--Question 1: Create a Table for Employee Information -----------

CREATE TABLE EmployeeInfo (

employee\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

first\_name VARCHAR2(50) NOT NULL,

last\_name VARCHAR2(50) NOT NULL,

date\_of\_birth DATE NOT NULL,

email VARCHAR2(100),

phone\_number VARCHAR2(20),

hire\_date DATE NOT NULL,

department VARCHAR2(50)

);

-------------- Question 1: Altering the EmployeeInfo Table ---------------

1. Add a new column named address of type VARCHAR2(200) to store employee addresses.

ALTER TABLE EmployeeInfo ADD address varchar2(200);

2.Modify the email column to allow null values.

ALTER TABLE EmployeeInfo MODIFY email VARCHAR2(100);

3. Rename the column phone\_number to contact\_number.

ALTER TABLE EmployeeInfo RENAME COLUMN phone\_number TO contact\_number;

4. Delete the department column from the table.

ALTER TABLE EmployeeInfo DROP COLUMN department;

5. Add a primary key constraint on the employee\_id column.

ALTER TABLE EmployeeInfo ADD CONSTRAINT pk\_employee\_id PRIMARY KEY (employee\_id);

6. Remove the primary key constraint from the employee\_id column.

ALTER TABLE EmployeeInfo DROP CONSTRAINT pk\_employee\_id;

------- ADD Employeeinfo table record ---------

insert into EmployeeInfo(first\_name,last\_name,date\_of\_birth,email,phone\_number,hire\_date,department)

values ('jainish','barbhaya','01-feb-2003','jainish@gmail.com',6352811628,'16-apr-2023','mca');

-------------- Update Queries for EmployeeInfo Table: -----------------

1. You need to update the email address of an employee with employee\_id 1. Set their email to 'newemail@example.com'

UPDATE EmployeeInfo SET email = 'newemail@example.com' WHERE employee\_id = 1;

2. An employee with employee\_id 203 recently changed their last name to 'Johnson.' Update their last name in the EmployeeInfo table accordingly. Write the SQL query to make this change.

UPDATE EmployeeInfo SET last\_name = 'Johnson' WHERE employee\_id = 203;

3. An employee with employee\_id 305 has been promoted and their salary needs to be increased by $5,000.

---note: salary name column in not available employeeinfo table ---------

UPDATE EmployeeInfo SET salary = salary + 5000 WHERE employee\_id = 305;

4. Due to a system error, the hire date of an employee with employee\_id 402 was recorded incorrectly as '2023-06-15' instead of '2023-06-01'. Write an SQL query to correct this hire date.

UPDATE EmployeeInfo SET hire\_date = TO\_DATE('01-jun-2023') WHERE employee\_id = 402;

5. You need to assign a new department to an employee with employee\_id 507. Update their department to 'Marketing'.

UPDATE EmployeeInfo SET department = 'Marketing' WHERE employee\_id = 507;

----------Question 2: Create a Table for Product Inventory -----------

CREATE TABLE ProductInventory (

product\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

product\_name VARCHAR2(100) NOT NULL,

description VARCHAR2(200),

price NUMBER(10, 2) NOT NULL,

quantity\_in\_stock NUMBER(5) NOT NULL,

manufacturer VARCHAR2(50),

category VARCHAR2(50),

date\_added DATE NOT NULL);

----------- Question 2: Altering the ProductInventory Table -------

1. Increase the precision of the price column to allow prices up to $9999.99.

ALTER TABLE ProductInventory MODIFY price NUMBER(11, 2);

2. Add a new column named manufacturer\_location of type VARCHAR2(100) to store the location of the manufacturer.

ALTER TABLE ProductInventory ADD manufacturer\_location VARCHAR2(100);

3. Rename the column category to product\_category.

ALTER TABLE ProductInventory RENAME COLUMN category TO product\_category;

4. Set a default value of 0 for the quantity\_in\_stock column.

ALTER TABLE ProductInventory MODIFY quantity\_in\_stock NUMBER(5) DEFAULT 0;

5. Add a primary key constraint on the product\_id column.

ALTER TABLE ProductInventory ADD CONSTRAINT pk\_product\_id PRIMARY KEY (product\_id);

6. Remove the primary key constraint from the product\_id column.

ALTER TABLE ProductInventory DROP CONSTRAINT pk\_product\_id;

-------- Update Queries for ProductInventory Table: ---------------

1. The price of a product with product\_id 101 has been increased by 10%. Write an SQL query to update the price accordingly.

