Arlington Sprouts Database Documentation

By: Cesar Frayre

Overview

This document details various SQL queries and operations performed on the Arlington Sprouts database, which manages a sprouts retail business. The queries cover various aspects including customer management, inventory control, sales analysis, and data integrity checks.

Query Descriptions

Customer Information Queries

1. **Customer Name Pattern Search (Q1)**
   - Purpose: Retrieve customer information based on name patterns
   - Filters: Names beginning with 'J' or 'M'
   - Returns: Full name, address, city, state, zipcode
   - Sample Results: Matched customers like Jasper Johns, Ekene Maduka, Tatsuo Miyajima

2. **Contract Schedule Analysis (Q2)**
   - Purpose: Monitor dealer-shop course schedules
   - Filters: Events after 2:00 PM
   - Returns: Contract and course scheduling information

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

```
SELECT D.Dname AS DealerName, C.Cname AS CourseName, C.Sdate AS CourseDate, DATE_FORMAT(C.Ctime, '%h:%i %p') AS CourseTime FROM contract AS C JOIN dealer AS D ON C.Did = D.Did WHERE TIME(C.Ctime)
> '14:00:00';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

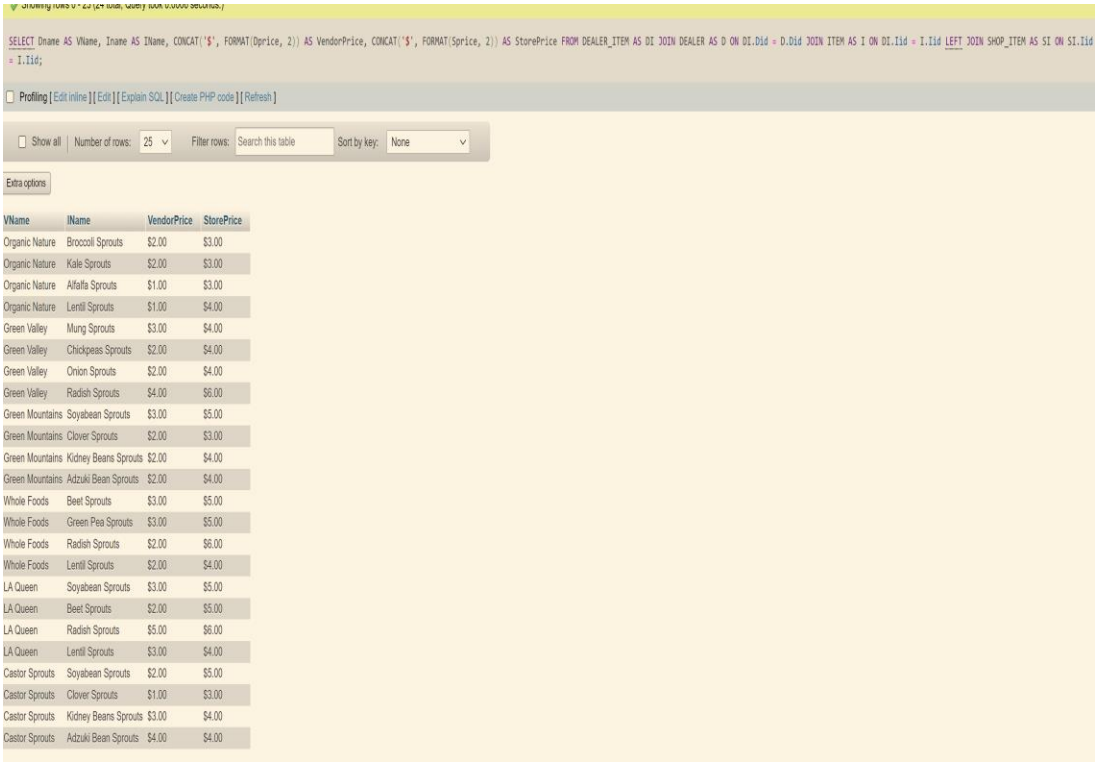
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

DealerName	CourseName	CourseDate	CourseTime
Green Mountains	Sprout Greens	2024-06-25	03:00 PM
Whole Foods	Sprout Certifications	2024-06-25	04:00 PM
Whole Foods	Sprout Certifications	2024-06-26	04:00 PM
LA Queen	Sprout Sandwiches	2024-07-26	03:00 PM
LA Queen	Sprout Sandwiches	2024-07-26	04:00 PM
LA Queen	Microgreens	2024-07-26	05:00 PM

Inventory and Pricing Analysis

3. **Product Pricing Overview (Q3)**
   - Purpose: Comprehensive price comparison across dealers
   - Returns: Vendor and store prices for all sprout products by dealer



