|  |  |
| --- | --- |
|  | 2814ICT – Data Management  7003ICT – Database Design  School of Information & Communication Technology  Trimester 1, 2021  **Assignment Part 2:**  **Implementing a Database for BigM** |

**ASSIGNMENT TITLE: Analysing ERDs and Writing corresponding SQL statements for BigM’s Database.**

|  |  |  |
| --- | --- | --- |
| **Group ID** | **011** | |
| **Student 1** | **s-number: s5196431** | **Full name: Duwon Ha** |
| **Student 2** | **s-number: s5203114** | **Full name: Kavya Krishnakumar** |
| **Student 3** | **s-number: s5213262** | **Full name: Shinzo Tanimoto** |
| **Course Code: 2814ICT** | | **Workshop/Lab day & time: Friday 9:00am – 10:45am** |
| **Tutor’s name: Emon Kumar Dey** | | **Date submitted: 25/05/2021** |

Contents

[**List of Illustrations** 2](#_Toc72872190)

[**Acknowledgements** 2](#_Toc72872191)

[**Task 1: Creating SQL Tables** 3](#_Toc72872192)

[**Task 2: Inserting Records** 5](#_Toc72872193)

[**Task 3: SQL Queries** 7](#_Toc72872194)

[**Task 4: Inserting Additional Data** 12](#_Toc72872195)

# 

# **List of Illustrations**

|  |  |
| --- | --- |
| **Illustration Name** | **Page Number** |
| Table 1 | 7 |
| Table 2 | 7 |
| Table 3 | 8 |
| Table 4 | 8 |
| Table 5 | 8 |
| Table 6 | 9 |
| Table 7 | 9 |
| Table 8 | 10 |
| Table 9 | 10 |
| Table 10 | 11 |

**Acknowledgements**

1. **Emon Kumar Dey**

# **Task 1: Creating SQL Tables**

CREATE DATABASE IF NOT EXISTS BigM\_s5213262;

USE bigm\_s5213262;

SHOW TABLES;

CREATE TABLE IF NOT EXISTS CUSTOMER(

Cust\_Number INT PRIMARY KEY AUTO\_INCREMENT,

Cust\_Fname VARCHAR(30),

Cust\_Lname VARCHAR(30),

Cust\_Phone CHAR(10)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS PRODUCT(

Prod\_Num INT PRIMARY KEY AUTO\_INCREMENT,

Prod\_Desc VARCHAR(30),

Prod\_Size VARCHAR(30),

Prod\_Price DECIMAL

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS DEPARTMENT(

Dept\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Dept\_Name VARCHAR(30)

) ENGINE=InnoDB;

CREATE TABLE  IF NOT EXISTS EMPLOYEE(

Emp\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Emp\_FName   VARCHAR(30),

Emp\_LName   VARCHAR(30),

Emp\_Phone   INTEGER,

Emp\_DoB     DATE,

Emp\_StartDate   DATE,

Emp\_TaxFNum   INTEGER,

Emp\_HourlySalary DECIMAL(10,2),

StrDept\_ID INT,

SupervisorID INT

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS STORE(

Str\_Num INT PRIMARY KEY AUTO\_INCREMENT,

Str\_Name VARCHAR(50),

Str\_Phone CHAR(10),

Str\_Fax CHAR(10),

Str\_Email VARCHAR(40),

StoreManagerID INT,

SupStore\_Num INT,

FOREIGN KEY (StoreManagerID) REFERENCES EMPLOYEE(Emp\_ID),

FOREIGN KEY (SupStore\_Num) REFERENCES STORE(Str\_Num)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS CUSTOMERORDER(

CustOrd\_ID INT PRIMARY KEY AUTO\_INCREMENT,

CustOrd\_Date DATE,

Cust\_Number INT NOT NULL,

Str\_Num INT NOT NULL,

FOREIGN KEY (Cust\_Number) REFERENCES CUSTOMER (Cust\_Number),

FOREIGN KEY (Str\_Num) REFERENCES STORE (Str\_Num)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS ORDERLINE(