UPDATE ProductInventory SET price = price \* 1.10 WHERE product\_id = 101;

2. A product with product\_id 205 has been discontinued, and its quantity in stock should be set to 0. Write an SQL query to update the quantity\_in\_stock for this product.

UPDATE ProductInventory SET quantity\_in\_stock = 0 WHERE product\_id = 205;

3. The manufacturer of a product with product\_id 303 has changed their location. Update the manufacturer\_location to 'New York' for this product.

UPDATE ProductInventory SET manufacturer\_location = 'New York' WHERE product\_id = 303;

4. Update the category of products with names containing the word 'Electronics' to 'Electrical Appliances'.

UPDATE ProductInventory SET product\_category = 'Electrical Appliances' WHERE INSTR(product\_name, 'Electronics');

5. A product with product\_id 408 has been recalled and is no longer available. Update its quantity\_in\_stock to -1 to mark it as unavailable.

UPDATE ProductInventory SET quantity\_in\_stock = -1 WHERE product\_id = 408;

-----------Question 3: Create a Table for Library Books ----------

CREATE TABLE LibraryBooks (

book\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

title VARCHAR2(200) NOT NULL,

author VARCHAR2(150) NOT NULL,

publication\_date DATE NOT NULL,

isbn VARCHAR2(20),

genre VARCHAR2(50),

available\_copies NUMBER(5),

total\_copies NUMBER(5)

);

--------- Question 3: Altering the LibraryBooks Table ------------

1. Add a new column named language of type VARCHAR2(50) to store the language of the book.

ALTER TABLE LibraryBooksADD language VARCHAR2(50);

2. Modify the isbn column to allow null values.

ALTER TABLE LibraryBooks MODIFY isbn VARCHAR2(20) NULL;

3. Rename the column available\_copies to available\_quantity.

ALTER TABLE LibraryBooks RENAME COLUMN available\_copies TO available\_quantity;

4. Delete the total\_copies column from the table.

ALTER TABLE LibraryBooks DROP COLUMN total\_copies;

5. Add a primary key constraint on the book\_id column.

ALTER TABLE LibraryBooks ADD CONSTRAINT pk\_book\_id PRIMARY KEY (book\_id);

6. Remove the primary key constraint from the book\_id column.

ALTER TABLE LibraryBooks DROP CONSTRAINT pk\_book\_id;

--------------- Update Queries for LibraryBooks Table: --------------------------------

1. A book with book\_id 101 has received a new edition, and its title needs to be updated to 'The New Book Title'. Write an SQL query to update the books title.

UPDATE LibraryBooks SET title = 'The New Book Title' WHERE book\_id = 101;

2.Correct the publication date of a book with book\_id 203, which was mistakenly recorded as '2021-05-15' instead of '2021-05-01'.

UPDATE LibraryBooks SET publication\_date = TO\_DATE('01-may-2021', 'YYYY-MM-DD') WHERE book\_id = 203;

3. Update the genre of all books published before the year 2000 to 'Classics'.

UPDATE LibraryBooks SET genre = 'Classics' WHERE EXTRACT(YEAR FROM publication\_date) < 2000;

4. The total copies of a book with book\_id 305 should be increased by 5 due to high demand. Write an SQL query to update the total\_copies accordingly.

UPDATE LibraryBooks SET total\_copies = total\_copies + 5 WHERE book\_id = 305;

5. A book with book\_id 402 has been removed from the librarys collection. Set its available\_quantity to 0 to mark it as unavailable.

