

```
use my_db;
```

```
CREATE TABLE IF NOT EXISTS retail_sales (  
    transactions_id SERIAL PRIMARY KEY,  
    sale_date DATE NOT NULL,  
    sale_time TIME NOT NULL,  
    customer_id INT NOT NULL,  
    gender VARCHAR(15) NOT NULL,  
    age INT NOT NULL,  
    category VARCHAR(15) NOT NULL,  
    quantity INT NOT NULL,  
    price_per_unit DECIMAL(10, 2) NOT NULL,  
    cogs DECIMAL(10, 2) NOT NULL, -- Cost of Goods Sold  
    total_sale DECIMAL(10, 2) NOT NULL  
);
```

```
-- First, alter the table to allow NULLs in columns that will have  
NULLs in your data
```

```
ALTER TABLE retail_sales  
MODIFY COLUMN sale_time TIME NULL,  
MODIFY COLUMN customer_id INT NULL,  
MODIFY COLUMN gender VARCHAR(15) NULL,  
MODIFY COLUMN age INT NULL,  
MODIFY COLUMN category VARCHAR(15) NULL,  
MODIFY COLUMN quantity INT NULL,  
MODIFY COLUMN price_per_unit DECIMAL(10,2) NULL,  
MODIFY COLUMN cogs DECIMAL(10,2) NULL,  
MODIFY COLUMN total_sale DECIMAL(10,2) NULL;
```

```
-- Clear existing data (if any)  
TRUNCATE TABLE retail_sales;
```

```
INSERT INTO retail_sales (transactions_id, sale_date, sale_time,  
customer_id, gender, age, category, quantity, price_per_unit, cogs,  
total_sale) VALUES  
(1, '2022-11-05', '09:15:00', 1001, 'Female', 28, 'Clothing', 2, 45.99,  
25.00, 91.98),  
(2, '2022-11-15', '14:30:00', 1001, 'Female', 28, 'Beauty', 3, 35.50,  
18.00, 106.50),  
(3, '2022-12-10', '11:20:00', 1001, 'Female', 28, 'Clothing', 1,  
129.99, 75.00, 129.99),  
(4, '2023-01-20', '16:45:00', 1001, 'Female', 28, 'Beauty', 2, 89.99,  
45.00, 179.98),
```

```
(5, '2022-11-05', '11:30:00', 1002, 'Male', 35, 'Electronics', 1,  
299.99, 180.00, 299.99),  
(6, '2022-12-22', '15:30:00', 1002, 'Male', 35, 'Electronics', 1,  
799.99, 480.00, 799.99),  
(7, '2023-02-14', '13:15:00', 1002, 'Male', 35, 'Electronics', 2,  
199.99, 120.00, 399.98),
```

```
-- Customer 1003 with some NULLs (but sale_date is NOT NULL)  
(8, '2022-11-05', '14:45:00', 1003, 'Female', 42, 'Home', 3, 89.99,  
45.00, 269.97),  
(9, '2022-12-05', '10:30:00', 1003, 'Female', 42, 'Home', 1, 249.99,  
125.00, 249.99),  
(10, '2023-01-25', '14:15:00', 1003, 'Female', 42, 'Home', 2, 149.99,  
75.00, 299.98),
```

```

-- Customer 1007 with NULL gender but valid sale_date
