Case Study - Retail Analytics - IV

Challenge16:

Write an SQL query that segments customers based on the total quantity of products they have purchased. Also, count the number of customers in each segment which will help us target a particular segment for marketing.

Hint:

- *Use the customer_profiles and sales_transaction tables.*
- *Create a separate table named customer_segment and create the segments on the total quantity of the purchased products.*
- *To segment customers based on their purchasing behavior for targeted marketing campaigns. Create Customer segments on the following criteria-*

Total Quantity of Products Purchased	Customer Segment
1-10	Low
11-30	Mid
>30	High

- *The resulting table should count the number of customers in different customer segments.*

- *The resulting table should count the number of customers in different customer segments.*
- *Return the result table in any order.*

Output Format:

CustomerSegment	COUNT(*)
Med	NUM
Low	NUM
High	NUM

Note: *The NUM in the output table denotes Number of customers.*

CustomerID	Age	Gender	Location	JoinDate
1	63	Other	East	01/01/20
2	63	Male	North	02/01/20
3	34	Other	North	03/01/20
4	19	Other	Unknown	04/01/20
5	57	Male	North	05/01/20
6	22	Other	South	06/01/20
7	56	Other	East	07/01/20
8	65	Female	East	08/01/20
9	33	Male	West	09/01/20
10	34	Male	East	10/01/20
11	44	Other	North	11/01/20
12	24	Other	East	13/01/20
13	69	Male	East	14/01/20
14	25	Male	North	15/01/20
15	25	Male	North	16/01/20

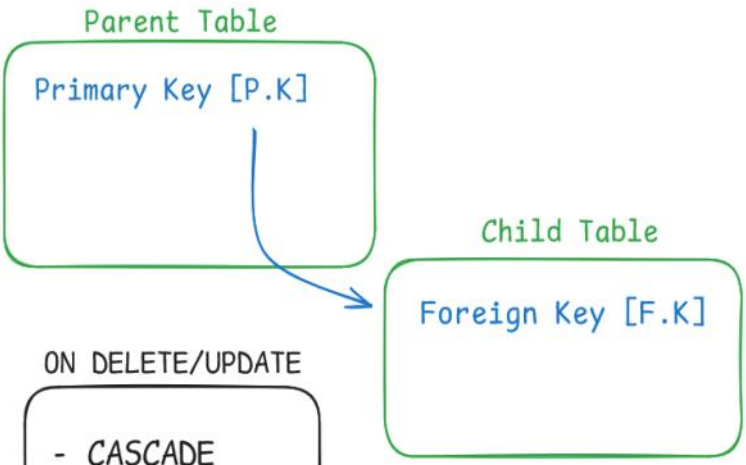
TransactionID	CustomerID	ProductID	QuantityPurchased	TransactionDate	Price	TransactionDate_updated
1	103	120	3	01/01/23	30.43	2023-01-01
2	436	126	1	01/01/23	15.19	2023-01-01
3	861	55	3	01/01/23	67.76	2023-01-01
4	271	27	2	01/01/23	65.77	2023-01-01
5	107	118	1	01/01/23	14.55	2023-01-01
6	72	53	1	01/01/23	26.27	2023-01-01
7	701	39	2	01/01/23	95.92	2023-01-01
8	21	65	4	01/01/23	17.19	2023-01-01
9	615	145	4	01/01/23	66	2023-01-01
10	122	158	2	01/01/23	22.27	2023-01-01
11	467	181	2	01/01/23	69	2023-01-01
12	215	13	3	01/01/23	18.78	2023-01-01
13	331	21	1	01/01/23	14.29	2023-01-01
14	459	147	3	01/01/23	53.98	2023-01-01
15	88	53	2	01/01/23	26.27	2023-01-01

```sql
-- Challenge 16: Customer Segmentation by Quantity

USE retail_analytics;
SELECT * FROM customers;
SELECT * FROM sales;

CREATE TABLE customer_segment AS
    SELECT
        CustomerID,
        CASE
            WHEN TotalQuantity BETWEEN 1 AND 10 THEN 'Low'
            WHEN TotalQuantity BETWEEN 11 AND 30 THEN 'Med'
            WHEN TotalQuantity > 30 THEN 'High'
        ELSE 'None'
            END AS CustomerSegment
    FROM (
        SELECT
            c.CustomerID,
            SUM(s.QuantityPurchased) AS TotalQuantity
        FROM Customers c
        JOIN sales s
        ON c.CustomerID = s.CustomerID
        GROUP BY CustomerID
    ) AS customer_totals;

SELECT * FROM customer_segment;
```

