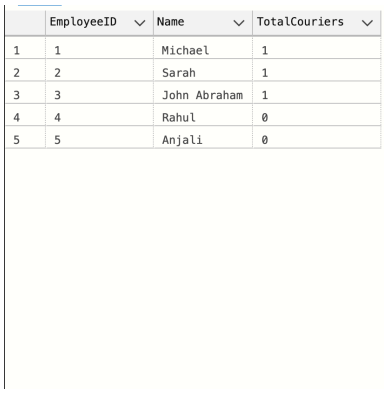
```
--TASK3 GroupBy, Aggregate Functions, Having, Order By, where

OUTPUT:

--14. Find the total number of couriers handled by each employee.

SELECT E.EmployeeID, E.Name, COUNT(C.CourierID) AS TotalCouriers
FROM Employee E
LEFT JOIN Courier C ON E.EmployeeID = C.AssignedEmployeeID
GROUP BY E.EmployeeID, E.Name
ORDER BY TotalCouriers DESC;
```



	EmployeeID	Name	TotalCouriers
1	1	Michael	1
2	2	Sarah	1
3	3	John Abraham	1
4	4	Rahul	0
5	5	Anjali	0

```
--15. Calculate the total revenue generated by each location

SELECT L.LocationID, L.LocationName, SUM(P.Amount) AS TotalRevenue
FROM Payment P
JOIN Location L ON P.LocationID = L.LocationID
GROUP BY L.LocationID, L.LocationName
ORDER BY TotalRevenue;
```

Results Messages

	LocationID	LocationName	TotalRevenue
1	1	Chennai Hub	100.00
2	2	Bangalore Hub	250.00
3	3	Mumbai Hub	500.00

```sql
--16. Find the total number of couriers delivered to each
location.

SELECT L.LocationID,
       L.LocationName,
       COUNT(C.CourierID) AS TotalDelivered
FROM Location L
LEFT JOIN Courier C ON C.DeliveryLocationID = L.LocationID
AND C.Status = 'Delivered'
GROUP BY L.LocationID, L.LocationName
ORDER BY TotalDelivered DESC;
```

	LocationID	LocationName	TotalDelivered
1	3	Mumbai Hub	1
2	4	Delhi Hub	0
3	1	Chennai Hub	0
4	2	Bangalore Hub	0

```sql
--17. Find the courier with the highest average delivery time:


SELECT TOP 1 C.CourierID, C.TrackingNumber, AVG(DATEDIFF(DAY,
C.SentDate, C.DeliveryDate)) AS AvgDeliveryTime
FROM Courier C
WHERE C.DeliveryDate IS NOT NULL
GROUP BY C.CourierID, C.TrackingNumber
ORDER BY AvgDeliveryTime DESC;
```

	CourierID	TrackingNumber	AvgDeliveryTime
1	2	TRK789123	13

```sql
--18. Find Locations with Total Payments Less Than a Certain
Amount

SELECT L.LocationID, L.LocationName, SUM(P.Amount) AS
TotalPayments
FROM Payment P
JOIN Location L ON P.LocationID = L.LocationID
GROUP BY L.LocationID, L.LocationName
HAVING SUM(P.Amount) < 5000
ORDER BY TotalPayments ASC;
```

	LocationID	LocationName	TotalPayments
1	1	Chennai Hub	100.00
2	2	Bangalore Hub	250.00
3	3	Mumbai Hub	500.00

```sql
--19. Calculate Total Payments per Location

SELECT L.LocationID, L.LocationName, SUM(P.Amount) AS
TotalPayments
FROM Payment P
JOIN Location L ON P.LocationID = L.LocationID
GROUP BY L.LocationID, L.LocationName
ORDER BY TotalPayments DESC;
```

	LocationID	LocationName	TotalPayments
1	3	Mumbai Hub	500.00
2	2	Bangalore Hub	250.00
3	1	Chennai Hub	100.00

```sql
/*20. Retrieve couriers who have received payments totaling more
than $1000 in a specific location
(LocationID = X):*/

SELECT C.CourierID, C.SenderName, SUM(P.Amount) AS TotalPayments
FROM Courier C
JOIN Payment P ON C.CourierID = P.CourierID
WHERE P.LocationID = 1
GROUP BY C.CourierID, C.SenderName
HAVING SUM(P.Amount) > 1000
ORDER BY TotalPayments DESC;
```


```sql
/*21. Retrieve couriers who have received payments totaling more
than $1000 after a certain date
(PaymentDate > 'YYYY-MM-DD'):*/

SELECT C.CourierID, C.SenderName, SUM(P.Amount) AS TotalPayments
FROM Courier C
JOIN Payment P ON C.CourierID = P.CourierID
WHERE P.PaymentDate > '2025-01-01'
GROUP BY C.CourierID, C.SenderName
HAVING SUM(P.Amount) > 1000
ORDER BY TotalPayments DESC;
```

```sql
/*22. Retrieve locations where the total amount received is more
than $5000 before a certain date
(PaymentDate > 'YYYY-MM-DD')*/

SELECT L.LocationID, L.LocationName, SUM(P.Amount) AS
TotalPayments
FROM Payment P
JOIN Location L ON P.LocationID = L.LocationID
WHERE P.PaymentDate < '2025-01-01'
GROUP BY L.LocationID, L.LocationName
HAVING SUM(P.Amount) > 5000
ORDER BY TotalPayments DESC;
```