(11, '2022-11-08', '13:15:00', 1007, NULL, 38, 'Electronics', 1,
1299.99, 850.00, 1299.99),
(12, '2022-11-10', '11:20:00', 1007, 'Male', 38, 'Electronics', 1,
899.99, 550.00, 899.99),
(13, '2022-12-05', '14:45:00', 1007, 'Male', 38, 'Clothing', 3, 89.99,
50.00, 269.97),
(14, '2022-12-10', '11:00:00', 1007, 'Male', 38, 'Electronics', 1,
1599.99, 1000.00, 1599.99),
(15, '2023-01-08', '16:30:00', 1007, 'Male', 38, 'Clothing', 2, 199.99,
120.00, 399.98),

-- Customer 1020 with NULL sale_time but valid sale_date
(16, '2023-01-15', NULL, 1020, 'Female', 44, 'Electronics', 1, 2499.99,
1600.00, 2499.99),
(17, '2023-01-12', '16:30:00', 1020, 'Female', 44, 'Beauty', 5, 89.99,
40.00, 449.95),
(18, '2023-02-18', '10:15:00', 1020, 'Female', 44, 'Home', 2, 129.99,
65.00, 259.98),
(19, '2023-02-25', '14:00:00', 1020, 'Female', 44, 'Beauty', 3, 149.99,
75.00, 449.97),

-- Customer 1101 with NULL age but valid sale_date
(20, '2022-11-07', '10:00:00', 1101, 'Female', NULL, 'Clothing', 5,
39.99, 22.00, 199.95),
(21, '2022-12-15', '11:45:00', 1101, 'Female', 25, 'Clothing', 2,
59.99, 35.00, 119.98),
(22, '2023-01-22', '13:30:00', 1101, 'Female', 25, 'Clothing', 3,
45.99, 25.00, 137.97),

-- Customer 1102 with NULL category but valid sale_date
(23, '2022-10-12', '14:30:00', 1102, 'Male', 31, NULL, 1, 399.99,
250.00, 399.99),
(24, '2023-01-25', '13:15:00', 1102, 'Male', 31, 'Electronics', 1,
799.99, 500.00, 799.99),

-- Customer 1103 with NULL quantity but valid sale_date
(25, '2022-11-10', '15:30:00', 1103, 'Female', 45, 'Beauty', NULL,
65.00, 32.50, 130.00),
(26, '2022-12-20', '11:30:00', 1103, 'Female', 45, 'Beauty', 1, 89.99,
40.00, 89.99),
(27, '2023-02-05', '15:45:00', 1103, 'Female', 45, 'Beauty', 4, 24.99,
12.00, 99.96),

-- Customer 1201 with NULL price but valid sale_date
(28, '2022-11-12', '11:45:00', 1201, 'Male', 33, 'Clothing', 3, NULL,
30.00, 167.97),
(29, '2023-01-05', '10:45:00', 1201, 'Male', 33, 'Clothing', 1, 39.99,
20.00, 39.99),

-- Customer 1202 with NULL cogs but valid sale_date
(30, '2022-11-15', '14:00:00', 1202, 'Female', 27, 'Home', 2, 45.50,
NULL, 91.00),

-- Customer 1203 with NULL total_sale but valid sale_date
(31, '2023-02-25', '15:45:00', 1203, 'Female', 45, 'Electronics', 1,
799.99, 500.00, NULL),

