

|  |  |
| --- | --- |
| **Course Number and Name:**  CSE 4308  Database Management Systems Lab | |
| **Student Name:**  Nafisa Maliyat | **Student ID:**  200042133 |
| **Report Submission Date:**  08 September, 2022 | **Name of Lab Instructor:**  Md. Bakhtiar Hasan, Lecturer, CSE  Zannatun Naim Sristy, Lecturer, CSE |

**Overview:**

|  |
| --- |
| The lab required applying appropriate conditions using different queries to find the information specified in the question. This was done using combination of various clauses and nested queries with help from the lab manual and a few internet resources.  On the next pages, I have mentioned the following :   * the problem statement, * the problem analysis, * the code snippets of the queries, * problems faced (if any) during solution of the tasks, and * results obtained from the queries. |

**Task 1**

**Problem Statement:**

|  |
| --- |
|  |

**Analysis of the problem:**

|  |
| --- |
| Three tables were created with the mentioned attributes and constraints. |

**SQL Query:**

|  |
| --- |
| DROP TABLE DEPOSITOR\_INFO;  DROP TABLE ACCOUNT;  DROP TABLE CUSTOMER;  CREATE TABLE ACCOUNT  (      ACCOUNT\_NO CHAR(5),      BALANCE NUMBER NOT NULL,      CONSTRAINT PK\_ACCOUNT PRIMARY KEY(ACCOUNT\_NO)  );  CREATE TABLE CUSTOMER  (      CUSTOMER\_NO CHAR(5),      CUSTOMER\_NAME VARCHAR2(20) NOT NULL,      CUSTOMER\_CITY VARCHAR2(10),      CONSTRAINT PK\_CUSTOMER PRIMARY KEY(CUSTOMER\_NO)  );  CREATE TABLE DEPOSITOR  (      ACCOUNT\_NO CHAR(5),      CUSTOMER\_NO CHAR(5),      CONSTRAINT PK\_DEPOSITOR PRIMARY KEY(ACCOUNT\_NO, CUSTOMER\_NO)  ); |

**Any problems faced and how it was solved:**

|  |
| --- |
| When the query was run multiple times, the existing tables created problems. This was solved using Drop Table command at the beginning.  Since the DEPOSITOR table is renamed later on, using the Drop Table for ‘DEPOSITOR’ caused an error since it does not exist. Thus Drop Table was used for ‘DEPOSITOR\_INFO’ instead., which solved the issue.  DEPOSTIOR\_INFO contains references to ACCOUNT and CUSTOMER tables thus dropping these tables first gave an error. This was resolved by dropping DEPOSITOR\_INFO first and then the other two tables. |

**Results:**

|  |
| --- |
|  |

**Task 2**

**Problem Statement:**

|  |
| --- |
|  |

**Analysis of the problem:**

|  |
| --- |
| The queries for these were written using Alter Table queries, as provided in the lab manual.  (c) required two alter table queries for renaming two attributes, as well as (e).  Task 2 thus required seven query statements. |

**SQL Query:**

|  |
| --- |
| --2A--  ALTER TABLE CUSTOMER ADD DATE\_OF\_BIRTH DATE;  --2B--  ALTER TABLE ACCOUNT MODIFY BALANCE NUMBER(12,2);  --2C--  ALTER TABLE DEPOSITOR RENAME COLUMN ACCOUNT\_NO TO A\_NO;  ALTER TABLE DEPOSITOR RENAME COLUMN CUSTOMER\_NO TO C\_NO;  --2D--  ALTER TABLE DEPOSITOR RENAME TO DEPOSITOR\_INFO;  --2E--  ALTER TABLE DEPOSITOR\_INFO  ADD CONSTRAINT FK\_DEPOSITOR\_ACCOUNT  FOREIGN KEY (A\_NO) REFERENCES  ACCOUNT(ACCOUNT\_NO);  ALTER TABLE DEPOSITOR\_INFO  ADD CONSTRAINT FK\_DEPOSITOR\_CUSTOMER  FOREIGN KEY (C\_NO) REFERENCES  CUSTOMER(CUSTOMER\_NO); |

**Any problems faced and how it was solved:**

|  |
| --- |
| Since the lab manual provided the necessary instructions, there were no issues encountered. |