CustOrd\_ID INT,

Prod\_Num INT,

OrdLn\_DateArrived DATE,

OrdLn\_DatePicked DATE,

OrdLn\_Qnty INT,

PRIMARY KEY (CustOrd\_ID, Prod\_Num),

FOREIGN KEY (CustOrd\_ID) REFERENCES CUSTOMERORDER(CustOrd\_ID),

FOREIGN KEY (Prod\_Num) REFERENCES PRODUCT (Prod\_Num)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS STOREDEPARTMENT(

StrDept\_ID INT PRIMARY KEY AUTO\_INCREMENT,

StrDept\_Phone CHAR(10),

StrDept\_Email VARCHAR(40),

DeptSupervisorID INT,

Str\_Num INT,

Dept\_ID INT,

FOREIGN KEY (DeptSupervisorID) REFERENCES EMPLOYEE(Emp\_ID),

FOREIGN KEY (Str\_Num) REFERENCES STORE(Str\_Num),

FOREIGN KEY (Dept\_ID) REFERENCES DEPARTMENT(Dept\_ID)

) ENGINE=InnoDB;

ALTER TABLE EMPLOYEE

ADD FOREIGN KEY (StrDept\_ID) REFERENCES STOREDEPARTMENT (StrDept\_ID);

ALTER TABLE EMPLOYEE

ADD FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (Emp\_ID);

CREATE TABLE IF NOT EXISTS PAYSLIP(

Pay\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Pay\_date DATE,

Pay\_num\_of\_hours INT NOT NULL,

Pay\_amount\_gross DOUBLE NOT NULL,

Emp\_ID INT NOT NULL,

Str\_Num INT NOT NULL,

FOREIGN KEY (Emp\_ID) REFERENCES EMPLOYEE (Emp\_ID),

FOREIGN KEY (Str\_Num) REFERENCES STORE (Str\_Num)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS INVENTORY(

ProductNum INT,

Str\_Num INT,

Inv\_QntyOnHand INT NOT NULL,

Inv\_QtyOrdered INT NOT NULL,

PRIMARY KEY (ProductNum, Str\_Num),

FOREIGN KEY (ProductNum) REFERENCES PRODUCT (Prod\_Num),

FOREIGN KEY (Str\_Num) REFERENCES STORE (Str\_Num)

) ENGINE=InnoDB;

# **Task 2: Inserting Records**

USE bigm\_s5213262;

INSERT INTO CUSTOMER VALUES

(NULL, 'Buffy', 'Winters', '0412345678'),

(NULL, 'Duwon', 'Ha', '0455803205'),

(NULL, 'Kavya', 'Krishnakumar', '0489789774'),

(NULL, 'Jack', 'Tomlinson', '0484787854'),

(NULL, 'Shinzo', 'Tanimoto', '0435607767'),

(NULL, 'Jone', 'Tyler', '0445679873'),

(NULL, 'Princess', 'Ha', '0413247477'),

(NULL, 'David', 'Bieber', '0412522377');

INSERT INTO PRODUCT VALUES

(NULL, 'Gardening Scissors','Medium', 10),

(NULL, 'Shirt', 'Medium', 30),

(NULL, 'Kitchen Knives Set Deluxe', 'Medium', 120),

(NULL, 'Basketball Regulation Size', 'Small', 350),

(NULL, 'Nickson James Hits', 'Small', 50),

(NULL, '100% Polyester sleepwear', 'Large', 120),

(NULL, 'Jordan Shoes', 'Small', 700);

INSERT INTO DEPARTMENT VALUES

(NULL, 'Sports'),

(NULL, 'Clothes'),

(NULL, 'Kitchen'),

(NULL, 'Homeware'),

(NULL, 'Garden');

