UNIVERSITY OF DAYTON

COURT CASE MANAGEMENT SYSTEM

(Final Report)

DATA ADDED, SQL QUERIES CREATED

AND

TABLES CREATED

By

SANDEEP SRIVATSAVA SINGARAJU (1017432390)

SAI ARAVIND NAKARIKANTI (1017453510)

SOURAB DEB (1017442430)

Under the Guidance of

CEMIL KIRBAS

DATABASE MANAGEMENT SYSTEMS

CPS 542

DEPARTMENT OF COMPUTER SCIENCE

Table of Contents

Introduction 3

Entities 3

Attributes 4

Relationships 5

ER Diagram 7

Relational Schema 9

Normalization 11

Logical Model 13

Physical Model 14

Table Views 17

Data Manipulation 21

Conclusion 26

Court Case Database Management Design

Introduction:

A court database is a collection of data about court cases and legal proceedings that is usually kept up to date by a government organization. It might contain information about the parties involved, the accusations or charges, the verdict, and any supporting records or transcripts. Attorneys, academics, journalists, and the general public can access information regarding court cases and the legal system through court databases. They might be accessible online, where you can search for them using the case number, the parties' names, or other terms. Access to court schedules, dockets, and other details regarding forthcoming court appearances is also available through some court databases.

Entities:

In DBMS (Database Management System), an entity is a real-world object or concept that has attributes that describe its characteristics. Entities can be anything from physical objects, such as people or products, to abstract concepts, such as accounts or transactions.

In a database, an entity is typically represented as a table, with each row representing an instance of that entity and each column representing an attribute of that entity. For example, a client entity in a database might have attributes such as client name, address, phone number, and email.

Entities are important in database design because they help to organize and structure the data in a way that makes sense for the business or application. By identifying the entities and their attributes, designers can create a database schema that is optimized for storing and retrieving data efficiently.

In addition to attributes, entities can also have relationships with other entities. For example, a client entity might have a relationship with an attorney entity, indicating that a client can place multiple orders. These relationships are also important in database design because they help to define the overall structure of the database and ensure that the data is consistent and accurate. In this database we have 7 entities and they are:

- Client
- Case

- Attorney
- Court
- Judge
- Jury
- Bar Council

Attributes:

In database management systems, an attribute refers to a characteristic or property of an entity. Attributes are used to describe the entities that make up a database, and they provide the details and information that can be stored, retrieved, and analysed within the database.

Attributes can be of different types, depending on the kind of data they represent. Some common types of attributes include:

- Numeric attributes: These attributes represent numeric values such as integers or floating-point numbers. Examples include price, quantity, and age.
- Text attributes: These attributes represent textual data, such as names, addresses, and descriptions.
- Boolean attributes: These attributes represent a binary value, such as true/false or yes/no.
- Date/time attributes: These attributes represent dates and times, such as birthdates, order dates, or delivery times.

Attributes can be further classified as primary or foreign key attributes, which help define relationships between different entities in a database. A primary key is a unique identifier for an entity, while a foreign key is a reference to the primary key of another entity.

Attributes play a crucial role in the design and development of a database. They provide the structure for storing data and enable efficient searching, sorting, and querying of data within the database. By carefully selecting and organizing attributes, database designers can create a database that is optimized for the specific needs of the organization or application. In this database the attributes are as follows:

- Client – C_ID, C_Lname, C_Fname, C_Address, C_Sex
- Case – Case_ID, Case_Desp, Attorney_ID
- Attorney – A_ID, A_Name, A_Email, A_Address
- Court – Co_ID, Co_Name, Co_Address, Co_City, Co_Zipcode
- Judge – J_ID, J_Name, J_PhoneNo, J_Address
- Jury – No_of_Jurors, Juries_Verdicts
- Bar Council – A_General, A_ID, Bar_council_ID

Relationships:

In database management systems, a relationship is an association between two or more entities that can be used to represent the real-world connections and interactions between those entities. Relationships allow data to be organized in a meaningful way and enable queries to be performed that can retrieve related data from multiple tables.

There are three main types of relationships that can exist between entities in a database:

One-to-one relationship: In this type of relationship, one instance of an entity is associated with only one instance of another entity. For example, in a database of employees and their office locations, each employee can have only one office location, and each office location can be associated with only one employee.

One-to-many relationship: In this type of relationship, one instance of an entity can be associated with multiple instances of another entity, but each instance of the other entity can be associated with only one instance of the first entity. For example, in a database of customers and their orders, each customer can place multiple orders, but each order is associated with only one customer.

Many-to-many relationship: In this type of relationship, multiple instances of one entity can be associated with multiple instances of another entity. For example, in a database of students and classes, each student can enroll in multiple classes, and each class can have multiple students enrolled.

To represent relationships between entities in a database, foreign keys are often used. A foreign key is a field in one table that refers to the primary key of another table, creating a link between the two tables. The foreign key allows related data to be linked and retrieved from multiple tables when performing queries.

Overall, relationships are an essential concept in database design as they help to organize and structure data in a meaningful way, enabling efficient retrieval and analysis of data.

- registers: Client registers case
- choose: Client chooses Attorney
- retains: Attorney retains case
- members_of: Attorneys are members of bar council
- assigned_to: Case is assigned to court
- works_for: Judge works for court
- passes_to: Judge passes the case to Jury

ER DIAGRAM

An Entity-Relationship (ER) diagram is a graphical representation of entities and their relationships to each other in a database. It is used to model and design the database schema and can help to visualize the relationships between entities.

An ER diagram consists of three main components:

Entities: Entities are objects or concepts that exist independently and have distinct attributes or characteristics. In a database, an entity is typically represented as a table.

Relationships: Relationships define the associations or connections between entities. There are three types of relationships: one-to-one, one-to-many, and many-to-many.

Attributes: Attributes describe the properties or characteristics of an entity. Each entity has one or more attributes.

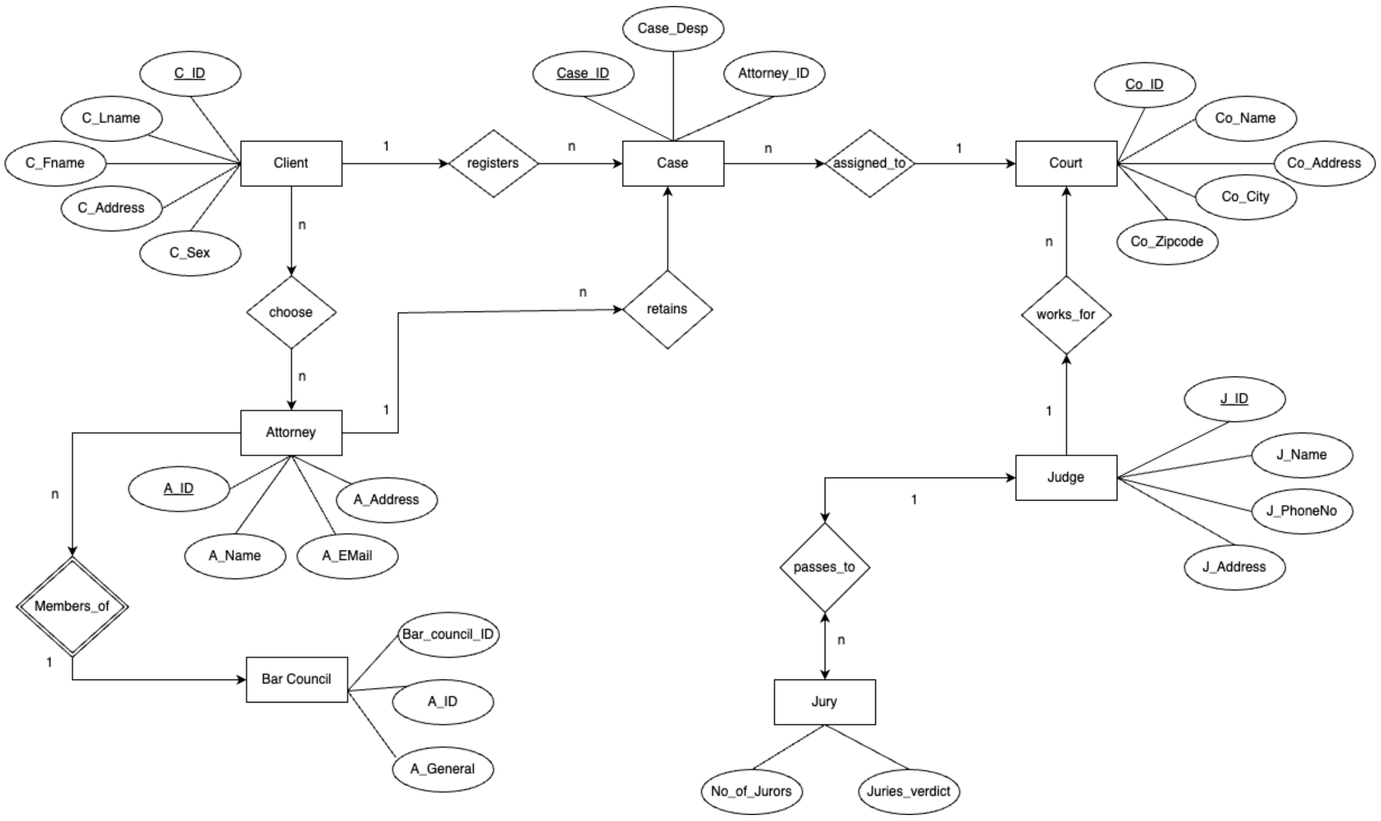
ER diagrams are typically created using symbols and notation to represent entities, relationships, and attributes. The symbols used may vary depending on the specific notation system being used, but some common symbols include:

Entity: represented by a rectangle with the entity name inside

Relationship: represented by a diamond shape connecting two entities

Attribute: represented by an oval shape connected to the entity rectangle

By creating an ER diagram, designers can visualize the relationships between entities and design a database schema that is efficient, flexible, and scalable. The ER Diagram for Court case management database is as follows


C_ID
C_Lname
C_Fname
C_Address
C_Sex
Client
1
registers
n
Case_ID
Case_Desp
Attorney_ID
Case
n
assigned_to
1
Co_ID
Co_Name
Co_Address
Co_City
Co_Zipcode
Court
n
choose
n
retains
n
works_for
1
Attorney
1
A_ID
A_Address
A_Name
A_EMail
n
Members_of
1
Bar Council
Bar_council_ID
A_ID
A_General
J_ID
J_Name
J_PhoneNo
J_Address
Judge
1
passes_to
n
Jury
No_of_Jurors
Juries_verdict

Relational Schema:

A relational schema is a blueprint that describes the structure and organization of a relational database. It defines the tables, fields, and relationships between tables that make up the database, and it provides a formal description of the data that is stored in the database.

A relational schema consists of several components, including:

Tables: Tables are the main data containers in a relational database. They are used to store data in rows and columns, with each column representing a field and each row representing a record.

Fields: Fields are individual data elements that make up a table. They are also called columns or attributes, and they represent specific characteristics or properties of the data being stored.

Primary keys: A primary key is a field or combination of fields that uniquely identify each record in a table. It is used to ensure that each record is unique and to create relationships between tables.

