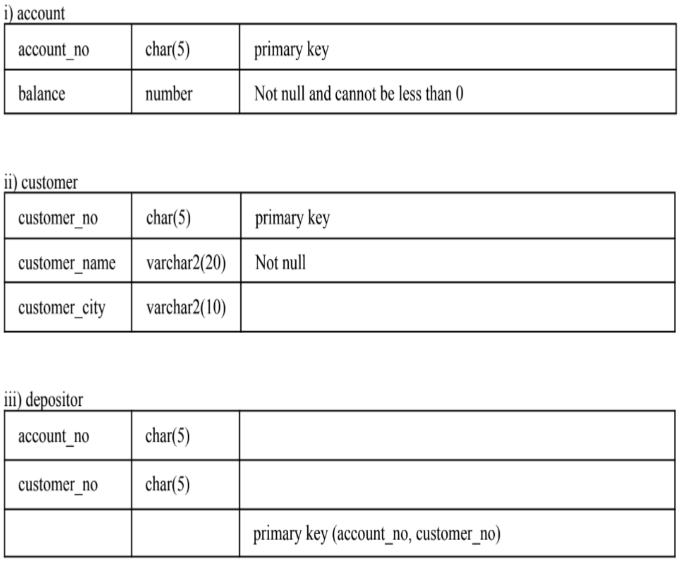
LAB-2
Assignment

Prepared by:

Md. Asif Hossain [2022-3-60-007]
Section: 07

Lab Task # 01 (Schema Definition):

Write SQL statements to create the following tables with the given constraints.



i) account

account_no	char(5)	primary key
balance	number	Not null and cannot be less than 0

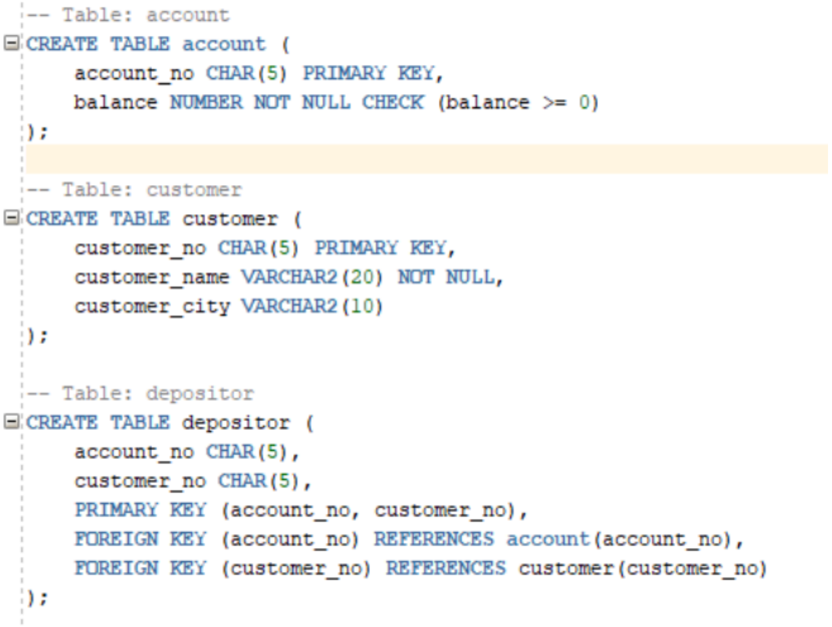
ii) customer

customer_no	char(5)	primary key
customer_name	varchar2(20)	Not null
customer_city	varchar2(10)	

iii) depositor

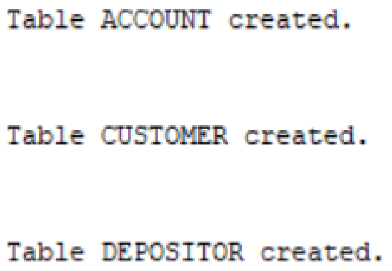
account_no	char(5)	
customer_no	char(5)	
		primary key (account_no, customer_no)

