Cover page

Multimedia University

CCP6114 Programming Fundamentals 2430

Lecture section:TC3L
Tutorial section:TT5L
Group number: G15
Group leader student name: Ayden

Num	Student ID	Student Name by alphabetical order	Task Descriptions	Percentage (%)
1	242UC24570	Ayden Bin Wira	Pseudocode, Create database and view database Create table	
2	242UC24571	Dania	Insert rows to the table Table support two data types	
3	242UC244G9	Nur Qistina Atashah	Documentation screenshots, create base, view table in csv mode	
4	241UC2417Q	Vinesh	Flowchart	

Every student is responsible for 100% (task percentage) of this group assignment work.

Mark sheet checklist (30%)
Assignment programming and documentation (30%)
You are required to submit assignment milestone 1 to your respective tutor also before the submission deadline.
Also document all your assignment tasks with this marking table that contain cover page, table of contents, page numbering, inputs, outputs, screenshots, explanations, and others.

Criteria	**Max**	**A1**	**A2**	**Mark**
Q1. Create database and view database name Create table, view table name Table supports two data types i.e. INT, TEXT Insert rows to the table View table in csv mode	5	*	*	?
Q2. Reading from a file, outputting to screen, writing to a file (0 if no files used or no screen outputs)	3		*	?
Q3. Update table rows and view table Delete table rows and view table	4		*	?
Q4. Count and output number of rows in the table	2		*	?
Q5. Must use vectors or arrays, functions or classes, to store file output contents	2		*	?
Q6. Inline comments, function or class comments, indentation, following proper C++ naming and styling conventions Any violation is penalized by a reduction of 1 mark.	2		*	?
Q7. The program demonstrates error handlings. [0: Below Expectation, 1: Within Expectation, 2: Exceed Expectation]	2		*	?
Q8. Correct structured diagrams	2		*	?
Q9. Correct flowcharts or pseudocodes with explanations for all the file input statements. Any missing flowchart or pseudocode will cause you to lose 1 mark.	2		*	?
Q10. Sample file inputs at least 3, their screen outputs, their file outputs with screenshots and explanations.	3		*	?
Q11. User documentation done and is coherence with the all implementations. Any missing input statement will cause you to lose 1 mark.	3		*	?

Total	30			?

Additional comments

You are required to fill in your task percentage and task descriptions.
Every student is responsible for 100% (task percentage) of this group assignment work.

Student 1

Student ID	?
Student name	?
Task percentage	?
Task descriptions	?
Total score (30m)	?

Student 2

Student ID	?
Student name	?
Task percentage	?
Task descriptions	?
Total score (30m)	?

Student 3

Student ID	?
Student name	?
Task percentage	?
Task descriptions	?
Total score (30m)	?

Student 4

Student ID	?
Student name	?
Task percentage	?
Task descriptions	?

Total score (30m)	?

Each feature will be evaluated based on documentation, fulfilment of requirements, correctness, compilation without warnings and errors, error free during runtime, error handlings, quality of comments, user friendliness, good coding format and style.

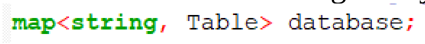
Table of contents with page numbers and links

Question Section

Q01, Q09, Q11 [5] Database name, table name, table of two data types, insert table rows, view tables

