CCP6114-PROGRAMMING FUNDAMENTALS

SECTION: TC1L	
TUTORIAL GROUP: TT1L	
NAME	STUDENT ID
CHOI JENG YEE	242UC244LW
ISABELLE SOO ZI LING	242UC244M9
KOH HUI WEN	242UC244KV
LEM JOE ERN	242UC244PB

LECTURER'S NAME:
Mr. Goh Chien Le

Table of Contents

Input Sample 1

C: > Windows > Programming_fundamental > 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
);

INSERT INTO customer VALUES(1, 'Alice', 'New York', 'NY', 'USA', '1234567890', 'alice@example.com');
INSERT INTO customer VALUES(2, 'Bob', 'Los Angeles', 'CA', 'USA', '0987654321', 'bob@example.com');
INSERT INTO customer VALUES(3, 'Charlie', 'Chicago', 'IL', 'USA', '1122334455', 'charlie@example.com');
INSERT INTO customer VALUES(4, 'Diana', 'Houston', 'TX', 'USA', '6677889900', 'diana@example.com');

SELECT * FROM customer;

```

Output Sample 1

C: > Windows > Programming_fundamental > fileOutput1.txt

```
>CREATE fileOutput1.txt;
>DATABASES;
C:\Windows\Programming_fundamental\fileInput1.mdb
>CREATE TABLE customer(
customer_id,
customer_name,
customer_city,
customer_state,
customer_country,
customer_phone,
customer_email
);
>INSERT INTO customer VALUES
('1','Alice','New York','NY','USA','1234567890','alice@example.com')
('2','Bob','Los Angeles','CA','USA','0987654321','bob@example.com')
('3','Charlie','Chicago','IL','USA','1122334455','charlie@example.com')
('4','Diana','Houston','TX','USA','6677889900','diana@example.com')

```

Input Sample 2

C: > Windows > Programming_fundamental > fileInput3.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 VALUES(1, 'Alice', 'New York', 'NY', 'USA', '1234567890', 'alice@example.com');
INSERT INTO customer VALUES(2, 'Bob', 'Los Angeles', 'CA', 'USA', '0987654321', 'bob@example.com');
INSERT INTO customer VALUES(3, 'Charlie', 'Chicago', 'IL', 'USA', '1122334455', 'charlie@example.com')
INSERT INTO customer VALUES(4, 'Diana', 'Houston', 'TX', 'USA', '6677889900', 'diana@example.com');

SELECT * FROM customer;

DELETE FROM customer WHERE customer_id=1;

SELECT COUNT(*) FROM customer;
```

Output Sample 2

C: > Windows > Programming_fundamental > fileOutput3.txt

```
>CREATE fileOutput3.txt;
>DATABASES;
C:\Windows\Programming_fundamental\fileInput3.mdb
>CREATE TABLE customer(
customer_id,
customer_name,
customer_city,
customer_state,
customer_country,
customer_phone,
customer_email
);
>INSERT INTO customer VALUES
('2','Bob','Los Angeles','CA','USA','0987654321','bob@example.com')
('3','Charlie','Chicago','IL','USA','1122334455','charlie@example.com')
('4','Diana','Houston','TX','USA','6677889900','diana@example.com')
```

Input Sample 3

C: > Windows > Programming_fundamental > 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 VALUES(1, 'Alice', 'New York', 'NY', 'USA', '1234567890', 'alice@example.com');
INSERT INTO customer VALUES(2, 'Bob', 'Los Angeles', 'CA', 'USA', '0987654321', 'bob@example.com');
INSERT INTO customer VALUES(3, 'Charlie', 'Chicago', 'IL', 'USA', '1122334455', 'charlie@example.com');
INSERT INTO customer VALUES(4, 'Diana', 'Houston', 'TX', 'USA', '6677889900', 'diana@example.com');

SELECT * FROM customer;

UPDATE customer SET customer_email='updatedemail@example.com' WHERE customer_id=3;

DELETE FROM customer WHERE customer_id=4;