INSERT INTO EMPLOYEE(Emp\_ID, Emp\_FName, Emp\_LName, Emp\_Phone, Emp\_DoB, Emp\_StartDate, Emp\_TaxFNum, Emp\_HourlySalary) VALUES

(NULL, 'Koupa', 'Taylor', '0321224224', '2002-03-12', '2020-05-02', 12345678, 30),

(NULL, 'Ben', 'Flint', '0434551234', '1999-06-12', '2020-05-02', 15125677, 60),

(NULL, 'Harry', 'Robin', '0449879343', '2002-07-07', '2020-09-12', 98765432, 30),

(NULL, 'Elizabeth', 'Then', '0421252322', '1998-03-23', '2021-03-02', 9823124, 30),

(NULL, 'Jordan', 'Nickson', '0414279846', '1989-08-01', '2010-04-20', 09872525, 40),

(NULL, 'Jason', 'Micheal', '0412340987', '1997-05-04', '2019-12-12', 102938475, 30);

INSERT INTO STORE(Str\_Num, Str\_Name, Str\_Phone, Str\_Fax, Str\_Email, StoreManagerID) VALUES

(NULL, 'Elizabeth-Street-Branch', '0312452524', '0145252434', 'elizabeth@bigm.au', 3),

(NULL, 'Adelaide-Street-Branch', '0124352321', '291232423', 'adelaide@bigm.au', 1),

(NULL, 'Southbank-Branch', '023674845', '026882683', 'southbank@bigm.au', 4),

(NULL, 'Surfers-Paradise-Branch', '0123452352', '2948274817', 'surfers.paradise@bigm.au', 5),

(NULL, 'Mount-Gravatt-Branch', '0937192322', '2982142351', 'mountGravatt@bigm.au', 6);

INSERT INTO CUSTOMERORDER VALUES

(NULL, '2021-05-03', 1, 1),

(NULL, '2021-05-07', 2, 2),

(NULL, '2021-05-22', 3, 3),

(NULL, '2021-05-12', 4, 4),

(NULL, '2021-05-25', 5, 5),

(NULL, '2021-05-30', 6, 3),

(NULL, '2021-05-22', 7, 4);

INSERT INTO ORDERLINE VALUES

(1, 1, '2017-06-03', '2017-06-04', 3),

(2, 2, '2021-06-07', '2021-06-09', 2),

(3, 3, '2015-06-22', '2015-06-23', 2),

(4, 4, '2021-06-12', '2021-06-13', 1),

(5, 5, '2014-06-17', '2014-06-20', 3);

INSERT INTO STOREDEPARTMENT VALUES

(NULL, '1234567891', 'elizabeth.kitchen@bigm.au', 3, 1, 3),

(NULL, '1209384756', 'adelaide.sports@bigm.au', 5, 2, 1),

(NULL, '1252156215', 'southbank.kitchen@bigm.au', 4, 3, 3),

(NULL, '9830484921', 'surfers.paradice.homewear@bigm.au', 1, 4, 4),

(NULL, '8273182931', 'mountGravatt.garden@bigm.au', 6, 5, 5),

(NULL, '2910242312', 'elizabeth.garden@bigm.au', 3, 1, 5),

(NULL, '2923144212', 'adelaide.garden@bigm.au', 2, 2, 5);

UPDATE EMPLOYEE SET StrDept\_ID = 4 WHERE Emp\_ID = 1;

UPDATE EMPLOYEE SET StrDept\_ID = 7 WHERE Emp\_ID = 2;

UPDATE EMPLOYEE SET StrDept\_ID = 1 WHERE Emp\_ID = 3;

UPDATE EMPLOYEE SET StrDept\_ID = 3 WHERE Emp\_ID = 4;

UPDATE EMPLOYEE SET StrDept\_ID = 2 WHERE Emp\_ID = 5;