```
SELECT Dname AS VName, Iname AS IName, CONCAT('$', FORMAT(Dprice, 2)) AS VendorPrice, CONCAT('$', FORMAT(Sprice, 2)) AS StorePrice FROM DEALER_ITEM AS DI JOIN DEALER AS D ON DI.Did = D.Did JOIN ITEM AS I ON DI.Iid = I.Iid LEFT JOIN SHOP_ITEM AS SI ON SI.Iid
= I.Iid;
```

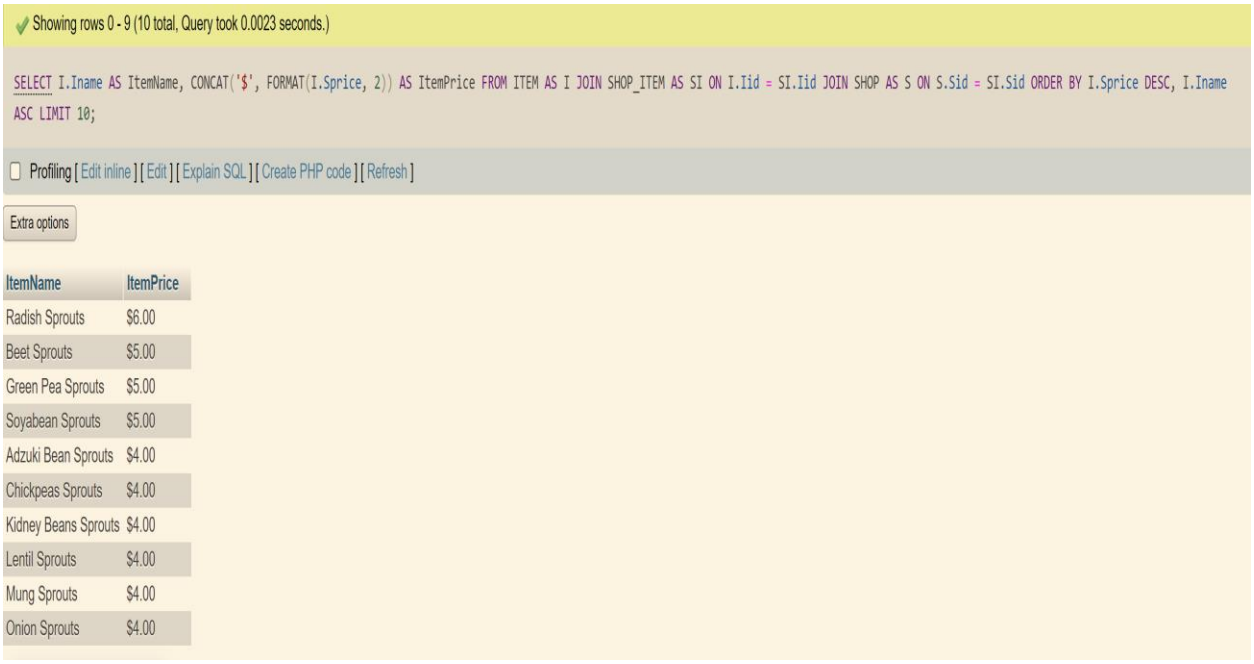
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

VName	IName	VendorPrice	StorePrice
Organic Nature	Broccoli Sprouts	$2.00	$3.00
Organic Nature	Kale Sprouts	$2.00	$3.00
Organic Nature	Alfalfa Sprouts	$1.00	$3.00
Organic Nature	Lentil Sprouts	$1.00	$4.00
Green Valley	Mung Sprouts	$3.00	$4.00
Green Valley	Chickpeas Sprouts	$2.00	$4.00
Green Valley	Onion Sprouts	$2.00	$4.00
Green Valley	Radish Sprouts	$4.00	$6.00
Green Mountains	Soyabean Sprouts	$3.00	$5.00
Green Mountains	Clover Sprouts	$2.00	$3.00
Green Mountains	Kidney Beans Sprouts	$2.00	$4.00
Green Mountains	Adzuki Bean Sprouts	$2.00	$4.00
Whole Foods	Beet Sprouts	$3.00	$5.00
Whole Foods	Green Pea Sprouts	$3.00	$5.00
Whole Foods	Radish Sprouts	$2.00	$6.00
Whole Foods	Lentil Sprouts	$2.00	$4.00
LA Queen	Soyabean Sprouts	$3.00	$5.00
LA Queen	Beet Sprouts	$2.00	$5.00
LA Queen	Radish Sprouts	$5.00	$6.00
LA Queen	Lentil Sprouts	$3.00	$4.00
Castor Sprouts	Soyabean Sprouts	$2.00	$5.00
Castor Sprouts	Clover Sprouts	$1.00	$3.00
Castor Sprouts	Kidney Beans Sprouts	$3.00	$4.00
Castor Sprouts	Adzuki Bean Sprouts	$4.00	$4.00

4. **Top-Priced Items (Q4)**
   - Purpose: Identify highest-priced inventory
   - Returns: Top 10 items sorted by price in descending order



Showing rows 0 - 9 (10 total, Query took 0.0023 seconds.)