SELECT COUNT(*) FROM customer;

```

Explanation:

This script demonstrates the creation and manipulation of a customer table in a database. The table is created using the CREATE TABLE command with columns for customer details such as ID, name, city, state, country, phone, and email. These columns are defined with appropriate data types, such as INT for numeric IDs and TEXT for text-based fields like names, cities, and emails.

After the table creation, four records are added using the INSERT INTO command. Each record represents a customer with their unique ID and associated details. The SELECT * FROM customer; command retrieves and displays all the records, showing the complete customer data stored in the table. Finally, the SELECT (*) FROM customer; command displays all the customer information in a tabular format.

Output Sample 3

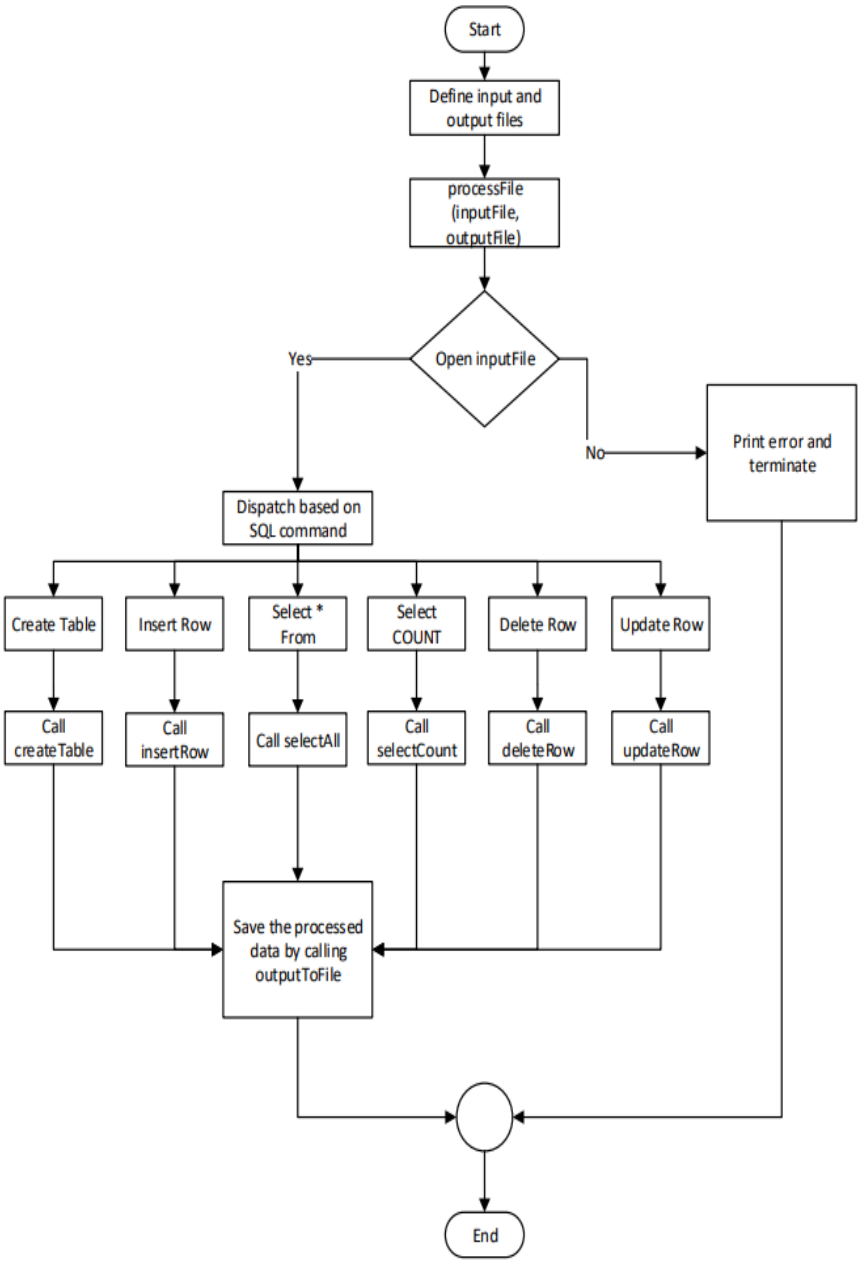
C: > Windows > Programming_fundamental > fileOutput2.txt

