Chapter 42: Functions (Aggregate)

Section 42.1: Conditional aggregation

Create a table named Orders:

```
CREATE TABLE Orders
(OrderID INT PRIMARY KEY, CustomerID INT, OrderDate DATE)
```

Insert some data into the Orders table:

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate)
VALUES
(1, 101, '2023-01-01'),
(2, 102, '2023-01-02'),
(3, 103, '2023-01-03')
```

Section 42.2: Subqueries

Subqueries can be used to return data that is used in the FROM clause:

```
SELECT * FROM Orders
WHERE CustomerID IN
(SELECT CustomerID FROM Customers
WHERE Age > 30)
```

Section 42.3: List Concatenation

Create a table named Products:

```
CREATE TABLE Products
(ProductID INT PRIMARY KEY, ProductName VARCHAR(50), Category VARCHAR(50))
```

Insert some data into the Products table:

```
INSERT INTO Products (ProductID, ProductName, Category)
VALUES
(1, 'Product A', 'Electronics'),
(2, 'Product B', 'Clothing'),
(3, 'Product C', 'Home')
```

Use a subquery to list the product names:

```
SELECT ProductName FROM Products
WHERE Category = 'Clothing'
```

Chapter 42: Functions (Aggregate)

Section 42.1: Conditional aggregation

Create a table named Payment:

```
CREATE TABLE Payment
(PaymentID INT PRIMARY KEY, PaymentType VARCHAR(50), Amount DECIMAL(10, 2))
```

Insert some data into the Payment table:

```
INSERT INTO Payment (PaymentID, PaymentType, Amount)
VALUES
(1, 'Credit', 150.00),
(2, 'Debit', 200.00),
(3, 'Cash', 50.00)
```

Use a conditional aggregate function to calculate the total amount for each payment type:

```
SELECT PaymentType, SUM(Amount), COUNT(*) FROM Payment
GROUP BY PaymentType
```

Chapter 52: Subqueries

Section 52.1: Subquery in FROM clause

A subquery in the FROM clause is used to filter the data before it is used in the main query:

```
SELECT * FROM Employees
WHERE Department IN
(SELECT Department FROM Departments
WHERE Location = 'New York')
```

Section 52.2: Subquery in SELECT clause

Use a subquery in the SELECT clause to filter the data based on a condition:

```
SELECT EmployeeName FROM Employees
WHERE Department = (SELECT Department FROM Departments
WHERE Location = 'New York')
```

Section 52.3: Subquery in WHERE clause

Use a subquery in the WHERE clause to filter the data based on a condition:

```
SELECT * FROM Employees
WHERE Department IN
(SELECT Department FROM Departments
WHERE Location = 'New York')
```

Section 52.4: Correlated Subqueries

Subqueries that return a result set that is correlated to the main query are called correlated subqueries:

```
SELECT EmployeeName FROM Employees
WHERE Department = (SELECT Department FROM Departments
WHERE Location = 'New York')
```

Section 52.5: Filter query results using query on different table

This query filters all employees who are from the same state as the headquarters:

```
SELECT * FROM Employees
WHERE State = (SELECT State FROM Headquarters)
```
# Contents

**About**

<table>
<thead>
<tr>
<th>Chapter 1: Getting started with SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 1.1: Overview</td>
</tr>
</tbody>
</table>

**Chapter 2: Identifier**

| Section 2.1: Unquoted Identifiers |

**Chapter 3: Data Types**

<table>
<thead>
<tr>
<th>Section 3.1: DECIMAL and NUMERIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 3.2: FLOAT and REAL</td>
</tr>
<tr>
<td>Section 3.3: Integers</td>
</tr>
<tr>
<td>Section 3.4: MONEY and SMALLMONEY</td>
</tr>
<tr>
<td>Section 3.5: BINARY and VARBINARY</td>
</tr>
<tr>
<td>Section 3.6: CHAR and VARCHAR</td>
</tr>
<tr>
<td>Section 3.7: NCHAR and NVARCHAR</td>
</tr>
<tr>
<td>Section 3.8: UNIQUEIDENTIFIER</td>
</tr>
</tbody>
</table>

**Chapter 4: NULL**

<table>
<thead>
<tr>
<th>Section 4.1: Filtering for NULL in queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 4.2: Nullable columns in tables</td>
</tr>
<tr>
<td>Section 4.3: Updating fields to NULL</td>
</tr>
<tr>
<td>Section 4.4: Inserting rows with NULL fields</td>
</tr>
</tbody>
</table>

**Chapter 5: Example Databases and Tables**

<table>
<thead>
<tr>
<th>Section 5.1: Auto Shop Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 5.2: Library Database</td>
</tr>
<tr>
<td>Section 5.3: Countries Table</td>
</tr>
</tbody>
</table>

**Chapter 6: SELECT**

<table>
<thead>
<tr>
<th>Section 6.1: Using the wildcard character to select all columns in a query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 6.2: SELECT Using Column Aliases</td>
</tr>
<tr>
<td>Section 6.3: Select Individual Columns</td>
</tr>
<tr>
<td>Section 6.4: Selecting specified number of records</td>
</tr>
<tr>
<td>Section 6.5: Selecting with Condition</td>
</tr>
<tr>
<td>Section 6.6: Selecting with CASE</td>
</tr>
<tr>
<td>Section 6.7: Select columns which are named after reserved keywords</td>
</tr>
<tr>
<td>Section 6.8: Selecting with table alias</td>
</tr>
<tr>
<td>Section 6.9: Selecting with more than 1 condition</td>
</tr>
<tr>
<td>Section 6.10: Selecting without Locking the table</td>
</tr>
<tr>
<td>Section 6.11: Selecting with Aggregate functions</td>
</tr>
<tr>
<td>Section 6.12: Select with condition of multiple values from column</td>
</tr>
<tr>
<td>Section 6.13: Get aggregated result for row groups</td>
</tr>
<tr>
<td>Section 6.14: Selection with sorted Results</td>
</tr>
<tr>
<td>Section 6.15: Selecting with null</td>
</tr>
<tr>
<td>Section 6.16: Select distinct (unique values only)</td>
</tr>
<tr>
<td>Section 6.17: Select rows from multiple tables</td>
</tr>
</tbody>
</table>

**Chapter 7: GROUP BY**

<table>
<thead>
<tr>
<th>Section 7.1: Basic GROUP BY example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 7.2: Filter GROUP BY results using a HAVING clause</td>
</tr>
<tr>
<td>Section 7.3: USE GROUP BY to COUNT the number of rows for each unique entry in a given column</td>
</tr>
<tr>
<td>Section 7.4: ROLAP aggregation (Data Mining)</td>
</tr>
</tbody>
</table>
End of ebook preview
Download the full PDF tutorial from the link below:

Click Here