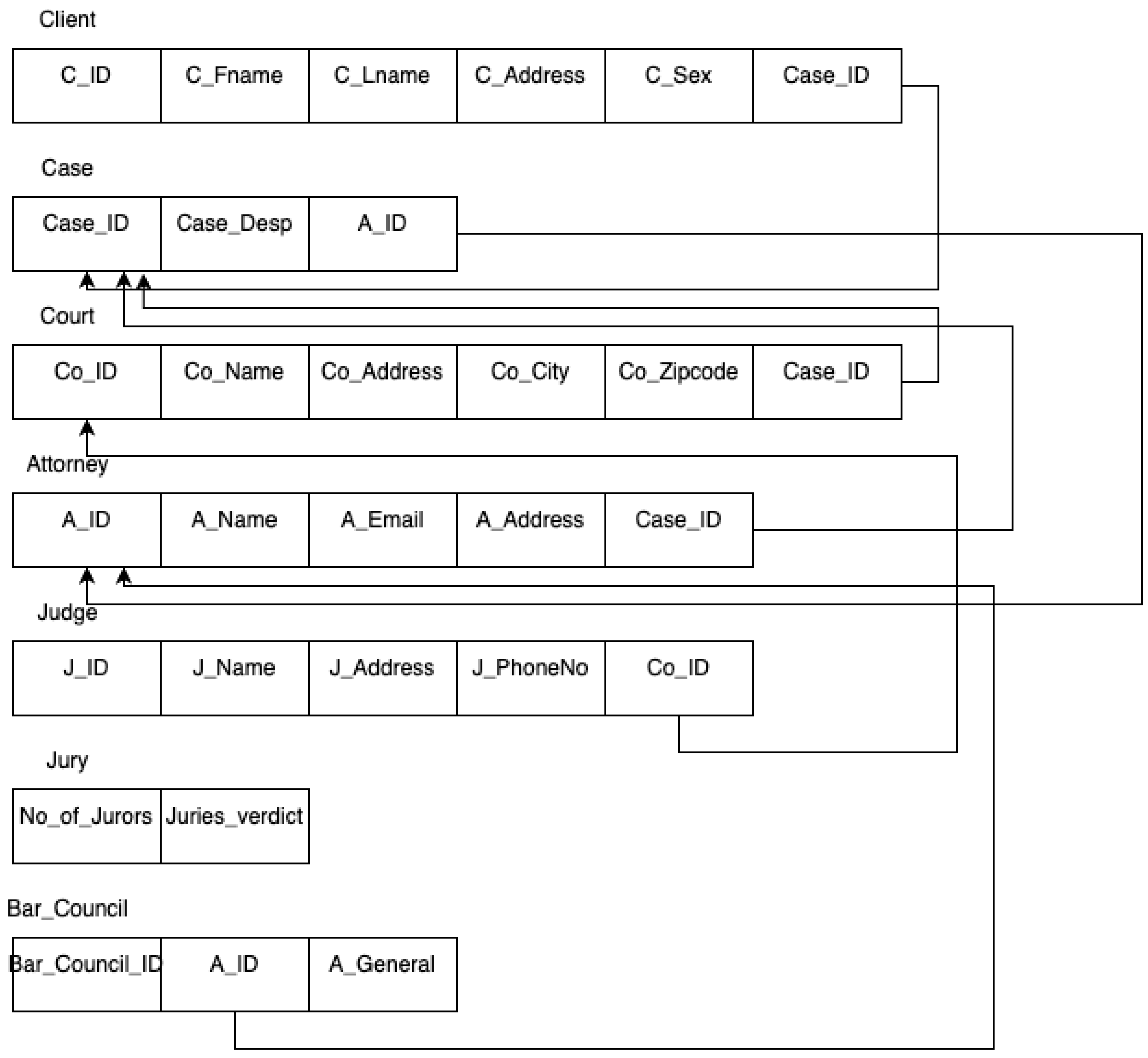
Foreign keys: A foreign key is a field in one table that refers to the primary key of another table. It is used to create relationships between tables and ensure data consistency.

Relationships: Relationships define the associations or connections between tables in a database. There are three main types of relationships: one-to-one, one-to-many, and many-to-many.

A relational schema can be represented graphically using an Entity-Relationship (ER) diagram, which uses symbols to represent tables, fields, and relationships between tables. The schema can also be expressed in a formal language, such as SQL (Structured Query Language), which is used to create, modify, and retrieve data from relational databases.

Overall, a relational schema is a key component of database design and helps to ensure that data is organized, structured, and stored in a way that is efficient and effective for the specific needs of the organization or application.

Relational Schema for the Database:



Normalization:

To normalize the given database, we need to apply normalization rules to eliminate data redundancy and improve data consistency.

1. First Normal Form (1NF)

To achieve 1NF, we need to make sure that each attribute in a table contains only atomic values. In the given database, all attributes appear to be atomic, so the database is already in 1NF.

2. Second Normal Form (2NF)

To achieve 2NF, we need to make sure that every non-key attribute is fully dependent on the primary key. In the given database, we can see that the Case table has a partial dependency on the primary key, Attorney_ID. To remove this dependency, we can create a new table for Attorney and move the Attorney-related attributes to that table. This will give us the following tables:

Client (C_ID, C_Lname, C_Fname, C_Address, C_Sex)

Attorney (A_ID, A_Name, A_Email, A_Address)

Case (Case_ID, Case_Desp, Attorney_ID)

Court (Co_ID, Co_Name, Co_Address, Co_City, Co_Zipcode)

Judge (J_ID, J_Name, J_PhoneNo, J_Address)

Jury (No_of_Jurors, Juries_Verdicts)

Bar_Council (A_General, A_ID, Bar_council_ID)

3. Third Normal Form (3NF)

To achieve 3NF, we need to make sure that there are no transitive dependencies in the tables. In the given database, we can see that the Attorney table has a transitive dependency on the Bar_Council table through the A_ID attribute. To remove this dependency, we can create a new table for the relationship between Attorney and Bar_Council. This will give us the following normalized tables:

Client (C_ID, C_Lname, C_Fname, C_Address, C_Sex)

Attorney (A_ID, A_Name, A_Email, A_Address, Bar_council_ID)

Attorney_Bar_Council (A_ID, Bar_council_ID, A_General)

Case (Case_ID, Case_Desp, Attorney_ID)

Court (Co_ID, Co_Name, Co_Address, Co_City, Co_Zipcode)

Judge (J_ID, J_Name, J_PhoneNo, J_Address, Co_ID)

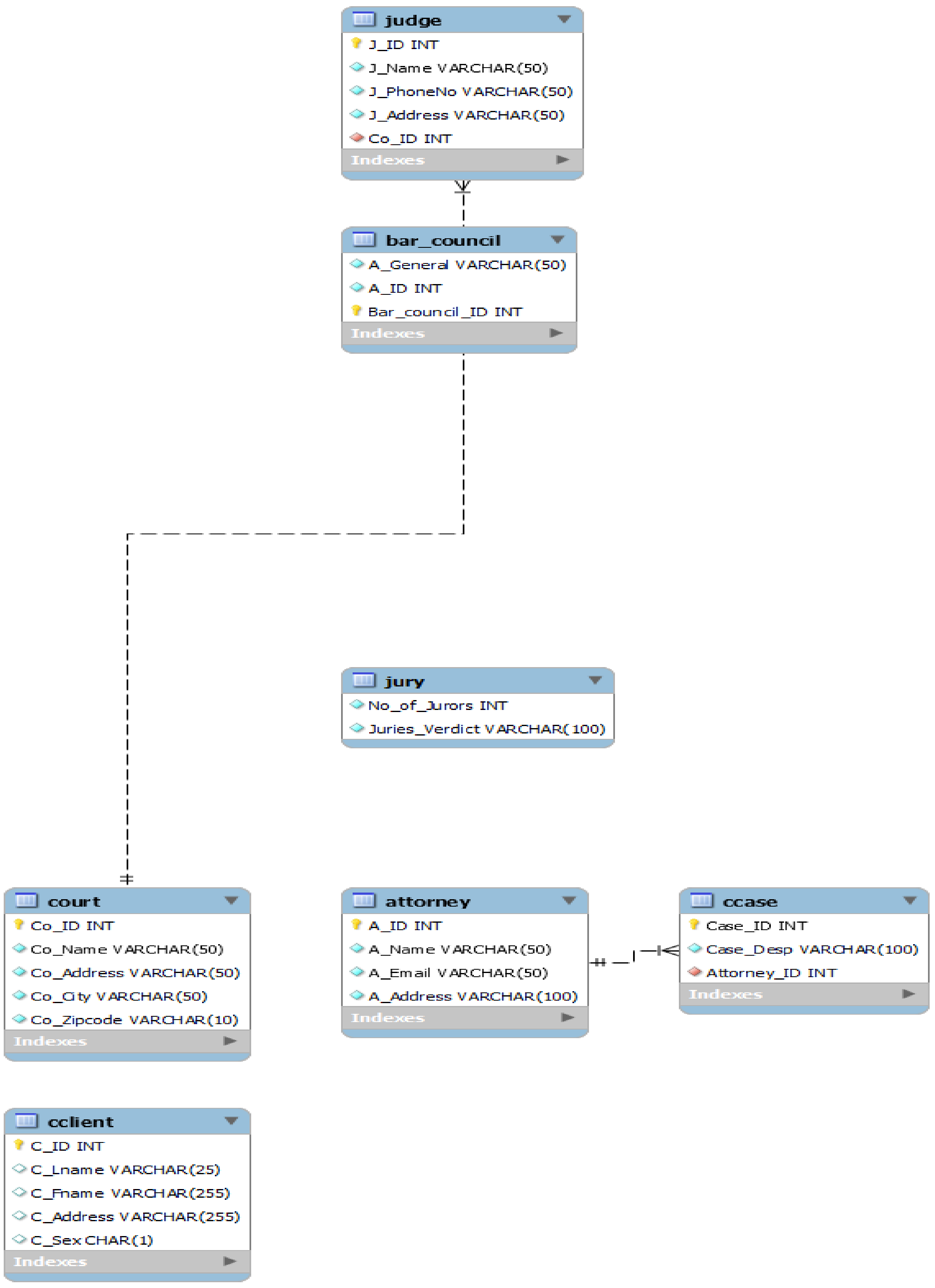
Jury (No_of_Jurors, Juries_Verdicts, Case_ID)

Bar_Council (Bar_council_ID)

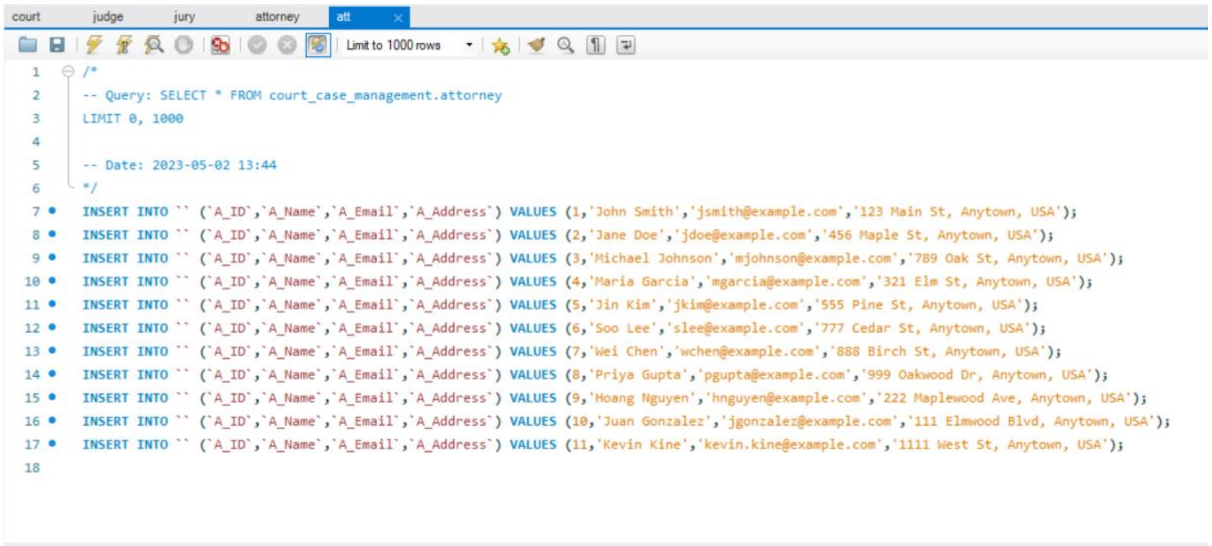
In the above tables, we have removed the transitive dependency between Attorney and Bar_Council by creating a new table, Attorney_Bar_Council, that represents the relationship between them. Now, all tables are in 3NF, and the database is normalized.

