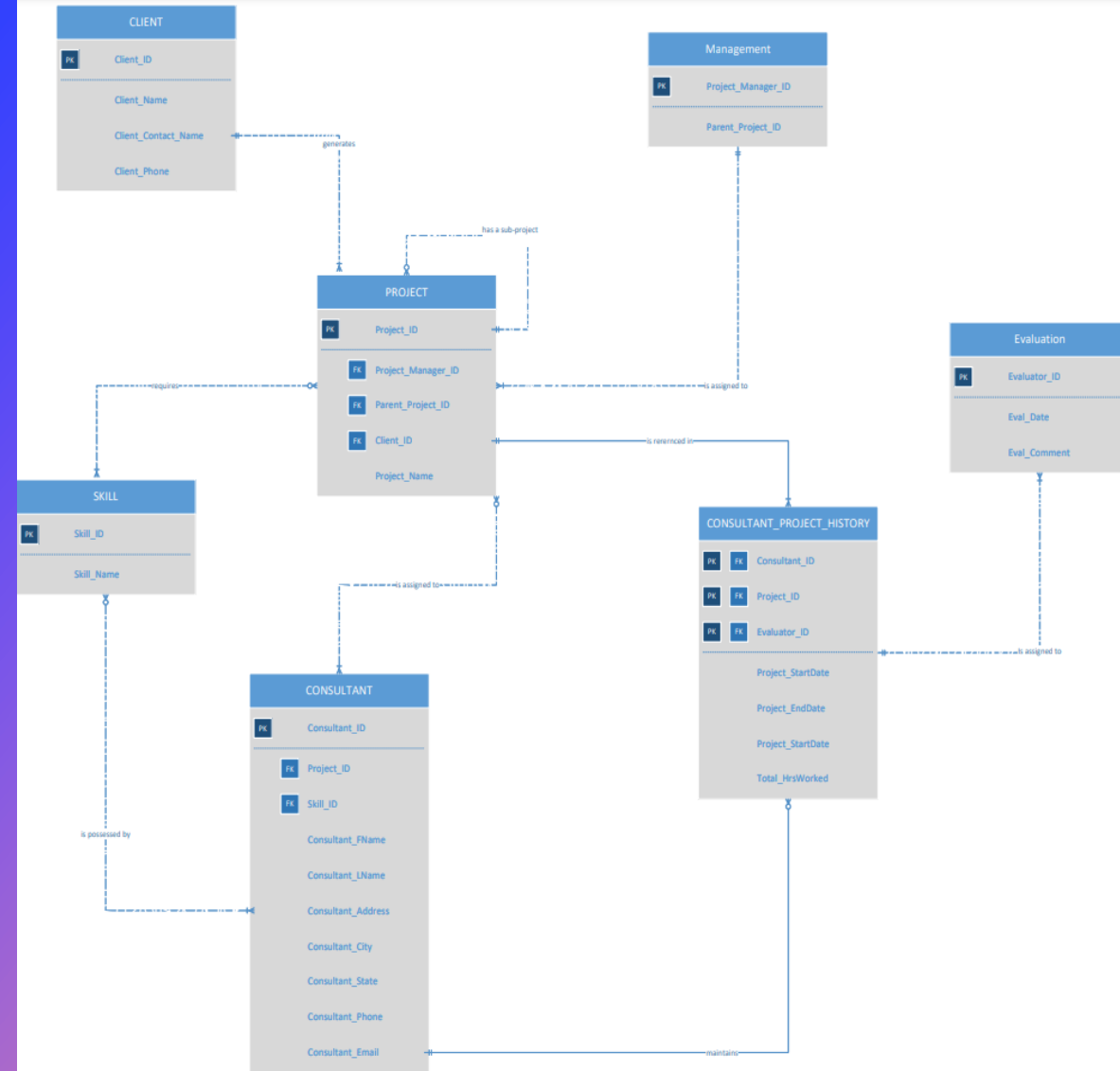
○

+

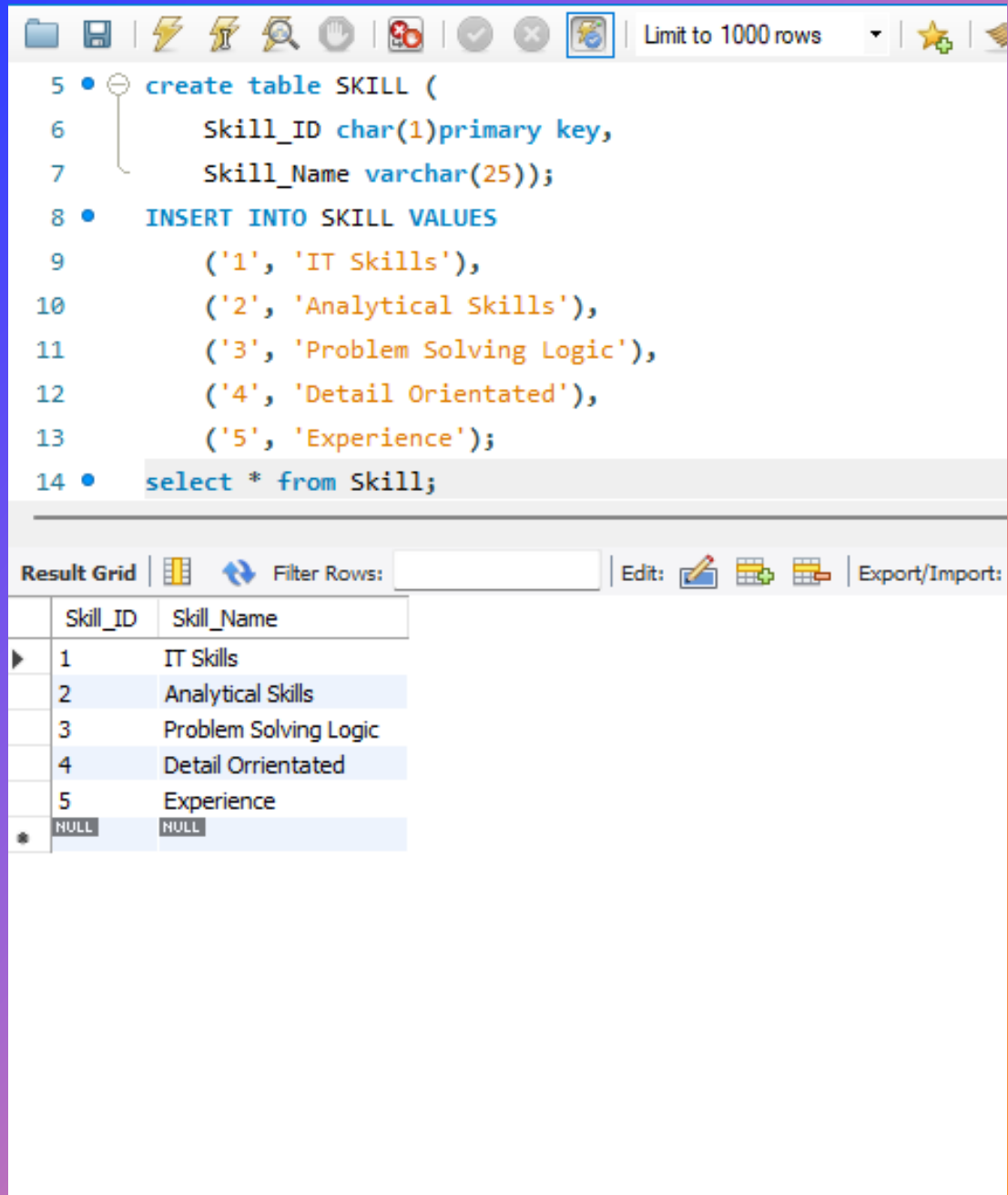
○

•

Revised ERD with two added tables for organization



• ⁺_◦ TABLE CONSTRUCTION ⁺_◦



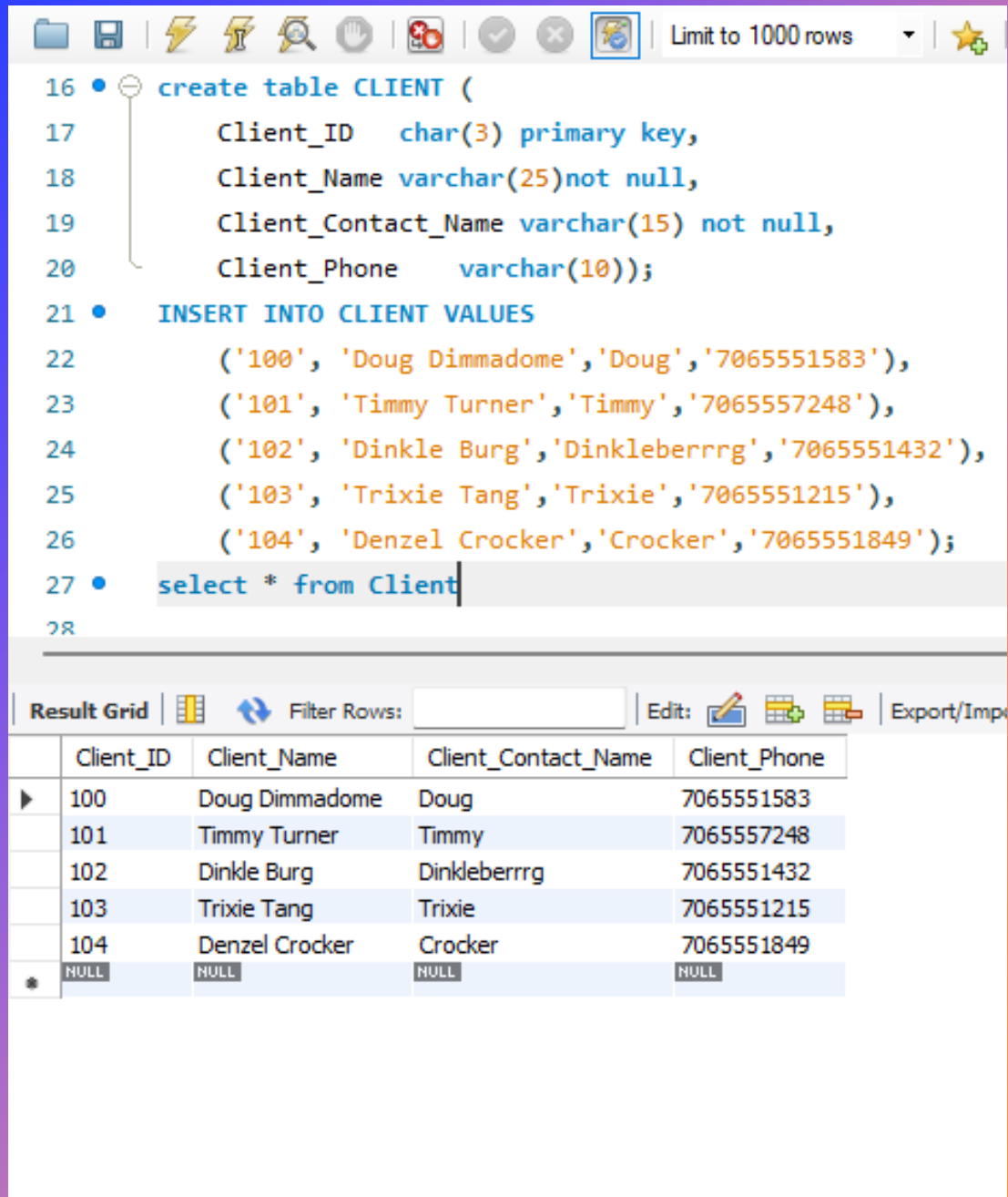
The screenshot displays a database management interface. The top toolbar includes icons for file operations, a search icon, a refresh icon, a limit to 1000 rows dropdown, and a star icon. The SQL editor contains the following code:

```
5 • create table SKILL (  
6     Skill_ID char(1)primary key,  
7     Skill_Name varchar(25));  
8 • INSERT INTO SKILL VALUES  
9     ('1', 'IT Skills'),  
10    ('2', 'Analytical Skills'),  
11    ('3', 'Problem Solving Logic'),  
12    ('4', 'Detail Orientated'),  
13    ('5', 'Experience');  
14 • select * from Skill;
```

Below the SQL editor is the 'Result Grid' section, which includes a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Filter Rows'. The result grid shows the following data:

Skill_ID	Skill_Name
1	IT Skills
2	Analytical Skills
3	Problem Solving Logic
4	Detail Orientated
5	Experience
NULL	NULL

The Skill Table



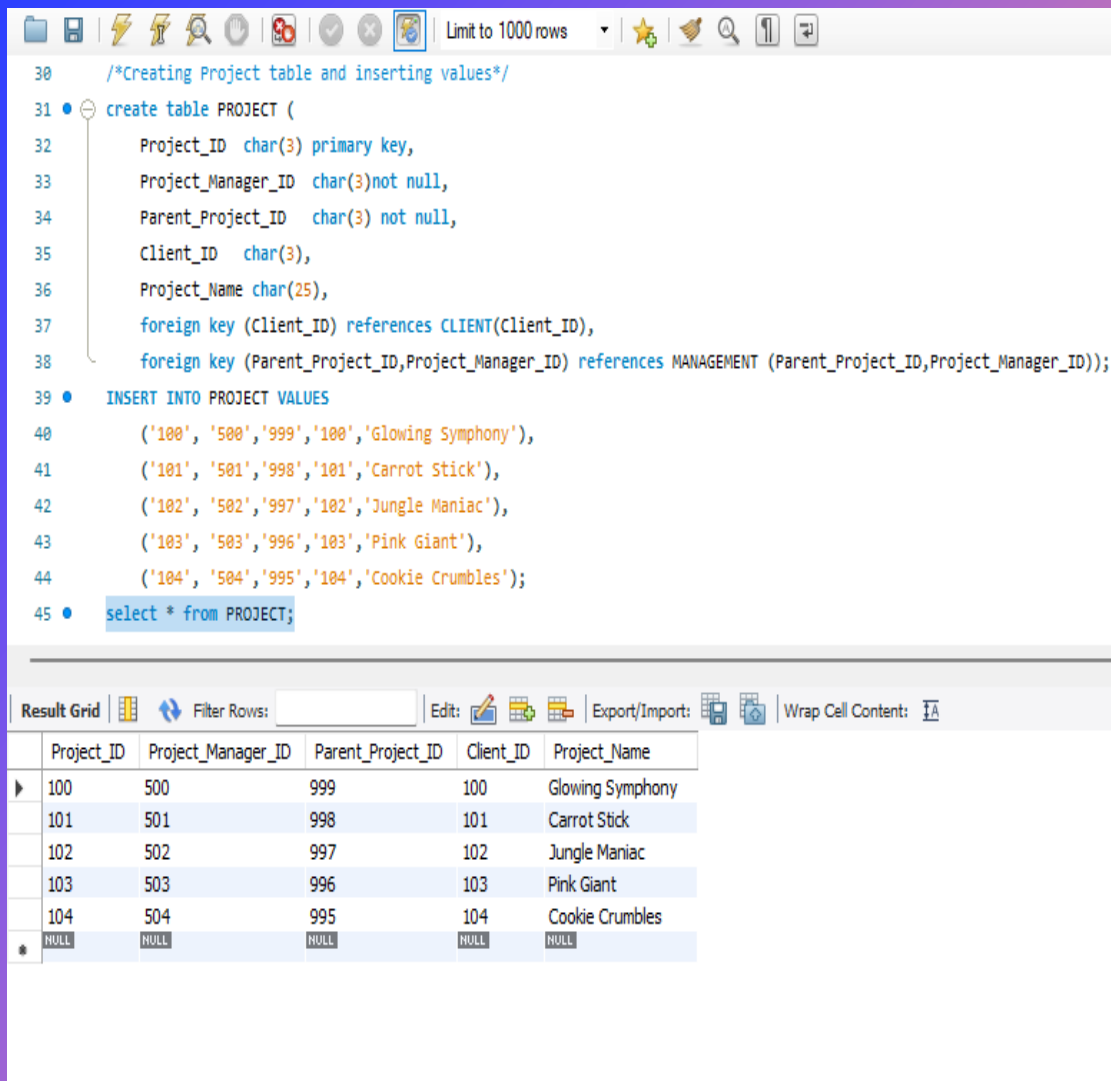
The screenshot displays a database management interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
16 • create table CLIENT (  
17     Client_ID char(3) primary key,  
18     Client_Name varchar(25) not null,  
19     Client_Contact_Name varchar(15) not null,  
20     Client_Phone varchar(10));  
21 • INSERT INTO CLIENT VALUES  
22     ('100', 'Doug Dimmadome', 'Doug', '7065551583'),  
23     ('101', 'Timmy Turner', 'Timmy', '7065557248'),  
24     ('102', 'Dinkle Burg', 'Dinkleberrrg', '7065551432'),  
25     ('103', 'Trixie Tang', 'Trixie', '7065551215'),  
26     ('104', 'Denzel Crocker', 'Crocker', '7065551849');  
27 • select * from Client
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input and an 'Edit' button. The grid displays the following data:

	Client_ID	Client_Name	Client_Contact_Name	Client_Phone
▶	100	Doug Dimmadome	Doug	7065551583
	101	Timmy Turner	Timmy	7065557248
	102	Dinkle Burg	Dinkleberrrg	7065551432
	103	Trixie Tang	Trixie	7065551215
	104	Denzel Crocker	Crocker	7065551849
*	NULL	NULL	NULL	NULL

The Client Table

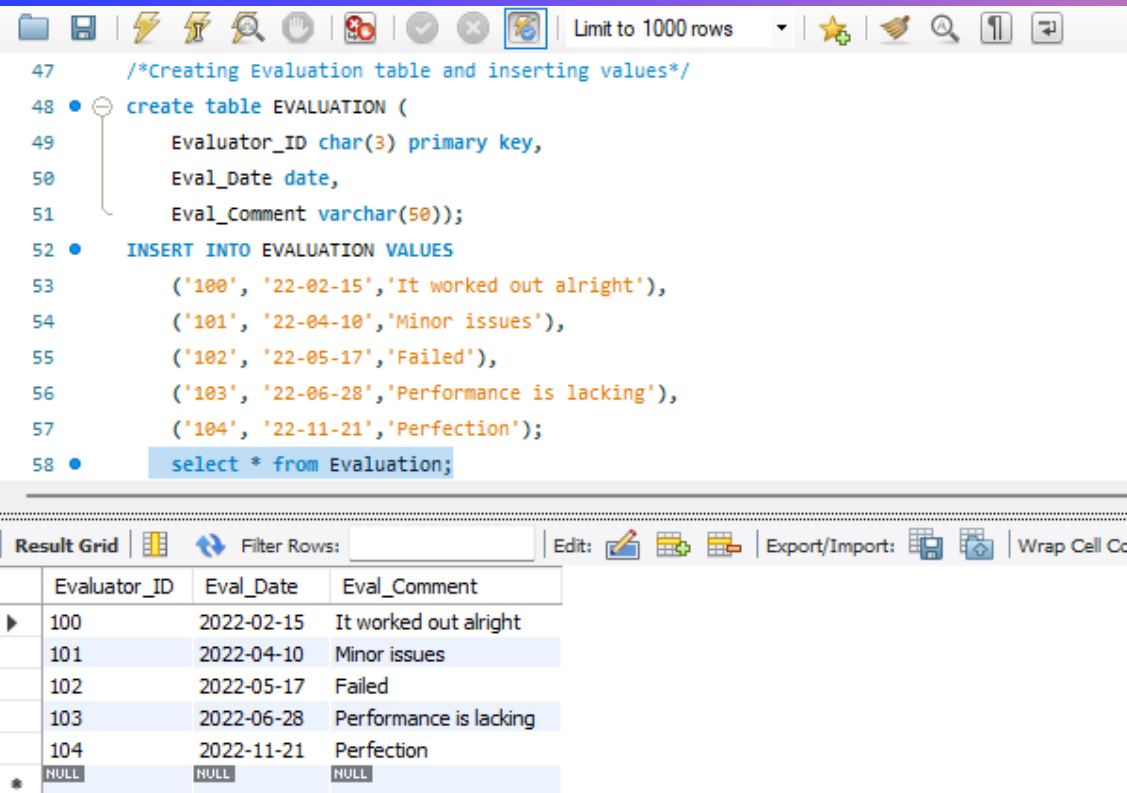


The screenshot shows a database management interface. The top section contains SQL code for creating a table and inserting data. The bottom section shows a result grid with 6 columns: Project_ID, Project_Manager_ID, Parent_Project_ID, Client_ID, and Project_Name. The grid displays 5 rows of data, including a row with NULL values.

```
30 /*Creating Project table and inserting values*/
31 create table PROJECT (
32     Project_ID char(3) primary key,
33     Project_Manager_ID char(3) not null,
34     Parent_Project_ID char(3) not null,
35     Client_ID char(3),
36     Project_Name char(25),
37     foreign key (Client_ID) references CLIENT(Client_ID),
38     foreign key (Parent_Project_ID,Project_Manager_ID) references MANAGEMENT (Parent_Project_ID,Project_Manager_ID));
39 INSERT INTO PROJECT VALUES
40     ('100', '500', '999', '100', 'Glowing Symphony'),
41     ('101', '501', '998', '101', 'Carrot Stick'),
42     ('102', '502', '997', '102', 'Jungle Maniac'),
43     ('103', '503', '996', '103', 'Pink Giant'),
44     ('104', '504', '995', '104', 'Cookie Crumbles');
45 select * from PROJECT;
```

Project_ID	Project_Manager_ID	Parent_Project_ID	Client_ID	Project_Name
100	500	999	100	Glowing Symphony
101	501	998	101	Carrot Stick
102	502	997	102	Jungle Maniac
103	503	996	103	Pink Giant
104	504	995	104	Cookie Crumbles
NULL	NULL	NULL	NULL	NULL

The Project Table



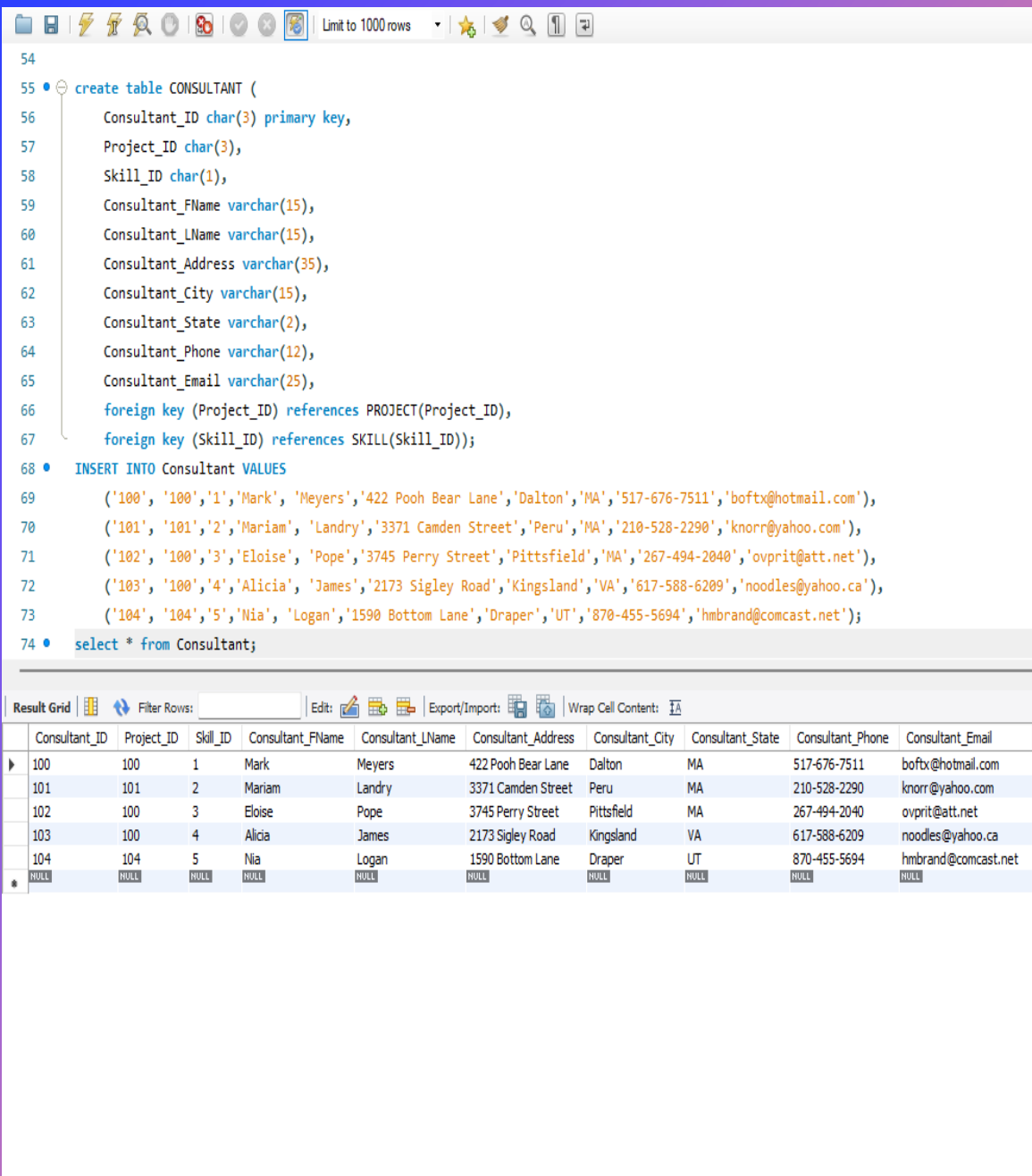
The screenshot shows a database management interface. The top pane displays SQL code for creating an 'EVALUATION' table and inserting five rows of data. The bottom pane shows the 'Result Grid' with the executed query's output. The table has columns 'Evaluator_ID', 'Eval_Date', and 'Eval_Comment'. The data rows are as follows:

Evaluator_ID	Eval_Date	Eval_Comment
100	2022-02-15	It worked out alright
101	2022-04-10	Minor issues
102	2022-05-17	Failed
103	2022-06-28	Performance is lacking
104	2022-11-21	Perfection
* NULL	NULL	NULL

The Evaluation Table

*Created for Foreign Key Use

The Consultant Table

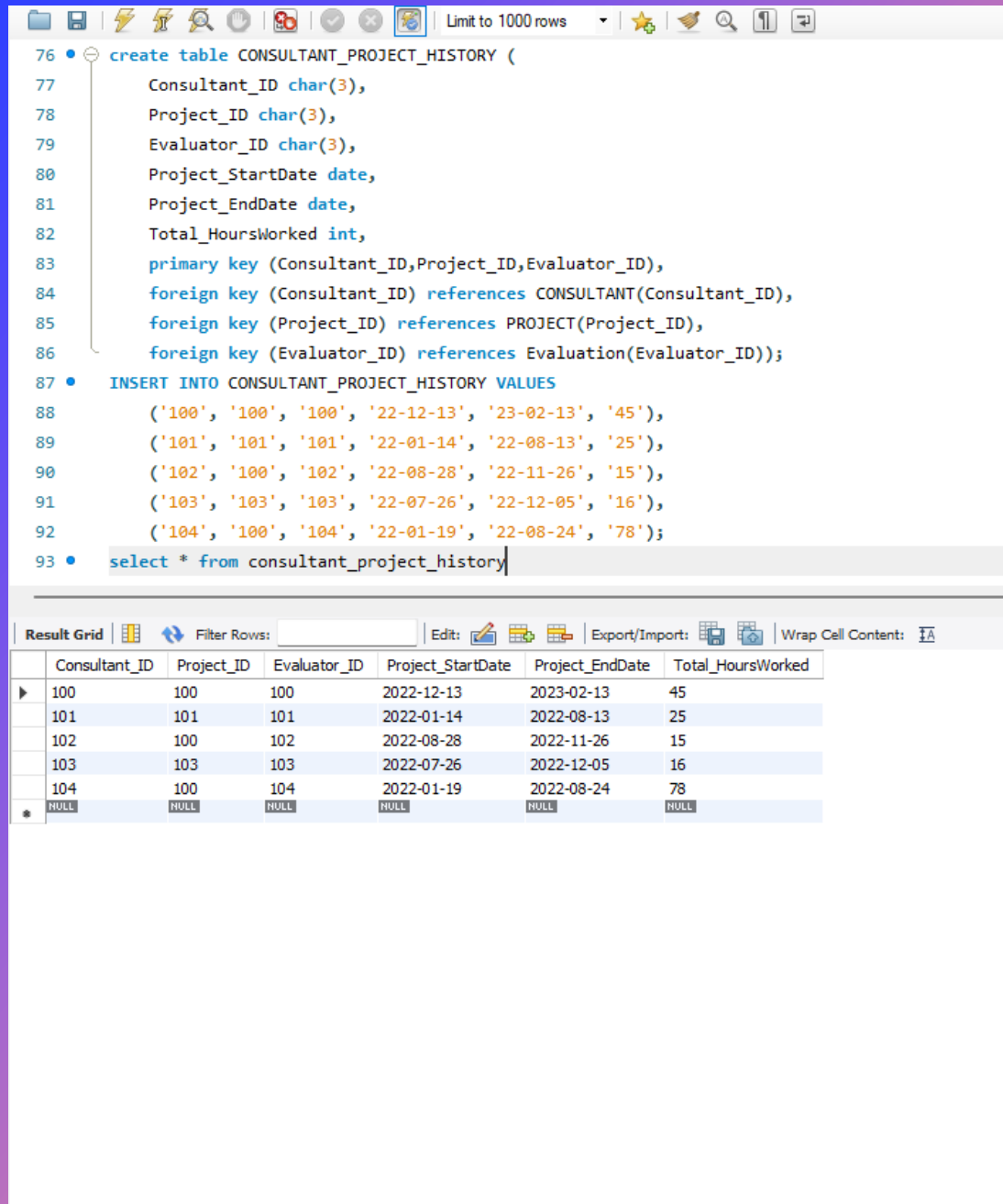


The screenshot displays a database management interface. The top section shows SQL code for creating and inserting data into a table named 'CONSULTANT'. The code includes column definitions with data types and constraints, foreign key references to 'PROJECT' and 'SKILL' tables, and an 'INSERT INTO' statement with five rows of data. The bottom section shows the 'Result Grid' with the same five rows of data, including columns for Consultant_ID, Project_ID, Skill_ID, Consultant_FName, Consultant_LName, Consultant_Address, Consultant_City, Consultant_State, Consultant_Phone, and Consultant_Email. The first four rows contain specific data, and the fifth row shows NULL values for all columns.

```
54
55 • create table CONSULTANT (
56     Consultant_ID char(3) primary key,
57     Project_ID char(3),
58     Skill_ID char(1),
59     Consultant_FName varchar(15),
60     Consultant_LName varchar(15),
61     Consultant_Address varchar(35),
62     Consultant_City varchar(15),
63     Consultant_State varchar(2),
64     Consultant_Phone varchar(12),
65     Consultant_Email varchar(25),
66     foreign key (Project_ID) references PROJECT(Project_ID),
67     foreign key (Skill_ID) references SKILL(Skill_ID));
68 • INSERT INTO Consultant VALUES
69     ('100', '100', '1', 'Mark', 'Meyers', '422 Pooh Bear Lane', 'Dalton', 'MA', '517-676-7511', 'boftx@hotmail.com'),
70     ('101', '101', '2', 'Mariam', 'Landry', '3371 Camden Street', 'Peru', 'MA', '210-528-2290', 'knorr@yahoo.com'),
71     ('102', '100', '3', 'Eloise', 'Pope', '3745 Perry Street', 'Pittsfield', 'MA', '267-494-2040', 'ovprit@att.net'),
72     ('103', '100', '4', 'Alicia', 'James', '2173 Sigley Road', 'Kingsland', 'VA', '617-588-6209', 'noodles@yahoo.ca'),
73     ('104', '104', '5', 'Nia', 'Logan', '1590 Bottom Lane', 'Draper', 'UT', '870-455-5694', 'hmbrand@comcast.net');
74 • select * from Consultant;
```