UPDATE EMPLOYEE SET StrDept\_ID = 5 WHERE Emp\_ID = 6;

UPDATE STORE SET SupStore\_Num = 2 WHERE Str\_Num = 1;

UPDATE STORE SET SupStore\_Num = 3 WHERE Str\_Num = 2;

UPDATE STORE SET SupStore\_Num = 4 WHERE Str\_Num = 3;

UPDATE STORE SET SupStore\_Num = 5 WHERE Str\_Num = 4;

UPDATE STORE SET SupStore\_Num = 1 WHERE Str\_Num = 5;

INSERT INTO PAYSLIP VALUES

(NULL, '2021-06-30', 80, 2400 ,1,2),

(NULL, '2021-06-30', 72, 4320, 2,2),

(NULL, '2021-06-30', 50, 1500, 3,1),

(NULL, '2021-06-30', 60, 1800, 4,3),

(NULL, '2021-06-30', 80, 3200, 5,4),

(NULL, '2021-06-30', 68, 2040, 6,5);

INSERT INTO INVENTORY VALUES

(2,2,3,4),

(1,3,4,5),

(3,2,1,10),

(2,3,10,2),

(5,4,2,4),

(5,3,2,2),

(6,5,5,2),

(7,2,5,2);

# 

# **Task 3: SQL Queries**

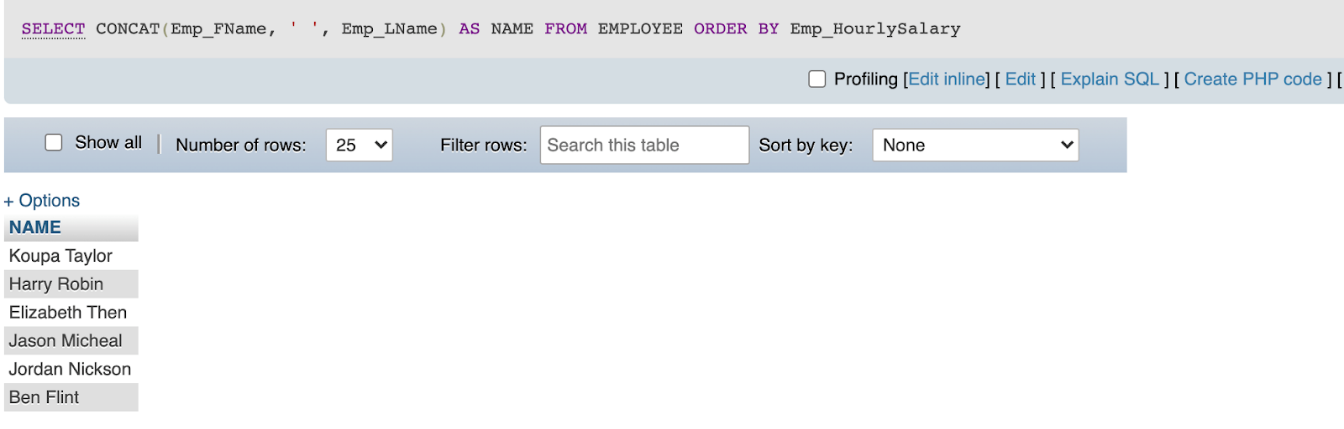
**Query 1:** List of names of all employees sorted by their hourly salary.

SELECT CONCAT(Emp\_FName, ' ', Emp\_LName) AS ‘Name’

FROM EMPLOYEE

ORDER BY Emp\_HourlySalary;

**Output table:**



Table

**Query 2:**  The date on which the most recent customer order has been made. The customer’s name and date of the order will be sufficient.