Queries:



```
-- Table: account
CREATE TABLE account (
    account_no CHAR(5) PRIMARY KEY,
    balance NUMBER NOT NULL CHECK (balance >= 0)
);

-- Table: customer
CREATE TABLE customer (
    customer_no CHAR(5) PRIMARY KEY,
    customer_name VARCHAR2(20) NOT NULL,
    customer_city VARCHAR2(10)
);

-- Table: depositor
CREATE TABLE depositor (
    account_no CHAR(5),
    customer_no CHAR(5),
    PRIMARY KEY (account_no, customer_no),
    FOREIGN KEY (account_no) REFERENCES account(account_no),
    FOREIGN KEY (customer_no) REFERENCES customer(customer_no)
);
```

Output:



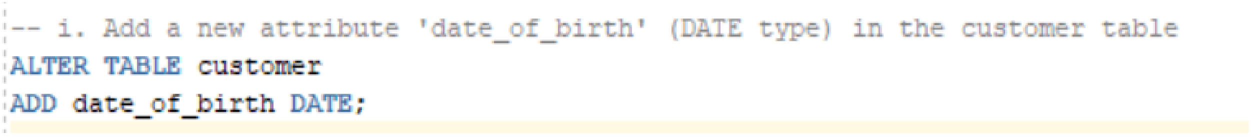
```
Table ACCOUNT created.


Table CUSTOMER created.


Table DEPOSITOR created.
```

Lab Task # 02 (Schema Modification):

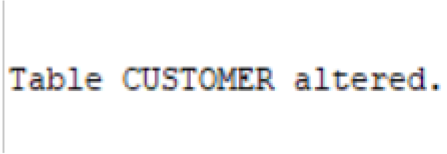
After executing each of these SQL statements execute the command – **desc <table_name>** to confirm the changes.

i. Write SQL statement to add a new attribute 'date_of_birth' (date type) in customer table. ii. Write SQL statement to drop the attribute 'date_of_birth' from customer table. iii. Write SQL statement to rename the attribute account_no, customer_no from depositor table to a_no and c_no, respectively.
iv. Write SQL statements to add two foreign key constraints 'depositor_fk1' and 'depositor_fk2' which identifies a_no and c_no as a foreign key.

i. Write SQL statement to add a new attribute 'date_of_birth' (date type) in customer table.

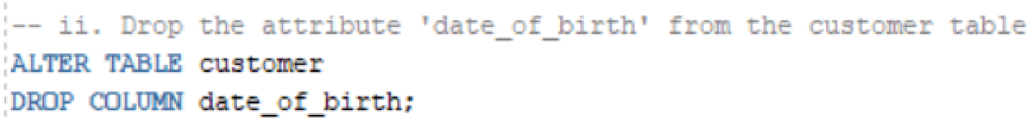


```
-- i. Add a new attribute 'date_of_birth' (DATE type) in the customer table
ALTER TABLE customer
ADD date_of_birth DATE;
```

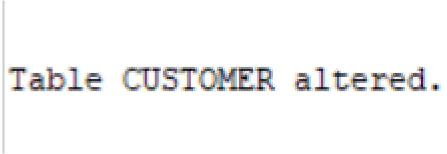


```
Table CUSTOMER altered.
```

ii. Write SQL statement to drop the attribute 'date_of_birth' from customer table.

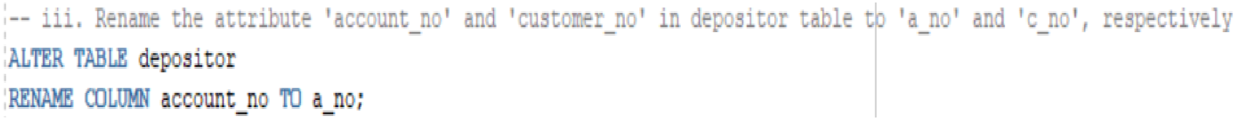


```
-- ii. Drop the attribute 'date_of_birth' from the customer table
ALTER TABLE customer
DROP COLUMN date_of_birth;
```