```
SELECT I.Iname AS ItemName, CONCAT('$', FORMAT(I.Sprice, 2)) AS ItemPrice FROM ITEM AS I JOIN SHOP_ITEM AS SI ON I.Iid = SI.Iid JOIN SHOP AS S ON S.Sid = SI.Sid ORDER BY I.Sprice DESC, I.Iname
ASC LIMIT 10;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

ItemName	ItemPrice
Radish Sprouts	$6.00
Beet Sprouts	$5.00
Green Pea Sprouts	$5.00
Soyabean Sprouts	$5.00
Adzuki Bean Sprouts	$4.00
Chickpeas Sprouts	$4.00
Kidney Beans Sprouts	$4.00
Lentil Sprouts	$4.00
Mung Sprouts	$4.00
Onion Sprouts	$4.00

5. **Price Margin Analysis**
   - Purpose: Identify high-margin items
   - Condition: Store price > 2 × vendor price

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT D.Dname AS Vname, I.Iname AS Iname, CONCAT('$', FORMAT(DI.Dprice, 2)) AS VendorPrice, CONCAT('$', FORMAT(I.Sprice, 2)) AS StorePrice FROM DEALER_ITEM AS DI JOIN DEALER AS D ON DI.Did =
D.Did JOIN ITEM AS I ON DI.Iid = I.Iid JOIN SHOP_ITEM AS SI ON SI.Iid = I.Iid WHERE I.Sprice > (DI.Dprice * 2);
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

Vname	Iname	VendorPrice	StorePrice
Organic Nature	Alfalfa Sprouts	$1.00	$3.00
Organic Nature	Lentil Sprouts	$1.00	$4.00
Whole Foods	Radish Sprouts	$2.00	$6.00
LA Queen	Beet Sprouts	$2.00	$5.00
Castor Sprouts	Soyabean Sprouts	$2.00	$5.00
Castor Sprouts	Clover Sprouts	$1.00	$3.00

