# Data Modelling

## Entity Relationship Diagram (ERD)

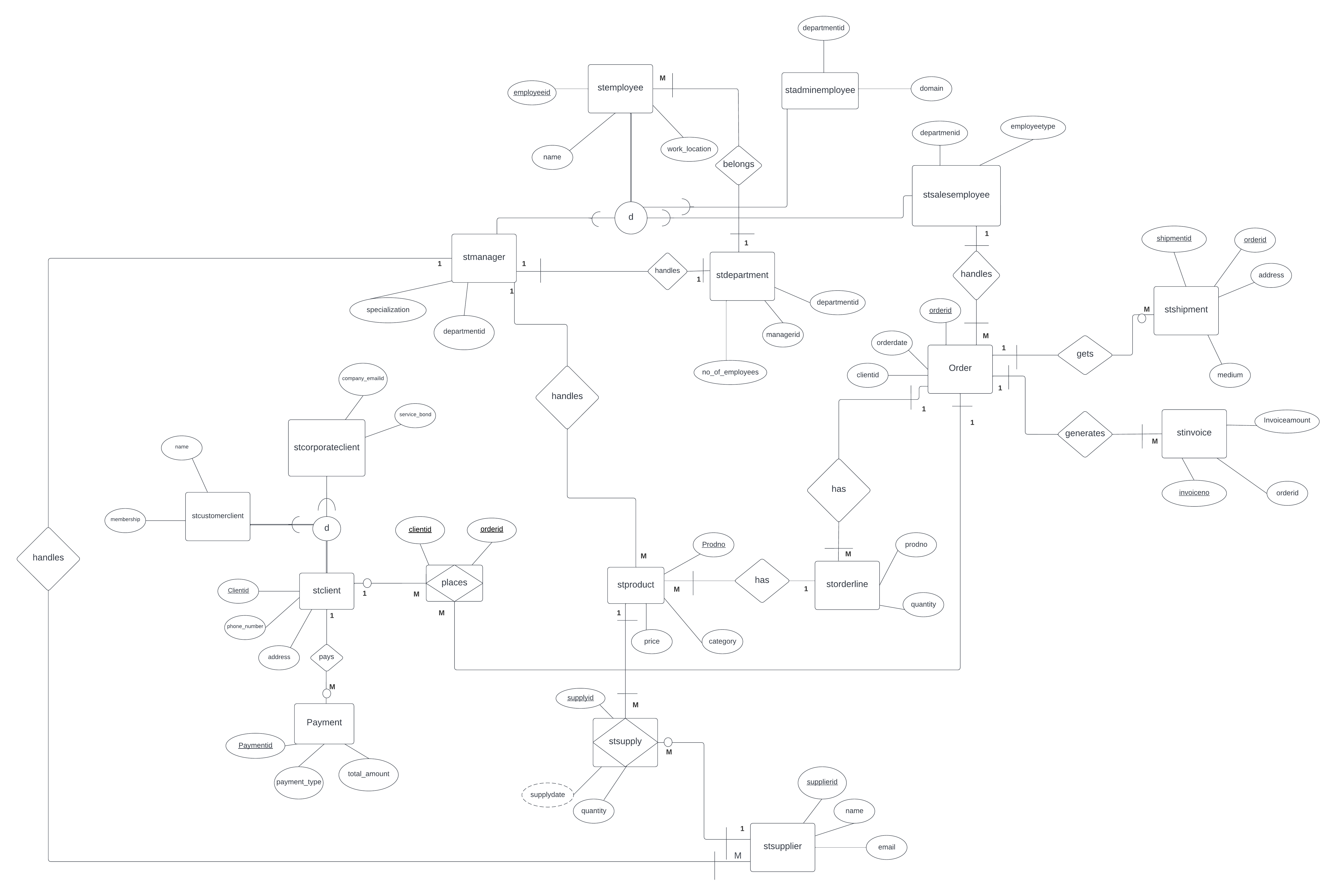


Figure : Entity Relationship diagram

## Business Relational model

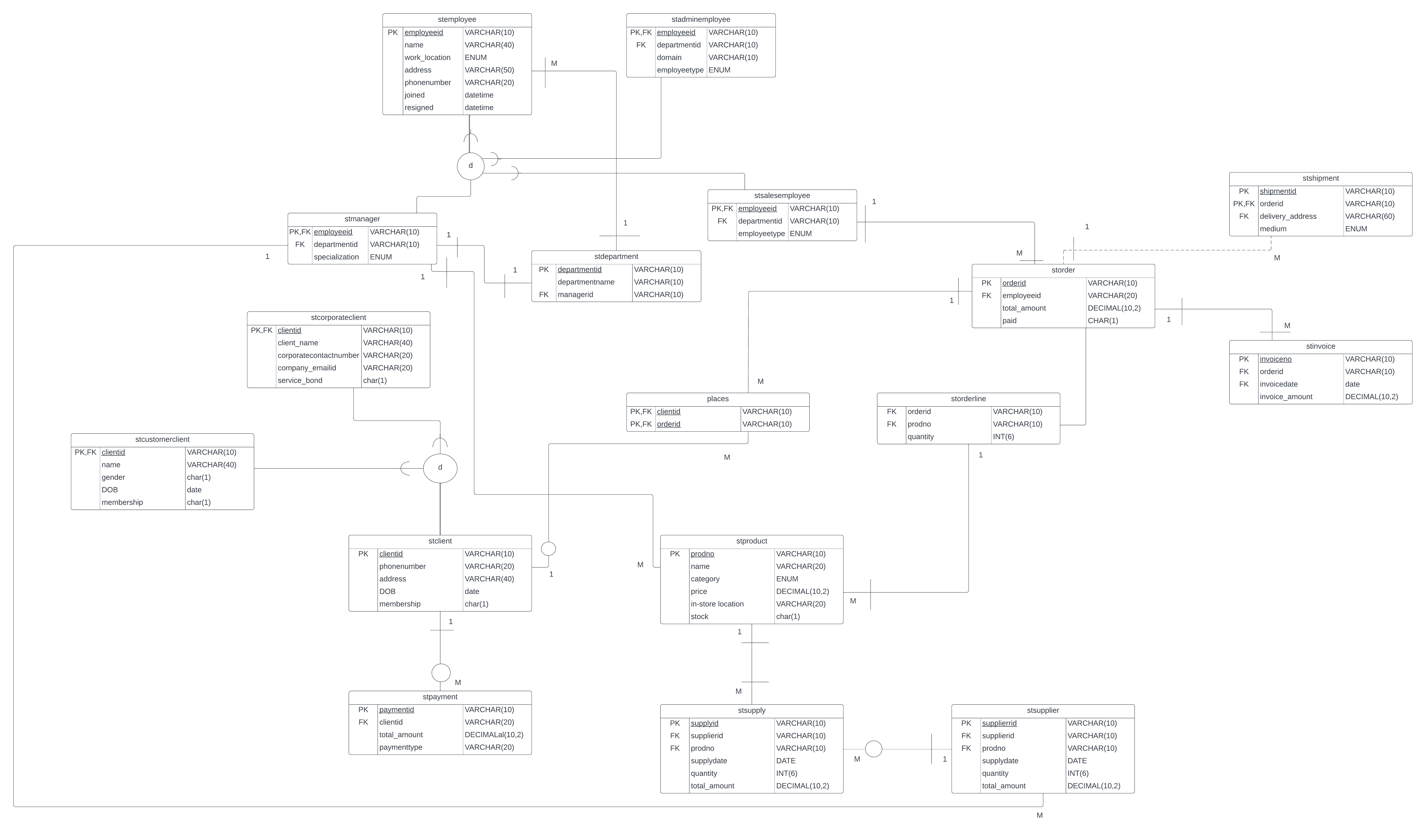