```

```

-- ONE-TIME CUSTOMERS (all with valid sale_dates)
(32, '2022-11-18', '10:30:00', 1301, 'Female', 30, 'Clothing', 4,
29.99, 15.00, 119.96),
(33, '2022-11-20', '16:45:00', 1302, 'Male', 40, 'Clothing', 6, 24.99,
12.50, 149.94),
(34, '2022-11-22', '12:15:00', 1303, 'Female', 35, 'Clothing', 5,
49.99, 28.00, 249.95),
(35, '2022-11-25', '15:30:00', 1304, 'Male', 28, 'Clothing', 7, 19.99,
10.00, 139.93),
(36, '2022-11-03', '09:45:00', 1305, 'Female', 32, 'Beauty', 1, 125.00,
60.00, 125.00),
(37, '2022-11-17', '14:20:00', 1306, 'Female', 29, 'Beauty', 2, 45.99,
20.00, 91.98),
(38, '2022-11-28', '16:10:00', 1307, 'Female', 41, 'Beauty', 3, 32.50,
15.00, 97.50),

-- Customer 2001 with valid sale_date (removed NULL sale_date)
(39, '2022-11-10', '10:45:00', 2001, 'Female', 65, 'Clothing', 1,
79.99, 40.00, 79.99),
(40, '2022-12-03', '11:15:00', 2001, 'Female', 65, 'Home', 2, 39.99,
20.00, 79.98),
(41, '2023-01-18', '10:30:00', 2001, 'Female', 65, 'Clothing', 1,
89.99, 45.00, 89.99),

-- Customer 2002 with NULL customer_id but valid sale_date
(42, '2022-12-20', '13:15:00', NULL, 'Male', 68, 'Electronics', 1,
899.99, 550.00, 899.99),
(43, '2023-02-08', '12:30:00', 2002, 'Male', 68, 'Electronics', 1,
199.99, 120.00, 199.99),

-- Customer 3001 with valid sale_date (removed NULL sale_date)
(44, '2022-11-25', '13:45:00', 3001, 'Female', 22, 'Clothing', 4,
24.99, 12.50, 99.96),
(45, '2022-12-12', '15:30:00', 3001, 'Female', 22, 'Beauty', 2, 19.99,
10.00, 39.98),
(46, '2023-02-03', '09:15:00', 3001, 'Female', 22, 'Clothing', 2,
34.99, 17.50, 69.98),

-- Customer 3002 with valid sale_date (removed NULL sale_date)
(47, '2023-01-15', '14:30:00', 3002, 'Male', 24, 'Clothing', 2, 44.99,
25.00, 89.98),

-- CORPORATE CUSTOMERS
(48, '2022-12-15', '09:30:00', 5001, 'Male', 44, 'Clothing', 10, 29.99,
15.00, 299.90),
(49, '2023-01-22', '11:00:00', 5001, 'Male', 44, 'Electronics', 5,
89.99, 54.00, 449.95),

-- Customer 5002 with NULL price/cogs/total but valid sale_date
(50, '2022-11-14', '09:45:00', 5002, 'Male', 50, 'Home', 8, NULL, NULL,
NULL),
(51, '2023-02-12', '16:15:00', 5002, 'Male', 50, 'Home', 6, 49.99,
25.00, 299.94),

-- More records with NULLs but valid sale_dates
(52, '2022-11-06', '08:30:00', 1027, 'Male', 32, 'Clothing', 2, 59.99,
35.00, 119.98),