6. **Specific Margin Items (Q6)**
   - Purpose: Find items with exact double markup
   - Condition: Store price = 2 × vendor price
   - Sorting: Ordered by vendor name

```
SELECT D.Dname AS VendorName, I.Iname AS ItemName, CONCAT('$', FORMAT(DI.Dprice, 2)) AS VendorPrice, CONCAT('$', FORMAT(I.Sprice, 2)) AS StorePrice FROM DEALER_ITEM AS DI JOIN DEALER AS D ON
DI.Did = D.Did JOIN ITEM AS I ON DI.Iid = I.Iid WHERE ABS(I.Sprice - DI.Dprice) = 2 ORDER BY D.Dname;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

VendorName	ItemName	VendorPrice	StorePrice
Castor Sprouts	Clover Sprouts	$1.00	$3.00
Green Mountains	Soyabean Sprouts	$3.00	$5.00
Green Mountains	Kidney Beans Sprouts	$2.00	$4.00
Green Mountains	Adzuki Bean Sprouts	$2.00	$4.00
Green Valley	Chickpeas Sprouts	$2.00	$4.00
Green Valley	Onion Sprouts	$2.00	$4.00
Green Valley	Radish Sprouts	$4.00	$6.00
LA Queen	Soyabean Sprouts	$3.00	$5.00
Organic Nature	Alfalfa Sprouts	$1.00	$3.00
Whole Foods	Beet Sprouts	$3.00	$5.00
Whole Foods	Green Pea Sprouts	$3.00	$5.00
Whole Foods	Lentil Sprouts	$2.00	$4.00

Sales and Order Analysis

7. **Customer Order Summary (Q7)**
   - Purpose: Detailed order analysis
   - Returns: Order date, customer name, total amount, items ordered
   - Grouping: By order date and customer

Showing rows 0 - 24 (27 total, Query took 0.0007 seconds.)

```