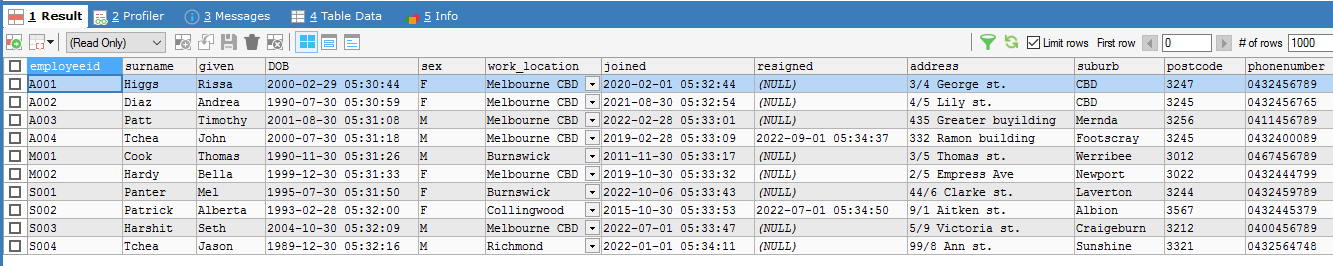
Figure : Business relational model

# 2.0 SQL Scripts

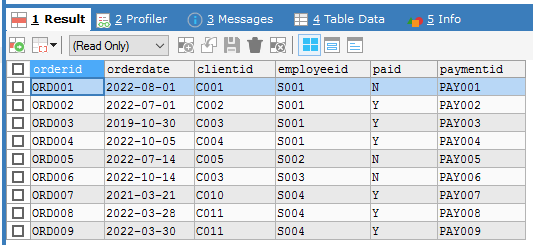
## 2.1 Function to calculate commission for all sales employees based on sales performance