CustomerID	CustomerSegment
103	Med
436	Med
861	Med
271	Low
107	Low
72	Low
701	Med
21	Med
615	Low
122	Med
467	Med
215	Med
331	Low
459	Med
88	Med
373	Low

```sql
SELECT
    CustomerSegment,
    COUNT(*)
FROM customer_segment
GROUP BY CustomerSegment;
```

CustomerSegment	COUNT(*)
Med	559
Low	423
High	7



ON DELETE/UPDATE

- CASCADE
- SET NULL
- SET DEFAULT
- RESTRICT
- NO ACTION

```
CREATE TABLE Customers(
    customer_id INT PRIMARY KEY,
    name VARCHAR(50)
);

CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE
);
```

```
416 •  DESC Customers;
417
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
name	varchar...	YES		NULL	

```
417 •  DESC Orders;
418
```

Result Grid | Filter Rows: | Export: | W

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES		NULL	
order_date	date	YES		NULL	

```
419 •  ALTER TABLE Orders
420    ADD CONSTRAINT fk_customer
421    FOREIGN KEY(customer_id)
422    REFERENCES Customers(customer_id)
423    ON DELETE CASCADE;
424
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	

```
INSERT INTO customers (customer_id, name) VALUES
(1, 'Alice Johnson'),
(2, 'Bob Smith'),
(3, 'Charlie Lee'),
(4, 'Diana Patel'),
(5, 'Ethan Clark'),
(6, 'Fiona Davis'),
(7, 'George Brown'),
(8, 'Hannah Wilson'),
(9, 'Ian Thompson'),
(10, 'Julia Martinez');
SELECT * FROM customers;
```

customer_id	name
1	Alice Johnson
2	Bob Smith
3	Charlie Lee
4	Diana Patel
5	Ethan Clark
6	Fiona Davis
7	George Brown
8	Hannah Wilson
9	Ian Thompson
10	Julia Martinez

```
INSERT INTO orders (order_id, customer_id, order_date) VALUES
(101, 1, '2024-01-05'),
(102, 2, '2024-01-15'),
(103, 1, '2024-02-20'),
(104, 3, '2024-02-25'),
(105, 4, '2024-03-05'),
(106, 5, '2024-03-15'),
(107, 2, '2024-04-01'),
(108, 6, '2024-04-10'),
(109, 7, '2024-04-12'),
(110, 3, '2024-04-25'),
(111, 8, '2024-05-01'),
(112, 9, '2024-05-10'),
(113, 10, '2024-05-20');

SELECT * FROM Orders;
```

order_id	customer_id	order_date
101	1	2024-01-05
102	2	2024-01-15
103	1	2024-02-20
104	3	2024-02-25
105	4	2024-03-05
106	5	2024-03-15
107	2	2024-04-01
108	6	2024-04-10
109	7	2024-04-12
110	3	2024-04-25
111	8	2024-05-01
112	9	2024-05-10
113	10	2024-05-20
NULL	NULL	NULL

```
-- ON DELETE CASCADE
DELETE FROM Customers WHERE customer_id = 10;
```

customer_id	name
1	Alice Johnson
2	Bob Smith
3	Charlie Lee
4	Diana Patel
5	Ethan Clark
6	Fiona Davis
7	George Brown
8	Hannah Wilson
9	Ian Thompson
NULL	NULL