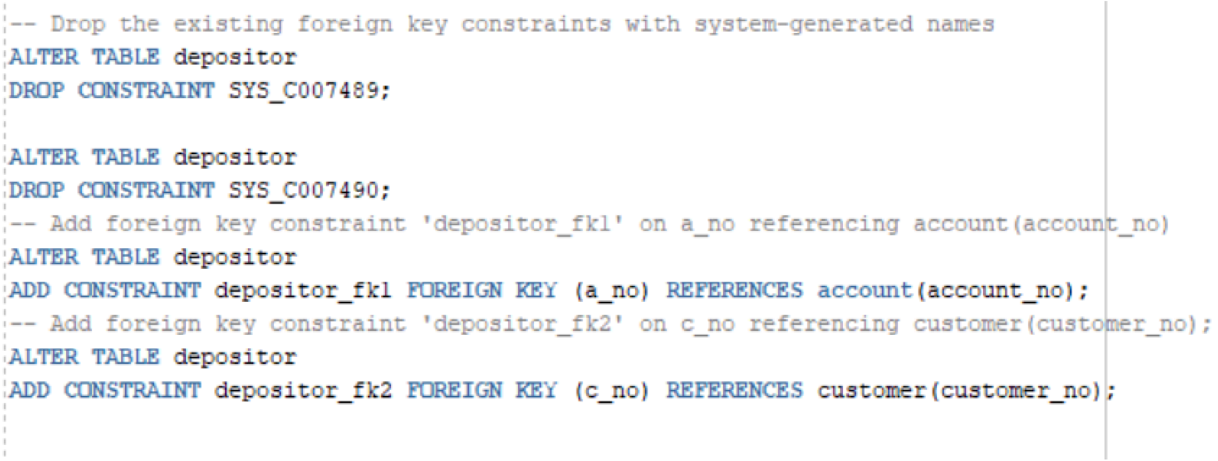
```
Table CUSTOMER altered.
```

iii. Write SQL statement to rename the attribute account_no, customer_no from depositor table to a_no and c_no, respectively.

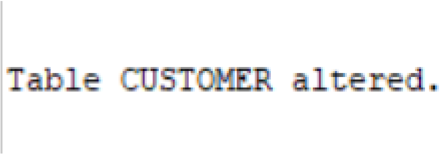


```
-- iii. Rename the attribute 'account_no' and 'customer_no' in depositor table to 'a_no' and 'c_no', respectively
ALTER TABLE depositor
RENAME COLUMN account_no TO a_no;
```

iv. Write SQL statements to add two foreign key constraints ‘depositor_fk1’ and ‘depositor_fk2’ which identifies a_no and c_no as a foreign key.



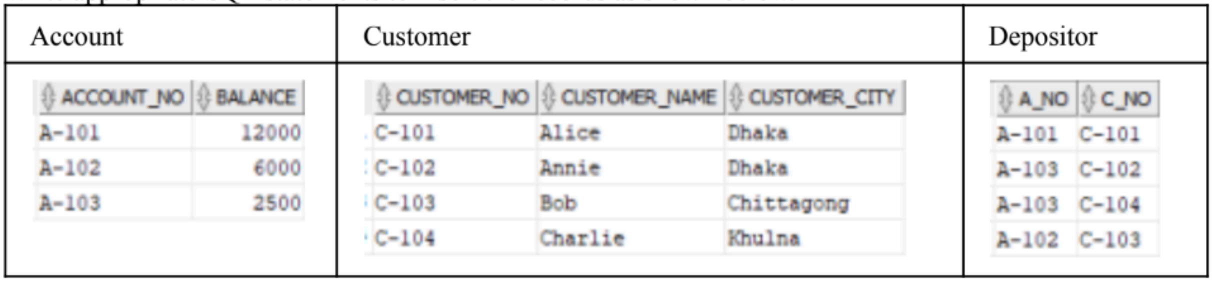
```
-- Drop the existing foreign key constraints with system-generated names
ALTER TABLE depositor
DROP CONSTRAINT SYS_C007489;

ALTER TABLE depositor
DROP CONSTRAINT SYS_C007490;
-- Add foreign key constraint 'depositor_fk1' on a_no referencing account(account_no)
ALTER TABLE depositor
ADD CONSTRAINT depositor_fk1 FOREIGN KEY (a_no) REFERENCES account(account_no);
-- Add foreign key constraint 'depositor_fk2' on c_no referencing customer(customer_no);
ALTER TABLE depositor
ADD CONSTRAINT depositor_fk2 FOREIGN KEY (c_no) REFERENCES customer(customer_no);
```



```
Table CUSTOMER altered.
```

