**Q9. Stored Procedures in SQL with parameters**

a. Create a stored procedure Get\_country\_payments which takes in year and country as inputs and gives year wise, country wise total amount as an output. Format the total amount to nearest thousand unit (K)

Tables: Customers, Payments

CREATE DEFINER=`root`@`localhost` PROCEDURE `Get\_country\_payments`(

IN in\_year INT,

IN in\_country VARCHAR(50)

)

BEGIN

SELECT

in\_year AS Year,

in\_country AS Country,

CONCAT(ROUND(SUM(p.amount) / 1000), 'K') AS Total\_Amount

FROM

customers c

JOIN

payments p ON c.customerNumber = p.customerNumber

WHERE

YEAR(p.paymentDate) = in\_year

AND c.country = in\_country;

END

**Q12. ERROR HANDLING in SQL**

Create the table Emp\_EH. Below are its fields.

* EmpID (Primary Key)
* EmpName
* EmailAddress

Create a procedure to accept the values for the columns in Emp\_EH. Handle the error using exception handling concept. Show the message as “Error occurred” in case of anything wrong.

CREATE DEFINER=`root`@`localhost` PROCEDURE `InsertIntoEmp\_EH`(

IN p\_EmpID INT,

IN p\_EmpName VARCHAR(100),

IN p\_EmailAddress VARCHAR(100)

)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

SELECT 'Error occurred' AS Message;

END;

INSERT INTO Emp\_EH (EmpID, EmpName, EmailAddress)

VALUES (p\_EmpID, p\_EmpName, p\_EmailAddress);

SELECT 'Insert successful' AS Message;

END

**Q13. TRIGGERS**

Create the table Emp\_BIT. Add below fields in it.

* Name
* Occupation
* Working\_date
* Working\_hours

CREATE DEFINER=`root`@`localhost` TRIGGER `emp\_bit\_BEFORE\_INSERT` BEFORE INSERT ON `emp\_bit` FOR EACH ROW BEGIN

IF NEW.Working\_hours < 0 THEN

SET NEW.Working\_hours = ABS(NEW.Working\_hours);

END IF;

END