order_id	customer_id	order_date
101	1	2024-01-05
102	2	2024-01-15
103	1	2024-02-20
104	3	2024-02-25
105	4	2024-03-05
106	5	2024-03-15
107	2	2024-04-01
108	6	2024-04-10
109	7	2024-04-12
110	3	2024-04-25
111	8	2024-05-01
112	9	2024-05-10

```
-- ON UPDATE CASCADE
ALTER TABLE Orders
ADD CONSTRAINT fk_update_customer
FOREIGN KEY(customer_id)
REFERENCES Customers(customer_id)
ON UPDATE CASCADE;

UPDATE Customers SET customer_id = 10
WHERE customer_id = 9;
```

customer_id	name
1	Alice Johnson
2	Bob Smith
3	Charlie Lee
4	Diana Patel
5	Ethan Clark
6	Fiona Davis
7	George Brown
8	Hannah Wilson
10	Ian Thompson
NULL	NULL

order_id	customer_id	order_date
101	1	2024-01-05
102	2	2024-01-15
103	1	2024-02-20
104	3	2024-02-25
105	4	2024-03-05
106	5	2024-03-15
107	2	2024-04-01
108	6	2024-04-10
109	7	2024-04-12
110	3	2024-04-25
111	8	2024-05-01
112	10	2024-05-10
NULL	NULL	NULL

```
-- ON UPDATE SET NULL
ALTER TABLE Orders
ADD CONSTRAINT fk_set_null
FOREIGN KEY(customer_id)
REFERENCES Customers(customer_id)
ON UPDATE SET NULL;

UPDATE Customers SET customer_id = 9
WHERE customer_id = 10;
```

customer_id	name
1	Alice Johnson
2	Bob Smith
3	Charlie Lee
4	Diana Patel
5	Ethan Clark
6	Fiona Davis
7	George Brown
8	Hannah Wilson
9	Ian Thompson
NULL	NULL

order_id	customer_id	order_date
101	1	2024-01-05
102	2	2024-01-15
103	1	2024-02-20
104	3	2024-02-25
105	4	2024-03-05
106	5	2024-03-15
107	2	2024-04-01
108	6	2024-04-10
109	7	2024-04-12
110	3	2024-04-25
111	8	2024-05-01
112	NULL	2024-05-10
NULL	NULL	NULL

```
-- ON DELETE SET NULL
ALTER TABLE Orders
ADD CONSTRAINT fk_delete_set_null
FOREIGN KEY(customer_id)
REFERENCES Customers(customer_id)
ON DELETE SET NULL;

DELETE FROM Customers WHERE customer_id = 8;
```

customer_id	name
1	Alice Johnson
2	Bob Smith
3	Charlie Lee
4	Diana Patel
5	Ethan Clark
6	Fiona Davis
7	George Brown
9	Ian Thompson
NULL	NULL

order_id	customer_id	order_date
101	1	2024-01-05
102	2	2024-01-15
103	1	2024-02-20
104	3	2024-02-25
105	4	2024-03-05
106	5	2024-03-15
107	2	2024-04-01
108	6	2024-04-10
109	7	2024-04-12
110	3	2024-04-25
111	NULL	2024-05-01
112	NULL	2024-05-10
NULL	NULL	NULL