SELECT CONCAT(C.Cust\_FName, ' ' , C.Cust\_LName) AS 'Name', O.CustOrd\_Date AS 'Date Of The Order'

FROM CUSTOMER AS C, CUSTOMERORDER AS O

WHERE C.Cust\_Number = O.Cust\_Number

Order by CustOrd\_Date DESC

limit 0,1;

**Output table:**

Graphical user interface

Description automatically generated with medium confidence

Table 2

**Query 3:** List of all the store names and their manager names, sorted in dictionary order of the store name.

SELECT S.Str\_Name AS 'Store Name', CONCAT(E.Emp\_FName, ' ', E.Emp\_LName) AS 'Name'

FROM STORE AS S, EMPLOYEE AS E

WHERE S.StoreManagerID = E.Emp\_ID

ORDER BY S.Str\_Name ASC;

**Output table:**

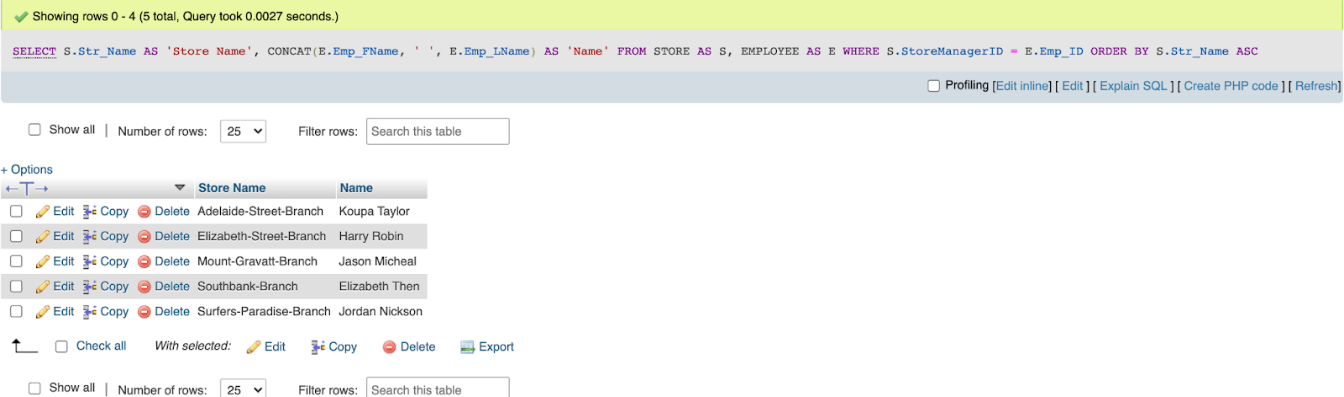


Table 3

**Query 4:** A list of all customers that have not placed an order yet. Displaying customer number and name will be sufficient.

SELECT C.Cust\_Number AS 'Custom Number', CONCAT(C.Cust\_FName, ' ', C.Cust\_LName) AS 'Name'

FROM CUSTOMER AS C

WHERE C.Cust\_Number NOT IN (SELECT Cust\_Number FROM CUSTOMERORDER);

**Output table:**

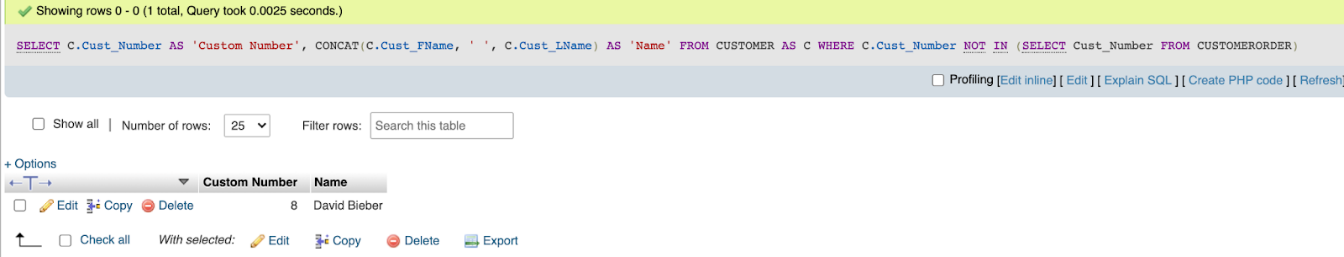


Table 4

**Query 5:**  A list containing the name of employees, who work as supervisors for ‘Sports’ departments in various stores. Show store names and the supervisors of Sports department.