Logical Model

From the Normalized Schema we can design the logical design in MySQL Work Bench:

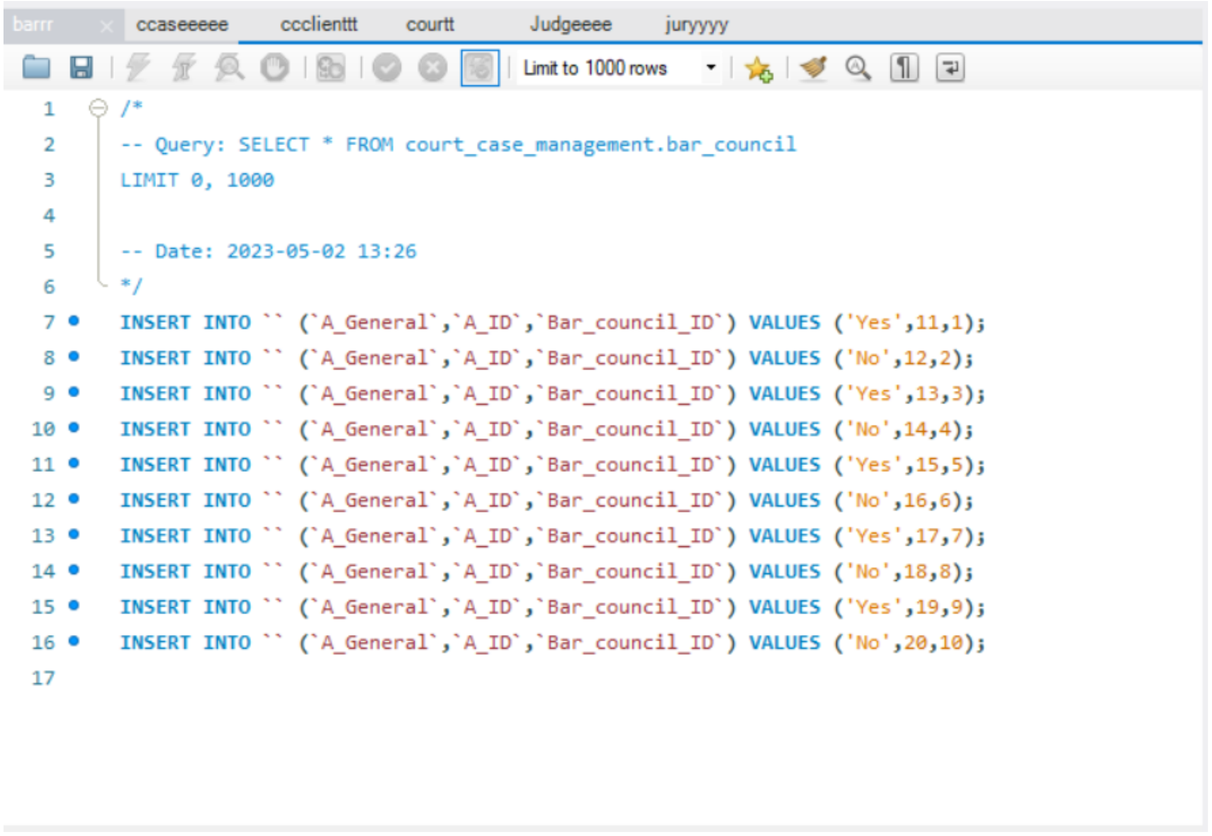


Physical Model of Court case Management System done in MYSQL:



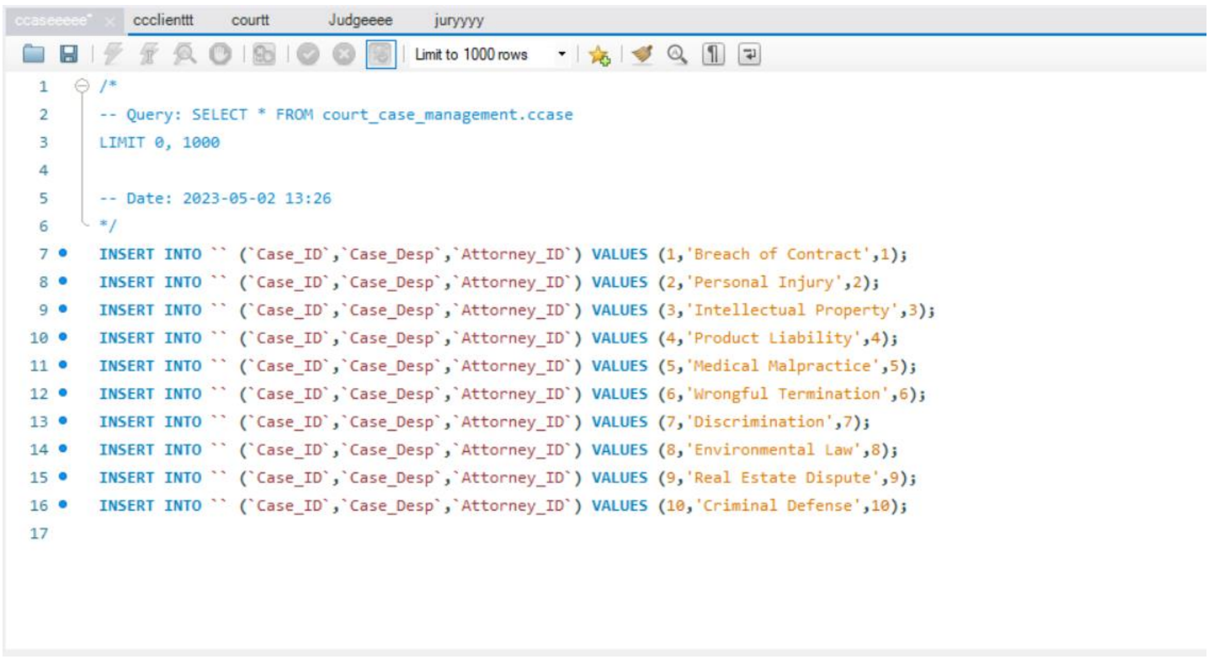
```sql
/*
-- Query: SELECT * FROM court_case_management.attorney
LIMIT 0, 1000

-- Date: 2023-05-02 13:44
*/
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (1,'John Smith','jsmith@example.com','123 Main St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (2,'Jane Doe','jdoe@example.com','456 Maple St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (3,'Michael Johnson','mjohnson@example.com','789 Oak St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (4,'Maria Garcia','mgarcia@example.com','321 Elm St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (5,'Jin Kim','jkim@example.com','555 Pine St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (6,'Soo Lee','slee@example.com','777 Cedar St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (7,'Wei Chen','wchen@example.com','888 Birch St, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (8,'Priya Gupta','pgupta@example.com','999 Oakwood Dr, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (9,'Hoang Nguyen','hnguyen@example.com','222 Maplewood Ave, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (10,'Juan Gonzalez','jgonzalez@example.com','111 Elmwood Blvd, Anytown, USA');
INSERT INTO `` (`A_ID`,`A_Name`,`A_Email`,`A_Address`) VALUES (11,'Kevin Kine','kevin.kine@example.com','1111 West St, Anytown, USA');
```



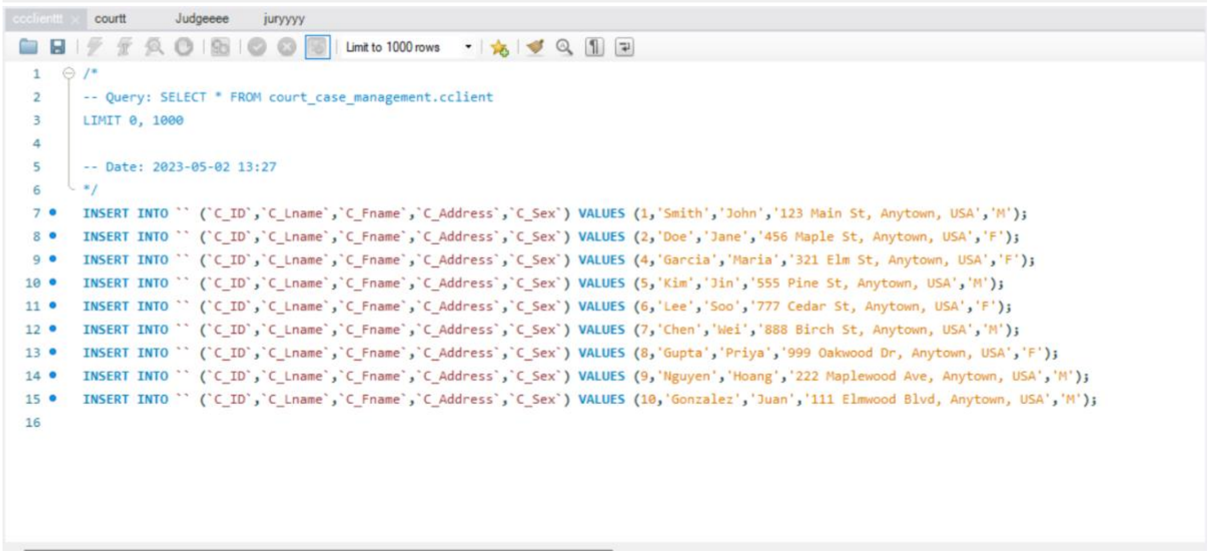
```sql
/*
-- Query: SELECT * FROM court_case_management.bar_council
LIMIT 0, 1000

-- Date: 2023-05-02 13:26
*/
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('Yes',11,1);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('No',12,2);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('Yes',13,3);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('No',14,4);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('Yes',15,5);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('No',16,6);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('Yes',17,7);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('No',18,8);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('Yes',19,9);
INSERT INTO `` (`A_General`,`A_ID`,`Bar_council_ID`) VALUES ('No',20,10);
```



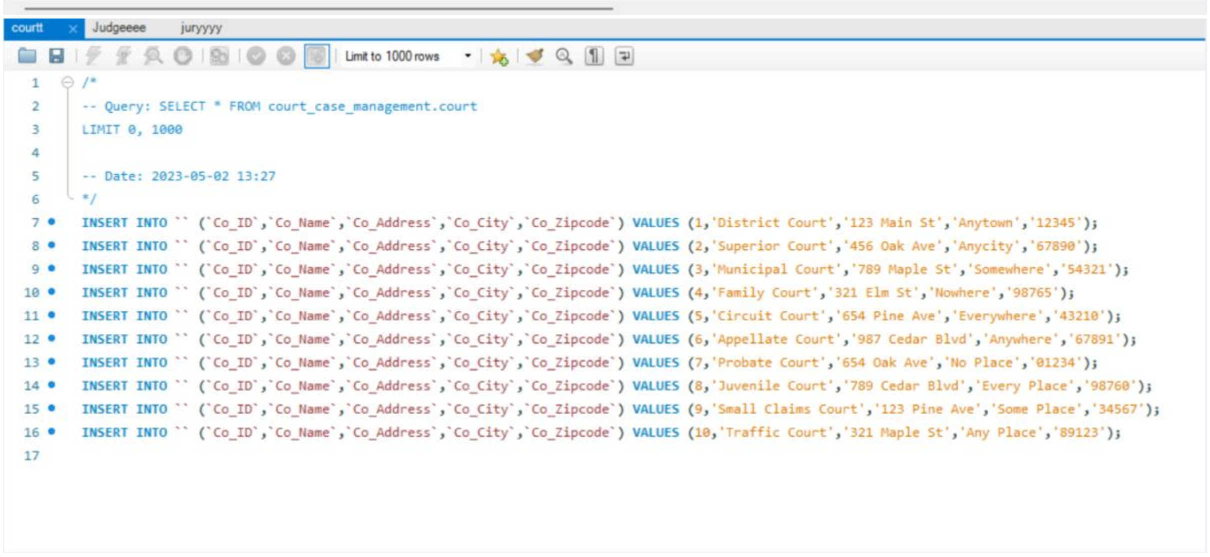
```sql
/*
-- Query: SELECT * FROM court_case_management.ccase
LIMIT 0, 1000

-- Date: 2023-05-02 13:26
*/
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (1,'Breach of Contract',1);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (2,'Personal Injury',2);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (3,'Intellectual Property',3);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (4,'Product Liability',4);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (5,'Medical Malpractice',5);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (6,'Wrongful Termination',6);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (7,'Discrimination',7);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (8,'Environmental Law',8);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (9,'Real Estate Dispute',9);
INSERT INTO `` (`Case_ID`,`Case_Desp`,`Attorney_ID`) VALUES (10,'Criminal Defense',10);
```