SELECT O.Odate AS OrderDate, C.Cname AS CustomerName, CONCAT('$', FORMAT(SUM(O.Amount), 2)) AS Amount, SUM(OI.Icount) AS ItemsOrdered FROM ORDERS AS O JOIN CUSTOMER AS C ON O.Cid = C.cId JOIN ORDER_ITEM AS OI ON O.Oid =
OI.oId JOIN ITEM AS I ON OI.iId = I.Iid GROUP BY O.Odate, C.Cname ORDER BY O.Odate ASC, C.Cname ASC;
```

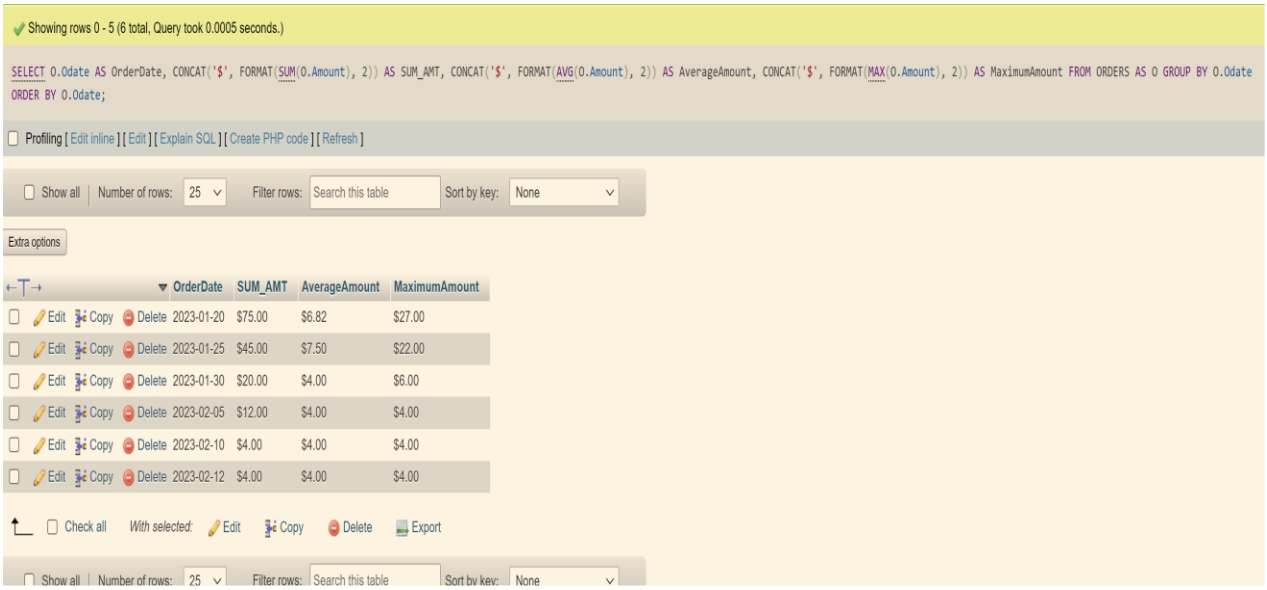
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 | > >> | Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

OrderDate	CustomerName	Amount	ItemsOrdered
2023-01-20	Abdur Rahman	$24.00	3
2023-01-20	Abed Abdi	$3.00	1
2023-01-20	Ismail Gulg	$3.00	1
2023-01-20	Jasper Johns	$5.00	1
2023-01-20	Kalipada Ghoshal	$4.00	1
2023-01-20	Manishi Dey	$81.00	6
2023-01-20	Nandalal Bose	$4.00	1
2023-01-20	Raja Ravi Varma	$5.00	1
2023-01-20	Shakir Ali	$4.00	1
2023-01-20	Sunil Das	$5.00	1
2023-01-20	Winslow Homer	$3.00	1
2023-01-25	Albert Bierstadt	$4.00	1
2023-01-25	Edward Hopper	$4.00	1
2023-01-25	Georgia O Keeffe	$4.00	1
2023-01-25	Ismail Gulg	$88.00	5
2023-01-25	Kalipada Ghoshal	$5.00	1
2023-01-25	Sunil Das	$6.00	1
2023-01-30	Ekene Maduka	$3.00	1
2023-01-30	Ismail Gulg	$4.00	1
2023-01-30	Modupeola Fadugba	$6.00	1
2023-01-30	Olu Amoda	$3.00	1
2023-01-30	Sunil Das	$4.00	1
2023-02-05	Koki Tanaka	$4.00	1
2023-02-05	Li Chen	$4.00	1
2023-02-05	Tatsuo Miyajima	$4.00	1

8. **Daily Sales Statistics (Q8)**
   - Purpose: Daily sales performance metrics
   - Returns: Sum, average, and maximum order amounts per date



Showing rows 0 - 5 (6 total, Query took 0.0005 seconds.)