UPDATE LibraryBooks SET available\_quantity = 0 WHERE book\_id = 402;

------------ Question 4: Create a Table for Customer Orders ----------

CREATE TABLE CustomerOrders (

order\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

customer\_name VARCHAR2(100) NOT NULL,

order\_date DATE NOT NULL,

total\_amount NUMBER(10, 2) NOT NULL,

shipping\_address VARCHAR2(200),

payment\_method VARCHAR2(50),

status VARCHAR2(20),

tracking\_number VARCHAR2(30)

);

---------- Question 4: Altering the CustomerOrders Table -----------

1. Add a new column named delivery\_date of type DATE to store the expected delivery date of orders.

ALTER TABLE CustomerOrders ADD delivery\_date DATE;

2. Modify the shipping\_address column to allow longer addresses (e.g., VARCHAR2(250)).

ALTER TABLE CustomerOrders MODIFY shipping\_address VARCHAR2(250);

3. Rename the column payment\_method to payment\_type.

ALTER TABLE CustomerOrders RENAME COLUMN payment\_method TO payment\_type;

4. Set a default value of 'Processing' for the status column.

ALTER TABLE CustomerOrders MODIFY status DEFAULT 'Processing';

5. Add a primary key constraint on the order\_id column.

ALTER TABLE CustomerOrders ADD CONSTRAINT pk\_order\_id PRIMARY KEY (order\_id);

6. Remove the primary key constraint from the order\_id column

ALTER TABLE CustomerOrders DROP CONSTRAINT pk\_order\_id;

----------- Update Queries for CustomerOrders Table: ------------------

1. An order with order\_id 101 has a new shipping address due to a customers recent move. Update the shipping\_address for this order.

UPDATE CustomerOrders SET shipping\_address = 'New Address Here' WHERE order\_id = 101;

2. Change the payment type of an order with order\_id 203 from 'Credit Card' to 'PayPal'.

UPDATE CustomerOrders SET payment\_method = 'PayPal' WHERE order\_id = 203;

3. An order with order\_id 303 is delayed and will be delivered two days later. Update its delivery\_date accordingly.

UPDATE CustomerOrders SET order\_date = order\_date + 2 WHERE order\_id = 303;

4. Update the total amount of an order with order\_id 405 to $175.50, reflecting a change in the orders contents.

UPDATE CustomerOrders SET total\_amount = 175.50 WHERE order\_id = 405;

5. An order with order\_id 501 has been canceled. Change its status to 'Canceled' and remove the tracking number.

UPDATE CustomerOrders SET status = 'Canceled', tracking\_number = NULL WHERE order\_id = 501;

------------ Delete Queries for EmployeeInfo Table: ---------------

1. You need to remove an employee with employee\_id 101 who has left the company. Write an SQL query to delete this employees record from the EmployeeInfo table.

DELETE FROM EmployeeInfo WHERE employee\_id = 101;

2. Delete all employees with a hire\_date before '2020-01-01' who are no longer with the company. Write an SQL query to remove these records.

DELETE FROM EmployeeInfo WHERE hire\_date < TO\_DATE('01-jan-2020');

------------- Delete Queries for ProductInventory Table: ------------

1. A product with product\_id 201 is discontinued and should be removed from the inventory. Write an SQL query to delete this products record from the ProductInventory table.

DELETE FROM ProductInventory WHERE product\_id = 201;

2. Delete all products with a price greater than $500.00 that are no longer in stock (quantity\_in\_stock = 0). Write an SQL query to remove these records.

DELETE FROM ProductInventory WHERE price > 500.00 AND quantity\_in\_stock = 0;

---------------- Delete Queries for LibraryBooks Table: --------------

1. A book with book\_id 102 has been permanently removed from the library's collection. Write an SQL query to delete this book's record from the LibraryBooks table.

DELETE FROM LibraryBooks WHERE book\_id = 102;

2. Remove all books published before the year 1990 that have less than 5 available copies. Write an SQL query to remove these records.

DELETE FROM LibraryBooks WHERE EXTRACT(YEAR FROM publication\_date) < 1990 AND available\_quantity < 5;

------------- Delete Queries for CustomerOrders Table: ---------------

1. An order with order\_id 301 was mistakenly duplicated in the system and needs to be deleted. Write an SQL query to remove one of the duplicate orders.

DELETE FROM CustomerOrders WHERE order\_id = 301;

2. Delete all orders with a total\_amount less than $50.00 that are in 'Canceled' status. Write an SQL query to remove these records.

DELETE FROM CustomerOrders WHERE total\_amount < 50.00 AND status = 'Canceled';