Lab Task # 03 (Inserting Records into Tables):

Write appropriate SQL statements to insert the records as shown below.



Account

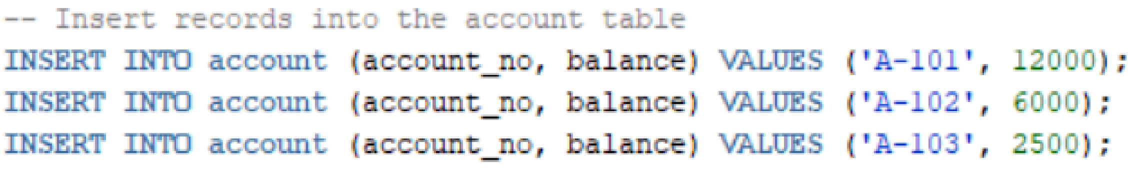
ACCOUNT_NO	BALANCE
A-101	12000
A-102	6000
A-103	2500

Customer

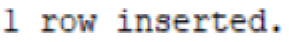
CUSTOMER_NO	CUSTOMER_NAME	CUSTOMER_CITY
C-101	Alice	Dhaka
C-102	Annie	Dhaka
C-103	Bob	Chittagong
C-104	Charlie	Khulna

Depositor

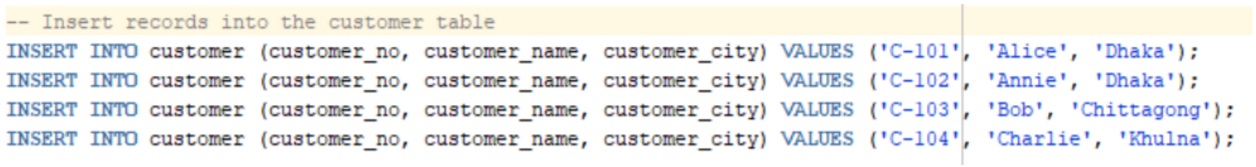
A_NO	C_NO
A-101	C-101
A-103	C-102
A-103	C-104
A-102	C-103



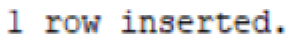
```
-- Insert records into the account table
INSERT INTO account (account_no, balance) VALUES ('A-101', 12000);
INSERT INTO account (account_no, balance) VALUES ('A-102', 6000);
INSERT INTO account (account_no, balance) VALUES ('A-103', 2500);
```



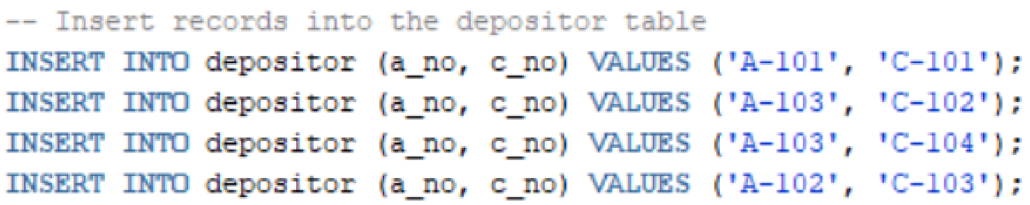
```
1 row inserted.
```



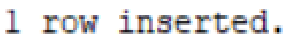
```sql
-- Insert records into the customer table
INSERT INTO customer (customer_no, customer_name, customer_city) VALUES ('C-101', 'Alice', 'Dhaka');
INSERT INTO customer (customer_no, customer_name, customer_city) VALUES ('C-102', 'Annie', 'Dhaka');
INSERT INTO customer (customer_no, customer_name, customer_city) VALUES ('C-103', 'Bob', 'Chittagong');
INSERT INTO customer (customer_no, customer_name, customer_city) VALUES ('C-104', 'Charlie', 'Khulna');
```



```
1 row inserted.
```