```

```
(53, '2022-11-07', '12:30:00', 1029, 'Male', 41, 'Electronics', 1,
499.99, 300.00, 499.99),
(54, '2022-11-07', '14:15:00', 1030, 'Female', 36, 'Home', 3, 35.99,
17.50, 107.97),
(55, '2022-11-08', '19:30:00', 1032, 'Female', 45, 'Beauty', 2, 55.00,
25.00, 110.00),
```

-- Records with various NULLs but valid sale_dates

```
(56, '2022-11-16', '12:30:00', 1035, 'Female', 42, 'Beauty', 2, 39.99,
18.00, 79.98),
(57, '2022-11-18', '15:15:00', 1036, 'Male', 31, 'Home', 1, 249.99,
125.00, 249.99),
(58, '2022-11-20', '17:45:00', 1037, 'Female', 28, 'Clothing', 3,
34.99, 17.50, 104.97),
(59, '2022-11-22', '10:00:00', 1038, 'Male', 46, 'Electronics', 1,
599.99, 350.00, 599.99),
```

-- More NULL patterns with valid sale_dates

```
(60, '2022-11-17', '14:30:00', 1040, 'Male', 39, 'Beauty', 1, 199.99,
90.00, 199.99),
(61, '2022-12-10', '16:45:00', 1041, 'Female', 27, 'Home', 2, 89.99,
45.00, 179.98),
(62, '2022-12-20', '13:15:00', NULL, 'Female', 38, 'Electronics', 1,
899.99, 550.00, 899.99),
(63, '2023-01-05', '10:45:00', 1044, NULL, 32, 'Clothing', 3, 64.99,
35.00, 194.97),
(64, '2023-01-12', '15:30:00', 1045, 'Female', NULL, 'Beauty', 2,
74.99, 35.00, 149.98),
(65, '2023-01-18', '17:15:00', 1046, 'Male', 29, NULL, 1, 349.99,
175.00, 349.99),
(66, '2023-01-22', '11:00:00', 1047, 'Female', 36, 'Clothing', NULL,
84.99, 45.00, 169.98),
(67, '2023-01-28', '14:45:00', 1048, 'Male', 43, 'Electronics', 1,
NULL, 950.00, 1499.99),
(68, '2023-02-08', '12:30:00', 1050, 'Male', 47, 'Beauty', 1, 299.99,
NULL, 299.99),
(69, '2023-02-18', '10:30:00', 1052, 'Male', 31, 'Clothing', 2, 94.99,
50.00, NULL),
(70, '2023-02-20', '15:45:00', 1021, 'Male', 36, 'Electronics', 1,
1899.99, 1200.00, 1899.99),
```

-- Partial NULLs with valid sale_date

```
(71, '2023-03-01', '11:00:00', 1060, 'Male', 40, 'Clothing', 2, 59.99,
30.00, 119.98),
(72, '2022-10-05', '10:15:00', 1018, 'Female', 26, 'Clothing', 2,
35.99, 18.00, 71.98),
(73, '2022-12-08', '11:45:00', 1022, 'Female', 39, 'Beauty', 1, 89.99,
40.00, 89.99),
(74, '2022-12-15', '16:20:00', 1023, 'Male', 27, 'Clothing', 3, 42.99,
22.00, 128.97),
(75, '2023-01-10', '09:30:00', 1024, 'Female', 34, 'Home', 2, 65.50,
32.75, 131.00);
```

-- Verify data insertion

```
SELECT COUNT(*) AS total_records FROM retail_sales;
SELECT * FROM retail_sales LIMIT 10;
```

```

-- Checked for NULL values across all columns.
SELECT * FROM retail_sales
WHERE transactions_id IS NULL;

SELECT * FROM retail_sales
WHERE sale_time IS NULL;

SELECT * FROM retail_sales
WHERE sale_date IS NULL;

SELECT * FROM retail_sales
WHERE customer_id IS NULL;

SELECT * FROM retail_sales
WHERE gender IS NULL;

SELECT * FROM retail_sales
WHERE age IS NULL;

SELECT * FROM retail_sales
WHERE category IS NULL;

SELECT * FROM retail_sales
WHERE quantity IS NULL;

SELECT * FROM retail_sales
WHERE cogs IS NULL;

SELECT * FROM retail_sales
WHERE total_sale IS NULL;

-- Removed incomplete records to ensure data integrity.
SELECT * FROM retail_sales
WHERE transactions_id IS NULL
   OR sale_date IS NULL
   OR sale_time IS NULL
   OR customer_id IS NULL
   OR gender IS NULL
   OR age IS NULL
   OR category IS NULL
   OR quantity IS NULL
   OR price_per_unit IS NULL
   OR cogs IS NULL
   OR total_sale IS NULL;

set sql_safe_updates = 0;

DELETE FROM retail_sales
WHERE transactions_id IS NULL
   OR sale_date IS NULL
   OR sale_time IS NULL
   OR customer_id IS NULL
   OR gender IS NULL
   OR age IS NULL
   OR category IS NULL
   OR quantity IS NULL
   OR price_per_unit IS NULL
   OR cogs IS NULL