```
SELECT O.Odate AS OrderDate, CONCAT('$', FORMAT(SUM(O.Amount), 2)) AS SUM_AMT, CONCAT('$', FORMAT(AVG(O.Amount), 2)) AS AverageAmount, CONCAT('$', FORMAT(MAX(O.Amount), 2)) AS MaximumAmount FROM ORDERS AS O GROUP BY O.Odate
ORDER BY O.Odate;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	OrderDate	SUM_AMT	AverageAmount	MaximumAmount
Edit Copy Delete	2023-01-20	$75.00	$6.82	$27.00
Edit Copy Delete	2023-01-25	$45.00	$7.50	$22.00
Edit Copy Delete	2023-01-30	$20.00	$4.00	$6.00
Edit Copy Delete	2023-02-05	$12.00	$4.00	$4.00
Edit Copy Delete	2023-02-10	$4.00	$4.00	$4.00
Edit Copy Delete	2023-02-12	$4.00	$4.00	$4.00

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

9. **Customer Spending Analysis (Q9)**
   - Purpose: Track customer spending patterns
   - Returns: Total amount spent per customer
   - Sorting: Ascending order by amount spent

```
SELECT C.Cname AS CustomerName, CONCAT('$', FORMAT(SUM(O.Amount), 2)) AS TotalAmountSpent FROM CUSTOMER AS C JOIN ORDERS AS O ON C.cId = O.Cid JOIN SHOP AS S ON O.Sid = S.Sid GROUP BY C.Cname
ORDER BY SUM(O.Amount) ASC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

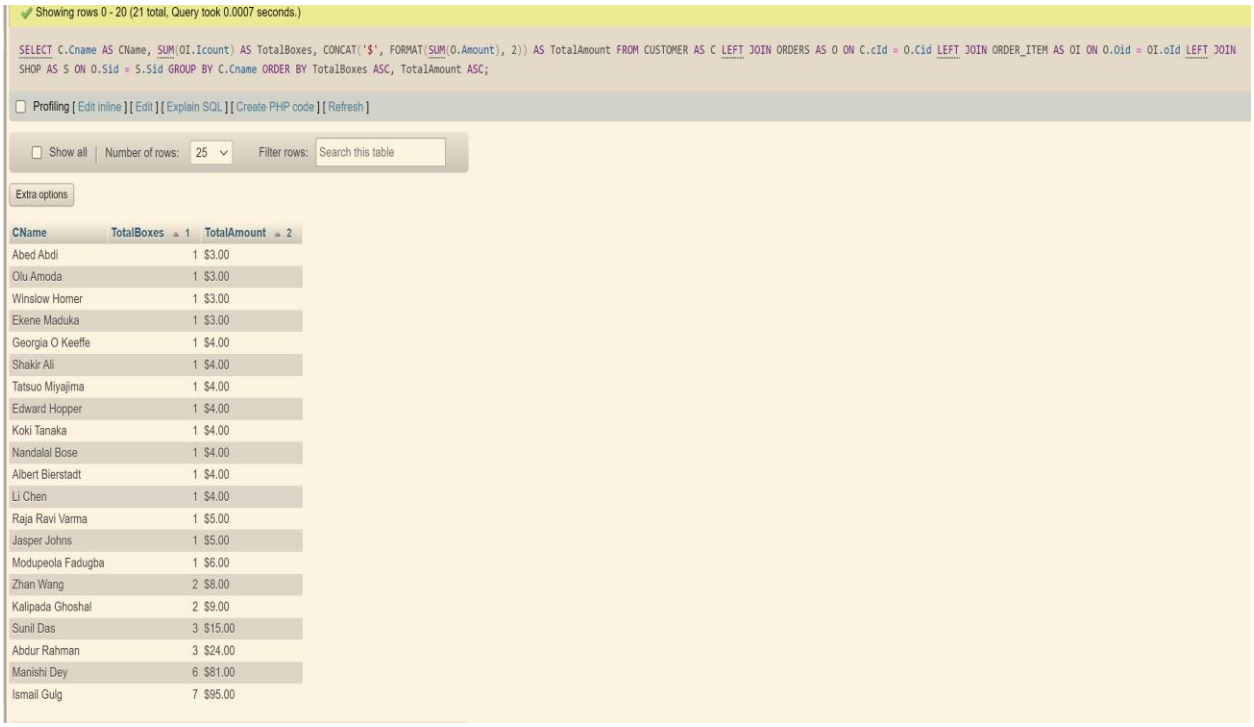
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

CustomerName	TotalAmountSpent
Abed Abdi	$3.00
Olu Amoda	$3.00
Winslow Homer	$3.00
Ekene Maduka	$3.00
Georgia O Keeffe	$4.00
Shakir Ali	$4.00
Tatsuo Miyajima	$4.00
Edward Hopper	$4.00
Koki Tanaka	$4.00
Nandalal Bose	$4.00
Albert Bierstadt	$4.00
Li Chen	$4.00
Raja Ravi Varma	$5.00
Jasper Johns	$5.00
Modupeola Fadugba	$6.00
Zhan Wang	$8.00
Kalipada Ghoshal	$9.00
Abdur Rahman	$12.00
Sunil Das	$15.00
Manishi Dey	$27.00
Ismail Gulg	$29.00

10. Customer Order Metrics (Q10)

- Purpose: Combined order analysis
- Returns: Total boxes ordered and amount spent per customer
- Sorting: By number of boxes ordered



Showing rows 0 - 20 (21 total, Query took 0.0007 seconds.)