SELECT S.Str\_Name AS 'Store Name', CONCAT(E.Emp\_FName, ' ', E.Emp\_LName) AS 'Supervisor Name'

FROM STORE AS S, EMPLOYEE AS E, STOREDEPARTMENT AS SD

WHERE SD.DeptSupervisorID = E.Emp\_ID

AND SD.Dept\_ID = (SELECT Dept\_ID FROM DEPARTMENT WHERE Dept\_Name = 'Sports')

AND SD.Str\_Num = S.Str\_Num;

**Output table:**



Table 5

**Query 6:** A list containing the total quantity on hand for each product (product number and description) regardless of stores.

SELECT P.Prod\_Num AS 'Product Number', P.Prod\_Desc AS 'Description', SUM(Inv\_QntyOnHand) AS 'Quantity On Hand'

FROM PRODUCT AS P, INVENTORY AS I

WHERE P.Prod\_Num = I.ProductNum

GROUP BY P.Prod\_Num;

**Output table:**

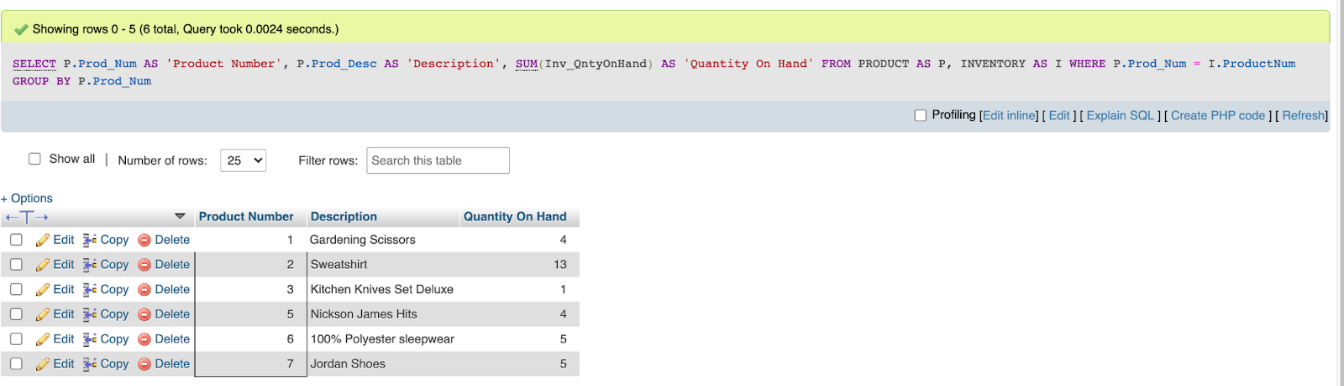


Table 6

**Query 7:** A list showing each product sold (picked) on or before May 20, 2018. Show product number, name and quantity sold, sorted by product number and then quantity sold.

SELECT P.Prod\_Num AS 'Product Number', P.Prod\_Desc AS 'Description',  SUM(O.OrdLn\_Qnty) AS 'Quantity Sold'

FROM ORDERLINE AS O, PRODUCT AS P

WHERE   O.Prod\_Num = P.Prod\_Num AND OrdLn\_DatePicked <= '2018-05-20'

GROUP BY P.Prod\_Num

ORDER BY P.Prod\_Num, sum(O.OrdLn\_Qnty);

**Output table:**



Table 7

**Query 8:** A list of products (show product number, description and price) whose price is less than or equal to the average product price.