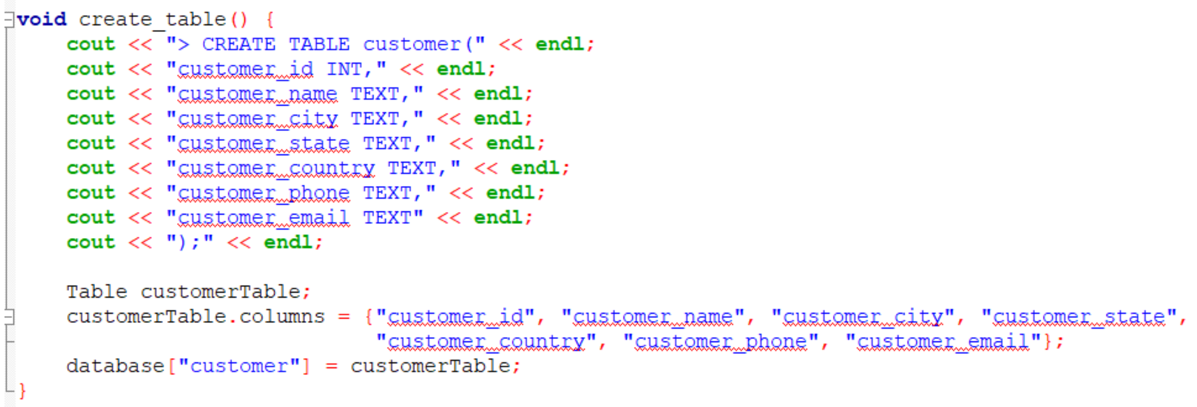
Create database and view database name

- Declaration of database globally



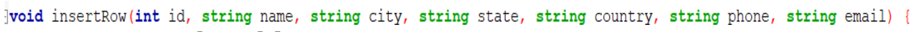
```
map<string, Table> database;
```

Create table, view table name



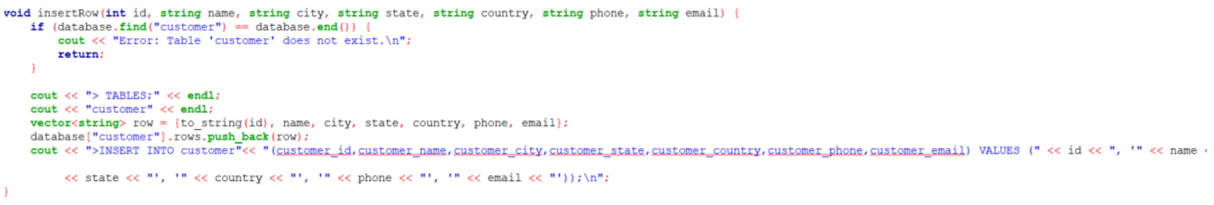
```
void create_table() {
    cout << "> CREATE TABLE customer(" << endl;
    cout << "customer_id INT," << endl;
    cout << "customer_name TEXT," << endl;
    cout << "customer_city TEXT," << endl;
    cout << "customer_state TEXT," << endl;
    cout << "customer_country TEXT," << endl;
    cout << "customer_phone TEXT," << endl;
    cout << "customer_email TEXT" << endl;
    cout << ");" << endl;

    Table customerTable;
    customerTable.columns = {"customer_id", "customer_name", "customer_city", "customer_state",
                             "customer_country", "customer_phone", "customer_email"};
    database["customer"] = customerTable;
}
```

Table supports two data types i.e. INT, TEXT



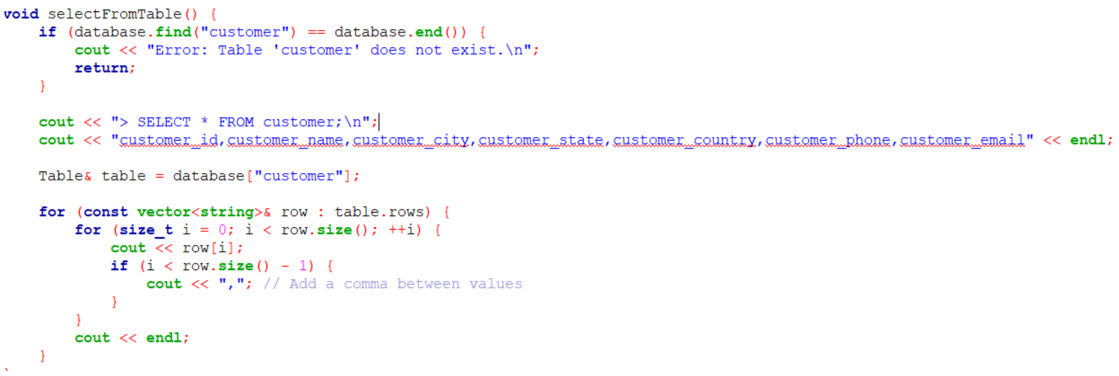
```
void insertRow(int id, string name, string city, string state, string country, string phone, string email) {
```