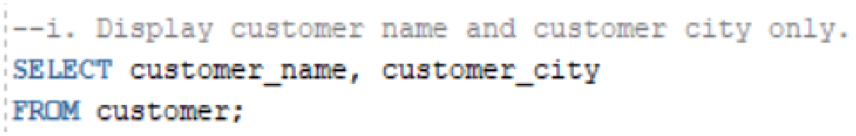
```sql
-- Insert records into the depositor table
INSERT INTO depositor (a_no, c_no) VALUES ('A-101', 'C-101');
INSERT INTO depositor (a_no, c_no) VALUES ('A-103', 'C-102');
INSERT INTO depositor (a_no, c_no) VALUES ('A-103', 'C-104');
INSERT INTO depositor (a_no, c_no) VALUES ('A-102', 'C-103');
```



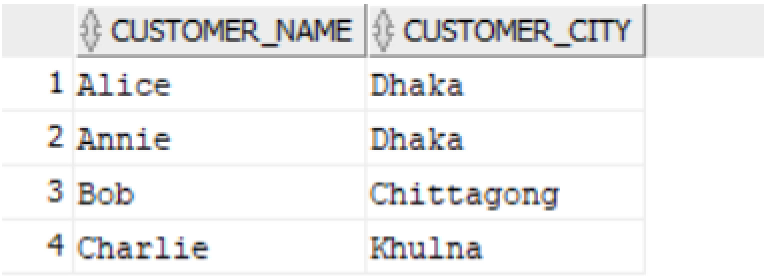
```
1 row inserted.
```

Lab Task # 04 (Queries):

i. Display customer name and customer city only.

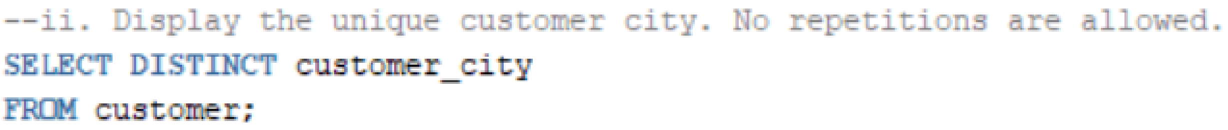


```sql
--i. Display customer name and customer city only.
SELECT customer_name, customer_city
FROM customer;
```

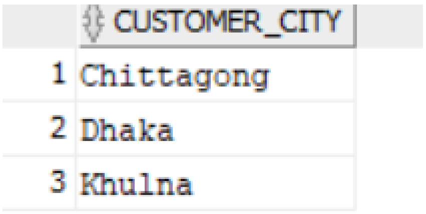


	CUSTOMER_NAME	CUSTOMER_CITY
1	Alice	Dhaka
2	Annie	Dhaka
3	Bob	Chittagong
4	Charlie	Khulna

ii. Display the unique customer city. No repetitions are allowed.