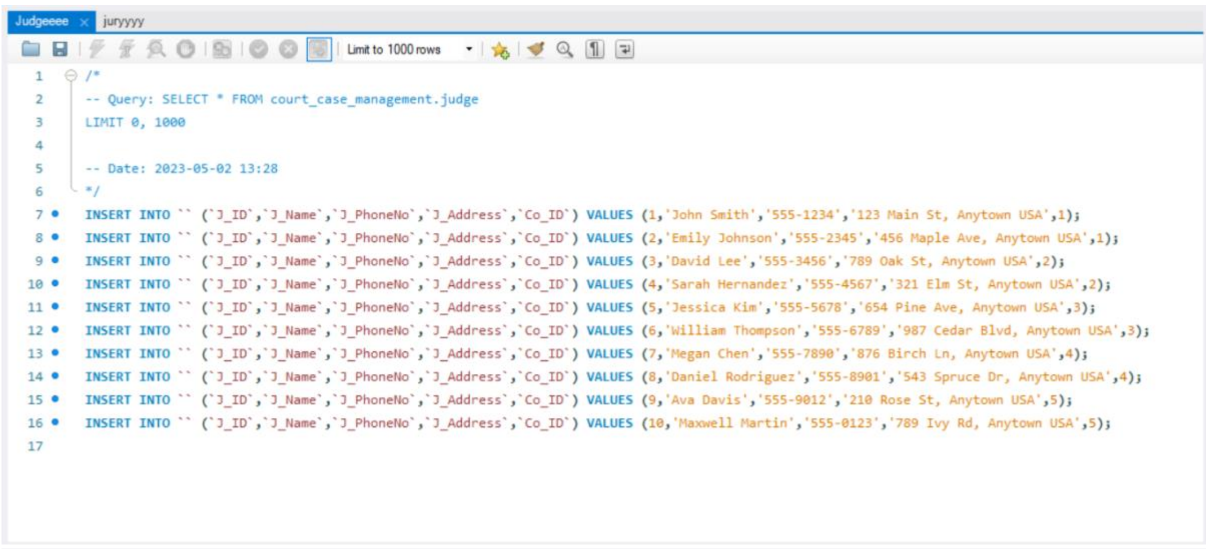
```sql
/*
-- Query: SELECT * FROM court_case_management.cclient
LIMIT 0, 1000

-- Date: 2023-05-02 13:27
*/
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (1,'Smith','John','123 Main St, Anytown, USA','M');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (2,'Doe','Jane','456 Maple St, Anytown, USA','F');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (4,'Garcia','Maria','321 Elm St, Anytown, USA','F');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (5,'Kim','Jin','555 Pine St, Anytown, USA','M');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (6,'Lee','Soo','777 Cedar St, Anytown, USA','F');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (7,'Chen','Wei','888 Birch St, Anytown, USA','M');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (8,'Gupta','Priya','999 Oakwood Dr, Anytown, USA','F');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (9,'Nguyen','Hoang','222 Maplewood Ave, Anytown, USA','M');
INSERT INTO `` (`C_ID`,`C_Lname`,`C_Fname`,`C_Address`,`C_Sex`) VALUES (10,'Gonzalez','Juan','111 Elmwood Blvd, Anytown, USA','M');
```



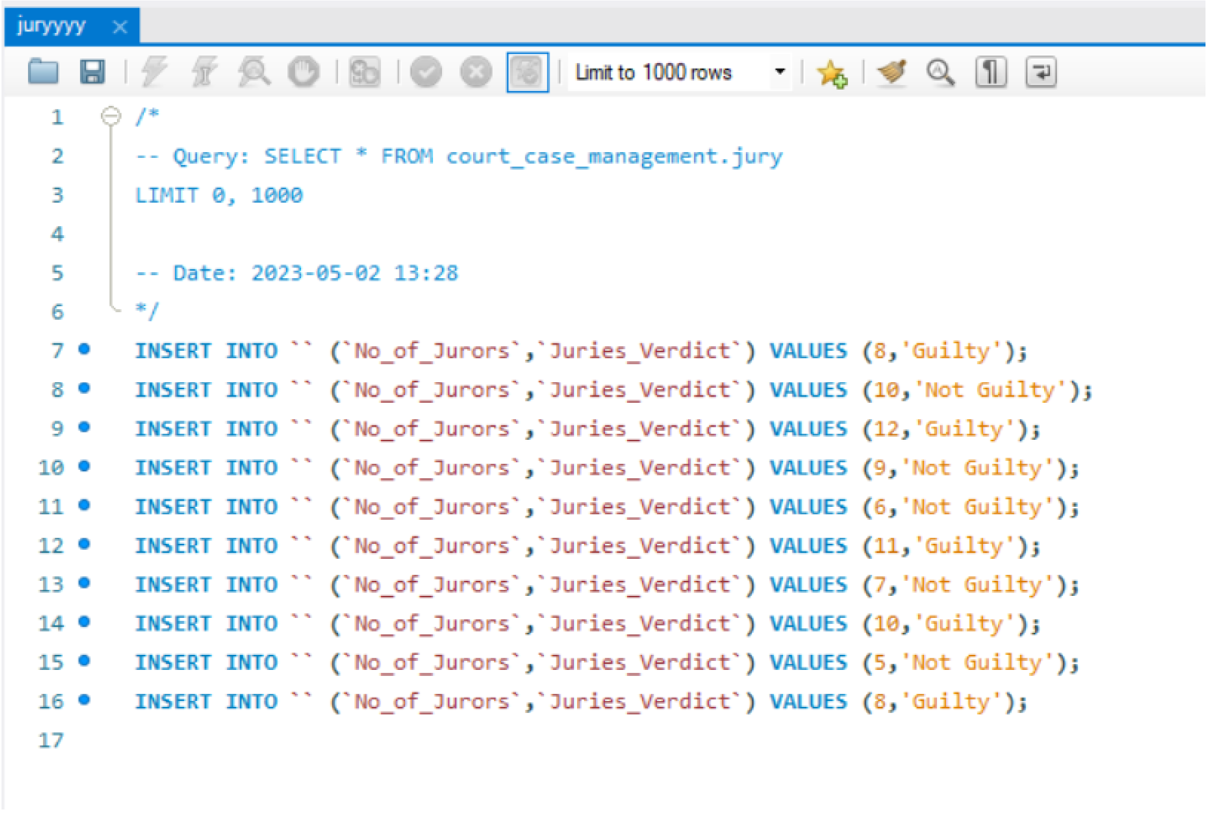
```sql
/*
-- Query: SELECT * FROM court_case_management.court
LIMIT 0, 1000

-- Date: 2023-05-02 13:27
*/
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (1,'District Court','123 Main St','Anytown','12345');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (2,'Superior Court','456 Oak Ave','Anycity','67890');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (3,'Municipal Court','789 Maple St','Somewhere','54321');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (4,'Family Court','321 Elm St','Nowhere','98765');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (5,'Circuit Court','654 Pine Ave','Everywhere','43210');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (6,'Appellate Court','987 Cedar Blvd','Anywhere','67891');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (7,'Probate Court','654 Oak Ave','No Place','01234');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (8,'Juvenile Court','789 Cedar Blvd','Every Place','98760');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (9,'Small Claims Court','123 Pine Ave','Some Place','34567');
INSERT INTO `` (`Co_ID`,`Co_Name`,`Co_Address`,`Co_City`,`Co_Zipcode`) VALUES (10,'Traffic Court','321 Maple St','Any Place','89123');
```



```sql
/*
-- Query: SELECT * FROM court_case_management.judge
LIMIT 0, 1000

-- Date: 2023-05-02 13:28
*/
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (1,'John Smith','555-1234','123 Main St, Anytown USA',1);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (2,'Emily Johnson','555-2345','456 Maple Ave, Anytown USA',1);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (3,'David Lee','555-3456','789 Oak St, Anytown USA',2);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (4,'Sarah Hernandez','555-4567','321 Elm St, Anytown USA',2);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (5,'Jessica Kim','555-5678','654 Pine Ave, Anytown USA',3);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (6,'William Thompson','555-6789','987 Cedar Blvd, Anytown USA',3);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (7,'Megan Chen','555-7890','876 Birch Ln, Anytown USA',4);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (8,'Daniel Rodriguez','555-8901','543 Spruce Dr, Anytown USA',4);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (9,'Ava Davis','555-9012','210 Rose St, Anytown USA',5);
INSERT INTO `` (`J_ID`,`J_Name`,`J_PhoneNo`,`J_Address`,`Co_ID`) VALUES (10,'Maxwell Martin','555-0123','789 Ivy Rd, Anytown USA',5);
```



```sql
/*
-- Query: SELECT * FROM court_case_management.jury
LIMIT 0, 1000

-- Date: 2023-05-02 13:28
*/
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (8,'Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (10,'Not Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (12,'Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (9,'Not Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (6,'Not Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (11,'Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (7,'Not Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (10,'Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (5,'Not Guilty');
INSERT INTO `` (`No_of_Jurors`,`Juries_Verdict`) VALUES (8,'Guilty');
```

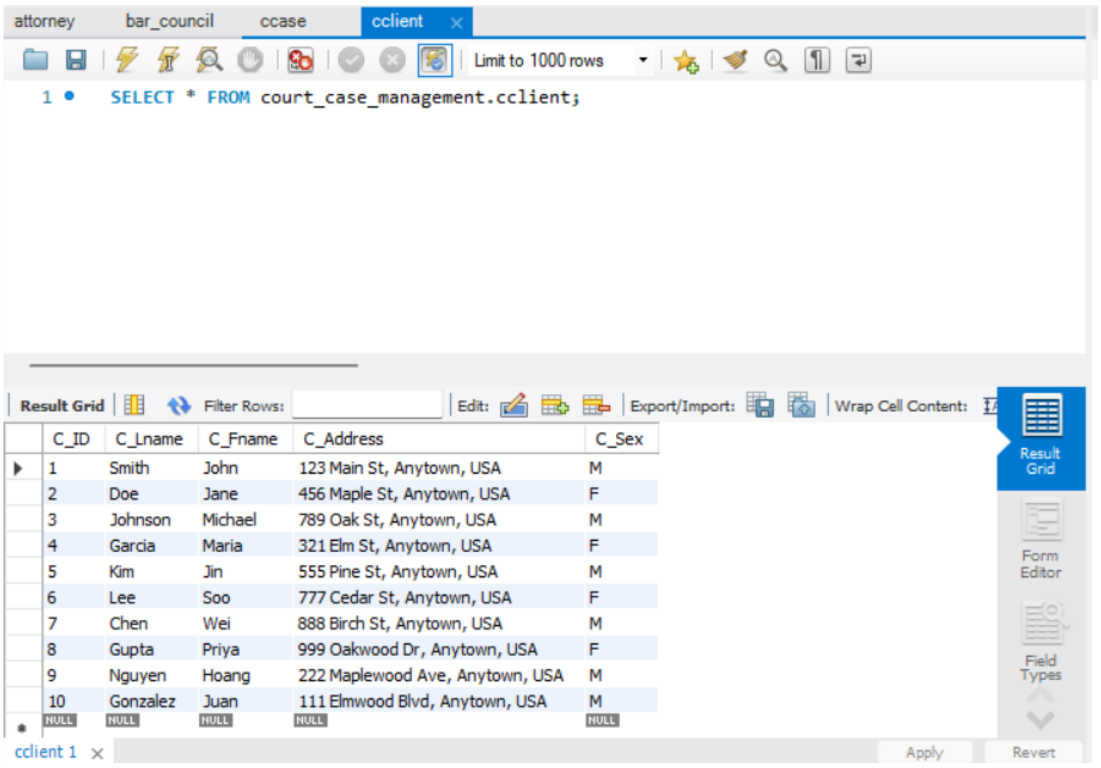
MYSQL

In MySQL, users can create databases, tables, views, stored procedures, triggers, and functions. They can also insert, update, delete, and query data using SQL (Structured Query Language) commands. MySQL also supports transactions, which are sequences of database operations that are executed as a single unit of work to ensure data consistency.

MySQL has a number of advantages, including its ease of use, scalability, reliability, and performance. It is also widely supported by hosting providers and has a large user community, which makes it easy to find resources and support for the platform. Additionally, MySQL is open source, which means that it is free to use and can be customized to meet the specific needs of users.

TABLE VIEWS:

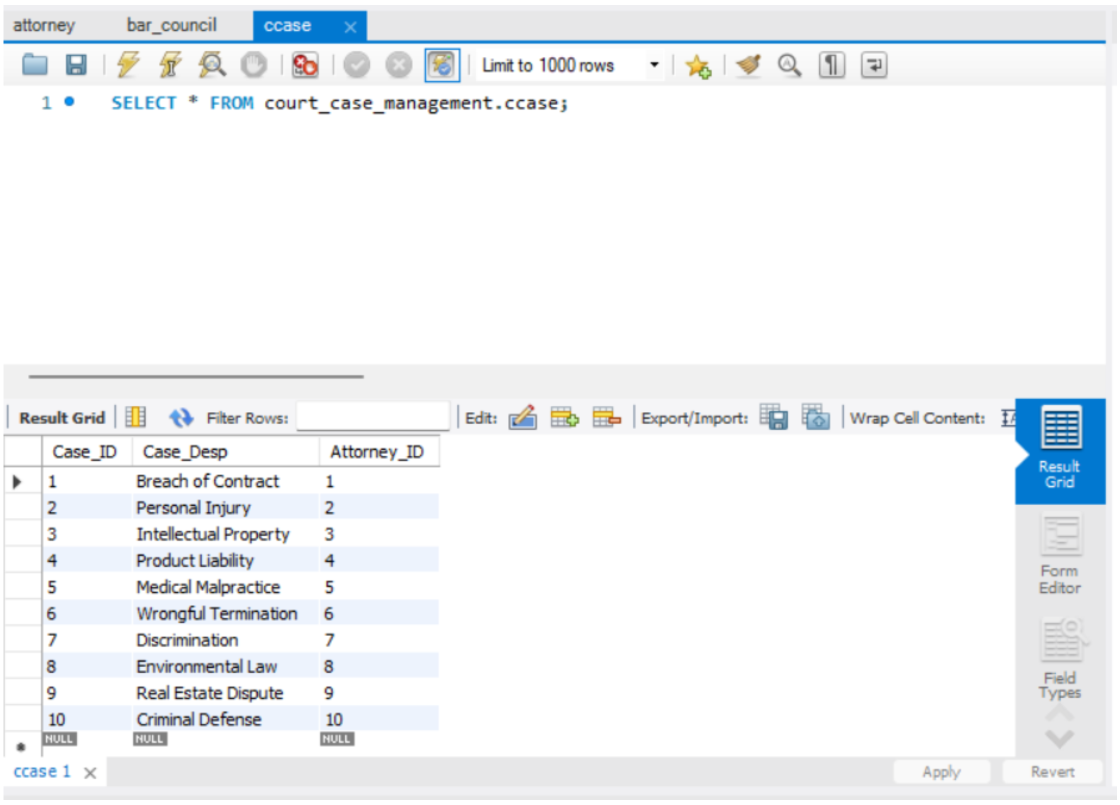
1. Client Table



```
SELECT * FROM court_case_management.cclient;
```