**Details of all the tables involved in the query**

SELECT \* FROM stemployee;



SELECT \* FROM storder;



**Function**

DELIMITER $$

CREATE FUNCTION fnc\_calc\_commission(empid VARCHAR(10))

RETURNS DOUBLE

DETERMINISTIC

BEGIN

DECLARE commission DOUBLE;

DECLARE rph DOUBLE;

DECLARE count\_order INT;

SELECT COUNT(\*)

INTO count\_order

FROM storder

GROUP BY employeeid

HAVING employeeid = empid;

SELECT rateperhour

INTO rph

FROM stemployee

WHERE employeeid = empid;

IF count\_order >= 3 THEN

SET commission = CEIL(0.1 \* rph);

ELSE

SET commission = CEIL(0.07 \* rph);

END IF;

RETURN commission;

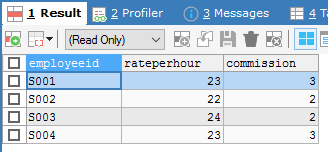
END$$

**Output**

SELECT employeeid, rateperhour, fnc\_calc\_commission(employeeid) AS commission

FROM stemployee

WHERE employeeid LIKE '%S%';



**Design of the SQL statement**

The function has ‘empid’ which is the ‘employee id’ passed as an argument. The first select query generates the number of orders an employee has processed and the second select query provides the rate per hour of that employee. If the count of order is more than or equal to 3, then the employee gets 10 percent commission of the rate per hour or else the commission is 7 percent of the rate per hour. This value is returned through a commission variable which is declared in the function.

**Business Need and business value**

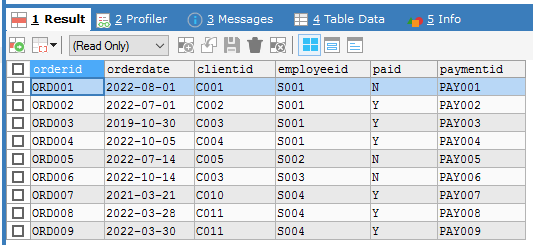
## 

This function enables Stella Pty. Ltd.'s management to give commision allowances to its sales employees. The commision is based on the employee's performance, so the higher the sales performance, the bigger the commission granted. The business need of this function is to encourages employees to perform better in the organisation for personal gain; yet, this benefits the corporation because it boosts sales. Furthermore, Stella Pty. Ltd. can determine the company's expenses in the form of employee commission through this function.

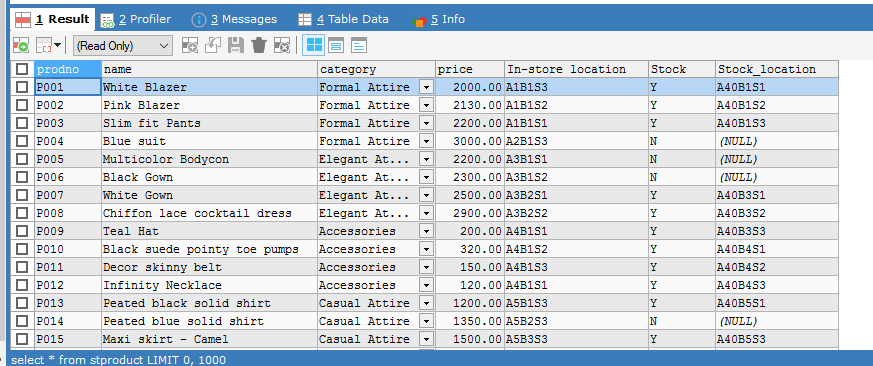
## 2.2 Find the most valuable client by finding the highest order purchase

**Details of all the tables involved in the query**

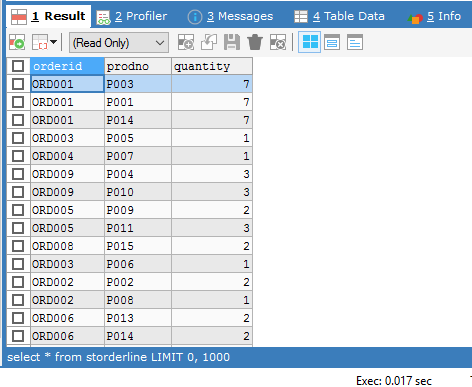
SELECT \* FROM storder;