Insert rows to the table



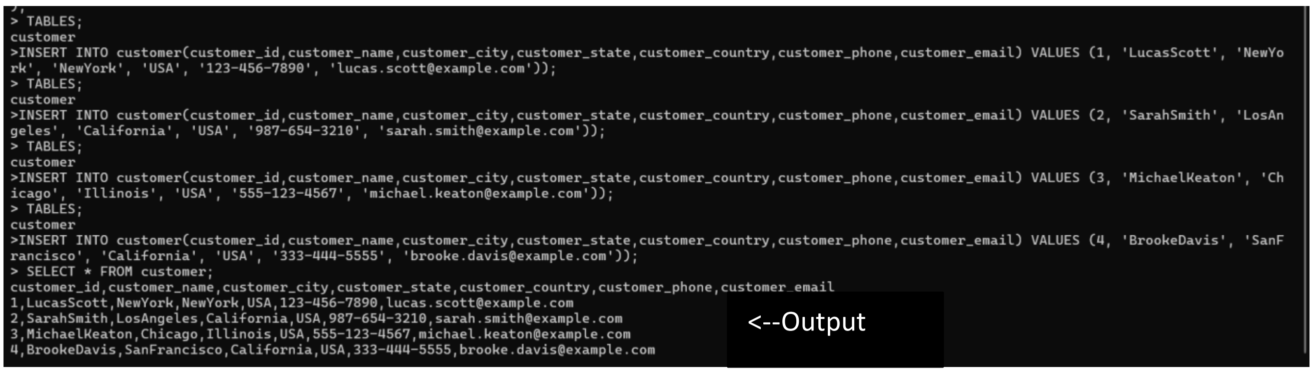
```
void insertRow(int id, string name, string city, string state, string country, string phone, string email) {
    if (database.find("customer") == database.end()) {
        cout << "Error: Table 'customer' does not exist.\n";
        return;
    }

    cout << "> TABLES;" << endl;
    cout << "customer" << endl;
    vector<string> row = {to_string(id), name, city, state, country, phone, email};
    database["customer"].rows.push_back(row);
    cout << ">INSERT INTO customer"<< "(customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email) VALUES (" << id << ", '" << name <

         << state << "', '" << country << "', '" << phone << "', '" << email << "');\n";
}
```

View table in csv mode



```
void selectFromTable() {
    if (database.find("customer") == database.end()) {
        cout << "Error: Table 'customer' does not exist.\n";
        return;
    }

    cout << "> SELECT * FROM customer;\n";
    cout << "customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email" << endl;

    Table& table = database["customer"];

    for (const vector<string>& row : table.rows) {
        for (size_t i = 0; i < row.size(); ++i) {
            cout << row[i];
            if (i < row.size() - 1) {
                cout << ","; // Add a comma between values
            }
        }
        cout << endl;
    }
```