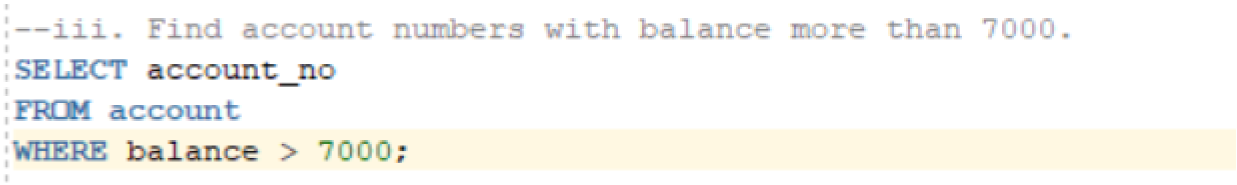


```sql
--ii. Display the unique customer city. No repetitions are allowed.
SELECT DISTINCT customer_city
FROM customer;
```

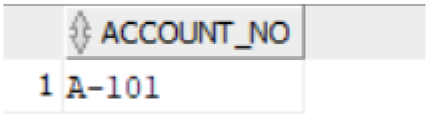


	CUSTOMER_CITY
1	Chittagong
2	Dhaka
3	Khulna

iii. Find account numbers with balance more than 7000.

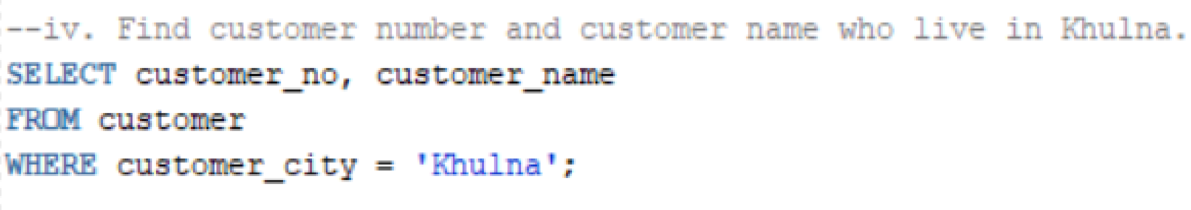


```
--iii. Find account numbers with balance more than 7000.
SELECT account_no
FROM account
WHERE balance > 7000;
```

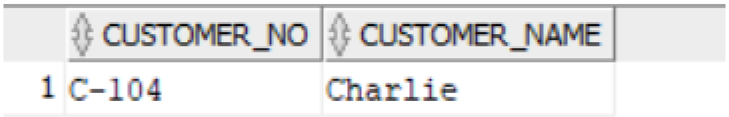


	ACCOUNT_NO
1	A-101

iv. Find customer number and customer name who live in Khulna.

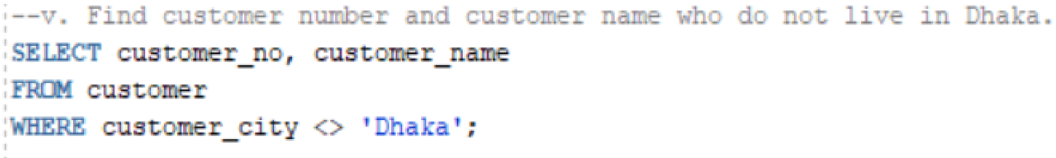


```
--iv. Find customer number and customer name who live in Khulna.
SELECT customer_no, customer_name
FROM customer
WHERE customer_city = 'Khulna';
```

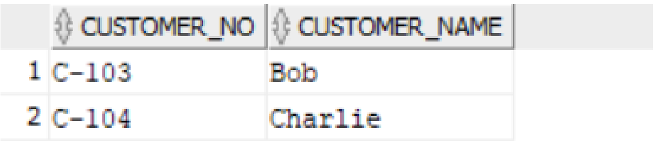


	CUSTOMER_NO	CUSTOMER_NAME
1	C-104	Charlie

v. Find customer number and customer name who do not live in Dhaka.