Consultant_ID	Project_ID	Skill_ID	Consultant_FName	Consultant_LName	Consultant_Address	Consultant_City	Consultant_State	Consultant_Phone	Consultant_Email
100	100	1	Mark	Meyers	422 Pooh Bear Lane	Dalton	MA	517-676-7511	boftx@hotmail.com
101	101	2	Mariam	Landry	3371 Camden Street	Peru	MA	210-528-2290	knorr@yahoo.com
102	100	3	Eloise	Pope	3745 Perry Street	Pittsfield	MA	267-494-2040	ovprit@att.net
103	100	4	Alicia	James	2173 Sigley Road	Kingsland	VA	617-588-6209	noodles@yahoo.ca
104	104	5	Nia	Logan	1590 Bottom Lane	Draper	UT	870-455-5694	hmbrand@comcast.net
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The Consultant Project History Table



The screenshot shows a database management interface. The top section contains SQL code for creating a table and inserting data. The bottom section shows a 'Result Grid' with the data inserted into the 'CONSULTANT_PROJECT_HISTORY' table.

```
76 • create table CONSULTANT_PROJECT_HISTORY (  
77     Consultant_ID char(3),  
78     Project_ID char(3),  
79     Evaluator_ID char(3),  
80     Project_StartDate date,  
81     Project_EndDate date,  
82     Total_HoursWorked int,  
83     primary key (Consultant_ID,Project_ID,Evaluator_ID),  
84     foreign key (Consultant_ID) references CONSULTANT(Consultant_ID),  
85     foreign key (Project_ID) references PROJECT(Project_ID),  
86     foreign key (Evaluator_ID) references Evaluation(Evaluator_ID));  
87 • INSERT INTO CONSULTANT_PROJECT_HISTORY VALUES  
88     ('100', '100', '100', '22-12-13', '23-02-13', '45'),  
89     ('101', '101', '101', '22-01-14', '22-08-13', '25'),  
90     ('102', '100', '102', '22-08-28', '22-11-26', '15'),  
91     ('103', '103', '103', '22-07-26', '22-12-05', '16'),  
92     ('104', '100', '104', '22-01-19', '22-08-24', '78');  
93 • select * from consultant_project_history
```

Result Grid

	Consultant_ID	Project_ID	Evaluator_ID	Project_StartDate	Project_EndDate	Total_HoursWorked
▶	100	100	100	2022-12-13	2023-02-13	45
	101	101	101	2022-01-14	2022-08-13	25
	102	100	102	2022-08-28	2022-11-26	15
	103	103	103	2022-07-26	2022-12-05	16
	104	100	104	2022-01-19	2022-08-24	78
*	NULL	NULL	NULL	NULL	NULL	NULL


```
100      /*Creating Management table and inserting values*/
101 • create table MANAGEMENT (
102     Project_Manager_ID char(3)not null,
103     Parent_Project_ID char(3)not null,
104     primary key (Parent_Project_ID,Project_Manager_ID));
105 • INSERT INTO MANAGEMENT VALUES
106     ('500','999'),
107     ('501','998'),
108     ('502','997'),
109     ('503','996'),
110     ('504','995');
111 • select * from management;
112
```

Result Grid | Filter Rows: | Edit: | Export/Import

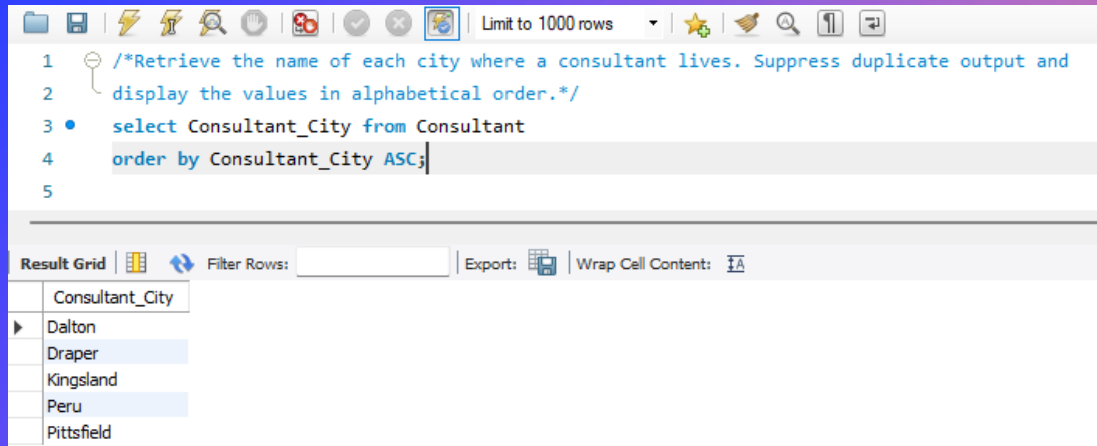
	Project_Manager_ID	Parent_Project_ID
▶	504	995
	503	996
	502	997
	501	998
	500	999

The Management Table

*Created for Foreign Key Use



QUERY BUILDING



The screenshot shows a database query editor with a toolbar at the top containing icons for file operations, execution, and navigation. The query text is as follows:

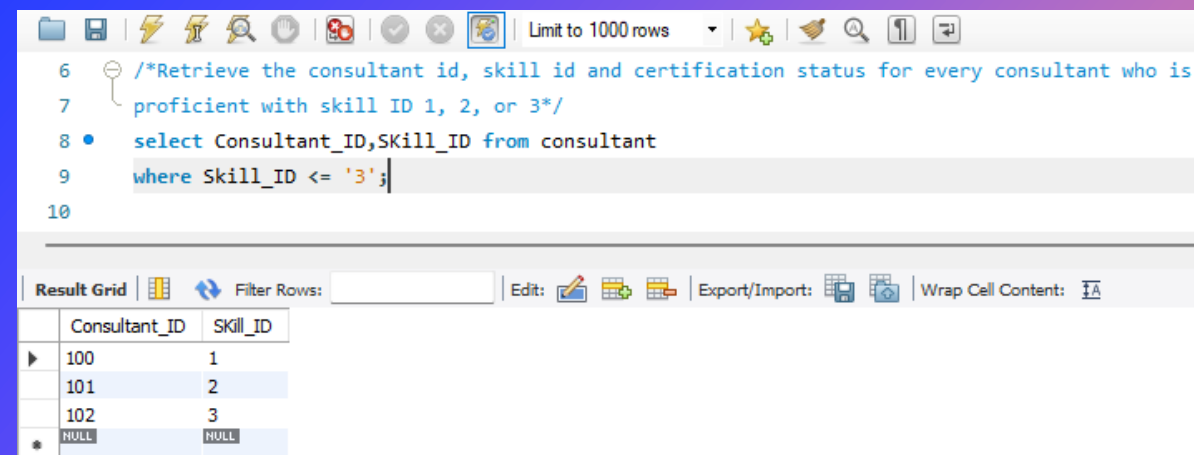
```
1 /*Retrieve the name of each city where a consultant lives. Suppress duplicate output and
2 display the values in alphabetical order.*/
3 • select Consultant_City from Consultant
4 order by Consultant_City ASC;
5
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The grid displays a single column titled 'Consultant_City' with the following values:

Consultant_City
Dalton
Draper
Kingsland
Peru
Pittsfield

Retrieve the name of
each city where a
consultant lives.
Suppress duplicate
output and display
the values in
alphabetical order

Retrieve the
consultant id, skill id
and certification
status for every
consultant who is
proficient with skill ID
1, 2, or 3



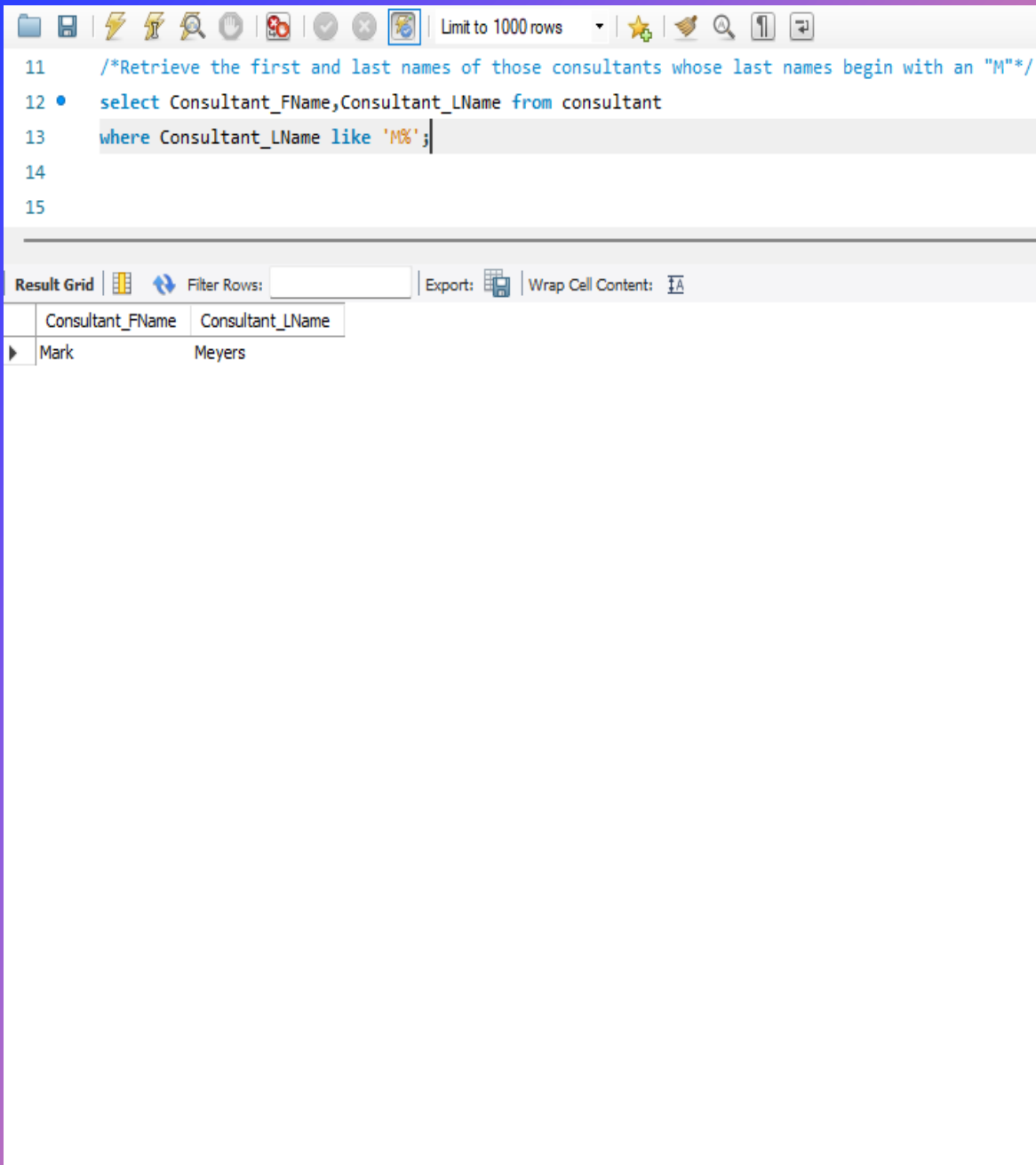
The screenshot shows a database query editor with a toolbar at the top containing icons for file operations, execution, and navigation. Below the toolbar, a SQL query is entered in a text area. The query is as follows:

```
6  /*Retrieve the consultant id, skill id and certification status for every consultant who is
7  proficient with skill ID 1, 2, or 3*/
8  •  select Consultant_ID,Skill_ID from consultant
9     where Skill_ID <= '3';
10
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid displays the following data:

	Consultant_ID	Skill_ID
▶	100	1
	101	2
	102	3
*	NULL	NULL

Retrieve the first
and last names of
those consultants
whose last names
begin with an "M"

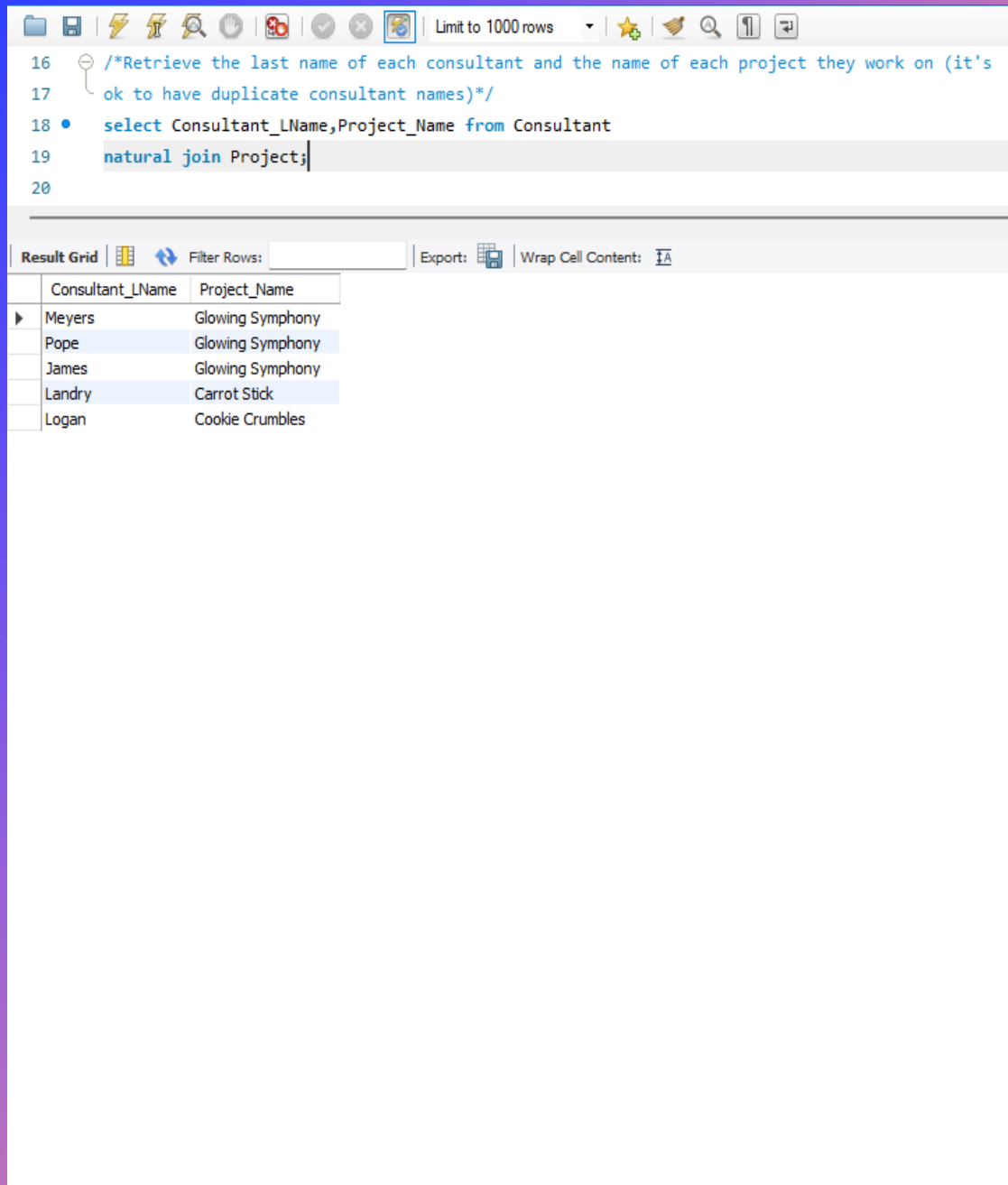


The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, a SQL query is entered in a text area:

```
11 /*Retrieve the first and last names of those consultants whose last names begin with an "M"*/  
12 • select Consultant_FName,Consultant_LName from consultant  
13 where Consultant_LName like 'M%';  
14  
15
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid itself is a table with two columns: "Consultant_FName" and "Consultant_LName". The first row of data shows "Mark" and "Meyers".

Consultant_FName	Consultant_LName
Mark	Meyers



The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered in a text area. The query is as follows:

```
16 /*Retrieve the last name of each consultant and the name of each project they work on (it's  
17 ok to have duplicate consultant names)*/  
18 • select Consultant_LName,Project_Name from Consultant  
19 natural join Project;  
20
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'Consultant_LName' and 'Project_Name'.

