

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

## Learner Notes and Code Examples

```
--Exploring all the tables

SELECT * FROM Customers;

SELECT * FROM Categories;

SELECT * FROM Employees;

SELECT * FROM OrderDetails;

SELECT * FROM Orders;

SELECT * FROM Products;

SELECT * FROM Shippers;

SELECT * FROM Suppliers;


-- We need to award bonus to employees who had top 5 order amounts


-- If we do not consider units of products purchased, the following
question answers the question "Which employees sold highest price product"


SELECT E.FirstName, E.LastName, P.Price

FROM Employees AS E

INNER JOIN Orders AS O ON

E.EmployeeID = O.EmployeeID

INNER JOIN OrderDetails AS OD ON

O.OrderID = OD.OrderID

INNER JOIN Products AS P ON
```

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

```
OD.ProductID = P.ProductID
```

```
ORDER BY P.Price DESC
```

```
LIMIT 5;
```

```
-- Output:->
```

```
-- Number of Records: 5
```

```
-- FirstName      LastName      Price
```

```
-- Margaret      Peacock      263.5
```

```
-- Nancy         Davolio      263.5
```

```
-- Robert        King         263.5
```

```
-- Margaret      Peacock      263.5
```

```
-- Steven        Buchanan     263.5
```

```
-- To answer the original question we need to calculate the total amount  
an Employee was able to sell,
```

```
--it could be done by multiply quantity by product price and adding all  
products sold by an employee
```

```
SELECT E.FirstName, E.LastName, SUM((P.Price * OD.Quantity)) AS  
TotalAmount
```

```
FROM Employees AS E
```

```
INNER JOIN Orders AS O ON
```

```
E.EmployeeID = O.EmployeeID
```

```
INNER JOIN OrderDetails AS OD ON
```

```
O.OrderID = OD.OrderID
```

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

```
INNER JOIN Products AS P ON
```

```
OD.ProductID = P.ProductID
```

```
GROUP BY O.OrderID
```

```
ORDER BY TotalAmount DESC
```

```
LIMIT 5;
```

```
-- Output:->
```

```
-- Number of Records: 5
```

```
-- FirstName      LastName      TotalAmount
```

```
-- Steven         Buchanan    15353.6
```

```
-- Robert         King       14366.5
```

```
-- Margaret       Peacock    14104
```

```
-- Robert         King       13427
```

```
-- Margaret       Peacock    9244.250000000002
```

```
-- In this result, 2 of the employees are repeated as they had more than one great sales.
```

```
-- But we need to select top 5 distinct employees to award incentives.
```

```
-- The guide suggests to include a having clause which consists of ID's of top performers.
```

```
-- But in real time we'll need to calculate that for ourselves, so I did my own version of the answer.
```

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

```
-- Guide:->

SELECT E.FirstName, E.LastName, O.OrderID, SUM((P.Price * OD.Quantity)) AS
TotalAmount

FROM Employees AS E

INNER JOIN Orders AS O ON

E.EmployeeID = O.EmployeeID

INNER JOIN OrderDetails AS OD ON

O.OrderID = OD.OrderID

INNER JOIN Products AS P ON

OD.ProductID = P.ProductID

GROUP BY O.OrderID

HAVING O.OrderID IN (10372, 10424, 10417, 10324, 10351)

ORDER BY TotalAmount DESC

LIMIT 5;

-- My Version:->

SELECT E.FirstName, E.LastName, O.OrderID, SUM((P.Price * OD.Quantity)) AS
TotalAmount

FROM Employees AS E

INNER JOIN Orders AS O ON

E.EmployeeID = O.EmployeeID

INNER JOIN OrderDetails AS OD ON

O.OrderID = OD.OrderID

INNER JOIN Products AS P ON
```

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

```
OD.ProductID = P.ProductID

GROUP BY O.OrderID

HAVING O.OrderID NOT IN (10360, 10353, 10440, 10430)

ORDER BY TotalAmount DESC

LIMIT 5;

-- Since we won't know which IDs have highest sales amount and we not able
to use DISTINCT clause on the W3Schools editor

-- I made sure that the other than highest of the employee no other entry
comes in the top 5

-- Output:->

-- FirstName      LastName      OrderID      TotalAmount
-- Steven          Buchanan      10372         15353.6
-- Robert          King          10424         14366.5
-- Margaret        Peacock       10417         14104
-- Anne            Dodsworth    10324         7698.45
-- Nancy           Davolio       10351         7103.599999999999
```

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

## Key Takeaways

### Task 1

#### Title: Project and Database Introduction

- A database is a collection of related tables.
- Data is stored in tables and is organized into rows and columns.
- The W3Schools SQL Tryit Editor provides a place to write and run SQL code.

### Task 2

#### Title: A Look at the Question and the Suggested Solution

- The data analyst strives to use data to answer business questions.
- A business problem is best solved by examining the data available and visualizing which pieces of data should make up the solution.
- Data from multiple tables may be required to solve a business problem.

### Task 3

#### Title: Joining Tables Together in SQL to Obtain Data for Analysis

- To join two tables, they must share a common column.
- The SQL INNER JOIN command returns only rows that match between two tables.
- An SQL query can be keyed into the SQL Tryit editor without regard to case; however, correct spelling and punctuation is critical

### Task 4

#### Title: Calculate and Summarize Sales for each Order

- New, temporary fields can be created as a result of a calculation in SQL.
- Aggregating or grouping data can make it more useful for decision making.

# Award Sales-incentive Bonuses Using the W3Schools SQL Tool

- In SQL code, the SUM() function, together with the GROUP BY clause, can be used to aggregate data.

## Task 5

### Title: Display the Solution

- Limiting the number of rows that display as the result of an SQL query can be accomplished using the LIMIT command.
- In SQL code, the HAVING command applies a filter after aggregation.
- A data analyst often strives to anticipate alternative types of data that may address a business problem.

### Sample SQL Code:

```
SELECT LastName, FirstName, Orders.OrderID, sum(Quantity * Price) as SalesAmt
FROM employees
  inner join orders
    on employees.employeeID = orders.employeeid
  inner join orderDetails
    on orders.orderid = orderdetails.orderid
  inner join products
    on orderdetails.productid = products.productid
group by orders.orderid
having orders.orderid in (10372, 10424, 10417, 10324, 10351)
order by salesamt desc
```