Screenshots (inputs, outputs), explanations



```
);
> TABLES;
customer
>INSERT INTO customer(customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email) VALUES (1, 'LucasScott', 'NewYo
rk', 'NewYork', 'USA', '123-456-7890', 'lucas.scott@example.com'));
> TABLES;
customer
>INSERT INTO customer(customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email) VALUES (2, 'SarahSmith', 'LosAn
geles', 'California', 'USA', '987-654-3210', 'sarah.smith@example.com'));
> TABLES;
customer
>INSERT INTO customer(customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email) VALUES (3, 'MichaelKeaton', 'Ch
icago', 'Illinois', 'USA', '555-123-4567', 'michael.keaton@example.com'));
> TABLES;
customer
>INSERT INTO customer(customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email) VALUES (4, 'BrookeDavis', 'SanF
rancisco', 'California', 'USA', '333-444-5555', 'brooke.davis@example.com'));
> SELECT * FROM customer;
customer_id,customer_name,customer_city,customer_state,customer_country,customer_phone,customer_email
1,LucasScott,NewYork,NewYork,USA,123-456-7890,lucas.scott@example.com
2,SarahSmith,LosAngeles,California,USA,987-654-3210,sarah.smith@example.com
3,MichaelKeaton,Chicago,Illinois,USA,555-123-4567,michael.keaton@example.com
4,BrookeDavis,SanFrancisco,California,USA,333-444-5555,brooke.davis@example.com
```

Pseudocode parts, explanations

```
START

//DEFINE Structure: Table
DEFINE Columns as Vector of strings
DEFINE Rows as Vector of vector of strings

//Declare Global Database
Map<string, Table>

//Functions to create fileOutput
FUNCTION create_fileoutput
PRINT "CREATE fileOutput1.txt;"
END FUNCTION

//Functions for filepath
FUNCTION database_fileInput(filePath):
PRINT "DATABASES;"
PRINT filePath
END FUNCTION

//Function to create table
FUNCTION create_table
PRINT SQL command to create table NAMED Customer
DEFINE Table with columns AS [ "customer_id" : "integer", "customer_name" : "text", "customer_city" :
"text", "customer_state" : "text", "customer_country" : "text", "customer_phone" : "integer",
"customer_email" : "text"]
   ADD Table to database["customer"]
 END FUNCTION

//Function to insert rows
FUNCTION insertRow(id, name, city, state, country, phone, email):
IF "customer" table does not exist:
PRINT "Error: Table 'customer' does not exist.\n"
RETURN
PRINT SQL INSERT command
ADD row to database["customer"].rows

//Function to select from table
FUNCTION selectFromTable:
IF "customer" table does not exist:
PRINT "Error: Table 'customer' does not exist.\n"
RETURN
PRINT SQL SELECT command
PRINT Columns
PRINT Rows

//View table in csv mode
FOR EACH row IN table.rows DO
FOR i FROM 0 TO size of row - 1 DO
PRINT row[i]
IF i IS NOT the last index THEN
PRINT ","
END IF
END FOR
 PRINT a new line
END FOR

//MAIN FUNCTION
DECLARE filePath AS STRING = "C:/CCP6114_2430_TC3L_G15/fileInput1.mdb"
CALL create_fileoutput()
CALL database_fileInput(filePath)
CALL create_table()

CALL insertRow(1, "LucasScott", "NewYork", "NewYork", "USA", "123-456-7890",
"lucas.scott@example.com")
CALL insertRow(2, "SarahSmith", "LosAngeles", "California", "USA", "987-654-3210",
"sarah.smith@example.com")
CALL insertRow(3, "MichaelKeaton", "Chicago", "Illinois", "USA", "555-123-4567",
"michael.keaton@example.com")
CALL insertRow(4, "BrookeDavis", "SanFrancisco", "California", "USA", "333-444-5555",
"brooke.davis@example.com")

CALL selectFromTable()

RETURN 0
```

Q02, Q09, Q11 [3] Reading from a file, outputting to screen, writing to a file
0 if no files used or no screen outputs

Screenshots (inputs, outputs), explanations
?

Pseudocode parts, explanations

```
```

Q03, Q09, Q11 [4] Update table rows, delete table rows, view table

Screenshots (inputs, outputs), explanations
?

Pseudocode parts, explanations

Q04, Q09, Q11 [2] Count and output number of rows in the table

Screenshots (inputs, outputs), explanations
?

Pseudocode parts, explanations

Q05, Q11 [2] Must use vectors or arrays, functions or classes, to store file output contents

Screenshots (inputs, outputs), explanations
?