Consultant_LName	Project_Name
Meyers	Glowing Symphony
Pope	Glowing Symphony
James	Glowing Symphony
Landry	Carrot Stick
Logan	Cookie Crumbles

Retrieve the last
name of each
consultant and the
name of each project
they work on (it's ok
to have duplicate
consultant names)

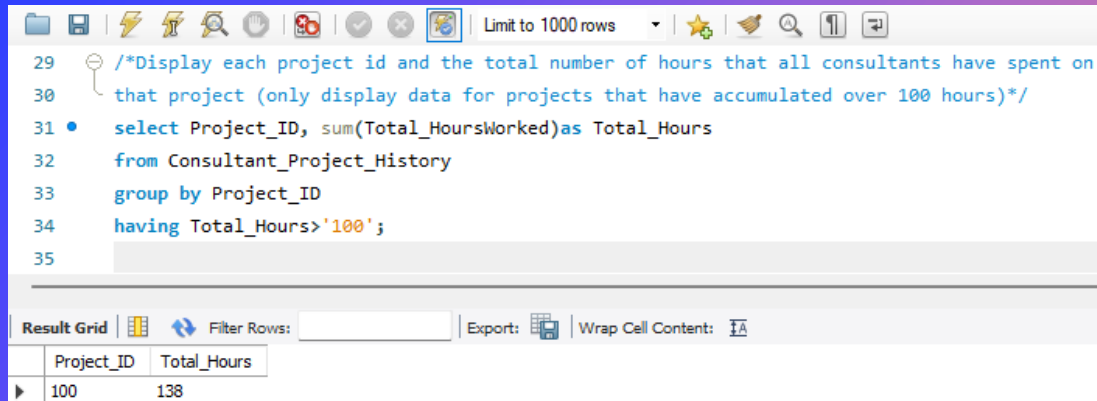
```
21  /*List the first and last name of every consultant who has worked with Mark Meyers (use a
22  subquery)*/
23  •  select distinct Consultant_FName, Consultant_LName
24  from consultant where Project_ID = ( select distinct Project_ID
25  from consultant where Consultant_LName = 'Meyers');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Consultant_FName	Consultant_LName
▶	Mark	Meyers
	Eloise	Pope
	Alicia	James

List the first and last name of every consultant who has worked with Mark Meyers (use a subquery)

Display each project id and the total number of hours that all consultants have spent on that project (only display data for projects that have accumulated over 100 hours)



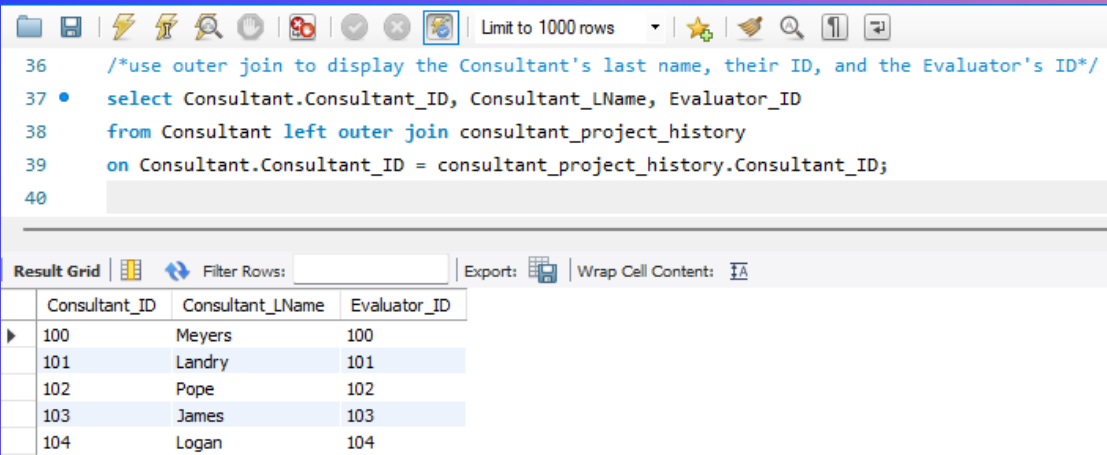
The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered in a text area. The query is as follows:

```
29 /*Display each project id and the total number of hours that all consultants have spent on
30 that project (only display data for projects that have accumulated over 100 hours)*/
31 • select Project_ID, sum(Total_HoursWorked)as Total_Hours
32 from Consultant_Project_History
33 group by Project_ID
34 having Total_Hours>'100';
35
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'Project_ID' and 'Total_Hours'.

Project_ID	Total_Hours
100	138

Use outer join to display the Consultant's last name, their ID, and the Evaluator's ID

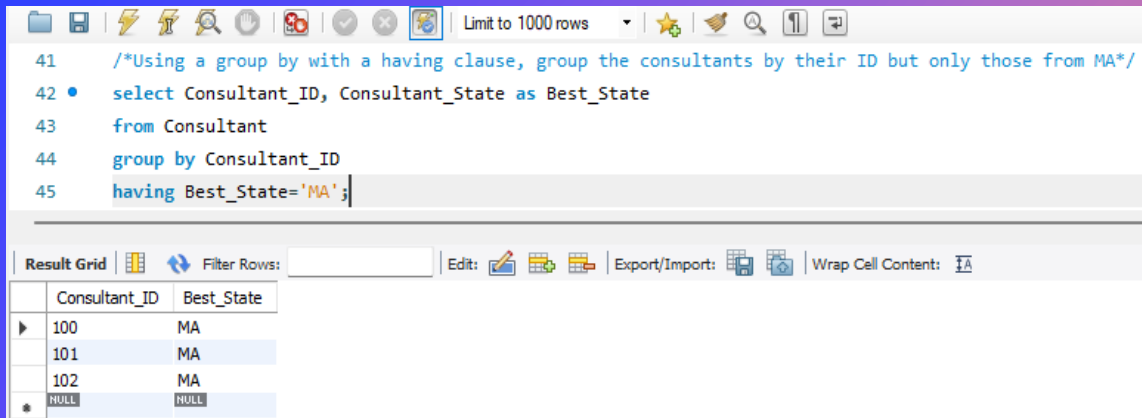


The screenshot shows a database query editor with a toolbar at the top containing icons for file operations, execution, and navigation. The query text is as follows:

```
36 /*use outer join to display the Consultant's last name, their ID, and the Evaluator's ID*/
37 • select Consultant.Consultant_ID, Consultant_LName, Evaluator_ID
38 from Consultant left outer join consultant_project_history
39 on Consultant.Consultant_ID = consultant_project_history.Consultant_ID;
40
```

Below the query editor is the 'Result Grid' section, which includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays a table with the following data:

	Consultant_ID	Consultant_LName	Evaluator_ID
▶	100	Meyers	100
	101	Landry	101
	102	Pope	102
	103	James	103
	104	Logan	104



The screenshot shows a database query editor with a toolbar at the top. The SQL query is as follows:

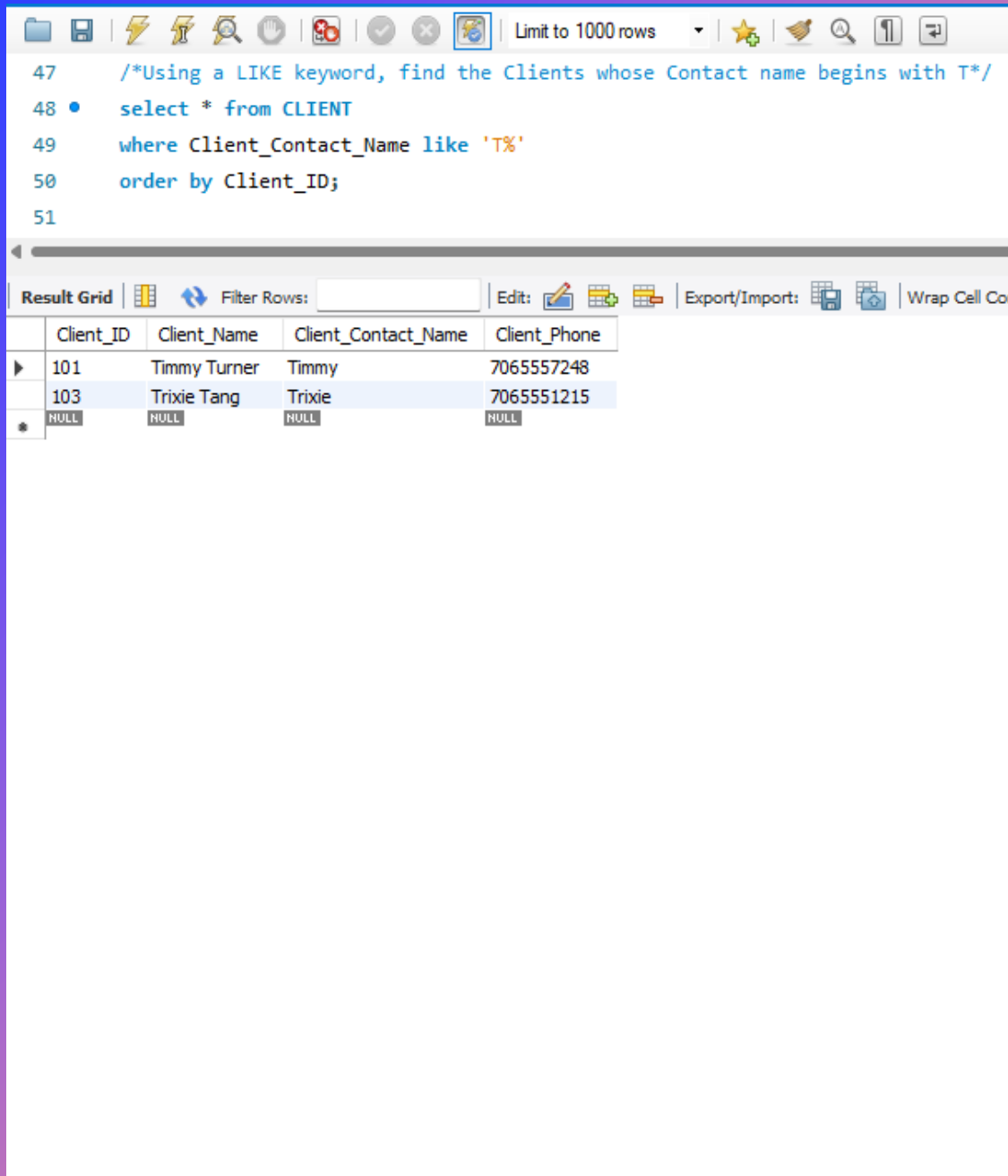
```
41 /*Using a group by with a having clause, group the consultants by their ID but only those from MA*/
42 • select Consultant_ID, Consultant_State as Best_State
43 from Consultant
44 group by Consultant_ID
45 having Best_State='MA';
```

Below the query, the 'Result Grid' is displayed with the following data:

Consultant_ID	Best_State
100	MA
101	MA
102	MA
NULL	NULL

Using a “group by having” clause, group the consultants by their ID but only those from MA

Using a LIKE keyword, find the Clients whose Contact name begins with T

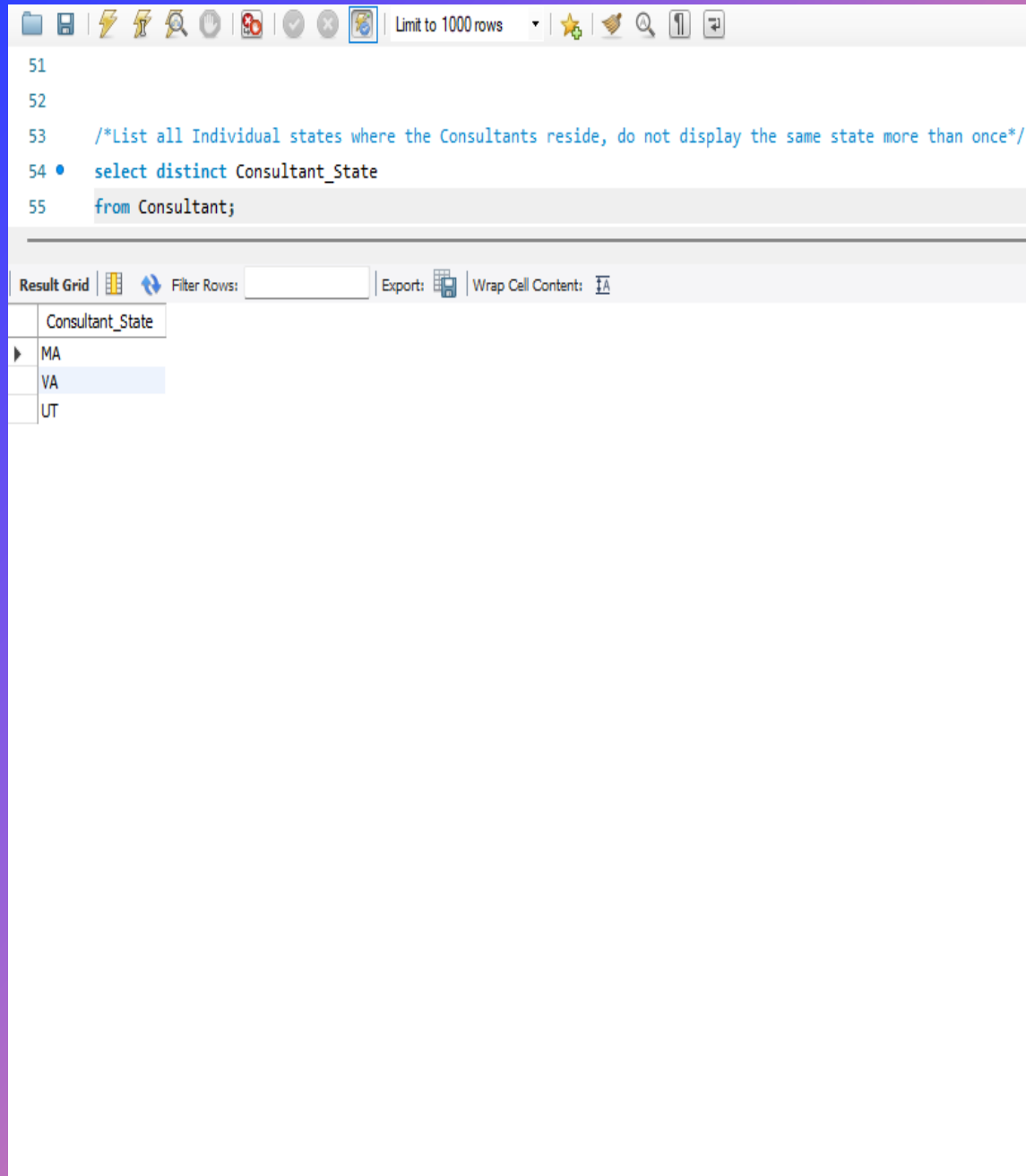


The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a text input field containing "Limit to 1000 rows". Below the toolbar, a SQL query is displayed in a text area. The query is as follows:

```
47 /*Using a LIKE keyword, find the Clients whose Contact name begins with T*/  
48 • select * from CLIENT  
49 where Client_Contact_Name like 'T%'  
50 order by Client_ID;  
51
```

Below the query, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The result grid itself is a table with the following data:

Client_ID	Client_Name	Client_Contact_Name	Client_Phone
101	Timmy Turner	Timmy	7065557248
103	Trixie Tang	Trixie	7065551215
NULL	NULL	NULL	NULL



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, a 'Limit to 1000 rows' dropdown, and various query tools. The SQL editor contains the following code:

```
51  
52  
53 /*List all Individual states where the Consultants reside, do not display the same state more than once*/  
54 • select distinct Consultant_State  
55 from Consultant;
```

Below the editor is a 'Result Grid' section with a 'Filter Rows' input and an 'Export' button. The grid displays the results of the query:

Consultant_State
MA
VA
UT

List all Individual states where the Consultants reside, do not display the same state more than once



USEFUL ADDITIONS



FEEDBACK AND COMMENTS ABOUT CLIENT

THE SRC CONSULTATION
PRESENTATION

- Being able to leave comments and feedback about certain clients and how the projects went while working with them would be useful in order to decide on how to go about business with them later down the road. Not only would it be a good indicator of what clients work well, it could also be useful in order to show who are returning customers.
- Although it would add an additional table, it would also serve as a resource for more information on the clients that are good to work with.

TOO MANY PROJECTS ASSIGNED TO TOO LITTLE OF A CREW

THE SRC CONSULTATION
PRESENTATION

- By using 1:M on so many people alone, we are stretching employees too thin
- It would make sense assigning managers based on certain certifications, for them to be better suited on certain projects, rather than have them work on many projects regardless of certifications.

+

o

.

THANK YOU

Any Questions?