```

```

        OR total_sale IS NULL;

-- Exploratory Data Analysis (EDA)
/*
Q1. Total Records Count
*/
SELECT COUNT(*) AS TotalRecords
FROM retail_sales;

/*
Q2. Count of unique customers
*/
SELECT COUNT(DISTINCT customer_id) AS TotalCustomers
FROM retail_sales;

/*
Q3. HOW MANY CATEGORIES DO WE HAVE ?
*/
SELECT COUNT(DISTINCT Category) AS TotalCategories
FROM retail_sales;

-- DATA ANALYSIS OVER BUSINESS KEY PROBLEM
/*
Q1. Write a SQL query to retrieve all columns for sales made on
'2022-11-05':
*/
SELECT
    *
FROM retail_sales
WHERE sale_date = '2022-11-05';

/*
Q2. Write a SQL query to retrieve all transactions where the category is
'Clothing' and the quantity sold is more than 3 in the month of Nov-
2022
*/

SELECT
    *
FROM retail_sales
WHERE category = 'Clothing'
    AND
        quantity > 3
    AND
        sale_date BETWEEN '2022-11-01' AND '2022-11-30';

/*
Q3. Write a SQL query to calculate the total sales (total_sale) for each
category.
*/
SELECT
    category,
    SUM(total_sale) AS TotalSales
FROM retail_sales
GROUP BY

```

```
        category
ORDER BY
    TotalSales DESC;
```

```
SELECT
    category,
    count(category) AS totalquantity
FROM retail_sales
GROUP BY category;
```

```
/*
Q4. Write a SQL query to find the average age of customers who
purchased items from the 'Beauty' category.
*/
```

```
SELECT
    category,
    ROUND(AVG(age)) AS AverageAge
FROM retail_sales
WHERE category = 'Beauty';
```

```
/*
Q5. Write a SQL query to find all transactions where the total_sale is
greater than 1000.
*/
```

```
SELECT
    transactions_id,
    total_sale
FROM retail_sales
WHERE total_sale > 1000;
```

```
/*
Q6. Write a SQL query to find the total number of transactions
(transaction_id) made by each gender in each category.
*/
```

```
SELECT
    gender,
    category,
    COUNT(*) AS TotalTransactions
FROM retail_sales
GROUP BY 1,2
ORDER BY 2;
```

```
/*
Q7. Write a SQL query to calculate the average sale for each month. Find
out best selling month in each year:
*/
```

```
SELECT
    Year,
    Month,
    AverageSale
FROM (
    SELECT
        YEAR(sale_date) AS Year,
        MONTH(sale_date) AS Month,
        ROUND(AVG(total_sale),2) AS AverageSale,
```

```

        DENSE_RANK() OVER(PARTITION BY YEAR(sale_date) ORDER BY
AVG(total_sale) DESC) AS AverageRank
        FROM retail_sales
        GROUP BY 1,2
    )t
WHERE AverageRank = 1;

```

```

/*
Q8. Write a SQL query to find the top 5 customers based on the highest
total sales:
*/

```

```

SELECT
    customer_id,
    COUNT(*) AS Purchase,
    SUM(total_sale) AS TotalSales
FROM retail_sales
GROUP BY
    customer_id
ORDER BY
    TotalSales DESC
LIMIT 5;

```

```

SELECT
    *
FROM retail_sales
WHERE customer_id = 1003;

```

```

-- Beauty category attracts younger customers
SELECT category,
    ROUND(AVG(age), 1) as avg_age,
    MIN(age) as min_age,
    MAX(age) as max_age
FROM retail_sales
GROUP BY category
ORDER BY avg_age;

```

```

/*
Q9. Write a SQL query to find the number of unique customers who
purchased items from each category:
*/

```

```

SELECT
    category,
    COUNT(DISTINCT(customer_id)) AS unique_customers
FROM retail_sales
GROUP BY category;

```

```

/*
Q10. Write a SQL query to create each shift and number of orders
(Example Morning <12, Afternoon Between 12 & 17, Evening >17):
*/

```

```

WITH hourly_sale
AS
(

```



```
SELECT
    *,
    CASE
        WHEN HOUR(sale_time) < 12 THEN "Morning"
        WHEN HOUR(sale_time) BETWEEN 12 AND 17 THEN "Afternoon"
        ELSE "Evening"
    END AS Shift
FROM retail_sales
)
SELECT
    Shift,
    COUNT(*) AS TotalOrders
FROM hourly_sale
GROUP BY Shift;
```