C_ID	C_Lname	C_Fname	C_Address	C_Sex
1	Smith	John	123 Main St, Anytown, USA	M
2	Doe	Jane	456 Maple St, Anytown, USA	F
3	Johnson	Michael	789 Oak St, Anytown, USA	M
4	Garcia	Maria	321 Elm St, Anytown, USA	F
5	Kim	Jin	555 Pine St, Anytown, USA	M
6	Lee	Soo	777 Cedar St, Anytown, USA	F
7	Chen	Wei	888 Birch St, Anytown, USA	M
8	Gupta	Priya	999 Oakwood Dr, Anytown, USA	F
9	Nguyen	Hoang	222 Maplewood Ave, Anytown, USA	M
10	Gonzalez	Juan	111 Elmwood Blvd, Anytown, USA	M
NULL	NULL	NULL	NULL	NULL

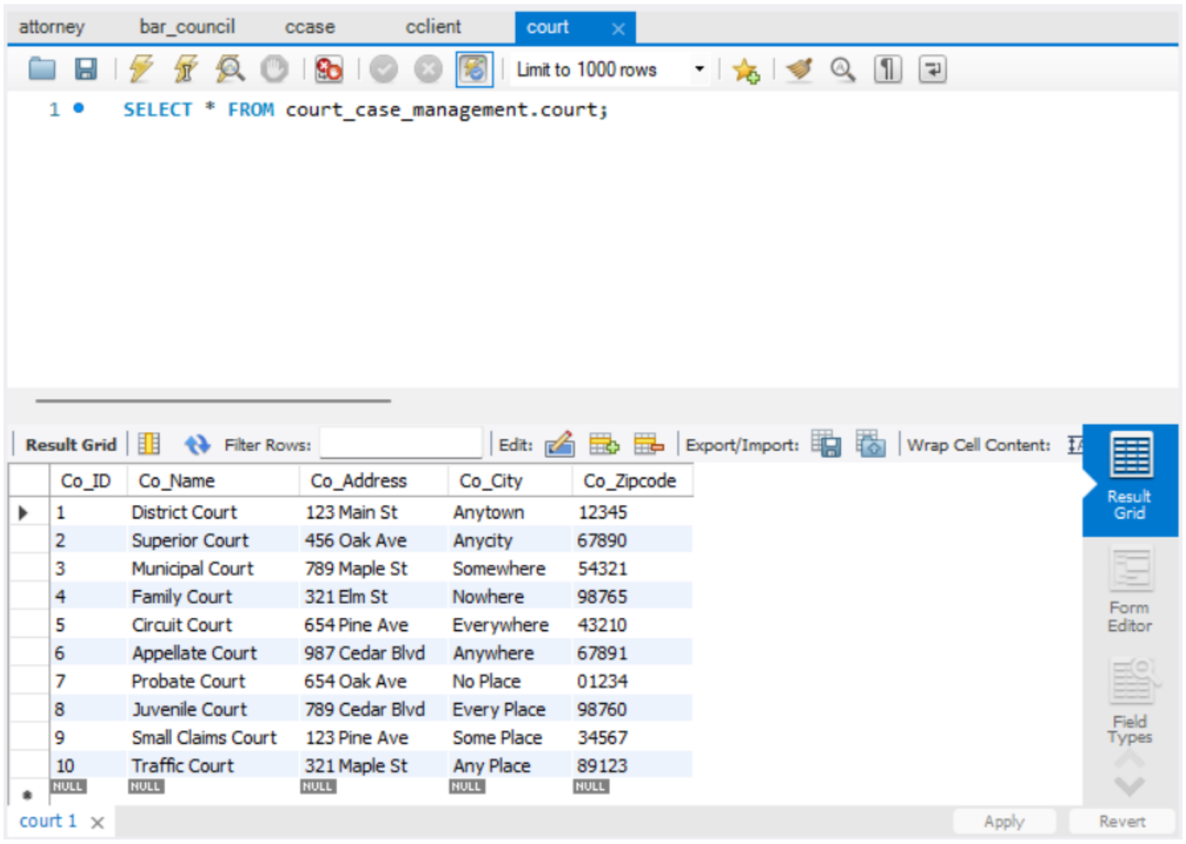
2. Case Table



```
SELECT * FROM court_case_management.ccase;
```

Case_ID	Case_Desp	Attorney_ID
1	Breach of Contract	1
2	Personal Injury	2
3	Intellectual Property	3
4	Product Liability	4
5	Medical Malpractice	5
6	Wrongful Termination	6
7	Discrimination	7
8	Environmental Law	8
9	Real Estate Dispute	9
10	Criminal Defense	10
NULL	NULL	NULL

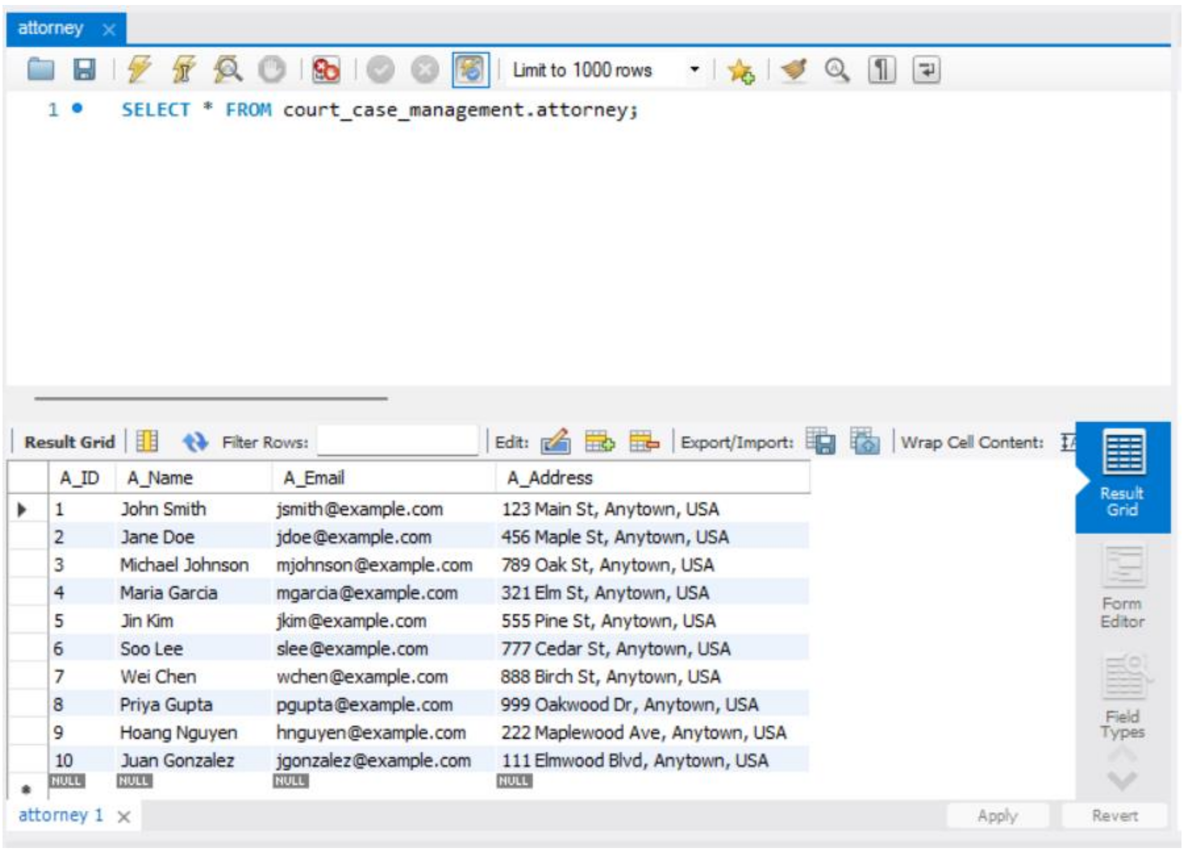
3. Court Table



```
SELECT * FROM court_case_management.court;
```

Co_ID	Co_Name	Co_Address	Co_City	Co_Zipcode
1	District Court	123 Main St	Anytown	12345
2	Superior Court	456 Oak Ave	Anycity	67890
3	Municipal Court	789 Maple St	Somewhere	54321
4	Family Court	321 Elm St	Nowhere	98765
5	Circuit Court	654 Pine Ave	Everywhere	43210
6	Appellate Court	987 Cedar Blvd	Anywhere	67891
7	Probate Court	654 Oak Ave	No Place	01234
8	Juvenile Court	789 Cedar Blvd	Every Place	98760
9	Small Claims Court	123 Pine Ave	Some Place	34567
10	Traffic Court	321 Maple St	Any Place	89123
NULL	NULL	NULL	NULL	NULL

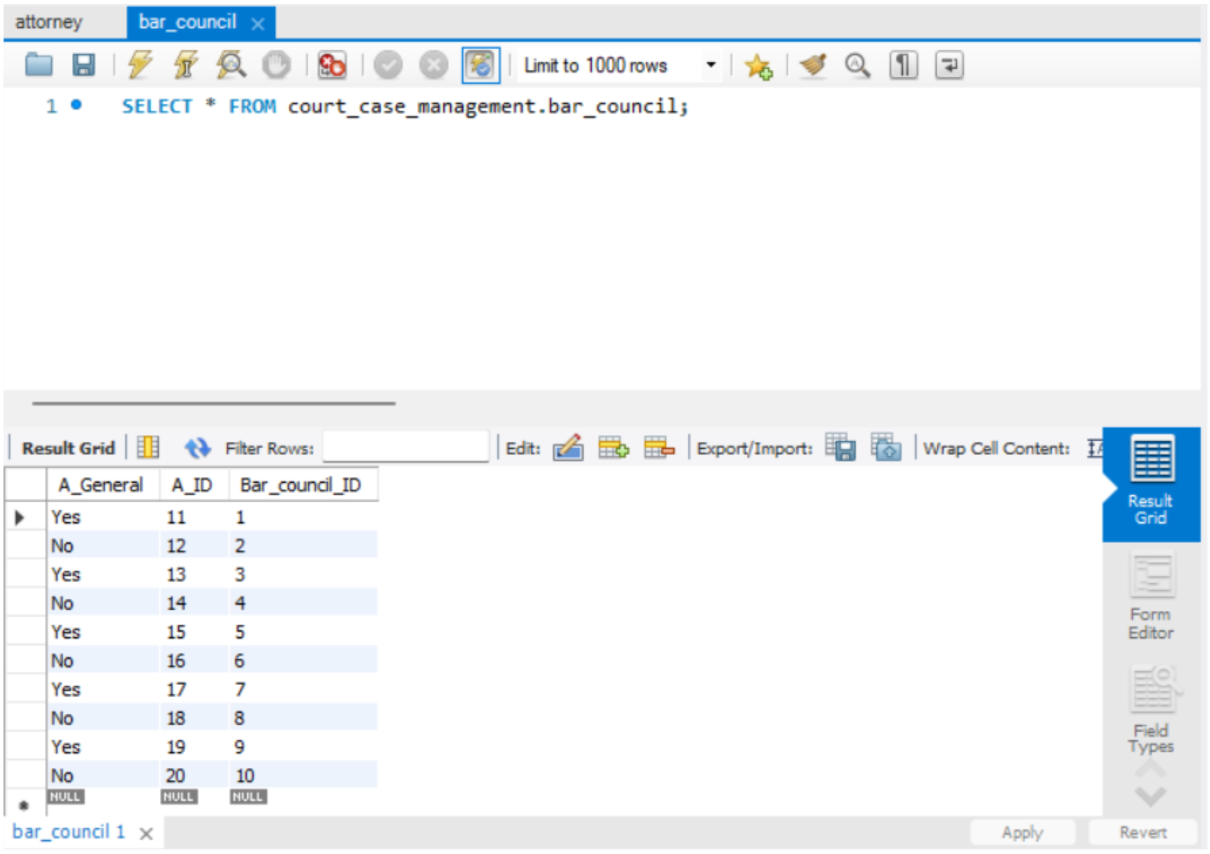
4. Attorney Table



```
SELECT * FROM court_case_management.attorney;
```