```
SELECT C.Cname AS CName, SUM(OI.Icount) AS TotalBoxes, CONCAT('$', FORMAT(SUM(O.Amount), 2)) AS TotalAmount FROM CUSTOMER AS C LEFT JOIN ORDERS AS O ON C.cId = O.Cid LEFT JOIN ORDER_ITEM AS OI ON O.Oid = OI.oId LEFT JOIN
SHOP AS S ON O.Sid = S.Sid GROUP BY C.Cname ORDER BY TotalBoxes ASC, TotalAmount ASC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

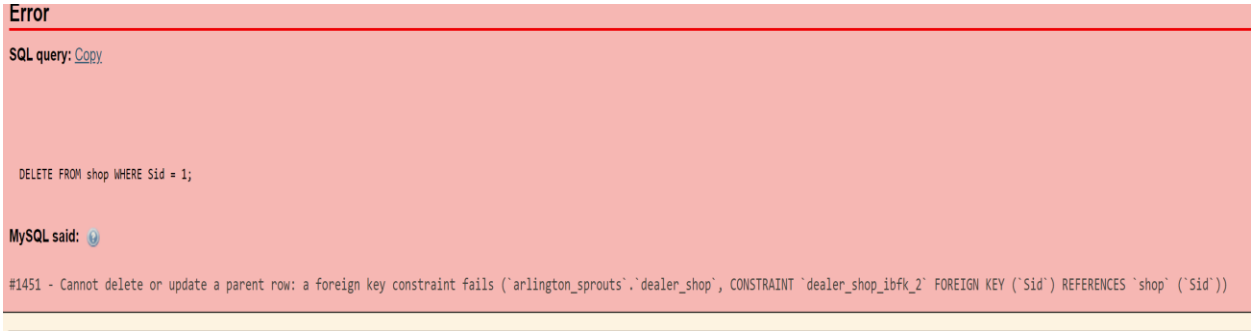
CName	TotalBoxes ▲ 1	TotalAmount ▲ 2
Abed Abdi	1	$3.00
Olu Amoda	1	$3.00
Winslow Homer	1	$3.00
Ekene Maduka	1	$3.00
Georgia O Keeffe	1	$4.00
Shakir Ali	1	$4.00
Tatsuo Miyajima	1	$4.00
Edward Hopper	1	$4.00
Koki Tanaka	1	$4.00
Nandalal Bose	1	$4.00
Albert Bierstadt	1	$4.00
Li Chen	1	$4.00
Raja Ravi Varma	1	$5.00
Jasper Johns	1	$5.00
Modupeola Fadugba	1	$6.00
Zhan Wang	2	$8.00
Kalipada Ghoshal	2	$9.00
Sunil Das	3	$15.00
Abdur Rahman	3	$24.00
Manishi Dey	6	$81.00
Ismail Gulg	7	$95.00

Error Handling Cases

Database Integrity Errors

11. Foreign Key Constraint Error (Q11)

- Scenario: Attempting to delete shop record
- Error: Cannot delete due to existing dealer_shop references
- Reason: Foreign key constraint violation



Error

SQL query: Copy

```
DELETE FROM shop WHERE Sid = 1;
```

MySQL said:

#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`arlington_sprouts`.`dealer_shop`, CONSTRAINT `dealer_shop_ibfk_2` FOREIGN KEY (`Sid`) REFERENCES `shop` (`Sid`))

12. Primary Key Violation (Q12)

- Scenario: Shop insertion with duplicate ID
- Error: Cannot insert duplicate primary key
- Affected Table: shop table

Error

SQL query: Copy

```
INSERT into shop VALUES
(1,'Shop1','Dwight','Dallas','TX','75211-1111','2024-12-20','4696056710','www.ok.com');
```

MySQL said:

```
#1062 - Duplicate entry '1' for key 'PRIMARY'
```

Error

SQL query: Copy

```
INSERT into item VALUES
(NULL,'CESAR',35);
```

MySQL said:

```
#1048 - Column 'Iid' cannot be null
```

Error

SQL query: Copy

```
INSERT into dealer_shop VALUES
(3,4);
```

MySQL said:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`arlington_sprouts`.`dealer_shop`, CONSTRAINT `dealer_shop_ibfk_2` FOREIGN KEY (`Sid`) REFERENCES `shop` (`Sid`))
```

13. Foreign Key Reference Error (Q13)

- Scenario: Updating shop_customer record
- Error: Invalid shop ID reference
- Reason: Foreign key constraint violation

Notes

- All queries are designed to maintain data integrity
- Error cases demonstrate proper database constraint enforcement
- Queries focus on business analytics and operational monitoring