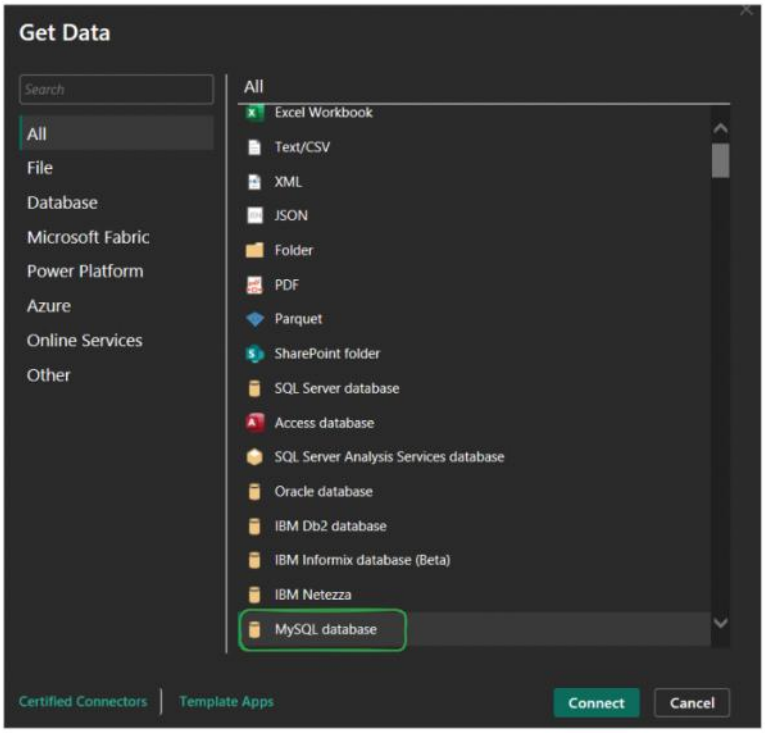
```
SELECT
    CONSTRAINT_NAME,
    CONSTRAINT_TYPE,
    TABLE_NAME
FROM
    INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE
    TABLE_NAME = 'orders'
    AND TABLE_SCHEMA = 'trigger_demo';
```

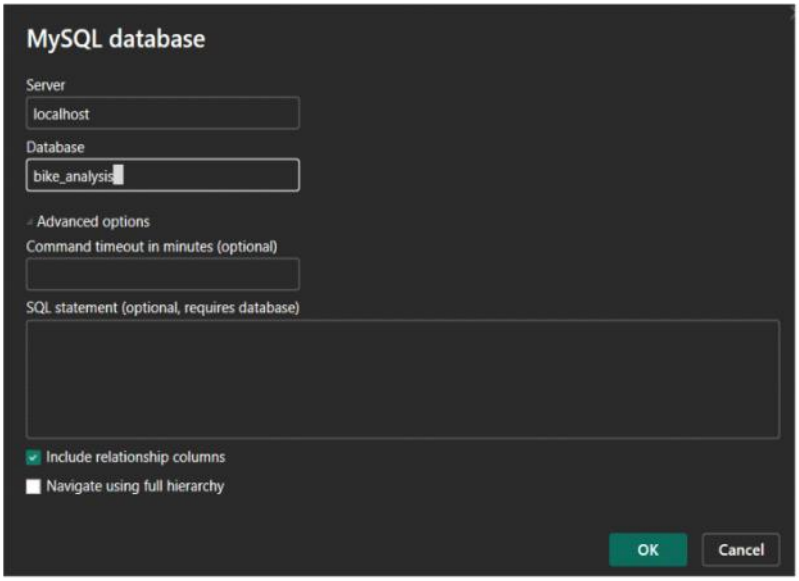
CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
PRIMARY	PRIMARY KEY	orders
fk_customer	FOREIGN KEY	orders
fk_delete_set_null	FOREIGN KEY	orders
fk_set_null	FOREIGN KEY	orders
fk_update_customer	FOREIGN KEY	orders

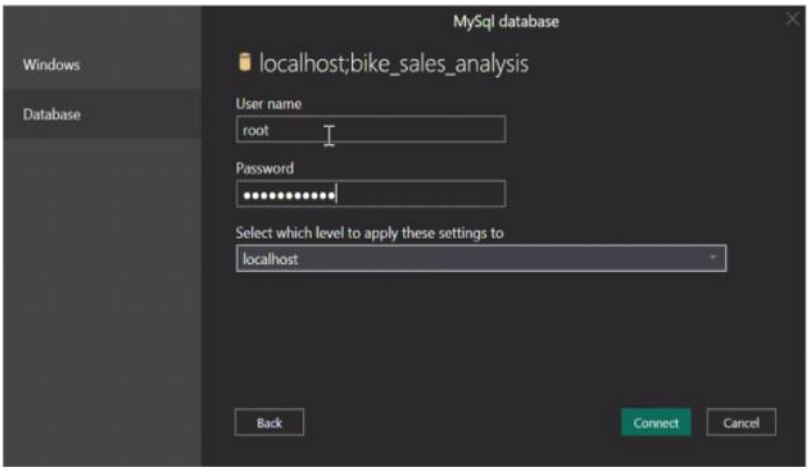
```
ALTER TABLE Orders
ADD CONSTRAINT fk_update_restrict
FOREIGN KEY (customer_id)
REFERENCES Customers(customer_id)
ON UPDATE RESTRICT;

UPDATE Customers
SET customer_id = 12
WHERE customer_id = 2;
```