A_ID	A_Name	A_Email	A_Address
1	John Smith	jsmith@example.com	123 Main St, Anytown, USA
2	Jane Doe	jdoe@example.com	456 Maple St, Anytown, USA
3	Michael Johnson	mjohnson@example.com	789 Oak St, Anytown, USA
4	Maria Garcia	mgarcia@example.com	321 Elm St, Anytown, USA
5	Jin Kim	jkim@example.com	555 Pine St, Anytown, USA
6	Soo Lee	slee@example.com	777 Cedar St, Anytown, USA
7	Wei Chen	wchen@example.com	888 Birch St, Anytown, USA
8	Priya Gupta	pgupta@example.com	999 Oakwood Dr, Anytown, USA
9	Hoang Nguyen	hnguyen@example.com	222 Maplewood Ave, Anytown, USA
10	Juan Gonzalez	jgonzalez@example.com	111 Elmwood Blvd, Anytown, USA
NULL	NULL	NULL	NULL

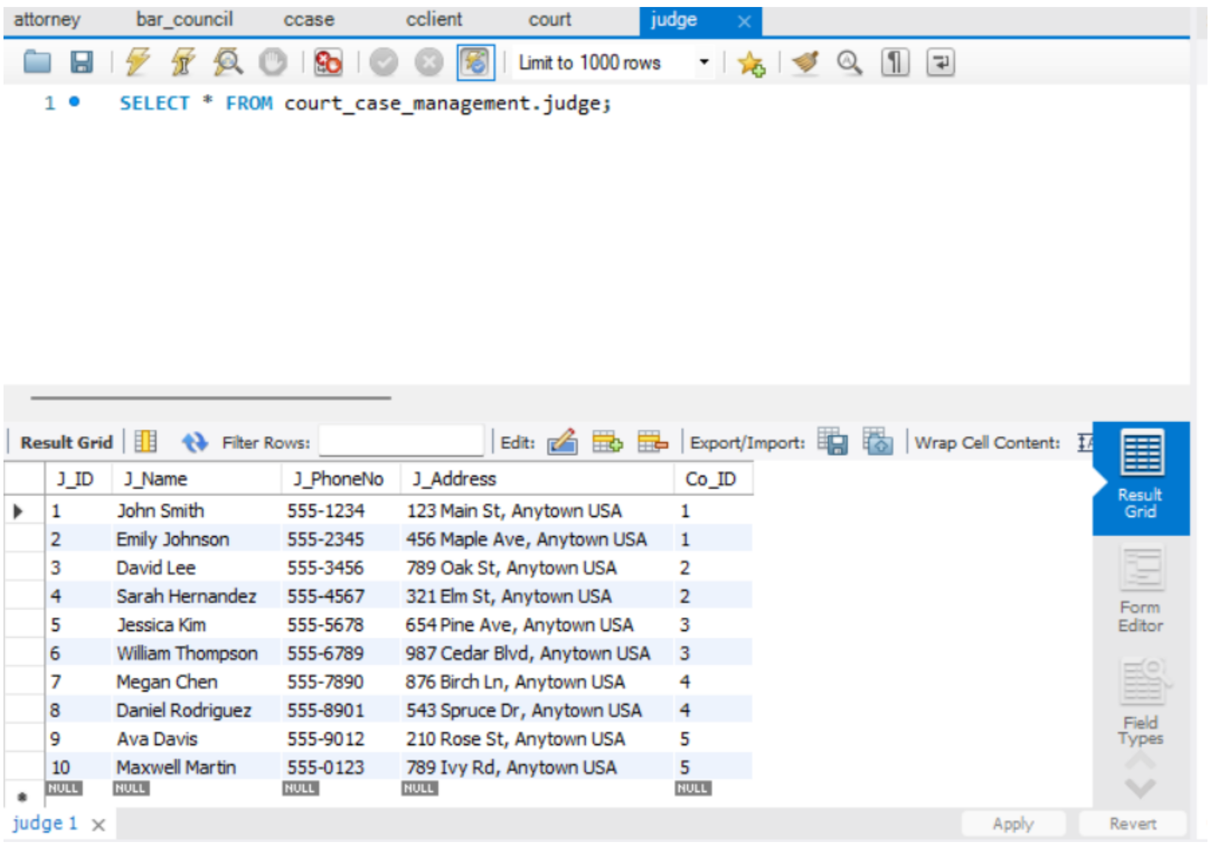
5. Bar Council Table



```
SELECT * FROM court_case_management.bar_council;
```

A_General	A_ID	Bar_council_ID
Yes	11	1
No	12	2
Yes	13	3
No	14	4
Yes	15	5
No	16	6
Yes	17	7
No	18	8
Yes	19	9
No	20	10
NULL	NULL	NULL

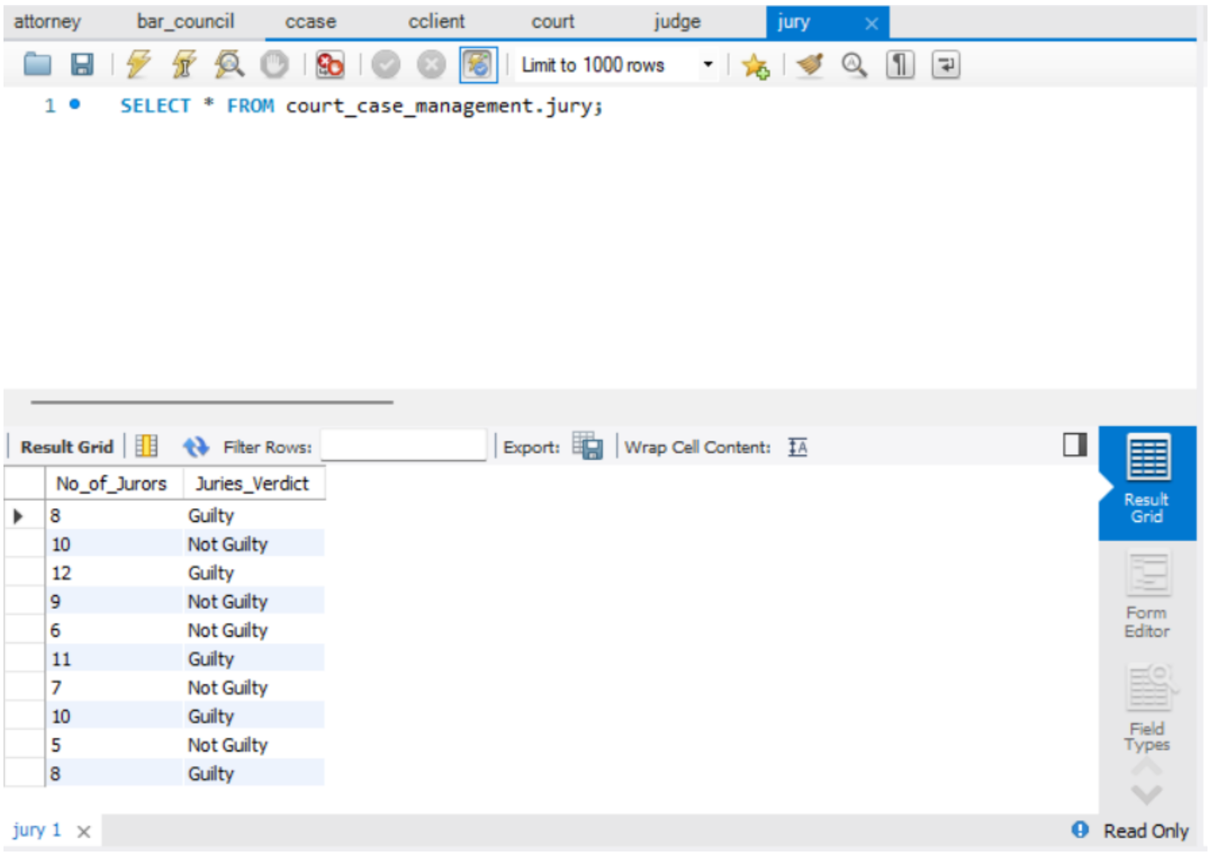
6. Judge Table



```
SELECT * FROM court_case_management.judge;
```

J_ID	J_Name	J_PhoneNo	J_Address	Co_ID
1	John Smith	555-1234	123 Main St, Anytown USA	1
2	Emily Johnson	555-2345	456 Maple Ave, Anytown USA	1
3	David Lee	555-3456	789 Oak St, Anytown USA	2
4	Sarah Hernandez	555-4567	321 Elm St, Anytown USA	2
5	Jessica Kim	555-5678	654 Pine Ave, Anytown USA	3
6	William Thompson	555-6789	987 Cedar Blvd, Anytown USA	3
7	Megan Chen	555-7890	876 Birch Ln, Anytown USA	4
8	Daniel Rodriguez	555-8901	543 Spruce Dr, Anytown USA	4
9	Ava Davis	555-9012	210 Rose St, Anytown USA	5
10	Maxwell Martin	555-0123	789 Ivy Rd, Anytown USA	5
NULL	NULL	NULL	NULL	NULL

7. Jury Table



```
SELECT * FROM court_case_management.jury;
```

No_of_Jurors	Juries_Verdict
8	Guilty
10	Not Guilty
12	Guilty
9	Not Guilty
6	Not Guilty
11	Guilty
7	Not Guilty
10	Guilty
5	Not Guilty
8	Guilty

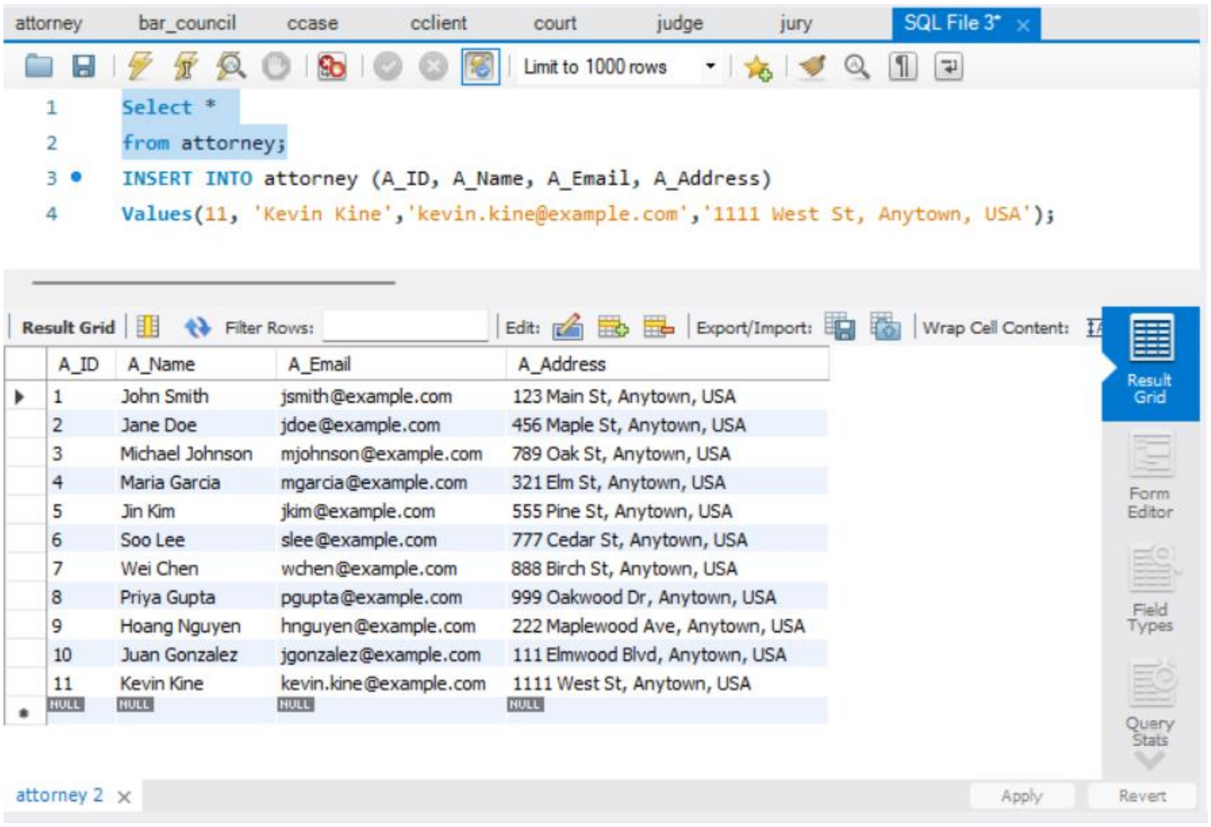
Data Manipulation

Data manipulation refers to the process of modifying, organizing, and analyzing data in a database. It involves performing various operations such as adding, deleting, updating, and retrieving data from the database.