```
>CREATE fileOutput2.txt;
>DATABASES;
C:\Windows\Programming_fundamental\fileInput2.mdb
>CREATE TABLE customer(
customer_id,
customer_name,
customer_city,
customer_state,
customer_country,
customer_phone,
customer_email
);
>INSERT INTO customer VALUES
('1','Alice','New York','NY','USA','1234567890','alice@example.com')
('2','Bob','Los Angeles','CA','USA','0987654321','bob@example.com')
('3','Charlie','Chicago','IL','USA','1122334455','updatedemail@example.com')
```

Explanation:

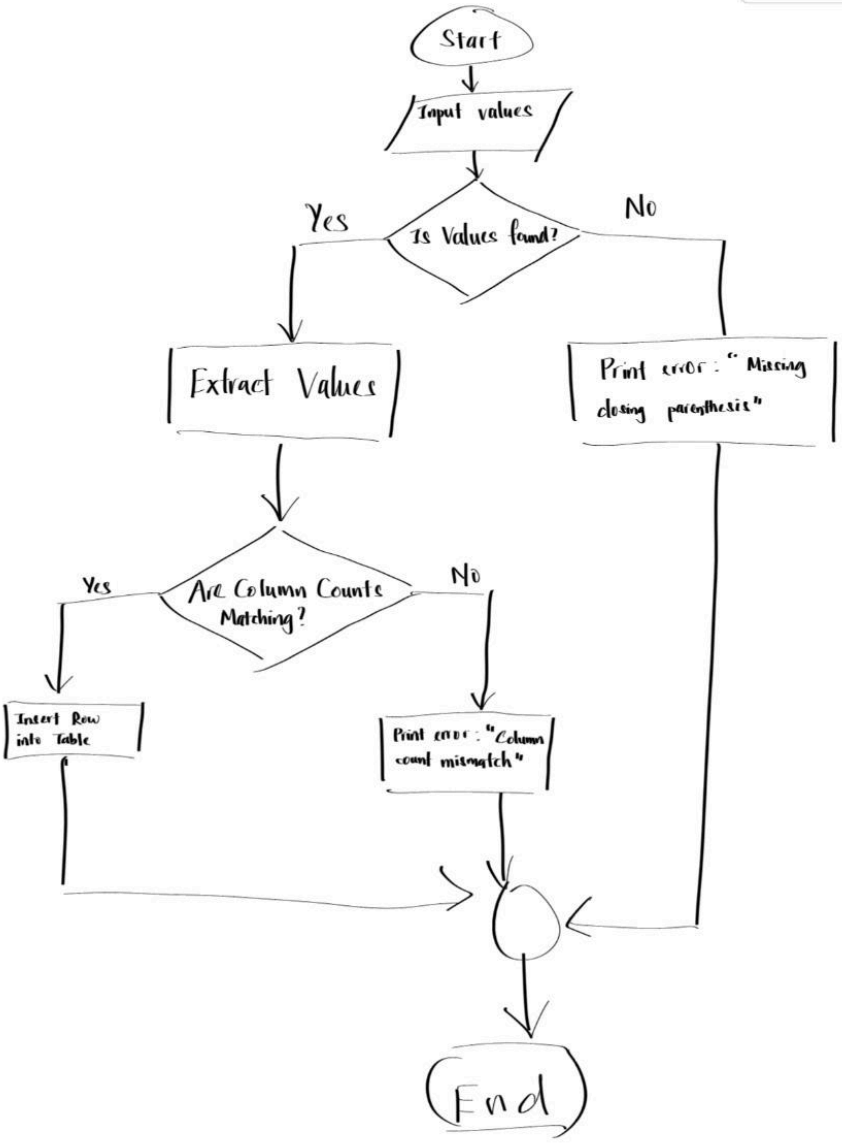