**Results:**

|  |
| --- |
|  |

**Task 3**

**Problem Statement:**

|  |
| --- |
|  |

**Analysis of the problem:**

|  |
| --- |
| From (a) to (f), simple queries in the format of select-from-where were performed on single table and appropriate conditions were applied.  (g) required performing query on both DEPOSITOR\_INFO and CUSTOMER since DEPOSITOR\_INFO stores the customer number of those customers associated with an account number. CUSTOMER table was required to select customer names and their city.  In (h), DEPOSITOR\_INFO and ACCOUNT tables were used to first find the list of customer numbers where the corresponding accounts has a balance greater than 1000. Another query was performed on the obtained result to retrieve the customer information of those selected customers from CUSTOMER table.  ‘Distinct’ keyword was used to ensure even if a customer has more than one account that fulfilled the conditions, their information would be printed just once.  For (i), queries were performed on all three tables. All the conditions were ANDED together; another alternative query could have been nested query similar to (h).  Here, ‘distinct’ was not used because one account cannot be associated with more than one person logically, |

**SQL Query:**

|  |
| --- |
| --3A--  SELECT ACCOUNT\_NO  FROM ACCOUNT  WHERE BALANCE<100000;  --3B--  SELECT CUSTOMER\_NAME  FROM CUSTOMER  WHERE CUSTOMER\_CITY='KHL';  --3C--  SELECT CUSTOMER\_NO  FROM CUSTOMER  WHERE CUSTOMER\_NAME LIKE '%A%';  --3D--  SELECT DISTINCT A\_NO  FROM DEPOSITOR\_INFO;  --3E--  SELECT \*  FROM ACCOUNT, DEPOSITOR\_INFO;  --3F--  ALTER TABLE DEPOSITOR\_INFO RENAME COLUMN C\_NO TO CUSTOMER\_NO;  SELECT \*  FROM CUSTOMER NATURAL JOIN DEPOSITOR\_INFO;  ALTER TABLE DEPOSITOR\_INFO RENAME COLUMN CUSTOMER\_NO TO C\_NO;  --3G--  SELECT CUSTOMER.CUSTOMER\_NAME, CUSTOMER.CUSTOMER\_CITY  FROM DEPOSITOR\_INFO, CUSTOMER  WHERE CUSTOMER.CUSTOMER\_NO = DEPOSITOR\_INFO.C\_NO;  --3H--  SELECT CUSTOMER.CUSTOMER\_NO, CUSTOMER.CUSTOMER\_NAME, CUSTOMER.CUSTOMER\_CITY  FROM CUSTOMER  WHERE CUSTOMER\_NO IN  (      SELECT DISTINCT DEPOSITOR\_INFO.C\_NO      FROM DEPOSITOR\_INFO, ACCOUNT      WHERE DEPOSITOR\_INFO.A\_NO = ACCOUNT.ACCOUNT\_NO          AND ACCOUNT.BALANCE>1000  );  --3I--  SELECT ACCOUNT.ACCOUNT\_NO, ACCOUNT.BALANCE  FROM DEPOSITOR\_INFO, ACCOUNT, CUSTOMER  WHERE DEPOSITOR\_INFO.A\_NO = ACCOUNT.ACCOUNT\_NO      AND DEPOSITOR\_INFO.C\_NO = CUSTOMER\_NO      AND ACCOUNT.BALANCE BETWEEN 5000 AND 10000      AND CUSTOMER.CUSTOMER\_CITY='DHK'; |

**Problems faced and how they were solved:**

|  |
| --- |
| In 3(f), the Natural Join did not occur as expected since DEPOSITOR\_INFO and CUSTOMER tables no longer have a commonly named attribute (CUSTOMER\_NO was renamed to C\_NO in DEPOSITOR\_INFO table). Renaming the attribute in DEPOSITOR\_INFO table temporarily for the Natural Join query solved the problem. |

**Test Entries:**

|  |
| --- |
| INSERT INTO ACCOUNT VALUES('101', 1000);  INSERT INTO ACCOUNT VALUES('102', 6000);  INSERT INTO ACCOUNT VALUES('103', 10000);  INSERT INTO ACCOUNT VALUES('104', 3000);  INSERT INTO CUSTOMER VALUES('201', 'NAFISA', 'DHK', null);  INSERT INTO CUSTOMER VALUES('202', 'ANIKA', 'KHL', null);  INSERT INTO CUSTOMER VALUES('203', 'MALIYAT', 'KHL', null);  INSERT INTO CUSTOMER VALUES('204', 'AJAY', 'DHK', null);  INSERT INTO DEPOSITOR\_INFO VALUES('101', '201');  INSERT INTO DEPOSITOR\_INFO VALUES('102', '202');  INSERT INTO DEPOSITOR\_INFO VALUES('104', '203');  INSERT INTO DEPOSITOR\_INFO VALUES('101', '204'); |

**Results:**

|  |
| --- |
| (a)    (b)    (c)    (d)    (e)    (f)    (g)    (h)    (i) |