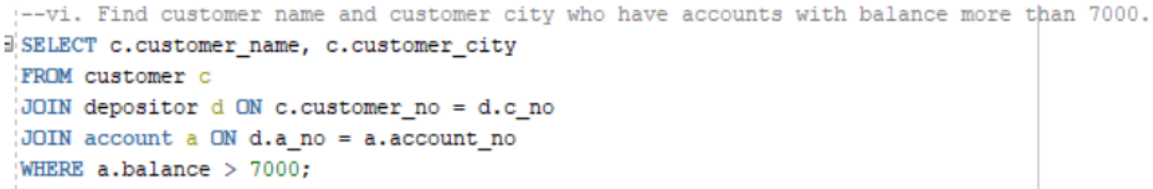


```
--v. Find customer number and customer name who do not live in Dhaka.
SELECT customer_no, customer_name
FROM customer
WHERE customer_city <> 'Dhaka';
```

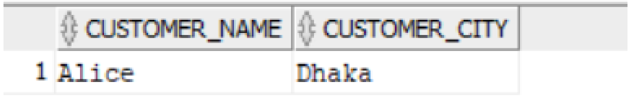


	CUSTOMER_NO	CUSTOMER_NAME
1	C-103	Bob
2	C-104	Charlie

vi. Find customer name and customer city who have accounts with balance more than 7000.

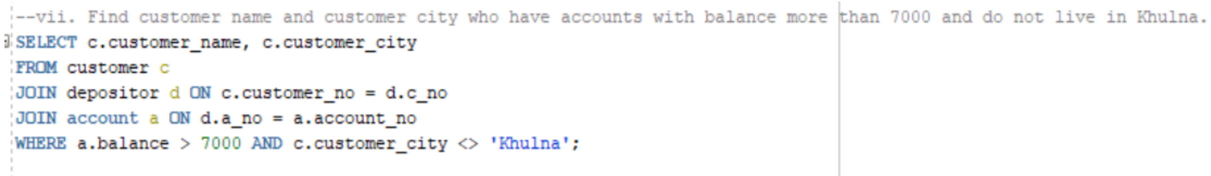


```
--vi. Find customer name and customer city who have accounts with balance more than 7000.
SELECT c.customer_name, c.customer_city
FROM customer c
JOIN depositor d ON c.customer_no = d.c_no
JOIN account a ON d.a_no = a.account_no
WHERE a.balance > 7000;
```

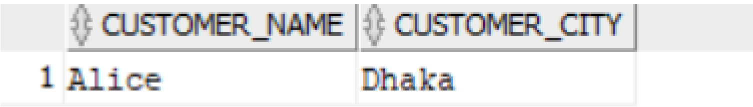


	CUSTOMER_NAME	CUSTOMER_CITY
1	Alice	Dhaka

Vii.Find customer name and customer city who have accounts with balance more than 7000 and do not live in Khulna.

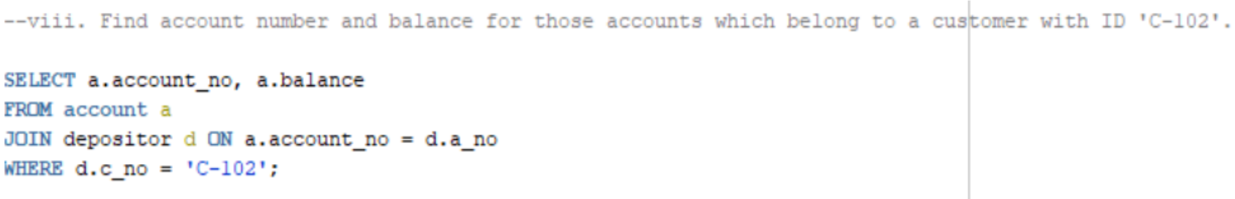


```
--vii. Find customer name and customer city who have accounts with balance more than 7000 and do not live in Khulna.
SELECT c.customer_name, c.customer_city
FROM customer c
JOIN depositor d ON c.customer_no = d.c_no
JOIN account a ON d.a_no = a.account_no
WHERE a.balance > 7000 AND c.customer_city <> 'Khulna';
```

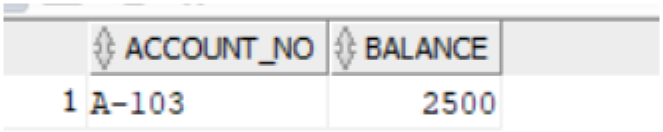


	CUSTOMER_NAME	CUSTOMER_CITY
1	Alice	Dhaka

viii. Find account number and balance for those accounts which belong to a customer with id ‘C-102’.