SELECT Prod\_Num AS 'Product Number', Prod\_Desc 'Description', Prod\_Price AS '($) Price'

FROM PRODUCT

WHERE Prod\_Price <= (SELECT AVG(Prod\_Price) FROM PRODUCT);

**Output table:**

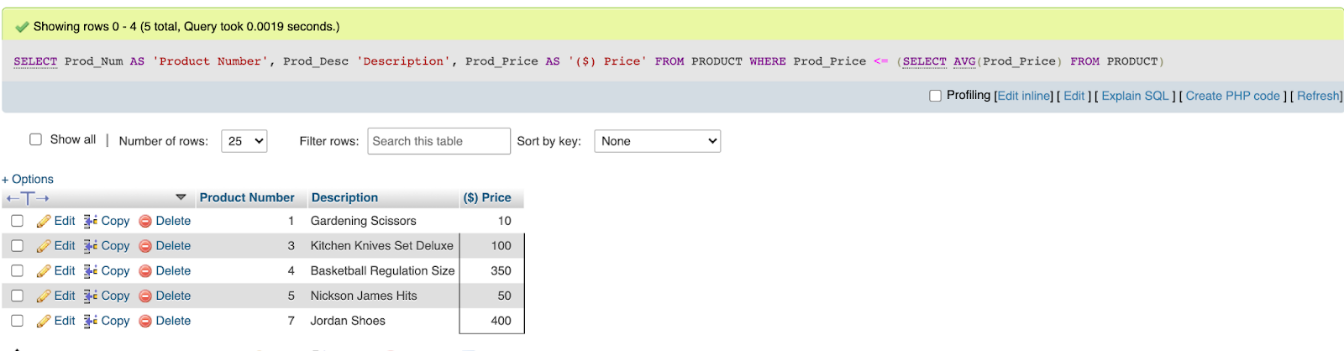


Table 8

**Query 9:**  Increase each employee’s salary by 7.5% and show the updated salary of all employees (name and salary).

UPDATE EMPLOYEE

SET Emp\_HourlySalary = Emp\_HourlySalary\*1.075;

SELECT CONCAT(Emp\_FName, ' ', Emp\_LName) AS 'Name', Emp\_HourlySalary AS 'Hourly Salary'

FROM EMPLOYEE;

**Output table:**

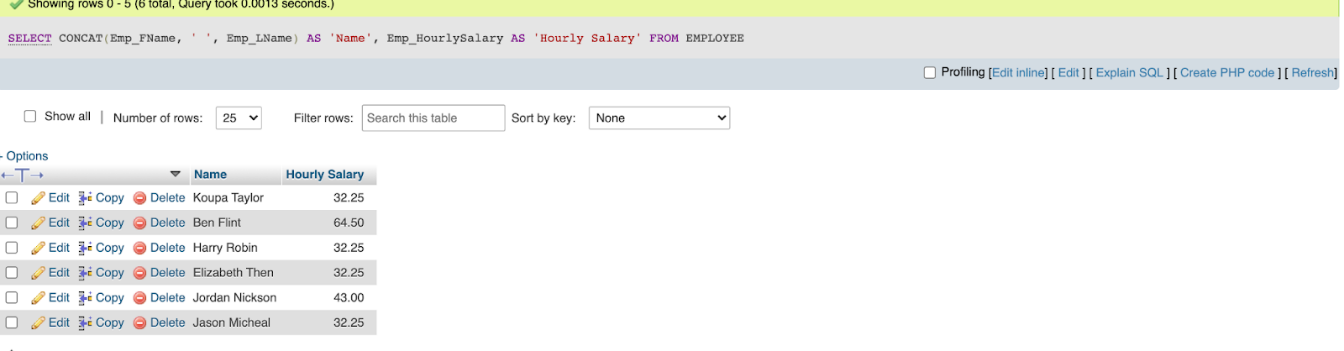


Table 9

**Query 10:**  Show the pay information (employee name, hours paid, amount paid) of all employees in the most recent pay date.