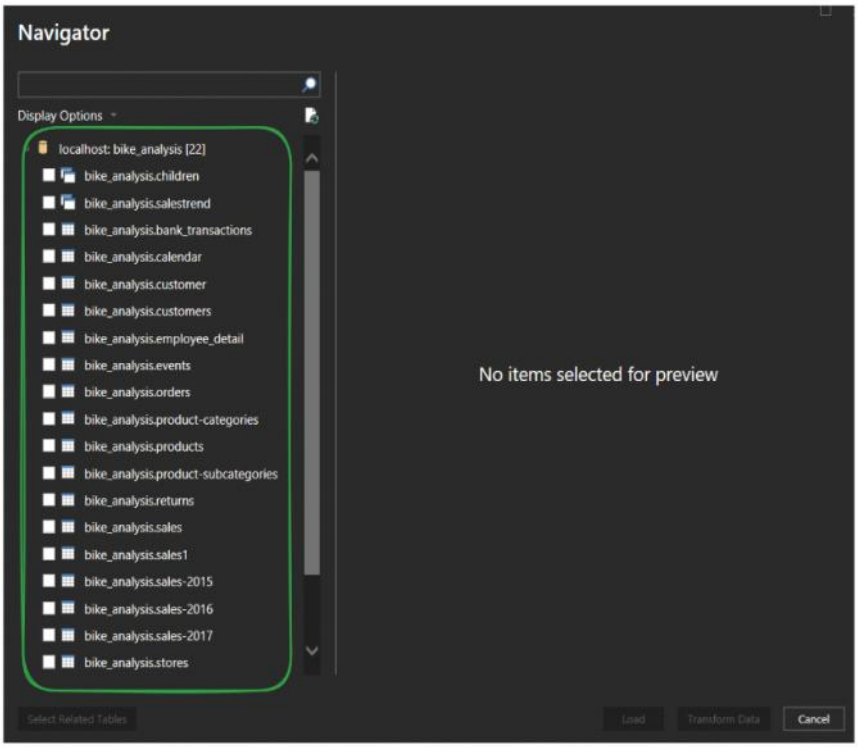
```
Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails
(`trigger_demo`.`orders`, CONSTRAINT `fk_update_restrict` FOREIGN KEY (`customer_id`)
REFERENCES `customers` (`customer_id`) ON UPDATE RESTRICT)
```

MySQL Connector




MySQL database
Server
localhost
Database
bike_analysis
Advanced options
Command timeout in minutes (optional)
SQL statement (optional, requires database)
Include relationship columns
Navigate using full hierarchy
OK
Cancel


MySql database
Windows
Database
localhost;bike_sales_analysis
User name
root
Password
Select which level to apply these settings to
localhost
Back
Connect
Cancel


Navigator
Display Options
localhost: bike_analysis [22]
bike_analysis.children
bike_analysis.salestrend
bike_analysis.bank_transactions
bike_analysis.calendar
bike_analysis.customer
bike_analysis.customers
bike_analysis.employee_detail
bike_analysis.events
bike_analysis.orders
bike_analysis.product-categories
bike_analysis.products
bike_analysis.product-subcategories
bike_analysis.returns
bike_analysis.sales
bike_analysis.sales1
bike_analysis.sales-2015
bike_analysis.sales-2016
bike_analysis.sales-2017
bike_analysis.stores
No items selected for preview
Select Related Tables
Load
Transform Data
Cancel