Code parts, explanations

Q06, Q11 [2] Inline comments, function or class comments, indentation, proper C++ naming with styling conventions
Any violation is penalized by a reduction of 1 mark.

Screenshots (inputs, outputs), explanations
?

Code parts, explanations

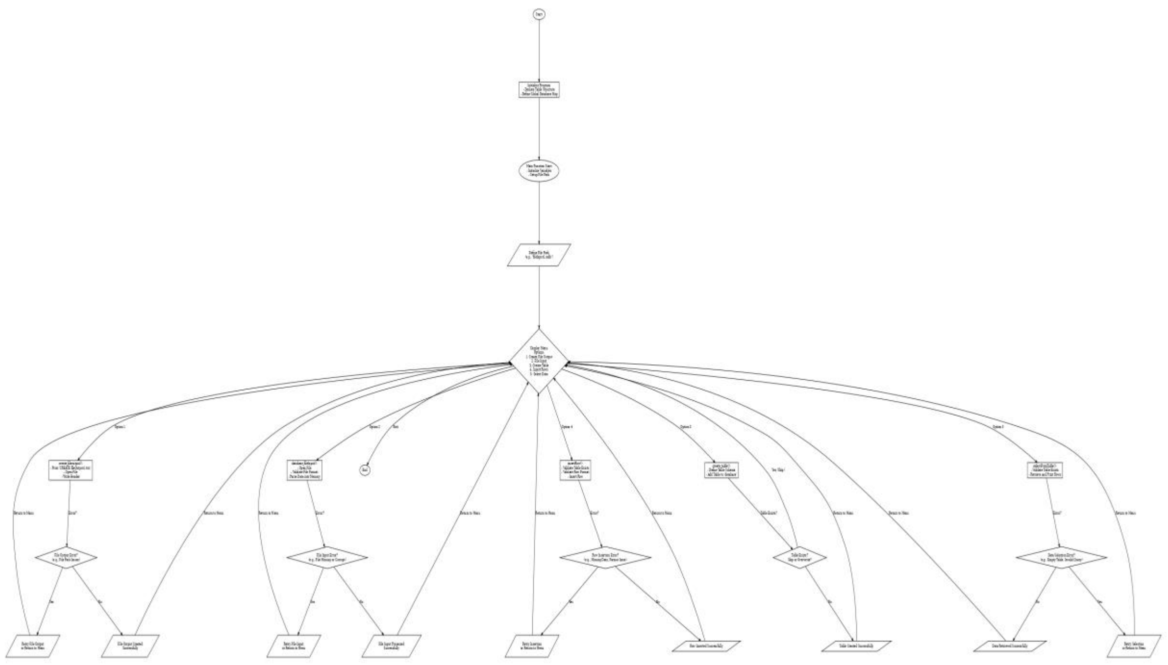
Q07, Q09, Q11 [2] The program demonstrates error handlings
[0: Below Expectation, 1: Within Expectation, 2: Exceed Expectation]

Screenshots (inputs, outputs), explanations
?

Pseudocode parts, explanations

Q08, Q11 [2] Structured diagrams

Figures, explanations



Q10, Q11 [3] Three sample input files, step by step screenshot outputs, output files, explanations

Sample 1 for A1
input file
filename: fileInput1.mdb

```
CREATE fileOutput1.txt;
DATABASES;

CREATE TABLE customer(
customer_id INT,
customer_name TEXT,
customer_city TEXT,
customer_state TEXT,
customer_country TEXT,
customer_phone TEXT,
customer_email TEXT
);
TABLES;

INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(1,'name1','city1','state1','country1','phone1','email1');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(2,'name2','city2','state2','country2','phone2','email2');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(3,'name3','city3','state3','country3','phone3','email3');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(4,'name4','city4','state4','country4','phone4','email4');

SELECT * FROM customer;
```

