1) Update the categoryName From “Beverages” to "Drinks" in the categories table.

```
1  UPDATE categories
2  SET categoryname = 'Drinks'
3  WHERE categoryname = 'Beverages';
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 145 msec.
```

2) Insert into shipper new record (give any values) Delete that new record from shippers table.

```
INSERT INTO shippers (ShipperID, companyName)
VALUES (4 , 'UnitedPostal');
DELETE FROM shippers
WHERE ShipperID = 'Speedy Express' AND companyName = 'UnitedPostal';
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 140 msec.
```

```
DELETE FROM shippers
WHERE ShipperID = 4 AND companyName = 'UnitedPostal';
```

Data Output | Messages | Notifications

```
DELETE 1

Query returned successfully in 118 msec.
```

3) Update categoryID=1 to categoryID=1001. Make sure related products update their categoryID too. Display the both category and products table to show the cascade.

```sql
ALTER TABLE products
ADD CONSTRAINT fk_category
FOREIGN KEY (categoryID)
REFERENCES categories(categoryID)
ON UPDATE CASCADE;

UPDATE categories
SET categoryID = 1001
WHERE categoryID = 1;

SELECT * FROM categories WHERE categoryID = 1001;

SELECT * FROM products WHERE categoryID = 1001;
```

Data Output | Messages | Notifications

	categoryid [PK] integer	categoryname character varying (60)	description character varying (260)
1	1001	Drinks	Soft drinks, coffees, teas, beers, and ales

```sql
ALTER TABLE products
ADD CONSTRAINT fk_category
FOREIGN KEY (categoryID)
REFERENCES categories(categoryID)
ON UPDATE CASCADE;

UPDATE categories
SET categoryID = 1001
WHERE categoryID = 1;

SELECT * FROM categories WHERE categoryID = 1001;

SELECT * FROM products WHERE categoryID = 1001;
```

ata Output | Messages | Notifications

Showir

productid [PK] integer	productname character varying (100)	quantityperunit character varying (100)	unitprice numeric (10,2)	discontinued boolean	categoryid integer
1	Chai	10 boxes x 20 bags	18.00	false	1001
2	Chang	24 - 12 oz bottles	19.00	false	1001
24	Guarana Fantastica	12 - 355 ml cans	4.50	true	1001

otal rows: 12 | Query complete 00:00:00.217

Delete the categoryID= “3” from categories. Verify that the corresponding records are deleted automatically from products.(HINT: Alter the foreign key on products(categoryID) to add ON UPDATE CASCADE, ON DELETE CASCADe

```sql
ALTER TABLE products
ADD CONSTRAINT fk_category
FOREIGN KEY (categoryID)
REFERENCES categories(categoryID)
ON UPDATE CASCADE
ON DELETE CASCADE;

DELETE FROM categories
WHERE categoryID = 3;

SELECT * FROM products
WHERE categoryID = 3;
```

ata Output | Messages | Notifications

productid [PK] integer	productname character varying (100)	quantityperunit character varying (100)	unitprice numeric (10,2)	discontinued boolean	categoryid integer

Total rows: 0 | Query complete 00:00:00.178

4) Delete the customer = “VINET” from customers. Corresponding customers in orders table should be set to null (HINT: Alter the foreign key on orders(customerID) to use ON DELETE SET NULL)

```sql
ALTER TABLE orders1
ADD CONSTRAINT fk_orders_customers
FOREIGN KEY (customerID)
REFERENCES customers(customerID)
ON DELETE SET NULL;

DELETE FROM customers
WHERE customerID = 'VINET';

SELECT * FROM orders1
WHERE customerID IS NULL;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5

orderid [PK] integer	customerid character varying (5)	employeeid integer	orderdate date	requireddate date	shippeddate date	shipperid integer	freight numeric (10,2)
10248	[null]	5	2013-07-04	2013-08-01	2013-07-16	3	32.38
10274	[null]	6	2013-08-06	2013-09-03	2013-08-16	1	6.01
10295	[null]	2	2013-09-02	2013-09-30	2013-09-10	2	1.15

Total rows: 5 | Query complete 00:00:00.192

5) Insert the following data to Products using UPSERT:

product_id = 100, product_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=5

product_id = 101, product_name = White bread, quantityperunit=5 boxes,unitprice = 13, discontinued = 0, categoryID=5

product_id = 100, product_name = Wheat bread, quantityperunit=10 boxes,unitprice = 13, discontinued = 0, categoryID=5