SELECT CONCAT(E.Emp\_FName, ' ', E.Emp\_LName) AS 'Name', CAST((P.Pay\_amount\_gross / E.Emp\_HourlySalary) AS INT) AS 'Hours Paid', P.Pay\_amount\_gross AS 'Total Amount Paid'

FROM EMPLOYEE AS E, PAYSLIP AS P

WHERE P.Emp\_ID = E.Emp\_ID

AND P.Pay\_date = (SELECT Pay\_date

                  FROM PAYSLIP

                  ORDER BY Pay\_date DESC

                  limit 0, 1);

**Output table:**

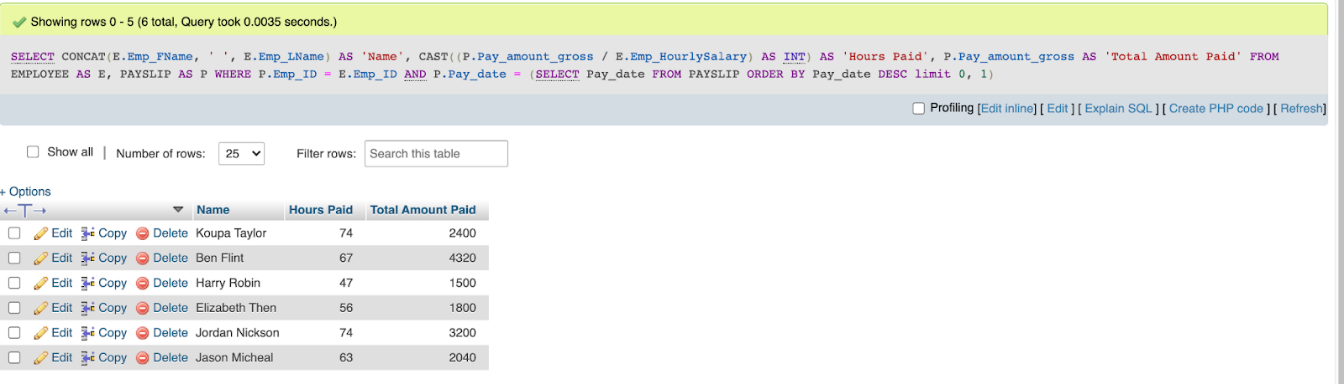


Table 10

# **Task 4: Inserting Additional Data**

USE bigm\_s5213262;

INSERT INTO CUSTOMER VALUES

(NULL, 'Daniel', 'Ortega', '0431xxx668');

INSERT INTO CUSTOMERORDER VALUES

(NULL, '2018-09-06', (SELECT Cust\_Number FROM CUSTOMER ORDER BY Cust\_Number DESC limit 0, 1), 2);

INSERT INTO ORDERLINE VALUES

((SELECT CustOrd\_ID FROM CUSTOMERORDER ORDER BY CustOrd\_ID DESC limit 0, 1), 2, '2018-09-08', '2018-09-10', 2);

UPDATE INVENTORY

SET Inv\_QtyOnHand = Inv\_QtyOnHand - (SELECT OrdLn\_Qnty FROM ORDERLINE WHERE CustOrd\_ID = (SELECT CustOrd\_ID FROM CUSTOMERORDER ORDER BY CustOrd\_ID DESC limit 0, 1) AND Prod\_Num = 2)

WHERE ProductNum = 2

AND Str\_Num = 2;

UPDATE INVENTORY

SET Inv\_QtyOrdered = Inv\_QtyOrdered + (SELECT OrdLn\_Qnty FROM ORDERLINE WHERE CustOrd\_ID = (SELECT CustOrd\_ID FROM CUSTOMERORDER ORDER BY CustOrd\_ID DESC limit 0, 1) AND Prod\_Num = 2)

WHERE ProductNum = 2

AND Str\_Num = 2;