

Introduction to DBMS

ASSESSMENT

Below are concise, well-formatted statements you can run in any SQL RDBMS (MySQL/Postgres/SQL Server). I include CREATE TABLE and sample INSERTs so you can test, followed by the required query using JOIN. I also show a LEFT JOIN variant (if you want customers that might not have an assigned salesman).

-- 1) Create tables

```
CREATE TABLE Salesman (  
    salesman_id INT PRIMARY KEY,  
    name       VARCHAR(100) NOT NULL,  
    city       VARCHAR(50),  
    commission DECIMAL(6,2)  
);
```

```
CREATE TABLE Customer (  
    customer_id INT PRIMARY KEY,  
    customer_name VARCHAR(100) NOT NULL,  
    city       VARCHAR(50),  
    grade      INT,  
    salesman_id INT,  
    FOREIGN KEY (salesman_id) REFERENCES Salesman(salesman_id)  
);
```

-- 2) Sample data (optional, for testing)

```
INSERT INTO Salesman (salesman_id, name, city, commission) VALUES
(1001, 'Robert', 'New York', 0.10),
(1002, 'Emily', 'Chicago', 0.12),
(1003, 'Amit', 'Mumbai', 0.08);
```

```
INSERT INTO Customer (customer_id, customer_name, city, grade,
salesman_id) VALUES
(1, 'Alpha Corp', 'New York', 2, 1001),
(2, 'Beta Traders', 'Chicago', 3, 1002),
(3, 'Gamma LLC', 'Pune', 1, 1003),
(4, 'Delta Co', 'Bangalore', 2, NULL);
```

-- 3) Required query using INNER JOIN (returns only customers who have a matching salesman)

```
SELECT
    c.customer_name    AS CustomerName,
    c.city             AS CustomerCity,
    s.name             AS Salesman,
    s.commission       AS Commission
FROM
    Customer c
    INNER JOIN Salesman s ON c.salesman_id = s.salesman_id
ORDER BY
    c.customer_name;
```

-- 4) Alternative: LEFT JOIN (returns all customers; Salesman columns NULL if none assigned)

SELECT

c.customer_name AS CustomerName,

c.city AS CustomerCity,

s.name AS Salesman,

s.commission AS Commission

FROM

Customer c

LEFT JOIN Salesman s ON c.salesman_id = s.salesman_id

ORDER BY

c.customer_name;