Some common data manipulation operations are:

1. Insertion: This operation is used to add new data to the database. The INSERT statement is used for this purpose.
2. Deletion: This operation is used to remove data from the database. The DELETE statement is used for this purpose.
3. Updating: This operation is used to modify existing data in the database. The UPDATE statement is used for this purpose.
4. Retrieval: This operation is used to retrieve data from the database. The SELECT statement is used for this purpose.
5. Sorting: This operation is used to sort data in the database based on a specific criterion. The ORDER BY clause is used for this purpose.
6. Filtering: This operation is used to retrieve specific data from the database based on a condition. The WHERE clause is used for this purpose.
7. Aggregation: This operation is used to perform calculations on the data in the database. The GROUP BY and aggregate functions (e.g. SUM, AVG, MAX, MIN) are used for this purpose.

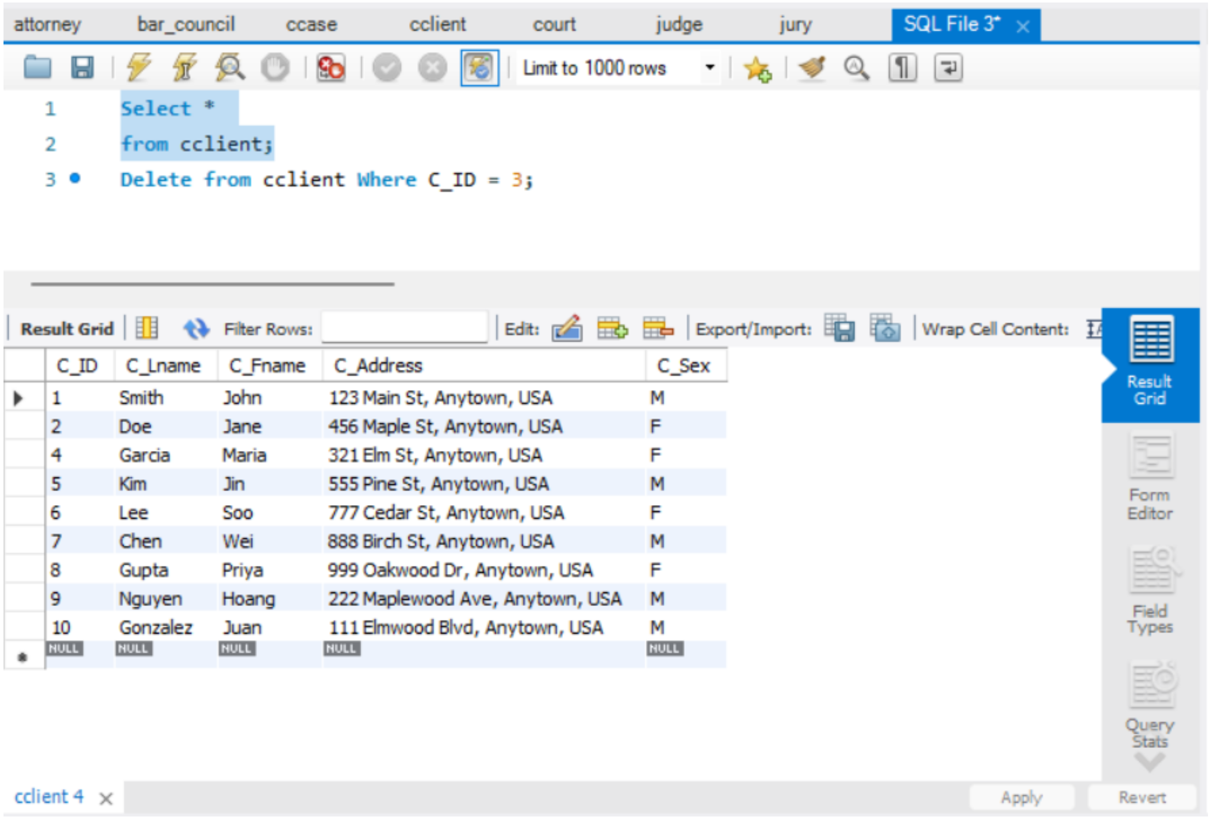
1. INSERT



```sql
Select *
from attorney;
INSERT INTO attorney (A_ID, A_Name, A_Email, A_Address)
Values(11, 'Kevin Kine','kevin.kine@example.com','1111 West St, Anytown, USA');
```

A_ID	A_Name	A_Email	A_Address
1	John Smith	jsmith@example.com	123 Main St, Anytown, USA
2	Jane Doe	jdoe@example.com	456 Maple St, Anytown, USA
3	Michael Johnson	mjohnson@example.com	789 Oak St, Anytown, USA
4	Maria Garcia	mgarcia@example.com	321 Elm St, Anytown, USA
5	Jin Kim	jkim@example.com	555 Pine St, Anytown, USA
6	Soo Lee	slee@example.com	777 Cedar St, Anytown, USA
7	Wei Chen	wchen@example.com	888 Birch St, Anytown, USA
8	Priya Gupta	pgupta@example.com	999 Oakwood Dr, Anytown, USA
9	Hoang Nguyen	hnguyen@example.com	222 Maplewood Ave, Anytown, USA
10	Juan Gonzalez	jgonzalez@example.com	111 Elmwood Blvd, Anytown, USA
11	Kevin Kine	kevin.kine@example.com	1111 West St, Anytown, USA
NULL	NULL	NULL	NULL

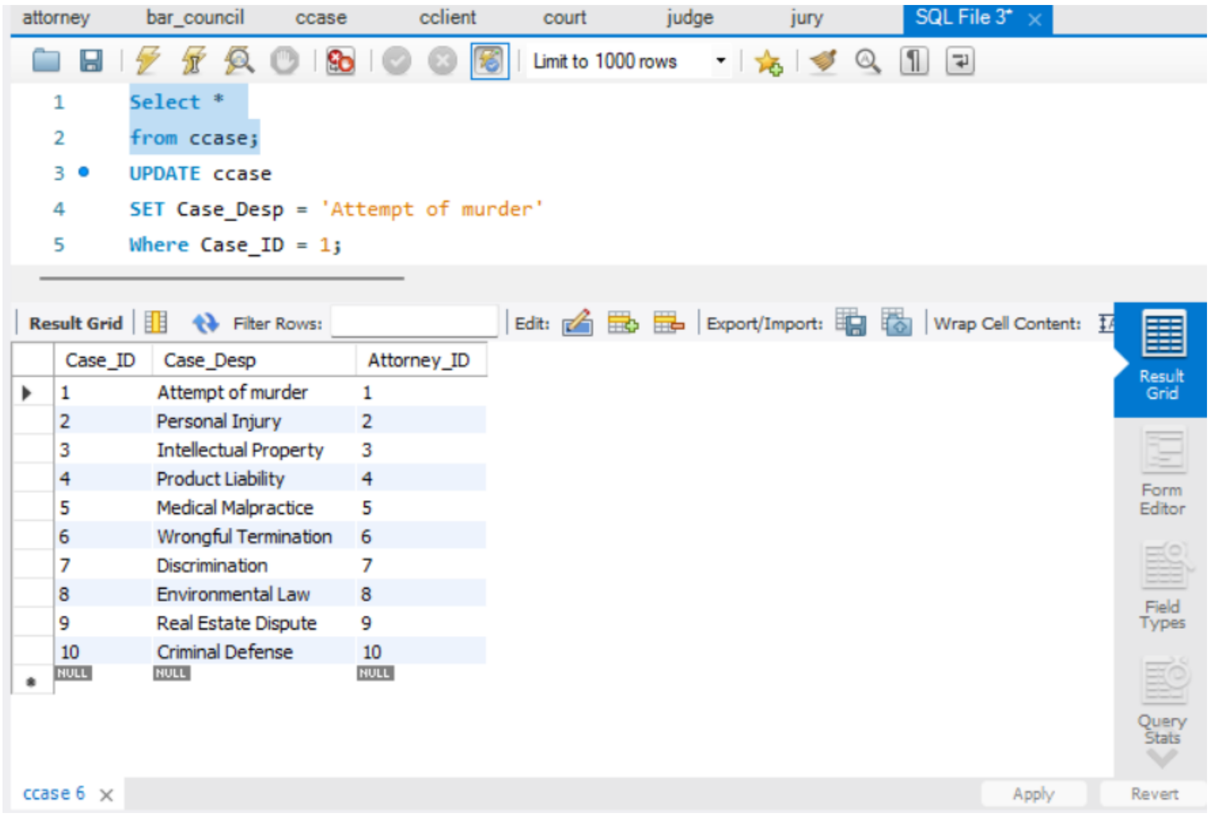
2. DELETE



```sql
Select *
from cclient;
Delete from cclient Where C_ID = 3;
```

C_ID	C_Lname	C_Fname	C_Address	C_Sex
1	Smith	John	123 Main St, Anytown, USA	M
2	Doe	Jane	456 Maple St, Anytown, USA	F
4	Garcia	Maria	321 Elm St, Anytown, USA	F
5	Kim	Jin	555 Pine St, Anytown, USA	M
6	Lee	Soo	777 Cedar St, Anytown, USA	F
7	Chen	Wei	888 Birch St, Anytown, USA	M
8	Gupta	Priya	999 Oakwood Dr, Anytown, USA	F
9	Nguyen	Hoang	222 Maplewood Ave, Anytown, USA	M
10	Gonzalez	Juan	111 Elmwood Blvd, Anytown, USA	M
NULL	NULL	NULL	NULL	NULL

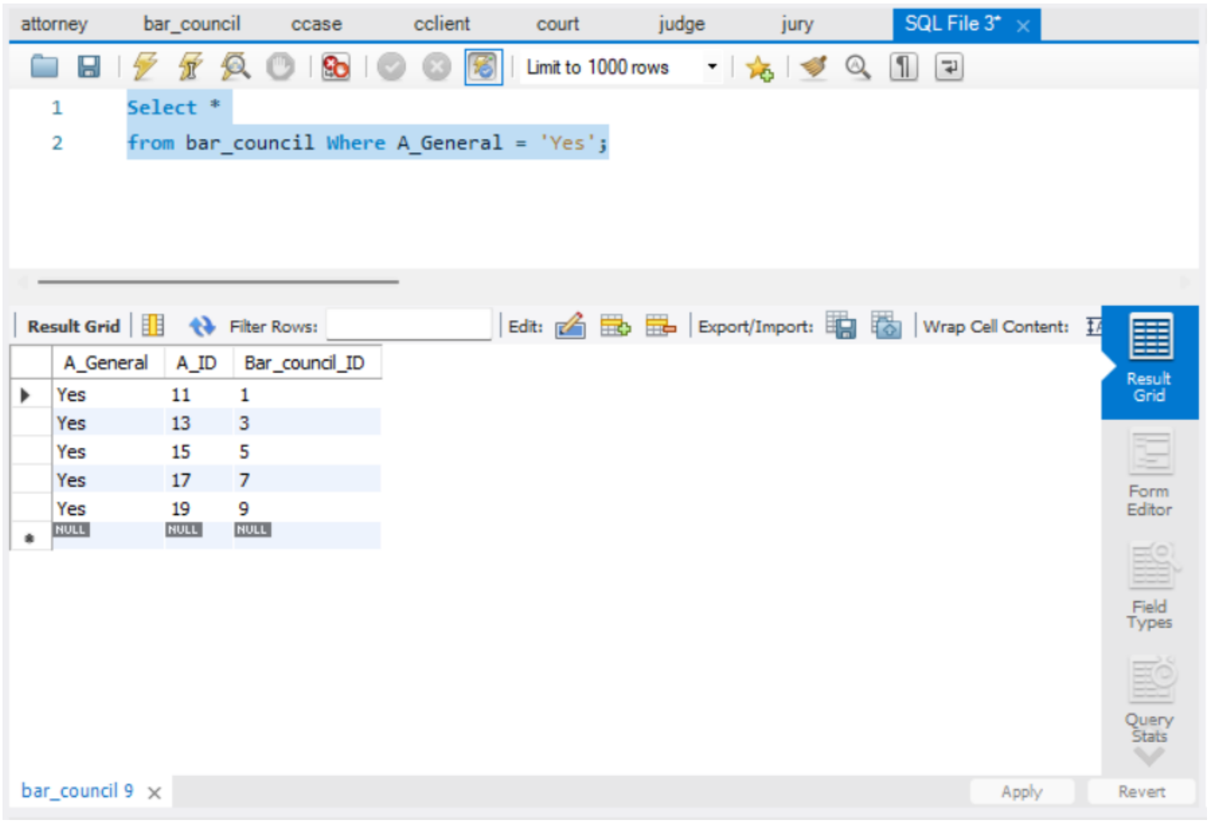
3. UPDATE



```sql
Select *
from ccase;
UPDATE ccase
SET Case_Desp = 'Attempt of murder'
Where Case_ID = 1;
```