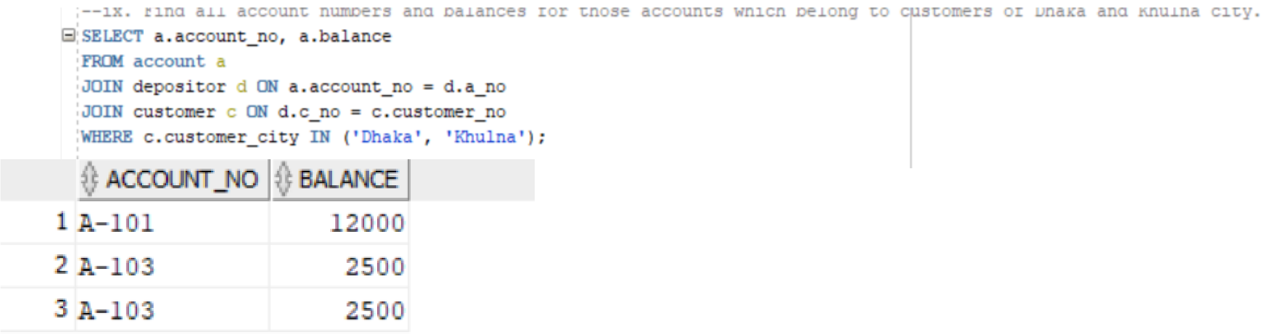


```
--viii. Find account number and balance for those accounts which belong to a customer with ID 'C-102'.

SELECT a.account_no, a.balance
FROM account a
JOIN depositor d ON a.account_no = d.a_no
WHERE d.c_no = 'C-102';
```



	ACCOUNT_NO	BALANCE
1	A-103	2500

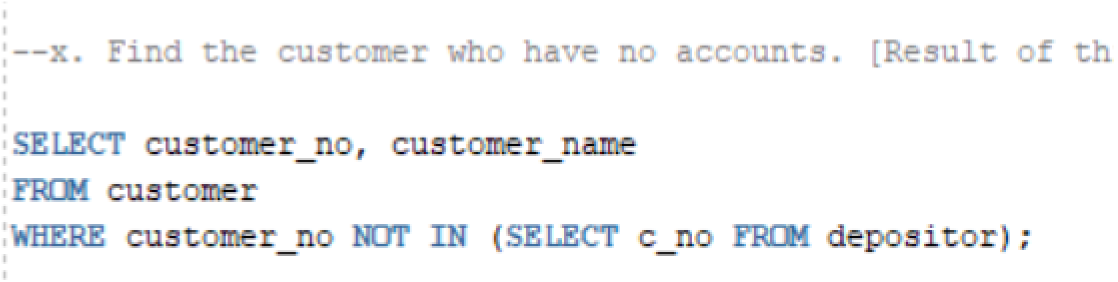
ix. Find all account number and balance for those accounts which belong to customers of Dhaka and Khulna city.



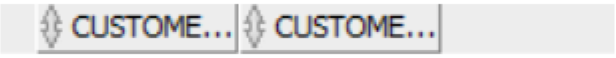
```
--ix. Find all account numbers and balances for those accounts which belong to customers of Dhaka and Khulna city.
SELECT a.account_no, a.balance
FROM account a
JOIN depositor d ON a.account_no = d.a_no
JOIN customer c ON d.c_no = c.customer_no
WHERE c.customer_city IN ('Dhaka', 'Khulna');
```

	ACCOUNT_NO	BALANCE
1	A-101	12000
2	A-103	2500
3	A-103	2500

x. Find the customer who have no accounts. [Result of this query will be empty for this dataset. However, you must write the correct SQL]



```
--x. Find the customer who have no accounts. [Result of th

SELECT customer_no, customer_name
FROM customer
WHERE customer_no NOT IN (SELECT c_no FROM depositor);
```



CUSTOME...	CUSTOME...