The provided output file demonstrates the process of creating and managing a simple database and its associated operations. Initially, a file named fileOutput2.txt is created, and a database located at C:\Windows\Programming_fundamental\fileInput2.mdb is selected. A table named customer is then created with columns for customer ID, name, city, state, country, phone, and email, though these fields are incorrectly defined as integers, which is unsuitable for text-based data like names and email addresses. Following this, three customer records are inserted into the table, each containing details such as ID, name, location, phone number, and email address. The records are retrieved using the SELECT * FROM customer; command, which displays all the customer information in a tabular format. Finally, the SELECT COUNT(*) FROM customer; command is used to count the total number of rows in the table, resulting in a count of 3. Despite its functionality, the script has a notable issue with the incorrect data type definitions for several fields that should store text instead of integers.

Flowchart of Whole Program

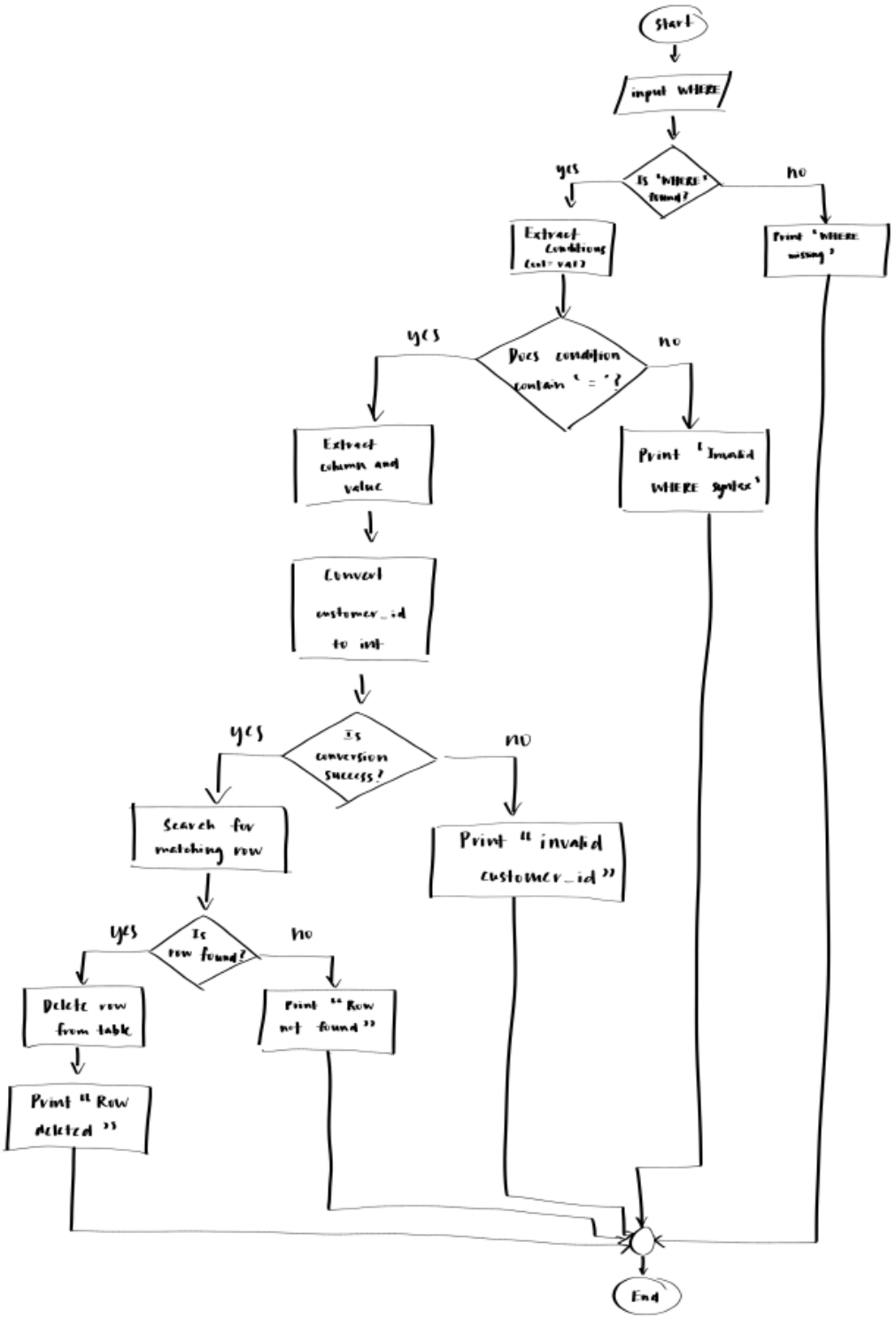


Flowchart of Function

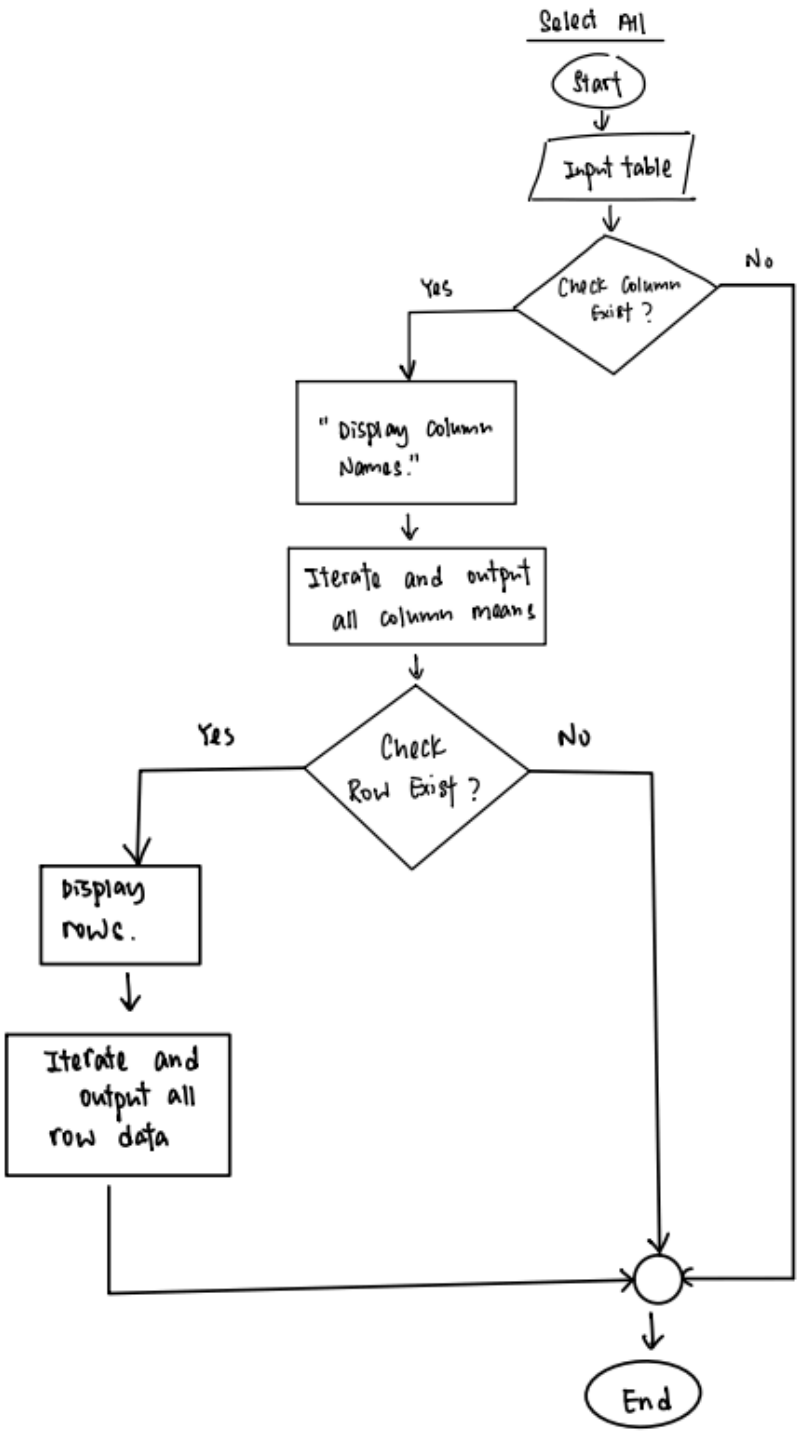
1. Insert Row