Case_ID	Case_Desp	Attorney_ID
1	Attempt of murder	1
2	Personal Injury	2
3	Intellectual Property	3
4	Product Liability	4
5	Medical Malpractice	5
6	Wrongful Termination	6
7	Discrimination	7
8	Environmental Law	8
9	Real Estate Dispute	9
10	Criminal Defense	10
NULL	NULL	NULL

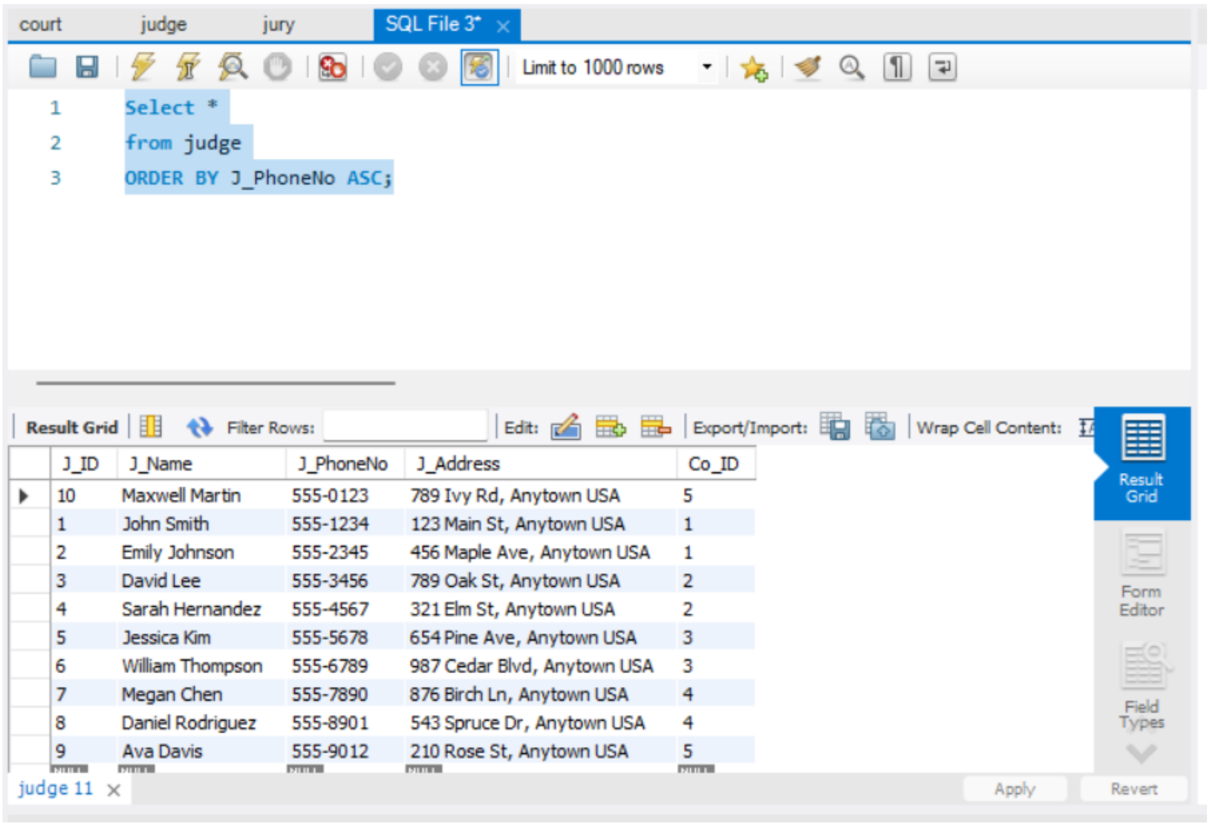
4. RETRIEVAL



```sql
Select *
from bar_council Where A_General = 'Yes';
```

A_General	A_ID	Bar_council_ID
Yes	11	1
Yes	13	3
Yes	15	5
Yes	17	7
Yes	19	9
NULL	NULL	NULL

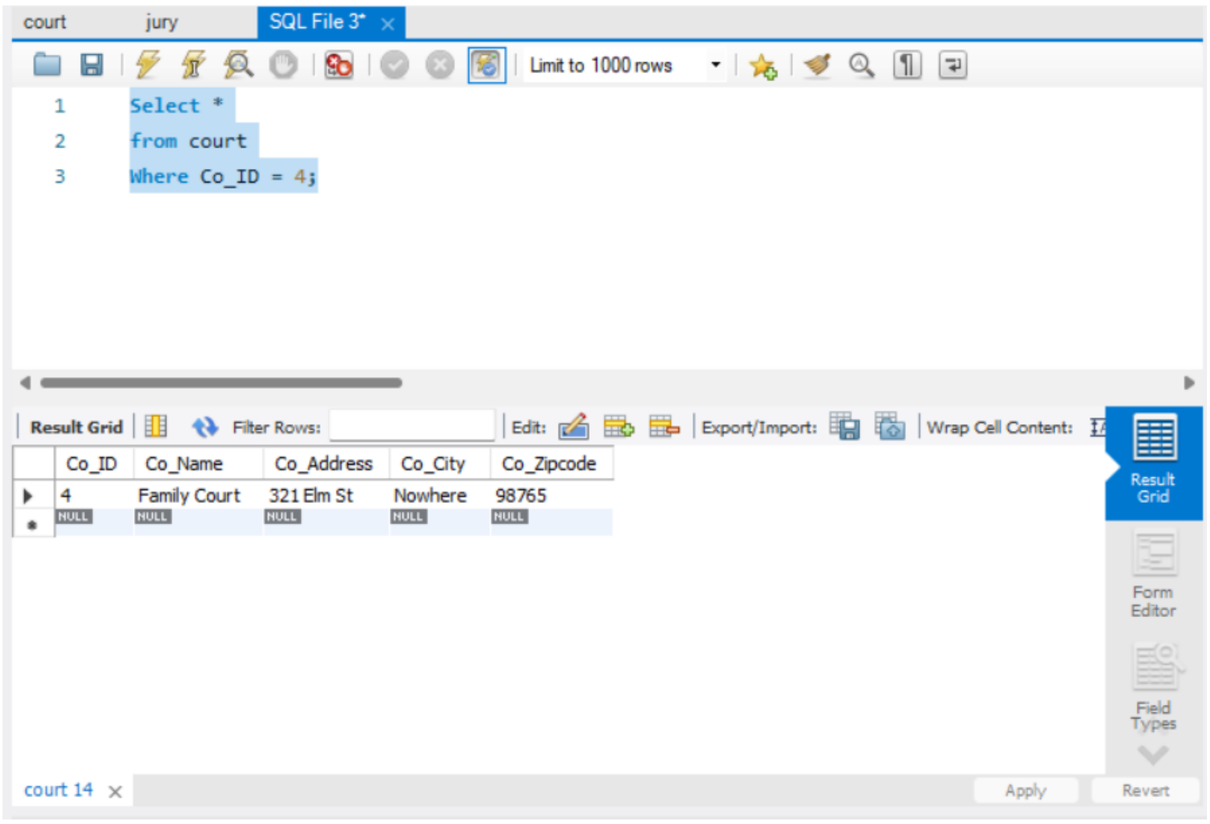
5. SORT



```
Select *
from judge
ORDER BY J_PhoneNo ASC;
```

J_ID	J_Name	J_PhoneNo	J_Address	Co_ID
10	Maxwell Martin	555-0123	789 Ivy Rd, Anytown USA	5
1	John Smith	555-1234	123 Main St, Anytown USA	1
2	Emily Johnson	555-2345	456 Maple Ave, Anytown USA	1
3	David Lee	555-3456	789 Oak St, Anytown USA	2
4	Sarah Hernandez	555-4567	321 Elm St, Anytown USA	2
5	Jessica Kim	555-5678	654 Pine Ave, Anytown USA	3
6	William Thompson	555-6789	987 Cedar Blvd, Anytown USA	3
7	Megan Chen	555-7890	876 Birch Ln, Anytown USA	4
8	Daniel Rodriguez	555-8901	543 Spruce Dr, Anytown USA	4
9	Ava Davis	555-9012	210 Rose St, Anytown USA	5

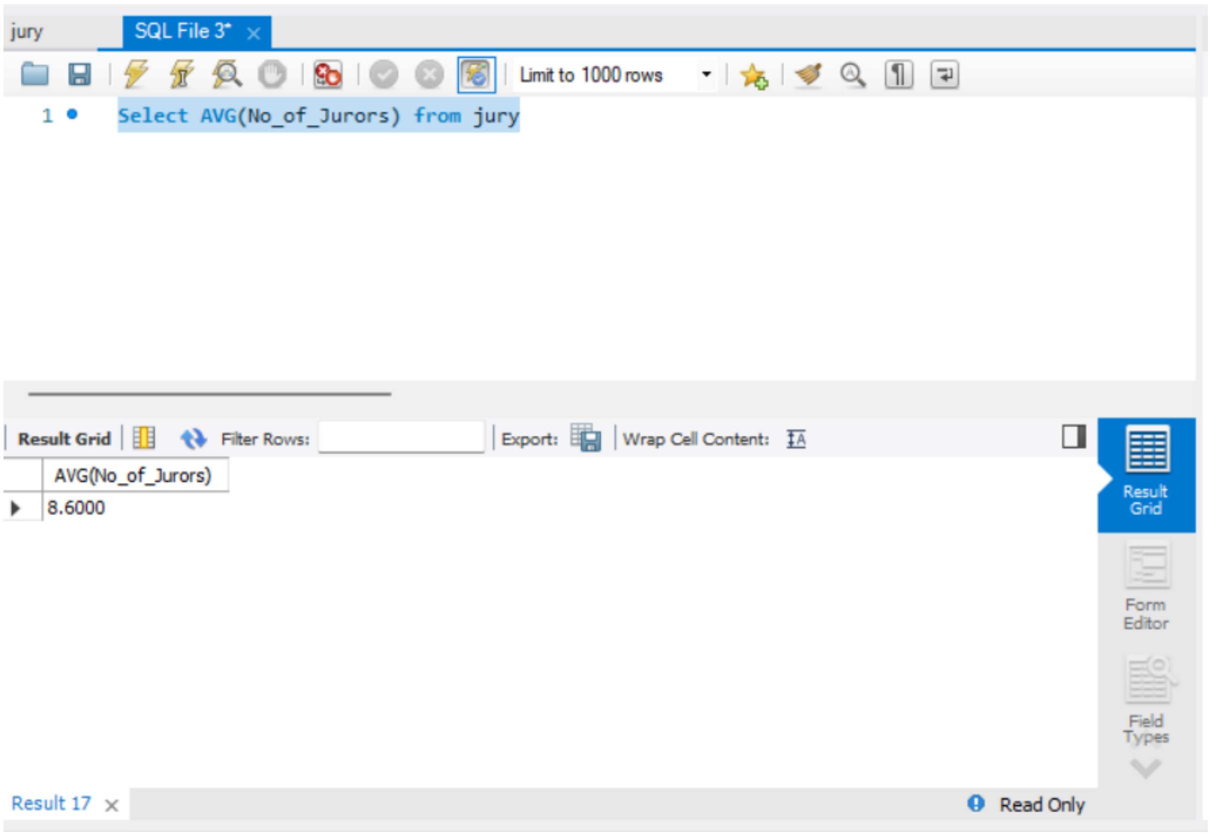
6. FILTER



```
Select *
from court
Where Co_ID = 4;
```

Co_ID	Co_Name	Co_Address	Co_City	Co_Zipcode
4	Family Court	321 Elm St	Nowhere	98765
NULL	NULL	NULL	NULL	NULL

7. AGGREGATION



```
Select AVG(No_of_Jurors) from jury
```

```
AVG(No_of_Jurors)
8.6000
```

Conclusion

The Court Case Management System project offers details on the various judges employed by the Court, the number of clients served by various attorneys, the issues faced by clients, and a summary of their cases. The following are some benefits of this project:

- Less redundancy.
- Information retrieval is simple.
- Updates and removals are easily accomplished.