* This my Database Login setting information screen and tables created as per pdf instructions.

Graphical user interface, text, application

Description automatically generated

1. How many tables have been created? List the names of the created tables

There are 12 tables created.

* select count(\*) as "Number of tables" from tab;

Text

Description automatically generated

-CONTACTS

-COUNTRIES

-CUSTOMERS

-EMPLOYEES

-INVENTORIES

-LOCATIONS

-ORDER\_ITEMS

-ORDERS

-PRODUCT\_CATEGORIES

-PRODUCTS

-REGIONS

-WAREHOUSES

1. Display or write down each tables column names and data types.

**CONTACTS:**

Text

Description automatically generated with medium confidence

**COUNTRIES:**

**Text

Description automatically generated with medium confidence**

**CUSTOMERS:**

**Text

Description automatically generated**

**EMPLOYEES:**

**Text

Description automatically generated**

**INVENTORIES:**

**Table

Description automatically generated**

**LOCATIONS:**

**Text

Description automatically generated**

**ORDER\_ITEMS:**

**Text

Description automatically generated**

**ORDERS:**

**Text

Description automatically generated**

**PRODUCT\_CATEGORIES:**

**Text

Description automatically generated**

**PRODUCTS:**

**Text

Description automatically generated**

**REGIONS:**

**A picture containing text

Description automatically generated**

**WAREHOUSES:**

**Text

Description automatically generated**

1. Write a SELECT query for each of the table created to display all the values of the tables and sort them descending with their primary keys (Hint: the number of SELECT queries depend on the number of tables created)

* **SELECT \* FROM contacts ORDER BY contact\_id DESC;**

**A screenshot of a computer

Description automatically generated with low confidence**

* **SELECT \* FROM countries ORDER BY country\_id DESC;**

**Table

Description automatically generated**

* **SELECT \* FROM customers ORDER BY customer\_id DESC;**

**A screenshot of a computer

Description automatically generated with low confidence**

* **SELECT \* FROM employees ORDER BY employee\_id DESC;**

**A picture containing table

Description automatically generated**

* **SELECT \* FROM inventories ORDER BY product\_id DESC;**

**Table

Description automatically generated**

* **SELECT \* FROM locations ORDER BY location\_id DESC;**

**A picture containing text, receipt, screenshot

Description automatically generated**

* **SELECT \* FROM order\_items ORDER BY item\_id DESC;**

**Table

Description automatically generated**

* **SELECT \* FROM orders ORDER BY order\_id DESC;**

**Table

Description automatically generated**

* **SELECT \* FROM product\_categories ORDER BY category\_id DESC;**

**Text

Description automatically generated**

* **SELECT \* FROM products ORDER BY product\_id DESC;**

**Text, letter

Description automatically generated**

* **SELECT \* FROM regions ORDER BY region\_id DESC;**

**Text

Description automatically generated**

* **SELECT \* FROM warehouses ORDER BY warehouse\_id DESC;**

**A picture containing text, screenshot, document

Description automatically generated**

1. Write a SQL statement to display DISTINCT credit limit for the customers table.

* **SELECT DISTINCT credit\_limit FROM customers;**

**Table

Description automatically generated**

1. Using Integrity constraints tab for each table. Write down the primary keys for each table and write relations between the tables like which are parent table, child table and what are their primary keys and foreign keys

* CONTACTS table:

Primary Key: **CONTACT\_ID**

Foreign Key: **CUSTOMER\_ID**

* COUNTRIES:

Primary Key: **COUNTRY\_ID**

Foreign Key: **REGION\_ID**

* CUSTOMERS:

Primary Key: **CUSTOMER\_ID**

Foreign Key: **NONE**

* EMPLOYEES:

Primary Key: **EMPLOYEE\_ID**

Foreign Key: **MANAGER\_ID**

* INVENTORIES:

Primary Key: **PRODUCT\_ID**

Foreign Key: **WAREHOUSE\_ID, PRODUCT\_ID**

* LOCATIONS:

Primary Key: **LOCATION\_ID**

Foreign Key: **COUNTRY\_ID**

* ORDER\_ITEMS:

Primary Key: **ORDER\_ID**

Foreign Key: **ITEM\_ID , PRODUCT\_ID**

* ORDERS:

Primary Key: **ORDER\_ID**

Foreign Key: **CUSTOMER\_ID , SALESMAN\_ID**

* PRODUCT\_CATEGORIES:

Primary Key: **CATEGORY\_ID**

Foreign Key: **NONE**

* PRODUCTS:

Primary Key: **PRODUCT\_ID**

Foreign Key: **CATEGORY\_ID**

* REGIONS:

Primary Key: **REGION\_ID**

Foreign Key: **NONE**

* WAREHOUSES:

Primary Key: **WAREHOUSE\_ID**

Foreign Key: **LOCATION\_ID**

1. INSERT 2 rows of values to EMPLOYEES table. Makeup your own values and display the newly entered information

* **INSERT INTO employees VALUES (121,'Prince','Jodhani','pdjodhani@myseneca.ca','8780906333','16-JAN-22',NULL,'Stock Manager');**
* **INSERT INTO employees VALUES (122,'Rohan','kapadiya','rskapadiya@myseneca.ca','7567302437','15-JAN-22',NULL,'Stock Manager');**

****

1. Create an empty new EMPLOYEE table called “TESTEMPLOYEE” from the existing EMPLOYEE table. Display the contents of TESTEMPLOYEE. Now copy all the values from EMPLOYEE table to TESTEMPLOYEE and display the contents.

* **CREATE TABLE TESTEMPLOYEE AS (SELECT \* FROM employees WHERE 1=2);**

**Table

Description automatically generated with medium confidence**

* **INSERT INTO testemployee SELECT \* FROM employees;**

**Text

Description automatically generated with medium confidence**

1. Write a SQL query to update ORDER\_ITEMS quantity to 100 for the order ids in the range of 1 to 10

* **UPDATE order\_items SET quantity = 100 WHERE order\_id BETWEEN 1 AND 10;**

**Table

Description automatically generated**

1. Delete the newly inserted values in TESTEMPLOYEE table

* DELETE FROM testemployee;

Graphical user interface, application

Description automatically generated

1. Using JOIN statements write the SQL query for the following data required
2. Display all ORDER\_ITEMS that are in common with ORDERS

* SELECT \* FROM order\_items INNER JOIN orders USING (order\_id);

Table

Description automatically generated

1. Display all ORDER\_ITEMS that are in common with PRODUCTS

* SELECT \* FROM order\_items INNER JOIN products USING(product\_id);

A picture containing qr code

Description automatically generated

1. Display all the ORDERS that are not in common with ORDER\_ITEMS

* SELECT \* FROM order\_items INNER JOIN orders ON order\_items.order\_id <> orders.order\_id;

Table

Description automatically generated

1. Display all the PRODUCTS that are not in common with ORDER\_ITEMS

* SELECT \* FROM products INNER JOIN order\_items ON order\_items.product\_id <> products.product\_id;

A screenshot of a computer

Description automatically generated with low confidence