```sql
INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
VALUES
    (100, 'Wheat bread', '1', 13, FALSE, 5)
ON CONFLICT (productid)
DO UPDATE
SET
    productname = EXCLUDED.productname,
    quantityperunit = EXCLUDED.quantityperunit,
    unitprice = EXCLUDED.unitprice,
    discontinued = EXCLUDED.discontinued,
    categoryID = EXCLUDED.categoryID;

-- Second insert (productid = 101)
INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
VALUES
    (101, 'White bread', '5 boxes', 13, FALSE, 5)
ON CONFLICT (productid)
DO UPDATE
SET
    productname = EXCLUDED.productname,
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 107 msec.
```

6) Write a **MERGE query**:Create **temp table with name:** 'updated_products' and insert

```sql
CREATE Temporary TABLE updated_products(
productID INT,
productName VARCHAR(100),
quantityPerUnit VARCHAR(60),
unitPrice DECIMAL,
discontinued INT,
categoryID INT
);
INSERT INTO updated_products (productid, productname, quantityperunit, unitpr
VALUES
    (100, 'Wheat bread', '10', 20 , 1, 5),
    (101, 'White bread', '5 boxes', 19.99, 0, 5),
    (102, 'Midnight Mango Fizz', '24 - 12 oz bottles', 19, 0, 1),
    (103, 'Savory Fire Sauce', '12 - 550 ml bottles', 10, 0, 2);
SELECT * FROM updated_products;
```

Data Output | Messages | Notifications

SQL

	productid integer	productname character varying (100)	quantityperunit character varying (60)	unitprice numeric	discontinued integer	categoryid integer
1	100	Wheat bread	10	20	1	5
2	101	White bread	5 boxes	19.99	0	5
3	102	Midnight Mango Fizz	24 - 12 oz bottles	19	0	1
4	103	Savory Fire Sauce	12 - 550 ml bottles	10	0	2

Total rows: 4 | Query complete 00:00:00.240

- Update the price and discontinued status for from below table 'updated_products' only if there are matching products and updated_products .discontinued =0

```sql
UPDATE products
SET
    unitprice = u.unitprice,
    discontinued = (u.discontinued = 0) ::boolean
FROM updated_products u
WHERE products.productid = u.productid
  AND u.discontinued = 0;
```

ta Output | Messages | Notifications

```
DATE 1

ery returned successfully in 230 msec.
```

If there are matching products and updated_products .discontinued =1 then delete

```sql
DELETE FROM products
WHERE productid IN (
    SELECT u.productid
    FROM updated_products u
    WHERE u.discontinued = 1
    AND products.productid = u.productid
);
```

Data Output | Messages | Notifications

```
DELETE 1

Query returned successfully in 72 msec.
```

- Insert any new products from updated_products that don't exist in products only if updated_products .discontinued =0.

```sql
INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
SELECT u.productid, u.productname, u.quantityperunit, u.unitprice, (u.discontinued = 0)::boolean, u.categoryID
FROM updated_products u
WHERE u.discontinued = 0
  AND NOT EXISTS (
      SELECT 1 FROM products p WHERE p.productid = u.productid
  );
```

Data Output | Messages | Notifications

```
INSERT 0 2

Query returned successfully in 123 msec.
```

7) List all orders with employee full names. (Inner join)

```sql
SELECT column_name
FROM information_schema.columns
WHERE table_name = 'employees';

SELECT o.order_id, o.order_date, e.first_name || ' ' || e.last_name AS employeeFullName
FROM orders o
INNER JOIN employees e ON o.employee_id = e.employee_id;
```

Data Output | Messages | Notifications

	order_id [PK] smallint	order_date date	employeefullname text
1	10248	1996-07-04	Steven Buchanan
2	10249	1996-07-05	Michael Suyama
3	10250	1996-07-08	Margaret Peacock
4	10251	1996-07-08	Janet Leverling
5	10252	1996-07-09	Margaret Peacock
6	10253	1996-07-10	Janet Leverling
7	10254	1996-07-11	Steven Buchanan
8	10255	1996-07-12	Anne Dodsworth

Total rows: 830 | Query complete 00:00:00.158