output file and screen output
filename: fileOutput1.txt

```
> CREATE fileOutput1.txt;
> DATABASES;
C:\mariadb\fileInput1.mdb
> CREATE TABLE customer(
customer_id INT,
customer_name TEXT,
customer_city TEXT,
customer_state TEXT,
customer_country TEXT,
customer_phone TEXT,
customer_email TEXT
);
> TABLES;
customer
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(1,'name1','city1','state1','country1','phone1','email1');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(2,'name2','city2','state2','country2','phone2','email2');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(3,'name3','city3','state3','country3','phone3','email3');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(4,'name4','city4','state4','country4','phone4','email4');
> SELECT * FROM customer;
customer_id,customer_name,customer_city,customer_state,customer_country,customer_ph
one,customer_email
1,name1,city1,state1,country1,phone1,email1
2,name2,city2,state2,country2,phone2,email2
3,name3,city3,state3,country3,phone3,email3
4,name4,city4,state4,country4,phone4,email4
```

Sample 1 for A2
input file
filename: fileInput2.mdb

```
CREATE fileOutput2.txt;
DATABASES;

CREATE TABLE customer(
customer_id INT,
customer_name TEXT,
customer_city TEXT,
customer_state TEXT,
customer_country TEXT,
customer_phone TEXT,
customer_email TEXT
);

INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(1,'name1','city1','state1','country1','phone1','email1');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(2,'name2','city2','state2','country2','phone2','email2');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(3,'name3','city3','state3','country3','phone3','email3');
INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(4,'name4','city4','state4','country4','phone4','email4');
SELECT * FROM customer;

TABLES;

UPDATE customer SET customer_email='email333' WHERE customer_id=3;
SELECT * FROM customer;

DELETE FROM customer WHERE customer_id=4;
SELECT * FROM customer;

SELECT COUNT(*) FROM customer;
```

output file and screen output
filename: fileOutput2.txt

```
> CREATE fileOutput2.txt;
> DATABASES;
C:\mariadb\fileInput2.mdb
> CREATE TABLE customer(
customer_id INT,
customer_name TEXT,
customer_city TEXT,
customer_state TEXT,
customer_country TEXT,
customer_phone TEXT,
customer_email TEXT
);
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(1,'name1','city1','state1','country1','phone1','email1');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(2,'name2','city2','state2','country2','phone2','email2');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(3,'name3','city3','state3','country3','phone3','email3');
> INSERT INTO
customer(customer_id,customer_name,customer_city,customer_state,customer_country,cus
tomer_phone,customer_email) VALUES
(4,'name4','city4','state4','country4','phone4','email4');
> SELECT * FROM customer;
customer_id,customer_name,customer_city,customer_state,customer_country,customer_ph
one,customer_email
1,name1,city1,state1,country1,phone1,email1
2,name2,city2,state2,country2,phone2,email2
3,name3,city3,state3,country3,phone3,email3
4,name4,city4,state4,country4,phone4,email4
> TABLES;
customer
> UPDATE customer SET customer_email='email333' WHERE customer_id=3;
> SELECT * FROM customer;
customer_id,customer_name,customer_city,customer_state,customer_country,customer_ph
one,customer_email
1,name1,city1,state1,country1,phone1,email1
2,name2,city2,state2,country2,phone2,email2
3,name3,city3,state3,country3,phone3,email333
4,name4,city4,state4,country4,phone4,email4
> DELETE FROM customer WHERE customer_id=4;
> SELECT * FROM customer;
customer_id,customer_name,customer_city,customer_state,customer_country,customer_ph
one,customer_email
1,name1,city1,state1,country1,phone1,email1
2,name2,city2,state2,country2,phone2,email2
3,name3,city3,state3,country3,phone3,email333
> SELECT COUNT(*) FROM customer;
3
```

Sample 2
Input file, step by step screenshot outputs, output file, explanations
Your own sample?

Sample 3
Input file, step by step screenshot outputs, output file, explanations
Your own sample?