SELECT \* FROM stproduct;

****

SELECT \* FROM storderline;



**Query**

SELECT stord.clientid, SUM(ordl.quantity \* prod.price) AS total

FROM stproduct AS prod JOIN (storderline ordl, storder stord)

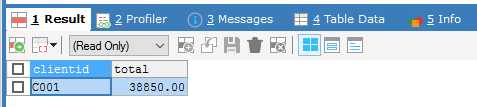
ON (prod.prodno = ordl.prodno AND stord.orderid = ordl.orderid)

GROUP BY stord.clientid

ORDER BY total DESC

LIMIT 1;

**Output**

****

**Design of the SQL statement**

This query uses JOIN to provide output from three tables – stproduct, storder and storderline. Storder table stores the information such as clientid and orderid. For each orderid, the total amount can be calculated by the taking the quantity of the number of products ordered from storderline table and the price of each product stored in the stproduct table. As each client may have multiple order, so GROUP BY has been used to calculate total amount spent by a client. This result is displayed in a descending order of the amount where the highest amount is displayed on the top of the query and LIMIT 1 has been shown to determine the highest spending client.

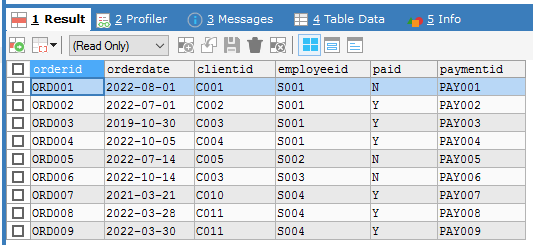
**Business Need**

## Stella Pty. Ltd. can use this query to identify its most valuable client based on the highest money spent by any client. Knowing who the most important client is, enables the organization to follow up with them, expressing gratitude via messaging or offering them discount offers and rewards for being the most valuable client. On the other hand, the client may raise the firm's sales as a result of this follow-up from the company. As a result, this query, assists Stella Pty. Ltd. in increasing sales by following up with the most valuable client. Furthermore, it can assist the organization in determining the most money spent by any client.

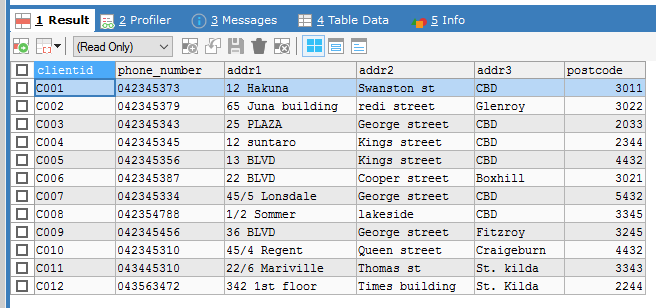
## 2.3 Find details of all the clients who have not paid the order amount

**Details of all the tables involved in the query**

SELECT \* FROM storder;



SELECT \* FROM stclient;



**Query**

SELECT \*

FROM stclient

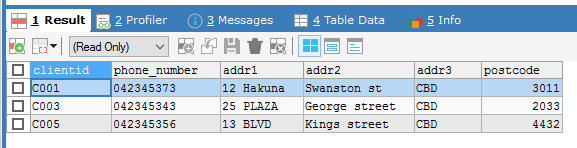
WHERE 1 <= (SELECT COUNT(\*)

FROM storder

WHERE paid = 'N' AND stclient.clientid = storder.clientid

GROUP BY clientid);

**Output**

****

**Design of the SQL statement**

This query employs nested queries to identify clients who have not paid the order amount by utilising the COUNT function and the GROUP BY in the 'client id'. The check occurs just once for a specific clientid since the storder table relates to the stclient table with the common column 'client id.' This streamlines the SQL query to produce results only for the referred 'client id.' The output of the inner query was used in the where condition to find the client whose value is greater than or equal to 1.

**Business Need**

Stella Pty. Ltd. can use this query to identify all clients from whom the order amount has yet to be paid. Because the company has various expenses, such as supplier payments, employee salaries, and so on, it is critical to know all of the clients from whom no income has been generated. This query returns the information for such a client, to whom the company can send a payment reminder. This query serves a business purpose by identifying all debtors clients of Stella Pty. Ltd.