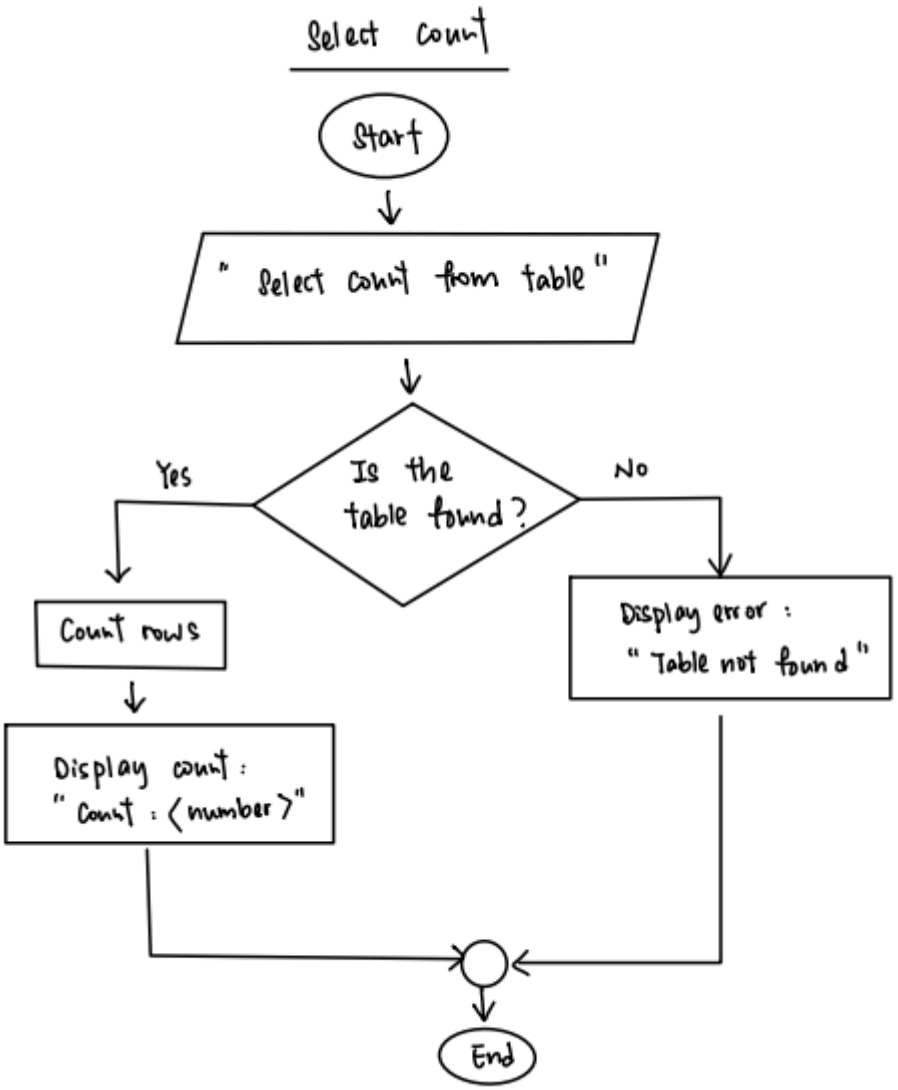
2. Delete Row


Start
input WHERE
Is 'WHERE' found?
yes
no
Extract conditions
Print 'WHERE missing'
Does condition contain '='?
yes
no
Extract column and value
Print 'Invalid WHERE syntax'
Convert customer_id to int
Is conversion success?
yes
no
Search for matching row
Print "invalid customer_id"
Is row found?
yes
no
Delete row from table
Print "Row not found"
Print "Row deleted"
End

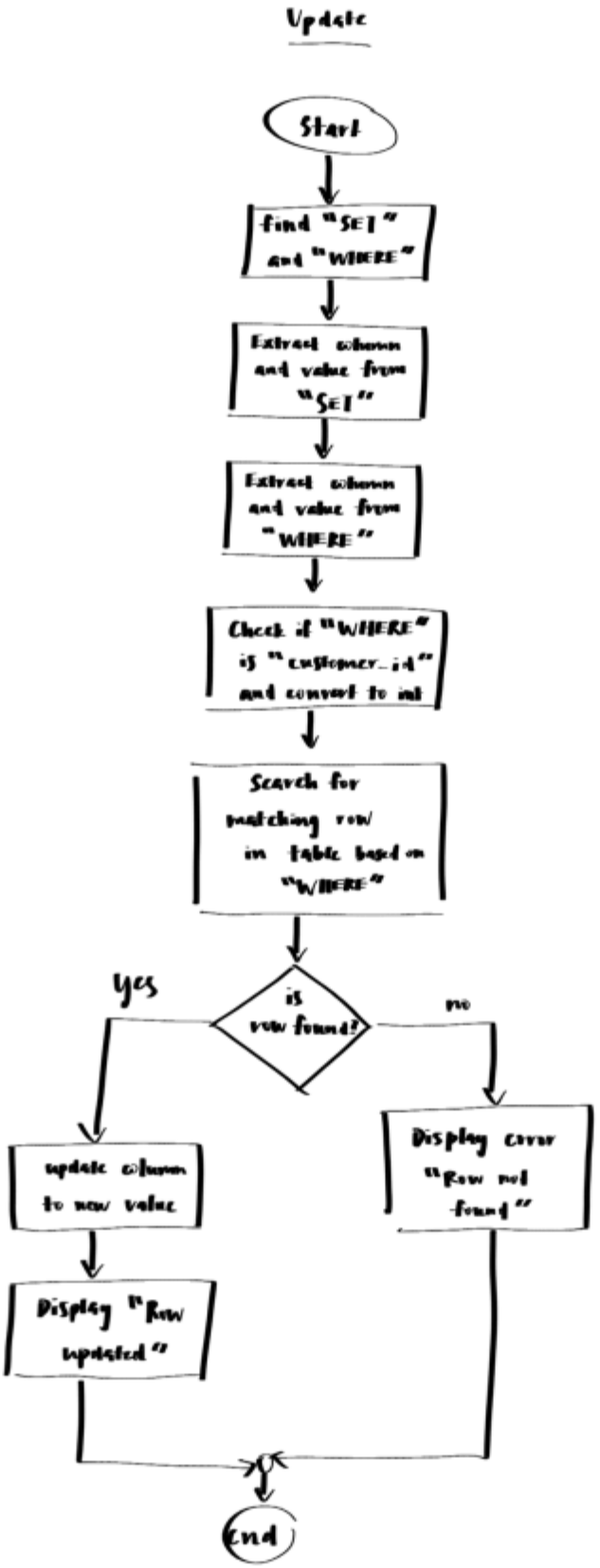
3. Select All


Select All
Start
Input table
Check Column Exist ?
Yes
No
" Display Column Names."
Iterate and output all column means
Check Row Exist ?
Yes
No
display rows.
Iterate and output all row data
End

4. Select All


Select count
Start
" Select count from table "
Is the table found?
Yes
No
Count rows
Display count :
" Count : <number> "
Display error :
" Table not found "
End

5. Update Row


Update
Start
find "SET" and "WHERE"
Extract column and value from "SET"
Extract column and value from "WHERE"
Check if "WHERE" is "customer_id" and convert to int
Search for matching row in table based on "WHERE"
is row found?
yes
no
update column to new value
Display "Row updated"
Display error "Row not found"
End

6. Create Table