AMERICAN INTERNATIONAL
UNIVERSITY-BANGLADESH

Choose an item.

Assignment Cover Page

Assignment Title:	ADMS lab task		
Assignment No:	ss	Date of Submission:	23 February 2023
Course Title:	ADVABNCED DATABASE MANAGEMENT SYSTEM		
Course Code:	Click here to enter text.	Section:	D
Semester:	Spring 2022-2023	Course Teacher:	REZWAN AHMED

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	MD Mera zHossain	20-42460-1	BSc [CSE]	
2			Choose an item.	
3			Choose an item.	
4			Choose an item.	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

Faculty use only

FACULTYCOMMENTS	Marks Obtained	
	Total Marks	

ADVABNCED DATABASE MANAGEMENT SYSTEM

NAME: MD Mearz Hossain

ID: 20-42460-1

Section: D

```
customer -> Customer_name,Customer_street, Customer_city
branch -> Branch_name, Branch_city, Assets
account -> Account_number, Bruch_name, Balance
depositor -> Customer_name, Account_number
 loan -> Loan_number, Branch_name, Ammount
borrower -> Customer_name, Loan_number
```

1. Find the names & cities of all borrowers

```
        SELECT DISTINCT customer.Customer_name, customer.Customer_city
        FROM borrower
        JOIN customer ON borrower.Customer_name = customer.Customer_name
```

2. Find the names & cities of customers who have a loan at "Perryridge" branch

```
        SELECT DISTINCT customer.Customer_name, customer.Customer_city
FROM borrower
JOIN customer ON borrower.Customer_name = customer.Customer_name
JOIN loan ON borrower.Loan_number = loan.Loan_number
JOIN branch ON loan.Branch_name = branch.Branch_name
WHERE branch.Branch_name = 'Perryridge'
```

3. Find the customers with accounts at a branch where "Hayes" has an account

```
select distinct D.customer_name
from depositor D, account A
where D.account_number = A.account_number and
  branch_name in
    (select branch_name
    from depositor Dh, account Ah
    where Dh.account_number = Ah.account_number and D.customer_name = 'Hayes');
```

4. Find the names of branch whose assets are greater than the asstes of some branch in "Brooklyn"

```
select distinct T.branch_name
from branch T, branch S
where T.assets > S.assets and S.branch_city = 'Brooklyn';
```

5. Show the name of branches having at least one account,with average balances of account at each branch ,if that average is above 700

```
select branch_name, avg(balance)
```

```
from account
group by branch_name
having avg(balance) > 700;
```

6. Show the Names of the branch having the largest average balance

```
select branch_name
from account
group by branch_name
having avg(balance) >= all(select avg(balance)
from account group by branch_name);
```

7. Find the name and branch name of the borrower

```
SELECT customer.Customer_name, loan.Branch_name
FROM customer
JOIN borrower ON customer.Customer_name = borrower.Customer_name
JOIN loan ON borrower.Loan_number = loan.Loan_number;
```

8. Find the customers who have a loan in downtown branch

```
SELECT DISTINCT Customer_name
FROM borrower JOIN loan ON borrower.Loan_number = loan.Loan_number
JOIN branch ON loan.Branch_name = branch.Branch_name
WHERE branch.Branch_city = 'downtown';
```

9. Find the customer with loan number at a branch where jonshon has a loan

```
SELECT customer.Customer_name
FROM customer JOIN depositor ON customer.Customer_name = depositor.Customer_name
JOIN account ON depositor.Account_number = account.Account_number
JOIN loan ON account.Branch_name = loan.Branch_name
JOIN borrower ON loan.Loan_number = borrower.Loan_number
WHERE borrower.Customer_name = 'jonshon' AND customer.Customer_name =
borrower.Customer_name;
```

10.Find the number of borrower in each branch

```
SELECT account.Branch_name, COUNT(DISTINCT borrower.Customer_name) AS
Number_of_borrowers
FROM account JOIN depositor ON account.Account_number = depositor.Account_number
JOIN borrower ON depositor.Customer_name = borrower.Customer_name
GROUP BY account.Branch_name;
```

11. Find the customers name who borrows the maximum amount

```
SELECT customer.Customer_name
FROM customer
JOIN borrower ON customer.Customer_name = borrower.Customer_name
JOIN loan ON borrower.Loan_number = loan.Loan_number
WHERE loan.Amount = (
  SELECT MAX(Amount)
  FROM loan
);
```

12. Find the customers name, their city and loan amount that they borrow

```
SELECT customer.Customer_name, customer.Customer_city, loan.Amount
FROM customer
JOIN borrower ON customer.Customer_name = borrower.Customer_name
JOIN loan ON borrower.Loan_number = loan.Loan_number;
```

13. Find the average balance of all customers in "Harrison" having at least two account.

```
SELECT AVG(account.Balance)
FROM account
JOIN depositor ON account.Account_number = depositor.Account_number
JOIN customer ON depositor.Customer_name = customer.Customer_name
WHERE customer.Customer_city = 'Harrison'
GROUP BY customer.Customer_name
HAVING COUNT(account.Account_number) >= 2;
```